Calcul formel
et
Mathématiques
avec
la HP49G
en mode algébrique

Renée De Graeve
Maître de Conférence à Grenoble I

Remerciements

Je remercie :
- Bernard Parisse pour ses précieux conseils et ses remarques sur ce texte,
- Sylvain Daudé pour sa relecture,
- Jean Tavenas pour l'intérêt porté à l'achèvement de ce guide,
- les élèves de Terminale du lycée Notre-Dame des Victoires de Voiron, ainsi que leur professeur Jean Marc Paucod, pour leur participation au test du sujet de bac avec la HP49.

Ce document est disponible à l'adresse Internet suivante :
http ://www-fourier.ujf-grenoble.fr/~parisse/hp49.pdf

Préface

On me pose parfois la question : pourquoi mettre du calcul formel dans une calculatrice, alors que les logiciels spécialisés sur ordinateur sont maintenant bon marché, voire gratuits ?
Selon moi, la calculatrice est l'instrument le mieux adapté à l' *intégration* des outils de calcul dans l'enseignement des mathématiques puisque vous pouvez l'emporter facilement avec vous et l'utiliser pendant une séance de travaux dirigés ou un cours.

Mais l'utilisation d'un logiciel de calcul formel n'est pas aussi simple que l'interface pourrait le laisser croire... Il est donc important de disposer de la documentation adéquate. Le manuel de la HP49G décrit assez brièvement le logiciel, aussi ce guide en est le complément indispensable : il présente la HP49G dans la perspective de quelqu'un qui souhaite faire des mathématiques.

Le lecteur qui s'intéresse aux mathématiques peut très bien ne lire que ce texte puisque l'auteur commence par une présentation de la machine et détaille ensuite les commandes de calcul formel classées par thème (l'index permet de retrouver les commandes par ordre alphabétique) puis la programmation en mode algébrique. Chaque commande est illustrée par un exemple et certaines sont mises en pratique dans la résolution d'un sujet de bac. La partie programmation comporte de nombreux programmes, en particulier d'arithmétique.

En résumé, ceci est le guide que j'aurais dû écrire si j'en avais eu la patience ! Je remercie Renée de l'avoir réalisé...

Bernard Parisse
Maître de Conférences à l'Université de Grenoble I
Développeur du logiciel de calcul formel de la HP49G

Pour commencer

0.1 Présentation générale

0.1.1 Mise en route

Appuyer sur la touche `ON`.
En cours de travail, cette touche `ON` permet de sortir d'une application : elle joue le rôle de EXIT ou de CANCEL.
Pour éteindre la calculatrice, taper `shift-rouge` puis sur `ON`.
Si malgré plusieurs `ON` (`CANCEL`), la calculatrice ne repond pas, appuyer simultanément sur `ON` et `F3` pour la réinitialiser.

0.1.2 Que voit-on ?

De haut en bas :
1. l'écran
1.a l'état de la calculatrice
1.b l'historique des calculs
1.c un bandeau contenant des commandes
2. le clavier

1.L'écran :
1.a L'état de la calculatrice décrit les modes mis en œuvre :
- `RAD` ou `DEG` selon que l'on travaille en radians ou en degrés.
- `XYZ` pour indiquer que l'on travaille en coordonnées rectangulaires.
- `HEX` pour indiquer que les entiers binaires précédés de `#` sont écrits en base 16.
- `R` ou `C` selon que l'on travaille en mode RÉEL ou en mode COMPLEXE.
- `=` ou $\sim$ selon que l'on travaille en mode EXACT (calcul formel) ou en mode APPROCHÉ (calcul numérique).
- `'X'` indique le nom de la variable courante contenu dans `VX` : en général c'est `'X'`
- `ALG` ou `RPN` selon que l'on travaille en mode ALGÉBRIQUE ou en mode RPN.
- `{HOME}` ou `{HOME ESSAI}` pour indiquer le nom du répertoire dans lequel on se trouve (par exemple le répertoire principale HOME ou le sous répertoire ESSAI).

1.b L'historique des calculs
Principe : sur l'écran, le calcul demandé (précédé de `:`) s'inscrit à gauche et le résultat s'inscrit à droite.

1.c Le bandeau :
Les commandes du bandeau sont accessibles par les touches :
`F1 F2 F3 F4 F5 F6`.
Lorsque le bandeau comporte plus de 6 commandes, la suite du bandeau est visible lorsqu'on appuie sur la touche `NXT`. Le bandeau peut contenir des répertoires (contenant un ensemble de commandes), ils sont repérables par leur forme de valise. Pour activer une commande du bandeau, il suffit de taper sur la touche `Fi` correspondante.
2. Le clavier :
Il faut repérer :

- la touche `ON` pour la mise en route ou pour arrêter un calcul en cours . Pour éteindre la calculatrice, taper `shift-rouge` puis sur `ON`.
- les deux touches "shift", une bleue et une rouge qui permettent à une même touche d'avoir plusieurs fonctions.
- la touche `ALPHA` pour taper du texte (en majuscules par défaut). Pour rester en mode de saisie alphabétique il faut appuyer deux fois sur la touche `ALPHA`. Pour sortir de ce mode taper à nouveau sur la touche `ALPHA`. Pour basculer entre l'écriture en majuscules ou en minuscules, taper sur `shift-bleu` puis sur la touche `ALPHA` (lorsqu'on est en mode de saisie alphabétique).
- la touche `ENTER` qui sert à valider une commande.
- les quatres flèches (gauche, droite, haut, bas) qui permettent de déplacer le curseur lorsqu'on est dans l'éditeur ou dans un menu.

0.2 Les différents modes

Cette calculatrice permet de travailler dans différents modes.
On peut choisir :
-le mode algébrique ou écriture polonaise inversée (`ALG ou RPN`)
-le mode réel ou le mode complexe (`R` ou `C`)
-le mode exact ou le mode approximatif (= ou ∼)
-le mode direct ou le mode pas a pas...
ATTENTION : ce qui suit suppose que la calculatrice est dans le mode algébrique réel exact direct (`R = ALG`).
Tapez : `CASCFG` (Computer Algebra System ConFiG) pour mettre la calculatrice en mode réel exact direct. Au cours de votre travail, il est conseillé de taper `CASCFG` pour se remettre dans cette configuration (en effet la calculatrice change -en vous demandant l'autorisation- de mode quand c'est nécessaire !).
VÉRIFICATION
Vérifiez maintenant que la calculatrice est bien en mode algébrique réel exact.

Pour cela faire :

touche `MODE` puis vérifier que l'operating mode est bien `algebraic` sinon choisir `algebraic` à l'aide de `choos` du bandeau ou taper sur la touche $^+/_-$.

Vérifier, pendant que vous êtes dans `MODE`, à l'aide de `cas` du bandeau que ni `numeric`, ni `approx`, ni `complex` ne sont cochés (sinon enlever la croix à l'aide de `chk` du bandeau).
Remarque : pour des applications pédagogiques, il est souvent intéressant de cocher `step/step` pour que la calculatrice fasse les calculs en pas à pas.

Puis `ok` du bandeau pour valider le choix fait dans `cas` puis `ok` du bandeau pour valider le choix fait dans `MODE`.

On est alors en mode algébrique réel exact.

ATTENTION :
CE QUI SUIT SUPPOSE QUE LA CALCULATRICE EST DANS CET ÉTAT .

Vous êtes à nouveau dans le répertoire `HOME`.

Il suffit maintenant de taper les calculs à effectuer par exemple :

$1+1$ suivi de `ENTER`

Le résultat s'affiche (à droite) alors que l'expression $1+1$ précédée de : remonte dans l'historique (à gauche).

On pourra ainsi recopier cette expression dans la ligne de commande en appuyant sur la touche `HIST` (la flèche vers le haut permet de sélectionner l'expression et `echo` du bandeau de la recopier et de la simplifier).

On peut aussi utiliser le dernier résultat (noté `ANS(1)`) grâce à la touche `ANS` (`shift-bleu ENTER`) et aussi les résultats précédents (notés `ANS(2)`...).

Vous avez la possibilité de faire soit des calculs exacts, soit des calculs approchés par exemple : $\sqrt{2}$ suivi de `ENTER` n'évalue pas $\sqrt{2}$ et effectue des calculs exacts mais suivi de `shift-rouge ENTER (→NUM)` donne une approximation de $\sqrt{2}$ avec 12 chiffres significatifs, tout en restant en mode exact.

Bien sûr, si vous ne voulez faire que du calcul numérique il suffit de cocher `approx` (`MODE` puis `cas` du bandeau), dans ce cas, la touche `ENTER` effectue le calcul numérique en évaluant les constantes et les variables.

0.3 Notations

Les quatre flèches de direction du curseur sont ici représentées par les quatre triangles :

$$\triangle \ \triangleleft \ \triangleright \ \triangledown$$

La flèche d'effacement (effacement du caractère se trouvant avant le curseur) est représentée par :

$$\Leftarrow$$

La flèche rouge au dessus du 0 est représentée par :

$$\rightarrow$$

La touche `STO` est représentée dans un programme par :

`STO`$\triangleright$ ou $\triangleright$

Le retour à la ligne (en rouge au dessus du point) est representé par :

$$\hookleftarrow$$

0.4 Les flags

La plupart des commandes tiennent compte des indicateurs (flags) du système.

Chaque flag est repéré par un numéro et a une valeur par défaut.
Si on veut changer la valeur d'un flag, on peut le faire en tapant sur la touche `MODE`, puis sur `F1` pour **`flags`** du bandeau, on accède ainsi au gestionnaire des flags. On coche le flag que l'on veut changer et sa nouvelle fonction apparaît.
Quand on connait le numéro d'un flag, on peut aussi changer sa valeur à l'aide des commandes `SF` ou `CF`. Par exemple pour changer le flag de numéro 117 (c'est le flag qui gére l'affichage des menus) on tape :
`SF(-117)` (les menus se trouvent alors inscrits dans le bandeau) et alors :
`FS?(-117)` est égal à `1.` et `FC?(-117)` est égal à `0.`.
Pour avoir à nouveau des menus déroulants il suffit de taper :
`CF(-117)` (`FS?(-117)` est alors égal à `0.` et `FC?(-117)` à `1.`).

Chapitre 1

Touches importantes

1.1 La touche APPS

Cette touche ouvre le menu des différentes applications.

1.1.1 Plot functions

On trouve :
`Equation entry`. Ce menu est identique à la suite de touches `shift-bleu F1 (Y=)`.
`Plot window`. Ce menu est identique à la suite de touches `shift-bleu F2 (WIN)`.
`Graph display`. Ce menu est identique à la suite de touches `shift-bleu F3 (GRAPH)`.
`Plot setup`. Ce menu est identique à la suite de touches `shift-bleu F4 (2D/3D)`.
`Table setup`. Ce menu est identique à la suite de touches `shift-bleu F5 (TBLSET)`.
`Table display`. Ce menu est identique à la suite de touches `shift-bleu F6 (TABLE)`.
On se reportera au chapitre 3 pour avoir plus de détails

1.1.2 I/O functions

Ce sont les fonctions qui permettent de faire dialoguer votre calculatrice avec votre ordinateur.
Par exemple on trouve en 5 : `Transfer`.
Si on tape sur 5 puis `ok` du bandeau on ouvre la fenêtre `Transfer` :
Port : Wire
Type : Kermit (ou XModem)
Par exemple, voilà comment on utilise le programme Kermit sous Linux :
-On branche la calculatrice au cordon de transfert.
-Sur l'ordinateur on tape :
`kermit`
puis `serv`
-Sur la `HP49G` on tape :
`SEND('NOM')`

pour que la variable de nom `NOM` qui se trouve sur votre `HP49G` soit recopiée sur votre ordinateur.
-Ou
Sur la `HP49G` on tape :
`KGET('NOM')`
pour que la variable de nom `NOM` qui se trouve sur votre ordinateur soit recopiée sur votre `HP49G`.

1.1.3 Constants library

Cela ouvre une liste de 40 constantes de la physique.
Ces constantes sont définies par leurs abréviations et leurs noms ou leurs valeurs (si `value` du bandeau est coché).
Elles sont suivies de leurs unités si `unit` du bandeau est coché.
Elles peuvent être recopiées dans la ligne de commande quand on appuie sur `->stk` du bandeau.

1.1.4 Numeric solver

Ce menu est identique au menu obtenu à partir de la suite de touches `shift-rouge 7 (NUM.SLV)`.

1.1.5 Time & date

Ce menu est identique au menu obtenu à partir de la suite de touches `shift-rouge 9 (TIME)`.

1.1.6 Equation writer

Ce menu est identique au menu obtenu à partir de la touche `EQW`.
Pour plus de détails, on se reportera à la section 2.1.

1.1.7 File manager

Ce menu est identique au menu obtenu à partir de la suite de touches `shift-bleu APPS (FILES)`.
Pour plus de détails, on se reportera à la section 2.5.

1.1.8 Matrix writer

Ce menu est identique au menu obtenu à partir de la suite de touches `shift-bleu EQW (MTRW)`.
Pour plus de détails, on se reportera à la section 2.2.

1.1.9 Text editor

Cela ouvre la ligne de commande : il faut remarquer que ce que l'on écrit peut s'écrire sur plusieurs lignes (lorsqu'on tape sur $\mathtt{shift}-\mathtt{rouge}\ \bullet\ (\hookleftarrow)$).

1.1.10 Math menu

Ce menu est identique au menu obtenu à partir de la suite de touches `shift-bleu SYMB (MTH)`.

1.1.11 CAS menu

On trouve :
1.`ARITHMETIC` correspondant au menu de `shift-bleu 1 (ARIT)`
2.`ALGEBRA` correspondant au menu de `shift-rouge 4 (ALG)`
3.`COMPLEX` correspondant au menu de `shift-rouge 1 (CMPLX)`
4.`CALCULUS` correspondant au menu de `shift-bleu 4 (CALC)`
5.`EXP&LN` correspondant au menu de `shift-bleu 8 (EXP&LN)`
6.`SYMBOLIC SOLVER` correspondant au menu de `shift-bleu 7 (S.SLV)`
7.`MATRICES` correspondant au menu de `shift-bleu 5 (MATRICES)`
8.`CONVERT` correspondant au menu de `shift-bleu 6 (CONVERT)`
9.`TRIGONOMETRIC` correspondant au menu de `shift-rouge 8 (TRIG)`
Pour plus de détails, on se reportera au chapitre 4.

1.2 La touche MODE

Cette touche permet de régler le mode de fonctionnement de votre calculatrice : mode `Algeraic` ou `RPN`, de régler les `flags` (touche `F1`), de régler le fonctionnement du `cas` (touche `F3`) et de régler la taille de l'affichage avec `disp` (touche `F4`).
Par exemple (cf page 8) le `flag 117` peut être :
`choose boxes` pour avoir des menus déroulants
ou
`soft menu` pour avoir les menus dans le bandeau.

1.3 La touche TOOL

Cette touche fait apparaître un bandeau contenant :
`edit` pour éditer la première ligne (ou la ligne mise en surbrillance).
`view` pour visualiser la première ligne (ou la ligne mise en surbrillance).
`rcl` identique à la suite de touches `shift − bleu STO` ▷ (`RCL`) (cf page 20).
`sto` ▷ identique à la touche `STO`▷.
`purge` identique à la commande `PURGE` (cf page 21).
`clear` efface la ligne de commande en cours en laissant le curseur en début de ligne (n'est pas identique à `CANCEL` qui annule la ligne de commande en cours!!!).
Attention, `clear` efface tout l'historique en l'absence de ligne de commande et est alors identique à `shift − rouge` ← (`CLEAR`).

1.4 La touche UNDO (shift-rouge HIST)

Cette touche est très pratique puisqu'elle permet d'annuler la dernière commande.

1.5 La touche VAR

Cette touche fait apparaître un bandeau contenant le nom de toutes les variables utilisées (appuyer sur `NXT` pour tout voir !!!).
Pour plus de détails, on se reportera à la section 2.4.

1.6 La touche EQW

Elle permet d'ouvrir l'éditeur d'équations.
Cette touche peut être utilisée à tout moment même à l'intérieur de l'éditeur de matrices.
On peut aussi accéder à l'historique depuis l'éditeur d'équations (cf 1.11). Pour plus de détails, on se reportera à la section 2.1.

1.7 La touche MTRW (shift-bleu EQW)

Elle permet d'ouvrir l'éditeur de matrices pour éditer des tableaux. Si vous voulez entrer un vecteur, veillez à ce que `vect` du bandeau soit coché.
Pour écrire une matrice :
On édite la première ligne, puis on fait revenir le curseur au début de la deuxième ligne, puis on écrit les lignes suivantes, le curseur se met automatiquement au début des autres lignes.
Pour plus de détails, on se reportera à la section 2.2.

1.8 La touche SYMB

Cela ouvre le menu des fonctions symboliques de base classées par thème.
Les différents sous-menus contiennent les fonctions du `cas` utiles à un élève de terminale. On retrouve ces fonctions (et d'autres !) dans les menus correspondants du `cas`.
Exemple :
Le `SYMBOLIC ARITH MENU` est une partie du sous-menu `INTEGER` du menu `ARITH` `(shift-bleu 1)`.

1.9 La touche MTH (shift-bleu SYMB)

Cela ouvre le menu des fonctions mathématiques.
On notera :
Les fonctions hyperboliques (sous -menu 4) comme :
`SINH ASINH COSH ACOSH TANH ATANH`
Les fonctions :
`EXPM(X)=EXP(X)-1 LNP1(X)=LN(X+1)`
et les fonctions utiles pour les réels (sous -menu 5) comme :
`FLOOR(X)` qui donne la partie entière de `X`.
`CEIL(X)` qui donne la partie entière de `X+1` si `X` n'est pas entier et `X` sinon.
`RND(X,n)` qui arrondit `X` avec `n` décimales.
`TRNC(X,n)` qui tronque `X` avec `n` décimales.

1.10 La touche UNITS (shift-rouge 6)

Le menu `UNITS` contient 127 unités classées par catégories.
Pour utiliser des unités il faut écrire l'unité précedée de `_` (`shift-rouge -`).
On peut faire des changements d'unités grâce à la fonction `CONVERT` (qui se trouve dans le sous-menu `Tools` du menu `UNITS`).
Exemple :
On tape :
`CONVERT(12_cm,1_m)`
On obtient :
`0.12_m`

1.11 La touche HIST

Cette touche permet d'accéder à l'historique lorsqu'on est en train de taper une commande. Elle permet aussi d'y accéder depuis l'éditeur d'équations ou de matrices.
Il faut savoir que ce que l'on recopie est recopié ET évalué.
Si on veut réutiliser un résultat sans qu'il soit évalué, il faut utiliser :
`ANS(1)` ou `ANS(2)`...(`shift-bleu ENTER (ANS(1)`).
Si on veut réutiliser une commande, on peut aussi utiliser `shift-bleu HIST (CMD)` qui donne la liste des dernières commandes utilisées.

Chapitre 2

Saisie

2.1 L'éditeur d'équations

2.1.1 Accès à l'equationwriter

La touche `EQW` (pour EQuationWriter) vous permet d'entrer dans l'éditeur d'équations, à tout moment, lors de la saisie de la ligne de commande.
C'est un éditeur très performant pour écrire, simplifier et travailler sur des expressions mathématiques.

Lorsque l'on est dans l'éditeur d'équations on peut taper des expressions en sachant que l'opérateur que l'on utilise porte toujours sur l'expression adjacente ou sur l'expression sélectionnée. On ne se préoccupe pas de mettre des parenthèses, on sélectionne ! ! !
Il faut voir les expressions mathématiques comme un arbre (pas forcément binaire) et comprendre que les quatre flèches permettent de parcourir l'arbre de façon naturelle (les flèches droite et gauche permettent d'aller d'un sous-arbre à l'autre, les flèches haut et bas de monter ou de descendre dans l'arbre, les flèches droite et gauche "shiftées" permettent diverses sélections (cf page 16 l'exemple 2)).

2.1.2 Comment sélectionner ?

On peut entrer dans le mode sélection de deux façons

- La flèche $\triangle$ vous fait entrer dans le mode sélection et sélectionne l'élément adjacent.
- La flèche $\triangleright$ vous fait entrer dans le mode sélection et sélectionne le sous-arbre adjacent.
- Attention : si on est en train de taper une fonction ayant plusieurs arguments (comme par exemple une $\sum$ ou une $\int$, la flèche $\triangleright$ permet de progresser dans l'écriture, en changeant le curseur d'emplacement (la flèche $\triangleright$ permet le passage d'un argument à l'autre)). Il faut donc toujours dans ce cas sélectionner avec la flèche $\triangle$ (cf 2.1.4)

Exemples de fonctionnement de cet éditeur :

- Exemple 1
  On tape :

$$\texttt{2} + \texttt{X} * \texttt{3} - \texttt{X}$$

et on obtient

$$2 + \mathtt{X} \cdot 3 - \mathtt{X}$$

`ENTER ENTER` donne le résultat

$$2 + 2 \cdot \mathtt{X}$$

On tape :

$$2 \ + \ \mathtt{X} \ \triangleright \ * \, 3 \ - \ \mathtt{X}$$

et on obtient

$$(2 + \mathtt{X}) \cdot 3 - \mathtt{X}$$

`ENTER ENTER` donne le résultat

$$6 + 2 \cdot \mathtt{X}$$

On tape :

$$2 \ + \ \mathtt{X} \ \triangleright \ * \, 3 \, \triangle \ - \ \mathtt{X}$$

et on obtient

$$(2 + \mathtt{X}) \cdot (3 - \mathtt{X})$$

`ENTER ENTER` donne le résultat

$$-(\mathtt{X}^2 - \mathtt{X} - 6)$$

- Exemple 2
  Si on veut taper :
  $$\mathtt{X}^2 - 3 \cdot \mathtt{X} + 1$$
  On tape :
  $$\mathtt{X} \ \mathtt{y}^\mathtt{x} \ 2 \ \triangleright \ - \ 3 \ \mathtt{X} \ \triangleright \ \triangleright \ + \ 1$$
  En effet il faut sélectionner $-3 \cdot \mathtt{X}$ avant de taper $+\ 1$
- Exemple 3
  Si on veut taper :
  $$\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5}$$
  Ici, le sommet de l'arbre est un $+$ et il y a 4 sous arbres ; chacun de ces sous-arbres a comme sommet un $\div$ et possède deux feuilles.
  On tape tout d'abord `EQW`, puis le premier sous -arbre :
  $$1 \div 2$$
  puis on sélectionne cet arbre avec
  $$\triangleright$$
  puis on tape
  $$+$$
  et le second sous-arbre :
  $$1 \div 3$$
  puis on sélectionne cet arbre avec
  $$\triangleright$$

puis on tape

$$+$$

et le troisième sous-arbre :

$$\mathtt{1} \div \mathtt{4}$$

puis on sélectionne cet arbre avec

$$\triangleright$$

puis on tape

$$+$$

et le quatrième sous-arbre :

$$\mathtt{1} \div \mathtt{5}$$

puis on sélectionne cet arbre avec

$$\triangleright$$

Maintenant, l'expression voulue

$$\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5}$$

se trouve ecrite dans l'equationwriter et $\frac{1}{5}$ est sélectionnée.
Parcourez l'arbre pour sélectionner

$$\frac{1}{3} + \frac{1}{4}$$

Il faut taper

$$\triangleleft$$

pour sélectionner $\frac{1}{4}$ puis

$$\mathtt{shift} - \mathtt{rouge}\triangleleft$$

permet de sélectionner deux sous-arbres contigus ici

$$\frac{1}{3} + \frac{1}{4}$$

Intérêt : On peut demander d'effectuer le calcul de la partie sélectionnée en faisant

$$\mathtt{shift} - \mathtt{rouge}\ \mathtt{SYMB}\ (\mathtt{EVAL})$$

On obtient :

$$\frac{1}{2} + \frac{7}{12} + \frac{1}{5}$$

Si on veut effectuer maintenant le calcul partiel

$$\frac{1}{2} + \frac{1}{5}$$

il faut tout d'abord faire une permutation pour que $\frac{1}{2}$ et $\frac{1}{5}$ soient côte à côte en tapant

$$\mathtt{shift} - \mathtt{bleu}\triangleleft$$

qui échange l'élément sélectionné avec son voisin de gauche.
On obtient

$$\frac{7}{12}+\frac{1}{2}+\frac{1}{5}$$

et $\frac{7}{12}$ est sélectionné, puis

$$\triangleright\texttt{shift}-\texttt{rouge}\triangleright$$

sélectionne

$$\frac{1}{2}+\frac{1}{5}$$

On peut alors faire à nouveau `EVAL`.

2.1.3 Comment modifier une expression

Pour remplacer la sélection par une expression, il suffit de taper l'expression.
Pour supprimer la sélection on tape :

$$\texttt{shift}-\texttt{rouge}\ \Leftarrow$$

Pour supprimer un opérateur unaire, sommet de l'arbre sélectionné, on tape :

$$\texttt{shift}-\texttt{bleu}\ \Leftarrow$$

par exemple pour remplacer $\sin(expr)$ par $\cos(expr)$, on efface sin (en sélectionnant $\sin(expr)$ puis $\texttt{shift}-\texttt{bleu}\ \Leftarrow$) puis on tape cos.
Pour supprimer un opérateur binaire, il faut utiliser `edit` du bandeau, corriger dans l'éditeur, et revenir à l'éditeur d'équations avec `ENTER`.
La touche `HIST` (utilisée depuis l'éditeur d'équations) permet de revenir à l'historique et de recopier un élément de l'historique avec `echo` du bandeau.

2.1.4 Comment écrire $\int$ et $\sum$

Pour entrer le signe $\int$ il suffit de taper

$$\texttt{shift}-\texttt{rouge}\ \texttt{TAN}\ (\int)$$

Pour entrer le signe $\sum$ il suffit de taper

$$\texttt{shift}-\texttt{rouge}\ \texttt{SIN}\ (\sum)$$

le curseur se place aux endroits voulus et se déplace à l'aide de

$$\triangleright$$

Les expressions que l'on rentre suivent la loi de la sélection expliquée précédemment, mais il faut entrer dans le mode sélection avec $\triangle$.

Attention, ne pas utiliser l'indice i pour définir la somme car i désigne le nombre complexe solution de $x^2+1=0$.

Il faut savoir que $\sum$ sait calculer symboliquement les sommes de fractions rationnelles et les series hypergéométriques qui admettent une primitive discrète (à partir de la ROM version 1.11).
En mode numérique $\sum$ effectue des calculs approchés (par exemple $\sum_{k=0}^{4}\frac{1}{k!}=2.70833333334$ alors que $1+\frac{1}{1!}+\frac{1}{2!}+\frac{1}{3!}+\frac{1}{4!}=\frac{65}{24}$) (le symbole ! s'obtient en tapant `alpha` $\texttt{shift}-\texttt{rouge}$ `2`).

2.1.5 Le mode curseur

Le mode curseur permet de sélectionner une grande expression rapidement : pour cela taper `shift-rouge EQW (')` pour passer en mode curseur puis, utiliser les flèches pour inclure votre sélection dans une boite puis, `ENTER` pour sélectionner le contenu de la boite, ou `CANCEL` si vous voulez annuler votre sélection.

2.2 L'éditeur de tableaux

Pour ouvrir l'éditeur de tableaux ou le matrix writer taper :
`shift-bleu EQW (MTRW)`.
Vous pouvez alors entrer les éléments de la première ligne en appuyant sur `ENTER` après chaque entrée (vous pouvez bien sûr vous servir de l'éditeur d'équations pour les écrire!!!), puis il faut ramener le curseur avec les flèches au début de la deuxième ligne etc...(le curseur revient ensuite automatiquement au début de la troisième ligne).
Si vous entrez un nombre négatif, par exemple -2, tapez $^+/_-$ 2.
Si vous voulez entrer un vecteur pensez à vérifier que `vect` du bandeau est coché.
A noter qu'en mode `Algébrique` on doit entrer les éléments un à un (il faut appuyer sur `ENTER` après chaque élément), alors qu'en mode `RPN` on peut écrire plusieurs éléments en les séparant par des espaces.

2.3 L'éditeur de texte

C'est la ligne qui s'ouvre sous l'historique pour taper une commande.
C'est un véritable éditeur de texte où l'on peut : sélectionner une expression (avec `BEGIN END`), la couper (`CUT`) ou la recopier dans le buffer (`COPY`), puis la recopier là où se trouve le curseur (`PASTE`).
Il faut noter que toutes ces commandes fonctionnent aussi dans `EQW` et `MTRW`.

2.3.1 BEGIN END

Mettre le curseur sur le premier élément du texte à sélectionner puis taper :
`shift-rouge APPS (BEGIN)`.
Puis déplacer le curseur sur l'élément suivant le dernier caractère et taper :
`shift-rouge MODE (END)`.
Votre sélection apparaît.

2.3.2 COPY

`shift-rouge VAR (COPY)` recopie la sélection dans le buffer.

2.3.3 CUT

`shift-rouge STO (CUT)` recopie la sélection dans le buffer et l'efface.

2.3.4 PASTE

shift-rouge NXT (PASTE) recopie la sélection là où se trouve le curseur (il faut avoir fait avant, soit COPY, soit CUT, pour que la sélection soit dans le buffer).

2.4 Les variables

Vous pouvez stocker des objets dans des variables, et les réutiliser en utilisant le nom de la variable.
Bien voir la différence entre A et 'A' :
A est évalué (désigne l'exécution du contenu) et 'A' n'est pas évalué (désigne le nom de la variable).
Par exemple :
STO(B,'A') : le contenu de B est mis dans A.
STO('B','A') signifie qu'à tout moment B et A ont même contenu.
VAR affiche un bandeau qui contient toutes les variables que vous avez définies ainsi que les sous-répertoires (ils se différencient des variables, dans le bandeau, par leur forme de "valise"). Il faut savoir que shift-bleu APPS (FILES) affiche l'arborescence des variables de HOME ainsi que la mémoire d'archive et facilite la gestion des variables.

2.4.1 STO

STO permet de créer une variable et de stocker un objet dans cette variable.
ATTENTION STO est préfixé si on tape la commande en mode alpha, et est infixée si on utilise la touche STO (notée dans la suite STO▷ ou ▷).
Exemples :
On tape :
STO(1,'A')
ou
on utilise la touche STO▷ qui se traduit à l'écran par ▷ :
on tape :
1 STO ▷ A (1 ▷ A).
On remarque qu'ici les ' ' autour de A sont inutiles.
La variable A est alors créée et cette variable contient 1.
On tape :
≪ 12 ≫ STO ▷ P
P est une variable contenant le programme ≪ 12 ≫ qui affiche 12.

2.4.2 RCL

RCL a comme paramètre le nom de la variable entouré de ' et permet d'afficher le contenu de cette variable.
Pour récupérer le contenu d'une variable, il suffit de taper le nom de cette variable, SAUF, si cette variable contient un programme (car alors le programme est exécuté).
Dans l'exemple précédent :
A affiche 1 et P affiche 12

alors que :
RCL('A') affiche 1 et RCL('P') affiche ≪ 12 ≫ .

2.4.3 PURGE

PURGE permet d'effacer le nom de la variable et son contenu.
On trouve PURGE dans le menu TOOL
On tape :
PURGE('A')

2.4.4 Les variables prédéfinies

Le nom de la variable symbolique courante se trouve dans VX (ce sera en général X, il ne faut donc pas utiliser X comme nom de variable ou effacer le contenu de X avant de faire du calcul symbolique.
EPS contient la valeur de epsilon utilisé dans la commande EPSX0 (cf 4.20.1).
EQ contient l'équation du dernier graphe réalisé.
MATRIX contient la matrice utilisée comme argument de JORDAN,EGV ou EGVL.
MODULO contient la valeur de p quand on fait du calcul symbolique dans $Z/p.Z$.
PERIOD doit contenir la période de la fonction dont on veut les coefficients de Fourier (cf 4.7.16).
PRIMIT contient la primitive de la dernière fonction intégrée.
REALASSUME contient le nom des variables symboliques que l'on considère comme réelles (par défaut X, t et toutes les variables d'intégration utilisées).
SYSTEM contient le système utilisé comme argument de rref ou RREF si ce système a au moins un paramètre.

2.5 Les répertoires

Au début, vous n'avez que le répertoire HOME qui sera la racine principale de vos répertoires futurs (répertoire père).

2.5.1 Création d'un répertoire

Taper shift-bleu APPS (FILES) qui affiche la structure en arbre de vos répertoires.
Sélectionner le répertoire que vous choisissez commme père (par exemple HOME au début) puis ok du bandeau.
Un bandeau contenant edit copy move... s'affiche, faire NXT et sélectionner avec F3 new (new variable or directory).
Ne pas remplir Object mais Name (il suffit de taper le nom que vous avez choisi puis ok du bandeau).
Puis cocher Directory avec F3 (chk), puis ok du bandeau.
Puis CANCEL pour revenir dans HOME.
Vérifier en tapant sur VAR que votre répertoire a bien été créé.
On peut aussi créer un répertoire grâce à la commande CRDIR.
On se met dans le répertoire qui va être le père et on tape :
CRDIR('NOMREP')
Un sous-répertoire de nom NOMREP est alors créé.

2.5.2 Travailler dans un répertoire

Travailler dans un répertoire est facile : il suffit d'appuyer sur `VAR` pour faire apparaître le nom des sous répertoires dans le bandeau, puis d'ouvrir le répertoire voulu en appuyant sur la touche `Fi` correspondant à son nom, puis `ENTER`.
Pour remonter dans l'arbre des répertoires il suffit de taper :
`shift-bleu VAR (UPDIR)`

2.5.3 Effacer, renommer, déplacer un répertoire

Taper `shift-bleu APPS (FILES)` qui affiche la structure en arbre de vos répertoires.
Sélectionner le répertoire que vous voulez effacer, renommer, déplacer, puis `ok` du bandeau.
Un bandeau contenant `edit copy move...purge rename...` s'affiche.
`purge` efface ce répertoire à condition qu'il soit vide.
`rename` lui donne un autre nom.
`copy` le copie (on utilise les flèches pour désigner la destination puis `ok`).
`move` le déplace (on utilise les flèches pour désigner la destination puis `ok`).

Chapitre 3

Graphique

3.1 Les différentes fenêtres

3.1.1 Equation entry

Cette fenêtre est obtenue avec la suite de touches :
`shift-bleu F1 (Y=)`. Elle permet de définir l'équation du graphique.

3.1.2 Plot window

Cette fenêtre est obtenue avec la suite de touches :
`shift-bleu F2 (WIN)`.
Elle permet de définir la fenêtre de visualisation et de donner les bornes entre lesquelles on veut faire varier le paramètre indépendant.
Si le paramètre indépendant est à `Default` cela veut dire qu'il varie comme le paramètre horizontal de la fenêtre.
Pour remettre un paramètre à `Default` il faut taper sur `NXT` puis sur `reset` du bandeau.

3.1.3 Graph display

Cette fenêtre est obtenue avec la suite de touches :
`shift-bleu F3 (GRAPH)`.
Elle permet d'obtenir le graphe quand tous les paramètres ont été choisis.

3.1.4 Plot setup

Cette fenêtre est obtenue avec la suite de touches :
`shift-bleu F4 (2D/3D)`.
Elle permet de définir le type du graphique, l'équation et les variables.

3.1.5 Table setup

Cette fenêtre est obtenue avec la suite de touches :
`shift-bleu F5 (TBLSET)`.
Elle permet d'initialiser un tableau de valeurs.

3.1.6 Table display

Cette fenêtre est obtenue avec la suite de touches :
`shift-bleu F6 (TABLE)`.
Elle donne le tableau de valeurs qui a été initialisé par `TBLSET`.

3.2 Les différents champs à définir

3.2.1 Le type

Le type de tracé que l'on veut peut être choisi grâce au `choos` du bandeau de la fenêtre `PLOT SETUP (shift-bleu F4 (2D/3D))`.
On détaillera ici les tracés les plus utilisés comme :
`Function` pour tracer le graphe de fonctions en coordonnées cartésiennes.
`Polar` pour tracer des courbes en coordonnées polaires.
`Parametric` pour tracer des courbes en coordonnées paramétriques.
`Truth` pour tracer les solutions d'inéquations (le pixel (x,y) est allumé si `EQ` est vraie).
`Diff Eq` pour tracer les solutions de l'équation différentielle $y' = f(x, y)$.
On peut tracer les solutions sur l'intervalle $[\,a, b]$ vérifiant $y(x_0) = y_0$.
Pour cela, on met dans `H-View` les valeurs de a et b puis x_0 dans `Init` et y_0 dans `Init-Soln`.
Puis on fait le tracé en deux temps : on met `Final` à b pour avoir la solution sur $[x_0, b]$, on trace, puis on met `Final` à a pour avoir la solution sur $[\,a, x_0]$ et on trace.
`Slopefield` pour tracer le champ des tangentes d'une équation différentielle de la forme $y' = f(x, y)$.
`Fast3D` pour tracer une surface définie par $z = f(x, y)$.
On peut ensuite faire tourner le repère à l'aide des touches `NXT`, `TOOL` et des flèches $\triangle$ $\triangleleft$ $\triangleright$ $\triangledown$... ce qui permet d'avoir une bonne vision de la surface.

3.2.2 L'équation

L'équation peut être entrée de plusieurs façons :
-on peut la stocker dans la variable `EQ`.
-on peut la taper dans la fenêtre ouverte avec `shift-bleu F1 (Y=)`.
-on peut la taper dans le champ `EQ` de la fenêtre `PLOT SETUP` ouverte avec `shift-bleu F4 (2D/3D)`.
-on peut aussi utiliser la fonction du `cas PLOT` qui prend comme argument une équation, la stocke dans `EQ` et ouvre la fenêtre `PLOT SETUP`.
Il faut noter que `EQ` peut être une liste d'équations, dans ce cas on aura sur le même graphe les courbes correspondant aux éléments de la liste.
On peut aussi grâce à la fonction du `cas PLOTADD` rajouter une équation à la liste d'équations contenue dans `EQ`.

3.2.3 Variable indépendante et forme de l'équation

La forme de l'équation depend du type de graphique choisi et du choix de la variable indépendante.
Selon les cas, on tape une équation de la forme :

`f(x)` pour le tracé du graphe de $y = f(x)$ en coordonnées cartésiennes, si `x` est la variable indépendante et si le type est `Function`.
`f(t)` pour le tracé en coordonnées polaires de la courbe $r = f(t)$, si `t` est la variable indépendante et si le type est `Polar`.
`x(t) + i.y(t)` pour tracer la courbe $(x = x(t), y = y(t))$ en coordonnées paramétriques, si `t` est la variable indépendante et si le type est `Parametric`.
`f(x, y) > 0` pour hachurer la zone correspondante, si `x` est la variable indépendante et si le type est `Truth`.
`f(t, y)` pour tracer les solutions de l'équation différentielle $y' = f(t, y)$ si, `t` est la variable indépendante et si le type est `Diff Eq`.
`f(t, y)` pour tracer le champ des tangentes d'une équation différentielle de la forme $y' = f(t, y)$, si `t` est la variable indépendante et si le type est `Slopefield`.
`f(x, y)` pour tracer une surface définie par $z = f(x, y)$ si `x` est la variable indépendante et si le type est `Fast3D`.
Quelquefois, le nom de la deuxième variable peut être changé, par défaut son nom est `y`. Ce nom est toujours précédé de `Depend` même si c'est une variable indépendante!!!! Ne pas tenir compte du mot `Depend`.

3.3 Le tracé

Avant de faire un tracé, il faut régler différents paramètres.
Quand tous les paramètres sont entrés, pour effectuer le tracé il suffit d'appuyer sur :
`erase draw` (si on veut effacer le graphique précédent) ou
`draw` (si on veut garder le graphique précédent)
du bandeau de l'une des fenêtres :
`PLOT SETUP (shift-bleu F4 (2D/3D))`
`PLOT (shift-bleu F1 (Y=))`
`PLOT WINDOW (shift-bleu F2 (WIN))`.
On peut aussi taper :
`shift-bleu F3 (GRAPH)` pour effectuer le tracé sans effacer le précédent.
On peut revoir le dernier graphe en appuyant sur $\lhd$.

Chapitre 4

Calcul formel

4.1 Les entiers (et les entiers de Gauss)

Dans tout ce paragraphe, on peut utiliser des entiers de Gauss à la place des entiers dans les différentes fonctions.

4.1.1 `Ecriture normale`

La calculatrice peut gérer des nombres entiers en précision infinie, essayez :

$$100!$$

Le symbole ! s'obtient soit en tapant `alpha shift` $-$ `rouge 2`, soit en utilisant `shift-rouge CAT (CHARS)`. Dans ce cas, on sélectionne ! dans `CHARS` (avec les flèches) puis on le recopie avec `echo1` du bandeau.

L'écriture décimale de 100! etant très longue, on peut voir le résultat grâce à la touche `TOOL` puis `view` du bandeau.
La touche `HIST`, puis la flèche vers le haut, permet de remonter dans l'historique et `view` du bandeau de visualiser les résultats.

4.1.2 `DEFINE`

Soit l'exercice suivant :
Calculer les six premiers nombres de Fermat $F_k = 2^{2^k} + 1$ pour $k = 1..6$ et dire s'ils sont premiers.

On tape l'expression

$$2^{2^2} + 1$$

on trouve 17, puis on lance la commande `ISPRIME?()` avec comme argument `ANS(1)`. Cette commande se trouve dans le menu `ARITH (shift-bleu 1)` puis dans le sous menu `1 INTEGER` (ou on l'écrit en mode α).

La réponse est `1.`, ce qui veut dire `vrai`

Grâce à l'historique (`HIST`) je recopie l'expression $2^{2^2} + 1$ dans la ligne de commande et je la modifie en

$$2^{2^3} + 1$$

Ou bien, je tape l'expression $2^{2^K}+1$ `STO FK` puis `3 STO K` etc...
Ou bien, et c'est la meilleure méthode, on définit la fonction `F(K)` à l'aide de `DEF (shift-bleu 2)` ou en tapant

$$\texttt{DEFINE(F(K)} = 2^{2^{\texttt{K}}}+1)$$

La réponse est `NOVAL` et `F` s'inscrit parmi les variables (appuyer sur `VAR` pour le vérifier).
Pour $K = 5$ on tape :

$$\texttt{F}(5)$$

On obtient :

$$4294967297$$

On peut factoriser F_5 avec `FACTOR` que l'on trouve dans le menu :
`ALG (shift-rouge 4)`.
On tape :

$$\texttt{FACTOR(F}(5))$$

On obtient

$$641 \cdot 6700417$$

Pour `F(6)` on trouve :

$$18446744073709551617$$

On factorise avec `FACTOR`, on trouve :

$$274177.67280421310721$$

Attention à la différence entre :

$$2 \cdot 5 = \frac{5}{2}$$

et

$$2.5 = 10$$

4.1.3 `GCD`

`GCD` désigne le PGCD de deux entiers (ou de deux listes d'entiers de même longueur).
On tape :

$$\texttt{GCD}(18, 15)$$

On obtient :

$$3$$

On tape :

$$\texttt{GCD}(\{18, 28\}, \{15, 21\})$$

On obtient :

$$\{3, 7\}$$

en effet $\texttt{GCD}(18, 15) = 3$ et $\texttt{GCD}(28, 21) = 7$

4.1.4 LGCD

LGCD désigne le PGCD d'une liste de nombres entiers.
On tape :

LGCD({18, 15, 21, 36})

On obtient :

3

4.1.5 SIMP2

SIMP2 a comme paramètre deux entiers (ou deux listes d'entiers). Ces deux entiers sont considérés comme représentants d'une fraction. SIMP2 renvoie la fraction simplifiée sous la forme d'une liste de deux entiers.
On tape :

SIMP2(18, 15)

On obtient :

{6, 5}

On tape :

SIMP2({18, 28}, {15, 21})

On obtient :

{6, 5, 4, 3}

4.1.6 LCM

LCM désigne le PPCM de deux entiers (ou de deux listes d'entiers).
On tape :

LCM(18, 15)

On obtient :

90

4.1.7 FACTOR

FACTOR décompose l'entier en produit de facteurs premiers.
On tape :

FACTOR(90)

On obtient :

$2.3^2.5$

4.1.8 FACTORS

FACTORS effectue aussi cette décomposition, mais le résultat est donné sous la forme d'une liste, formée par les diviseurs premiers et leur multiplicité.
On tape :

FACTORS(90)

On obtient :

{2, 1., 3, 2., 5, 1.}

4.1.9 DIVIS

`DIVIS` donne la liste des diviseurs d'un entier.
On tape :

`DIVIS(36)`

On obtient :

`{1, 3, 9, 2, 6, 18, 4, 12, 36}`

4.1.10 IQUOT

`IQUOT` désigne le quotient entier de la division euclidienne de deux entiers.
On tape :

`IQUOT(148, 5)`

On obtient :

`29`

4.1.11 IREMAINDER MOD

`IREMAINDER` désigne le reste entier de la division euclidienne de deux entiers.
On tape :

`IREMAINDER(148, 5)`

ou

`MOD(148, 5)`

On obtient :

`3`

`IREMAINDER` travaille avec des entiers ou des entiers de Gauss, c'est ce qui le différencie de `MOD`.
`MOD` accepte des réels mais pas des entiers de Gauss.
Essayer :

`IREMAINDER(148!, 5! + 2)`

(! s'obtient avec `alpha shift-rouge 2`).

4.1.12 IDIV2

`IDIV2` donne la liste du quotient et du reste entier de la division euclidienne de deux entiers.
On tape :

`IDIV2(148, 5)`

On obtient :

`{29, 3}`

En mode pas à pas, la division se fait comme à l'école, avec l'algorithme dit de la "potence".

4.1.13 ISPRIME ?

`ISPRIME?(N)` si `N` est pseudo-premier renvoie `1.` (vrai) et renvoie `0.` (faux) si `N` n'est pas premier.
Définition : Pour les nombres inférieurs à 10^{14} être pseudo-premier et premier c'est la même chose! ...mais au delà de 10^{14} un nombre pseudo-premier est premier avec une probabilité très forte (cf l'algorithme de Rabin section7.6).
On tape :

`ISPRIME?(13)`

On obtient :

`1.`

On tape :

`ISPRIME?(14)`

On obtient :

`0.`

4.1.14 NEXTPRIME

`NEXTPRIME(N)` désigne le premier nombre pseudo-premier trouvé après `N`.
On tape :

`NEXTPRIME(75)`

On obtient :

`79`

4.1.15 PREVPRIME

`PREVPRIME(N)` désigne le premier nombre pseudo-premier trouvé avant `N`.
On tape :

`PREVPRIME(75)`

On obtient :

`73`

4.1.16 IEGCD

`IEGCD(A,B)` désigne le PGCD étendu (identité de Bézout) de deux entiers.
`IEGCD(A,B)` renvoie `{D,U,V}` vérifiant `AU+BV=D` et `D=PGCD(A,B)`.
On tape :

`IEGCD(48,30)`

On obtient :

$$\{6, 2, -3\}$$

En effet :

$$2 \cdot 48 + (-3) \cdot 30 = 6$$

4.1.17 IABCUV

`IABCUV(A,B,C)` donne `{U ,V}` vérifiant `AU+BV=C`.
Il faut bien sûr que `C` soit un multiple du `PGCD(A,B)` pour obtenir une solution.
On tape :

$$\texttt{IABCUV}(48, 30, 18)$$

On obtient :

$$\{6, -9\}$$

4.1.18 ICHINREM

`ICHINREM([A,P],[B,Q])` désigne un nombre `X` vérifiant :
`X=A (mod P)` et `X=B (mod Q)`.
Il existe toujours une solution `X` si `P` et `Q` sont premiers entre eux, et toutes les solutions sont congrues modulo `N=P.Q`
Exemple :
Trouver les solutions de :

$$\begin{cases} X = & 3 \pmod 5 \\ X = & 9 \pmod{13} \end{cases}$$

On tape :

$$\texttt{ICHINREM}([3, 5], [9, 13])$$

On obtient :

$$[-147, 65]$$

ce qui veut dire que `X=-147 (mod 65)`

4.1.19 PA2B2

`PA2B2` décompose un entier p premier, congru à 1 modulo 4 en $p = a^2 + b^2$. La calculatrice donne le résultat sous la forme $a + b \cdot i$
On tape :

$$\texttt{PA2B2}(17)$$

On obtient :

$$4 + \mathtt{i}$$

en effet $17 = 4^2 + 1^2$

4.1.20 EULER

`EULER` désigne l'indicadrice d'EULER d'un entier.
EULER(n) est égale au cardinal de l'ensemble des nombres inférieurs à n et premiers avec n.
On tape :

$$\texttt{EULER}(21)$$

On obtient :

$$12$$

En effet l'ensemble :
E={2,4,5,7,8,10,11,13,15,16,17,19} correspond aux nombres inférieurs à 21 qui sont premiers avec 21, et E a comme cardinal 12.

4.2 Les rationnels

Essayez :

$$\frac{123}{12}+\frac{57}{21}$$

puis `ENTER` la réponse est :

$$\frac{363}{28}$$

avec `shift-rouge ENTER (→NUM)` la réponse est :

$$12.9642857143$$

Si on mélange les deux représentations par exemple :

$$\frac{1}{2}+0.5$$

la caculatrice demande à passer en mode `approx` pour faire le calcul ; il faut alors répondre `yes` pour obtenir :

$$1.$$

Revenez ensuite en mode exact (`MODE cas` du bandeau etc...).

4.2.1 PROPFRAC

`PROPFRAC(A/B)` écrit la fraction $\frac{A}{B}$ sous la forme :

$$Q+\frac{R}{B} \;\; avec \;\; 0 \leq R < B$$

On tape :

$$\texttt{PROPFRAC}(43 \div 12)$$

On obtient :

$$3+\frac{7}{12}$$

4.2.2 FXND

`FXND` a comme argument une fraction et renvoie, la liste formée par le numérateur et le dénominateur de cette fraction simplifiée.
On tape :

$$\texttt{FXND}(42 \div 12)$$

On obtient :

$$\{7,2\}$$

4.2.3 SIMP2

`SIMP2` (cf 4.1.5) a pour argument une liste de deux entiers représentant une fraction, et renvoie comme `FXND` une liste formée par le numérateur et le dénominateur de cette fraction simplifiée.
On tape :

$$\texttt{SIMP2}(\{42,12\})$$

On obtient :

$$\{7,2\}$$

4.3 Les réels

Essayez :

$$\texttt{EXP}(\pi * \sqrt{20})$$

puis `ENTER` la réponse est :

$$\texttt{EXP}(2 * \sqrt{5} * \pi)$$

avec `shift-rouge ENTER (`→`NUM)` la réponse est :

$$1263794.7537$$

4.4 Les complexes

On tape :

$$(\texttt{1} + \texttt{2.i})^2$$

puis `ENTER` .

Si on n'est pas en mode `complex`, la caculatrice demande à changer de mode : il faut alors répondre `yes` pour obtenir la réponse :

$$-(\texttt{3} - \texttt{4} \cdot \texttt{i})$$

Il faut noter que cette expression ne sera pas simplifiée davatange (les résultats mettront toujours en évidence un nombre complexe de partie réelle positive en mode exact).
Vous trouverez dans le menu `shift-rouge 1 (CMPLX)` les fonctions suivantes ayant comme paramètre une expression à valeur complexe :
`ARG` pour déterminer l'argument du paramètre.
`ABS` pour déterminer le module du paramètre.
`CONJ` pour déterminer le conjugué du paramètre.
`RE` pour déterminer la partie réelle du paramètre.
`IM` pour déterminer la partie imaginaire du paramètre.
`NEG` pour déterminer l'opposé du paramètre.
`SIGN` pour déterminer le quotient du paramètre par son module.
Exemple :
On tape :

$$\texttt{ARG}(\texttt{3} + \texttt{4.i})$$

On obtient :

$$\texttt{ATAN}(\frac{\texttt{4}}{\texttt{3}})$$

4.5 Les expressions algébriques

4.5.1 `FACTOR`

`FACTOR` a comme paramètre une expression qu'il factorise.
Exemple :
Factoriser

$$x^4 + 1$$

On tape :

$$\mathtt{FACTOR}(\mathtt{X}^4 + 1)$$

On trouve `FACTOR` dans le menu de `ALG (shift-rouge 4)` (ou on le tape en mode α).
On trouve en mode réel :

$$(\mathtt{X}^2 + \sqrt{2} \cdot \mathtt{X} + 1) \cdot (\mathtt{X}^2 - \sqrt{2} \cdot \mathtt{X} + 1)$$

On trouve en mode complexe (pour cela cocher `complex` : touche `MODE` puis `cas` du bandeau puis cocher avec `chk` puis `ok ok`) :

$$\frac{(2.\mathtt{X} + (1 + \mathtt{i}).\sqrt{2}).(2.\mathtt{X} - (1 + \mathtt{i}).\sqrt{2}).(2.\mathtt{X} + (1 - \mathtt{i}).\sqrt{2}).(2.\mathtt{X} - (1 - \mathtt{i}).\sqrt{2})}{16}$$

4.5.2 EXPAND EVAL

`EXPAND` et `EVAL`) ont comme paramètre une expression qu'ils développent et simplifient.
Exemple :
En faisant `EXPAND(ANS(1))` on obtient à nouveau

$$\mathtt{X}^4 + 1.$$

4.5.3 SUBST

`SUBST` a deux paramètres : une expression dependant d'un paramètre et une égalité (paramètre=valeur de substitution).
`SUBST` effectue la substitution demandée dans l'expression.
On tape :

$$\mathtt{SUBST}(\mathtt{A}^2 + 1, \mathtt{A} = 2)$$

On obtient :

$$2^2 + 1$$

4.5.4 PREVAL

`PREVAL` a trois paramètres : une expression (`F(VX)`) dépendant de la variable contenue dans `VX` et deux expressions `A` et `B` .
`PREVAL` effectue `F(B)-F(A)`.
`PREVAL` est utile pour calculer une intégrale définie à partir d'une primitive : on évalue cette primitive entre les deux bornes de l'intégrale.
On tape :

$$\mathtt{PREVAL}(\mathtt{X}^2 + \mathtt{X}, 2, 3)$$

On obtient :

$$12 - 6$$

4.6 Les fonctions

4.6.1 DERVX

Soit

$$f(x) = \frac{x}{x^2-1} + \ln(\frac{x+1}{x-1})$$

Calculer la dérivée de f.

On trouve DERVX dans le menu :
CALC (shift-bleu 4) sous menu 1.DERIV. & INT... en position 3 (ou on le tape en mode α).
On tape :

$$\texttt{DERVX}(\frac{\texttt{X}}{\texttt{X}^2-1} + \texttt{LN}(\frac{\texttt{X}+1}{\texttt{X}-1}))$$

ou si on a stocké l'expression de $f(x)$ dans F

DERVX(F)

ou si on a défini $F(X)$ à l'aide de DEFINE : (DEFINE(F(X) $= \frac{\texttt{X}}{\texttt{X}^2-1} + \texttt{LN}(\frac{\texttt{X}+1}{\texttt{X}-1})$))

DERVX(F(X))

On trouve une expression compliquée que l'on simplifie en la recopiant ($\triangle$ ENTER ENTER).
On obtient :

$$-\frac{3\cdot\texttt{X}^2-1}{\texttt{X}^4-2\cdot\texttt{X}^2+1}$$

4.6.2 DERIV

DERIV a deux arguments : une expression (ou une fonction) et une variable (ou un vecteur contenant le nom des variables) (voir fonctions de plusieurs variables paragraphe 4.16.1).
DERIV renvoie la dérivée de l' expression (ou de la fonction) par rapport à la variable donnée comme deuxième paramètre (utile pour calculer des dérivées partielles !).
Exemple :
Soit à calculer :

$$\frac{\partial(x.y^2.z^3+x.y)}{\partial z}$$

On tape :

$$\texttt{DERIV}(\texttt{X.Y}^2.\texttt{Z}^3 + \texttt{X.Y}\ ,\ \texttt{Z})$$

On obtient :

$$\texttt{3.X.Y}^2.\texttt{Z}^2$$

4.6.3 INTVX

Soit

$$f(x) = \frac{x}{x^2-1} + \ln(\frac{x+1}{x-1})$$

Calculer une primitive de f.
On trouve INTVX dans le menu CALC (shift-bleu 4) sous menu 1. DERIV.

`& INT...` en position 8 (ou on le tape en mode α).
On tape :

$$\mathtt{INTVX}(\frac{\mathtt{X}}{\mathtt{X}^2-1}+\mathtt{LN}(\frac{\mathtt{X}+1}{\mathtt{X}-1}))$$

ou si on a stocké l'expression de $f(x)$ dans `F`

$$\mathtt{INTVX(F)}$$

ou si on a défini $F(X)$ à l'aide de `DEFINE` ($\mathtt{DEFINE(F(X)} = \frac{\mathtt{X}}{\mathtt{X}^2-1}+\mathtt{LN}(\frac{\mathtt{X}+1}{\mathtt{X}-1})$)

$$\mathtt{INTVX(F(X))}$$

On trouve :

$$\mathtt{X}\cdot\mathtt{LN}(\frac{\mathtt{X}+1}{\mathtt{X}-1})+\frac{3}{2}\cdot\mathtt{LN}(|\mathtt{X}-1|)+\frac{3}{2}\cdot\mathtt{LN}(|\mathtt{X}+1|)$$

Exercice 1
Calculer :

$$\int\frac{2}{x^6+2\cdot x^4+x^2}\,dx$$

On tape :

$$\mathtt{INTVX}(\frac{2}{\mathtt{X}^6+2\cdot\mathtt{X}^4+\mathtt{X}^2})$$

On trouve :

$$-3\cdot\mathtt{ATAN(X)}-\frac{2}{\mathtt{X}}-\frac{\mathtt{X}}{\mathtt{X}^2+1}$$

Remarque :
On peut aussi taper dans l'equation writer (touche `EQW`) :

$$\int_1^{\mathtt{X}}\frac{2}{\mathtt{X}^6+2\cdot\mathtt{X}^4+\mathtt{X}^2}\,\mathtt{dX}$$

qui donne le même résultat plus une constante d'intégration égale à $\frac{3.\pi+10}{4}$.
Exercice 2
Calculer :

$$\int\frac{1}{\sin(x)+\sin(2\cdot x)}\,dx$$

On tape :

$$\mathtt{INTVX}(\frac{1}{\sin(\mathtt{X})+\sin(2\cdot\mathtt{X})})$$

On trouve :

$$-\frac{1}{6}\cdot\mathtt{LN}(|\mathtt{COS(X)}-1|)+\frac{1}{2}\cdot\mathtt{LN}(|\mathtt{COS(X)}+1|)+\frac{-2}{3}\cdot\mathtt{LN}(|2\cdot\mathtt{COS(X)}+1|)-\mathtt{LN(2)}$$

4.6.4 LIMIT

Trouver pour $n>2$, la limite quand x tend vers 0 de :

$$\frac{n\times\tan(x)-\tan(n\times x)}{\sin(n\times x)-n\times\sin(x)}$$

On utilise la commande `LIMIT` que l'on trouve dans le menu :
`CALC (shift-bleu 4)` sous-menu 2 `LIMIT & SERIES` (ou on le tape en mode

α).
On tape :

$$\mathtt{LIMIT}\left(\frac{\mathtt{N}\cdot\mathtt{TAN(X)}-\mathtt{TAN(N}\cdot\mathtt{X)}}{\mathtt{SIN(N}\cdot\mathtt{X)}-\mathtt{N}\cdot\mathtt{SIN(X)}},\mathtt{0}\right)$$

On obtient :

$$\mathtt{2}$$

Trouver la limite quand x tend vers $+\infty$ de :

$$\sqrt{x+\sqrt{x+\sqrt{x}}}-\sqrt{x}$$

On tape :

$$\mathtt{LIMIT}(\sqrt{\mathtt{X}+\sqrt{\mathtt{X}+\sqrt{\mathtt{X}}}}-\sqrt{\mathtt{X}},+\infty)$$

On obtient au bout d'un moment :

$$\frac{1}{2}$$

Attention, $+\infty$ s'obtient en tapant :

$$^{+}/_{-}\ \ ^{+}/_{-}\ \ \infty\ (\mathtt{shift}-\mathtt{bleu}\ \mathtt{0})$$

4.6.5 LIMIT et $\int$

Déterminer la limite quand a tend vers l'infini de :

$$\int_2^a \left(\frac{x}{x^2-1}+\ln(\frac{x+1}{x-1})\right)\,dx$$

On tape dans l'`equationwriter` :

$$\int_2^{+\infty}\left(\frac{\mathtt{X}}{\mathtt{X}^2-1}+\mathtt{LN}(\frac{\mathtt{X}+1}{\mathtt{X}-1})\right)\,\mathtt{dX}$$

Attention, $+\infty$ s'obtient en tapant :

$$^{+}/_{-}\ \ ^{+}/_{-}\ \ \infty\ (\mathtt{shift}-\mathtt{bleu}\ \mathtt{0})$$

On obtient :

$$+\infty-\frac{\mathtt{7.LN(3)}}{\mathtt{2}}$$

et après simplification :

$$+\infty$$

4.6.6 IBP

`IBP` a deux paramètres : une expression de la forme $u(x).v'(x)$ et $v(x)$.
`IBP` renvoie une liste formée de $u(x).v(x)$ et de $-v(x).u'(x)$ c'est à dire des termes que l'on doit calculer quand on fait une intégration par parties.
Il reste alors à calculer l'intégrale du deuxième terme puis à faire la somme avec le premier terme pour obtenir une primitive de $u(x).v'(x)$(c'est très pratique en

mode RPN!!!).
On tape :

$$\mathtt{IBP(LN(X),X)}$$

On obtient :

$$\{\mathtt{X.LN(X)},-1\}$$

Remarque : Si le premier paramètre de IBP est une liste de deux éléments, IBP n'agit que sur le dernier élément de cette liste et ajoute le terme intégré au premier élément de la liste (de façon à pouvoir faire plusieurs IBP à la suite en mode Algébrique).

4.6.7 RISCH

RISCH a deux paramètres : une expression et un nom de variable.
RISCH renvoie une primitive du premier paramètre par rapport à la variable spécifiée en deuxième paramètre.
On tape :

$$\mathtt{RISCH((2.X^2+1).EXP(X^2+1),X)}$$

On obtient :

$$\mathtt{X.EXP(X^2+1)}$$

4.7 Les expressions trigonométriques

4.7.1 TEXPAND

TEXPAND a comme argument une expression trigonométrique.
TEXPAND développe cette expression en fonction de $\sin(x)$ et $\cos(x)$.
Exemple 1 :
On tape :

$$\mathtt{TEXPAND(COS(X+Y))}$$

On obtient :

$$\mathtt{COS(Y).COS(X)-SIN(Y).SIN(X)}$$

Exemple 2 :
On tape :

$$\mathtt{TEXPAND(COS(3.X))}$$

On obtient :

$$\mathtt{4.COS(X)^3-3.COS(X)}$$

Exemple 3 :
On tape :

$$\mathtt{TEXPAND(\frac{SIN(3.X)+SIN(7.X)}{SIN(5.X)})}$$

On obtient après une simplification ($\triangle$ ENTER) :

$$\mathtt{4.COS(X)^2-2}$$

4.7.2 TLIN

TLIN a comme argument une expression trigonométrique.
TLIN linéarise cette expression en fonction de $\sin(n.x)$ et $\cos(n.x)$.
Exemple :
On tape :

$$\mathtt{TLIN(COS(X).COS(Y))}$$

On obtient :

$$\frac{1}{2}.\mathtt{COS(X-Y)}+\frac{1}{2}.\mathtt{COS(X+Y)}$$

Exemple 2 :
On tape :

$$\mathtt{TLIN(COS(X)^3)}$$

On obtient :

$$\frac{1}{4}.\mathtt{COS(3.X)}+\frac{3}{4}.\mathtt{COS(X)}$$

Exemple 3 :
On tape :

$$\mathtt{TLIN(4.COS(X)^2-2)}$$

On obtient :

$$\mathtt{2.COS(2.X)}$$

4.7.3 TCOLLECT

TCOLLECT a comme argument une expression trigonométrique.
TCOLLECT linéarise cette expression en fonction de $\sin(n.x)$ et $\cos(n.x)$ puis rassemble **en mode réel** les sinus et les cosinus de même angle.
On tape :

$$\mathtt{TCOLLECT(SIN(X)+COS(X))}$$

On obtient :

$$\sqrt{2}.\mathtt{COS(X-\frac{\pi}{4})}$$

4.7.4 ACOS2S

ACOS2S a comme argument une expression trigonométrique.
ACOS2S transforme cette expression en remplaçant $\arccos(x)$ par $\frac{\pi}{2}-\arcsin(x)$.
On tape :

$$\mathtt{ACOS2S(ACOS(X)+ASIN(X))}$$

On obtient :

$$\frac{\pi}{2}$$

4.7.5 ASIN2C

`ASIN2C` a comme argument une expression trigonométrique.
`ASIN2C` transforme cette expression en remplaçant $\arcsin(x)$ par $\frac{\pi}{2} - \arccos(x)$.
On tape :

$$\texttt{ASIN2C(ACOS(X)} + \texttt{ASIN(X))}$$

On obtient :

$$\frac{\pi}{2}$$

4.7.6 ASIN2T

`ASIN2T` a comme argument une expression trigonométrique.
`ASIN2T` transforme cette expression en remplaçant $\arcsin(x)$ par $\arctan(\frac{x}{\sqrt{1-x^2}})$.
On tape :

$$\texttt{ASIN2T(ASIN(X))}$$

On obtient :

$$\texttt{ATAN}(\frac{\texttt{TAN(X)}}{\sqrt{1 - \texttt{TAN(X)}^2}})$$

4.7.7 ATAN2S

`ATAN2S` a comme argument une expression trigonométrique.
`ATAN2S` transforme cette expression en remplaçant :
$\arctan(x)$ par $\arcsin(\frac{x}{\sqrt{1+x^2}})$.
On tape :

$$\texttt{ATAN2S(ATAN(X))}$$

On obtient :

$$\texttt{ASIN}(\frac{\texttt{X}}{\sqrt{\texttt{X}^2 + 1}})$$

4.7.8 SINCOS

`SINCOS` a comme argument une expression contenant des exponentielles complexes.
`SINCOS` transforme cette expression en fonction de $\sin(x)$ et de $\cos(x)$.
On tape :

$$\texttt{SINCOS(EXP(i.X))}$$

On obtient :

$$\mathtt{COS(X)} + \mathtt{i.SIN(X)}$$

4.7.9 TAN2SC

TAN2SC a comme argument une expression trigonométrique.
TAN2SC transforme cette expression en remplaçant $\tan(x)$ par $\frac{\sin(x)}{\cos(x)}$.
On tape :

$$\mathtt{TAN2SC(TAN(X))}$$

On obtient :

$$\frac{\mathtt{SIN(X)}}{\mathtt{COS(X)}}$$

4.7.10 TAN2SC2

TAN2SC2 a comme argument une expression trigonométrique.
TAN2SC2 transforme cette expression en remplaçant $\tan(x)$ par $\frac{\sin(2.x)}{1+\cos(2.x)}$ (ou par $\frac{1-\cos(2.x)}{\sin(2.x)}$ si l'on préfère les sinus c'est à dire quand `Prefer sin()` du flag 116 est coché cf 0.4).
On tape :

$$\mathtt{TAN2SC2(TAN(X))}$$

On obtient :

$$\frac{\mathtt{SIN(2.X)}}{1 + \mathtt{COS(2.X)}}$$

4.7.11 HALFTAN

HALFTAN a comme argument une expression trigonométrique.
HALFTAN transforme les $\sin(x)\cos(x)$ et $\tan(x)$ contenus dans l'expression en fonction de $\tan(\frac{x}{2})$.
On tape :

$$\mathtt{HALFTAN(SIN(X)^2 + COS(X)^2)}$$

On obtient ($\mathtt{SQ(X)} = \mathtt{X}^2$) :

$$\left(\frac{\mathtt{2.TAN(\frac{X}{2})}}{\mathtt{SQ(TAN(\frac{X}{2}))} + 1}\right)^2 + \left(\frac{1 - \mathtt{SQ(TAN(\frac{X}{2}))}}{\mathtt{SQ(TAN(\frac{X}{2}))} + 1}\right)^2$$

On obtient après simplification :

4.7.12 TRIG

TRIG a comme argument une expression trigonométrique.
TRIG simplifie cette expression à l'aide de $\sin(x)^2 + \cos(x)^2 = 1$.
On tape :

$$\texttt{TRIG(SIN(X)}^2 + \texttt{COS(X)}^2 + \texttt{1)}$$

On obtient :

$$\texttt{2}$$

4.7.13 TRIGSIN

TRIGSIN a comme argument une expression trigonométrique.
TRIGSIN simplifie cette expression à l'aide de $\sin(x)^2 + \cos(x)^2 = 1$ en privilégiant les sinus.
On tape :

$$\texttt{TRIGSIN(SIN(X)}^4 + \texttt{COS(X)}^2 + \texttt{1)}$$

On obtient :

$$\texttt{SIN(X)}^4 - \texttt{SIN(X)}^2 + \texttt{2}$$

4.7.14 TRIGCOS

TRIGCOS a comme argument une expression trigonométrique.
TRIGCOS simplifie cette expression à l'aide de $\sin(x)^2 + \cos(x)^2 = 1$ en privilégiant les cosinus.
On tape :

$$\texttt{TRIGCOS(SIN(X)}^4 + \texttt{COS(X)}^2 + \texttt{1)}$$

On obtient :

$$\texttt{COS(X)}^4 - \texttt{COS(X)}^2 + \texttt{2}$$

4.7.15 TRIGTAN

TRIGTAN a comme argument une expression trigonométrique.
TRIGTAN simplifie cette expression à l'aide de $\sin(x)^2 + \cos(x)^2 = 1$ en privilégiant les tangentes.
On tape :

$$\texttt{TRIGTAN(SIN(X)}^4 + \texttt{COS(X)}^2 + \texttt{1)}$$

On obtient :

$$\frac{\texttt{2.TAN(X)}^4 + \texttt{3.TAN(X)}^2 + \texttt{2}}{\texttt{TAN(X)}^4 + \texttt{2.TAN(X)}^2 + \texttt{1}}$$

4.7.16 FOURIER

`FOURIER` a deux paramètres : une expression $f(x)$ et un entier n.
`FOURIER` renvoie le coefficient de Fourier c_n de $f(x)$ considérée comme une fonction définie sur $[0, T]$ et périodique de période T (T etant égale au contenu de la variable `PERIOD`).
On a si f est continue par morceaux :

$$f(x) = \sum_{n=-\infty}^{+\infty} c_n e^{\frac{2inx\pi}{T}}$$

Exemple : Déterminer les coefficients de Fourier de la fonction f périodique de période $2.\pi$ et définie sur $[0\ 2.\pi[$ par $f(x) = x^2$.
On tape :

$$2.\pi \ \texttt{STO} \triangleright \texttt{PERIOD}$$

$$\texttt{FOURIER}(\texttt{X}^2, \texttt{N})$$

On obtient après simplification :

$$\frac{2.\mathtt{i}.\mathrm{N}.\pi + 2}{\mathtt{N}^2}$$

Donc si $n \neq 0$ on a :

$$c_n = \frac{2.i.N.\pi + 2}{N^2}$$

Puis on tape :

$$\texttt{FOURIER}(\texttt{X}^2, 0)$$

On obtient :

$$\frac{4.\pi^2}{3}$$

Donc si $n = 0$ on a :

$$c_0 = \frac{4.\pi^2}{3}$$

4.8 Les Exponentielles et les Logarithmes

4.8.1 EXPLN

`EXPLN` a comme argument une expression trigonométrique.
`EXPLN` transforme les fonctions trigonométriques en exponentielles et logarithmes SANS linéariser.
`EXPLN` fait passer en `mode complexe`.
On tape :

$$\texttt{EXPLN(SIN(X))}$$

On obtient :

$$\frac{\texttt{EXP(i.X)} - \frac{1}{\texttt{EXP(i.X)}}}{2.\texttt{i}}$$

4.8.2 LIN

LIN a comme argument une expression contenant des exponentielles et des fonctions trigonométriques.
LIN linéarise cette expression (l'exprime en fonction de $\exp(n.x)$).
LIN fait passer en `mode complexe` quand il y a des fonctions trigonométriques.
Exemple 1 :
On tape :

$$\mathtt{LIN((SIN(X))}$$

On obtient :

$$-(\frac{\mathtt{i}}{2}.\mathtt{EXP(i.X)}) + \frac{\mathtt{i}}{2}.\mathtt{EXP}(-(\mathtt{i.X}))$$

Exemple 2 :
On tape :

$$\mathtt{LIN((COS(X)^2)}$$

On obtient :

$$-(\frac{1}{4}.\mathtt{EXP(2.i.X)}) + \frac{1}{2} + \frac{1}{4}.\mathtt{EXP}(-(\mathtt{2.i.X}))$$

Exemple 3 :
On tape :

$$\mathtt{LIN((EXP(X)+1)^3)}$$

On obtient :

$$\mathtt{3.EXP(X)+1+3.EXP(2.X)+EXP3.X)}$$

4.8.3 LNCOLLECT

LNCOLLECT a comme argument une expression contenant des logarithmes.
LNCOLLECT regroupe les termes en logarithmes. Il est donc préférable de l'utiliser sur une expression factorisée (en utilisant FACTOR).
On tape :

$$\mathtt{LNCOLLECT(LN(X+1)+LN(X-1))}$$

On obtient :

$$\mathtt{LN((X+1)(X-1))}$$

4.8.4 TSIMP

TSIMP simplifie toutes les expressions en les transformant en exponentielles complexes (fait passer en `mode complexe`) puis en réduisant le nombre de variable au sens de LVAR (cf 4.20.2).
On n'utilise TSIMP qu'en dernier ressort.

On tape :

$$\mathtt{TSIMP}(\frac{\mathtt{SIN(3.X)} + \mathtt{SIN(7.X)}}{\mathtt{SIN(5.X)}})$$

On obtient après simplification (en recopiant 2 fois le résultat) :

$$\frac{\mathtt{EXP(i.X)}^4 + 1}{\mathtt{EXP(i.X)}^2}$$

4.9 Les polynômes

4.9.1 GCD

GCD désigne le pgcd (plus grand commun diviseur) de deux polynômes (ou de deux listes de polynômes de même longueur).
On tape :

$$\mathtt{GCD}(\mathtt{X}^2 + 2 \cdot \mathtt{X} + 1, \mathtt{X}^2 - 1)$$

On obtient :

$$\mathtt{X} + 1$$

On tape :

$$\mathtt{GCD}(\{\mathtt{X}^2 + 2 \cdot \mathtt{X} + 1, \mathtt{X}^3 - 1\}, \{\mathtt{X}^2 - 1, \mathtt{X}^2 + \mathtt{X} - 2\})$$

On obtient :

$$\{\mathtt{X} + 1, \mathtt{X} - 1\}$$

4.9.2 LGCD

LGCD désigne le pgcd (plus grand commun diviseur) d'une liste de polynômes.
LGCD renvoie une liste constituée par la liste des polynômes d'entrée, et le PGCD des polynômes de cette liste.
On tape :

$$\mathtt{LGCD}(\{\mathtt{X}^2 + 2 \cdot \mathtt{X} + 1, \mathtt{X}^3 + 1, \mathtt{X}^2 - 1, \mathtt{X}^2 + \mathtt{X}\})$$

On obtient :

$$\{\{\mathtt{X}^2 + 2 \cdot \mathtt{X} + 1, \mathtt{X}^3 + 1, \mathtt{X}^2 - 1, \mathtt{X}^2 + \mathtt{X}\}\ ,\ \mathtt{X} + 1\}$$

4.9.3 SIMP2

SIMP2 a comme paramètre deux polynômes (ou deux listes de polynômes de même longueur). Ces deux polynômes sont considérés comme représentant une fraction rationnelle.
SIMP2 renvoie la fraction rationnelle simplifiée sous la forme d'une liste de deux polynômes.
On tape :

$$\mathtt{SIMP2}(\mathtt{X}^3 - 1, \mathtt{X}^2 - 1)$$

On obtient :

$$\{\mathtt{X}^2 + \mathtt{X} + 1, \mathtt{X} + 1\}$$

4.9.4 LCM

LCM désigne le ppcm (plus petit commun multiple) de deux polynômes (ou de deux listes de polynômes de même longueur).
On tape :

$$\mathtt{LCM}(\mathtt{X}^2+2\cdot\mathtt{X}+1,\mathtt{X}^2-1)$$

On obtient :

$$(\mathtt{X}^2+2\cdot\mathtt{X}+1)\cdot(\mathtt{X}-1)$$

4.9.5 FACTOR

FACTOR a pour argument un polynôme ou une liste de polynômes.
FACTOR factorise le polynôme ou la liste de polynômes.
On tape :

$$\mathtt{FACTOR}(\mathtt{X}^2+2\cdot\mathtt{X}+1)$$

On obtient :

$$(\mathtt{X}+1)^2$$

On tape :

$$\mathtt{FACTOR}(\mathtt{X}^4-2.\mathtt{X}^2+1)$$

On obtient :

$$(\mathtt{X}-1)^2.(\mathtt{X}+1)^2$$

On tape :

$$\mathtt{FACTOR}(\{\mathtt{X}^3-2.\mathtt{X}^2+1,\mathtt{X}^2-\mathtt{X}\})$$

On obtient :

$$\{\frac{(\mathtt{X}-1).(2.\mathtt{X}+-1+\sqrt{5}).(2.\mathtt{X}-(1+\sqrt{5}))}{4}\ ,\ \mathtt{X}(\mathtt{X}-1)\}$$

4.9.6 FACTORS

FACTORS a pour argument un polynôme ou une liste de polynômes.
FACTORS donne la liste des facteurs du polynôme avec leur multiplicité.
On tape :

$$\mathtt{FACTORS}(\mathtt{X}^2+2\cdot\mathtt{X}+1)$$

On obtient :

$$\{\mathtt{X}+1,2\}$$

On tape :

$$\mathtt{FACTORS}(\mathtt{X}^4-2.\mathtt{X}^2+1)$$

On obtient :

$$\{\mathtt{X}-1,2.\ ,\ \mathtt{X}+1,2.\}$$

On tape :

$$\mathtt{FACTORS}(\{\mathtt{X}^3-2.\mathtt{X}^2+1,\mathtt{X}^2-\mathtt{X}\})$$

On obtient :

$$\{\{\mathtt{X}-1,1.\ ,\ 2.\mathtt{X}+-1+\sqrt{5},1.\ ,\ 2.\mathtt{X}-(1+\sqrt{5}),1.\ ,\ 4,-1.\},$$

$$\{\mathtt{X},1.\ ,\ \mathtt{X}-1,1.\}\}$$

4.9.7 DIVIS

DIVIS a pour argument un polynôme (ou une liste de polynômes de même longueur) et renvoie la liste des diviseurs.
On tape :

$$\mathtt{DIVIS}(\mathtt{X}^4 - 1)$$

On obtient :

$$\{1, \mathtt{X}^2 + 1, \mathtt{X} - 1, \mathtt{X}^3 - \mathtt{X}^2 + \mathtt{X} - 1, \mathtt{X} + 1, \mathtt{X}^3 + \mathtt{X}^2 + \mathtt{X} + 1, \mathtt{X}^2 - 1, \mathtt{X}^4 - 1\}$$

4.9.8 QUOT

QUOT donne le quotient de deux polynômes dans la division selon les puissances décroissantes.
On tape :

$$\mathtt{QUOT}(\mathtt{X}^2 + 2 \cdot \mathtt{X} + 1, \mathtt{X})$$

On obtient :

$$\mathtt{X} + 2$$

4.9.9 REMAINDER

REMAINDER donne le reste de la division de deux polynômes (division selon les puissances décroissantes).
On tape :

$$\mathtt{REMAINDER}(\mathtt{X}^3 - 1, \mathtt{X}^2 - 1)$$

On obtient :

$$\mathtt{X} - 1$$

4.9.10 DIV2

Donne la liste, du quotient et du reste de la division selon les puissances décroissantes de deux polynômes.
On tape :

$$\mathtt{DIV2}(\mathtt{X}^2 + 2 \cdot \mathtt{X} + 1, \mathtt{X})$$

On obtient :

$$\{\mathtt{X} + 2, 1\}$$

Le mode pas à pas est ici intéressant car il donne les restes successifs...

4.9.11 EGCD

Il s'agit de l'identité de Bézout (Extended Greatest Common Divisor). $\mathtt{EGCD}(\mathtt{A}[\mathtt{X}], \mathtt{B}[\mathtt{X}])$ renvoie $\{\mathtt{D}[\mathtt{X}], \mathtt{U}[\mathtt{X}], \mathtt{V}[\mathtt{X}]\}$ avec D,U,V vérifiant :

$$\mathtt{D}[\mathtt{X}] = \mathtt{U}[\mathtt{X}] * \mathtt{A}[\mathtt{X}] + \mathtt{V}[\mathtt{X}] * \mathtt{B}[\mathtt{X}]$$

On tape :

$$\mathtt{EGCD}(\mathtt{X}^2 + 2 \cdot \mathtt{X} + 1, \mathtt{X}^2 - 1)$$

On obtient :

$$\{2 \cdot \mathtt{X} + 2, 1, -1\}$$

4.9.12 ABCUV

Il s'agit encore de l'identité de Bézout. Les paramètres sont trois polynômes, A, B, C (C doit être un multiple du PGCD(A,B)) :
$\mathtt{ABCUV(A[X],B[X],C[X])}$ renvoie $\{\mathtt{U[X],V[X]}\}$ avec U,V vérifiant :

$$\mathtt{C[X]} = \mathtt{U[X]} * \mathtt{A[X]} + \mathtt{V[X]} * \mathtt{B[X]}$$

On tape :

$$\mathtt{ABCUV(X^2+2\cdot X+1, X^2-1, X+1)}$$

On obtient :

$$\{\frac{1}{2}, \frac{-1}{2}\}$$

On tape :

$$\mathtt{ABCUV(X^2+2\cdot X+1, X^2-1, X^3+1)}$$

On obtient :

$$\{\frac{X^2-X+1}{2}, -\frac{X^2-X+1}{2}\}$$

4.9.13 HORNER

HORNER a deux paramètres : un polynôme $P[X]$ et un nombre a, et renvoie une liste composée du polynôme $Q[X]$ (quotient de $P[X]$ par $X-a$), de a et de $P[a]$.
On tape :

$$\mathtt{HORNER(X^4+2.X^3-3.X^2+X-2, 1)}$$

On obtient :

$$\{X^3+3.X^2+1\ ,\ 1\ ,\ -1\}$$

4.9.14 PTAYL

Il s'agit d'écrire un polynôme $P[X]$ selon les puissances de $X-a$.
PTAYL a deux paramètres : un polynôme P et un nombre a.
On tape :

$$\mathtt{PTAYL(X^2+2\cdot X+1, 2)}$$

On obtient le polynôme Q[X] :

$$X^2+6\cdot X+9$$

Attention, on a :

$$\mathtt{P[X]} = \mathtt{Q[X-2]}$$

4.9.15 ZEROS

ZEROS a deux paramètres : un polynôme P et le nom de la variable.
ZEROS renvoie la liste des zéros SANS leur multiplicité.
On tape :

$$\mathtt{ZEROS(X^4-1, X)}$$

On obtient :
-en mode réel

$$\{-1 \ , \ 1\}$$

-en mode complexe

$$\{-1 \ , \ 1 \ , \ -i \ , \ i\}$$

4.9.16 PROOT

PROOT est la commande numérique de la HP48.
PROOT a comme argument un vecteur de composantes les coefficients d'un polynôme (par ordre décroissant).
PROOT renvoie un vecteur de composantes les racines du polynôme.
Pour chercher les racines de $P[x] = x^5 - 2.x^4 + x^3$, on tape :

$$\texttt{PROOT}([1, -2, 1, 0, 0, 0])$$

On obtient :

$$[0., 0., 0., 1., 1.]$$

ce qui signifie que : 0 est racine triple et 1 est racine double de $P[x]$.

4.9.17 FROOTS

FROOTS a comme argument une fraction rationnelle $F[x]$.
FROOTS renvoie un vecteur de composantes les racines et les pôles de $F[x]$ suivis de leur multiplicité.
On tape :

$$\texttt{FROOTS}(\frac{X^5 - 2.X^4 + X^3}{X - 2})$$

On obtient :

$$[2, -1., 0, 3., 1, 2.]$$

ce qui signifie que : 2 est un pôle d'ordre 1, 0 est racine triple et 1 est racine double de $F[x] = \frac{x^5-2.x^4+x^3}{x-2}$.

4.9.18 PCOEF

PCOEF est la commande numérique de la HP48.
PCOEF a comme argument, un vecteur de composantes les racines d'un polynôme $P[x]$.
PCOEF renvoie un vecteur de composantes, les coefficients du polynôme unitaire $P[x]$ (par ordre décroissant).
On tape :

$$\texttt{PCOEF}([1, 2, 0, 0, 3])$$

On obtient :

$$[1., -6., 11., -6., 0., 0.]$$

c'est à dire le développement du polynôme $P[x] = (x - 1).(x - 2).x.x.(x - 3)$: $x^5 - 6.x^4 + 11.x^3 - 6.x^2$.

4.9.19 FCOEF

FCOEF a comme argument un vecteur de composantes les racines et les pôles d'une fraction rationnelle $F[x]$ suivis de leur multiplicité.
FCOEF renvoie la fraction rationnelle $F[x]$.
On tape :

$$\mathtt{FCOEF}([1,2,0,3,2,-1])$$

On obtient :

$$\frac{\mathtt{X}^5 - 2.\mathtt{X}^4 + \mathtt{X}^3}{\mathtt{X} - 2}$$

puisque $(x-1)^2.x^3 = x^5 - 2.x^4 + x^3$

4.9.20 CHINREM

CHINREM a comme argument deux vecteurs ayant chacun comme composantes deux polynômes.
CHINREM renvoie un vecteur de composantes deux polynômes.
`CHINREM([A[X],R[X]],[B[X],Q[X]])` désigne les polynômes `P[X]` et `S[X]` vérifiant :
`S[X]=R[X].Q[X]`,
`P[X]=A[X] (mod R[X])` et `P[X]=B[X] (mod Q[X])`.
Il existe toujours une solution `P[X]` si `R[X]` et `Q[X]` sont premiers entre eux, et toutes les solutions sont congrues modulo `S[X]=R[X].Q[X]`
Trouver les solutions $P[X]$ de :

$$\begin{cases} P[X] = & X \pmod{X^2+1} \\ P[X] = & X-1 \pmod{X^2-1} \end{cases}$$

On tape :

$$\mathtt{CHINREM}([\mathtt{X},\mathtt{X}^2+1],[\mathtt{X}-1,\mathtt{X}^2-1])$$

On obtient :

$$[-\frac{\mathtt{X}^2 - 2.\mathtt{X} + 1}{2}, -\frac{\mathtt{X}^4 - 1}{2}]$$

donc $P[X] = -\frac{X^2-2.X+1}{2}$ (mod $-\ \frac{X^4-1}{2}$)

4.9.21 TRUNC

TRUNC permet de tronquer un polynôme à un ordre donné (utile quand on fait des développements limités).
TRUNC a deux arguments : un polynôme et X^n.
TRUNC renvoie le polynôme tronqué à l'ordre $n-1$ (pas de termes $\geq$ à X^n).
On tape :

$$\mathtt{TRUNC}((1 + \mathtt{X} + \frac{1}{2}.\mathtt{X}^2)^3\ ,\ \mathtt{X}^4)$$

On obtient :

$$4.\mathtt{X}^3 + \frac{9}{2}.\mathtt{X}^2 + 3.\mathtt{X} + 1$$

4.9.22 LAGRANGE

LAGRANGE a comme argument une matrice de deux lignes et n colonnes : la première ligne correspond à des valeurs d'abscisses x_i, et la deuxième ligne correspond à des valeurs d'ordonnés y_i $(i = 1..n)$.
LAGRANGE renvoie le polynôme P de degré $n-1$ tel que $P(x_i) = y_i$.
On tape :

$$\texttt{LAGRANGE}([[1,3],[0,1]])$$

On obtient :

$$\frac{\mathtt{X}-1}{2}$$

en effet $\frac{x-1}{2} = 0$ pour $x = 1$ et $\frac{x-1}{2} = 1$ pour $x = 3$

4.9.23 LEGENDRE

LEGENDRE a comme argument un entier n.
LEGENDRE renvoie le polynôme non nul solution de l'équation différentielle :

$$(x^2-1).y'' - 2.x.y' - n(n+1).y = 0$$

On tape :

$$\texttt{LEGENDRE}(4)$$

On obtient :

$$\frac{35.\mathtt{X}^4 - 30.\mathtt{X}^2 + 3}{8}$$

4.9.24 HERMITE

HERMITE a comme argument un entier n.
HERMITE renvoie le polynôme de HERMITE de degré n.
On tape :

$$\texttt{HERMITE}(6)$$

On obtient :

$$64.\mathtt{X}^6 - 480.\mathtt{X}^4 + 720.\mathtt{X}^2 - 120$$

4.9.25 TCHEBYCHEFF

TCHEBYCHEFF a comme argument un entier n.
Si $n > 0$, TCHEBYCHEFF renvoie le polynôme T_n tel que :

$$T_n[x] = \cos(n.\arccos(x))$$

Si $n < 0$ TCHEBYCHEFF renvoie le polynôme de Tchebycheff de seconde espèce :

$$T_n[x] = \frac{\sin(n.\arccos(x))}{\sin(\arccos(x))}$$

On tape :

$$\texttt{TCHEBYCHEFF}(4)$$

On obtient :

$$\mathtt{8.X^4 - 8.X^2 + 1}$$

en effet :
$\cos(4.x) = Re((\cos(x) + i.\sin(x))^4)$
$\cos(4.x) = \cos(x)^4 - 6.\cos(x)^2.(1 - \cos(x)^2) + ((1 - \cos(x)^2)^2.$
$\cos(4.x) = T_4(\cos(x)).$
On tape :

$$\mathtt{TCHEBYCHEFF}(-4)$$

On obtient :

$$\mathtt{8.X^3 - 4.X}$$

en effet :
$\sin(4.x) = \sin(x).(8.\cos(x)^3 - 4.\cos(x)).$

4.9.26 REORDER

REORDER a deux paramètres : une expression et un vecteur contenant les noms des variables dans un certain ordre.
REORDER développe l'expression selon l'ordre des variables donné dans le second paramètre.
On tape :

$$\mathtt{REORDER(X^2 + 2.X.A + A^2 + Z^2 - X.Z, [A, X, Z])}$$

On obtient :

$$\mathtt{A^2 + 2.X.A + X^2 - Z.X + Z^2}$$

4.10 Les fractions rationnelles

4.10.1 FXND

FXND a comme argument une fraction rationnelle et renvoie la liste formée par le numérateur et le dénominateur de cette fraction simplifiée.
On tape :

$$\mathtt{FXND(\frac{X^2 - 1}{X - 1})}$$

On obtient :

$$\{\mathtt{X + 1, 1}\}$$

4.10.2 SIMP2

SIMP2 a comme paramètre deux polynômes (ou deux listes de polynômes de même longueur). Ces deux polynômes sont considérés comme représentant une fraction rationnelle.
SIMP2 renvoie la fraction rationnelle simplifiée sous la forme d'une liste de deux polynômes.
On tape :

$$\mathtt{SIMP2(X^3 - 1, X^2 - 1)}$$

On obtient :

$$\{\mathtt{X^2 + X + 1, X + 1}\}$$

4.10.3 PROPFRAC

PROPFRAC a comme argument une fraction rationnelle.
PROPFRAC renvoie cette fraction rationnelle écrite de manière à mettre en évidence sa partie entière.
PROPFRAC(A[X]/B[X]) écrit la fraction rationnelle $\frac{A[X]}{B[X]}$ sous la forme :

$$Q[X] + \frac{R[X]}{B[X]}$$

avec $R[X] = 0$ ou $0 \leq deg(R[X]) < deg(B[X])$.
On tape :

$$\texttt{PROPFRAC}(\frac{(5.X + 3).(X - 1)}{X - 2})$$

On obtient :

$$5.X - 12 + \frac{21}{X + 2}$$

4.10.4 PARTFRAC

Décomposer en éléments simples la fraction rationnelle :

$$\frac{x^5 - 2 \times x^3 + 1}{x^4 - 2 \times x^3 + 2 \times x^2 - 2 \times x + 1}$$

On utilise la commande PARTFRAC.
Cette commande se trouve dans le menu ARITH (shift-bleu 1) sous menu 2.POLYNOMIAL... en position 14 (ou on l'écrit en mode α).
On tape :

$$\texttt{PARTFRAC}(\frac{X^5 - 2 * X^3 + 1}{X^4 - 2 * X^3 + 2 * X^2 - 2 * X + 1})$$

On obtient en mode réel :

$$X + 2 + \frac{\frac{-1}{2}}{X - 1} + \frac{\frac{X-3}{2}}{X^2 + 1}$$

On obtient en mode complexe :

$$X + 2 + \frac{\frac{1-3.i}{4}}{X + i} + \frac{\frac{-1}{2}}{X - 1} + \frac{\frac{1+3.i}{4}}{X - i}$$

4.11 Le calcul modulaire

On peut faire des calculs modulo p c'est à dire dans Z/pZ ou dans $Z/pZ[X]$.
Attention, pour certaines commandes il faut choisir un nombre p premier.
La représentation choisie est la représentation symétrique (-1 au lieu de 6 modulo 7).
Il faut mettre dans la variable MODULO, du répertoire HOME, la valeur de p.
DANS LA SUITE LES EXEMPLES SERONT TRAÎTÉS AVEC p=13

4.11.1 MODSTO

Pour mettre dans MODULO la valeur de p (par exemple p=13) on peut utiliser :
MODE cas MODULO ... ou 13 STO MODULO (si on est dans HOME) ou encore MODSTO(13).
MODSTO permet de changer la valeur de la variable MODULO du répertoire HOME.
On taperait par exemple : MODSTO(5) ou 5 STO MODULO pour faire des calculs modulo 5.
DANS LA SUITE LES EXEMPLES SERONT TRAÎTÉS AVEC p=13

4.11.2 ADDTMOD

ADDTMOD réalise uneaddition dans $Z/pZ[X]$.
On tape :

$$\mathtt{ADDTMOD}(\mathtt{11X}+5, \mathtt{8X}+6)$$

On obtient :

$$\mathtt{6X}-2$$

4.11.3 SUBTMOD

SUBTMOD réalise une soustraction dans $Z/pZ[X]$.
On tape :

$$\mathtt{SUBTMOD}(\mathtt{11X}+5, \mathtt{8X}+6)$$

On obtient :

$$\mathtt{3X}-1$$

4.11.4 MULTMOD

MULTMOD réalise une multiplication dans $Z/pZ[X]$.
On tape :

$$\mathtt{MULTMOD}(\mathtt{11X}+5, \mathtt{8X}+6)$$

On obtient :

$$-(\mathtt{3X}^2-\mathtt{2X}-4)$$

4.11.5 DIV2MOD

Les arguments sont deux polynômes A[X] et B[X]. Le résultat est la liste du quotient et du reste de la division euclidienne de A[X] par B[X] dans $Z/pZ[X]$.
On tape :

$$\mathtt{DI2VMOD}(\mathtt{X}^3+\mathtt{X}^2+1, \mathtt{2X}^2+4)$$

puisque

$$X^3+X^2+1=(2X^2+4).(\frac{X+1}{2})+\frac{5X-4}{4}$$

On obtient :

$$\{\frac{\mathtt{X}+1}{2}\ ,\ \frac{\mathtt{5X}-4}{4}\}$$

puis

$$\mathtt{EXPANDMOD}(\{\frac{\mathtt{X}+1}{2}\ ,\ \frac{\mathtt{5X}-4}{4}\})$$

on obtient :

$$\{-(\mathtt{6X}+6)\ ,\ -(\mathtt{2X}+1)\}$$

4.11.6 DIVMOD

Les arguments sont deux polynômes `A[X]` et `B[X]`. Le résultat est la fraction rationnelle $\frac{\mathtt{A[X]}}{\mathtt{B[X]}}$ simplifiée dans $Z/pZ[X]$.
On tape :

$$\mathtt{DIVMOD}(2\mathtt{X}^2+5, 5\mathtt{X}^2+2\mathtt{X}-3)$$

On obtient :

$$\frac{5\mathtt{X}+3}{6\mathtt{X}+6}$$

4.11.7 POWMOD

`POWMOD(X,N)` calcule `X` à la puissance `N` dans $Z/pZ[X]$.
Le contenu p de `MODULO` doit être un nombre premier inférieur à 100.
On tape :

$$\mathtt{POWMOD}(2\mathtt{X}+1, 5)$$

On obtient :

$$6.\mathtt{X}^5+2.\mathtt{X}^4+2.\mathtt{X}^3+\mathtt{X}^2-3.\mathtt{X}+1$$

car :
$10=-3 \pmod{13}$ $40=1 \pmod{13}$ $80=2 \pmod{13}$ $32=6 \pmod{13}$.

4.11.8 INVMOD

`INVMOD` a comme argument un entier.
`INVMOD` calcule l'inverse de cet entier dans Z/pZ.
On tape :

$$\mathtt{INVMOD}(5)$$

On obtient (car $5\times-5=-25=1 \pmod{13}$) :

$$-5$$

4.11.9 GCDMOD

`GCDMOD` a deux polynômes comme arguments.
`GCDMOD` calcule le PGCD des deux polynômes dans $Z/pZ[X]$.
On tape :

$$\mathtt{GCDMOD}(2\mathtt{X}^2+5, 5\mathtt{X}^2+2\mathtt{X}-3)$$

On obtient :

$$-(4\mathtt{X}-5)$$

4.11.10 EXPANDMOD

`EXPANDMOD` a comme argument une expression polynomiale.
`EXPANDMOD` développe cette expression dans $Z/pZ[X]$.
On tape :

$$\mathtt{EXPANDMOD}((2\mathtt{X}^2+12).(5\mathtt{X}-4))$$

On obtient :

$$-(3\mathtt{X}^3-5\mathtt{X}^2+5\mathtt{X}-4)$$

4.11.11 FACTORMOD

FACTORMOD a comme argument un polynôme.
FACTORMOD factorise ce polynôme dans $Z/pZ[X]$ à condition que l'on ait $\mathtt{p} \leq 97$ et p premier.
On tape :

$$\mathtt{FACTORMOD}(-(3X^3 - 5X^2 + 5X - 4))$$

On obtient :

$$-((3X - 5)(X^2 + 6))$$

4.11.12 RREFMOD

RREFMOD permet de résoudre un système d'équations linéaires de la forme $AX = B$ dans Z/pZ.
L'argument est une matrice formée par A bordée avec B comme dernier vecteur colonne. Le résultat est une matrice formée de A1 et de B1 où A1 a des zéros de part et d'autre de la diagonale et le système $A1X = B1$ est équivalent à $AX = B$.
On tape :

$$\mathtt{RREFMOD}([[1,2,9][3,10,0]])$$

pour résoudre

$$\begin{cases} x + 2\cdot y = 9 \\ 3\cdot x + 10\cdot y = 0 \end{cases}$$

On obtient :

$$\begin{bmatrix} 2 & 0 & 6 \\ 0 & 4 & -1 \end{bmatrix}$$

ce qui veut dire que $2.X = 6$ et $4.Y = -1$ ou encore $X = 3 \quad Y = 3$ (puisque $-4*3 = 1 \pmod{13}$).

4.12 Développements limités et asymptotiques

4.12.1 DIVPC

DIVPC a trois arguments : deux polynômes A[X], B[X] (avec $B(0) \neq 0$) et un entier n.
DIVPC renvoie le quotient Q[X] de la division de A[X] par B[X] selon les puissances croissantes avec $\deg(Q) \leq n$ ou $Q = 0$.
$Q[X]$ est donc le développement limité d'ordre n de $\frac{A[X]}{B[X]}$ au voisinage de $X = 0$.
On tape :

$$\mathtt{DIVPC}(1 + X^2 + X^3, 1 + X^2, 5)$$

On obtient :

$$1 + X^3 - X^5$$

Attention : la machine demande a passer en "puissances croissantes", répondre yes.

4.12.2 TAYLOR0

TAYLOR0 a un seul argument : la fonction de x à développer, et renvoie son développement limité à l'ordre relatif 4 au voisinage de $x = 0$ (si x est la variable courante).
On tape :

$$\texttt{TAYLOR0}(\frac{\texttt{TAN(P.X)} - \texttt{SIN(P.X)}}{\texttt{TAN(Q.X)} - \texttt{SIN(Q.X)}})$$

On obtient :

$$\frac{P^5 - Q^2.P}{4.Q^3}.X^2 + \frac{P^3}{Q^3}$$

Attention : l'ordre 4 veut dire que l'on développe à l'ordre relatif 4 le numérateur et le dénominateur (ici ordre absolu 5 pour le numérateur et le dénominateur, ce qui donne en fin de compte, un ordre 2 (5-3) puisque la valuation du dénominateur est égale à 3).

4.12.3 TAYLR

Donner un developpement limité d'ordre 2 au voisinage de x=0 de :

$$\frac{3 \times \tan(x) - \tan(3 \times x)}{x \times (1 - \cos(3 \times x))}$$

Le dénominateur etant d'ordre 3, pour obtenir un developpement limité d'ordre 2 au voisinage de $x = 0$, il faut faire un développement d'ordre 5 au voisinage de $x = 0$ du numérateur. On utilise la commande TAYLR que l'on trouve dans le menu de CALC (shift-bleu 4) sous menu 2. LIMITS & SER... en position 5 (ou on le tape en mode α). TAYLR est compatible avec la HP48.
On tape :

$$\texttt{TAYLR}(\frac{3 \cdot \texttt{TAN(X)} - \texttt{TAN}(3 \cdot X)}{X \cdot (1 - \texttt{COS}(3 \cdot X))}, X, 5)$$

On obtient :

$$-\frac{16}{9} \cdot (1 + \frac{19}{4} \cdot X^2)$$

4.12.4 SERIES

– développement au voisinage de x=a
  Exemple :
  Donner un développement limité à l'ordre 4 au voisinage de $x = \frac{\pi}{6}$ de $\cos(2 \times x)^2$.
  On utilise la commande SERIES que l'on trouve dans le menu de CALC (shift-bleu 4) en position 8 (ou on le tape en mode α) .
  On tape :

$$\texttt{SERIES}(\texttt{COS}(2 \cdot X)^2, X = \frac{\pi}{6}, 4)$$

  On obtient :

$$\{\{\texttt{Limit} : \frac{1}{4} \;\; \texttt{Equiv} : \frac{1}{4}$$

$$\texttt{Expans} : \; (-\frac{8}{3}h^4 + \frac{8\sqrt{3}}{3}h^3 + 2h^2 - \sqrt{3}h + \frac{1}{4})$$

$$\texttt{Remain}: \frac{\texttt{h}^5}{4}\}\ \texttt{h} = \texttt{X} - \frac{\pi}{6}\}$$

- développement au voisinage de `x=+∞` ou `x=-∞`
  Exemple 1 :
  Donner un développement de $\arctan(x)$ à l'ordre 5 au voisinage de `x=+∞` en prenant comme infiniment petit $h = \frac{1}{x}$.
  On tape :
  $$\texttt{SERIES(ATAN(X)}, \texttt{X} = +\infty, 5)$$
  On obtient :
  $$\{\{\texttt{Limit}: \frac{\pi}{2}\ \texttt{Equiv}: \frac{\pi}{2}$$
  $$\texttt{Expans}: (\frac{\pi}{2} - \texttt{h} + \frac{\texttt{h}^3}{3} - \frac{\texttt{h}^5}{5})\ \texttt{Remain}: \frac{\pi\texttt{h}^6}{2}\}\ \texttt{h} = \frac{1}{\texttt{X}}\}$$
  Exemple 2 :
  Donner un développement de $(2x-1)e^{\frac{1}{x-1}}$ à l'ordre 2 au voisinage de `x=+∞` en prenant comme infiniment petit $h = \frac{1}{x}$.
  On tape :
  $$\texttt{SERIES}((2\texttt{X}-1)\cdot \texttt{EXP}(\frac{1}{\texttt{X}-1}), \texttt{X} = +\infty, 3)$$
  On obtient :
  $$\{\{\texttt{Limit}: +\infty\ \texttt{Equiv}: \frac{2}{\texttt{h}}$$
  $$\texttt{Expans}: (\frac{2+\texttt{h}+2\texttt{h}^2+\frac{17\texttt{h}^3}{6}}{\texttt{h}})\ \texttt{Remain}: 2\texttt{h}^3\}\ \texttt{h} = \frac{1}{\texttt{X}}\}$$
  Exemple 3 :
  Donner un développement de $(2x-1)e^{\frac{1}{x-1}})$ à l'ordre 2 au voisinage de `x=-∞` en prenant comme infiniment petit $h = -\frac{1}{x}$.
  On tape :
  $$\texttt{SERIES}((2\texttt{X}-1)\cdot \texttt{EXP}(\frac{1}{\texttt{X}-1}), \texttt{X} = -\infty, 3)$$
  On obtient :
  $$\{\{\texttt{Limit}: -\infty\ \texttt{Equiv}: -\frac{2}{\texttt{h}}$$
  $$\texttt{Expans}: (\frac{-2+\texttt{h}-2\texttt{h}^2+\frac{17\texttt{h}^3}{6}}{\texttt{h}})\ \texttt{Remain}: -2\texttt{h}^3\}\ \texttt{h} = -\frac{1}{\texttt{X}}\}$$
- développement unidirectionnel
  Il faut utiliser pour l'ordre un réel positif (par exemple 4.) pour faire un developpement au voisinage de $x = a$ avec $x > a$ et un réel négatif (par exemple -4.) pour faire un developpement au voisinage de $x = a$ avec $x < a$.
  Exemple 1 :
  Donner un développement de $\frac{(1+X)^{\frac{1}{X}}}{X^3}$ à l'ordre 2, au voisinage de $X = 0^+$.
  On tape :
  $$\texttt{SERIES}(\frac{(1+\texttt{X})^{\frac{1}{\texttt{X}}}}{\texttt{X}^3}, \texttt{X}, 2.)$$
  On obtient :
  $$\{\{\texttt{Limit}: +\infty\ \texttt{Equiv}: \frac{\texttt{e}}{\texttt{h}^3}\ \texttt{Expans}: (-\frac{\texttt{e.h}-2.\texttt{e}}{2.\texttt{h}^3})\ \texttt{Remain}: \frac{\texttt{e}}{\texttt{h}}\}\ \texttt{h} = \texttt{X}\}$$

Exemple 2 :
Donner un développement de $\frac{(1+X)^{\frac{1}{X}}}{X^3}$ à l'ordre 2, au voisinage de $X = 0^-$.
On tape :

$$\mathtt{SERIES}(\frac{(1+\mathtt{X})^{\frac{1}{\mathtt{X}}}}{\mathtt{X}^3}, \mathtt{X}, -2.)$$

On obtient :

$$\{\{\mathtt{Limit}: -\infty\ \mathtt{Equiv}: \frac{\mathtt{e}}{\mathtt{h}^3}\ \mathtt{Expans}:\ (-\frac{\mathtt{e.h}-2.\mathtt{e}}{2.\mathtt{h}^3})\ \mathtt{Remain}: \frac{\mathtt{e}}{\mathtt{h}}\}\ \mathtt{h} = \mathtt{X}\}$$

Exemple 3 :
Donner un développement de $\frac{(1+X)^{\frac{1}{X}}}{X^3}$ à l'ordre 2, au voisinage de $X = 0$.
On tape :

$$\mathtt{SERIES}(\frac{(1+\mathtt{X})^{\frac{1}{\mathtt{X}}}}{\mathtt{X}^3}, \mathtt{X}, 2)$$

On obtient :

$$\{\{\mathtt{Limit}: \infty\ \mathtt{Equiv}: \frac{\mathtt{e}}{\mathtt{h}^3}\ \mathtt{Expans}:\ (-\frac{\mathtt{e.h}-2.\mathtt{e}}{2.\mathtt{h}^3})\ \mathtt{Remain}: \frac{\mathtt{e}}{\mathtt{h}}\}\ \mathtt{h} = \mathtt{X}\}$$

4.12.5 LIMIT

`LIMIT` a deux paramètres : une expression dépendant d'une variable et une égalité (variable =la valeur où l'on veut calculer la limite).
Il est souvent préférable d'écrire l'expression avec des quotes, pour éviter une réécriture de cette expression sous forme normale (pour ne pas avoir une simplification rationnelle des arguments) avant l'exécution de la commande `LIMIT`.
On tape par exemple :

$$\mathtt{LIMIT}('(2\mathtt{X}-1)\cdot\mathtt{EXP}(\frac{1}{\mathtt{X}-1})', \mathtt{X} = +\infty)$$

On obtient :

$$+\infty$$

4.13 Les matrices

4.13.1 TRAN

`TRAN` a comme argument une matrice A.
`TRAN` renvoie la matrice transposée de A.
On tape :

$$\mathtt{TRAN}(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix})$$

On obtient :

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

4.13.2 TRN

TRN a comme argument une matrice A.
TRN renvoie la matrice adjointe (transposée de la conjuguée) de A (c'est la commande de la HP48).
On tape :

$$\texttt{TRN}(\begin{bmatrix} i & 1+i \\ 1 & 1-i \end{bmatrix})$$

On obtient après simplification :

$$\begin{bmatrix} -i & 1 \\ 1-i & 1+i \end{bmatrix}$$

4.13.3 MAD

MAD a comme argument une matrice carrée A d'ordre n.
MAD renvoie une liste composée du déterminant de A, de l'inverse de A, d'une liste contenant les coefficients matriciels d'un polynôme Q, et du polynôme caractéristique P de A.
On a :

$$P(x) = (-1)^n . \det(A - x.I)$$

Le polynôme à coefficients matriciels $P(A) - P(x).I$ est divisible par $A - x.I$ (puisqu'il s'annule pour $x = A$). Soit $Q(x)$ leur quotient.
Puisque $P(A) = 0$, on a $P(A) - P(x).I = -P(x).I = (A - x.I).Q(x)$.
$Q(x)$ est donc aussi la comatrice de $A - x.I$ et on a :
$Q(x) = I.x^{n-1} + ... + B_0$ où $B_0 =$ la comatrice de A (au signe près si n est pair !).
On tape :

$$\texttt{MAD}(\begin{bmatrix} 4 & 1 & -2 \\ 1 & 2 & -1 \\ 2 & 1 & 0 \end{bmatrix})$$

On obtient :

$$\{8, \begin{bmatrix} \frac{1}{8} & -\frac{1}{4} & \frac{3}{8} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ -\frac{3}{8} & -\frac{1}{4} & \frac{7}{8} \end{bmatrix},$$

$$\{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -2 & 1 & -2 \\ 1 & -4 & -1 \\ 2 & 1 & -6 \end{bmatrix}, \begin{bmatrix} 1 & -2 & 3 \\ -2 & 4 & 2 \\ -3 & -2 & 7 \end{bmatrix}\},$$

$$\mathtt{X}^3 - 6.\mathtt{X}^2 + 12.\mathtt{X} - 8\}$$

4.13.4 HADAMARD

HADAMARD a comme arguments deux matrices A et B de même ordre.
HADAMARD renvoie la matrice constituée par le produit terme à terme des éléments de A et B.
On tape :

$$\texttt{HADAMARD}(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix})$$

On obtient :

$$\begin{bmatrix} 5 & 12 \\ 21 & 32 \end{bmatrix}$$

4.13.5 AXM

Si `AXM` a comme argument une matrice symbolique, `AXM` renvoie une matrice numérique et réciproquement.
On tape :

$$\texttt{AXM}([[1/2,2],[3,4]])$$

On obtient :

$$[[0.5,2],[3,4]]$$

4.13.6 AXL

`AXL` a comme argument une matrice.
`AXL` renvoie cette matrice écrite sous la forme d'une liste de listes.
Et réciproquement `AXL` transforme une liste de listes en une matrice.
On tape :

$$\texttt{AXL}([[1,2],[3,4]])$$

On obtient :

$$\{\{1,2\}\{3,4\}\}$$

On tape :

$$\texttt{AXL}(\{\{1,2\}\{3,4\}\})$$

On obtient :

$$[[1,2],[3,4]]$$

4.13.7 EGVL

`EGVL` a comme argument une matrice A d'ordre n.
`EGVL` renvoie un vecteur constitué par les n valeurs propres de A.
REMARQUE : Si A est symbolique on n'aura que les valeurs propres que le CAS sait calculer (car il faut savoir factoriser le polynôme caractéristique formellement!)
On tape :

$$\texttt{EGV}(\begin{bmatrix} 4 & 1 & -2 \\ 1 & 2 & -1 \\ 2 & 1 & 0 \end{bmatrix})$$

On obtient :

$$[2,2,2]$$

4.13.8 EGV

`EGV` a comme argument une matrice A d'ordre n.
`EGV` renvoie une liste composée d'une matrice de passage dans une base caractéristique et d'un vecteur constitué par les n valeurs propres de A (même remarque

que précédemment).
On tape :

$$\texttt{EGV}(\begin{bmatrix} 4 & 1 & -2 \\ 1 & 2 & -1 \\ 2 & 1 & 0 \end{bmatrix})$$

On obtient :

$$\{\begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 0 \\ 1 & 2 & 0 \end{bmatrix}, [2, 2, 2]\}$$

4.13.9 PCAR

`PCAR` a comme argument une matrice A d'ordre n.
`PCAR` renvoie le polynôme caractéristique P de A ($P[x] = (-1)^n.\det(A - x.I)$)
On tape :

$$\texttt{PCAR}(\begin{bmatrix} 4 & 1 & -2 \\ 1 & 2 & -1 \\ 2 & 1 & 0 \end{bmatrix})$$

On obtient :

$$\texttt{X}^3 - 6.\texttt{X}^2 + 12.\texttt{X} - 8$$

4.13.10 JORDAN

`JORDAN` a comme argument une matrice A d'ordre n.
`JORDAN` renvoie une liste composée du polynôme minimal M de A, du polynôme caractéristique P de A, de la liste des vecteurs propres (eigen) et caractéristiques (char) (chaque vecteur est précédé par sa valeur propre ou caractéristique) et d'un vecteur constitué par les n valeurs propres.
On tape :

$$\texttt{JORDAN}(\begin{bmatrix} 4 & 1 & -2 \\ 1 & 2 & -1 \\ 2 & 1 & 0 \end{bmatrix})$$

On obtient :

$$\{\texttt{X}^3 - 6\texttt{X}^2 + 12\texttt{X} - 8, \texttt{X}^3 - 6\texttt{X}^2 + 12\texttt{X} - 8,$$
$$\{\texttt{Char} : 2 : [1, 0, 0], \texttt{Char} : 2 : [2, 1, 2], \texttt{Eigen} : 2 : [1, 0, 1]\}, [2, 2, 2]\}$$

4.13.11 HILBERT

`HILBERT` a comme argument un entier n.
`HILBERT` renvoie la matrice de Hilbert carrée d'ordre n d'éléments :

$$a_{i,j} = \frac{1}{i + j - 1}$$

On tape :

$$\texttt{HILBERT}(4)$$

On obtient :

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

4.13.12 VANDERMONDE

`VANDERMONDE` a comme argument un vecteur de composantes x_i.
`VANDERMONDE` renvoie la matrice de Vandermonde correspondante(la kième ligne est le vecteur de composantes x_i^{k-1}).
On tape :

$$\mathtt{VANDERMONDE([A,B,C])}$$

On obtient :

$$\begin{bmatrix} 1 & 1 & 1 \\ \mathtt{A} & \mathtt{B} & \mathtt{C} \\ \mathtt{A}^2 & \mathtt{B}^2 & \mathtt{C}^2 \end{bmatrix}$$

4.13.13 LCXM

`LCXM` a comme arguments deux entiers n et p et un programme qui prend en entrée i (un numéro de ligne), j (un numéro de colonne) et qui renvoie la valeur de $a_{i,j}$.
`LCXM` renvoie la matrice de coefficients $a_{i,j}$ de dimension $n.p$
On tape :

$$\mathtt{LCXM}(2,3,\ll\rightarrow \mathtt{I\ J}\ \ll \mathtt{I}+\mathtt{J} \gg\ \gg)$$

On obtient :

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

4.14 Les vecteurs

Dans le menu `shift-bleu SYMB (MTH)` on trouve les fonctions permettant de calculer :
-la norme d'un vecteur : `ABS`
-un produit scalaire avec la commande : `DOT`
-un produit vectoriel avec la commande : `CROSS`

4.15 Les formes quadratiques

4.15.1 QXA

`QXA` a deux arguments : une forme quadratique q et le vecteur de composantes les variables utilisées.
`QXA` renvoie une liste de 2 termes : la matrice A associée à q et le vecteur indiquant les variables utilisées.
On tape :

$$\mathtt{QXA(2.X.Y\ ,\ [X,Y])}$$

On obtient :

$$\{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\ ,\ [\mathtt{X},\mathtt{Y}]\}$$

4.15.2 AXQ

`AXQ` a deux arguments : une matrice symétrique A représentant une forme quadratique q et le vecteur de composantes les variables utilisées.
`AXQ` renvoie une liste de 2 termes : la forme quadratique q et le vecteur indiquant les variables utilisées.
On tape :

$$\mathtt{AXQ([[0,1],[1,0]]\ ,\ [X,Y])}$$

On obtient :

$$\{2.X.Y\ ,\ [\mathtt{X},\mathtt{Y}]\}$$

4.15.3 GAUSS

`GAUSS` a deux arguments : une forme quadratique q et le vecteur de composantes les variables utilisées.
`GAUSS` renvoie une liste de 4 termes : les termes diagonaux d'une matrice diagonale B (obtenus par la décomposition de q en somme de carrés), la matrice Q de changement de base, l'écriture de q sous forme d'une somme de carrés et le vecteur indiquant les variables utilisées.
On a (si on note A la matrice assosiée à q) :

$${}^tQ.B.Q = A$$

On tape :

$$\mathtt{GAUSS(2.X.Y\ ,\ [X,Y])}$$

On obtient :

$$\{[\frac{1}{2},-2]\ ,\ \begin{bmatrix} 1 & 1 \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix}\ ,\ -2.(\frac{\mathtt{Y}-\mathtt{X}}{2})^2+\frac{1}{2}.(\mathtt{Y}+\mathtt{X})^2\ ,\ [\mathtt{X},\mathtt{Y}]\}$$

4.15.4 SYLVESTER

`SYLVESTER` a un seul argument : une matrice symétrique A représentant une forme quadratique q.
`SYLVESTER` renvoie une liste de 2 termes : les termes diagonaux d'une matrice diagonale B (obtenus par la décomposition de q en somme de carrés) et la matrice Q de changement de base.
On a :

$${}^tQ.B.Q = A$$

On tape :

$$\mathtt{SYLVESTER([[0,1],[1,0]])}$$

On obtient :

$$\{[\frac{1}{2},-2]\ ,\ \begin{bmatrix} 1 & 1 \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix}\}$$

4.16 Les fonctions de plusieurs variables

4.16.1 DERIV

DERIV a deux paramètres une application F de R^n dans R et un vecteur de R^n indiquant le nom des variables.
DERIV renvoie le gradient de F ($[\frac{\partial F}{\partial X}, \frac{\partial F}{\partial Y}, \frac{\partial F}{\partial Z}]$ si $n = 3$).
On tape :

$$\mathtt{DERIV}(2.\mathtt{X}^2.\mathtt{Y} - \mathtt{X}.\mathtt{Z}^3, [\mathtt{X}, \mathtt{Y}, \mathtt{Z}])$$

On obtient après simplification :

$$[4.\mathtt{Y}.\mathtt{X} - \mathtt{Z}^3, 2.\mathtt{X}^2, -(3.\mathtt{Z}^2.\mathtt{X})]$$

4.16.2 LAPL

LAPL a deux paramètres : une application F de R^n dans R et un vecteur de R^n indiquant le nom des variables.
LAPL renvoie le laplacien de F ($\frac{\partial^2 F}{\partial X^2} + \frac{\partial^2 F}{\partial Y^2} + \frac{\partial^2 F}{\partial Z^2}$ si $n = 3$).
On tape :

$$\mathtt{LAPL}(2.\mathtt{X}^2.\mathtt{Y} - \mathtt{X}.\mathtt{Z}^3\ ,\ [\mathtt{X}, \mathtt{Y}, \mathtt{Z}])$$

On obtient :

$$4.\mathtt{Y} - 6.\mathtt{X}.\mathtt{Z}$$

4.16.3 HESS

HESS a deux paramètres : une application F de R^n dans R et un vecteur de R^n indiquant le nom des variables.
HESS renvoie une liste constituée de la hessienne de F, du gradient de F et de la liste des variables.
On tape :

$$\mathtt{HESS}(2.\mathtt{X}^2.\mathtt{Y} - \mathtt{X}.\mathtt{Z}\ ,\ [\mathtt{X}, \mathtt{Y}, \mathtt{Z}])$$

On obtient :

$$\{ \begin{bmatrix} 4.\mathtt{Y} & 4.\mathtt{X} & -1 \\ 4.\mathtt{X} & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}\ ,\ [4.\mathtt{X}.\mathtt{Y} - \mathtt{Z}, 2.\mathtt{X}^2, -\mathtt{X}]\ ,\ [\mathtt{X}, \mathtt{Y}, \mathtt{Z}]\}$$

Pour avoir maintenant les points critiques, il suffit, en mode RPN de taper :

$$\mathtt{SOLVE}$$

puisque sur la pile il y a $[4.\mathtt{X}.\mathtt{Y} - \mathtt{Z}, 2.\mathtt{X}^2, -\mathtt{X}]$ et $[\mathtt{X}, \mathtt{Y}, \mathtt{Z}]$
alors qu'en mode ALGÉBRIQUE il faut taper :

$$\mathtt{SOLVE}([4.\mathtt{X}.\mathtt{Y} - \mathtt{Z}, 2.\mathtt{X}^2, -\mathtt{X}]\ ,\ [\mathtt{X}, \mathtt{Y}, \mathtt{Z}])$$

4.16.4 DIV

DIV a deux paramètres : une fonction vectorielle F (application de R^n dans R^n) et un vecteur de R^n indiquant le nom des variables.
DIV désigne la divergence de F.

$$\mathtt{DIV([A,B,C],[X,Y,Z])} = \frac{\partial \mathtt{A}}{\partial \mathtt{X}} + \frac{\partial \mathtt{B}}{\partial \mathtt{Y}} + \frac{\partial \mathtt{C}}{\partial \mathtt{Z}} \text{ (si } n = 3\text{)}$$

On tape :

$$\mathtt{DIV([X.Z, -Y^2, 2.X^Y], [X, Y, Z])}$$

On obtient :

$$\mathtt{Z - 2.Y}$$

4.16.5 CURL

Ici $n = 3$.
CURL a deux paramètres : une fonction vectorielle F (application de R^3 dans R^3) et un vecteur de R^3 indiquant le nom des variables.
CURL désigne le rotationnel de F.

$$\mathtt{CURL([A,B,C],[X,Y,Z])} = [\frac{\partial \mathtt{C}}{\partial \mathtt{Y}} - \frac{\partial \mathtt{B}}{\partial \mathtt{Z}}, \frac{\partial \mathtt{A}}{\partial \mathtt{Z}} - \frac{\partial \mathtt{C}}{\partial \mathtt{X}}, \frac{\partial \mathtt{B}}{\partial \mathtt{X}} - \frac{\partial \mathtt{A}}{\partial \mathtt{Y}}]$$

On tape :

$$\mathtt{CURL([X.Z, -Y^2, 2.X^Y], [X, Y, Z])}$$

On obtient :

$$\mathtt{[2.X^2, X - 2.Y.2X, 0]}$$

4.17 Équations

4.17.1 EXLR

EXLR a comme paramètre une équation.
EXLR renvoie une liste composée des deux membres de l'équation.
On tape :

$$\mathtt{EXLR(A = B)}$$

On obtient :

$$\{\mathtt{A, B}\}$$

4.17.2 SOLVEVX

SOLVEVX a comme paramètre une équation entre deux expressions de la variable contenue dans VX ou une expression (=0 est alors sous-entendu).
SOLVEVX résout l'équation.
Exemple 1 :

On tape :

$$\mathtt{SOLVEVX}(\mathtt{X}^4 - 1 = 3)$$

On obtient en mode réel :

$$\{\mathtt{X} = -\sqrt{2},\ \mathtt{X} = \sqrt{2}\}$$

On obtient en mode complexe :

$$\{\mathtt{X} = -\sqrt{2},\ \mathtt{X} = \sqrt{2},\ \mathtt{X} = -(\mathtt{i}.\sqrt{2}),\ \mathtt{X} = \mathtt{i}\sqrt{2}\}$$

Exemple 2 :
On tape :

$$\mathtt{SOLVEVX}((\mathtt{X} - 2).\mathtt{SIN}(\mathtt{X}))$$

On obtient en mode réel :

$$\{\mathtt{X} = -(2.\pi.\mathtt{n}_1),\ \mathtt{X} = 2.\pi.\mathtt{n}_1,\ \mathtt{X} = 2\}$$

4.17.3 `SOLVE`

`SOLVE` a deux paramètres : une équation entre deux expressions ou une expression (`=0` est alors sous-entendu), et le nom d'une variable.
`SOLVE` permet aussi de résoudre un système d'équations : il suffit pour cela de mettre les équations sous la forme d'un vecteur et le nom des variables dans un vecteur.
`SOLVE` résout l'équation ou le système d'équations.
Exemple 1 :
On tape :

$$\mathtt{SOLVE}(\mathtt{X}^4 - 1 = 3, \mathtt{X})$$

On obtient en mode réel :

$$\{\mathtt{X} = -\sqrt{2},\ \mathtt{X} = \sqrt{2}\}$$

On obtient en mode complexe :

$$\{\mathtt{X} = -\sqrt{2},\ \mathtt{X} = \sqrt{2},\ \mathtt{X} = -(\mathtt{i}.\sqrt{2}),\ \mathtt{X} = \mathtt{i}\sqrt{2}\}$$

Exemple 2 :
On tape :

$$\mathtt{SOLVE}([\mathtt{X} + \mathtt{Y} = 1, \mathtt{X} - \mathtt{Y}], [\mathtt{X}, \mathtt{Y}])$$

On obtient :

$$\{[\mathtt{X} = \frac{1}{2},\ \mathtt{Y} = \frac{1}{2}]\}$$

4.17.4 ISOL

ISOL isole une variable dans une expression ou une équation (la variable doit n'apparaître qu'une seule fois).
Cette commande est la même que pour la HP48.
ISOL a deux paramètres une expression ou une équation et le nom de la variable à isoler.
On tape :

$$\mathtt{ISOL}(\mathtt{X}^4 - 1 = 3, \mathtt{X})$$

On obtient en mode réel :

$$\{\mathtt{X} = -\sqrt{2},\ \mathtt{X} = \sqrt{2}\}$$

On obtient en mode complexe :

$$\{\mathtt{X} = -\sqrt{2},\ \mathtt{X} = \sqrt{2},\ \mathtt{X} = -(\mathtt{i}.\sqrt{2}),\ \mathtt{X} = \mathtt{i}\sqrt{2}\}$$

Attention : si le flag 01 est coché (`Principal value`), ISOL ne renvoie qu'une solution.

4.18 Les systèmes linéaires

Dans tout ce paragraphe, on appelle "matrice augmentée" du système A.X=B (ou matrice représentant le système A.X=B), la matrice formée par la matrice A bordée à droite par le vecteur colonne B.

4.18.1 REF

REF permet de résoudre un système d'équations linéaires que l'on écrit sous forme matricielle :

$$\mathtt{A} \cdot \mathtt{X} = \mathtt{B}$$

La commande REF se trouve dans le menu `MATRICES (shift-bleu 5)`, sous-menu `5 LINEAR SYST...`.
Le paramètre de REF est la matrice augmentée du système (celle formée par la matrice `A` du système bordée à droite par le second membre `B`).
Le résultat est une matrice `[A1,B1]` : `A1` a des zéros sous sa diagonale et les solutions de :

$$\mathtt{A1} \cdot \mathtt{X} = \mathtt{B1}$$

sont les mêmes que celles de :

$$\mathtt{A} \cdot \mathtt{X} = \mathtt{B}$$

Par exemple, soit à résoudre le système :

$$\left\{ \begin{array}{rcr} 3 \cdot x + y & = & -2 \\ 3 \cdot x + 2 \cdot y & = & 2 \end{array} \right.$$

On tape (on utilise `shift-bleu EQW (MTRW)` pour entrer la matrice) :

$$\mathtt{REF}([[3, 1, -2][3, 2, 2]])$$

On obtient :

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

4.18.2 rref

`rref` permet de résoudre un système d'équations linéaires que l'on écrit sous forme matricielle :

$$\mathtt{A} \cdot \mathtt{X} = \mathtt{B}$$

La commande `rref` se trouve dans le menu `MATRICES (shift-bleu 5)`, sous-menu `5 LINEAR SYST...`.
Le paramètre de `rref` est la matrice augmentée du système (celle formée par la matrice `A` du système et ayant comme dernier vecteur colonne le second membre `B`).
Le résultat est une matrice `[A1,B1]` : `A1` a des zéros de part et d'autre de sa diagonale et les solutions de :

$$\mathtt{A1} \cdot \mathtt{X} = \mathtt{B1}$$

sont les mêmes que celles de :

$$\mathtt{A} \cdot \mathtt{X} = \mathtt{B}$$

Il est intéressant d'utiliser `rref` en mode pas à pas en cochant `Step/Step (MODE cas chk)`.
Par exemple, soit à résoudre le système :

$$\left\{ \begin{array}{lcr} 3 \cdot x + y & = & -2 \\ 3 \cdot x + 2 \cdot y & = & 2 \end{array} \right.$$

On tape (on utilise `shift-bleu EQW (MTRW)` pour entrer la matrice) :

$$\mathtt{rref}([[3,1,-2][3,2,2]])$$

On obtient :

$$\left\{ \mathtt{Pivots} : \{1\ 1.\} \begin{bmatrix} 3 & 0 & -6 \\ 0 & 1 & 4 \end{bmatrix} \right\}$$

4.18.3 RREF

`RREF` est identique à `rref` sauf qu'elle ne donne pas les pivots.
`RREF` est la commande de la HP48, voici ce que cela donne en pas à pas avec la HP49G pour résoudre le système :

$$\left\{ \begin{array}{lcr} 3 \cdot x + y & = & -2 \\ 3 \cdot x + 2 \cdot y & = & 2 \end{array} \right.$$

On tape (on utilise `shift-bleu EQW (MTRW)` pour entrer la matrice) :

$$\mathtt{RREF}([[3,1,-2][3,2,2]])$$

On obtient :

$\mathrm{L}_2 = \mathrm{L}_2 - \mathrm{L}_1$

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

puis `ok`

$\mathtt{L}_1 = \mathtt{L}_1 - \mathtt{L}_2$

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

puis `ok`

Reduction result

$$\begin{bmatrix} 3 & 0 & -6 \\ 0 & 1 & 4 \end{bmatrix}$$

puis `ok`
et le résultat (avec des 1 sur la diagonale) s'inscrit dans l'historique :

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

4.18.4 LINSOLVE

`LINSOLVE` permet de résoudre un système d'équations linéaires.
La commande `LINSOLVE` se trouve dans le menu `MATRICES (shift-bleu 5)`, sous-menu `5.LINEAR SYST...` en position 1.
On tape :

`LINSOLVE()`

Puis on entre dans `MATRIXWRITER` en tapant sur `shift-bleu EQW (MTRW)` (le curseur étant dans les parenthéses de `LINSOLVE`) . On coche la touche du bandeau `vect`(si ce n'est pas déjà fait !), puis on tape la première équation (éventuellement à l'aide de `EQW`).

$$\mathtt{2 \cdot X + Y + Z = 1\ ENTER}$$

$$\mathtt{X + Y + 2 \cdot Z = 1\ ENTER}$$

$$\mathtt{X + 2 \cdot Y + Z = 4\ ENTER}$$

`ENTER`
puis, on tape les inconnues :

`[X Y Z]`

et `ENTER`

On obtient si on a coché le mode pas à pas (`MODE cas Step/Step`) :

`L2=2L2-L1`

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

puis `ok`

`L3=2L3-L1`

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

etc...à la fin
`Result`

$$\begin{bmatrix} 8 & 0 & 0 & 4 \\ 0 & 8 & 0 & -20 \\ 0 & 0 & 8 & -4 \end{bmatrix}$$

puis `ok`

$$\{\mathtt{X} = -\frac{1}{2}\ \mathtt{Y} = \frac{5}{2}\ \mathtt{Z} = -\frac{1}{2}\}$$

s'inscrit dans l'historique.

4.19 Les équations différentielles

4.19.1 LDEC

`LDEC` permet de résoudre directement les équations linéaires (`LAP ILAP` (cf 4.19.3) y sont utilisés en interne).
Pour les équations linéaires du second ordre, les paramètres sont le second membre et l'équation caractéristique.
Pour les systèmes differentiels du premier ordre les paramètres sont le second membre (un vecteur) et la matrice du système.
Exemple 1 :
Résoudre :

$$y'' - 6.y' + 9.y \ = \ x.e^{3.x}$$

On tape :

$$\mathtt{LDEC}(\mathtt{X} \cdot \mathtt{EXP}(3 \cdot \mathtt{X}), \mathtt{X}^2 - 6 \cdot \mathtt{X} + 9)$$

On trouve :

$$(\frac{\mathtt{X}^3}{6} - (3 \cdot \mathtt{C0} - \mathtt{C1}) \cdot \mathtt{X} + \mathtt{C0}) \cdot \mathtt{EXP}(3 \cdot \mathtt{X})$$

`C0` et `C1` sont les constantes d'intégration ($\mathtt{y}(0) = \mathtt{C0}$ $\mathtt{y}'(0) = \mathtt{C1}$).
Exemple 2 :
Résoudre :

$$Z' = \begin{bmatrix} 0 & 1 \\ -9 & 6 \end{bmatrix} .Z + \begin{bmatrix} 0 \\ X.EXP(3.X) \end{bmatrix}$$

C'est le même exemple que précédemment avec $Z = [y, y']$.
On tape :

$$\mathtt{LDEC}([0, \mathtt{X} \cdot \mathtt{EXP}(3 \cdot \mathtt{X})], [[0, 1][-9, 6]])$$

On trouve :

$$[(\frac{\mathtt{X}^3}{6} - (3 \cdot \mathtt{V1} - \mathtt{V2}) \cdot \mathtt{X} + \mathtt{V1}) \cdot \mathtt{EXP}(3 \cdot \mathtt{X}),$$
$$(\frac{\mathtt{X}^3}{2} + \frac{\mathtt{X}^2}{2} - (9.\mathtt{V1} - 3.\mathtt{V2}).\mathtt{X} + \mathtt{V2}).\mathtt{EXP}(3.\mathtt{X})]$$

V1 et V2 sont les constantes d'intégration ($\mathtt{Z}(0) = [\mathtt{V1}, \mathtt{V2}]$).

4.19.2 DESOLVE et SUBST

La commande DESOLVE se trouve dans le menu S.SLV (shift-bleu 7) en position 1 ou dans le menu CALC (shift-bleu 4) sous-menu 1 DIFFERENTIAL EQNS....
DESOLVE permet de résoudre d'autres équations différentielles.
Les paramètres sont : l'équation différentielle (où $y\prime$ s'écrit d1Y(X)) et l'inconnue Y(X).
Exemple 1 :
Résoudre :

$$y'' + y = \cos(x) \;\; y(0) = c_0 \;\; y'(0) = c_1$$

On tape :

$$\mathtt{DESOLVE}(\mathtt{d_1 d_1 Y(X)} + \mathtt{Y(X)} = \mathtt{COS(X)}, \mathtt{Y(X)})$$

On trouve :

$$\mathtt{Y(X)} = \mathtt{C0} \cdot \mathtt{COS(X)} + \frac{\mathtt{X} + 2 \cdot \mathtt{C1}}{2} \cdot \mathtt{SIN(X)}$$

On peut ensuite donner une valeur aux constantes en utilisant la commande SUBST qui se trouve dans le menu ALG (shift-rouge 4) en position 6. On écrit, si veut les solutions vérifiant $y(0) = 1$:

$$\mathtt{SUBST}(\mathtt{Y(X)} = \mathtt{C0} \cdot \mathtt{COS(X)} + \frac{\mathtt{X} + 2 \cdot \mathtt{C1}}{2} \cdot \mathtt{SIN(X)}, \mathtt{C0} = 1)$$

On obtient :

$$\mathtt{Y(X)} = \frac{2 \cdot \mathtt{COS(X)} + (\mathtt{X} + 2 \cdot \mathtt{C1}) \cdot \mathtt{SIN(X)}}{2}$$

Exemple 2 :
Résoudre :

$$y'' + y = \cos(x) \;\; y(0) = 1 \;\; y'(0) = c_1$$

Pour avoir les solutions vérifiant $y(0) = 1$ on peut aussi taper directement :

$$\mathtt{DESOLVE}([\mathtt{d_1 d_1 Y(X)} + \mathtt{Y(X)} = \mathtt{COS(X)}, \mathtt{Y(0)} = 1], \mathtt{Y(X)})$$

On trouve alors :

$$\mathtt{Y(X)} = \mathtt{COS(X)} + \frac{\mathtt{X} + 2 \cdot \mathtt{C1}}{2} \cdot \mathtt{SIN(X)}$$

4.19.3 LAP ILAP

Ces commandes se trouvent dans le menu `CALC (shift-bleu 4)` sous-menu `3.DIFFERENTIAL...` en positions 2 et 3.
On utilise les transformées de Laplace (`LAP`) et les transformées de Laplace inverses (`ILAP`) pour résoudre des équations différentielles linéaires à coefficients constants, par exemple :

$$y'' + p.y' + q.y \;=\; f(x) \;\; y(0) = a \;\; y'(0) = b$$

On a :

$$\mathtt{LAP(Y)(P)} = \int_{\mathtt{0}}^{+\infty} \mathtt{e}^{-\mathtt{P.X}}\mathtt{Y(X)dX}$$

$$\mathtt{ILAP(F)(T)} = \frac{\mathtt{1}}{\mathtt{2.i.}\pi} \int_{\mathtt{C}} \mathtt{e}^{\mathtt{Z.T}}\mathtt{dZ}$$

`C` etant une courbe fermée contenant les pôles de `F`
On utilise la propriété suivante :

$$\mathtt{LAP(Y')(P)} = -\mathtt{Y(0)} + \mathtt{P.LAP(Y)(P)}$$

La solution est alors :

$$\mathtt{ILAP}\left(\frac{\mathtt{LAP(F(X))} + (\mathtt{X} + \mathtt{P}) \cdot \mathtt{A} + \mathtt{B}}{\mathtt{X}^2 + \mathtt{P} \cdot \mathtt{X} + \mathtt{Q}}\right)$$

Exemple :
Résoudre :

$$y'' - 6.y' + 9.y \;=\; x.e^{3.x}$$

$$y(0) = a$$

$$y'(0) = b$$

On tape :

$$\mathtt{LAP(X \cdot EXP(3 \cdot X))\ ENTER}$$

On obtient :

$$\frac{\mathtt{1}}{\mathtt{X}^2 - \mathtt{6} \cdot \mathtt{X} + \mathtt{9}}$$

On tape :

$$\mathtt{ILAP}\left(\frac{\mathtt{ANS(1)} + (\mathtt{X} - \mathtt{6}) \cdot \mathtt{A} + \mathtt{B}}{\mathtt{X}^2 - \mathtt{6} \cdot \mathtt{X} + \mathtt{9}}\right)$$

On obtient la solution y :

$$\left(\frac{\mathtt{X}^3}{\mathtt{6}} - (\mathtt{3} \cdot \mathtt{A} - \mathtt{B}) \cdot \mathtt{X} + \mathtt{A}\right) \cdot \mathtt{EXP(3 \cdot X)}$$

4.20 Autres fonctions

4.20.1 EPSX0

EPSX0 a comme paramètre une expression de X et renvoie l'expression où les valeurs plus petites que EPS ont été remplacées par zéro.
On tape :

$$\mathtt{EPSX0}(0.001 + \mathtt{X})$$

On obtient (avec EPS=0.01) :

$$0 + \mathtt{X}$$

On obtient (avec EPS=0.0001) :

$$.001 + \mathtt{X}$$

4.20.2 LVAR

LVAR a comme paramètre une expression et renvoie une liste composée de l'expression et d'un vecteur de composantes le nom des variables indépendantes utilisées dans cette expression.
On tape :

$$\mathtt{LVAR(X.Y.SIN(X))}$$

On obtient :

$$\{\mathtt{X.Y.SIN(X)}, [\mathtt{SIN(X)}, \mathtt{X}, \mathtt{Y}]\}$$

4.20.3 LNAME

LNAME a comme paramètre une expression et renvoie un vecteur de composantes le nom des variables symboliques utilisées dans cette expression.
On tape :

$$\mathtt{LNAME(X.Y.SIN(X))}$$

On obtient :

$$[\mathtt{X}, \mathtt{Y}]$$

4.20.4 XNUM

XNUM a comme paramètre une expression ou un tableau.
XNUM fait passer en mode approximatif et renvoie la valeur numérique de l'expression ou du tableau.
On tape :

$$\mathtt{XNUM}(\sqrt{2})$$

On obtient :

1.41421356237

4.20.5 `XQ`

`XQ` a comme paramètre une expression numérique réelle.
`XQ` fait passer en mode exact et donne une approximation rationnelle ou réelle de l'expression.
On tape :

`XQ(1.41422)`

On obtient :

$$\frac{66441}{46981}$$

On tape :

`XQ(1.414213562)`

On obtient :

$$\sqrt{2}$$

Chapitre 5

Bac 99 et HP49G

5.1 Introduction

Commencez par taper `CASCFG` (Computer Algebra System ConFiG) pour mettre la calculatrice en mode algébrique et pour l'initialiser.
Les différentes commandes utilisées se trouvent dans le menu de la touche `SYMB` :
sous menus :
`ALGEBRA (FACTOR LIN SUBST)`
`ARITHMETIC (IEGCD ISPRIME ? PROPFRAC)`
`CALCULUS (DERIVX DERIV INTVX INT LIMIT)`
`GRAPH (SIGNTAB TABVAR)`
`TRIGONOMETR (TEXPAND)`
et dans le menu `shift-rouge 1 (CMPLX)`
`RE IM`
Après chaque commande, il faut taper `ENTER`, on oubliera souvent de le spécifier ! ! !
Dans ce qui suit, vous trouverez l'épreuve de mathématiques 1999 (série S) du Bac.
On a essayé de faire faire le plus de choses possibles à la HP49G... On remarquera, qu'il reste quand même à l'élève le soin de justifier les calculs et de faire un peu de raisonnement....

5.2 Exercice 1

L'objet de cet exercice est de tracer la courbe Γ décrite par M d'affixe $\frac{1}{2} \cdot z^2 - z$, lorsque m d'affixe z décrit le cercle C de centre O et de rayon 1. Soit t un réel de $[-\pi, \pi]$ et m le point de C d'affixe $z = e^{i \cdot t}$.

1. Calcul des coordonnées de M :
   On entre tout d'abord l'expression $\frac{1}{2} \cdot z^2 - z$ à l'aide de `EQW`.
   On tape :

   `EQW alpha Z` $\mathtt{y^x}$ `2` $\triangleright$ $\div$ `2` $\triangleright$ $-$ `alpha Z ENTER`

   L'expression est alors dans la ligne de commande et on la stocke dans la variable `M` :

   `STO` $\triangleright$ `M`

Puisque $z = e^{i \cdot t}$ on tape :

$$\texttt{SUBST(M, Z} = \texttt{EXP(i} \times \texttt{t))}$$

la réponse est :

$$\frac{\texttt{EXP(i} \cdot \texttt{t)}^2 - 2 \cdot \texttt{EXP(i} \cdot \texttt{t)}}{2}$$

On linéarise ensuite l'expression, on utilise l'historique pour recopier l'expression précédente :

LIN(HIST ENTER) ENTER

la réponse est :

$$\frac{1}{2} \cdot \texttt{EXP(2} \cdot \texttt{i} \cdot \texttt{t)} + -1 \cdot \texttt{EXP(i} \cdot \texttt{t)}$$

A noter, qu'en recopiant l'expression, on la simplifie :

$\triangle$ ENTER ENTER

donne :

$$\frac{\texttt{EXP(2} \cdot \texttt{i} \cdot \texttt{t)} - 2 \cdot \texttt{EXP(i} \cdot \texttt{t)}}{2}$$

– On cherche maintenant la partie réelle de cette expression :

  RE(HIST ENTER) ENTER

  la réponse est :

  $$\frac{\texttt{COS(t} \cdot 2) - 2 \cdot \texttt{COS(t)}}{2}$$

  On définit alors la fonction $x(t)$, on tape :

  DEFINE (X(t) = HIST ENTER) ENTER

– On cherche ensuite la partie imaginaire (il faut remonter dans l'historique pour retrouver l'expression $\frac{\texttt{EXP(2} \cdot \texttt{i} \cdot \texttt{t)} - 2 \cdot \texttt{EXP(i} \cdot \texttt{t)}}{2}$), on tape :

  IM(HIST $\triangle$ $\triangle$ $\triangle$ $\triangle$ ENTER) ENTER

  la réponse est :

  $$\frac{\texttt{SIN(t} \cdot 2) - 2 \cdot \texttt{SIN(t)}}{2}$$

  On définit alors la fonction $y(t)$, on tape :

  DEFINE(Y(t) = HIST ENTER)ENTER

2. On cherche un axe de symétrie de Γ, pour cela on calcule $x(-t)$ et $y(-t)$ en tapant :

   X(−t) ENTER

   la réponse est :

   $$\frac{\texttt{COS(t} \cdot 2) - 2 \cdot \texttt{COS(t)}}{2}$$

On a donc : $x(-t) \ = \ x(t)$
puis :

$$\texttt{Y}(-\texttt{t})\ \texttt{ENTER}$$

la réponse est :

$$\frac{-\texttt{SIN}(\texttt{t}\cdot 2)+2\cdot\texttt{SIN}(\texttt{t})}{2}$$

On a donc : $y(-t) \ = \ -y(t)$
Si $M_1(x(t), y(t))$ est sur Γ, $M_2(x(-t), y(-t)$ est aussi sur Γ.
On vient de monter que M_1 et M_2 sont symétriques par rapport à Ox, donc on en déduit que l'axe Ox est un axe de symétrie de Γ.

3. Calcul de $x'(t)$:
   On tape :

$$\texttt{DERIV(X(t),t)}$$

   la réponse est :

$$\frac{2\cdot(-2\cdot\texttt{SIN}(\texttt{t}\cdot 2))-2\cdot(-\texttt{SIN}(\texttt{t}))}{4}$$

   après simplification ($\triangle$ `ENTER ENTER`) :

$$-(\texttt{SIN}(\texttt{t}\cdot 2)-\texttt{SIN}(\texttt{t}))$$

   On développe l'expression (transformation de $\texttt{SIN}(2\cdot\texttt{t})$), on tape :

   `TEXPAND( HIST ENTER) ENTER`

   la réponse est :

$$-(\texttt{SIN}(\texttt{t})\cdot 2\cdot\texttt{COS}(\texttt{t})-\texttt{SIN}(\texttt{t}))$$

   puis on factorise :

   `FACTOR( HIST ENTER) ENTER`

   la réponse est :

$$-\texttt{SIN}(\texttt{t})\cdot(2\cdot\texttt{COS}(\texttt{t})-1)$$

   On peut alors définir la fonction $x'(t)$, on tape :

   `DEFINE( X1(t) = HIST ENTER) ENTER`

4. Calcul de $y'(t)$:
   On tape :

$$\texttt{DERIV(Y(t),t)}$$

   la réponse est :

$$\frac{2\cdot(2\cdot\texttt{COS}(\texttt{t}\cdot 2))-2\cdot\texttt{COS}(\texttt{t})}{4}$$

   après simplification ($\triangle$ `ENTER ENTER`) :

$$\texttt{COS}(\texttt{t}\cdot 2)-\texttt{COS}(\texttt{t})$$

   On développe l'expression (transformation de $\texttt{COS}(2\cdot\texttt{t})$), on tape :

   `TEXPAND( HIST ENTER) ENTER`

la réponse est :

$$2 \cdot \mathtt{COS(t)}^2 - 1 - \mathtt{COS(t)})$$

puis on factorise :

`FACTOR( HIST ENTER) ENTER`

la réponse est :

$$(\mathtt{COS(t)} - 1) \cdot (2 \cdot \mathtt{COS(t)} + 1)$$

On peut alors définir la fonction $y'(t)$, on tape :

`DEFINE( Y1(t) = HIST ENTER) ENTER`

5. Variations de $x(t)$ et de $y(t)$
   Pour cela on trace sur le même graphique $x(t)$ et $y(t)$, on tape :
   `shift-bleu F4 (2D/3D)` : la fenêtre `PLOT SETUP` s'ouvre.
   On choisit comme type `function` à l'aide de `choos` du bandeau.
   Puis on entre

   $$\{\mathtt{X(t)}, \mathtt{Y(t)}\}$$

   comme équation `EQ`, puis `t` comme paramètre indépendant, puis `ENTER`.
   Ensuite on tape, `shift-bleu F2 (WIN)`, pour régler les paramètres de la fenêtre.

6. Tracé de la courbe Γ :
   – Valeurs de $x(t)$ et de $y(t)$
     On trouve les valeurs de $x(t)$ et de $y(t)$ pour $t = 0, \frac{\pi}{3}, \frac{2\cdot\pi}{3}, \pi$ en tapant successivement :

     `X(0) ENTER`

     réponse : $\frac{-1}{2}$

     $\mathtt{X(}\pi \div 3\mathtt{)\ ENTER}$

     réponse : $\frac{-3}{4}$

     $\mathtt{X(}2 \times \pi \div 3\mathtt{)\ ENTER}$

     réponse : $\frac{1}{4}$

     $\mathtt{X(}\pi\mathtt{)\ ENTER}$

     réponse : $\frac{3}{2}$

     `Y(0) ENTER`

     réponse : 0

     $\mathtt{Y(}\pi \div 3\mathtt{)\ ENTER}$

     réponse : $\frac{-\sqrt{3}}{4}$

     $\mathtt{Y(}2 \times \pi \div 3\mathtt{)\ ENTER}$

     réponse : $\frac{-3\cdot\sqrt{3}}{4}$

     $\mathtt{Y(}\pi\mathtt{)\ ENTER}$

     réponse : 0

- Pente des tangentes ($m = \frac{y'(t)}{x'(t)}$)
  On trouve les valeurs de $\frac{y'(t)}{x'(t)}$ pour $t = 0, \frac{\pi}{3}, \frac{2\cdot\pi}{3}, \pi$ en tapant successivement :

$$\texttt{LIMIT(Y1(t)/X1(t)}, \texttt{t} = \texttt{0) ENTER}$$

  réponse : 0

$$\texttt{LIMIT(Y1(t)/X1(t)}, \texttt{t} = \pi \div \texttt{3) ENTER}$$

  réponse : ∞

$$\texttt{LIMIT(Y1(t)/X1(t)}, \texttt{t} = \texttt{2} \times \pi \div \texttt{3) ENTER}$$

  réponse : 0

$$\texttt{LIMIT(Y1(t)/X1(t)}, \texttt{t} = \pi\texttt{) ENTER}$$

  réponse : ∞
  Voici les variations de $x(t)$ et de $y(t)$

t	0		$\frac{\pi}{3}$		$\frac{2\pi}{3}$		π
$x'(t)$	0	$-$	0	$+$	$?$	$+$	0
$x(t)$	$\frac{-1}{2}$	$\downarrow$	$\frac{-3}{4}$	$\uparrow$	$\frac{1}{4}$	$\uparrow$	$\frac{3}{2}$
$y(t)$	0	$\downarrow$	$\frac{\sqrt{3}}{4}$	$\downarrow$	$\frac{-3\sqrt{3}}{4}$	$\uparrow$	0
$y'(t)$	0	$-$	$?$	$-$	0	$+$	$?$
m	0		∞		0		∞

- Courbe Γ :
  On fait ensuite le tracé de la courbe en paramétrique.
  On tape :
  `shift-bleu F4 (2D/3D)` et la fenêtre `PLOT SETUP` s'ouvre.
  On choisit comme type `parametric`, à l'aide de `choos` du bandeau. Puis on entre

$$\texttt{X(t)} + \texttt{i} \times \texttt{Y(t)}$$

  comme équation `EQ` , puis `t` comme paramètre independant, puis `ENTER`.
  Ensuite on tape `shift-bleu F2 (WIN)` pour régler les paramètres de la fenêtre

5.3 Exercice 2 (de spécialité)

On définit pour n entier naturel :

$$a_n = 4 \times 10^n - 1,\ b_n = 2 \times 10^n - 1 \text{ et } c_n = 2 \times 10^n + 1$$

On tape donc :

$$\texttt{DEFINE(A(N)} = \texttt{4} \cdot \texttt{10}^{\texttt{N}} - \texttt{1)}$$

$$\texttt{DEFINE(B(N)} = \texttt{2} \cdot \texttt{10}^{\texttt{N}} - \texttt{1)}$$

$$\texttt{DEFINE(C(N)} = \texttt{2} \cdot \texttt{10}^{\texttt{N}} + \texttt{1)}$$

1. – a) Calcul de $a_1, b_1, c_1, a_2, b_2, c_2, a_3, b_3, c_3$:
   Il suffit de taper :

   `A(1)`

   réponse 39

   `B(1)`

   réponse 19

   `C(1)`

   réponse 21

   `A(2)`

   réponse 399

   `B(2)`

   réponse 199

   `C(2)`

   réponse 201

   `A(3)`

   réponse 3999

   `B(3)`

   réponse 1999

   `C(3)`

   réponse 2001

   – b) nombre de chiffres et divisibilité
   Ici, la calculatrice n'est là que pour faire des essais pour différentes valeurs de n...
   On sait que les entiers n vérifiant :

$$10^n \leq n < 10^{n+1}$$

   ont $(n+1)$ chiffres dans l'écriture décimale.
   On a :

$$\begin{array}{rcccl} 3 \cdot 10^n & < & a_n & < & 4 \cdot 10^n \\ 10^n & < & b_n & < & 2 \cdot 10^n \\ 2 \cdot 10^n & < & c_n & < & 3 \cdot 10^n \end{array}$$

   donc $a_n,\ b_n,\ c_n$ ont $(n+1)$ chiffres dans l'écriture décimale.
   De plus $d_n = 10^n - 1$ est divisible par 9, car son écriture décimale ne comporte que des 9.
   On a

$$a_n = 3 \cdot 10^n \ + \ d_n$$

   et

$$c_n = 3 \cdot 10^n \ - \ d_n$$

   donc a_n et c_n sont divisibles par 3.

- c) b_3 est premier
  On tape :

  `ISPRIME?(B(3))`

  On obtient :

  `1.`

  ce qui veut dire `vrai`
  Pour montrer que $b_3 = 1999$ est premier, il suffit de tester si 1999 est divisible par tous les nombres premiers inférieurs ou égaux à $\sqrt{1999}$.
  Comme on a $1999 < 2025 = 45^2$, on teste la divisibilité de 1999 avec $n = 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41$.
  1999 n'etant divisible par aucun de ces nombres on en déduit que 1999 est premier.
- d) $a_n = b_n \times c_n$
  On tape :

  $\mathtt{B(N) \cdot C(N)}$

  On obtient :

  $\mathtt{4 \cdot (10^{N})^2 - 1}$

  qui est bien la valeur de a_n
  Décomposition en facteur premier de a_6
  On tape :

  `FACTOR(A(6))`

  On obtient :

  $3 \cdot 23 \cdot 29 \cdot 1999$
- e) b_n et c_n sont premiers entre eux.
  Ici, la calculatrice n'est là que pour faire des essais pour différentes valeurs de n...
  Pour montrer que c_n et b_n sont premiers entre eux il suffit de remarquer que :

  $$c_n = b_n + 2$$

  Ainsi, les diviseurs communs à c_n et b_n sont les diviseurs communs à b_n et 2 et sont aussi, les diviseurs communs à c_n et 2. b_n et 2 sont premiers entre eux car b_n est un nombre premier différent de 2. Donc

  $$PGCD(c_n,\ b_n) = PGCD(c_n,\ 2) = PGCD(b_n,\ 2) = 1$$

2. On considère l'équation :

$$b_3 \cdot x + c_3 \cdot y = 1$$

- a) Il y a au moins une solution car il s'agit de l'identité de Bézout.
  En effet, le théorème de Bézout dit :
  Si a et b sont premiers entre eux, il existe x et y vérifiant :

  $$a \cdot x + b \cdot y = 1$$

  Donc, l'équation :

  $$b_3 \cdot x + c_3 \cdot y = 1$$

  a au moins une solution.

- b)On tape :

  $$\texttt{IEGCD(B(3),C(3))}$$

  On obtient :

  $$\{\texttt{1},\texttt{1000},-\texttt{999}\}$$

  ce qui veut dire que l'on a :

  $$1 = b_3 \times 1000 + c_3 \times (-999)$$

  on a donc une solution particulière :
  $x = 1000,\ y = -999$.
  À la main, on écrit :
  $c_3 = b_3 + 2$ et $b_3 = 999 \times 2 + 1$
  donc, $b_3 = 999 \times (c_3 - b_3) + 1$ ainsi :

  $$b_3 \times 1000 + c_3 \times (-999) = 1$$

- c) Ici, la calculatrice ne peut pas trouver la solution générale.
  On a :

  $$b_3 \cdot x + c_3 \cdot y = 1$$

  et

  $$b_3 \times 1000 + c_3 \times (-999) = 1$$

  donc par soustraction, on a :

  $$b_3 \cdot (x - 1000) + c_3 \cdot (y + 999) = 0$$

  ou encore :

  $$b_3 \cdot (x - 1000) = -c_3 \cdot (y + 999)$$

  D'après le théorème de Gauss : c_3 est premier avec b_3 donc, c_3 divise $(x - 1000)$.
  Il existe donc $k \in Z$ tel que :

  $$(x - 1000) = k \times c_3$$

  et

  $$-(y + 999) = k \times b_3$$

  Réciproquement, soit

  $$x = 1000 + k \times c_3$$

  et

  $$y = -999 - k \times b_3 \quad \text{pour} \quad k \in Z$$

  On a :

  $$b_3 \cdot x + c_3 \cdot y = b_3 \times 1000 + c_3 \times (-999) = 1$$

  La solution générale est donc pour tout $k \in Z$:

  $$x = 1000 + k \times c_3$$

  $$y = -999 - k \times b_3$$

5.4 Exercice 2 (pas de spécialité)

On considère la suite

$$u_n = \int_0^2 \frac{2x+3}{x+2}\ e^{\frac{x}{n}}\ dx$$

1. – a) Variation de $g(x) = \frac{2x+3}{x+2}$ pour $x \in [0,2]$
   On tape :

   $$\mathtt{DEFINE(G(X)} = \frac{\mathtt{2X+3}}{\mathtt{X+2}}\mathtt{)}$$

   puis :

   $$\mathtt{TABVAR(G(X))}$$

   On obtient :

   $$\begin{array}{cccccc} -\infty & + & -2 & + & +\infty & X \\ 2 & \uparrow & \infty & \uparrow & 2 & F \end{array}$$

   La première ligne donne le signe de `f'(x)` selon x, et la deuxième ligne les variations de `f(x)`.
   On en déduit donc que $g(x)$ est croissante sur $[0,2]$.
   À noter que si on est en mode pas à pas (pour cela il faut appuyer sur `MODE` puis `cas` du bandeau et cocher `Step/Step` avec `chk` du bandeau puis `ok ok`), on obtient alors (quoiqu'il arrive la fonction est notée `F`) :

   $$\mathtt{F} =: \frac{\mathtt{2\cdot X+3}}{\mathtt{X+2}}$$

   puis, `ok` du bandeau

   $$\mathtt{F'} := \frac{\mathtt{2\cdot(X+2)-(2\cdot X+3)}}{\mathtt{SQ(X+2)}}$$

   puis, en se servant de la flèche $\bigtriangledown$ pour faire défiler l'écran

   $$\to \frac{\mathtt{1}}{\mathtt{(X+2)^2}}$$

   puis, `ok` du bandeau pour obtenir le tableau de variations.
   Si on n'est pas en mode pas à pas, on peut aussi demander le calcul de la dérivée en tapant :

   $$\mathtt{DERVX(G(X))}$$

   ce qui donne le calcul ci-dessus.
   On calcule $g(0)$ et $g(2)$, pour cela on tape :

   $$\mathtt{G(0)}$$

   réponse $\frac{3}{2}$

   $$\mathtt{G(2)}$$

   réponse $\frac{7}{4}$
   d'où, l'encadrement

   $$\frac{3}{2} \leq g(x) \leq \frac{7}{4} \quad \text{pour } x \in [0,2]$$

   .

- b) Là, la calculatrice ne peut rien ...il suffit de dire que

$$e^{\frac{x}{n}} \geq 0 \quad \text{pour } x \in [0,2]$$

pour montrer que, pour $x \in [0,2]$, on a :

$$\frac{3}{2}e^{\frac{x}{n}} \leq g(x)e^{\frac{x}{n}} \leq \frac{7}{4}e^{\frac{x}{n}}$$

- c) On intègre l'inégalité ci-dessus, on tape :

$$\int_0^2 \mathtt{e}^{\frac{\mathtt{X}}{\mathtt{N}}}\ \mathtt{dX}$$

On obtient :

$$\mathtt{N} \cdot \mathtt{e}^{\frac{2}{\mathtt{N}}} - \mathtt{N}$$

On en déduit donc :

$$\frac{3}{2}(ne^{\frac{2}{n}} - n) \leq u_n \leq \frac{7}{4}(ne^{\frac{2}{n}} - n)$$

Pour justifier le calcul précédent, il faut dire qu'une primitive de $e^{\frac{x}{n}}$ est $n \cdot e^{\frac{x}{n}}$.
Si on ne le sait pas, on peut toujours taper :

$$\mathtt{INTVX(EXP(X \div N))}$$

la réponse est : $\mathtt{N} \cdot \mathtt{e}^{\frac{\mathtt{X}}{\mathtt{N}}}$
- d) On cherche la limite de $(ne^{\frac{2}{n}} - n)$ quand $n \to +\infty$:

$$\mathtt{LIMIT(N \cdot EXP(2 \div N) - N\ ,\ N = +\infty)}$$

On obtient :

$$\mathtt{2}$$

Pour justifier ce résultat, il faut dire que :

$$\lim_{x \to 0} \frac{e^x - 1}{x} = 1$$

et donc que :

$$\lim_{n \to +\infty} \frac{e^{\frac{2}{n}} - 1}{\frac{2}{n}} = 1$$

ou encore :

$$\lim_{n \to +\infty} (e^{\frac{2}{n}} - 1) \cdot n = 2$$

Si L existe, en faisant tendre n vers $+\infty$ dans les inégalités de 1b), on obtient :

$$\frac{3}{2} \cdot 2 \leq L \leq \frac{7}{4} \cdot 2$$

2. – a) $g(x) = 2 - \frac{1}{x+2}$ et calcul de $I = \int_0^2 g(x)dx$
   On tape :

   $$\texttt{PROPFRAC(G(X))}$$

   On obtient

   $$2 - \frac{1}{\texttt{X}+2}$$

   Pour le calcul de l'intégrale I, on tape dans l'éditeur d'équations (touche `EQW`) :

   $$\int_0^2 \texttt{G(X)dX}$$

   On obtient :

   $$-(\texttt{LN}(2) - 4)$$

   À la main, on a $2x + 3 = 2(x + 2) - 1$ donc

   $$g(x) = 2 - \frac{1}{x+2}$$

   On intègre ensuite terme à terme entre 0 et 2, on obtient :

   $$\int_0^2 g(x)dx = [2x - \ln(x + 2)]_{x=0}^{x=2}$$

   c'est à dire, puisque $\ln 4 = 2\ln 2$:

   $$\int_0^2 g(x)dx = 4 - \ln 2$$

   – b) Là, la calculatrice ne peut rien...il suffit de dire que $e^{\frac{x}{n}}$ est croissante pour $x \in [0, 2]$, pour obtenir l'inégalité :

   $$1 \ \leq \ e^{\frac{x}{n}} \ \leq e^{\frac{2}{n}}$$

   puis par multiplication, $g(x)$ étant positif sur $[0, 2]$, on a :

   $$g(x) \ \leq \ g(x)e^{\frac{x}{n}} \ \leq g(x)e^{\frac{2}{n}}$$

   puis en intégrant on a :

   $$I \ \leq u_n \ \leq \ e^{\frac{2}{n}} I$$

   – c) Convergence de u_n
   On cherche la limite de $e^{\frac{2}{n}}$ quand $n \to +\infty$:

   $$\texttt{LIMIT(EXP(2} \div \texttt{N)} \ , \ \texttt{N} = +\infty)$$

   On obtient :

   $$1$$

   En effet, $\frac{2}{n}$ tend vers 0 lorsque n tend vers $+\infty$ donc, $e^{\frac{2}{n}}$ tend vers $e^0 = 1$ lorsque n tend vers $+\infty$.
   Lorsque n tend vers $+\infty$, u_n reste compris entre I et une quantité qui tend vers I (cf inégalités 2b)).
   Donc u_n converge et sa limite vaut I.
   On a donc montré que :

   $$L = I = 4 - \ln 2$$

5.5 Problème

Partie A

On considère la fonction f définie sur $]0,\ +\infty[$ par :

$$f(x) = \left(1 - \frac{1}{x}\right)(\ln x - 2)$$

On tape donc (en se servant de l'EQuationWriter) :

$$\texttt{DEFINE(F(X)} = (1 - 1 \div \texttt{X}) \times (\texttt{LN(X)} - 2))$$

Voici le détail de ce qu'il faut taper($\triangleleft$ représente le curseur) :

$$\texttt{DEFINE(F(X)} = \triangleleft)$$

Pour rentrer l'expression, en se servant de l'éditeur d'équations, on appuie sur la touche `EQW`.
On est ainsi dans l'éditeur d'équations, on tape :

$$1\ -\ 1\ \div\ \texttt{X} \triangleright\ \triangleright\ \triangleright\ \times \texttt{LN(X)}\ \triangleright\ -2\ \texttt{ENTER}$$

Ce qui donne dans la ligne de commande :

$$\texttt{DEFINE(F(X)} = (1 - \frac{1}{\texttt{X}}) \cdot (\texttt{LN(X)} - 2))$$

puis, `ENTER` pour valider.
`F` s'inscrit dans le bandeau des variables et `NOVAL` s'affiche à l'écran.
On vérifie en tapant `F(X)`, on obtient :

$$\frac{(\texttt{X} - 1) \cdot \texttt{LN(X)} - (2 \cdot \texttt{X} - 2)}{\texttt{X}}$$

1. Limite de f en $+\infty$ et 0.
   On tape :
   $$\texttt{LIMIT(F(X)}, +\infty)$$
   réponse $+\infty$
   puis,
   $$\texttt{LIMIT(F(X)}, 0)$$
   réponse ∞
2. Calcul de $f'(x)$.
   On tape :
   $$\texttt{DERVX(F(X))}$$
   On obtient :
   $$\frac{\texttt{LN(X)} + \texttt{X} - 3}{\texttt{X}^2}$$
3. $u(x) = \ln x + x - 3$
   On tape donc :
   $$\texttt{DEFINE(U(X)} = \texttt{LN(X)} + \texttt{X} - 3)$$

- a) Variations de u.
  On tape :

  `TABVAR(U(X)`

  La calculatrice demande à passer en mode complexe : répondre `YES`
  On obtient :

  $$\begin{array}{ccccccccc} -\infty & + & -1 & - & 0 & + & +\infty & X \\ -\infty & \uparrow & i\pi-4 & \downarrow & -\infty & \uparrow & +\infty & F \end{array}$$

  ATTENTION!!!!
  Seule la partie du tableau concernant les $x > 0$ est à prendre en compte (pour $x < 0$ la calculatrice considère une fonction ln à valeurs complexes).
- b) $u(x) = 0$ possède une solution unique α dans $[2, \ 3]$.
  D'après a) u est croissante sur $]0, \ +\infty[$.
  On tape :

  `U(2) shift − rouge ENTER`

  réponse $-0.306...$
  puis,

  `U(3) shift − rouge ENTER`

  réponse 1.098...
  On calcule ensuite :

  `U(2.20) shift − rouge ENTER`

  réponse $-0.306...$
  et

  `U(2.21) shift − rouge ENTER`

  réponse $-0.306...$
  D'après le théorème des valeurs intermédiaires (u est croissante et continue sur $[2, \ 3]$, u s'annule une seule fois entre 2 et 3 ($u(2) < 0$ et $u(3) > 0$)).
  Donc, si on appelle α l'unique solution de u dans $[2, \ 3]$, on a :

  $$2.20 < \alpha < 2.21$$

  puisque $u(2.20) < 0$ et $u(2.21) > 0$.
- c) Signe de $u(x)$ sur $]0, \ +\infty[$
  Le signe de $u(x)$ se déduit du tableau de variation de u on a :

  $$\begin{cases} u(x) < 0 & \text{pour } x < \alpha \\ u(x) = 0 & \text{pour } x = \alpha \\ u(x) > 0 & \text{pour } x > \alpha \end{cases}$$

4. - a)Variations de f.
  On fait le tableau de variations à la main car la dérivée de f n'est pas rationnelle...et la calculatrice ne sait pas encore faire!
  Le signe de $f'(x)$ étant celui de $u(x)$ on a :

$f'(x)$	0	$-$	α	$+$	$+\infty$
$f(x)$	$+\infty$	$\downarrow$	$\frac{\alpha-1}{\alpha}(ln(\alpha)-2)$	$\uparrow$	$+\infty$

– b)$f(\alpha) = -\frac{[\alpha-1)^2}{\alpha}$
On a :
$u(\alpha) = 0$ donc $\ln(\alpha) = 3 - \alpha$
On tape dans l'EQuationWriter :

$$(1 - \frac{1}{A})(LN(A) - 2)$$

puis
on met en inverse vidéo LN(A),
on ouvre le menu ALG (shift-rouge 4),
on sélectionne le N°6 (SUBST),
on complète la commande $SUBST(LN(A), LN(A) = 3 - A)$
puis, ENTER ENTER
On obtient :

$$-\frac{A^2 - 2 \cdot A + 1}{A}$$

puis,

$$FACTOR(-\frac{A^2 - 2 \cdot A + 1}{A})$$

donne

$$-\frac{(A - 1)^2}{A}$$

On tape :

$$DERIV(-\frac{(A - 1)^2}{A}, A)$$

on obtient :

$$-\frac{(A^2 - 1)}{A^2}$$

La fonction $v(x) = -\frac{(x-1)^2}{x}$ est donc décroissante pour $x > 1$.
L'encadrement de $f(\alpha)$ s'obtient donc en calculant :
$v(2.21)$ et $v(2.20)$.
On tape :

$$-\frac{(1.21)^2}{2.21} \text{ shift} - \text{rouge ENTER}$$

réponse -0.662488

$$-\frac{(1.2)^2}{2.2} \text{ shift} - \text{rouge ENTER}$$

réponse $-0.65454...$
on a donc :

$$-0.663 < f(\alpha) < -0.654$$

qui est un encadrement à $9 \cdot 10^{-3}$ près $(0.663 - 0.654 = 9 \cdot 10^{-3}$
ou encore :

$$-0.67 < f(\alpha) < -0.65$$

qui est un encadrement à $2 \cdot 10^{-2}$ près $(0.67 - 0.65 = 2 \cdot 10^{-2}$

5. – a) Signe de f
   On remarque que $f(1) = 0$ et que $f(e^2) = 0$
   On tape :

   `F(1)`

   réponse : 0

   `F(EXP(2))`

   réponse : 0
   Voici les variations de f et le signe de $f(x)$:

$f'(x)$	0	−	1	−	α	+	e^2	+	$+\infty$
$f(x)$	$+\infty$	$\downarrow$	0	$\downarrow$	$\frac{\alpha-1}{\alpha}(ln(\alpha)-2)$	$\uparrow$	0	$\uparrow$	$+\infty$
$f(x)$	$+\infty$	+	0	−	$\simeq -0.66$	−	0	+	$+\infty$

   – b) Graphe C de f
   on ouvre la fenêtre `PLOT SETUP (shift_bleu F4)` on choisit `function` et pour `EQ F(X)` puis, on règle les paramètres de la fenêtre dans `WIN (shift_bleu F2)`

Partie B

Soit H la primitive de f sur $]0,\ +\infty[$ et Γ son graphe.
ATTENTION aux notations qui ne sont pas les mêmes que le texte!!!

1. – a)Variations de H
   Puisque $H'(x) = f(x)$ on a le tableau de variations :

$f(x)$	0	+	1	−	e^2	+	$+\infty$
$H(x)$	?	$\uparrow$	0	$\downarrow$	?	$\uparrow$	?

   – b)Les tangentes en $x = 1$ et $x = e^2$
   On a $f(1) = 0$ et $f(e^2) = 0$. Les tangentes à Γ aux points d'abscisses 1 et e^2 sont donc de pente nulle, les tangentes sont horizontales.

2. Calcul de $H(x)$

   – a) Calcul de $\int_1^x \ln t dt$
   On tape dans l'éditeur d'équations :

$$\int_1^{\mathtt{X}} \mathtt{LN(T)dT}$$

   On obtient :

$$\mathtt{X} \cdot \mathtt{LN(X)} - (\mathtt{X} - 1)$$

   On peut aussi demander la primitive de $\ln x$, on tape :

   `INTVX(LN(X))`

   On obtient :

$$\mathtt{X} \cdot \mathtt{LN(X)} - \mathtt{X}$$

   À la main on pose $u = \ln(t)$ et $dv = dt$ donc $du = \frac{dt}{t}$ et $v = t$ on a :

$$\int_1^x \ln t\ dt = [t \cdot \ln t]_{t=1}^{t=x} - \int_1^x t \cdot \frac{dt}{t} = x \cdot \ln x - \int_1^x dt = x \cdot \ln x - (x-1)$$

– b) Il suffit de développer l'expression de $f(x)$ pour obtenir :

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

– c) Expression de $H(x)$
On tape (en se servant de l'EQuationWriter) :

$$\texttt{DEFINE(H(X)} = \int_{1}^{\texttt{X}} \texttt{F(T)dT)}$$

puis,

`H(X) ENTER`

On obtient :

$$-\frac{\texttt{LN(X)}^2 - (2 \cdot \texttt{X} + 4) \cdot \texttt{LN(X)} + 6 \cdot \texttt{X} - 6}{2}$$

On fait le calcul à la main en intégrant terme à terme l'expression de $f(x)$ trouvée en 2b), on a :

$$\int_1^x \ln t \, dt = x \ln x - x + 1$$

$$-\int_1^x \frac{\ln t}{t} \, dt = -\frac{(\ln x)^2}{2}$$

$$\int_1^x \frac{2}{t} \, dt = 2 \ln x$$

d'où :

$$H(x) = x \ln x - x + 1 - \frac{(\ln x)^2}{2} + 2 \ln x - 2x + 2$$

$$H(x) = -\frac{(\ln x)^2}{2} + (x+2) \ln x - 3x + 3$$

3. – a) On tape

`LIMIT(LN(X)/X, X = 0)`

réponse 0
D'après le cours, on sait que "x l'emporte sur $\ln x$ " d'où le résultat !
On tape :

`LIMIT(H(X), X = 0)`

réponse $-\infty$
On a :

$$H(x) = x \ln x + \ln x \frac{(4 - \ln x)}{2} - 3x + 3$$

Quand x tend vers 0, le premier terme de $H(x)$ tend vers 0, son deuxième terme tend vers $-\infty$ (car $\ln x$ tend vers $-\infty$ quand x tend vers 0) et ses derniers termes tendent vers 3.
Donc

$$\lim_{x \to 0} H(x) = -\infty$$

- b) On tape :

  $$\mathtt{LIMIT(H(X), X = +\infty)}$$

  réponse $+\infty$
  On met $x \ln x$ en facteur dans le début de $H(x)$ on obtient :

  $$H(x) = x \ln x(1 - \frac{\ln x}{2x} + \frac{2}{x} - \frac{3}{\ln x}) + 3$$

  $$\lim_{x \to +\infty} x \ln x = +\infty$$

  et la parenthèse tend vers 1 quand x tend vers $+\infty$, donc

  $$\lim_{x \to +\infty} H(x) = +\infty$$

- c)Variations de $H(x)$
  On calcule $H(e^2)$
  On tape :

  $$\mathtt{H(EXP(2))}$$

  On trouve l'expression obtenue en remplaçant dans `H(X)`, `X` par `EXP(2)`.
  On recopie cette expression pour la simplifier ($\triangle$ `ENTER ENTER`) et on obtient :

  $$-(\mathtt{EXP(2)} - 5)$$

  puis,

  $$\triangle \ \mathtt{ENTER\ shift - rouge\ ENTER}$$

  réponse -2.38905...
  On a ainsi une valeur approchée de $H(e^2)$.
  On reprend le tableau fait en 1a)

$f(x)$	0	$+$	1	$-$	e^2	$+$	$+\infty$
$H(x)$	$-\infty$	$\uparrow$	0	$\downarrow$	$\simeq -2.39$	$\uparrow$	$+\infty$

- d) Graphe C de f et graphe Γ de H
  on ouvre la fenêtre `PLOT SETUP (shift_bleu F4)` on choisit `function` et pour `EQ {F(X),H(X)}` puis, on régle les paramètres de la fenêtre dans `WIN (shift_bleu F2)`

4. Calcul d'une aire
   On a calculé :

   $$\int_1^{e^2} f(x)dx$$

   On a obtenu :

   $$-(\mathtt{EXP(2)} - 5)$$

   la courbe étant sous l'axe des x entre 1 et e^2, cette intégrale est négative.
   Les unités étant de 2cm, l'aire en cm ^{2} est donc égale à $-4H(e^2)$ cm^2, soit :

   $$4 \cdot (\mathtt{EXP(2)} - 5) \ \ \mathrm{cm}^2$$

   soit

   $$A = (4e^2 - 20) \ \ \mathrm{cm}^2$$

   ou encore

   $$9.55 \ \ \mathrm{cm}^2 \ < \ A \ < \ 9.56 \ \ \mathrm{cm}^2$$

5.6 Conclusion

On voit qu'un bon maniement de la calculatrice HP49G permet de faire une bonne partie des questions...
Il faut cependant noter, qu'en arithmétique il faut faire plus de raisonnements : la calculatrice permet alors de faire des vérifications....

Chapitre 6

Programmation

6.1 Implémentation en mode Algébrique

6.1.1 Comment éditer un programme

Un programme s'écrit dans la ligne de commande entre les délimiteurs $\ll \gg$

6.1.2 Comment sauver un programme

Il suffit de faire suivre le dernier $\gg$ par :

`STO ▷ NOMDUPROGRAMME`

6.1.3 Comment corriger un programme

Si la syntaxe est mauvaise, la machine vous met automatiquement le curseur là où le compilateur a détecté l'erreur. Il suffit donc de corriger !!!
Si l'erreur est détectée au cours de l'exécution du programme il faut taper :
`VISIT('NOMDUPROGRAMME')` qui édite votre programme.
On corrige, puis `ENTER` sauve votre programme corrigé.

6.1.4 Comment exécuter un programme

Si le programme n'a pas de paramètres, il suffit de taper son nom dans la ligne de commande.
S'il y a des paramètres, on fait suivre le nom du programme de parenthèses dans lesquelles on met les valeurs des paramètres séparées par une virgule.
Exemple : `PGCD(45,75)`

6.1.5 Comment améliorer puis sauver sous un autre nom un programme

On tape :
`RCL('NOMDUPROGRAMME')` puis `edit` du bandeau.
On fait les améliorations et on fait suivre le dernier $\gg$ par :

`STO ▷ NOUVEAUNOM`

6.2 Les commentaires

Il faut prendre l'habitude de commenter ses programmes.
En algorithmique un commentaire commence par `//` et se termine par un passage à la ligne.
Pour la `HP49G`, un commentaire commence par `@` et se termine par un passage à la ligne ou, est entouré de deux `@`.
Le caractère `@` est obtenu en tapant `shift-rouge ENTER`
Attention !!! le compilateur efface les commentaires... donc pour garder vos commentaires, il faut écrire votre programme sous la forme d'un texte qu'il faut ensuite compiler avec `STR` → ce qui complique un peu...

6.3 Les variables

6.3.1 Leurs noms

Ce sont les endroits où l'on peut stocker des valeurs, des nombres, des expressions, des objets.
Avec la `HP49G`, on peut utiliser des noms ayant jusqu'à 8 caractères.

6.3.2 Notion de variables locales

La `HP49G` peut utiliser des variables locales.
Les variables locales sont déclarées et initialisées (initialisation obligatoire !) grâce à → (`shift − rouge 0`)

En mode `RPN`, on peut définir et initialiser plusieurs variables locales à la fois, exemple :

$$\ll 1\ 2\ \rightarrow\ \mathtt{A\ B} \ll \mathtt{corpsdusousprogramme} \gg\gg$$

En mode `Algébrique`, chaque déclaration doit être suivie par un sous programme (délimiteurs ≪≫)
La flèche doit être entourée d'espaces (ces espaces sont mis automatiquement quand on n'est pas en mode α).
Exemple :

$$\ll 3.14\ \rightarrow\ \mathtt{PI} \ll 2 * \mathtt{PI} * \mathtt{R} \gg\gg \mathtt{STO} \triangleright\ \mathtt{PER}$$

Dans cet exemple, on a écrit le programme `PER`.
`PI` est une variable locale qui est déclarée et affectée par `3.14 → PI`. Cette variable est locale pour le programme qui suit sa déclaration (ici $\ll 2 * \mathtt{PI} * \mathtt{R} \gg$).
Par contre, `R` est une variable globale (qui doit exister avant l'exécution du programme `PER`). Si, au cours d'un programme, on veut stocker une valeur dans une variable (locale ou globale) il faut bien sûr utiliser `STO▷`.

6.3.3 Notion de paramètres

Quand on écrit une fonction il est possible d'utiliser des paramètres.
Par exemple si on veut que `R` soit le paramètre de la fonction `PER` on écrit :

$$\ll\ \rightarrow\ \mathtt{R} \ll 3.14\ \rightarrow\ \mathtt{PI} \ll 2 * \mathtt{PI} * \mathtt{R} \gg\gg\gg \mathtt{STO} \triangleright\ \mathtt{PER}$$

Ce paramètre R se comporte alors comme une variable locale, la seule différence est qu'il est initialisé lors de l'appel de la fonction.
L'exécution se fait en demandant par exemple : `PER(5)`.
La syntaxe en RPN et en `Algébrique` est :

$$\ll \ \rightarrow \ \mathtt{A\ B} \ll \ \ldots$$

pour écrire une fonction a deux paramètres.

6.4 Les Entrées

6.4.1 Traduction en Algorithmique

Pour que l'utilisateur puisse entrer une valeur dans la variable A au cours de l'exécution d'un programme, on écrira, en algorithmique : `saisir A`
Pour entrer des valeurs dans A et B on écrira : `saisir A,B`

6.4.2 Traduction HP49G mode RPN

```
'A' PROMPTSTO
```
ou
```
"A" "" INPUT STR-> EVAL 'A' STO
```

6.4.3 Traduction HP49G mode Algébrique

Pour entrer une variable locale A, on tape :
....0 $\rightarrow$ A $\ll$ PROMPTSTO('A')... $\gg$ ou
....0 $\rightarrow$ A $\ll$ PROMPTSTO('A'); 0 $\rightarrow$ B $\ll$ PROMPTSTO('B')..... $\gg$ $\gg$
Si on tape seulement :
`...PROMPTSTO('A')` alors A est une variable globale ou
`...PROMPTSTO('A') ; PROMPTSTO('B')` , alors A et B sont des variables globales.
On peut aussi utiliser INPUT pour créer une variable locale A on utilise, si A doit contenir une chaine de caractères :

$$\mathtt{INPUT}(''\mathtt{A} ='','' \ '') \ \rightarrow \ \mathtt{A}$$

ou si A doit contenir un nombre :
.....INPUT("A ="," ") $\rightarrow$ A
$\ll$ OBJ $\rightarrow$ (A) $\rightarrow$ A
$\ll$ ..... $\gg$
$\gg$

6.5 Les Sorties

6.5.1 Traduction en Algorithmique

En algorithmique on écrit :
```
Afficher "A=",A
```

6.5.2 Traduction HP49G mode RPN

En général on affiche simplement les résultats sur la pile pour une réutilisation éventuelle, on écrira simplement :
`A B`
On peut aussi afficher le résultat tagué, on écrira alors :
`A "A=" ->TAG`
`HALT` arrête le programme.

6.5.3 Traduction HP49G mode Algébrique

Seul le dernier résultat s'inscrit dans l'historique.
Pour des résultats intermédiaires on tape :

$$\mathtt{DISP}(\text{"}\mathtt{A} = \text{"} + \mathtt{A}, 3)$$

3 représente le numéro de la ligne
ou

$$\mathtt{MSGBOX}(\text{"}\mathtt{A} = \text{"}+ \rightarrow \mathtt{STR}(\mathtt{A}))$$

`CLEAR` efface l'écran.
`FREEZE(7)` arrête le programme et permet de visualiser les 7 lignes de l'affichage.

6.6 La séquence d'instructions ou action

Une action est une séquence d'une ou plusieurs instructions.

6.6.1 Traduction en Algorithmique

En langage algorithmique, on utilisera l'espace ou le passage à la ligne pour terminer une instruction.

6.6.2 Traduction HP49G mode RPN

Comme en algorithmique, il n'y a pas de séparateur : on utilise soit l'espace soit le retour à la ligne.

6.6.3 Traduction HP49G mode Algébrique

Pour la `HP49G` en mode algébrique, le `;` est un séparateur d' instructions.
Le `;` s'obtient en tapant simultanément `shift-rouge SPC`.

6.7 L'instruction d'affectation

L'affectation est utilisée pour stocker une valeur ou une expression dans une variable.

6.7.1 Traduction en Algorithmique

En algorithmique on écrira par exemple :
`2*A->B` pour stocker `2*A` dans `B`

6.7.2 Traduction HP49G mode RPN

Avec la HP49G mode RPN, il faut utiliser la notation postfixée et la commande `STO` :
`2 A * 'B' STO`

6.7.3 Traduction HP49G mode Algébrique

Pour la `HP49G` en mode algébrique, on utilise la touche `STO` qui se traduit à l'écran de la calculatrice par : ▷ (on le notera ici, `STO▷`).

6.8 Les instructions conditionnelles

6.8.1 Traduction en Algorithmique

`Si` *`condition`* `alors` *`action`* `fsi`
`Si` *`condition`* `alors` *`action1`* `sinon` *`action2`* `fsi`
Exemple :
`Si A = 10 ou A < B alors B-A->B sinon A-B->A fsi`

6.8.2 Traduction HP49G mode RPN

`IF` *`condition`* `THEN` *`action`* `END`
`IF` *`condition`* `THEN` *`action1`* `ELSE` *`action2`* `END`
Attention : on utilise la notation postfixée et == pour traduire la condition d'égalité.
On écrit pour traduire l'exemple :
`IF A 10 == A B < OR THEN B A - 'B' STO ELSE A B - 'A' STO END`
on peut aussi écrire
`IF 'A==10' 'A < B' OR THEN ...`

6.8.3 Traduction HP49G mode Algébrique

`IF` *`condition`* `THEN` *`action`* `END`
`IF` *`condition`* `THEN` *`action1`* `ELSE` *`action2`* `END`
Exemple : Attention au == pour traduire la condition d'égalité !
`IF A == 10 OR A < B THEN B-A STO▷ B ELSE A-B STO▷ A END`

6.9 Les instructions Pour

6.9.1 Traduction en Algorithmique

`Pour I de A à B faire` *`action`* `fpour`
`Pour I de A à B (pas P) faire` *`action`* `fpour`

6.9.2 Traduction HP49G mode RPN

`A B FOR I` *`action`* `NEXT`
`A B FOR I P STEP` *`action`* `NEXT`

6.9.3 Traduction HP49G mode Algébrique

`FOR (I , A , B)` *action* `NEXT`
`FOR (I , A , B )` *action* `STEP P`

Il n'y a pas besoin de déclarer la variable `I`.
`I` est déclaré et initialisé automatiquement par : `FOR (I,.,.)...`

6.10 L'instruction Tant que

6.10.1 Traduction en Algorithmique

`Tant que` *condition* `faire` *action* `ftantque`

6.10.2 Traduction HP49G mode RPN

`WHILE` *condition* `REPEAT` *action* `END`

6.10.3 Traduction HP49G mode Algébrique

`WHILE` *condition* `REPEAT` *action* `END`

6.11 Les conditions ou expressions booléennes

Une condition est une fonction qui a comme valeur un booléen, à savoir elle est soit `vraie` soit `fausse`.

6.11.1 Traduction en Algorithmique

Pour exprimer une condition simple on utilise les opérateurs :
`= > >` $\leq$ $\geq$ $\neq$

6.11.2 Traduction HP49G mode RPN

Attention, on utilise la notation postfixée et $==$ pour traduire la condition d'égalité.

6.11.3 Traduction HP49G mode Algébrique

Attention, pour la calculatrice `HP49G`, l'égalité se traduit par :
`==`

6.12 Les opérateurs logiques

6.12.1 Traduction en Algorithmique

Pour traduire des conditions complexes, on utilise les opérateurs logiques :
`ou et non`

6.12.2 Traduction HP49G mode RPN

Attention, on utilise la notation postfixée.
`ou` `et` `non` se traduisent par `OR AND NOT`

6.12.3 Traduction HP49G mode Algébrique

`ou` `et` `non` se traduisent sur la `HP49G` par `OR AND NOT`

6.13 Les fonctions

Dans une fonction on ne fait pas de saisie de données :
on utilise des paramètres qui seront initialisés lors de l'appel.
Dans une fonction, on veut pouvoir réutiliser le résultat :
on n'utilise pas la commande `affichage` mais la commande `résultat`.

6.13.1 Traduction en Algorithmique

On écrit par exemple, en algorithmique :

```
fonction addition(A,B)
résultat A+B
ffonction
```

Cela signifie que :
-Si on fait exécuter la fonction, le résultat sera affiché.
-On peut utiliser la fonction dans une expression.

6.13.2 Traduction HP49G mode RPN

Pour la HP49G mode RPN, on suppose que les arguments de la fonction sont mis sur la pile avant l'appel de la fonction.

Dans l'écriture de la fonction les arguments sont des variables locales qui seront initialisées par les éléments mis sur la pile. Le résultat de la fonction est alors mis sur la pile.
L'exemple se traduit par :

```
≪→ A B
 ≪ A B + ≫
≫
'ADDITION' STO
```

puis on met les valeurs de A et B au niveau 2 et 1 de la pile, après l'exécution d'`ADDITION` leur somme se trouvera au niveau 1 de la pile.

6.13.3 Traduction HP49G mode Algébrique

L'exemple se traduit par :

```
≪→ A B
 ≪ A + B ≫
≫
STO ▷ ADDITION
```

Puis, on tape :

$$\texttt{ADDITION}(4,5)$$

6.14 Les listes

6.14.1 Traduction en Algorithmique

En algorithmique, on utilise les { } pour délimiter une liste.
Par exemple {} désigne la liste vide et {1, 2, 3} est une liste de 3 éléments.
Le + sera utilisé pour concaténer 2 listes ou une liste et un élément ou un élément et une liste :
`{1, 2, 3}->TAB`
`TAB + 4 ->TAB` (maintenant `TAB` désigne {1, 2, 3, 4})
`TAB[2]` désigne le deuxième élément de `TAB` ici 2.

6.14.2 Traduction HP49G mode RPN

Ici les listes peuvent avoir des longueurs non définies à l'avance.
On utilise les {} pour délimiter une liste.
Par exemple {1 2 3} est une liste de 3 éléments et {} désigne la liste vide.
On obtient le `P`ième élément de `L` sur la pile avec :
`L P GET` ou `'L' P GET`
Si on veut modifier le `P`ième élément de `L` (par exemple le mettre à 0) on écrira :
`'L' P 0 PUT` ou `L P 0 PUT 'L' STO`
En effet `L P 0 PUT` renvoie sur la pile la liste modifiée alors que :
`'L' P 0 PUT` modifie la liste `L` .
Pour concaténer deux listes ou une liste et un élément on utilisera le +

6.14.3 Traduction HP49G mode Algébrique

Pour la `HP49G` en mode algébrique, les listes peuvent avoir des longueurs non définies à l'avance.
On utilise les {} pour délimiter une liste.
Par exemple {1,2,3} est une liste de 3 éléments et {} désigne la liste vide.
On obtient le `P`ième élément de `L` sur la pile avec :
`L[P]` ou `GET (L, P )`
Si on veut modifier le `P`ième élément de `L` (par exemple le mettre à 0) on écrira :
`PUT(L, P, 0) STO▷ L`
ou
`PUT('L', P, 0)`
En effet `PUT(L, P, 0)` renvoie la liste modifiée (sans modifier `L`) alors que :
`PUT ('L', P, 0)` modifie la liste `L`.
Pour concaténer deux listes ou une liste et un élément on utilisera le `+`.
La commande `SEQ` permet de constituer une liste, on tape :

$$\mathtt{SEQ}('\mathtt{X}*\mathtt{X}','\mathtt{X}',4,10,1)$$

on obtient :

$$\{16,25,36,49,64,81,100\}$$

6.15 Un exemple : le crible d'Eratosthéne

6.15.1 Déscription

Pour trouver les nombres premiers inférieurs ou égaux à N :

1. On écrit les nombres de 1 à N dans une liste.
2. On barre 1 et on met 2 dans la case P.
   Si $P.P \leq N$ il faut traiter les éléments de P à N.
3. On barre tous les multiples de P à partir de $P.P$.
4. On augmente P de 1
   Si $P.P$ est inférieur ou égal à N, il reste à traiter les éléments non barrés de P à N.
5. On appelle P le plus petit élément non barré de la liste.
6. On refait les points 3 4 5 tant que $P.P$ reste inférieur ou égal à N.

6.15.2 Ecriture de l'algorithme

```
Fonction crible(N)
local TAB PREM I P
// TAB et PREM sont des listes
{} ->TAB
{} ->PREM
pour I de 2 à N faire
  TAB+I -> TAB
fpour
0 +TAB -> TAB
2 -> P
// On a fait les points 1 et 2
//barrer 1 a été réalisé en le remplaçant par 0
//TAB est la liste 0 2 3 4 ...N
tant que P*P ≤ N faire
  pour I de P à int(N/P) faire
    0 -> TAB[I*P]
  fpour
// On a barré tous les multiples de P à partir de P*P
  P+1 -> P
//On cherche le plus petit nombre <= N non barré entre P et N
  tant que (P*P ≤ N) et (TAB[P]=0) faire
    P+1 -> P
  ftantque
ftantque
//on écrit le résultat dans une liste PREM
pour I de 2 à N faire
  si TAB[I] ≠ 0 alors
    PREM +I -> PREM
  fsi
fpour
résultat: PREM
```

6.15.3 Traduction HP49G mode RPN

Voici le programme CRIBLE :
N est le paramètre qui doit être mis sur la pile.
Les variables locales sont :
P et I qui sont des entiers ,
TA et PREM qui sont des listes.

```
≪ {} {} 2 1 → N TA PREM P I
  ≪ 0 'X' 'X' 2 N 1 SEQ + 'TA' STO
    WHILE P P * N ≤ REPEAT
       P N P / FLOOR FOR I
          TA I P * 0 PUT 'TA' STO
       NEXT
       1 'P' STO+
       WHILE P P * N ≤ TA P GET 0 == AND REPEAT
          1 'P' STO+
       END
    END
    2 N FOR I
       IF TA I GET 0 ≠ THEN
          I 'PREM' STO+
       END
    NEXT
    PREM
  ≫
≫
```

6.15.4 Traduction HP49G mode Algébrique

Voici le programme CRIBLE écrit pour la `HP49G mode Algébrique` :
L'utilisateur doit taper par exemple :
`CRIBLE(100)`.

```
≪ → N
  ≪ 0+SEQ('I','I',1,N,1) → TA
    ≪ 2 → P
      ≪ WHILE P * P ≤ N REPEAT
         FOR (I , P , FLOOR(N/P))
           PUT('TA',I*P,0)
         NEXT ;
          P + 1 STO ▷ P;
         WHILE P*P ≤ N AND GET(TA,P) == 0 REPEAT
            P + 1 STO ▷ P
         END
      END ;
      {2} → PREM
        ≪ FOR (I, 3, N)
             IF TA(I) ≠ 0 THEN
                PREM + I STO ▷ PREM;
             END
```

```
            NEXT ;
            PREM
            ≫
         ≫
      ≫
   ≫
≫ STO ▷ CRIBLE
```

Chapitre 7

Programmes d'arithmétique

7.1 Calcul du PGCD par l'algorithme d'Euclide

Cet algorithme est basé sur la définition récursive du PGCD :

$$\begin{aligned} PGCD(A,0) &= A \\ PGCD(A,B) &= PGCD(B, A \ mod \ B) \ si \ B \neq 0 \end{aligned}$$

Voici la description de cet algorithme :
on effectue des divisions euclidiennes successives :

$$\begin{aligned} A = &\quad B \times Q_1 + R_1 \quad 0 \leq R_1 < B \\ B = &\quad R_1 \times Q_2 + R_2 \quad 0 \leq R_2 < R_1 \\ R_1 = &\quad R_2 \times Q_3 + R_3 \quad 0 \leq R_3 < R_2 \\ &\ldots\ldots. \end{aligned}$$

Après un nombre fini d'étapes, il existe un entier n tel que : $R_n = 0$.
on a alors :
$PGCD(A,B) = PGCD(B,R_1) = ...$
$PGCD(R_{n-1},R_n) = PGCD(R_{n-1},0) = R_{n-1}$

7.1.1 Traduction algorithmique

-Version itérative
Si B $\neq$ 0 on calcule R=A mod B, puis avec B dans le rôle de A (en mettant B dans A) et R dans le rôle de B (en mettant R dans B) on recommence jusqu'à ce que B=0, le PGCD est alors A.

```
Fonction PGCD(A,B)
Local R
tant que B ≠ 0 faire
  A mod B->R
  B->A
  R->B
ftantque
résultat A
ffonction
```

-Version récursive
On écrit simplement la définition récursive vue plus haut.

```
Fonction PGCD(A,B)
Si B ≠ 0 alors
  résultat PGCD(B,A mod B)
  sinon
  résultat A
fsi
ffonction
```

7.1.2 Traduction HP49G mode RPN

-Version itérative

```
≪ 0 → A B R
   ≪ WHILE B 0 ≠ REPEAT
        A B MOD 'R' STO
        B 'A' STO
        R 'B' STO
     END
     A
   ≫
≫
```

-Version récursive

```
≪ 0 → A B
   ≪ IF B 0 ≠ THEN
        B A B MOD PGCDR
     ELSE
        A
     END
   ≫
≫
```

Puis on stocke ce programme dans la variable 'PGCDR'.

7.1.3 Traduction HP49G mode Algébrique

On tape :
-version itérative

```
≪ → A,B
   ≪ 0 → R
      ≪ WHILE B ≠ 0 REPEAT
           A MOD B STO ▷ R;
           B STO ▷ A;
           R STO ▷ B
        END;
        A
      ≫
   ≫
≫ STO ▷ PGCD
```

Puis par exemple, `PGCD(45,75)` pour l'exécuter.

-version récursive

```
≪ → A,B
  ≪ IF B ≠ 0 THEN
       PGCDR(B,A MOD B)
     ELSE
       A
     END
  ≫
≫ STO ▷ PGCDR
```

Puis par exemple, `PGCDR(45,75)` pour l'exécuter.
Remarque :
Si on utilise la fonction du calcul symbolique `IREMAINDER` à la place de `MOD` dans les programmes précédents, `PGCD` (ou `PGCDR`) peut alors avoir comme paramètres des entiers de Gauss.

7.2 Identité de Bézout par l'algorithme d'Euclide

Dans ce paragraphe la fonction Bezout(A,B) est égale à la liste {U, V, PGCD(A,B)} où U et V vérifient :
$A \times U + B \times V = PGCD(A, B)$.

7.2.1 Version itérative sans les listes

L'algorithme d'Euclide permet de trouver un couple U et V vérifiant :
$A \times U + B \times V = PGCD(A, B)$
En effet, si on note A_0 etB_0 les valeurs de A et de B du début on a :

$$\begin{aligned} A &= A_0 \times U + B_0 \times V \quad \text{avec} \quad U = 1 \quad \text{et} \quad V = 0 \\ B &= A_0 \times W + B_0 \times X \quad \text{avec} \quad W = 0 \quad \text{et} \quad X = 1 \end{aligned}$$

Puis on fait évoluer A, B, U, V, W, X de façon que les deux relations ci-dessus soient toujours verifiées.
Si :
$A = B \times Q + R \quad 0 \leq R < B \quad (R = A \; mod \; B \; et \; Q = E(A/B))$
On écrit alors :

$$\begin{aligned} &R = A - B \times Q = A_0 \times (U - W \times Q) + B_0 \times (V - X \times Q) = \\ &A_0 \times S + B_0 \times T \quad \text{avec} \quad S = U - W \times Q \quad \text{et} \quad T = V - X \times Q \end{aligned}$$

Il reste alors à recommencer avec B dans le rôle de A (B->A W->U X->V) et R dans le rôle de B (R->B S->W T->X) d'où l'algorithme :

```
fonction Bezout (A,B)
local U,V,W,X,S,T,Q,R
1->U 0->V 0->W 1->X
tant que B ≠ 0 faire
A mod B->R
E(A/B)->Q
//R=A-B*Q
U-W*Q->S
```

```
V-X*Q->T
B->A W->U X->V
R->B S->W T->X
ftantque
résultat {U, V, A}
ffonction
```

7.2.2 Version itérative avec les listes

On peut simplifier l'écriture de l'algorithme ci-dessus en utilisant moins de variables : pour cela on utilise des listes LA LB LR pour mémoriser les triplets {U, V, A} {W, X, B} et {S, T, R}.

```
fonction Bezout (A,B)
local LA LB LR
{1, 0, A}->LA
{0, 1, B}->LB
tant que LB[3] ≠ 0 faire
LA-LB*E(LA[3]/LB[3])->LR
LB->LA
LR->LB
ftantque
résultat LA
ffonction
```

7.2.3 Version récursive avec les listes

On peut définir récursivement la fonction Bezout par :
$Bezout(A, 0) = \{1,\ 0,\ A\}$
Si $B \neq 0$ il faut définir $Bezout(A, B)$ en fonction de $Bezout(B, R)$ lorsque $R = A - B \times Q$ et $Q = E(A/B)$.
On a :

$$Bezout(B, R) = LT = \{W, X, pgcd(B, R)\}$$
$$\text{avec} \quad W \times B + X \times R = pgcd(B, R)$$

Donc :

$$\begin{aligned} W \times B + X \times (A - B \times Q) &= pgcd(B, R) \quad \text{ou encore} \\ X \times A + (W - X \times Q) \times B &= pgcd(A, B). \end{aligned}$$

D'où si $B \neq 0$ et si $Bezout(B, R) = LT$ on a :
$Bezout(A, B) = \{LT[2],\ LT[1] - LT[2] \times Q,\ LT[3]\}$.

```
fonction Bezout (A,B)
local LT Q R
Si B ≠ 0 faire
```

```
E(A/B)->Q
A-B*Q->R
Bezout(B,R)->LT
Résultat {LT[2], LT[1]-LT[2]*Q, LT[3]}
sinon Résultat {1, 0, A}
fsi
ffonction
```

7.2.4 Traduction HP49G mode RPN

-Version itérative avec les listes

Au début A et B contiennent les deux nombres pour lesquels on cherche l'identité de Bézout, puis A et B désignent les listes LA et LB de l'algorithme.

```
≪ {} → A B R
  ≪ {1 0} 'A' STO+
    {0 1} 'B' STO+
    WHILE B 3 GET 0 ≠ REPEAT
      A B A 3 GET B 3 GET / FLOOR * - 'R' STO
      B 'A' STO
      R 'B' STO
    END
    A
      ≫
≫
```

-Version récursive avec les listes

```
≪ {} → A B T
  ≪ IF B 0 ≠ THEN
      B A B MOD BEZOUR 'T' STO
      T 2 GET DUP A B / FLOOR *
      T 1 GET SWAP -
      T 3 GET  + + +
    ELSE
      {1 0} A +
    END
  ≫
≫
```

Puis on stocke ce programme dans la variable BEZOUR

7.2.5 Traduction HP49G mode Algébrique

-Version itérative avec les listes

Au début A et B contiennent les deux nombres pour lesquels on cherche l'identité de Bézout, puis A et B désignent les listes LA et LB de l'algorithme.

```
≪ → A,B
  ≪ {} → R
    ≪ {1,0,A} STO ▷ A;
      {0,1,B} STO ▷ B;
      WHILE B[3] ≠ 0 REPEAT
        A − B ∗ FLOOR(A[3]/B[3]) STO ▷ R;
        B STO ▷ A;
```

```
         R STO ▷ B
      END
      A
   ≫
≫
≫ STO ▷ BEZOUT
```

Puis par exemple, `BEZOUT(45,75)` pour l'exécuter.

-Version récursive avec les listes

```
≪ → A,B
   ≪ {} → T
      ≪ IF B ≠ 0 THEN
            BEZOUR(B,A MOD B) STO ▷ T;
            {T[2],T[1] − T[2] * FLOOR(A/B),T[3]}
         ELSE
            {1,0,A}
         END
      ≫
   ≫
≫ STO ▷ BEZOUR
```

Puis par exemple, `BEZOUR(45,75)` pour l'exécuter.
Remarque :
Si on utilise la fonction du calcul symbolique `IREMAINDER` à la place de `MOD` dans les programmes précédents, `BEZOUT` peut alors avoir comme paramètres des entiers de Gauss.

7.3 Décomposition en facteurs premiers

7.3.1 Les algorithmes et leurs traductions

-Premier algorithme
On teste, pour tous les nombres D de 2 à N, la divisibilité de N par D.
Si D divise N, on cherche alors les diviseurs de N/D etc...
On met les diviseurs trouvés dans la liste FACT.

```
fonction facprem(N)
local D FACT
2 -> D
{} -> FACT
tant que N ≠ 1 faire
  si N mod D = 0 alors
    FACT + D -> FACT
    N/D -> N
   sinon
    D+1 -> D
 fsi
ftantque
résultat FACT
ffonction
```

- Première amélioration
On ne teste que les diviseurs D entre 2 et $E(\sqrt{N})$.

```
fonction facprem(N)
local D FACT
2 -> D
{} -> FACT
tant que D*D ≤ N faire
  si N mod D = 0 alors
    FACT + D -> FACT
    N/D-> N
   sinon
    D+1 -> D
 fsi
ftantque
FACT + N -> FACT
résultat FACT
ffonction
```

-Deuxième amélioration
On cherche si 2 divise N, puis on teste les diviseurs impairs D entre 3 et $E(\sqrt{N})$.
Dans la liste FACT, on fait suivre chaque diviseur par son exposant :
decomp(12)={2,2,3,1}.

```
fonction  facprem(N)
local K D FACT
{}->FACT
0 -> K
tant que N mod 2 = 0 faire
    K+1 -> K
    N/2 -> N
ftantque
si K ≠0 alors
    FACT + {2 K} -> FACT
fsi
3 ->D
tant que D*D ≤ N faire
  0 -> K
  tant que N mod D = 0 faire
    K+1 -> K
    N/D -> N
  ftantque
  si K ≠0 alors
    FACT + {D K} -> FACT
  fsi
  D+2 -> D
ftantque
si N ≠ 1 alors
```

```
FACT + {N 1} -> FACT
fsi
résultat FACT
ffonction
```

7.3.2 Traduction HP49G mode RPN

Voici la traduction de la deuxième amélioration :

```
« 0 3 {} → N K D FACT
  « WHILE N 2 MOD 0 == REPEAT
       1 'K' STO+
       'N' 2 STO/
    END
    IF K 0 ≠ THEN
       {2 K} 'FACT' STO
    END
    WHILE N D D * ≥ REPEAT
       0 'K' STO
       WHILE N D MOD 0 == REPEAT
          1 'K' STO+
          'N' D STO/
       END
       IF K 0 ≠ THEN
          {D K} 'FACT' STO+
       END
       2 'D' STO+
    END
    IF N 1 ≠ THEN
       {N 1} 'FACT' STO+
    END
  »
»
```

7.3.3 Traduction HP49G mode Algébrique

```
« → N
  « 0 → K
    « WHILE N MOD 2 == 0 REPEAT
         K + 1 STO▷ K;
         N/2 STO▷ N
      END;
      {} → FACTO
        « IF K ≠ 0 THEN
             FACTO + {2,K} STO▷ FACTO
          END;
          3 → D
            « WHILE D * D ≤ N REPEAT
                 0 STO▷ K;
                 WHILE N MOD D == 0 REPEAT
                    K + 1 STO▷ K;
```

```
                N/D STO ▷ N;
             END;
             IF K ≠ 0 THEN
                FACTO + {D, K} STO ▷ FACTO
             END;
             D + 2 STO ▷ D
          END;
          IF N ≠ 1 THEN
             FACTO + {N, 1} STO ▷ FACTO
          END;
          FACTO;
       ≫
    ≫
  ≫
 ≫
≫ STO ▷ FACTEUR
```

Puis par exemple, `FACTEUR(45)` pour l'exécuter.

7.4 Calcul de A^P *mod* N

7.4.1 Traduction Algorithmique

-Premier algorithme
On utilise deux variables locales PUIS et I.
On fait un programme itératif de façon qu'à chaque étape PUIS représente $A^I \pmod N$.

```
fonction puismod (A, P, N)
local PUIS, I
1->PUIS
pour I de 1 a P faire
  A*PUIS mod N ->PUIS
fpour
resultat PUIS
ffonction
```

-Deuxième algorithme
On utilise une seule variable locale PUI mais on fait varier P de façon qu'à chaque étape de l'itération on ait :
$resultat = PUI * A^P \pmod N$

```
fonction puismod (A, P, N)
local PUI
1->PUI
tant que  P>0  faire
  A*PUI mod N ->PUI
  P-1->P
ftantque
resultat PUI
ffonction
```

-Troisième algorithme
On peut aisément modifier ce programme en remarquant que :
$A^{2*P} = (A * A)^P$.
Donc quand P est pair on a la relation :
$PUI * A^P = PUI * (A * A)^{P/2} \pmod N$
et quand P est impair on a la relation :
$PUI * A^P = PUI * A * A^{P-1} \pmod N$.
On obtient alors un algorithme rapide de $A^P \pmod N$.

```
fonction puismod (A, P, N)
local PUI
1->PUI
tant que  P>0  faire
  si P mod 2 =0 alors
    P/2->P
    A*A mod N->A
  sinon
    A*PUI mod N ->PUI
    P-1->P
  fsi
ftantque
resultat PUI
ffonction
```

On peut remarquer que si P est impair P-1 est pair.
On peut donc écrire :

```
fonction puismod (A, P, N)
local PUI
1->PUI
tant que  P>0  faire
  si P mod 2 =1 alors
    A*PUI mod N ->PUI
    P-1->P
  fsi
P/2->P
A*A mod N->A
ftantque
resultat PUI
ffonction
```

-Programme récursif
On peut définir la puissance par les relations de récurrence :
$A^0 = 1 \quad A^{P+1} \pmod N = (A^P \pmod N)) * A \pmod N$

```
fonction puimod(A, P, N)
si P>0 alors
resultat puimod(A, P-1, N)*A mod N
sinon
resultat 1
fsi
ffonction
```

-Programme récursif rapide

```
fonction puimod(A, P, N)
si P>0 alors
  si P mod 2 =0 alors
    resultat puimod(A*A, P/2, N)
  sinon
    resultat puimod(A, P-1, N)*A mod N
  fsi
sinon
resultat 1
fsi
ffonction
```

7.4.2 Traduction HP49G mode RPN

L'utilisateur doit mettre sur la pile :
A, P, N pour obtenir $A^P mod\ N$.
Voici la traduction de l'algorithme rapide itératif :

```
≪ 1 → A P N PUI
   ≪ WHILE P 0 > REPEAT
         IF P 2 MOD 1 == THEN
            A PUI * N MOD 'PUI' STO
            'P' STO-
         END
         P 2 / 'P' STO
         A A * N MOD 'A' STO
      END
      PUI
  ≫
≫
```

Puis on stocke ce programme dans PUIMOD ('PUIMOD' STO)

7.4.3 Traduction HP49G mode Algébrique

Voici la traduction de l'algorithme rapide itératif :

```
≪ → A P N
   ≪ 1 → PUI
      ≪ WHILE P > 0 REPEAT
            IF P MOD 2 == 1 THEN
               A * PUI MOD N STO▷ PUI
               P - 1 STO▷ P;
            END;
            P/2 STO▷ P;
            A * A MOD N STO▷ A;
         END;
         PUI
      ≫
   ≫
≫ STO▷ PUIMOD
```

Puis par exemple, `PUIMOD(45,32,13)` pour l'exécuter.

7.5 La fonction estpremier

7.5.1 Traduction Algorithmique

- Premier algorithme

On va écrire un fonction booléenne de paramètre N, qui sera egale à VRAI quand N est premier et à FAUX sinon.

Pour cela, on cherche si N posséde un diviseur $\neq 1$ et $\leq$ à $E(\sqrt{N})$ (partie entière de racine de N).

On traite le cas N=1 à part !

On utilise une variable booléenne PREM, qui est au départ à VRAI et qui passe à FAUX dès que l'on rencontre un diviseur de N.

```
Fonction estpremier(N)
local PREM, I, J
E(√N) ->J
Si N = 1 alors
  FAUX->PREM
  sinon
  VRAI->PREM
fsi
2->I
tant que PREM et I ≤J faire
  si N mod I = 0 alors
     FAUX->PREM
     sinon
     I+1->I
  fsi
ftantque
résultat PREM
ffonction
```

-Première amélioration

On peut remarquer que l'on peut tester si N est pair et sinon regarder si N possède un diviseur impair.

```
Fonction estpremier(N)
local PREM, I, J
E(√N) ->J
Si (N = 1) ou (N mod 2 = 0) et N≠2 alors
  FAUX->PREM
  sinon
  VRAI->PREM
fsi
3->I
tant que PREM et I ≤J faire
  si N mod I = 0 alors
     FAUX->PREM
     sinon
     I+2->I
```

```
  fsi
ftantque
résultat PREM
ffonction
```

-Deuxième amélioration

On regarde si N est divisible par 2 ou par 3, et sinon on regarde si N posséde un diviseur de la forme $6 \times k - 1$ ou $6 \times k + 1$.

```
Fonction estpremier(N)
local PREM, I, J
E(√N) ->J
Si (N = 1) ou (N mod 2 = 0) ou ( N mod 3 = 0) alors
  FAUX->PREM
  sinon
  VRAI->PREM
fsi
si N=2 ou N=3 alors
VRAI->PREM
fsi
5->I
tant que PREM et I ≤J faire
  si (N mod I = 0) ou (N mod I+2 =0) alors
     FAUX->PREM
     sinon
     I+6->I
  fsi
ftantque
résultat PREM
ffonction
```

7.5.2 Traduction HP49G mode RPN

On traduit le dernier algorithme : le résultat est 0 (FAUX) ou 1 (VRAI).

```
≪ DUP √ FLOOR 0 5 → N J PREM I
 ≪ IF N 1 == N 2 MOD 0 == OR N 3 MOD 0 == OR THEN
      0 'PREM' STO
    ELSE
       1 'PREM' STO
    END
    IF N 2 == N 3 == OR THEN
       1 'PREM' STO
    END
    WHILE PREM I J ≤ AND REPEAT
       IF N I MOD 0 == N I 2 + MOD 0 == OR THEN
          0 'PREM' STO
       ELSE
          I 6 + 'I' STO
       END
   END
```

```
  PREM
 ≫
≫
```

7.5.3 Traduction HP49G mode Algébrique

```
≪ → N
  ≪ 0 → P
    ≪ IF N MOD 2 == 0 OR N MOD 3 == 0 OR N == 1THEN
        0 STO▷ P
      ELSE;
        1 STO▷ P;
      END;
      IF N == 2 OR N == 3 THEN
        1 STO▷ P;
      END;
      FLOOR(√N) → J
        ≪ 5 → I
          ≪ WHILE I ≤ J AND P REPEAT
              IF N MOD I == 0 OR N MOD (I + 2) == 0 THEN
                0 STO▷ P;
              ELSE;
                I + 6 STO▷ I;
              END;
            END;
            P
          ≫
        ≫
    ≫
  ≫
≫ STO▷ PREM?
```

Puis par exemple `PREM?(45789)` pour l'exécuter.

7.6 Méthode probabiliste de Mr Rabin

Si N est premier alors tous les nombres K strictement inférieurs à N sont premiers avec N, donc d'après le petit théorème de Fermat on a :

$K^{N-1} = 1 \pmod N$

Si N n'est pas premier, les entiers K verifiant :

$K^{N-1} = 1 \pmod N$

sont très peu nombreux.

Plus précisement on peut montrer que si $N > 4$, la probabilité de tomber sur un tel nombre K est inférieure à 0.25.

Un nombre N vérifiant $K^{N-1} = 1 \pmod N$ pour 20 tirages de K est un nombre pseudo-premier. La méthode probabiliste de Rabin consiste à tirer au hasard un nombre K $(1 < K < N)$ et à calculer :

$K^{N-1} \pmod N$

Si $K^{N-1} = 1 \pmod N$ on refait un autre tirage et si $K^{N-1} \neq 1 \pmod N$ on est sûr que N n'est pas premier.

Si on obtient $K^{N-1} = 1 \pmod N$ pour 20 tirages de K on peut conclure que N est premier avec une probabilité d'erreur très faible inférieure à 0.25^{20} soit de l'ordre de 10^{-12}.

Bien sûr cette méthode est employée pour savoir si de grands nombres sont pseudo-premiers.

7.6.1 Traduction Algorithmique

On suppose que :

Hasard(N) donne un nombre au hasard entre 0 et N-1.

Le calcul de :

K^{N-1} *mod N*

se fait grâce à l'algorithme de la puissance rapide (cf page 115).

On notera :

puismod(K, P, N) la fonction qui calcule K^P *mod N*

```
Fonction estprem(N)
local K, I, P
1->I
1->P
Tant que P = 1 et I < 20 faire
hasard(N-2)+2->K
puismod(K, N-1, N)->P
I+1->I
ftantque
Si P =1 alors
resultat VRAI
sinon
resultat FAUX
fsi
ffonction
```

7.6.2 Traduction HP49G mode RPN

On suppose que l'on a ecrit la fonction PUIMOD qui prend sur la pile trois arguments A, K, N et qui renvoie $A^K mod\ N$.

```
≪ 1 0 1 → N I K P
 ≪ 0 RDZ
  WHILE P 1 == 20 I > AND REPEAT
     1 'I' STO+
    RAND N 2 - * FLOOR 2 + 'K' STO
    K N 1 - N PUIMOD 'P' STO
  END
  IF P 1 == THEN
     1
   ELSE
     0
  END
```

```
  ≫
≫
```

7.6.3 Traduction HP49G mode Algébrique

```
≪ → N
   ≪ 1 → I
      ≪ 0 → K
         ≪ 1 → P
            ≪ RDZ(0);
               WHILE P == 0 AND I < 20 REPEAT
                  1 + I STO ▷ I;
                  FLOOR((N − 2) ∗ RAND) + 2 STO ▷ K;
                  PUIMOD(K, N − 1, N) STO ▷ P;
               END;
               IF P == 1 THEN
                  1
               ELSE
                  0
               END;
               P
            ≫
         ≫
      ≫
   ≫
≫ STO ▷ RABIN
```

Puis par exemple, `RABIN(45313)` pour l'exécuter.
Remarque :
On peut aussi utiliser la commande du calcul formel `POWMOD` et on écrit alors :
En mode `RPN` :
`N MODSTO`
`K N 1 - POWMOD 'P' STO`
à la place de :
`K N 1 - N PUIMOD 'P' STO`
En mode `Algèbrique` :
`MODSTO(N);`
`POWMOD(K,N-1) STO▷ P`
à la place de :
`PUIMOD(K,N-1,N) STO▷ P`

Index

Table des matières