N° d'ordre : 956

THÈSE

présentée à

L'UNIVERSITÉ DES SCIENCES ET TECHNIQUES DE LILLE

pour obtenir le titre de

DOCTEUR DE 3ème CYCLE

par

Hafedh AMAMOU

CONCEPTION ET REALISATION DE LOGICIELS D'ARITHMETIQUE

Thèse soutenue le 5 mars 1982, devant la Commission d'Examen

Membres du Jury	Président	P. POUZET
	Rapporteur	C. BREZINSKI
	Examinateurs	J. VIGNES
		F. DURBIN

PROFESSEURS 1ère CLASSE

M.	BACCHUS Pierre	Mathématiques
M.	BEAUFILS Jean-Pierre (dét.)	Chimie
M.	BIAYS Pierre	G.A.S.
M.	BILLARD Jean	Physique
M.	BONNOT Ernest	Biologie
M.	BOUGHON Pierre	Mathématiques
M.	BOURIQUET Robert	Biologie
M.	CELET Paul	Sciences de la Terre
M.	COEURE Gérard	Mathématiques
M.	CONSTANT Eugène	I.E.E.A.
M.	CORDONNIER Vincent	I.E.E.A.
M.	DEBOURSE Jean-Pierre	S.E.S.
M.	DELATTRE Charles	Sciences de la Terre
M.	DURCHON Maurice	Biologie
M.	ESCAIG Bertrand	Physique
M.	FAURE Robert	Mathématiques
M.	FOURET René	Physique
M.	GABILLARD Robert	I.E.E.A.
M.	GRANELLE Jean-Jacques	S.E.S.
M.	GRUSON Laurent	Mathématiques
M.	GUILLAUME Jean	Biologie
M.	HECTOR Joseph	Mathématiques
M.	HEUBEL Joseph	Chimie
M.	LABLACHE COMBIER Alain	Chimie
M.	LACOSTE Louis	Biologie
M.	LANSRAUX Guy	Physique
M.	LAVEINE Jean-Pierre	Sciences de la Terre
M.	LEBRUN André	C.U.E.E.P.
M.	LEHMANN Daniel	Mathématiques
Mme	LENOBLE Jacqueline	Physique
M.	LHOMME Jean	Chimie
M.	LOMBARD Jacques	S.E.S.
M.	LOUCHEUX Claude	Chimie
M.	LUCQUIN Michel	Chimie

M.	MAILLET Pierre	S.E.S.
M.	MONTREUIL Jean	Biologie
M.	PAQUET Jacques	Sciences de la Terre
M.	PARREAU Michel	Mathématiques
M.	PROUVOST Jean	Sciences de la Terre
M.	SALMER Georges	I.E.E.A.
Mme	SCHWARTZ Marie-Hélène	Mathématiques
M.	SEGUIER Guy	I.E.E.A.
M.	STANKIEWICZ François	Sciences Economiques
M.	TILLIEU Jacques	Physique
M.	TRIDOT Gabriel	Chimie
M.	VIDAL Pierre	I.E.E.A.
M.	VIVIER Emile	Biologie
M.	WERTHEIMER Raymond	Physique
M.	ZEYTOUNIAN Radyadour	Mathématiques

PROFESSEURS 2ème CLASSE

M.	AL FAKIR Sabah	Mathématiques
M.	ANTOINE Philippe	Mathématiques
M.	BART André	Biologie
Mme	BATTIAU Yvonne	Géographie
M.	BEGUIN Paul	Mathématiques
M.	BELLET Jean	Physique
M.	BKOUCHE Rudolphe	Mathématiques
M.	BOBE Bernard	S.E.S.
M.	BODART Marcel	Biologie
M.	BOILLY Bénoni	Biologie
M.	BONNELLE Jean-Pierre	Chimie
M.	BOSQ Denis	Mathématiques
M.	BREZINSKI Claude	I.E.E.A.
M.	BRUYELLE Pierre (Chargé d'enseignement)	Géographie
M.	CAPURON Alfred	Biologie
M.	CARREZ Christian	I.E.E.A.
M.	CHAMLEY Hervé	E.U.D.I.L.

M. CHAPOTON Alain	C.U.E.E.P.
M. COQUERY Jean-Marie	Biologie
Mme CORSIN Paule	Sciences de la Terre
M. CORTOIS Jean	Physique
M. COUTURIER Daniel	Chimie
Mle DACHARRY Monique	Géographie
M. DEBRABANT Pierre	E.U.D.I.L.
M. DEGAUQUE Pierre	I.E.E.A.
M. DELORME Pierre	Biologie
M. DEMUNTER Paul	C.U.E.E.P.
M. DE PARIS Jean-Claude	Mathématiques
M. DEVRAINNE Pierre	Chimie
M. DHAINAUT André	Biologie
M. DORMARD Serge	S.E.S.
M. DOUKHAN Jean-Claude	E.U.D.I.L.
M. DUBOIS Henri	Physique
M. DUBRULLE Alain	Physique
M. DUEE Gérard	Sciences de la Terre
M. DYMENT Arthur	Mathématiques
M. FLAMME Jean-Marie	E.U.D.I.L.
M. FONTAINE Hubert	Physique
M. GERVAIS Michel	S.E.S.
M. GOBLOT Rémi	Mathématiques
M. GOSSELIN Gabriel	S.E.S.
M. GOUDMAND Pierre	Chimie
M. GREVET Patrice	S.E.S.
M. GUILBAULT Pierre	Biologie
M. HANGAN Théodore	Mathématiques
M. HERMAN Maurice	Physique
M. JACOB Gérard	I.E.E.A.
M. JACOB Pierre	Mathématiques
M. JOURNEL Gérard	E.U.D.I.L.
M. KREMBEL Jean	Biologie
M. LAURENT François	I.E.E.A.
Mle LEGRAND Denise	Mathématiques
Mle LEGRAND Solange	Mathématiques (Calais)
Mme LEHMANN Josiane	Mathématiques

M.	LEMAIRE Jean	Physique
M.	LENTACKER Firmin	G.A.S.
M.	LEVASSEUR Michel	I.P.A.
M.	LHENAFF René	G.A.S.
M.	LOCQUENEUX Robert	Physique
M.	LOSFELD Joseph	I.E.E.A.
M.	LOUAGE Francis	E.U.D.I.L.
M.	MACKE Bruno	Physique
M.	MAIZIERES Christian	I.E.E.A.
Mle	MARQUET Simone	Mathématiques
M.	MESSELYN Jean	Physique
M.	MIGEON Michel	E.U.D.I.L.
M.	MIGNOT Fulbert	Mathématiques
M.	MONTEL Marc	Physique
Mme	NGUYEN VAN CHI Régine	G.A.S.
M.	PARSY Fernand	Mathématiques
Mle	PAUPARDIN Colette	Biologie
M.	PERROT Pierre	Chimie
M.	PERTUZON Emile	Biologie
M.	PONSOLLE Louis	Chimie
M.	PORCHET Maurice	Biologie
M.	POVY Lucien	E.U.D.I.L.
M.	RACZY Ladislas	I.E.E.A.
M.	RICHARD Alain	Biologie
M.	RIETSCH François	E.U.D.I.L.
M.	ROGALSKI Marc	M.P.A.
M.	ROUSSEAU Jean-Paul	Biologie
M.	ROY Jean-Claude	Biologie
M.	SALAMA Pierre	S.E.S.
Mme	SCHWARZBACH Yvette (CCP)	M.P.A.
M.	SCHAMPS Joël	Physique
M.	SIMON Michel	S.E.S.
M.	SLIWA Henri	Chimie
M.	SOMME Jean	G.A.S.
Mle	SPIK Geneviève	Biologie
M.	STERBOUL François	E.U.D.I.L.
M.	TAILLIEZ Roger	Institut Agricole

M.	TOULOTTE Jean-Marc	I.E.E.A.
M.	VANDORPE Bernard	E.U.D.I.L.
M.	WALLART Francis	Chimie
M.	WATERLOT Michel	Sciences de la Terre
Mme	ZINN JUSTIN Nicole	M.P.A.

CHARGES DE COURS

M.	TOP Gérard	S.E.S.
M.	ADAM Michel	S.E.S.

CHARGES DE CONFERENCES

M.	DUVEAU Jacques	S.E.S.
M.	HOFLACK Jacques	I.P..A
M.	LATOUCHE Serge	S.E.S.
M.	MALAUSSENA DE PERNO Jean-Louis	S.E.S.
M.	OPIGEZ Philippe	S.E.S.

Je tiens à remercier Monsieur le Professeur P. POUZET, qui a bien voulu me faire l'honneur de présider le jury de cette thèse.

Je suis très reconnaissant envers Monsieur le Professeur C. BREZINSKI qui m'a proposé ce travail et m'a guidé de ses conseils.

Je remercie Monsieur le Professeur J. VIGNES qui a bien voulu s'intéresser à ce travail et le juger.

Je remercie également Monsieur DURBIN du Commissariat à l'Energie Atomique qui m'a régulièrement conseillé et guidé et qui a bien voulu participer à ce jury.

Que les Membres du Laboratoire d'Analyse Numérique soient assurés de ma gratitude pour m'avoir aidé de leurs conseils durant plus de deux ans.

Mes remerciements vont aussi à Madame Patricia CARON et à Monsieur et Madame DEBOCK qui ont assuré avec gentillesse, compétence et rapidité la réalisation matérielle de ce document.

SOMMAIRE

CHAPITRE I

CHAPITRE II

CHAPITRE III

CHAPITRE IV

I - INTRODUCTION

II - ETUDE DES SUITES DE CHOQUET

III - ETUDE DE L'ε-ALGORITHME

IV - ETUDE DU DEVELOPPEMENT EN SERIE ENTIERE DE LA FONCTION EXPONENTIELLE

CHAPITRE V

INTRODUCTION

Le calcul numérique sur ordinateur comporte deux catégories d'erreurs :

- l'erreur de calcul dûe à l'arithmétique mise en jeu sur ordinateur et qui n'est qu'une approximation des opérateurs mathématiques.
- l'erreur de méthode dûe à l'algorithme utilisé.

Cette erreur entâche les résultats des méthodes numériques approchées (quadrature, ...) par opposition aux méthodes numériques exactes (méthodes directes d'algèbre linéaire ...) qui donnent des résultats exacts en un nombre fini d'opérations arithmétiques.

De nombreux auteurs ont publié des résultats importants concernant le deuxième type d'erreur.

Le premier type d'erreur a été moins étudié du fait qu'il est directement dépendant de la conception même des ordinateurs.

En effet, le concept d'infiniment petit et d'infiniment grand n'est valable qu'en algèbre. En informatique numérique ces concepts ne sont plus valables. Donnons un exemple :

$$u = \frac{(y+x) - x}{y}$$

$$v = \frac{y + (x-x)}{y}$$

En algèbre quels que soient x et y, $y \neq 0$, on a :

$$u = v = 1$$

par contre, le calcul sur ordinateur donne des résultats tout à fait différents suivant les valeurs de x et y. Par exemple pour $x = 1$ et $y = 10^{-40}$ on trouve

$$u = 0$$

$$v = 1$$

Ceci est la preuve de la non associativité de l'addition.

Un algorithme défini dans le domaine algébrique où les calculs s'effectuent avec une précision infinie, si on le transpose dans le domaine informatique où les calculs s'effectuent avec une précision finie, risque de donner des résultats complétement différents.

Le travail qu'on propose dans ce mémoire a donc été réalisé pour remédier à ce type d'erreur.

Ces dernières années on a vu se développer plusieurs directions principales de recherche sur l'arithmétique de l'ordinateur :

- estimations statistiques de ces erreurs par M. La Porte et J. Vignes [1]
- arithmétique d'intervalle par Moore [2] et Nickel [3].
- une nouvelle arithmétique par Kulish et Miranker [4].
- arithmétique résiduelle et arithmétique p-adique par T. Gregory [5].

Ces recherches ont conduit à des méthodes de contrôle et de correction des erreurs dûes à l'arithmétique de l'ordinateur qui sont extrêmement performantes. Cependant leur mise en oeuvre est souvent longue et difficile à effectuer. C'est pour cette raison que ce travail ne s'interesse qu'à des techniques bien moins élaborées mais dont la mise en oeuvre effective est plus simple.

Dans le chapitre 1, on propose en première partie une méthode pour convertir en rationnels les nombres réels définis en virgule flottante. Dans la deuxième partie une approche de l'arithmétique rationnelle, définie sur des opérateurs rationnels, est donnée à travers l'ε-algorithme dans lequel l'erreur de calcul est souvent importante. Cette erreur est dûe à la soustraction, de deux nombres voisins, qui apparaît au dénominateur de la relation de l'ε-algorithme :

$$\varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + \frac{1}{\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}}$$

Le chapitre 2 est consacré à l'étude et à la programmation des opérateurs de l'arithmétique rationnelle définis sur des opérandes rationnels de longueur étendue, permettant de dépasser la précision de la machine.

Dans le chapitre 3, on développe l'arithmétique réelle sur des opérandes réels définis avec des mantisses de longueur étendue.

Le chapitre 4 est consacré aux différentes expériences numériques développées sur l'ordinateur IRIS 80 du centre de calcul de l'Université de Lille I.

Dans le chapitre 5, on donne un aperçu de l'ensemble des nombres p-adiques, ainsi que de l'arithmétique p-adique tronquée en montrant les problèmes qui se posent encore pour la rendre utilisable dans les algorithmes.

CHAPITRE I

ANALYSE ET CODAGE DE LA CONVERSION DES REELS EN RATIONNELS ET DE L'ε-ALGORITHME RATIONNEL

I - INTRODUCTION

Nous allons développer dans ce chapitre une arithmétique exacte : l'arithmétique rationnelle, en l'appliquant à un algorithme d'extrapolation : l'ε-algorithme. Cette arithmétique effectue un calcul exact à partir de données rationnelles exactes.

Dans la première partie nous donnerons une définition de l'arithmétique rationnelle.

Dans la deuxième partie nous analyserons l'algorithme de conversion des réels en rationnels : l'algorithme des fractions continues ; puis nous appliquerons l'arithmétique rationnelle à l'ε-algorithme scalaire.

En conclusion nous analyserons les résultats numériques obtenus par l'algorithme des fractions continues et l'ε-algorithme rationnel.

II - L'ARITHMÉTIQUE RATIONNELLE

II-1 DEFINITION D'UNE ARITHMETIQUE EXACTE

Soit le couple $(\mathbf{F},(w_1,w_2))$ où $\mathbf{F}$ est un sous-ensemble de $\mathbf{R}$, et w_1, w_2 deux opérateurs définis sur cet ensemble.

On dit que ce couple définit une arithmétique exacte si les propriétés suivantes sont vérifiées.

a / fermeture : $a,b \in \mathbf{F} \Longrightarrow a w_i b \in \mathbf{F}$, $i = 1,2$

b / commutativité : $a w_i b = b w_i a$, $\forall\ a, b \in \mathbf{F}$, $i = 1,2$

c / associativité : $(a w_i b)\ w_i c = a w_i (b w_i c)$, $\forall\ a,b,c \in \mathbf{F}$, $i = 1,2$

d / distributivité : $(a w_1 b)\ w_2 c = (a w_2 c)\ w_1\ (b w_2 c)$, $\forall\ a,b,c \in \mathbf{F}$

e / élément neutre : $a w_1 0 = a$, $\forall\ a \in \mathbf{F}$

f / élément unité : $a w_2 1 = a$, $\forall\ a \in \mathbf{F}$

g / opposé : $\forall\ a \in \mathbf{F}$, $\exists\ b \in \mathbf{F}$: $a w_1 b = 0$

h / inverse : $\forall\ a \in \mathbf{F} - \{0\}$, $\exists\ b \in \mathbf{F}$: $a w_2 b = 1$

Il est évident que le couple $(\mathbb{Q}, (\oplus,*))$, $\mathbb{Q}$ étant l'ensemble des rationnels vérifie toutes ces propriétés et donc il définit une arithmétique exacte appelée l'arithmétique rationnelle.

Remarque : on désigne par le symbole $\oplus$ aussi bien l'addition que la soustraction.

II-2 DEFINITION DES OPERATEURS SUR L'ENSEMBLE DES RATIONNELS

*** Addition, Soustraction**

Soient deux rationnels N1/D1 et N2/D2 appartenant à $\mathbb{Q}$, leur somme est égale à :

$$\frac{N3}{D3} = \frac{N1 * D2 \oplus N2 * D1}{D1 * D2}$$

* *Multiplication*

Soient deux rationnels N1/D1 et N2/D2 appartenant à $\mathbb{Q}$, leur produit est égal à :

$$\frac{N3}{D3} = \frac{N1 * N2}{D1 * D2}$$

Remarque : La division de deux rationnels se fait en multipliant par l'inverse du diviseur.

II-3 L'ε-ALGORITHME RATIONNEL

Les quantités calculées par l'ε-algorithme scalaire se placent habituellement dans le tableau à double entrée suivant :

$$\begin{array}{lllll}
\varepsilon_0^{(0)} = S_0 & & & & \\
 & \varepsilon_1^{(0)} & & & \\
\varepsilon_0^{(1)} = S_1 & & \varepsilon_2^{(0)} & & \\
 & \varepsilon_1^{(1)} & & \varepsilon_{k-1}^{(0)} & \\
\varepsilon_0^{(2)} = S_2 & & \varepsilon_2^{(1)} & & \varepsilon_k^{(0)} \\
 & \varepsilon_1^{(2)} & & & \\
\varepsilon_0^{(3)} = S_3 & & & &
\end{array}$$

tel que $\varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n)} + (\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)})^{-1}$ $\qquad n = 0,1,2,\dots$ et $k = 1,2,\dots$
avec $\varepsilon_o^{(n)} = S_n$, $\varepsilon_1^{(n)} = 0 \quad \forall n$

On suppose que la suite initiale est donnée sous forme rationnelle.
Soit $\varepsilon_0^{(n)} = S_n = \varepsilon N_0^{(n)} / \varepsilon D_0^{(n)}$, ce qui donne pour le terme général

$$\varepsilon_{k+1}^{(n)} = \frac{\varepsilon N_{K+1}^{(n)}}{\varepsilon D_{K+1}^{(n)}} = \frac{\varepsilon N_{K-1}^{(n)}}{\varepsilon D_{K-1}^{(n)}} + \frac{1}{\varepsilon N_K^{(n+1)} / \varepsilon D_K^{(n+1)} - \varepsilon N_K^{(n)} / \varepsilon D_K^{(n)}}$$

$$\varepsilon_{k+1}^{(n)} = \frac{\varepsilon N_{K-1}^{(n)} * [\varepsilon N_K^{(n+1)} * \varepsilon D_K^{(n)} - \varepsilon N_K^{(n)} * \varepsilon D_K^{(n+1)}] + \varepsilon D_{K-1}^{(n)} * \varepsilon D_K^{(n+1)} * \varepsilon D_K^{(n)}}{\varepsilon D_{K-1}^{(n)} * [\varepsilon N_K^{(n+1)} * \varepsilon D_K^{(n)} - \varepsilon N_K^{(n)} * \varepsilon D_K^{(n+1)}]}$$

Donc pour la programmation de l'ε-algorithme scalaire en rationnel on a besoin de connaître :

- la suite initiale sous forme rationnelle
- le sous-programme de la somme de deux rationnels.

III - ANALYSE DES ALGORITHMES

En calcul automatique il n'est pas possible de représenter tout l'ensemble $\mathbb{Q}$ sur ordinateur, ce qui se traduit par le fait que la propriété de fermeture n'est plus vérifiée, par contre les autres propriétés qui définissent une arithmétique exacte restent vraies.

Sur l'ordinateur IRIS 80 de CII-HB (sur lequel on a testé les programmes) la représentation des entiers est limitée supérieurement à $2^{31}-1$ et inférieurement à -2^{31}. Cette limitation nous conduit donc à :

- initialiser l'ε-algorithme avec une suite dont les termes écrits sous forme rationnelle sont donnés dans leur expression la plus réduite. (C'est-à-dire que le numérateur et le dénominateur doivent être premiers entre eux).
- contrôler le dépassement de capacité après chaque opération élémentaire
- simplifier chaque rationnel du calcul intermédiaire en le ramenant à sa forme réduite.

III-1 TRANSFORMATION DES TERMES D'UNE SUITE DE REELS EN RATIONNELS

Soit la suite $(S_n, n \in \mathbb{N})$ dont les termes S_n appartiennent à $\mathbf{F}$ ensemble des nombres réels écrits en virgule flottante normalisée et qui sont représentables exactement en machine. Le terme S_n étant écrit avec un nombre fini de chiffres on peut donc le transformer en rationnel de la forme (N/D) en le développant en fraction continue.

III-1-1 Rappel sur les fractions continues

On appelle fraction continue l'expression $b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2}}$ qu'on notera

$$b_0 + \frac{a_1|}{|b_1} + \frac{a_2|}{|b_2} + \ldots$$

On appelle $n^{\text{ème}}$ convergent ou réduite de la fraction continue le terme
$C_n = b_0 + \frac{a_1|}{|b_1} + \ldots + \frac{a_n|}{|b_n} = p_n/q_n$.

Le calcul des convergents successifs de la fraction continue est donné par les formules de récurrence suivantes :

$$p_{-1} = 1 \qquad p_0 = b_0$$

$$q_{-1} = 0 \qquad q_0 = 1$$

$$p_m = a_m p_{m-2} + b_m p_{m-1} \qquad m = 1,2,\ldots$$

$$q_m = a_m q_{m-2} + b_m q_{m-1} \qquad m = 1,2,\ldots$$

Théorème 1 [6] Deux convergents successifs vérifient la relation suivante :

$$p_{m-1}\, q_m - p_m\, q_{m-1} = (-1)^m\, a_1 \ldots a_m \qquad \forall\, m \geq 1.$$

III-1-2 Fractions continues et théorie des nombres

Soit X un nombre réel positif. Notons par [X] sa partie entière. On a alors :

$$X = [X] + r \qquad \text{avec} \qquad 0 < r < 1$$

appelons y l'inverse de r, y > 1 et :

$$X = [X] + 1/y.$$

Ce procédé peut être itéré et on obtient X sous forme de fraction continue dont les termes a_i sont tous égaux à 1, appelée fraction continue simple.

Théorème 2 [6] Un nombre réel est rationnel si et seulement si sa fraction continue simple est finie.

Dans le cas des fractions continues simples, la formule du théorème 1 devient :

$$p_{m-1}\, q_m - p_m\, q_{m-1} = (-1)^m$$

et les deux termes d'une même réduite sont premiers entre eux. On peut donc tirer des deux théorèmes précédents le théorème fondamental :

Théorème 3 Tout nombre réel écrit en virgule flottante avec un nombre fini de digits décimaux peut être transformé en fraction irréductible par un calcul fini de ses réduites successives.

III-1-3 Algorithme de conversion d'un réel en rationnel

Soit un réel positif X écrit en virgule flottante avec n chiffres. Pour le convertir en fraction irréductible IN/ID on utilise l'algorithme RAT suivant :

[RAT1] [initialisation] ; N ← 1 ; M ← 0 ; IN ← [X] ; ID ← 1
[RAT2] [critère d'arrêt] ; V = X - [X] ; si V = 0 STOP ;
[RAT3] [calcul des réduites] ; X ← 1/V

IR ← [X] * IN * N ; N ← IN ; IN ← IR ;
IR ← [X] * ID + M ; M ← ID ; ID ← IR ;

aller en [RAT2].

EXEMPLE Soit $X = \frac{1627}{2520} = 0{,}645634921 = 1 - \frac{1}{2} + \frac{1}{3} + \ldots - \frac{1}{10}$

i	1	2	3	4	5	6	7	8	9	10	11	12	13
p_i	1	1	2	9	11	20	31	51	82	133	747	880	1627
q_i	1	2	3	14	17	31	48	79	127	206	1157	1363	2520

donc $\frac{p_{13}}{q_{13}} = \frac{1627}{2520} = 0{,}645634921.$

III-2 CALCUL DE LA SOMME DE DEUX FRACTIONS

Le symbole $\oplus$ désignera aussi bien l'addition que la soustraction.

Soit à calculer $\frac{w}{w'} = \frac{u}{u'} \oplus \frac{v}{v'}$ sachant que u/u' et v/v' sont deux fractions réduites. On a vu au paragraphe II.2 qu'on a $w = u * v' \oplus u' * v$ et $w' = u'*v'$.

En général le résultat w/w' n'est pas une fraction réduite. Voyons comment obtenir une fraction réduite :

Si le plus grand commun diviseur de u' et v' noté $d_1 = \text{pgcd}(u', v')$ est plus grand que 1 on a :

$$w/w' = \frac{(u * v' \oplus v * u')/d_1}{(u' * v')/d_1}$$

posons $t = (u * v' \oplus v * u')/d_1$. On calcule $d_2 = \text{pgcd}(t,d_1)$ et on obtient le résultat

$$\frac{w}{w'} = \frac{t \,/\, d_2}{(u'/d_1) * (v'/d_2)}$$

Théorème 4 [7]

Si u et u' sont premiers entre eux ainsi que v et v' alors $\text{pgcd}(uv' \oplus u'v, u'v') = d_1 d_2$ avec $d_1 = \text{pgcd}(u',v')$ et $d_2 = \text{pgcd}(u(v'/d_1) \oplus v(u'/d_1), d_1)$.

Corollaire

Si d_1 est égal à 1 alors $uv' \oplus u'v$ et $u'v'$ sont premiers entre eux.

III-2-1 Algorithme de la somme de deux fractions

Soit les deux fractions irréductibles INB/IDB et INC/IDC. Leur somme sous forme de fraction irréductible INA/IDA est donnée par l'algorithme 'ADD' suivant :

```
[ADD1]   ID ← pgcd (IDB,IDC)
[ADD2]   Si ID = 1 alors INA ← INB * IDC + IDB * INC
                         IDA ← IDB * IDC              STOP
```

Sinon

[ADD3] $T \leftarrow INB * (IDC/ID) + INC * (IDB/ID)$

[ADD4] $ID2 \leftarrow pgcd\ (T,ID)$

[ADD5] $INA \leftarrow T/ID2$

$IDA \leftarrow (IDB/ID) * (IDC/ID2)$ STOP

L'utilisation de cet algorithme permet de réduire la taille des entiers qui apparaissent dans le calcul intermédiaire de telle sorte que la probabilité de dépassement de capacité diminue. On va le voir sur un exemple.

<u>EXEMPLE</u> Supposons que l'on travaille avec un ordinateur dont la capacité maximum d'un mot mémoire en entier est de 1000. On veut calculer l'addition

$$\frac{INA}{IDA} = \frac{17}{70} + \frac{5}{42}\,.$$

Si on utilise un algorithme classique on trouve

$INA = 17 \times 42 + 70 \times 5 = 1064$ et $IDA = 70 \times 42 = 2940$.

Le pgcd de INA et IDA étant 28, on simplifie et on trouve : $\frac{INA}{IDA} = \frac{38}{105}$.
Mais sur ordinateur ce calcul n'aurait pas été jusqu'au bout car le dépassement de capacité se produit dans le calcul intermédiaire de INA et IDA.

Utilisons maintenant l'algorithme exposé plus haut :

$$\begin{aligned} ID &= pgcd\ (70,42) = 14 \\ T &= 17 \times 3 * 5 \times 5 = 76 \\ ID2 &= pgcd\ (76,14) = 2 \\ INA &= 76/2 = 38 \\ IDA &= (70/14) * (42/2) = 105 \end{aligned}$$

Le résultat est donc obtenu sans qu'aucun dépassement de capacité ne se soit produit dans les calculs intermédiaires.

III-2-2 Algorithme d'Euclide

Le calcul du plus grand commun diviseur de deux nombres A et B se fait par l'algorithme d'Euclide

Soient A et B deux entiers positifs. On supposera sans restreindre la généralité que $A \geq B$. Divisons A par B , on a :

$$A = B * Q_0 + R_0 \qquad \text{avec} \qquad 0 \leq R_0 < B$$

Remplaçons A par B et B par R_0 et itérons le procédé, on a :

$$B = R_0 * Q_1 + R_1 \qquad \text{avec} \qquad 0 \leq R_1 < R_0$$

et ainsi de suite : $R_{k-2} = R_{k-1} * Q_k + R_k$ avec $0 \leq R_k < R_{k-1}$.

On obtient une suite (R_k) strictement décroissante. Il existe donc un entier n tel que : $R_{n+1} = 0$ et donc $R_{n-1} = R_n * Q_{n+1}$. Tout diviseur de A et B divise R_0 ainsi que R_1 jusqu'à R_n. Réciproquement R_n divise R_{n-1} et donc il divise A et B. On appelle R_n le plus grand commun diviseur (pgcd) de A et de B.

La méthode précédente pour trouver le pgcd de deux nombres s'appelle l'algorithme d'Euclide [7] on le notera PGCD :

on veut calculer $R = PGCD(A,B)$; A et $B \geq 1$

[PGCD1] si B divise A ; R = B STOP

[PGCD2] si (A mod B)=1; R = 1 STOP

Sinon remplacer (A,B) par (B,A mod B) et aller en PGCD1.

III-3 L'ε-ALGORITHME SCALAIRE

L'utilisation de l'ε-algorithme s'effectue en parallèle avec le calcul des termes de la suite à accèlérer selon la technique présentée dans [10].
Une fois calculé l'élément S_n on calcule la diagonale montante partant de S_n.
Si on numérote dans le tableau ε les éléments dans l'ordre où ils sont calculés, cela donne :

$\varepsilon_0^{(n)}$	$\varepsilon_1^{(n)}$	$\varepsilon_2^{(n)}$	$\varepsilon_3^{(n)}$	$\varepsilon_4^{(n)}$
1				
	3			
		6		
2			10	
	5			15
		9		
4			14	
	8			
7		13		
	12			
11				

Cette technique nous permet de rajouter des termes à la suite initiale au fur et à mesure que l'on en a besoin. Si on introduit le terme S_n la connaissance de la diagonale montante issue de S_{n-1} suffit pour continuer le calcul dans la table ε. L'algorithme va donc fonctionner de la façon suivante : on calculera un terme de la suite initiale, puis on rentrera dans le sous-programme pour calculer une diagonale montante. Une fois ce calcul terminé on sort du sous-programme pour calculer le terme suivant de la suite initiale et ainsi de suite. Pour le calcul de la diagonale l'algorithme convertit le terme de la suite initiale en rationnel, fait les calculs et délivre le résultat en rationnel.

IV - MISE EN OEUVRE DES ALGORITHMES ET ANALYSE DES RÉSULTATS

IV-1 CODAGE DE L'ALGORITHME DES FRACTIONS CONTINUES

L'algorithme est codé par deux sous-programmes différents 'RAT' et 'RATS'. Ces SP communiquent certains paramètres avec le programme d'appel par l'instruction CØMMØN avec label, que l'utilisateur initialise par l'instruction BLØCK DATA. Ces paramètres sont :

NM : plus grand entier positif représentable en machine

NI : numéro logique de l'imprimante.

PR : précision absolue du résultat, utilisée par le critère d'arrêt de l'algorithme. En général cette valeur doit être choisie entre 10^{-13} et 10^{-15} lorsque les réels à convertir sont codés sur 14 digits décimaux.

IV-1-1 Sous-programme RAT

*** RAT ***

<u>Appel</u> : CALL RAT (X, IN, ID, R, ITA)

<u>Paramètres</u> : X, R : réels en virgule flottante. IN, ID, ITA : entiers.
X est un paramètre en entrée. IN, ID, R, ITA sont des paramètres en sortie.

<u>Description</u> : Ce SP convertit le réel X donné en double précision en un rationnel IN/ID, en utilisant l'algorithme du paragraphe III-1-3. Le paramètre R donne la valeur de IN/ID en virgule flottante. L'algorithme

s'arrête dès que $|X-R|$ devient inférieur à la précision PR. Le deuxième critère d'arrêt est donné par le dépassement de capacité sur IN ou ID et dans ce cas ITA, qui est initialisé en début de programme à zéro, prend la valeur 1.

IV-1-2 Sous-programme 'RATS'

Ce SP est le même que celui du paragraphe précédent sauf que X est donné en simple précision.

IV-2 CODAGE DE L'ε-ALGORITHME SCALAIRE

Deux sous-programmes sont donnés dans ce paragraphe. Le premier 'APSIL' effectue des calculs en réel, les données étant des nombres réels double précision. Le second 'EPSIL' reprend l'algorithme 'APSIL' mais convertit les données en rationnels par le SP 'RAT' et effectue les calculs en rationnel ; EPSIL appelle de plus les SP 'ADD' et 'PGCD'. L'ensemble des SP échangent les variables NM, NI, PR avec le programme d'appel par l'instruction CØMMØN avec label, que l'utilisateur initialise par l'instruction BLØCK DATA. Ces paramètres ont la même signification qu'au paragraphe IV-1.

*** APSIL ***

Appel : CALL APSIL (X, ICM, IK, MM, ZE, P, D)

Paramètres : X : réel devant contenir la valeur du terme S_{n-1} de la suite initiale avant le $n^{\text{ème}}$ appel du sous-programme.

ICM : indice de la dernière colonne que l'on veut calculer. Cet entier ne doit pas dépasser MM.

MM : taille maximum des tableaux représentant les diagonales.

IK : entier devant être initialisé à zéro à chaque fois que l'on utilise le SP pour une nouvelle suite.

ZE : réel très petit, de l'ordre de 10^{-15}, qui permet de continuer le calcul dans la table ε en remplaçant dans une division un diviseur nul.

P : tableau de dimension MM on y mémorise la diagonale déjà calculée.

D : tableau de dimension MM. On y mémorise la diagonale en cours de calcul.

X, ICM, MM, IK, ZE sont des paramètres à initialiser en entrée.

Description : Ce sous-programme imprime directement certains éléments dans les colonnes paires, soit l'escalier descendant ε_o^o, ε_o^1, ε_2^o, ε_2^1,..., ε_{ICM}^o, ε_{ICM}^1, ε_{ICM}^2, ... ; Si ICM est égal à MM on s'arrête à ε_{ICM}^1.

Si ICM est impair c'est la colonne (ICM-1) qui est imprimée.

*** EPSIL ***

Appel : CALL EPSIL (X, ICM, IK, MM, PN, PD, DN, DD)

S.P. utilisés : RAT, ADD, PGCD

Paramètres : X, ICM, MM, IK ont la même signification que dans le SP 'APSIL'. PN, PD, DN, DD : tableaux dont la dimension est égale à MM. PN et PD représentant le numérateur et le dénominateur de la diagonale déjà calculée ; DN et DD ceux de la diagonale en cours de calcul.

Description : Ce SP reprend le même algorithme que APSIL mais convertit le réel X en un rationnel par le SP 'RAT', calcule et imprime les résultats en rationnel.

*** ADD ***

Appel : CALL ADD (INA, IDA, INB, IDB, INC, IDC, ETA)

S.P. utilisées : PGCD

Paramètres : Tous les paramètres sont des entiers. INB, IDB, INC, IDC sont des paramètres en entrée. INA, IDA, ETA sont des paramètres en sortie

Description : Ce S.P. calcule $\frac{INA}{IDA} = \frac{INB}{IDB} + \frac{INC}{IDC}$ en utilisant l'algorithme du paragraphe III-2-1. ETA est initialisée à zéro, et mise à 1 s'il y a dépassement de capacité dans l'addition et dans ce cas INA = IDA = 0.

*** PGCD ***

Appel : CALL PGCD (IN, ID, IP)

Paramètres : Tous les paramètres sont des entiers. IN, ID sont en entrée. IP est en sortie.

Description : Ce SP calcule le PGCD IP de IN et ID en utilisant l'algorithme du paragraphe III-2-2.

IV-3 ANALYSE DES RESULTATS

IV-3-1 Algorithmes des fractions continues

Les nombreuses expériences numériques faites sur l'ordinateur IRIS 80 de CII-HB ont montré que l'étape [RAT 3] dans l'algorithme de conversion induit une erreur de calcul dans le terme (1/V) et la convergence des réduites successives de la fraction continue donnée par l'algorithme 'RAT' n'est plus assurée après un nombre fini d'étapes.

Ces pourquoi on a été amené à choisir des critères d'arrêt pour l'algorithme. On donne dans les tableaux (1), (2), (3) les différents résultats obtenus avec différents critères d'arrêt pour des nombres réels dont on connait les valeurs rationnelles exactes. Ces nombres sont les termes de la suite S_n définie par :

$$S_n = \sum_{i=1}^{n} \frac{(-1)^{i+1}}{i}$$

D'après ces résultats les différents choix du nombre de chiffres significatifs exacts du résultat, calculé en utilisant la précision PR, font varier considérablement la réduite choisie pour représenter le nombre réel. Afin d'avoir une bonne précision des calculs et un choix de nombres entiers assez petits pour représenter la réduite, il faut que PR soit compris entre 10^{-13} et 10^{-15}

IV-3-2 L'ε-ALGORITHME SCALAIRE

Les deux sous-programmes 'APSIL' et 'EPSIL' ont été utilisés pour accélérer la convergence des suites $S_n = \frac{1}{n+1}$ et $U_n = \sum_{i=o}^{n} \frac{(-1)^{i+1}}{i+1}$. Le premier exemple est très intéressant car on sait alors que :

$$\varepsilon_{2k}^{(n)} \frac{1}{(k+1)(n+k+1)}$$

Pour l'ε-algorithme rationnel 'EPSIL' dont la suite de départ est donnée en réel, nous avons choisi un SP de conversion RAT avec la valeur initiale en double précision, la réduite choisie sera celle dont la valeur en réel virgule flottante double précision possède 7 chiffres significatifs exacts. Ce choix a été dicté par le souci de concilier la bonne précision des calculs et des valeurs entières pas trop grandes dans le numérateur et le dénominateur de la réduite. Les résultats comparatifs des deux algorithmes sont présentés dans les tableaux (4) et (5).

On voit d'après le tableau (4) que même pour des suites aussi règulières que $\frac{1}{n+1}$ les résultats des deux algorithmes commencent à différer dès la 8ème colonne de la table ε. Pour les termes de la série $\sum_{n=o}^{\infty} (-1)^{n+1}/(n+1)$ la comparaison n'a pas été possible puisqu'un dépassement de capacité sur les variables entières se produit dès la 10$^{\text{éme}}$ colonne de l'ε-algorithme.

V - CONCLUSION

Les expériences numériques présentées, bien qu'ayant un caractère trop académique, permettent de voir l'efficacité de l'utilisation de l'arithmétique rationnelle. D'autre part la capacité de l'ordinateur étant trop limitée pour des opérandes entières (et ceci est général pour tous les ordinateurs) les calculs numériques n'ont pu être menés jusqu'au bout. C'est pourquoi nous allons développer une arithmétique rationnelle multi-précision, c'est à dire où chaque opérande est défini sur plusieurs mots mémoire. Ceci fait l'objet du prochain chapitre où l'on va simuler les opérations arithmétiques de base en rationnel : l'addition, la soustraction, la multiplication, la division entière et le calcul du PGCD.

TABLEAU 1

ALGORITHME DES FRACTIONS CONTINUES AVEC DES REELS SIMPLE PRECISION,LE CALCUL EST ARRETE DES QUE LA SIMPLE PRECISION EST ATTEINTE PAR LE RESULTAT(10E-6)

Y		IN/ID			IN/ID CALCULE			R
.100000E+01	**	1/	1	**	1/	1	**	.100000000000000D+01
.500000E+00	**	1/	2	**	1/	2	**	.500000000000000D+00
.833333E+00	**	5/	6	**	5/	6	**	.833333333333333D+00
.583333E+00	**	7/	12	**	7/	12	**	.583333333333333D+00
.783333E+00	**	47/	60	**	47/	60	**	.783333333333333D+00
.616667E+00	**	37/	60	**	37/	60	**	.616666666666667D+00
.759524E+00	**	319/	420	**	319/	420	**	.759523809523810D+00
.634524E+00	**	533/	840	**	533/	840	**	.634523809523810D+00
.745635E+00	**	1879/	2520	**	299/	401	**	.745635910224439D+00
.645635E+00	**	1627/	2520	**	747/	1157	**	.645635263612792D+00
.736544E+00	**	20417/	27720	**	260/	353	**	.736543909348442D+00
.653211E+00	**	18107/	27720	**	356/	545	**	.653211009174312D+00
.730134E+00	**	263111/	360360	**	928/	1271	**	.730133752950433D+00
.658705E+00	**	237371/	360360	**	855/	1298	**	.658705701078582D+00
.725372E+00	**	52279/	72072	**	1804/	2487	**	.725371934057097D+00
.662872E+00	**	95549/	144144	**	757/	1142	**	.662872154115587D+00
.721695E+00	**	1768477/	2450448	**	1294/	1793	**	.721695482431679D+00
.666140E+00	**	1632341/	2450448	**	421/	632	**	.666139240506329D+00
.718771E+00	**	33464927/	46558512	**	1030/	1433	**	.718771807397069D+00
.668771E+00	**	155685007/	232792560	**	953/	1425	**	.668771929824561D+00

Y : VALEUR EN VIRGULE FLOTTANTE A CONVERTIR

IN/ID:VALEUR EXACTE DE Y EN RATIONNELLE

IN/ID CALCULE:VALEUR RATIONNELLE DE Y CALCULEE PAR FRACTION CONTINUE

R :VALEUR EN VIRGULE FLOTTANTE DE IN/ID CALCULE

TABLEAU 2

ALGORITHME DES FRACTIONS CONTINUES AVEC DES REELS DOUBLE PRECISION,LE CALCUL EST ARRETE DES QUE LA SIMPLE PRECISION EST ATTEINTE PAR LE RESULTAT(10E-7)

Y		IN/ID			IN/ID CALCULE			R
.100000000000000D+01	**	1/	1	**	1/	1	**	.100000000000000D+01
.500000000000000D+00	**	1/	2	**	1/	2	**	.500000000000000D+00
.833333333333333D+00	**	5/	6	**	5/	6	**	.833333333333333D+00
.583333333333333D+00	**	7/	12	**	7/	12	**	.583333333333333D+00
.783333333333333D+00	**	47/	60	**	47/	60	**	.783333333333333D+00
.616666666666667D+00	**	37/	60	**	37/	60	**	.616666666666667D+00
.759523809523810D+00	**	319/	420	**	319/	420	**	.759523809523810D+00
.634523809523810D+00	**	533/	840	**	533/	840	**	.634523809523810D+00
.745634920634921D+00	**	1879/	2520	**	1879/	2520	**	.745634920634921D+00
.645634920634921D+00	**	1627/	2520	**	1627/	2520	**	.645634920634921D+00
.736544011544012D+00	**	20417/	27720	**	20157/	27367	**	.736544012862206D+00
.653210678210678D+00	**	18107/	27720	**	3479/	5326	**	.653210664663913D+00
.730137551337555D+00	**	263111/	360360	**	928/	1271	**	.730133752950433D+00
.658705183705184D+00	**	237371/	360360	**	1048/	1591	**	.658705216844752D+00
.725371850371850D+00	**	52279/	72072	**	1804/	2487	**	.725371934057097D+00
.662871850371850D+00	**	95549/	144144	**	1805/	2723	**	.662871832537642D+00
.721695379783615D+00	**	1768477/	2450448	**	3729/	5167	**	.721695374491968D+00
.666139624228060D+00	**	1632341/	2450448	**	1686/	2531	**	.666139865665745D+00
.718771403175428D+00	**	33464927/	46558512	**	1053/	1465	**	.718771331058020D+00
.668771403175428D+00	**	155685007/	232792560	**	2542/	3801	**	.668771375953696D+00

Y : VALEUR EN VIRGULE FLOTTANTE A CONVERTIR

IN/ID:VALEUR EXACTE DE Y EN RATIONNELLE

IN/ID CALCULE:VALEUR RATIONNELLE DE Y CALCULEE PAR FRACTION CONTINUE

R :VALEUR EN VIRGULE FLOTTANTE DE IN/ID CALCULE

TABLEAU 3

ALGORITHME DES FRACTIONS CONTINUES AVEC DES REELS DOUBLE PRECISION,LE CALCUL EST ARRETE DES QUE LA DOUBLE PRECISION EST ATTEINTE PAR LE RESULTAT(10D-14)

Y		IN/ID			IN/ID CALCULE			R
.1000000000000000D+01	**	1/	1	**	1/	1	**	.1000000000000000D+01
.5000000000000000D+00	**	1/	2	**	1/	2	**	.5000000000000000D+00
.8333333333333333D+00	**	5/	6	**	5/	6	**	.8333333333333333D+00
.5833333333333333D+00	**	7/	12	**	7/	12	**	.5833333333333333D+00
.7833333333333333D+00	**	47/	60	**	47/	60	**	.7833333333333333D+00
.6166666666666667D+00	**	37/	60	**	37/	60	**	.6166666666666667D+00
.7595238095238100D+00	**	319/	420	**	319/	420	**	.7595238095238100D+00
.6345238095238100D+00	**	533/	840	**	533/	840	**	.6345238095238100D+00
.7456349206349210D+00	**	1879/	2520	**	1879/	2520	**	.7456349206349210D+00
.6456349206349210D+00	**	1627/	2520	**	1627/	2520	**	.6456349206349210D+00
.7365440115440120D+00	**	20417/	27720	**	20417/	27720	**	.7365440115440120D+00
.6532106782106780D+00	**	18107/	27720	**	18107/	27720	**	.6532106782106780D+00
.7301337551337550D+00	**	263111/	360360	**	263111/	360360	**	.7301337551337550D+00
.6587051837051840D+00	**	237371/	360360	**	237371/	360360	**	.6587051837051840D+00
.7253718503718500D+00	**	52279/	72072	**	52279/	72072	**	.7253718503718500D+00
.6628718503718500D+00	**	95549/	144144	**	95549/	144144	**	.6628718503718500D+00
.7216953797836150D+00	**	1768477/	2450448	**	1768477/	2450448	**	.7216953797836150D+00
.6661398242280600D+00	**	1632341/	2450448	**	1632341/	2450448	**	.6661398242280600D+00
.7187714031754280D+00	**	33464927/	46558512	**	3326763/	4628402	**	.7187714031754370D+00
.6687714031754280D+00	**	155685007/	232792560	**	9960169/	14893234	**	.6687714031754290D+00

Y : VALEUR EN VIRGULE FLOTTANTE A CONVERTIR

IN/ID:VALEUR EXACTE DE Y EN RATIONNELLE

IN/ID CALCULE:VALEUR RATIONNELLE DE Y CALCULEE PAR FRACTION CONTINUE

R :VALEUR EN VIRGULE FLOTTANTE DE IN/ID CALCULE

TABLEAU 4

Comparaison des résultats de APSIL et EPSIL pour la suite 1/N

Elément de la table EPSILON	Valeurs données par APSIL	Valeur donnée par EPSIL		
		Numérateur	Dénominateur	Valeur en réel
ε_0^0	0.100000000000000D+0	1	1	0.100000000000000D+1
ε_0^1	0.500000000000000D+0	1	2	0.500000000000000D+0
ε_2^0	0.250000000000000D+0	1	4	0.250000000000000D+0
ε_2^1	0.166666666666667D+0	1	6	0.166666666666667D+0
ε_4^0	0.111111111111111D+0	1	9	0.111111111111111D+0
ε_4^1	0.833333333333334D-1	1	12	0.833333333333334D-1
ε_6^0	0.625000000000000D-1	1	16	0.625000000000000D-1
ε_6^1	0.500000000000000D-1	1	20	0.500000000000000D-1
ε_8^0	0.399999999999983D-1	1	25	0.400000000000000D-1
ε_8^1	0.333333333334034D-1	1	30	0.333333333333333D-1
ε_{10}^0	0.277777777774389D-1	1	36	0.277777777777778D-1
ε_{10}^1	0.238095238107490D-1	1	42	0.238095238095238D-1
ε_{12}^0	0.20408163258885 D-1	1	49	0.204081632653061D-1
ε_{12}^1	0.178571428892341D-1	1	56	0.178571428571429D-1
ε_{14}^0	0.156249998067560D-1	1	64	0.156250000000000D-1
ε_{14}^1	0.138888897898642D-1	1	72	0.138888888888889D-1
ε_{16}^0	0.123456746110631D-1	1	81	0.123456790123457D-1
ε_{16}^1	0.111111275992232D-1	1	90	0.111111111111111D-1
ε_{18}^0	0.999993348482543D-1	1	100	0.100000000000000D-1
ε_{18}^1	0.909112654150919D-1	1	110	0.909090909090900D-1

TABLEAU 5

Comparaison des résultats de APSIL et EPSIL pour la série de terme général $\frac{(-1)^{N+1}}{N+1}$

Elément de la table EPSILON	Valeurs données par APSIL	Valeur donnée par EPSIL		
		Numérateur	Dénominateur	Valeur en réel
ε_0^0	0.100000000000000D+1	1	1	0.100000000000000D+1
ε_0^1	0.500000000000000D+0	1	2	0.500000000000000D+0
ε_2^0	0.700000000000000D+0	7	10	0.700000000000000D+0
ε_2^1	0.690476190476190D+0	29	42	0.690476190476190D+0
ε_4^0	0.693333333333333D+0	52	75	0.693333333333333D+0
ε_4^1	0.693089430894309D+0	341	492	0.693089430894309D+0
ε_6^0	0.6931524547803620D+0	1073	1548	0.693152454780362 D+0
ε_6^1	0.693145743145743D+0	9607	13860	0.693145743145743D+0
ε_8^0	0.693147332354381D+0	14161	20430	0.693147332354381D+0
ε_8^1	0.693147142487717D+0	13402	19335	0.693147142487717D+0
ε_{10}^0	0.693147184962132D+0	D.P.	D.P.	
ε_{10}^1	0.693147179517777D+0	D.P.	D.P.	
ε_{12}^0	0.693147180688164D+0	D.P.	D.P.	
ε_{12}^1	0.693147180530854D+0	D.P.	D.P.	
ε_{14}^0	0.69314718056369 D+0	D.P.	D.P.	
ε_{14}^1	0.693147180559123D+0	D.P.	D.P.	
ε_{16}^0	0.693147180560055D+0	D.P.	D.P.	
ε_{16}^1	0.693147180559922D+0	D.P.	D.P.	

* D.P. désigne le dépassement de capacité

N.B. : La précision PR pour le sous-programme 'RAT' a été choisie égale à 10^{-7}.

A N N E X E   1

ENSEMBLE DES SOUS-PROGRAMMES
INTERVENANT DANS LES CODES 'RAT' ET 'RATS',
AINSI QUE LE PROGRAMME D'APPEL

```
C*********************************************************************
C  ALGORITHME DE CONVERSION D UN REEL X EN RATIONNEL
C*********************************************************************
       COMMON NM,NI,PR
       DOUBLE PRECISION X,R1
100    FORMAT(1H1,//////////////////////,40X,'Y : VALEUR EN VIRGULE FLOTT
      1ANTE A CONVERTIR',//,40X,'IN/ID:VALEUR EXACTE DE Y EN RATIONNELLE'
      1,//,40X,'IN/ID CALCULE:VALEUR RATIONNELLE DE Y CALCULEE PAR FRACTI
      1ON CONTINUE',//,40X,'R :VALEUR EN VIRGULE FLOTTANTE DE IN/ID CALCU
      1LE')
104    FORMAT(1H1,26X,'ALGORITHME DES FRACTIONS CONTINUES AVEC DES REELS
      1SIMPLE  PRECISION,LE CALCUL EST',/,29X,'ARRETE DES QUE LA SIMPLE P
      1RECISION EST ATTEINTE PAR LE RESULTAT(10E-6)')
105    FORMAT(1H1,26X,'ALGORITHME DES FRACTIONS CONTINUES AVEC DES REELS
      1DOUBLE PRECISION,LE CALCUL EST',/,29X,' ARRETE DES QUE LA DOUBLE
      1PRECISION EST ATTEINTE PAR LE RESULTAT(10D-14)')
106    FORMAT(1H1,26X,'ALGORITHME DES FRACTIONS CONTINUES AVEC DES REELS
      1DOUBLE PRECISION,LE CALCUL EST',/,29X,' ARRETE DES QUE LA SIMPLE
      1PRECISION EST ATTEINTE PAR LE RESULTAT(10E-7)')
50     FORMAT(///,13X,'Y',23X,'IN/ID',21X,'IN/ID CALCULE',21X,'R')
51     FORMAT(6X,E15.6,4X,'**',2(I12,'/',I12,3X,'**'),D22.15)
52     FORMAT(E15.6,3X,'**',I12,'/',I12,3X,'**','DEPASSEMENT DE CAPACITEE
      1')
53     FORMAT(D22.15,3X,'**',2(I12,'/',I12,3X,'**'),D22.15)
54     FORMAT(D22.15,3X,'**',I12,'/',I12,3X,'**','DEPASSEMENT DE CAPACITE
      1E')
       NI=108
       NM=2147483647
C*********************************************************************
C X DOUBLE PRECISION ,LE RESULTAT EN SIMPLE PRECISION
C*********************************************************************
       PR=1.E-7
       WRITE(NI,100)
       WRITE(NI,106)
       WRITE(NI,50)
       X=0.
       IA=0
       IB=1
       DO 2 I=1,20
       ITA=0
       IS=1
       IF(2*(I/2).EQ.I) IS=-1
       IA=IA*I+IS*IB
       IB=IB*I
       CALL PGCD(IA,IB,NN)
       X=X+DFLOAT(IS)/I
       CALL RAT(X,IN,ID,R1,ITA)
       IF(ITA.EQ.1) GO TO 444
       WRITE(NI,53) X ,IA,IB,IN,ID,R1
       GO TO 2
444    WRITE(NI,54) X,IA,IB
2      CONTINUE
C*********************************************************************
C X DOUBLE PRECISION ,LE RESULTAT EN DOUBLE PRECISION
C*********************************************************************
       WRITE(NI,105)
       WRITE(NI,50)
       PR=1.E-14
       X=0.
       IB=1
       IA=0
       DO 3 I=1,20
       ITA=0
       IS=1
       IF(2*(I/2).EQ.I) IS=-1
       IA=IA*I+IS*IB
       IB=IB*I
       CALL PGCD(IA,IB,NN)
       X=X+DFLOAT(IS)/I
```

```
      CALL RAT(X,IN,ID,R1,ITA)
      IF(ITA.EQ.1) GO TO 555
      WRITE(NI,53) X ,IA,IB,IN,ID,R1
      GO TO 3
555   WRITE(NI,54) X,IA,IB
3     CONTINUE
C***********************************************************************
C X SIMPLE PRECISION ,LE RESULTAT EN SIMPLE PRECISION
C***********************************************************************
      PR=1.E-6
      WRITE(NI,104)
      WRITE(NI,50)
      X1=0
      IB=1
      IA=0
      DO 1 I=1,20
      IS=1
      ITA=0
      IF(2*(I/2).EQ.I) IS=-1
      IA=IA*I+IS*IB
      IB=IB*I
      CALL PGCD(IA,IB,NN)
      X1=X1+FLOAT(IS)/I
      CALL RATS(X1,IN,ID,R1,ITA)
      IF(ITA.EQ.1) GO TO 333
      WRITE(NI,51) X1,IA,IB,IN,ID,R1
      GO TO 1
333   WRITE(NI,52) X1,IA,IB
1     CONTINUE
      END
```

```
      SUBROUTINE RAT(X,IN,ID,R,ITA)
C***********************************************************************
C***CONVERSION D UN REEL X EN UN RATIONNEL IN/ID PAR LA METHODE DES
C***FRACTIONS CONTINUES,X EST DONNEE EN DOUBLE PRECISION
C***********************************************************************
8     FORMAT('LE TERME DE LA SUITE DE RANG',I3,3X,'EGAL A',D15.7,5X,'ETA
     1NT TROP PETIT CRE UN DEPASSEMENT DE CAPACITE')
9     FORMAT('LE TERME DE RANG',I3,3X,'X=',D15.7,5X,'A UNE PARTIE ENTIER
     1E QUI DEPASSE LA CAPACITE MACHINE')
10    FORMAT('L ERREUR RELATIVE COMMISE PAR LE CHOIX DE CETTE REPRESANTA
     1TION EST DE ',D15.7)
C***********************************************************************
      DOUBLE PRECISION Y,X,Z,R,V,A,B,RNM
      COMMON NM,NI,PR
C***********************************************************************
C INITIALISATIONS
C***********************************************************************
      RNM=NM
      IF(DABS(X).LT.(1/RNM)) GO TO 4
      IF(DABS(X).GE.(RNM+1)) GO TO 5
      IS=1
      IF(X.LT.0.)IS=-1
      Y=IS*X
      Z=Y
      N=1
      M=0
      IN=Y
      ID=1
      V=Y-IN
      R=DFLOAT(IN)/ID
      IF(V.EQ.0) GO TO 3
```

```
C************************************************************************
C CALCUL DES REDUITES DE LA FRACTION CONTINUE
C************************************************************************
1       Y=1./V
        J=Y
        V=Y-J
        A=DFLOAT(J)*IN+N
        IF(A.GT.NM)GO TO 2
        IR=A
        B=DFLOAT(J)*ID+M
        IF(B.GT.NM)GO TO 2
        N=IN
        IN=IR
        M=ID
        ID=B
        R=DFLOAT(IN)/ID
        IF(DABS(R-Z).LT.PR) GO TO 3
        GO TO 1
C************************************************************************
C CRITERE DE SORTIE
C************************************************************************
2       ITA=1
        RETURN
3       IN=IN*IS
        RETURN
4       WRITE(NI,8) I,X
        STOP
5       WRITE(NI,9) I,X
        STOP
        END
```

```
        SUBROUTINE RATS(X,IN,ID,R,ITA)
C************************************************************************
C***CONVERSION D UN REEL X EN UN RATIONNEL IN/ID PAR LA METHODE DES
C***FRACTIONS CONTINUES,X EST DONNEE EN SIMPLE PRECISION
C************************************************************************
        DOUBLE PRECISION A,B,RNM,R
        COMMON NM,NI,PR
C************************************************************************
C INITIALISATIONS
C************************************************************************
        RNM=NM
        IS=1
        IF(X.LT.0.)IS=-1
        Y=IS*X
        Z=Y
        N=1
        M=0
        IN=Y
        ID=1
        V=Y-IN
        R=DFLOAT(IN)/ID
        IF(V.EQ.0) GO TO 3
C************************************************************************
C CALCUL DES REDUITES DE LA FRACTION CONTINUE
C************************************************************************
1       Y=1./V
        J=Y
        V=Y-J
        A=DFLOAT(J)*IN+N
        IF(A.GT.NM)GO TO 2
        IR=A
```

```
      B=DFLOAT(J)*ID+M
      IF(B.GT.NM)GO TO 2
      N=IN
      IN=IR
      M=ID
      ID=B
      R=DFLOAT(IN)/ID
      IF(DABS(R-Z).LT.PR) GO TO 3
      GO TO 1
C***********************************************************************
C CRITERE DE SORTIE
C***********************************************************************
2     ITA=1
      RETURN
3     IN=IN*IS
      RETURN
      END
```

```
      SUBROUTINE PGCD(IN,ID,IP)
C***********************************************************************
C SIMPLIFICATION DE LA FRACTION IN/ID PAR LE CALCUL DE SON PGCD PAR
C L ALGORITHME D EUCLIDE
C***********************************************************************
      IF(ID.EQ.0) GO TO 7
      ISN=1
      ISD=1
      IF(IN.LT.0) ISN=-1
      IN=IN*ISN
      IF(ID.LT.0) ISD=-1
      ID=ID*ISD
      IA=IN
      IB=ID
3     IQ=IA/IB
      IR=IA-IQ*IB
      IF(IR) 4,6,4
4     CONTINUE
      IF(IR.EQ.1) GO TO 5
      IA=IB
      IB=IR
      GO TO 3
5     IP=1
      IN=ISN*IN
      ID=ISD*ID
      RETURN
6     IP=IB
      IN=ISN*IN/IB
      ID=ISD*ID/IB
      RETURN
7     WRITE(108,8) IN,ID
8     FORMAT('RATIONNEL FAUX  SON DENOMINATEUR EST NUL',5X,'NUMERATEUR='
     1,I10,5X,'DENOMINATEUR=',I10)
      STOP
      END
```

A N N E X E 2

ENSEMBLE DES SOUS-PROGRAMMES
INTERVENANT DANS LES CODES 'APSIL' ET 'EPSIL',
AINSI QUE LE PROGRAMME D'APPEL

```
C***********************************************************************
C*******************PROGRAMME DE L E-ALGORITHME RATIONNEL ET DE
C*******************L E-ALGORITHME VIRGULE FLOTTANTE POUR LA SUITE 1/N
C***********************************************************************
C X:TERME DE LA SUITE INITIALE
C Y: LIMITE EXACTE DE LA SUITE
C ZE: REMPLACE UN DIVISEUR NUL DANS LE SOUS PROGRAMME -APSIL-
C NM : CAPACITE MAXIMUM DE L ORDINATEUR EN ENTIER
C NI : ETIQUETTE D IMPRESSION
C PR : PRECISION DU RESULTAT DANS LE SOUS PROGRAMME-RAT-
C ICM : INDICE DE LA COLONNE MAXIMUM QUE L ON VEUT CALCULER
C IK1,IK2 : PARAMETRES A METTRE A ZERO  A L APPEL DU SOUS PROGRAMME POUR
C  UNE NOUVELLE SUITE
C  MM : DIMENSION DU TABLEAU REPRESENTANT UNE DIAGONALE
C***********************************************************************
      DOUBLE PRECISION X,Y,ZE
      INTEGER PN(21),PD(21),DN(21),DD(21)
      DOUBLE PRECISION P(21),D(21)
      COMMON/BLOK/NM,NI,PR
100   FORMAT(' RANG DE LA SUITE =',I2)
103   FORMAT(' LA LIMITE EXACTE DE LA SUITE=',D21.15,///)
105   FORMAT(' FIN******************************')
106   FORMAT(1H1,40(1H*),'ETUDE DE LA SUITE 1/N',/////)
107   FORMAT(1H1,40(1H*),'ETUDE DE LA SERIE DE TERME GENERAL EGAL A ((-1
     1)**(N+1))/N',/////)
      ZE=1.D-15
      ICM=20
      IK1=0
      IK2=0
      MM=21
      WRITE(NI,106)
      Y=0.
      WRITE(NI,103) Y
      X=0.
      DO 3 I=1,20
      WRITE(NI,100) I
      X=DFLOAT(1)/I
      CALL EPSIL(X,ICM,IK1,MM,PN,PD,DN,DD)
      CALL APSIL(X,ICM,IK2,MM,ZE,P,D)
3     WRITE(NI,105)
      END
```

```
      BLOCK DATA
      COMMON/BLOK/NM,NI,PR
      DATA NM,NI,PR/2147483647,108,1.E-7/
      END
```

```
C*************************************************************************
C********************PROGRAMME DE L E-ALGORITHME RATIONNEL ET DE
C********************L E-ALGORITHME VIRGULE FLOTTANTE POUR LA SERIE DE
C*********************TERME GENERAL EGAL A ((-1)**(N+1))/N
C*************************************************************************
C X:TERME DE LA SUITE INITIALE
C Y: LIMITE EXACTE DE LA SUITE
C ZE: REMPLACE UN DIVISEUR NUL DANS LE SOUS PROGRAMME -APSIL-
C NM : CAPACITE MAXIMUM DE L ORDINATEUR EN ENTIER
C NI : ETIQUETTE D IMPRESSION
C PR : PRECISION DU RESULTAT DANS LE SOUS PROGRAMME-RAT-
C ICM : INDICE DE LA COLONNE MAXIMUM QUE L ON VEUT CALCULER
C IK1,IK2 : PARAMETRES A METTRE A ZERO  A L APPEL DU SOUS PROGRAMME POUR
C  UNE NOUVELLE SUITE
C  MM : DIMENSION DU TABLEAU REPRESENTANT UNE DIAGONALE
C*************************************************************************
      DOUBLE PRECISION X,Y,ZE
      INTEGER PN(21),PD(21),DN(21),DD(21)
      DOUBLE PRECISION P(21),D(21)
      COMMON/BLOK/NM,NI,PR
100   FORMAT(' RANG DE LA SUITE =',I2)
103   FORMAT(' LA LIMITE EXACTE DE LA SUITE=',D21.15,///)
105   FORMAT(' FIN*******************************')
106   FORMAT(1H1,40(1H*),'ETUDE DE LA SUITE 1/N',/////)
107   FORMAT(1H1,40(1H*),'ETUDE DE LA SERIE DE TERME GENERAL EGAL A ((-1
     1)**(N+1))/N',/////)
      ZE=1.D-15
      ICM=20
      IK1=0
      IK2=0
      MM=21
      WRITE(NI,107)
      Y=DLOG(2)
      WRITE(NI,103) Y
      X=0.
      DO 1 I=1,20
      WRITE(NI,100) I
      IS=1
      IF(2*(I/2).EQ.I) IS=-1
      X=X+DFLOAT(IS)/I
      CALL EPSIL(X,ICM,IK1,MM,PN,PD,DN,DD)
      CALL APSIL(X,ICM,IK2,MM,ZE,P,D)
1     WRITE(NI,105)
      END
```

```
      BLOCK DATA
      COMMON/BLOK/NM,NI,PR
      DATA NM,NI,PR/2147483647,108,1.E-7/
      END
```

```
      SUBROUTINE EPSIL(X,ICM,IK,MM,PN,PD,DN,DD)
C***********************************************************************
C CALCUL DE LA TABLE EPSILON EN RATIONNEL
C***********************************************************************
C CARTES A CHANGER EVENTUELLEMENT PAR L UTILISATEUR
C***********************************************************************
      INTEGER TN,TD,RN,RD,SN,SD,BN,BD,ETA
      INTEGER PN(MM),PD(MM),DN(MM),DD(MM)
      DOUBLE PRECISION X,R
103   FORMAT(' DIMENSION INSUFFISANTE DANS LE TABLEAU EPSILON,AUGMENTER
     1MM ET REAJUSTER LA DIMENSION DE PN,PD,DN,DD')
101   FORMAT('  EPSILON(',I2,',',I2,')=',I15,'/',I15,'  =',D22.15,/)
102   FORMAT('  EPSILON(',I2,',',I2,')=','DEPASSEMENT DE CAPACITE QUI SE
     1 PRODUIT AU',I3,' TERME DE LA DIAGONALE')
104   FORMAT(' EPSILON RATIONNEL')
      COMMON/BLOK/NM,NI,PR
C***********************************************************************
C INITIALISATIONS
C***********************************************************************
      CALL RAT(X,TN,TD)
      IF(IK.NE.0) GOTO 1
      IK=1
      I1=0
      RN=TN
      RD=TD
      R=DFLOAT(RN)/RD
      PN(1)=TN
      PD(1)=TD
      M=1
      MI1=M
      L=M
      IF(2*(ICM/2).NE.ICM) ICM=ICM-1
      GO TO 66
C***********************************************************************
C CALCUL DE LA DIAGONALE MONTANTE DN/DD CORRESPONDANT AU TERME X DE LA
C SUITE DANS LA TABLE EPSILON  LA DIAGONALE PRECEDANTE ETANT PN/PD
C***********************************************************************
1     IF(I1.EQ.1) RETURN
      M=M+1
      MI1=MI1+1
      IF(M.GT.(ICM+1)) M=ICM+1
      IF(M.GT.MM) GO TO 9
      DN(1)=TN
      DD(1)=TD
      DO 2 J=2,M
      DN(J)=0
2     DD(J)=0
      DO 3 J=2,M
      BN=0
      BD=1
      IF(J.EQ.2) GO TO 11
      BN=PN(J-2)
      BD=PD(J-2)
11    JDN=DN(J-1)
      JDD=DD(J-1)
      JPN=-PN(J-1)
      JPD=PD(J-1)
      CALL ADD(SD,SN,JDN,JDD,JPN,JPD,ETA)
      IF(ETA.EQ.1) GO TO 4
      CALL ADD(DN(J),DD(J),SN,SD,BN,BD,ETA)
      IF(ETA.EQ.1) GO TO 4
3     CONTINUE
C***********************************************************************
C CRITERES DE SORTIE
C***********************************************************************
```

```
4     L=M
      IF(2*(M/2).EQ.M) L=M-1
      RN=DN(L)
      RD=DD(L)
      IS=1
      IF(RN.LT.0.AND.RD.LT.0) IS=-1
      RN=IS*RN
      RD=IS*RD
      R=0.
      DO 6 I=1,M
      PN(I)=DN(I)
6     PD(I)=DD(I)
      IF(RD.NE.0) R=DFLOAT(RN)/RD
66    WRITE(NI,104)
      JJ=L-1
      II=MI1-JJ-1
      IF(DABS(R).LT.PR) GO TO 7
      WRITE(NI,101) II,JJ,RN,RD,R
      GO TO 8
7     WRITE(NI,102) II,JJ,J
8     CONTINUE
      RETURN
9     WRITE(NI,104)
      WRITE(NI,103)
      I1=1
      RETURN
      END
```

```
      SUBROUTINE RAT(X,IN,ID)
C**********************************************************************
C***CONVERSION D UN REEL X EN UN RATIONNEL IN/ID PAR LA METHODE DES
C***FRACTIONS CONTINUES
C**********************************************************************
6     FORMAT(' DEPASSEMENT DE CAPACITE DANS LE S.P. DE CONVERSION FLOTTA
     1NT-RATIONNEL',/,D22.15,5X,'SERA REPRESENTE PAR',I15,'/',I15,'=',D2
     12.15)
7     FORMAT(' LE TERME DE LA SUITE',D22.15,10X,'SERA REPRESENTE  PAR ',
     1/,I12,'/',I12,'=',D22.15)
8     FORMAT(' LE TERME DE LA SUITE DE RANG',I3,3X,'EGAL A',D22.15,5X,'E
     1TANT TROP PETIT CRE UN DEPASOEMENT DE CAPACITE')
9     FORMAT(' LE TERME DE RANG',I3,3X,'X=',D22.15,5X,'A UNE PARTIE ENTI
     1ERE QUI DEPASSE LA CAPACITE MACHINE')
10    FORMAT(' L ERREUR RELATIVE COMMISE PAR LE CHOIX DE CETTE REPRESANT
     1ATION EST DE ',/,D22.15,/)
C**********************************************************************
      DOUBLE PRECISION Y,X,Z,R,V,A,B,RNM
      COMMON/BLOK/NM,NI,PR
C**********************************************************************
C INITIALISATIONS
C**********************************************************************
      RNM=NM
      IF(DABS(X).LT.(1/RNM)) GO TO 4
      IF(DABS(X).GE.(RNM+1)) GO TO 5
      IS=1
      IF(X.LT.0.)IS=-1
      Y=IS*X
      Z=Y
      N=1
      M=0
      IN=Y
      ID=1
      V=Y-IN
      R=DFLOAT(IN)/ID
      IF(V.EQ.0) GO TO 3
```

```
C***********************************************************************
C CALCUL DES REDUITES DE LA FRACTION CONTINUE
C***********************************************************************
1     Y=1./V
      J=Y
      V=Y-J
      A=DFLOAT(J)*IN+N
      IF(A.GT.NM)GO TO 2
      IR=A
      B=DFLOAT(J)*ID+M
      IF(B.GT.NM)GO TO 2
      N=IN
      IN=IR
      M=ID
      ID=B
      R=DFLOAT(IN)/ID
      IF(DABS(R-Z).LT.PR) GO TO 3
      GO TO 1
C***********************************************************************
C CRITERE DE SORTIE
C***********************************************************************
2     IN=IN*IS
      R=DFLOAT(IN)/ID
      WRITE(NI,6)X,IN,ID,R
      Z=DABS((R-Z)/Z)
      WRITE(NI,10) Z
      RETURN
3     IN=IN*IS
      WRITE(NI,7) X,IN,ID,R
      Z=DABS((Z-R)/Z)
      WRITE(NI,10) Z
      RETURN
4     WRITE(NI,8) I,X
      STOP
5     WRITE(NI,9) I,X
      STOP
      END
```

```
      SUBROUTINE ADD(INA,IDA,INB,IDB,INC,IDC,ETA)
C***********************************************************************
C CALCUL DE LA SOMME DE DEUX  NOMBRES RATIONNELS INB/IDB ET INC/IDC LE
C RESULTAT EST INA/IDA
C ETA EST UNE VARIABLE D ETAT QUI EST MISE A 1 QUAND ON A UN DEPASSEMENT
C DE CAPACITE LORS DU CALCUL ET DANS CE CAS INA=IDA=0
C***********************************************************************
      DOUBLE PRECISION T,DA
      INTEGER ETA
      COMMON/BLOK/NM,NI
      ETA=0
      IDDC=IDC
      CALL PGCD(IDB,IDC,ID)
      T=DFLOAT(INB)*IDC+IDB*INC
      IF(DABS(T).GT.NM) GO TO 2
      IT=T
      DA=DFLOAT(IDB)*IDC
      IF(DABS(DA).GT.NM) GO TO 2
      IDA=DA
      IF(ID.GT.1) GO TO 1
      INA=IT
      RETURN
```

```
 1      CALL PGCD(IT,ID,ID2)
        INA=IT
        IDA=(IDDC/ID2)*IDB
        RETURN
 2      ETA=1
        INA=0
        IDA=0
        RETURN
        END
```

```
        SUBROUTINE PGCD(IN,ID,IP)
C***********************************************************************
C SIMPLIFICATION DE LA FRACTION IN/ID PAR LE CALCUL DE SON PGCD PAR
C L ALGORITHME D EUCLIDE
C***********************************************************************
        IF(ID.EQ.0) GO TO 7
        ISN=1
        ISD=1
        IF(IN.LT.0) ISN=-1
        IN=IN*ISN
        IF(ID.LT.0) ISD=-1
        ID=ID*ISD
        IA=IN
        IB=ID
 3      IQ=IA/IB
        IR=IA-IQ*IB
        IF(IR) 4,6,4
 4      CONTINUE
        IF(IR.EQ.1) GO TO 5
        IA=IB
        IB=IR
        GO TO 3
 5      IP=1
        IN=ISN*IN
        ID=ISD*ID
        RETURN
 6      IP=IB
        IN=ISN*IN/IB
        ID=ISD*ID/IB
        RETURN
 7      WRITE(108,8) IN,ID
 8      FORMAT('RATIONNEL FAUX  SON DENOMINATEUR EST NUL',5X,'NUMERATEUR='
       1,I10,5X,'DENOMINATEUR=',I10)
        STOP
        END
```

```
        SUBROUTINE APSIL(T,ICM,IK,MM,ZE,P,D)
C***********************************************************************
C CALCUL DE LA TABLE DE L E-ALGORITHME EN VIRGULE FLOTTANTE DOUBLE
C PRECISION
C***********************************************************************
C CARTES A CHANGER EVENTUELLEMENT PAR L UTILISATEUR
C***********************************************************************
        DOUBLE PRECISION B,T,R,S,P(MM),D(MM)
        COMMON/BLOK/NM,NI
```

```
111   FORMAT(' DIMENSIONS INSUFISANTES DANS LA TABLE EPSILON')
102   FORMAT('  EPSILON(',I2,',',I2,')=',D22.15)
103   FORMAT(' EPSILON REEL VIRGULE FLOTTANTE')
C*********************************************************************
C INITIALISATIONS
C*********************************************************************
      IF(IK.NE.0) GOTO 1
      IK=1
      I1=0
      R=T
      P(1)=T
      M=1
      L=M
      MI=M
      IF(2*(ICM/2).NE.ICM) ICM=ICM-1
      GO TO 7
C*********************************************************************
C CALCUL DE LA DIAGONALE MONTANTE D CORRESPONDANT AU TERME X DE LA SUITE
C SUITE DANS LA TABLE EPSILON  LA DIAGONALE PRECEDANTE ETANT P
C*********************************************************************
1     IF(I1.EQ.1) RETURN
      M=M+1
      MI=MI+1
      IF(M.GT.(ICM+1)) M=ICM+1
      IF(M.GT.MM) GOTO 6
      D(1)=T
      DO 2 J=2,M
      B=0.
      IF(J.NE.2) B=P(J-2)
      S=D(J-1)-P(J-1)
      IF(DABS(S).LT.ZE) S=ZE
2     D(J)=B+1./S
C*********************************************************************
C OPERATIONS DE SORTIE
C*********************************************************************
      L=M
      IF(2*(M/2).EQ.M) L=M-1
      R=D(L)
      DO 5 J=1,M
5     P(J)=D(J)
7     WRITE(NI,103)
      JJ=L-1
      II=MI-JJ-1
      WRITE(NI,102) II,JJ,R
      RETURN
6     WRITE(NI,111)
      I1=1
      RETURN
      END
```

CHAPITRE II

L'ARITHMETIQUE RATIONNELLE ETENDUE

I - INTRODUCTION

L'arithmétique rationnelle est une arithmétique dont les données associées aux opérateurs sont des nombres rationnels. Une telle arithmétique est une arithmétique exacte. Elle a été utilisée dans le chapitre précédent pour le calcul de l'ε-algorithme, ce qui a permis d'obtenir des résultats exacts. Seulement l'utilisation d'une telle arithmétique entraîne la manipulation de nombres entiers qui deviennent de plus en plus grands au fur et à mesure que le calcul se déroule et très vite on atteint la capacité maximale de la machine pour des variables entières. Pour résoudre ce problème on a été amené à définir des entiers écrits sur plusieurs mots mémoire de la machine et de coder les différents opérateurs de l'arithmétique rationnelle sur ces données : c'est l'arithmétique rationnelle étendue.

II - ARITHMÉTIQUE RATIONNELLE ÉTENDUE

II-1 Définition

Un entier N en base 10 s'écrit sous la forme :

$N = S(N)\ (n_p * 10^p + \ldots + n_1 * 10 + n_0)$ avec $\left\{ \begin{array}{l} 0 \le n_i < 10 \text{ pour } i=0,\ldots,p \\ \text{et } n_i \text{ entier} \end{array} \right.$

S(N) désigne le signe de N.

On note symboliquement N par $N = S(N)\ (n_p \ldots n_0)_{10}$; et, en général, dans une base quelconque B on aura :

$N = S(N)(N_p * B^p + \ldots + N_1 * B + N_0)$ avec $0 \le N_i < B$ pour $i=0,\ldots,p$ et N_i entier.

Dans le cas où B est une puissance de 10, $B = 10^k$, on aura les relations suivantes :

$$
\begin{aligned}
N_0 &= (n_{k-1}\ n_{k-2} \cdots n_0) \\
N_1 &= (n_{2k-1}\ n_{2k-2} \cdots n_k) \\
&\vdots \\
N_i &= (n_{(i+1)k-1} \cdots n_{ik})
\end{aligned}
$$

et ceci jusqu'au dernier chiffre n_p.

Dans un ordinateur la représentation des entiers étant limitée par la capacité d'un mot mémoire (sur l'IRIS 80 elle est de $2^{31}-1$), pour représenter des entiers plus grands on définit un tableau de longueur L dont chaque mot contient un N_i c'est ce que l'on appelle tableau entier paramétré (TEP) de longueur variable

L+1 en base B. Le choix de cette base permet de diviser par K la zone mémoire nécessaire à la représentation d'un TEP.

On va maintenant définir les différents opérateurs : somme, différence, produit et division entière sur les TEP.

Soient deux TEP de longueur L+2 par rapport à la base B notés :

$$N = S(N)(N_L * B^L + \dots + N_1 * B + N_0) \quad \text{avec} \quad 0 \le N_i < B \quad \text{et} \quad N_i \text{ entier}$$
$$M = S(M)(M_L * B^L + \dots + M_1 * B + M_0) \quad \text{avec} \quad 0 \le M_i < B \quad \text{et} \quad M_i \text{ entier}$$

le premier mot du tableau N contient le signe S(N), il est égal à +1 ou -1 suivant que N est positif ou négatif. Le second mot N(2) porte la valeur N_L et ainsi de suite jusqu'au mot N(L+2) qui mémorisera N_0. Le même principe est utilisé pour M.

II-2 Addition

On suppose que N et M sont de même signe, sinon l'opération est une soustraction. La somme S de N et M est égale à :

$$S = N + M = S(S)(S_L * B^L + \dots + S_0)$$

avec S(S) = S(N)

$$S_0 = (N_0 + M_0) \bmod (B)$$
$$r_0 = [(N_0 + M_0)/B]$$
$$S_i = (N_i + M_i + r_{i-1}) \bmod (B) \text{ pour } i = 1,\dots, L$$
$$r_i = [(N_i + M_i + r_{i-1})/B] \quad \text{pour } i = 1,\dots, L$$

Dans le cas où $r_L \neq 0$ on considère qu'il y a un débordement dans le tableau S, le calcul sera arrêté et un message d'erreur signalera un dépassement de capacité.

II-3 Soustraction

On suppose que N et M sont de même signe sinon l'opération devient une addition. Prenons le cas où ils sont tous les deux positifs, le cas contraire s'étudie bien sûr de la même façon. On suppose que N ≥ M.

La différence D de N et M est égale à :

$$D = N - M = S(D)(D_L * B^B + \ldots + D_0)$$

avec S(D) = S(N)

$$D_0 = \begin{cases} N_0 - M_0 & \text{si } N_0 \geq M_0 \\ N_0 + B - M_0 & \text{si } N_0 < M_0 \quad \text{et dans ce cas} \\ \multicolumn{2}{l}{\text{on rajoute 1 à } M_1 \text{ et on recommence l'opération pour}} \\ \multicolumn{2}{l}{\text{les autres indices jusqu'à } D_L.} \end{cases}$$

Dans le cas où N < M on fait D = - (M-N).

Dans le cas où N et M sont tous les deux négatifs on a :

si $N \leq M$ alors $D = -(M-N)$
si $N > M$ alors $D = N - M$

II-4 Multiplication

Le produit P de N et M est égal à :

$$P = N * M = S(P)(P_{2L} * B^{2L} + \ldots + P_0)$$

Cette opération se fait de la façon suivante :

$$\begin{array}{rrrrr}
 & & N_L \ldots\ldots\ldots & N_0 & \\
 & & M_L \ldots\ldots\ldots & M_0 & \\
\hline
 & & P_{0L} \ldots\ldots & P_{00} & \\
 & P_{1L} \ldots\ldots & P_{10} & & \\
 & \vdots \quad \vdots & & & \\
P_{1L} \ldots\ldots & P_{1L} & & & \\
\hline
P_{2L} \ldots\ldots & P_L \ldots\ldots & P_0 & &
\end{array}$$

avec S(P) = S(N) * S(M)

$$\left.\begin{array}{ll} P_{i0} = (N_0 * M_i) \bmod (B) & \\ r_{i0} = [(N_0 * M_i)/B] & \\ P_{ij} = ((N_j * M_i) + r_{ij}) \bmod (B) & j = 1,\dots,L \end{array}\right\} \text{pour } i=0,\dots,L$$

et

$$P_0 = P_{00} \bmod (B)$$

$$r_0 = [P_{00}/B]$$

$$\left.\begin{array}{l} P_i = \left(\sum\limits_{k+j=i} P_{jk} + r_{i-1}\right) \bmod (B) \\ r_i = P_i \bmod (B) \end{array}\right\} \text{pour } i = 1,\dots,$$

Seulement comme les entiers sont définis par des T.E.P. de longueur L+2, dès que $r_{L0} \neq 0$ ou $r_L \neq 0$ les calculs seront arrêtés et un message d'erreur signalera un dépassement de capacité dans la multiplication. Il est évident que cette opération sera le plus souvent à l'origine des arrêts des algorithmes en raison du dépassement de capacité plus fréquent que dans l'addition.

II-5 Division entière

Soient deux entiers N et M définis par :

$$N = N_p * B^p + \dots + N_0 \quad \text{avec} \quad 0 \leq N_i < B \text{ pour } i = 0,\dots,p$$

$$M = M_k * B^k + \dots + M_0 \quad \text{avec} \quad 0 \leq M_i < B \text{ pour } i = 0,\dots,k$$

On suppose qu'ils sont de même signe et que $N \geq M$.

On cherche les entiers Q et R tel que :

$$N = Q * M + R$$

Dans le cas où N et M sont de signe contraire le calcul s'arrête et un message d'erreur est imprimé.

L'algorithme de la division entière est le suivant :

- On isole le premier mot des T.E.P. N et M de poids le plus fort noté N_p et M_k. On cherche leur quotient entier $C = [N_p/M_k]$. La valeur C est le mot de poids le plus fort de Q c'est-à-dire $Q_{p-k} = C$ ou $Q = C * B^{p-k}$.

- On calcule $Q' = M * (C * B^{p-k})$ et $R = N - Q'$. Si $R < M$ on est à la fin de l'algorithme, sinon on remplace N par R et on reprend le calcul.

<u>Exemple</u> : on prend B = 100, N = + 345750 et M = + 213. Les entiers N et M s'écrivent sous la forme d'un T.E.P. de longueur L = 4 :

N	+	34	57	50
M	+	00	02	13

<u>étape 1</u> : C1 = [34/2] = 17

Q	+	00	17	00
Q'	+	36	21	00
R	-	01	63	50

car $Q_{2-1} = Q_1 = 17$

R est négatif. Ceci veut dire que le quotient Q est surévalué, dans ce cas on remplace C par C-1 ce qui donne :

C = 16

Q	+	00	16	00
Q'	+	34	08	00
R	+	00	49	50

puisque $Q_{2-1} = Q_1 = 16$

Comme R > M on remplace N par R et on reprend l'algorithme.

<u>étape 2</u> : C = [49/2] = 24

Q	+	00	16	24
Q'	+	00	51	12
R	-	00	01	62

$Q_{1-1} = Q_0 = 24$

Là aussi R est négatif. On doit donc retrancher 1 à C :

C = 24 - 1 = 23

Q	+	00	16	23
Q'	+	00	48	99
R	+	00	00	51

Comme R est inférieur à M on est à la fin de l'algorithme et on a :

345750 = 213 * 1623 + 51

Remarque : Les algorithmes du calcul du plus grand commun diviseur et des quatre opérations de base sur les fractions rationnelles sont identiques à ceux présentés au chapitre I.

III - CODAGE DE L'ARITHMETIQUE RATIONNELLE ETENDUE

On a écrit l'ensemble des sous-programmes FØRTRAN permettant d'effectuer les quatre opérations arithmétiques sur les TEP.

III-1 Initialisation du code FORTRAN

Dans le but de définir un code général, certaines données sont introduites comme paramètres dans un bloc CØMMØN et doivent être fixées pour chaque programme d'appel par un sous-programme BLØKDATA. Ces paramètres sont :

IBASE : base dans laquelle sont écrits les TEP (voir paragraphe III-2)

LM : longueur maximale en chiffres des TEP intervenant dans le programme.

IN : numéro logique de l'imprimante.

KN : numéro logique du lecteur de carte.

De plus, pour tout les programmes d'appel, en tête des instructions exécutables, on fera appel au sous-programme d'initialisation INI :

CALL INI (L, LR).

III-2 Codage des nombres entiers paramétrés

a) Codage externe

Sur carte de données, tout entier $a_1 \ldots a_{k-1}$ sera représenté sous la forme suivante :

$\pm\, a_1 \ldots\ldots\ldots a_{k-1}\, b\!\!/_k$ avec $1 \leq k \leq LM$ et $b\!\!/$ désigne le caractère blanc

aucune limite n'est fixée pour LM ; Le sous-programme de lecture LEC (A, L, RA, LR) permet donc de lire n'importe quel entier écrit sur une ou plusieurs cartes de données à condition que la longueur L du TEP et la longueur LR du tableau RA aient été initialisées par INI(L, LR), sachant que le tableau RA doit être de dimension au moins égale à 80 caractères (nombre de caractère par carte de donnée).

Enfin l'entier $\pm\, a_1 \ldots a_{k-1}$ doit toujours être suivi d'un blanc $b\!\!/_k$ qui fait partie du nombre de caractères nécessaires à la représentation.

Sur listing le codage sera identique avec un nombre par ligne, à partir de la première colonne.

b) Codage interne

Un nombre entier N s'écrit dans une base quelconque B de la façon duivante :

$$N = S(N)\ (N_{p-2} * B^{p-2} + \ldots + N_1 * B + N_0)$$

S(N) est le signe de N

N_0 : les chiffres de poids le plus faible de N par rapport à la base B.

N_{p-2} : les chiffres de poids le plus fort de N par rapport à la base B.

Sur ordinateur cet entier sera représenté dans un tableau d'entiers de longueur L noté N(L). Le premier mot N(1) portera le signe de N, il sera égal à + 1 ou - 1 suivant que N est positif ou négatif. Le dernier mot N(L) mémorisera N_0, le mot N(L - 1) la valeur N_1 et ainsi de suite jusqu'au nombre N_{p-2} qui sera mémorisé dans le mot N(L - p+2). Le reste des mots du tableau de N(L - p+1) au mot N(2) sera complété de zéros. Pour la suite on appelera un mot l'une des valeurs N_i. Pour permettre la minimisation de la place mémoire occupée par ces entiers et la facilité de leur manipulation, la base B sera choisie comme une puissance de 10 et donc $N_{p-2},\ldots, N_0$ seront des entiers dont le nombre de chiffres en base 10 sera celui de (B - 1). De plus la base B sera fixée à une valeur maximale IBASE qui nous permettra d'effectuer les opérations arithmétiques élémentaires entre les différents mots de deux TEP sans qu'elles n'entraînent un dépassement de capacité. Sur l'IRIS 80 de la C.I.I. - H.B la valeur de IBASE est égale à 10^4.

c) Exemple

Considérons l'entier N égal à + 123456789 et la base de décomposition IBASE égale à 10^4. Dans ce cas l'entier N sera représenté par un TEP N(4) de la façon suivante :

+ 1	0001	2345	6789
N(1)	N(2)	N(3)	N(4)

III-3 Mode d'emploi des sous-programmes

Dans le but d'avoir des tableaux d'entiers (TEP) dynamiques, les variables de ce type même locales aux sous-programmes, sont déclarées en paramètres.

On désignera par L et LR les longueurs de ces tableaux.

L'arithmétique est évolutive par rapport à la longueur L, il suffit pour cela de fixer LM nombre de chiffres maximums des entiers et d'initialiser L et LR par le sous-programme INI.

*** INI ***

appel : CALL INI(L, LR)

paramètres : L longueur des TEP
LR longueur du tableau RA utilisé dans les sous-programmes LEC et ECR.

S.P. utilisé : néant

description : Ce SP fixe la longueur L des TEP nécessaire pour enregistrer les entiers de longueur LM, ainsi que la valeur LR.

remarque : L'appel à ce sous-programme doit toujours s'effectuer avant toute utilisation des autres sous-programmes.

III-3-1 Les sous-programmes primaires :

Ces sous-programmes sont ceux qui concernent le plus directement l'utilisateur. Ils font appel à une série d'autres sous-programmes, appelés des sous-programmes secondaires, dont certains aussi peuvent être utilisés directement par le programmeur.

*** LEC ***

appel : call LEC(A, L, RA, LR)

paramètres : A : TEP de longueur L
RA : (tableau en variable Hoolerigh de longeur LR)

S.P. utilisés : la fonction LOGBAS
sous-programme DIV 2

description : Ce SP lit la donnée sur cartes en mettant un caractère par mot dans le tableau RA, le critère d'arrêt sera le caractère blanc. Après on convertit les caractères du tableau RA en nombres entiers qu'on mémorisera dans le TEP A suivant la base IBASE.

remarque : La lecture des données se faisant carte par carte, la taille du tableau RA est égale à(L-1) * LOGBAS (IBASE) + 1 et au minimum à 80. La conversion des caractères de RA en chiffres se fait en utilisant le code EBCDIC. La dimension de RA doit rester un multiple de 80.

*** ECR ***

appel : call ECR(A, L, RA, LR)

paramètres : A : TEP de longueurs L
RA : tableau d'entiers de longueur LR

S.P. utilisé : la fonction LOGBAS

description : Ce SP permet la sortie sur l'imprimante, de code IN, du TEP A préalablement converti dans un tableau RA suivant la base 10.

remarque : La taille du tableau RA est égale à (L-1) * LOGBAS (IBASE) + 2 la conversion des mots du TEP A dans les mots du tableau RA se fait en utilisant le code EBCDIC.

*** STO ***

appel : CALL STO (A, B, L)
paramètres : A : entier ne dépassant pas la capacité maximale de l'ordinateur pour les variables entières.
B : un TEP de longueur L.

S.P. utilisé : néant

description : Ce SP mémorise un entier A dans un TEP B suivant la technique présentée au paragraphe III-2-b.

*** ADD ***

appel : call ADD (A,B,C,L,ITE)

paramètres : A,B,C : T.E.P. de longueur L
ITE : entier initialisé à zéro et prenant la valeur 1 dans le cas où il y a débordement dans le T.E.P. contenant le résultat de l'opération

S.P. utilisés: SOM, DIF, COMP

description : addition de A et B, le résultat est dans C. La longueur du T.E.P. C étant égale à L il y aura débordement, qui entraînera l'arrêt des calculs et l'impression d'un message d'erreur ainsi que le changement de la valeur de ITE à 1, dès que la retenue de la somme des mots de poids le plus fort d'indice 2 dans A et B est différente de zéro.

*** SOUS ***

appel : call SOUS (A,B,C,L)

paramètres : A,B,C : T.E.P. de longueur L

S.P. utilisés : ADD, SOM, DIF, COMP

description : soustraction de B à A le résultat est dans C. Pour cela on change le signe de B et on appelle le S.P. ADD

*** MUL ***

appel : call MUL (A,B,C,L, ITA)

paramètres : A,B,C : T.E.P. de longueur L
ITA : entier initialisé à zéro et prenant la valeur 1 dans le cas où il y a débordement dans le T.E.P. portant le résultat de l'opération

S.P. utilisé : néant

description : multiplication de A par B, le résultat est dans C. Comme la longueur de C est égale à L, il y aura débordement, qui entraînera l'arrêt des calculs et l'impression d'un message d'erreur ainsi que le changement de la valeur de ITA à 1, dès que la retenue sur le mot C(2) est non nulle

*** DIVE ***

appel : call DIVE (A,C,Q,R,Q1,R1,L)

paramètres : A,C,Q,R,Q1,R1 : T.E.P. de longueur L

S.P. utilisés : COMP, DIV1, MUL1, SOM1, DIF1, ADD, SOM, DIF

description : ce S.P. effectue la division entière de A par C sous la forme $A = C * Q + R$, Q étant le quotient et R le reste.
Q1 et R1 sont des T.E.P. qui servent pour le calcul local au S.P. et ne sont déclarés en paramètres que pour les avoir en longueur dynamique

Remarque : les T.E.P. A et C doivent être de même signe et A doit être supérieur en valeur absolue à C.

*** PGCD ***

appel : CALL PGCD (INA, IDA, PGC, INR, IDR, N1, N2, N3, L)

paramètres : tous les paramètres sont des TEP de longueur L

S.P. utilisés : COMP, DIVE, DIV 1, MUL 1, SOM 1, DIF 1, ADD, SOM, DIF

description : Ce SP calcule le PGCD (dans le TEP PGC) de INA et IDA, leurs valeurs réduites sont données dans le TEP INR et IDR. N1, N2, N3 sont des variables locales au SP et sont déclarées en paramètres pour rendre leur taille dynamique.

*** ADDFRA ***

appel : call ADDFRA (INA, IDA, INB, IDB, INC, IDC, N1, N2, N3, N4, N5, N6, N7, L, ITA)

paramètres : tous les paramètres sont des TEP de longueur L sauf ITA qui est un entier initialisé à zéro et dont la valeur passe à 1 dès qu'il y a débordement au cours de l'appel de ADD ou MUL. Les données $\frac{INB}{IDB}$ et $\frac{INC}{IDC}$ doivent être des fractions réduites. Le résultat $\frac{INA}{IDA}$ est donné sous forme réduite.

S.P. utilisés : PGCD, ADD, MUL, DIVE, SOM, DIF, COMP, DIV 1, MUL 1, SOM 1, DIF 1.

description : Ce SP effectue l'opération suivante $\frac{INA}{IDA} = \frac{INB}{IDB} \bullet \frac{INC}{IDC}$ en utilisant l'algorithme décrit au chapitre I. Les TEP N1, N2, N3, N4, N5, N6, N7 sont des variables locales au SP. Elles sont déclarées en paramètres pour rendre leur taille dynamique.

*** SOUFRA ***

appel : CALL SOUFRA (INA, IDA, INB, IDB, INC, IDC, N1, N2, N3, N4, N5, N6, N7, L, ITA)

paramètres : même chose que pour le SP ADDFRA

S.P. utilisés : ADDFRA, PGCD, ADD, MUL, DIVE, SOM, DIF, COMP, DIV 1, MUL 1, SOM 1, DIF 1.

description : Ce SP effectue l'opération suivante $\frac{INA}{IDA} = \frac{INB}{IBD} \ominus \frac{INC}{IDC}$ en changeant le signe de INC et en appelant le SP ADDFRA.

*** MULFRA ***

appel :CALL MULFRA (INA, IDA, INB, IDB, INC, IDC, N1, N2, N3, N4, N5, N6, N7, N8, L, ITA)

paramètres : même chose que pour le SP ADDFRA avec un TEP supplémentaire N8.

S.P. utilisés : MUL, DIVE, PGCD, COMP, ADD, SOM, DIF, DIV 1, MUL 1, SOM 1, DIF 1.

description : Ce SP effectue l'opération suivante $\frac{INA}{IDA} = \frac{INB}{IDB} \circledast \frac{INC}{IDC}$ en utilisant l'algorithme du chapitre I. Les TEP N1, N2, N3, N4, N5, N6, N7, N8 sont des variables locales au SP. Elles sont déclarées en paramètres pour rendre leur taille dynamique.

*** DIVFRA ***

appel : call DIVFRA (INA, IDA, INB, IDB, INC, IDC, N1, N2, N3, N4, N5, N6, N7, N8, L, ITA)

paramètres : même chose que pour le SP MULFRA

S.P. utilisés : MUL, DIVE, PGCD, COMP, ADD, SOM, DIF, DIV 1, MUL 1, SOM 1, DIF 1.

description : Ce SP effectue l'opération suivante $\frac{INA}{IDA} = \frac{INB}{IDB} : \frac{INC}{IDC}$ en inversant la deuxième fraction et en appelant le SP MULFRA.

III-3-2 Les sous-programmes secondaires

Ce sont des sous-programmes appelés par les S.P. primaires et donc ils n'appellent aucun autre S.P.

*** LOGBAS ***

appel : c'est une fonction qu'on appelle par LOGBAS (IBASE). Le paramètre IBASE est un entier description : la fonction LOGBAS donne le logarithme décimal de IBASE, c'est-à-dire le nombre des chiffres de (IBASE-1).

*** NOMCHI ***

appel : c'est une fonction qu'on appelle par NOMCHI (ENTI). Le paramètre ENTI est un entier

description : la fonction NOMCHI donne le nombre des chiffres de l'entier ENTI.

*** COMP ***

appel : call COMP (A,B,L,N)

paramètres : A,B : T.E.P. de longueur L
N : entier

description : comparaison en valeur absolue de A et B. Si $|A|$ est égale à $|B|$ alors N = 1, si $|A| < |B|$ alors N = 2 et si $|A| > |B|$ alors N = 3

*** SOM ***

appel : call SOM (A,B,C,L,IA,IB,ITA)

paramètres : A,B,C : T.E.P. de longueur L
IA, IB : entier représentant l'indice du mot de poids le plus fort dans A et B
ITA : entier initialisé à zéro et prenant la valeur 1 en cas de débordement dans le T.E.P. portant le résultat de l'opération

description : ce S.P. calcule dans C la somme de A et B qui sont deux T.E.P. de même signe, le calcul ne se faisant que sur les mots effectifs de A et de B c'est-à-dire jusqu'aux indices IA et IB des mots de poids le plus fort.

*** DIF ***

appel : call DIF (A,B,C,L,IA)

paramètres : A,B,C : T.E.P. de longueur L

IA : entier représentant l'indice du mot de poids le plus fort dans A

description : ce S.P. calcule dans C la différence des deux T.E.P. A et B sous l'hypothèse que $|A| > |B|$, ainsi le signe de C sera celui de A, le calcul se faisant jusqu'à l'indice IA du mot de poids le plus fort.

*** SOM1 ***

appel : call SOM1 (Q,C1,K3,L)

paramètres : Q : T.E.P. de longueur L

C1,K3 : entiers. C1 est strictement inférieur à IBASE.

description : addition à Q d'un T.E.P. dont seul le mot d'indice K3 est différent de zéro et est égal à C1, entier positif inférieur à (IBASE-1). Le résultat de cette opération est dans Q.

Remarque : ce S.P. ne peut être utilisé que par le S.P. DIVE.

*** DIF1 ***

appel : call DIF1 (Q,C1,K3,L)

paramètres : Q : T.E.P. de longueur L

C1,K3 : entiers. C1 est strictement inférieur à IBASE.

description : soustraction à Q d'un T.E.P. dont seul le mot d'indice K3 est différent de zéro et est égal à C1, entier positif inférieur à (IBASE-1). Le résultat de cette opération est dans Q.

Remarque : ce S.P. ne peut être utilisé que par le S.P. DIVE.

*** MUL1 ***

appel : call MUL1 (B,C1,Q1,K2,K3,L)

paramètres : B,Q : T.E.P. de longueur L
C1,K2,K3 : entiers. C1 est strictement inférieur à IBASE.

description : multiplication de B dont l'indice du mot de poids le plus fort est K2 par un T.E.P. dont seul le mot d'indice K3 est différent de zéro et est égal à C1. Le résultat étant porté par le T.E.P. Q1.

Remarque : ce S.P. est utilisé par le S.P. DIVE pour calculer le quotient intermédiaire Q1 et le comparer à A. Donc tout débordement dans le résultat Q1 provient d'une sur-évaluation du quotient C1 dans DIVE ; c'est pourquoi on diminue la valeur de C1 et on reprend le S.P. pour le calcul de Q1.

*** DIV1 ***

appel : call DIV1 (A,C2,Q,R,L)

paramètres : A,Q,R : T.E.P. de longueur L
C2 : entier

description : division entière de A par l'entier C2. Le quotient est dans Q, le reste dans R.

Remarque : ce S.P. ne peut être utilisé que par le S.P. DIVE.

*** DIV2 ***

appel : call DIV2 (X, C2, L)

paramètres : X : T.F.P. de longueur L
C2 : entier égal à une puissance de 10 et inférieur à IBASE.

description : division entière du T.E.P. X par C2.

III-3-3 Exemple d'appel de l'ensemble des sous-programmes

On présente ici un programme qui effectue les quatre opérations addition, soustraction, multiplication et division, de deux fractions $\frac{INB}{IDB}$ et $\frac{INC}{IDC}$. Les variables INB, IDC, INC et IDC sont des T.E.P. de longueur 7 soit un nombre de chiffres LM égal à 24. La base de décomposition est égale à 10^4. Le tableau TEL utilisé par les procédures ECR et LEC est dimensionné à 80 à cause du S.P. ECR.

```
      IMPLICIT INTEGER(A-Z)
      DIMENSION INA(7),IDA(7),INB(7),IDB(7),INC(7),IDC(7),N1(7),N2(7)
      DIMENSION N3(7),N4(7),N5(7),N6(7),N7(7),N8(7),TEL(80)
      COMMON/BLOK1/IBASE,LM,IN,KN
      ITA=0
      CALL INI(L,LR)
      CALL LEC(INB,L,TEL,LR)
      CALL LEC(IDB,L,TEL,LR)
      CALL LEC(INC,L,TEL,LR)
      CALL LEC(IDC,L,TEL,LR)
      CALL ECR(INB,L,TEL,LR)
      CALL ECR(IDB,L,TEL,LR)
      CALL ECR(INC,L,TEL,LR)
      CALL ECR(IDC,L,TEL,LR)
C***********************************************************************
C**   REDUCTION DE -INB,IDB- PAR LE CALCUL DE LEUR PGCD -N1- LEUR FORME REDUITE
C**   ETANT -INB,IDB .MEME CHOSE POUR -INC,IDC-
C***********************************************************************
      CALL PGCD(INB,IDB,N1,N2,N3,N6,N7,N8,L)
      DO 1 I=1,L
      INB(I)=N2(I)
    1 IDB(I)=N3(I)
      CALL PGCD(INC,IDC,N1,N2,N3,N6,N7,N8,L)
      DO 2 I=1,L
      INC(I)=N2(I)
    2 IDC(I)=N3(I)
      OUTPUT 'SIMPLIFICATION'
      CALL ECR(INB,L,TEL,LR)
      CALL ECR(IDB,L,TEL,LR)
      CALL ECR(INC,L,TEL,LR)
      CALL ECR(IDC,L,TEL,LR)
      OUTPUT 'ADDITION'
      CALL ADDFRA(INA,IDA,INB,IDB,INC,IDC,N1,N2,N3,N4,N5,N6,N7,L,ITA)
      CALL ECR(INA,L,TEL,LR)
      CALL ECR(IDA,L,TEL,LR)
      OUTPUT 'SOUSTRACTION'
      CALL SOUFRA(INA,IDA,INB,IDB,INC,IDC,N1,N2,N3,N4,N5,N6,N7,L,ITA)
      CALL ECR(INA,L,TEL,LR)
      CALL ECR(IDA,L,TEL,LR)
      OUTPUT 'MULTIPLICATION'
      CALL MULFRA(INA,IDA,INB,IDB,INC,IDC,N1,N2,N3,N4,N5,N6,N7,N8,L,ITA)
      CALL ECR(INA,L,TEL,LR)
      CALL ECR(IDA,L,TEL,LR)
      OUTPUT 'DIVISION'
      CALL DIVFRA(INA,IDA,INB,IDB,INC,IDC,N1,N2,N3,N4,N5,N6,N7,N8,L,ITA)
      CALL-ECR(INA,L,TEL,LR)
      CALL ECR(IDA,L,TEL,LR)
      END


      BLOCK DATA
      COMMON/BLOK1/IBASE,LM,IN,KN
      DATA IBASE,LM,IN,KN/10000,24,108,105/
      END
```

Les résultats numériques obtenus sont :

```
+ 20001000100010001
+ 999999999999
+ 999999999
- 987654321
 SIMPLIFICATION
+ 665700033336667
+ 333333333333
+ 111111111
- 109739369
 ADDITION
+ 731595339744344538160169
+ 36579789666530066877
 SOUSTRACTION
+ 731609413216344764106986
+ 36579789666630086877
 MULTIPLICATION
- 2234557990123345667889
+ 107849218216108369
 DIVISION
- 731632376781344301143123
+ 37037036799962962963
```

A N N E X E

Ensemble des sous-programmes de l'arithmétique rationnelle.

```
      SUBROUTINE INI(L,LR)
C*****************************************************************************
C** SOUS PROGRAMME INITIALISANT LES TEP A LA LONGUEUR L EN UTILISANT LM NOMBRE
C** DE CHIFFRES MAXIMUM UTILISES POUR REPRESENTER LES ENTIERS(LM EST DONNEE
C** DANS LE BLOC COMMON).
C*****************************************************************************
      IMPLICIT INTEGER(A-Z)
      COMMON/BLOK1/IBASE,LM
      PP=LOGBAS(IBASE)
      L=LM/PP+1
      IF(MOD(LM,PP).NE.0) L=L+1
      LR=(L-1)*PP+1
      IF(LR.LT.80) LR=80
      RETURN
      END
```

```
      SUBROUTINE ADDFRA(INA,IDA,INB,IDB,INC,IDC,N1,N2,N3,N4,N5,N6,N7,L,I
     1TA)
C*****************************************************************************
C** CALCUL DE LA SOMME DE DEUX FRACTIONS REDUITES -INB/IDB+INC/IDC- LE RESULTAT
C** ETANT -INA/IDA- (TOUT LES ENTIERS SONT DES TEP DE LONGUEUR -L-)
C** LES TABLEAUX -N1,N2,N3,N4,N5,N6,N7- SONT DES TEP SERVANT
C** POUR LE CALCUL INTERMEDIAIRE.
C** ITA : COMPTEUR NUL.IL EST MIS A 1 S'IL Y A UN DEBORDEMENT DANS LE RESULTAT
C** DANS LES OPERATIONS ADDITIONS OU MULTIPLICATIONS.
C** ALGORITHME : ON CALCULE LE PGCD -ID1- DE  -IDB ET IDC- ET ON A
C** INA=INB*IDC/ID1+INC*IDB/ID1 ET IDA=IDB/ID1*IDC/ID1.
C** SI ID1=1 ON ARRETE L'ALGORITHME,SINON ON CALCULE LE PGCD -ID2- DE -INA- ET
C** -ID1- ET LE RESULTAT SERA INA=INA/ID2 ET IDA=IDB/ID1*IDC/ID2.
C** REMARQUE : POUR REDUIRE LA TAILLE MEMOIRE DE CE SOUS-PROGRAMME LES TABLEAUX
C** DE DONNEES -INA,IDA- SERONT UTILISES POUR LE CALCUL INTERMEDIAIRE AVANT DE
C** PORTER LE RESULTAT FINAL.
C*****************************************************************************
      IMPLICIT INTEGER (A-Z)
      DIMENSION INA(L),IDA(L),INB(L),IDB(L),INC(L),IDC(L),N1(L),N2(L)
      DIMENSION N3(L),N4(L),N5(L),N6(L),N7(L)
      COMMON/BLOK1/IBASE,LM,IN
      ITA=0
C*****************************************************************************
C**   TEST DE NULLITE D'UN DES TERMES(INB,INC,IDB,IDC)
C*****************************************************************************
      DO 14 I=2,L
      IF(IDB(I).NE.0) GOTO 12
   14 CONTINUE
      GOTO 15
   12 DO 13,I=2,L
      IF(IDC(I).NE.0) GOTO 16
   13 CONTINUE
      GOTO 15
   16 DO 17 I=2,L
      IF(INB(I).NE.0) GOTO 18
   17 CONTINUE
      GO TO 19
   18 DO 20 I=2,L
      IF(INC(I).NE.0) GOTO 21
   20 CONTINUE
      GOTO 24
C*****************************************************************************
C** REDUCTION DE -IDB,IDC- PAR LE CALCUL DE LEUR PGCD -N3- LEUR FORME REDUITE
C** ETANT -N5,N6-.
C*****************************************************************************
 21   CALL PGCD(IDB,IDC,N3,N5,N6,INA,IDA,N2,L)
C ***************************************************************************
C** CALCUL DU NUMERATEUR -N2- DU RESULTAT PAR LA FORMULE
C** N2=INC*IDB/N3+INB*IDC/N3 .
C ***************************************************************************
      CALL MUL(INB,N6,INA,L,ITA)
```

```
      IF(ITA.NE.0) GO TO 7
      CALL MUL(N5,INC,IDA,L,ITA)
      IF(ITA.NE.0) GO TO 7
      CALL ADD(INA,IDA,N2,L,ITA)
      IF(ITA.NE.0) GO TO 7
C*****************************************************************************
C** TEST SUR LE PGCD -ID1-,S'IL EST EGAL A 1 ON CALCUL LE DENOMINATEUR
C** IDA=IDB/N3*IDC/N3 ET LE NUMERATEUR INA=N2 ,EN SUITE ON VA A L'ETIQUETTE 6
C** POUR SORTIR DU SOUS-PROGRAMME.
C*****************************************************************************
      DO 2 I=2,L-1
      IF(N3(I).NE.0) GO TO 3
2     CONTINUE
      IF(N3(L).NE.1) GO TO 3
      CALL MUL(N5,N6,IDA,L,ITA)
      IF(ITA.NE.0) GO TO 7
      DO 4 I=1,L
4     INA(I)=N2(I)
      GO TO 6
C*****************************************************************************
C** DANS LE CAS OU -N3- EST DIFFERENT DE 1 ON SIMPLIFIE LE NOUVEAU NUMERATEUR
C** -N2- ET -N3- EN CALCULANT LEUR PGCD -N4- LEURS VALEURS REDUITES SONT DANS
C**-INA ET IDA-,EN SUITE ON CALCULE LE NUMERATEUR ET LE DENOMINATEUR DU RESULTAT
C** FINAL -INA ET IDA-.
C*****************************************************************************
3     DO 11 I=1,L
11    N1(I)=N5(I)
      CALL PGCD(N2,N3,N4,INA,IDA,N5,N6,N7,L)
      ISC=1
      IF(IDC(1).LT.0) ISC=-1
      IDC(1)=IDC(1)*ISC
      CALL DIVE(IDC,N4,N5,N6,N7,N2,L)
      IDC(1)=IDC(1)*ISC
      IF(ISC.LT.0) N5(1)=-1
      CALL MUL(N5,N1,IDA,L,ITA)
      IF(ITA.NE.0) GO TO 7
      GO TO 6
C*****************************************************************************
C**   CRITERE DE SORTIE DANS LE CAS OU IL Y A UN DENOMINATEUR NUL
C*****************************************************************************
   15 WRITE(IN,100)
  100 FORMAT('ERREUR: APARITION D UN RATIONNEL DONT LE DENOMINATEUR EST
     1NUL DANS LA PROCEDURE (ADDFRA)')
      STOP
C*****************************************************************************
C**   CRITERE DE SORTIE DANS LE CAS OU IL Y A UN NUMERATEUR NUL
C*****************************************************************************
   19 DO 25 I=1,L
      INA(I)=INC(I)
   25 IDA(I)=IDC(I)
      GOTO 27
   24 DO 26 I=1,L
      INA(I)=INB(I)
   26 IDA(I)=IDB(I)
   27 CONTINUE
C*****************************************************************************
C** SORTIE DU RESULTAT CALCULE.
C*****************************************************************************
    6 IS=INA(1)*IDA(1)
      IF(IS.LT.0) GO TO 8
      INA(1)=1
      IDA(1)=1
7     CONTINUE
      RETURN
8     INA(1)=-1
      IDA(1)=1
      RETURN
      END
```

```
      SUBROUTINE SOUFRA(INA,IDA,INB,IDB,INC,IDC,N1,N2,N3,N4,N5,N6,N7,L,I
     1TA)
C***********************************************************************
C** CALCUL DE LA DIFFERENCE DE DEUX FRACTIONS -INB/IDB-INC/IDC-LE RESULTAT
C** ETANT -INA/IDA- ( TOUT LES ENTIERS SONT DES TEP DE LONGUEUR -L-).
C** LES TABLEAUX -N1,N2,N3,N4,N5,N6,N7- SONT DES TEP SERVANT POUR LE CALCUL
C** INTERMEDIAIRE
C** ITA : COMPTEUR NUL.IL EST MIS A 1 S'IL Y A DEBORDEMENT DANS LE RESULTAT DES
C** OPERATIONS D'ADDITION ET DE MULTIPLICATION
C** ALGORITHME : ON CHANGE LE SIGNE DE -INC- ET ON FAIT L'ADDITION DES DEUX
C**  FRACTIONS.
C***********************************************************************
      IMPLICIT INTEGER(A-Z)
      DIMENSION INA(L),IDA(L),INB(L),IDB(L),INC(L),IDC(L),N1(L),N2(L)
      DIMENSION N3(L),N4(L),N5(L),N6(L),N7(L)
      COMMON/BLOK1/IBASE,LM,IN
C***********************************************************************
C** CHANGEMENT DU SIGNE DE -INC-
C***********************************************************************
      INC(1)=-INC(1)
C***********************************************************************
C** APPEL DU SOUS-PROGRAMME -ADDFRA-
C***********************************************************************
      CALL ADDFRA(INA,IDA,INB,IDB,INC,IDC,N1,N2,N3,N4,N5,N6,N7,L,ITA)
      INC(1)=-INC(1)
      RETURN
      END
```

```
      SUBROUTINE MULFRA(INA,IDA,INB,IDB,INC,IDC,N1,N2,N3,N4,N5,N6,N7,N8,
     1L,ITA)
C***********************************************************************
C**   CALCUL DU PRODUIT DE DEUX FRACTIONS (INB/IDB*INC/IDC) LE RESULTAT
C**   ETANT (INA/IDA) (TOUT LES ENTIERS SONT DES TEP DE LONGUEUR .L.)
C**   LES TABLEAUX (N1,N2,N3,N4,N5,N6,N7,N8) SONT DES TEP SERVANT POUR LE
C**   CALCUL INTERMIDIAIRE.
C**   ITA : COMPTEUR NUL.IL EST MIS A 1 S'IL Y A UN DEBORDEMENT DANS LE RESULTAT
C**    '   DES ADDITIONS OU DES MULTIPLICATIONS.
C**   ALGORITHME : ON REDUIT LES DEUX FRACTIONS (INB/IDB ET INC/IDC).
C**   ON CALCULE LE PGCD -ID1- DE -IDB ET INC- AINSI QUE LE PGCD -ID2-DE -INB
C**   ET IDC-, ENFIN ON CALCULE .INA=INB/ID1*INC/ID2 ET IDA=IDB/ID2*IDC/ID1- .
C***********************************************************************
      IMPLICIT INTEGER (A-Z)
      DIMENSION INA(L),IDA(L),INB(L),IDB(L),INC(L),IDC(L),N1(L),N2(L)
      DIMENSION N3(L),N4(L),N6(L),N7(L),N8(L),N5(L)
      COMMON/BLOK1/IBASE,LM,IN
      ITA=0
C***********************************************************************
C**   TEST DE NULLITE D'UN DES TERMES(INB,INC,IDB,IDC)
C***********************************************************************
      DO 11 I=2,L
      IF(IDB(I).NE.0) GOTO 12
   11 CONTINUE
      GOTO 3
   12 DO 13,I=2,L
      IF(IDC(I).NE.0) GOTO 10
   13 CONTINUE
      GOTO 3
   10 DO 7 I=2,L
      IF(INB(I).NE.0) GOTO 8
    7 CONTINUE
      GOTO 5
    8 DO 9 I=2,L
      IF(INC(I).NE.0) GOTO 14
    9 CONTINUE
      GOTO 5
```

```
      C***********************************************************************
      C**   REDUCTION DE -INB,IDC- ET -IDB,INC-
      C***********************************************************************
      14    CALL PGCD(INB,IDC,N1,N2,N3,N6,N7,N8,L)
            CALL PGCD(IDB,INC,N1,N4,N5,N6,N7,N8,L)
      C***********************************************************************
      C**   CALCUL DU RESULTAT
      C***********************************************************************
            CALL MUL(N2,N5,INA,L,ITA)
            IF(ITA.NE.0) GOTO 4
            CALL MUL(N3,N4,IDA,L,ITA)
            IF(ITA.NE.0) GOTO 4
            IF(INA(1)*IDA(1).LT.0) GOTO 15
            INA(1)=IDA(1)=1
          4 CONTINUE
            RETURN
      15    INA(1)=-1
            IDA(1)=1
            RETURN
      C***********************************************************************
      C**   CRITERE DE SORTIE DANS LE CAS OU IL Y A UN DENOMINATEUR NUL
      C***********************************************************************
          3 WRITE(IN,100)
        100 FORMAT('ERREUR : APARITION D UN RATIONNEL DONT LE DENOMINATEUR EST
           1 NUL DANS LA PROCEDURE (MULFRA)')
            STOP
      C***********************************************************************
      C**   CRITERE DE SORTIE DANS LE CAS OU IL Y A UN NUMERATEUR NUL
      C***********************************************************************
          5 DO 6 I=2,L
            INA(I)=0
          6 IDA(I)=0
            INA(1)=1
            IDA(1)=1
            IDA(L)=1
            RETURN
            END
```

```
            SUBROUTINE DIVFRA(INA,IDA,INB,IDB,INC,IDC,N1,N2,N3,N4,N5,N6,N7,N8,
           1L,ITA)
      C***********************************************************************
      C**   CALCUL DU QUOTIENT DE DEUX FRACTIONS (INB/IDB:INC/IDC) LE RESULTAT
      C**   ETANT (INA/IDA) (TOUT LES ENTIERS SONT DES TEP DE LONGUEUR .L.)
      C**   LES TABLEAUX (N1,N2,N3,N4,N6,N7,N8) SONT DES TEP SERVANT POUR LE
      C**   CALCUL INTERMIDIAIRE.
      C**   ITA : COMPTEUR NUL.IL EST MIS A 1 S'IL Y A UN DEBORDEMENT DANS LE RESULTAT
      C**     '   DES ADDITIONS OU DES MULTIPLICATIONS.
      C** ALGORITHME : ON INVERSE LA FRACTION -INC/IDC- ET ON FAIT LA MULTIPLICATION
      C** -INB/IDB*IDC/INC.
      C***********************************************************************
            IMPLICIT INTEGER (A-Z)
            DIMENSION INA(L),IDA(L),INB(L),IDB(L),INC(L),IDC(L),N1(L),N2(L)
            DIMENSION N3(L),N4(L),N6(L),N7(L),N8(L)
            COMMON/BLOK1/IBASE,LM,IN
            CALL MULFRA(INA,IDA,INB,IDB,IDC,INC,N1,N2,N3,N4,N5,N6,N7,N8,L,ITA)
            RETURN
            END
```

```
      SUBROUTINE STO(A,B,L)
C*****************************************************************************
C** MEMORISATION D'UN ENTIER-A-DANS UN TEP B(L) DE LONGUEUR -L-.
C** ON MET DANS CHAQUE MOT DU TABLEAU UN NOMBRE DE CHIFFRES DE -A- EGAL
C** A CELUI DE (IBASE-1)
C*****************************************************************************
      IMPLICIT INTEGER (A-Z)
      DIMENSION B(L)
      COMMON/BLOK1/IBASE
      N=A
      IF(A.LT.0) N=-A
      DO 2 I=L,2,-1
      B(I)=MOD(N,IBASE)
2     N=N/IBASE
      B(1)=1
      IF(A.LT.0) B(1)=-1
      RETURN
      END
```

```
      SUBROUTINE LEC(A,L,RA,LR)
C*****************************************************************************
C** LECTURE D'UN ENTIER DONNE SUR CARTE DANS UN TEP -A(L)-.
C** L'ENTIER ECRIT SUR CARTE DOIT OBLIGATOIREMENT AVOIR UN SIGNE
C*****************************************************************************
      IMPLICIT INTEGER (A-Z)
      DIMENSION A(L),RA(LR)
      COMMON/BLOK1/IBASE,LM,IN,KN
      DATA C0/1H-/,C1/1H0/,C2/Z01000000/,C3/1H /
      DO 11 I=1,LR
11    RA(I)=C3
      DO 10 I=1,L
10    A(I)=0
      K=LM/80
      IF(MOD(LM,80).NE.0) K=K+1
      LL=0
      DO 8 I=1,K
      L1=LL*80+1
      L2=(LL+1)*80
      READ(KN,100) (RA(J),J=L1,L2)
      IF(RA(L2).EQ.C3) GO TO 4
      LL=LL+1
8     CONTINUE
100   FORMAT(80A1)
4     I=2
      J=2
      PP=LOGBAS(IBASE)
3     B=10**(PP-1)
      DO 1 K=1,PP
      A(J)=((RA(I)-C1)/C2)*B+A(J)
      B=B/10
      I=I+1
      IF(RA(I).EQ.C3)GO TO 2
1     CONTINUE
      J=J+1
      GO TO 3
2     IF(K.EQ.PP) GO TO 5
      D2=10**(PP-K)
      CALL DIV2(A,D2,L)
5     IF(J.EQ.L) GO TO 9
      KK=L-J+2
      DO 6 K=L,KK,-1
      A(K)=A(J)
6     J=J-1
      DO 7 K=KK-1,2,-1
7     A(K)=0
9     A(1)=1
      IF(RA(1).EQ.C0) A(1)=-1
      RETURN
      END
```

```
      SUBROUTINE ECR(A,L,RA,LR)
C***********************************************************************
C**IMPRESSION D'UN ENTIER REPRESENTE PAR UN TEP A(L)
C** ON TRANSFERE LES CHIFFRES DES MOTS DE A(L) DANS LE TABLEAU-RA-EN ECRIVANT UN
C** CHIFFRE PAR MOT,EN SUITE ON IMPRIME LE TABLEAU-RA-
C** A(L) : TEP DE LONGUEUR -L-.
C** IBASE : BASE DE DECOMPOSITION D'UN ENTIER DANS LE TEP A(L)
C** RA(LR): TABLEAU DANS LEQUEL ON TRANSFERE A(L) EN ECRIVANT UN CHIFFRE PAR
C**  MOT
C** K : NOMBRE DE CHIFFRES SIGNIFICATIFS DANS LE TABLEAU -A-.
C***********************************************************************
      IMPLICIT INTEGER (A-D,F-Z),LOGICAL (E)
      DIMENSION A(L),RA(LR)
      COMMON/BLOK1/IBASE,LM,IN
      DATA C0/1H+/,C1/1H-/,C3/1H /,C4/Z01000000/,C5/1H0/
C**************************************************
C** DETERMINATION DU PREMIER MOT NON NUL DU TABLEAU-A-.
C**************************************************
      DO 11 I=1,LR
11    RA(I)=C3
      DO 2 I=2,L
      IF(A(I).NE.0) GO TO 3
2     CONTINUE
C**************************************************
C** CAS OU L'ENTIER ECRIT DANS LE TABLEAU -A- EST NUL.
C**************************************************
      RA(1)=C0
      RA(2)=C3
      RA(3)=C5
      WRITE(IN,100) RA(1),RA(2),RA(3)
100   FORMAT(3A1)
      RETURN
C***********************************************************************
C** CAS OU L'ENTIER ECRIT DANS LE TABLEAU-A-EST NON NUL.
C** INITIALISATION DE-K-
C** VIDAGE DU PREMIER MOT NON NUL DU TABLEAU-A-DANS LE TABLEAU-RA-.
C***********************************************************************
3     K=LOGBAS(IBASE)*(L-I+1)
      II=I
      J=3
      E=.FALSE.
      X=A(I)
      BAS=IBASE/10
9     Q=X/BAS
      R=MOD(X,BAS)
      IF(Q.NE.0) GO TO 7
      IF(E) GO TO 6
      K=K-1
      GO TO 8
7     E=.TRUE.
6     RA(J)=Q*C4+C5
      J=J+1
8     BAS=BAS/10
      X=R
      IF(BAS.NE.0) GO TO 9
C**********************************************************************:
C** VIDAGE DU RESTE DES MOTS DU TABLEAU-A-DANS LE TABLEAU-RA-.
C**********************************************************************:
      IF(I.GE.L) GO TO 22
      II=I+1
      DO 5 I=II,L
      X=A(I)
      BAS=IBASE/10
91    Q=X/BAS
      R=MOD(X,BAS)
      RA(J)=Q*C4+C5
      J=J+1
      BAS=BAS/10
      X=R
      IF(BAS.NE.0) GO TO 91
5     CONTINUE
```

```
      C**********************************************************************
      C** IMPRESSION DU TABLEAU-RA-.
      C *****************************************************************
   22    RA(1)=C0
         IF(A(1).EQ.-1) RA(1)=C1
         RA(2)=C3
         WRITE(IN,110) (RA(I),I=1,K+2)
  110    FORMAT(132A1)
         RETURN
         END
```

```
         SUBROUTINE ADD(A,B,C,L,ITE)
      C**********************************************************************
      C** SOMME DE DEUX TEP -A,B- DE LONGUEUR -L-,LE RESULTAT EST DANS LE TEP -C- DE
      C** LONGUEUR -L-.
      C** S : SIGNE DE A*B.
      C** ITE : COMPTEUR NUL AU DEPART,EST MIS À 1 S'IL Y A DEBORDEMENT DANS LE
      C** RESULTAT -C-.
      C** IA : INDICE DU PREMIER MOT NON NUL DE -A-.
      C** IB : INDICE DU PREMIER MOT NON NUL DE -B-.
      C**********************************************************************
         IMPLICIT INTEGER (A-Z)
         DIMENSION A(L),B(L),C(L)
         COMMON/BLOK1/IBASE,LM,IN
         ITE=0
         LL=L+2
         S=A(1)*B(1)
      C*********************************************
      C** RECHERCHE DU PREMIER MOT NON NUL DU TABLEAU -A-.
      C*********************************************
         DO 7 I=2,L
         IF(A(I).NE.0) GO TO 8
    7    CONTINUE
      C*********************************************
      C** CAS OU LE TABLEAU -A- EST NUL.
      C*********************************************
         DO 9 I=1,L
    9    C(I)=B(I)
         RETURN
    8    IA=I
      C*********************************************
      C** RECHERCHE DU PREMIER MOT NON NUL DU TABLEAU -B-.
      C*********************************************
         DO 10 I=2,L
         IF(B(I).NE.0) GO TO 11
   10    CONTINUE
      C*********************************************
      C** CAS OU LE TABLEAU -B- EST NUL.
      C*********************************************
         DO 12 I=1,L
   12    C(I)=A(I)
         RETURN
      C*********************************************
      C** CAS OU LES TABLEAUX -A- ET -B- SONT NON NULS ET DE MEME SIGNE.
      C*********************************************
   11    IB=I
         IF(S.EQ.-1) GO TO 1
         CALL SOM(A,B,C,L,IA,IB,ITE)
         RETURN
```

```
C***************************************************************************
C** CAS OU LES TABLEAUX A ET B SONT NON NULS ET DE SIGNE CONTRAIRE,ON DISTINGUE
C** TROIS CAS : IAI = IBI , IAI > IBI , IAI < IBI .
C***************************************************************************
1       CALL COMP(A,B,L,N)
        GO TO (6,3,5) N
6       DO 4 I=2,L
4       C(I)=0
        C(1)=1
        RETURN
3       CALL DIF(A,B,C,L,IA)
        RETURN
5       CALL DIF(B,A,C,L,IB)
        RETURN
        END
```

```
        SUBROUTINE SOUS(A,B,C,L)
C***************************************************************************
C** SOUSTRACTION DE DEUX TEP -A,B- DE LONGUEUR -L-.LE RESULTAT EST DANS LE TEP
C** -C- DE LONGUEUR -L-.
C** ON CHANGE LE SIGNE DE -B- ET ON APPELLE LE SOUS-PROGRAME (ADD).
C***************************************************************************
        IMPLICIT INTEGER (A-Z)
        DIMENSION A(L),B(L),C(L)
        COMMON/BLOK1/IBASE,LM,IN
        B(1)=-B(1)
        CALL ADD(A,B,C,L,ITA)
        B(1)=-B(1)
        RETURN
        END
```

```
        SUBROUTINE MUL(A,B,C,L,ITA)
C***************************************************************************
C** MULTIPLICATION DE DEUX TEP -A ET B- DE LONGUEUR -L-.LE RESULTAT EST DANS LE
C** TEP -C- DE LONGUEUR -L-.
C** ITA : INDICATEUR DE DEBORDEMENT DANS LE RESULTAT -C-.
C** I1 : INDICE DU PREMIER MOT NON NUL DE -A-.
C** I2 : INDICE DU PREMIER MOT NON NUL DE -B-.
C***************************************************************************
        IMPLICIT INTEGER (A-Z)
        DIMENSION A(L),B(L),C(L)
        COMMON/BLOK1/IBASE,LM,IN
C *********************************************************************
C** INITIALISATION DE -ITA- ET DU TABLEAU -C-.
C** RECHERCHE DES PREMIERS ELEMENTS NON NULS DANS -A- ET -B-.
C *********************************************************************
        ITA=0
        C(1)=A(1)*B(1)
        DO 1 I=2,L
1       C(I)=0
        DO 9 I=2,L
        IF(A(I).NE.0) GO TO 10
9       CONTINUE
        GO TO 11
10      I1=I
        DO 12 I=2,L
        IF(B(I).NE.0) GO TO 13
12      CONTINUE
```

```
      C*************************************************
      C** CAS OU -A- OU -B- EST NUL.
      C*************************************************
      11    C(1)=1
            RETURN
      C*************************************************
      C** CAS OU -A- ET -B- SONT NON NULS.
      C** MULTIPLICATION DU TABLEAU -A- PAR LES ELEMENTS -B(I)-,POUR -I- VARIANT DE -L
      C** A -I2-.
      C*************************************************
      13    I2=I
            K=0
            DO 2 I=L,I2,-1
            R=0
            IF(B(I).EQ.0) GO TO 2
            DO 3 J=L,I1,-1
            JJ=J-K
            P=A(J)*B(I)+R
            R=P/IBASE
            IF(JJ.GE.2) GO TO 4
            IF(P.NE.0) GO TO 5
            GO TO 3
      4     P=MOD(P,IBASE)
            C(JJ)=C(JJ)+P
      3     CONTINUE
            IF(R.NE.0.AND.JJ.EQ.2) GO TO 5
            C(JJ-1)=C(JJ-1)+R
      2     K=K+1
      C ****************************************************************************
      C** NORMALISATION DU TABLEAU -C- SUIVANT LA BASE (IBASE).
      C ****************************************************************************
            DO 6 J=L,3,-1
            IF(C(J).EQ.0) GO TO 6
            R=C(J)/IBASE
            C(J)=MOD(C(J),IBASE)
            C(J-1)=C(J-1)+R
      6     CONTINUE
            R=C(2)/IBASE
            IF(R.NE.0) GO TO 5
            RETURN
      C*******************************************************************************
      C** IMPRESSION D'UN MESSAGE D'ERREUR DANS LE CAS OU IL Y A DEBORDEMENT DANS -C-.
      C*******************************************************************************
      5     WRITE(IN,7)
      7     FORMAT(' DEPASSEMENT DE CAPACITE  DANS LA MULTIPLICATION')
            ITA=1
            RETURN
            END
```

```
            SUBROUTINE DIVE(A,C,Q,R,Q1,R1,L)
      C*******************************************************************************
      C** DIVISION ENTIERE D'UN TEP -A- PAR UN TEP -C- DE LONGUEUR -L- LE RESULTAT EST
      C** DONNE PAR LES TEP -Q,R- SOUS LA FORME (A=Q*C+R).
      C** LES TABLEAUX -Q1- ET -R1- DE LONGUEUR -L- SONT DES TABLEAUX DE CALCUL
      C** INTERMEDIAIRE.
      C** ALGORITHME : ON ISOLE LES DEUX PREMIERS MOTS NON NULS -C1- ET -C2- DU POIDS
      C** LE PLUS FORT DE -A- ET -C-.ON CALCULE C1=C1/C2.ON FORME LE QUOTIENT Q=Q+C1
      C** ON CALCULE Q1=Q*C  PUIS R1=A-Q1,ET TANT QUE R1>C ON REMPLACE-A- PAR -R1-ET
      C** ON RECOMMENCE,SINON ON A R1=R ET ON A A=Q*C+R.
      C*******************************************************************************
            IMPLICIT INTEGER (A,C-Z),LOGICAL(B)
            DIMENSION A(L),C(L),Q(L),R(L),Q1(L),R1(L)
            COMMON/BLOK1/IBASE,LM,IN
```

```
C*************************************************
C***RECHERCHE DU PREMIER MOT NON NUL DU DIVISEUR -C-.
C*************************************************
      DO 3 I=2,L
      IF(C(I).NE.0) GO TO 4
3     CONTINUE
C*************************************************
C** CAS OU LE DIVISEUR -C- EST NUL.ARRET DES CALCULS.
C*************************************************
      WRITE(IN,25)
25    FORMAT('ERREUR DANS LA PROCEDURE''DIVE'',LE DENOMINATEUR EST NUL')
      STOP
C***********************************************************************
C** DANS LE CAS OU -A- ET -C- NE SONT PAS DE MEME SIGNE,LE CALCUL EST ARRETE ET
C** UN MESSAGE D'ERREUR EST IMPRIME.
C***********************************************************************
4     K2=I
      IS=A(1)*C(1)
      IF(IS.GE.0) GO TO 28
      WRITE(IN,29)
29    FORMAT('ERREUR:LA PROCEDURE''DIVE''N''ACCEPTE QUE DES ENTIERS DE
     1MEME SIGNE')
      STOP
C***********************************************************************
C** -A- ET -C- ETANT DE MEME SIGNE ON LES CONSIDERE POSITIFS,ET ON LES COMPARE
C** EN VALEUR ABSOLUE.
C***********************************************************************
28    A(1)=1
      C(1)=1
      CALL COMP(A,C,L,N)
C***********************************************************************
C** SI -A- EST PLUS PETIT QUE -C- EN VALEUR ABSOLUE,DEUX CAS : SI -A- EST NUL
C** ALORS Q=R=0,SINON ON IMPRIME UN MESSAGE D'ERREUR.
C***********************************************************************
      GO TO (40,41,42) N
42    DO 52 I=2,L
      IF(A(I).NE.0) GO TO 53
52    CONTINUE
      DO 54 I=2,L
      Q(I)=0
54    R(I)=0
      Q(1)=1
      R(1)=1
      RETURN
53    WRITE(IN,43)
43    FORMAT('ERREUR DANS LA PROCEDURE -DIV- LE DENOMINATEUR EST SUPERIE
     SUR AU NUMERATEUR')
      STOP
C*************************************************
C** SI A=C ALORS R=0 ET Q=1.
C*************************************************
40    DO 44 I=2,L
      Q(I)=0
44    R(I)=0
      Q(1)=1
      R(1)=1
      Q(L)=1
      RETURN
C***********************************************************************
C** SI -A-EST PLUS GRAND QUE-C-LE RESTE DU SOUS PROGRAMME VA FAIRE LA DIVISION
C** ON ETUDIE EN PREMIER LE CAS OU LA VALEUR DE-C-EST ECRITE DANS UN SEUL MOT
C** DANS CE CAS ON APPELLE LE SOUS-PROGRAMME(DIV1).
C***********************************************************************
41    C2=C(I)
      BOUL=.FALSE.
      IF(K2.LT.L) GO TO 30
      CALL DIV1(A,C2,Q,R,L)
      RETURN
```

```
C ***********************************************************************
C** INITIALISATION DU RESTE INTERMEDIAIRE -R1- A -A-.
C ***********************************************************************
30    DO 1 I=2,L
1     R1(I)=A(I)
      R1(1)=1
C***********************************************************************************
C** INITIALISATION DU QUOTIENT-Q-ET DU QUOTIENT INTERMEDIAIRE-Q1-A ZERO.LE SIGNE
C** DE -Q1-DEPEND DE CELUI DE-R1-.
C***********************************************************************************
      DO 2 I=2,L
2     Q(I)=0
      Q(1)=1
5     DO 11 I=2,L
11    Q1(I)=0
      Q1(1)=1
      IF(R1(1).GE.0) Q1(1)=-1
C***********************************************************************************
C** RECHERCHE DU PREMIER ELEMENT NON NUL DU DIVIDENDE A,ON LE MET DANS-C1-,-K1-
C** ETANT SON INDICE.
C***********************************************************************************
      DO 12 I=2,L
      IF(R1(I).NE.0) GO TO 13
12    CONTINUE
C***********************************************************************************
C***DIVISION DU PREMIER ELEMENT NON NUL DE -A- PAR LE PREMIER ELEMENT NON
C***NUL DE -C-.
C***********************************************************************************
13    K1=I
      C1=R1(I)
      IF(C1.GE.C2) GO TO 14
      C1=C1*IBASE
      K1=K1+1
14    C1=C1/C2
      K3=L-K2+K1
C************************************************
C** CALCUL DE Q1=C1*C PAR LA PROCEDURE(MUL1).
C************************************************
      CALL MUL1(C,C1,Q1,K2,K3,L)
C***********************************************************************************
C** ON CALCUL LE QUOTIENT-Q-EN LUI AJOUTANT OU EN LUI RETRANCHANT-C1-SUIVANT QUE
C** -R1- DE L'ETAPE PRECEDANTE EST POSITIF OU NEGATIF.
C***********************************************************************************
      IF(BOUL) GO TO 15
      CALL SOM1(Q,C1,K3,L)
      GO TO 17
15    CALL DIF1(Q,C1,K3,L)
C************************************************
C***CALCUL DU RESTE
C************************************************
17    CALL ADD(R1,Q1,R,L,ITE)
C************************************************
C** CRITERES D'ARRET.
C************************************************
      CALL COMP(R,C,L,N)
      GO TO (31,32,33) N
C***********************************************************************************
C** CAS OU -R-EST PLUS GRAND EN VALEUR ABSOLUE QUE-C-.ON RECOMMENCE LA DIVISION
C** EN REMPLACANT -A- PAR -R1-.
32    DO 23  I=1,L
23    R1(I)=R(I)
      BOUL=.FALSE.
      IF(R1(1).LT.0) BOUL=.TRUE.
      GO TO 5
```

```
C*****************************************************************************
C** CAS OU R=C EN VALEUR ABSOLUE,IL SUFFIT D'AJOUTER OU DE RETRANCHER 1 A-Q-
C** SUIVANT LE SIGNE DE-R-ET ON A A=Q*C.
C*****************************************************************************
 31     C1=1
        K3=L
        IF(R(1).GT.0) GO TO 34
        CALL DIF1(Q,C1,K3,L)
        DO 35 I=2,L
 35     R(I)=0
        R(1)=1
        RETURN
 34     CALL SOM1(Q,C1,K3,L)
        DO 36 I=2,L
 36     R(I)=0
        R(1)=1
        RETURN
C*****************************************************************************
C** CAS OU-R-EST PLUS PETIT QUE-C-.SI-R-EST POSITIF ON EST A LA FIN DE
C** L'ALGORITHME,SINON ON RETRANCHE 1 A -Q- ET -C- A -R- POUR TROUVER A=Q*C+R.
C*****************************************************************************
 33     IF(R(1).GT.0) RETURN
        C1=1
        K3=L
        CALL DIF1(Q,C1,K3,L)
        CALL ADD(C,R,R1,L,ITE)
        DO 37 I=1,L
 37     R(I)=R1(I)
        RETURN
        END
```

```
        SUBROUTINE PGCD(INA,IDA,PGC,INR,IDR,N1,N2,N3,L)
C*****************************************************************************
C** SIMPLIFICATION DE DEUX TEP -INA-,-IDA- EN -INR-,-IDR- PAR LE CALCUL DE
C** LEUR PGCD -PGC- PAR L'ALGORITHME D'EUCLIDE.TOUS LES ENTIERS ETANT DES
C** TEP DE LONGUEUR -L-.
C** LES TABLEAUX -N1,N2,N3- SERVENT POUR LE CALCUL INTERMEDIAIRE.
C*****************************************************************************
        IMPLICIT INTEGER (A-Z)
        DIMENSION INA(L),IDA(L),INR(L),PGC(L),IDR(L),N1(L),N2(L),N3(L)
        COMMON/BLOK1/IBASE,LM,IN
C*****************************************************************************
C** SI LE DENOMINATEUR DU RATIONNEL A REDUIRE EST NUL ON ARRETE LES CALCULS EN
C** IMPRIMANT UN MESSAGE D'ERREUR.
C*****************************************************************************
        DO 18 I=2,L
        IF(IDA(I).NE.0) GO TO 19
 18     CONTINUE
        WRITE(IN,100)
 100    FORMAT('ERREUR : APARITION D''UN RATIONNEL DONT LE DENOMINATEUR ES
       1T NUL  DANS LA PROCEDURE(PGCD)')
        STOP
C*****************************************************************************
C** SI LE NUMERATEUR DU RATIONNEL A REDUIRE EST NUL ON VA A LA FIN DU
C** SOUS-PROGRAMME EN PRENANT LE PGCD EGAL A 1 .
C*****************************************************************************
 19     DO 16 I=2,L
        IF(INA(I).NE.0) GO TO 17
 16     CONTINUE
        GO TO 6
```

```
      C***********************************************************************
      C** ON INITIALISE L'ALGORITHME D'EUCLIDE EN PRENANT UN RATIONNEL DONT LE
      C** NUMERATEUR EST PLUS GRAND QUE LE DENOMINATEUR .DANS LE CAS OU ILS SONT EGAU
      C** ON ARRETE LES CALCULS,LE RESULTAT ETANT EGAL A 1/1.
      C***********************************************************************
      17    CALL COMP(INA,IDA,L,N)
            GO TO (10,11,12) N
      10    DO 15 I=2,L
      15    PGC(I)=INA(I)
            PGC(1)=1
            DO 13 I=2,L-1
            INR(I)=0
      13    IDR(I)=0
            INR(L)=1
            IDR(L)=1
            INR(1)=INA(1)
            IDR(1)=IDA(1)
            RETURN
      11    DO 1 I=2,L
            INR(I)=INA(I)
      1     PGC(I)=IDA(I)
            INR(1)=1
            PGC(1)=1
            GO TO 8
      12    DO 14 I=2,L
            INR(I)=IDA(I)
      14    PGC(I)=INA(I)
            INR(1)=1
            PGC(1)=1
      C***********************************************************************
      C** ON APPLIQUE L'ALGORITHME D'EUCLIDE ,TANT QUE LE RESTE DE LA DIVISION
      C** ENTIERE N'EST PAS NUL OU EGAL A 1 ON REITERE LES CALCULS EN PRENANT POUR
      C** NOUVEAU RATIONNEL LE QUOTIENT DE LA DIVISION ET LE RESTE.
      C***********************************************************************
      8     CALL DIVE(INR,PGC,IDR,N1,N2,N3,L)
            DO 2 I=2,L
            IF(N1(I).NE.0) GO TO 3
      2     CONTINUE
      C***********************************************************************
      C** SI LE RESTE EST NUL : FIN DE L'ALGORITHME D'EUCLIDE LE PGCD DE -INA-ET-IDA-
      C** EST EGAL A -PGC- ,ON CALCULE LE RATIONNEL REDUIT -INR/IDR- DE -INA/IDA- EN
      C** DIVISANT CHAQUE TERME PAR LE PGCD -PGC-.
      C***********************************************************************
      C***********************************************************************
      C** LE SOUS-PROGRAME (DIVE) DIVISION ENTIERE N'UTILISANT QUE DES ENTIERS
      C** POSITIFS ON CHANGE LE SIGNE DES DONNEES NEGATIVES.
      C***********************************************************************
            SIN=1
            IF(INA(1).LT.0) SIN=-1
            INA(1)=INA(1)*SIN
            CALL DIVE(INA,PGC,INR,N1,N2,N3,L)
            INA(1)=INA(1)*SIN
            INR(1)=INA(1)
            SID=1
            IF(IDA(1).LT.0) SID=-1
            IDA(1)=IDA(1)*SID
            CALL DIVE(IDA,PGC,IDR,N1,N2,N3,L)
            IDA(1)=IDA(1)*SID
            IDR(1)=IDA(1)
            RETURN
      C***********************************************************************
      C** SI LE RESTE DE LA DIVISION ENTIERE EST EGAL A 1 ON ARRETE LES CALCULS EN
      C** ALLANT A L'ETIGUETTE 6.LE PGCD EST EGAL A 1, LE RATIONNEL ETANT IRREDUCTIBLE
      C***********************************************************************
```

```
3       IF(I.EQ.L.AND.N1(I).EQ.1) GO TO 6
        DO 7 I=1,L
        INR(I)=PGC(I)
7       PGC(I)=N1(I)
        GO TO 8
6       DO 9 I=2,L-1
9       PGC(I)=0
        PGC(L)=1
        PGC(1)=1
        DO 20 I=2,L
        INR(I)=INA(I)
20      IDR(I)=IDA(I)
        INR(1)=INA(1)
        IDR(1)=IDA(1)
        RETURN
        END
```

```
        INTEGER FUNCTION LOGBAS(IBASE)
C************************************************************************
C** CALCUL DE (LOGBAS) ,NOMBRE DE CHIFFRES DE (IBASE-1)
C************************************************************************
        IMPLICIT INTEGER (A-Z)
        LOGBAS=0
        IBAS=IBASE/10
        IF(IBAS.EQ.0) RETURN
2       LOGBAS=LOGBAS+1
        IBAS=IBAS/10
        IF(IBAS.NE.0) GO TO 2
        RETURN
        END
```

```
        INTEGER FUNCTION NOMCHI(ENTI)
C************************************************************************
C** CALCUL DE (NOMCHI) NOMBRE DE CHIFFRES DE L'ENTIER (ENTI).
C************************************************************************
        IMPLICIT INTEGER (A-Z)
        NOMCHI=1
        ENT=ENTI/10
        IF(ENT.EQ.0) RETURN
1       NOMCHI=NOMCHI+1
        ENT=ENT/10
        IF(ENT.NE.0) GO TO 1
        RETURN
        END
```

```
        SUBROUTINE COMP(A,B,L,N)
C************************************************************************
C** COMPARAISONS EN VALEUR ABSOLUE DE DEUX TEP -A ET B- DE LONGUEUR -L-.LE
C** RESULTAT EST DANS -N-.
C************************************************************************
```

```
      IMPLICIT INTEGER (A-Z)
      DIMENSION A(L),B(L)
C**********************************************
C** SI IAI=IBI ALORS N=1.
C**********************************************
      DO 1 I=2,L
      IF(A(I).NE.B(I)) GO TO 2
1     CONTINUE
      N=1
      RETURN
C**********************************************
C** SI IAI<IBI ALORS N=2.
C**********************************************
 2    IF(A(I).LT.B(I)) GO TO 3
      N=2
      RETURN
C*****************************************
C** SI IAI>IBI ALORS N=3.
C*****************************************
3     N=3
      RETURN
      END
```

```
      SUBROUTINE SOM(A,B,C,L,IA,IB,ITA)
C*****************************************************************
C** ADDITION DE DEUX ENTIERS DE MEME SIGNE -A,B- ECRITS DANS DES TEP DE LONGUEUR
C** -L-.LE RESULTAT EST ECRIT DANS LE TEP -C- DE LONGUEUR -L-.
C** IA : INDICE DU PREMIER MOT NON NUL DE -A-.
C** IB : INDICE DU PREMIER MOT NON NUL DE -B-.
C** ITA : COMTEUR NUL.IL EST MIS A 1 S'IL Y A  DEPASSEMENT DE CAPACITE.
C*****************************************************************
      IMPLICIT INTEGER (A-Z)
      DIMENSION A(L),B(L),C(L)
      COMMON/BLOK1/IBASE,LM,IN
C ***********************************************************
C** INITIALISATION DE -ITA-,DU SIGNE DE -C-(TOUJOURS EN C(1)),ET DE -C-.
C ***********************************************************
      ITA=0
      C(1)=A(1)
      DO 3 I=2,L
3     C(I)=0
C ***********************************************************
C** CALCUL DE L'INDICE -IC- DU PREMIER MOT NON NUL DE -C-.
C ***********************************************************
      IC=IA
      IF(IA.GT.IB) IC=IB
      R=0
C ***********************************************************
C** CALCUL DE C(I)=A(I)+B(I) EN COMMENCANT PAR LE DERNIER MOT -L-
C ***********************************************************
      DO 1 I=L,IC,-1
      SO=A(I)+B(I)+R
      R=SO/IBASE
1     C(I)=MOD(SO,IBASE)
      IF(R.NE.0.AND.IC.EQ.2) GO TO 4
      C(IC-1)=R
      RETURN
C*****************************************************************
C** IMPRESSION DE MESSAGE D'ERREUR DANS LE CAS OU IL Y A DEBORDEMENT DANS -C-.
C*****************************************************************
4     WRITE(IN,2)
2     FORMAT('DEPASSEMENT DE CAPACITE DANS L ADDITION')
      ITA=1
      RETURN
      END
```

```
      SUBROUTINE DIF(A,B,C,L,IA)
C***********************************************************************************
C** CALCUL DE LA DIFFERENCE DE DEUX TEP -A,B- DE LONGUEUR -L-.LE RESULTAT ETANT
C** DANS LE TEP -C- DE LONGUEUR -L-
C**-A- DOIT ETRE TOUJOURS PLUS GRAND QUE -B- EN VALEUR ABSOLUE,AINSI LE SIGNE DU
C** RESULTAT SERA CELUI DE -A-.
C***********************************************************************************
      IMPLICIT INTEGER (A-Z)
      DIMENSION A(L),B(L),C(L)
      COMMON/BLOK1/IBASE
C*************************************************
C** INITIALISATION DU TABLEAU -C-.
C*************************************************
      C(1)=A(1)
      DO 2 I=2,L
2     C(I)=0
C ****************************************************************************
C** CALCUL DE C=A-B MOT PAR MOT EN COMMENCANT PAR LE DERNIER MOT -L-.
C ****************************************************************************
      A3=B(L)
      DO 1 I=L,IA,-1
      A1=A(I)
      A2=A3
      IF(I.EQ.2) GO TO 1
      A3=B(I-1)
      IF(A2.EQ.0) GO TO 1
      IF(A1.GE.A2) GO TO 1
      A1=A1+IBASE
      A3=A3+1
1     C(I)=A1-A2
      RETURN
      END
```

```
      SUBROUTINE SOM1(Q,C1,K3,L)
C***********************************************************************************
C** ADDITION D'UN ENTIER POSITIF -C1- AU MOT D'INDICE -K3- DU TEP -Q-.DANS LE
C** CAS OU LA NOUVELLE VALEUR DE CE MOT EST SUPERIEURE A (IBASE-1) LA RETENUE
C** DOIT ETRE ADDITIONNEE AU MOT D'INDICE INFERIEUR.LE RESULTAT DE CETTE
C** OPERATION EST DANS -Q-.
C***********************************************************************************
      IMPLICIT INTEGER(A-Z)
      DIMENSION Q(L)
      COMMON/BLOK1/IBASE
      C4=C1
      DO 16 I=K3,3,-1
      C3=Q(I)+C4
      C4=C3/IBASE
      Q(I)=MOD(C3,IBASE)
      IF(C4.EQ.0) RETURN
16    CONTINUE
      Q(2)=Q(2)+C4
      RETURN
      END
```

```
      SUBROUTINE DIF1(Q,C1,K3,L)
C***********************************************************************************
C** SOUSTRACTION D'UN ENTIER POSITIF -C1- AU MOT D'INDICE -K3- DU TEP -Q-.LE
C** RESULTAT EST DANS -Q-.
C***********************************************************************************
```

```
      IMPLICIT INTEGER (A-Z)
      DIMENSION Q(L)
      COMMON/BLOK1/IBASE
      C4=C1
      DO 18 I=K3,3,-1
      IF(Q(I).GE.C4) GO TO 19
      Q(I)=Q(I)+IBASE-C4
      C4=1
18    CONTINUE
      Q(2)=Q(2)-1
      RETURN
19    Q(I)=Q(I)-C4
      RETURN
      END
```

```
      SUBROUTINE MUL1(B,C1,Q1,K2,K3,L)
C***********************************************************************
C** MULTIPLICATION D'UN TEP -B- DE LONGUEUR -L- DONT LE PREMIER MOT NON NUL EST
C** A L'INDICE K2- PAR UN TEP DONT SEUL LE MOT -C1- D'INDICE -K3- EST NON NUL
C** LE RESULTAT EST DANS -Q1-.
C***********************************************************************
      IMPLICIT INTEGER (A-Z)
      DIMENSION B(L),Q1(L)
      COMMON/BLOK1/IBASE
      L1=L+2
      K4=K3+2
      K5=L1-K2
C**************************************************
C** INITIALISATION DU RESULTAT-Q1-.
C**************************************************
26    DO 25 I=2,L
25    Q1(I)=0
C**************************************************
C** CALCUL DU RESULTAT-Q1-MOT PAR MOT
C**************************************************
      DO 20 I=2,K5
      IF(B(L1-I).EQ.0) GO TO 20
      C3=B(L1-I)*C1
      C4=C3/IBASE
      C3=MOD(C3,IBASE)
      Q1(K4-I)=Q1(K4-I)+C3
      IF(C4.EQ.0) GO TO 20
      IF((K4-I-1).LT.2) GO TO 21
      Q1(K4-I-1)=Q1(K4-I-1)+C4
20    CONTINUE
C**************************************************
C** NORMALISATION DE-Q-SUIVANT LA BASE(IBASE).
C**************************************************
      DO 24 I=2,K5
      C4=Q1(K4-I)/IBASE
      IF(C4.EQ.0) GO TO 24
      IF((K4-I-1).LT.2) GO TO 21
      Q1(K4-I)=MOD(Q1(K4-I),IBASE)
      Q1(K4-I-1)=Q1(K4-I-1)+C4
24    CONTINUE
      RETURN
21    C1=C1/(C4+1)
      GO TO 26
      END
```

```
      SUBROUTINE DIV1(A,C2,Q,R,L)
C**********************************************************************************
C** DIVISION ENTIERE D'UN TEP -A- DE LONGUEUR -L- PAR UN ENTIER -C2-.LE RESULTAT
C** EST DONNE SOUS LA FORME (A=C2*Q+R).
C**********************************************************************************
      IMPLICIT INTEGER (A-Z)
      DIMENSION A(L),Q(L),R(L)
      COMMON/BLOK1/IBASE
      DO 1 I=2,L
    1 Q(I)=0
      Q(1)=1
      C1=0
      I=2
    9 C1=C1*IBASE+A(I)
      IF(C1.EQ.0) GO TO 6
      IF(C1.GE.C2) GO TO 7
      I=I+1
      IF(I.GT.L) GO TO 8
      C1=C1*IBASE+A(I)
    7 Q(I)=C1/C2
      C1=MOD(C1,C2)
    6 I=I+1
      IF(I.LE.L) GO TO 9
    8 DO 10 I=2,L-1
   10 R(I)=0
      R(1)=1
      R(L)=C1
      RETURN
      END
```

```
      SUBROUTINE DIV2(X,C2,L)
C**********************************************************************************
C** DIVISION D'UN TEP X DE LONGUEUR L PAR UN NOMBRE C2 EGALE A UNE PUISSANCE
C** DE 10 ET INFERIEUR A IBASE.
C********************************************************************************
      IMPLICIT INTEGER(A-Z)
      DIMENSION X(L)
      COMMON/BLOK1/IBASE
      I=2
      C1=0
    1 C1=C1*IBASE+X(I)
      IF(C1.EQ.0) GO TO 2
      IF(C1.GE.C2) GO TO 3
      X(I)=0
      I=I+1
      IF(I.GT.L) GO TO 8
      C1=C1*IBASE+X(I)
    3 X(I)=C1/C2
      C1=MOD(C1,C2)
    2 I=I+1
      IF(I.LE.L) GO TO 1
    6 CONTINUE
      RETURN
      END
```

CHAPITRE III

ARITHMETIQUE REELLE ETENDUE

I - INTRODUCTION

On introduit dans ce chapitre la deuxième arithmétique étudiée : l'arithmétique réelle étendue [8, 9] Elle diffère de l'arithmétique rationnelle puisque les calculs s'effectuent de façon approchée sur des données approchées mais en utilisant des mantisses d'une longueur étendue ce qui améliore la précision des résultats. Dans ce but, on a simulé les quatre opérations de base (addition, soustraction, multiplication, division) en définissant les nombres dans des tableaux dont la longueur est fixée par l'utilisateur.

II - ARITHMÉTIQUE RÉELLE ÉTENDUE

II-1 DEFINITION

Tout nombre réel x représenté en virgule flottante normalisée s'écrit :

$$x = \pm m \times b^E$$

avec m une mantisse illimitée

b la base de représentation

E l'exposant entier

cette valeur est représentée en machine par :

$$X \in \mathbb{F} \qquad X = \pm M \times b^E$$

avec

$\mathbb{F}$ Sous-ensemble des nombres rationnels qu'il est possible de représenter avec une mantisse de longueur finie P.

M la mantisse limitée à P chiffres dans la base b.

Dans les ordinateurs courants l'arithmétique à virgule flottante normalisée est souvent disponible en simple précision (P = 6) ou en double précision (P = 15).

La représentation que l'on propose va permettre de disposer de nombres réels dont la mantisse est représentée avec un nombre de chiffres P aussi grand que l'on veut et fixé par l'utilisateur.

II-2 CODAGE EXTERNE DES NOMBRES REELS

Tout nombre réel sera représenté sous la forme normalisée par :

$$X = \pm \,.\, a_1\, a_2 \ldots a_m\, \alpha \pm e_1 \ldots e_n,\ a_1 \neq 0$$

m est le nombre de chiffres de la mantisse.

L'exposant $\pm\, e_1 \ldots e_n$ doit rester inférieur à la capacité maximale de la machine en variable entière.

α indique le mode de la fin de la mantisse.

On a défini quatre modes différents :

E si la mantisse est exacte

A si la mantisse est arrondie

T si la mantisse est tronquée

X si la mantisse se termine d'une façon inconnue.

II-3 CODAGE INTERNE DES NOMBRES REELS

a) *Définition* : La représentation du nombre X du paragraphe précédent est donnée en base 10. Dans une base quelconque B on aura :

$$X = \pm \left(\frac{A_i}{B^i} + \frac{A_{i+1}}{B^{i+1}} + \ldots + \frac{A_{i+j}}{B^{i+j}} + \ldots\right) B^E \quad \forall\, i \in \mathbb{Z}$$

ce qui s'écrit en représentation normalisée avec k chiffres après la virgule :

$$X = \pm \,.\, A_1 \ldots A_k\, \alpha \pm e_1 \ldots e_n$$

avec $0 \leq A_i < B \quad i = 1, \ldots, k$

et $\quad A_1 \neq 0.$

Dans le cas où B est une puissance de 10, $B = 10^p$, on trouve les relations suivantes :

$$A_1 = a_1 \ldots a_p$$

$$A_2 = a_{p+1} \ldots a_{2p}$$

$$\vdots$$

$A_i = a_{(i-1)p-1} \ldots a_{ip}$ et ceci jusqu'au dernier chiffre a_m.

Exemple 1

Soit le nombre réel X en base 10 :

$$X = + .23985602000\ \alpha + 41839$$

en base $B = 10^4$ s'écrit :

$$X = + \, . \, A_1 \, A_2 \, A_3 \, \alpha + E$$

avec $A_1 = (2398)_{10}$

$A_2 = (5602)_{10}$

$A_3 = (000)_{10}$

$E = (41839)_{10}$

b) Représentation interne

Pour chaque nombre réel, on réserve un tableau X de L mots entiers, L étant calculée par l'algorithme LØNG (paragraphe III-1) en utilisant MM chiffres de la mantisse en base 10 (MM = m), LP chiffres de B en base 10 et LS mots réservés dans le tableau (signe, mode de fin de la mantisse, etc ...).

Dans les exemples numériques que nous donnerons, on prendra toujours $L = 6$, $B = 10^4$.

Les L mots du tableau sont utilisés ainsi :

X(LS) : mot contenant l'exposant signé. Il est donc possible de ne représenter que les réels dont l'exposant ne dépasse pas la capacité de l'ordinateur en variable entière.

X(1) : mot contenant le mode de fin de la mantisse.

X(2) : mot contenant l'indice du dernier mot non nul du tableau.

X(3) : mot contenant le nombre de chiffres non nuls en base 10 de la mantisse.

X(4) : mot contenant le nombre de chiffres significatifs de la mantisse en base 10. Les zéros significatifs sont pris en compte.

X(5) : mot contenant le nombre de chiffres exacts de la mantisse en base 10, les zéros compris.

X(LS+1) : mot contenant le signe de la mantisse. Ce mot contiendra +1 si le nombre est positif sinon -1.

Les mots X(LS+2) jusqu'à X(L) contiennent les chiffres de la mantisse en base 10^{LP}.

Exemple 2 :

Reprenons le nombre réel de l'exemple 1 :

$$M = + \, . \, A_1 \, A_2 \, A_3 \, \alpha + E$$

prenons LP = 4 (donc $B = 10^4$)

LS = 6

MM = 20

pour présenter M on réserve un tableau de longueur L donnée par la formule suivante (algorithme LØNG) :

$$L = [MM/_{LP}] + LS + K \text{ avec } K = \begin{cases} 1 \text{ si MODULO (MM, LP)} = 0 \\ 2 \text{ si MODULO (MM, LP)} \neq 0 \end{cases}$$

l'opérateur [./.] représentant la division entière et MODULO (a, b) désigne le reste de la division de a par b avec a, b $\in \mathbb{N}$. Ce qui donne pour cet exemple, L = 12.

Donc le réel M = + . 23985602000 α + 41839 est représenté en mémoire de l'ordinateur par un tableau M(12) défini par :

α	9	8	11	11	+ 41839	+ 1	2398	5602	0	0	0

Dans le cas où l'on veut rajouter des mots réservés au tableau il suffit de prendre LS plus grand, la taille du tableau sera réajustée par l'algorithme LØNG et ce sont les mots d'indice (LS-1) jusqu'à 6 qui seront utilisés pour les nouveaux mots.

II-4 LES OPERATIONS ARITHMETIQUES DE BASE

Soient deux nombres réels X et Y définis par :

$$X = S(X) \;.\; a_1 \;\ldots\; a_m \;\alpha\; S(e) \; e_1 \;\ldots\; e_\ell$$

$$Y = S(Y) \;.\; b_1 \;\ldots\; b_n \;\beta\; S(c) \; c_1 \;\ldots\; c_k$$

S(X) et S(Y) représentent les signes de X et Y, α et β les modes de terminaison de leurs mantisses, S(e) et S(c) les signes de leurs exposants respectifs.

Pour la suite on utilise la notation $X = (M_x, E_x)$ M_x étant la mantisse dans la représentation normalisée et E_x l'exposant.

II-4-1 Addition

a) Définition : On suppose que X et Y sont de même signe et que Y > X. Les autres cas étant identiques.

La somme S de X et Y est donnée par :

$$S = (M_s, E_s) = .M_s \ \gamma \ E_s$$

tel que :

$$E_s = E_y - E_x$$
$$M_s = M_y + (M_x \times 10^{-E_s}).$$

Le résultat est toujours donné sous la forme normalisée.

Le mode de terminaison γ de la mantisse de S est défini par le tableau suivant :

α \ β	E	A	T	X
E	E	A	T	X
A	A	A	X	X
T	T	X	T	X
X	X	X	X	X

Tableau 1

Le calcul de M_s se fait avec le nombre de chiffres le plus élevé des deux mantisses M_y et $M_x * 10^{-E_s}$ l'autre mantisse étant complétée par des zéros.

b) Algorithme

Soient X(L) et Y(L) les tableaux représentant en machine les réels X et Y, et S(L) le résultat de leur addition. L'algorithme de l'addition (SOMR) est le suivant :

[SOMR 1] - S(LS) ← Y(LS) - X(LS)

- S(LS+1) ← Y(LS+1)

[SOMR 2] - décalage de la mantisse de X de S(LS) positions vers la droite en utilisant l'algorithme DECA (paragraphe III-3)

[SOMR 3] - RES ← 0

- pour I variant de L jusqu'à LS+2 par pas de -1 faire :
  . S(I) ← MODULO [Y(I) + X(I) + RES, B]
  . RES ← [(Y(I) + X(I) + RES)/B]

[SOMR 4] - si RES = 0 aller à [SOMR 5] sinon faire :

. décalage de la mantisse de S d'une position

. $S(LS+2) \leftarrow S(LS+2) + (RES \times B/_{10})$

[SOMR 5] - Calcul de S(1) par l'algorithme (FINM). Cet algorithme représente l'opérateur défini par le tableau 1 en utilisant le mode de terminaison de X et Y, (paragraphe III-3).

- Calcul de S(2) et S(3) par l'algorithme (CHNN) (paragraphe III-3)
- $S(4) \leftarrow$ Maximum (X(4), Y(4))
- $S(5) \leftarrow$ Minimum (X(5), Y(5)).
- fin.

II-4-2 Soustraction

Sous les mêmes hypothèses qu'au paragraphe II-4-1 l'algorithme de la soustraction (DIFR) est identique à celui de l'addition sauf pour le calcul de la mantisse du résultat M_s qui sera donné par :

$$M_s = M_y - (M_x \times 10^{-E_s})$$

II-4-3 Multiplication

a) Définition

Le produit P de Y par X est défini par :

$$P = (M_p, E_p)$$

tel que $E_p = E_x + E_y$

$$M_p = M_x \times M_y.$$

Le résultat est donné sous la forme normalisée. Le mode de terminaison γ de la mantisse est défini par les mêmes règles que pour l'addition, sauf dans le cas ou le résultat dépasse la dimension L du tableau P(L) mémorisant le résultat, dans ce cas on réapplique la règle du tableau 1 en utilisant la valeur de γ déjà trouvée à la place de α.

b) Algorithme

Sachant que X, Y et P sont représentés en machine dans les tableaux X(L), Y(L) et P(L), que RX est un tableau de dimension 2L qui sert pour le calcul intermédiaire, l'algorithme (MULR) définissant le produit est le suivant :

[MULR 1] - P(LS) ← Y(LS) + X(LS)

- P(LS+1) ← Y(LS+1) × X(LS+1)
- calcul de P(1) par l'algorithme (FINM) en utilisant le mode de terminaison de X et Y.
- calcul de P(2) et P(3) par l'algorithme (CHNN).

[MULR 2] - mise à zéro du tableau RX

- IND ← 0 ; RET ← 0 ; LL ← 2L - IND ; J = L
- pour I variant de L à LS+2 par pas de 1

faire :

. A ← RX(LL) + (X(I) XY(J)) + RET

. RX(LL) ← MODULO (A, B)

. RET ← [A/B]

. LL ← LL - 1

[MULR 4] - J ← J - 1 ; IND ← IND + 1

- si J > LS + 1 aller à [MULR 3]

[MULR 5] - normalisation du résultat intermédiaire RX par rapport à la base B par l'algorithme (NØRM) (paragraphe III-3)

[MULR 6] - pour I variant de LS+1 jusqu'à L par pas de -1 faire

. P(I) ← RX(I)

[MULR 7] - P(4) ← Maximum (X(4) + Y(4), MM)

- P(5) ← Maximum (X(5) + Y(5), MM)
- si RX(L+1) = 0 aller à [MULR 8] sinon recalculer P(1) par l'algorithme (FINM) en utilisant P(1) déjà calculé et l'arrondi.

[MULR 8] fin.

II-4-4 Division

a) Définition

Le quotient de X par Y est défini par :

$$Q = (M_q, E_q)$$

tel que

$$E_q = E_x - E_y$$

$$M_q = M_x \div M_y$$

Le résultat est donné sous la forme normalisée. Le mode de terminaison de la mantisse est défini par les mêmes règles que pour la multiplication.

b) *Algorithme*

On se donne les réels X et Y représentés dans les tableaux X(L) et Y(L), Q le quotient de X par Y sera représenté dans Q(L). Pour décrire l'algorithme de la division on a besoin de deux tableaux RY(L) et RZ(9,L)
Le principe de l'algorithme consiste à comparer le dividende X à Y, 2Y,...,9Y et de mettre dans le quotient le multiplicateur i de Y tel que i Y < X < (i+1)Y. On remplace X par X - (Y × Q) et on recommence l'opération de comparaison jusqu'à ce que l'on obtienne le nombre de chiffres nécessaires dans la mantisse de Q, ce nombre de chiffres M pourra être fixé par l'utilisateur.

L'algorithme est donc défini par (DIVR) :

[DIVR 1] - pour J variant de 1 à 9 par pas de 1 :
. RZ (J) = J * Y

[DIVR 2] - RY ← X

[DIVR 3] - Q(LS) ← X(LS) - Y(LS)
- Q(LS+1) ← X(LS+1) × Y(LS+1)
- Q(5), Q(4) ← Maximum (X(4) + Y(4), M)

[DIVR 4] - pour J variant de 1 à X(4) par pas de 1 faire :
. recherche de I tel que RZ(I) ≤ RY ≤ RZ (I+1)
. mettre I à la $J^{\text{ème}}$ position de la mantisse de Q.
. RY ← X - (Y × Q)

[DIVR 5] - calcul de Q(2) et Q(3) par l'algorithme (CHNN)
- calcul de Q(1) par l'algorithme (FINM) en utilisant X(1) et Y(1).

[DIVR 6] - si RY = 0 aller en [DIVR 7] sino faire :
. l'arrondi sur le dernier chiffre significatif de Q.
- le calcul de Q(1) par l'algorithme (FINM) en utilisant la valeur de Q(1) déjà calculée et l'arrondi.

[DIVR 7] - fin.

III - CODAGE DE L'ARITHMÉTIQUE RÉELLE ÉTENDUE

L'ensemble des algorithmes réalisant les quatre opérations arithmétiques ainsi que la lecture et l'écriture des données ont été écrits en code FØRTRAN.

III-1 INITIALISATION DU CODE

Dans le but d'avoir un code général, les données permettant de définir la représentation interne des nombres réels seront introduites comme paramètres du code par l'instruction FØRTRAN (BLØK DATA) eux-mêmes communiqués aux sous-programmes par l'instruction CØMMØN, et qui devront être initialisés par l'utilisateur.

L'ensemble de ces données est :

MM : nombre de chiffres des mantisses en base 10

LS : nombre de mots réservés dans les tableaux de la représentation interne

LP : nombre de chiffres de la base choisie pour la représentation interne en base 10, ($B = 10^{LP}$).

NN : le plus grand entier représenté en machine en variable entière.

NI, NJ : numéro logique de l'imprimante et du lecteur de carte.

Ces paramètres seront donc déclarés par l'instruction :

```
      BLØCK DATA
   CØMMØN / BLØCK / MM, LS, LP, NN, NI, NJ
   DATA / MM, LS, LP, NN, NI, NJ / 240, 6, 4, 2147483647, 108, 105 /
   END
```

les valeurs données à NN, NI, NJ sont celles qui ont été utilisées sur l'ordinateur IRIS 80 de CII du centre de calcul de l'Université de Lille I.

Enfin, pour tous les programmes d'appel, en tête des instructions exécutables, on doit faire appel au sous-programme d'initialisation LØNG qui permet de calculer les dimensions des tableaux nécessaires à l'ensemble des algorithmes. Ce calcul se fait suivant les formules :

$$L = [MM / LP] + LS + K \text{ avec } K = \begin{cases} 1 \text{ si Modulo } (MM, LP) = 1 \\ 2 \text{ sinon} \end{cases}$$

$$LI = [((MM+15)/80)+K]*80 \text{ avec } K = \begin{cases} 0 \text{ si Modulo } (MM + 15, 80) = 0 \\ 1 \text{ sinon} \end{cases}$$

$$LJ = 2 * L$$

III-2 ENTREE ET SORTIE DES DONNEES

III-2-2 Entrée des données

Tous les nombres réels sont introduits sur des cartes de données sous la forme suivante :

$$\pm \, . \, a_1 \, \ldots\ldots \, a_m \, \alpha \pm e_1 \, \ldots\ldots \, e_n$$

α étant le mode de terminaison de la mantisse comme on l'a défini au paragraphe II-2.

Les a_i sont les chiffres significatifs en base 10, les zéros compris, composant la mantisse.

La représentation peut porter sur plusieurs cartes et doit se terminer par un blanc.

Soit le nombre réel X. On le mémorise dans le tableau X de dimension L par l'algorithme (LECR) suivant :

On se donne un tableau intermédiaire RX (LI).

[LECR 1] - lire les données caractère par caractère dans le tableau RX, en s'arrêtant au premier blanc rencontré, soit IF l'indice du dernier mot de RX.

[LECR 2] - transformer la mantisse en base 10 en une mantisse en base B et l'écrire dans les mots X(LS+2) jusqu'à X(L) en complétant par des zéros : Soit IM l'indice du mot du tableau RX représentant le mode de terminaison de la mantisse.

[LECR 3] - mettre dans X(LS) l'exposant écrit caractère par caractère dans les mots RX (IM+1) jusqu'à RX (IF).

- mettre dans X (LS+1) les valeurs +1 ou -1 suivant que RX(1) est le signe + ou -.
- mettre dans X(1) l'une des valeurs 1, 2, 3 ou 4 suivant que RX(IM) est le caractère E, A, T ou X.
- calcul de X(2) et X(3) par l'algorithme (CHNN).

Remarques

1) On considère que pour les données en entrée le nombre de chiffres significatifs de la mantisse est égal au nombre de chiffres exacts (X(4) = X(5)). Il est donc nécessaire de donner tous les chiffres significatifs des données en entrée même s'il s'agit de zéros.

2) Les nombres entiers ne doivent pas être lus sur les cartes de données par l'algorithme (LECR), mais fournis directement à l'algorithme (VFL). Cet algorithme les transforme en un réel dans un tableau X, et initialise X(4) et X(5) par le nombre de chiffres maximum que peut avoir la mantisse d'un réel écrit dans le tableau X.

3) Le nombre zéro est initialisé dans un tableau X(L) par l'algorithme VFL de la façon suivante.

X(1) = 1

X(2) = 7

X(3) = 0

X(4) = X(5) = (L - LS - 1) × LP

X(LS) = 0

X(LS + 1) = +1

X(LS + 2) = ... = X(L) = 0.

III-2-2 Sortie des données

L'impression d'un résultat se fait par l'algorithme (ECRR) en utilisant un tableau intermédiaire RX (LI) dans lequel on transforme les chiffres en caractères avant de les imprimer. Cet algorithme est donc l'inverse de celui de la lecture (LECR).

III-3 MODE D'EMPLOI DES SOUS-PROGRAMMES

Pour l'ensemble des sous-programmes, et sauf spécifications contraires, lors de l'appel d'un SP la signification des différentes variables est la suivante : X, Y et Z désignent des réels codés dans des tableaux suivant la méthode du paragraphe II-3, Y et Z étant les opérandes X le résultat ; L est la dimension de ces tableaux. MM, LS, LP, B, NN désignent les mêmes variables que celles du paragraphe III-1.
L'ensemble des sous-programmes comprend un SP d'initialisation, des SP d'écriture et de lecture, des SP utilisateurs et des SP utilitaires.

III-3-1 Le sous-programme d'initialisation

Ce SP doit être appelé avant toute utilisation des autres sous-programmes.

*** LØNG ***

Appel : CALL LØNG (L, LI, LJ)

Paramètres : LI, LJ dimension des tableaux paramétrés de certains SP.

SP utilisé : néant

Description : ce SP calcule L, LI, LJ suivant l'algorithme LØNG du paragraphe III-1.

III-3-2 Les sous-programmes de lecture et d'écriture

Trois SP permettent la communication des données entre les programmes et l'utilisateur.

*** LECR ***

Appel : CALL LECR (X, L, RX, LI)

Paramètres : RX est le tableau de dimension LI dans lequel on tranfère les données sur cartes.

SP utilisé : CHNN

Description : ce SP lit dans X les données sur carte suivant l'algorithme LECR du paragraphe III-2-1. La conversion des caractères de RX en chiffres se fait en utilisant le code EBCDIC.

*** ECRR ***

Appel : CALL ECRR (X, L, RX, LI)

Paramètres : RX est un tableau de dimension LI dans lequel on transfère le réel X avant de l'imprimer.

SP utilisé : néant

Description : ce SP écrit les données dans X suivant l'algorithme ECRR du paragraphe III-2-2. Ce SP utilise pour la conversion des chiffres en caractères le code EBCDIC.

*** VFL ***

Appel : CALL VFL (X, N, L)

Paramètre : N est un entier relatif

SP utilisé : néant

Description : ce SP transforme l'entier N en un réel codé dans le tableau X suivant l'algorithme VFL du paragraphe III-2-1 remarque 2 et 3.

Remarque

Dans le cas où on utilise un ordinateur travaillant avec un code différent du code EBCDIC, il est nécessaire de modifier les SP LECR et ECRR.

III-3-3 Les sous-programmes utilisateurs

On présente ici l'ensemble des SP concernant les quatre opérations arithmétiques.

*** ADDR ***

Appel : CALL ADDR (X, Y, Z, L)

SP utilisés : SØMR, DIFR, CØMR, VFL

Description : Ce SP appelle les SP de la somme (SØMR) ou de la différence (DIFR) suivant que Y et Z sont de même signe ou de signe différent.

*** SØUR ***

Appel : CALL SØUR (X, Y, Z, L)

SP utilisé : ADDR

Description : ce SP change le signe de Z et appelle le SP d'addition ADDR.

*** SØMR ***

Appel : CALL SØMR (X, Y, Z, L)

SP utilisés : DECA, CHNN, FINM

Description : ce SP calcule dans X la somme de Y et Z suivant l'algorithme du paragraphe II-4-1.

*** DIFR ***

Appel : CALL DIFR (X, Y, Z, L)

SP utilisés : DECA, NØRM, CHNN, FINM

Description : ce SP calcule dans X la différence (Y-Z) suivant l'algorithme du paragraphe II-4-2.

*** MULR ***

Appel : CALL MULR (X, Y, Z, L, RX, LJ)

Paramètres : RX est un tableau de dimension LJ.

SP utilisés : VFL, FINM, NØRM, ARØN

Description : ce SP effectue dans X la multiplication de Y par Z suivant l'algorithme du paragraphe II-4-3.

*** DIVR ***

Appel : CALL DIVR (X, Y, Z, L, M, RZ, RY, LJ)

Paramètres : RZ un tableau de dimensions (9, LJ)
RY un tableau de dimension LJ
M un entier qui permet à l'utilisateur de fixer le nombre de chiffres de la mantisse du quotient.

SP utilisés : ARØN, MULR 1, VFL, FINM, CHNN

Description : ce SP effectue l'opération X = Y/Z suivant l'algorithme du paragraphe II-4-4.

III-3-4 Les sous-programmes utilitaires

L'ensemble des SP utilitaires sont appelés par les SP utilisateurs mais sont aussi à la disposition de l'utilisateur si certaines précautions d'utilisation sont prises.

*** DECA ***

Appel : CALL DECA (X, Z, L, DEC)

Paramètre : DEC est un paramètre entier donné en entrée.
Z est un tableau en entrée. X un tableau en sortie

SP utilisé : DIVR 1

Description : ce SP décale d'un nombre (DEC) de positions la mantisse de Z et place le résultat dans X. L'algorithme consiste à diviser par 10^{DEC} le réel écrit dans Z. Dans le cas où la dimension de X est insuffisante, on arrondit le dernier chiffre de la mantisse ainsi obtenue.

*** NØRM ***

Appel : CALL NØRM (X, L, DEC)

Paramètre : DEC est une variable entière de sortie du SP.
X est un tableau en entrée et en sortie.

SP utilisé : MULR 1

Description : ce SP normalise la mantisse de X en opérant des multiplications par 10 successives. Le nombre de multiplications faites est donné par DEC.

*** ARØN ***

Appel : CALL ARØN (X, L, IX, RET)

Paramètres : IX et RET sont des entiers en entrée.
X est un tableau en entrée et en sortie.

SP utilisés : DECA, CHNN.

Description : ce SP additionne la valeur RET au mot d'indice IX de X. Dans le cas où RET est égal à 1 l'algorithme fait l'arrondi sur le dernier chiffre du mot d'indice IX.

*** CHNN ***

Appel : CALL CHNN (X, L)

SP utilisé : néant

Description : ce SP calcule dans X(2) l'indice du dernier mot de X non nul et dans X(3) le nombre de chiffres non nuls en base 10 de la mantisse de X.

*** FINM ***

Appel : CALL FINM (X1, Y1, Z1)

Paramètres : X1, Y1 et Z1 sont des entiers variant de 1 à 4.
X1 est le résultat.
Y1 et Z1 sont les données.

SP utilisé : néant

Description : ce SP calcule X1 en utilisant le tableau 1 du paragraphe II-4-1 en remplaçant α et β par Y1 et Z1.

*** MULR 1 ***

Appel : CALL MULR 1 (X, B, L, FIX)

Paramètres : B et FIX deux entiers en entrée. X est un tableau en entrée sortie.

SP utilisé : ce SP permet de multiplier le réel écrit dans X par l'entier B. Il ne peut-être utilisé que si la multiplication ne fait pas apparaître de retenue sur le dernier mot d'indice FIX.

*** DIVR 1 ***

Appel : CALL DIVR 1 (X, C2, R, L)

Paramètres : C2 et R sont deux entiers en entrée.
X est un tableau en entrée, sortie.

SP utilisé : néant

Description : ce SP divise le réel écrit dans X par l'entier C2, R est le reste de cette division.

*** CØMR ***

Appel : CALL CØMR (X, Y, L, N)

Paramètres : N est une variable de sortie entière comprise entre 1 et 3.
X et Y sont des tableaux en entrée.

SP utilisé : néant

Description : ce SP compare en valeur absolue les deux réels écrits dans X et Y et donne le résultat :

$N = 1$ si $|X| > |Y|$

$N = 2$ si $|X| = |Y|$

$N = 3$ si $|X| < |Y|$

III-4 EXEMPLE D'APPEL DE L'ENSEMBLE DE SOUS-PROGRAMMES

On présente un programme qui éffectue les quatre opérations addition, soustraction, multiplication et division de deux nombres réels Y et Z représentés par des tableaux de dimension L = 22, soit un nombre de chiffres de la mantisse égal à 60.

```
      IMPLICIT INTEGER(A-Z)
      DIMENSION X(25),Y(25),Z(25),RZ(9,25),RY(50),RX(80)
      COMMON/BLOK/M,LS,LP,NN,NI,NJ
      M=10
      CALL LONG(L,LI,LJ)
      CALL LECR(Y,L,RX,LI)
      CALL ECRR(Y,L,RX,LI)
      CALL LECR(Z,L,RX,LI)
      CALL ECRR(Z,L,RX,LI)
      OUTPUT 'SOMME'
      CALL ADDR(X,Y,Z,L)
      CALL ECRR(X,L,RX,LI)
      OUTPUT 'DIFFERENCE'
      CALL SOUR(X,Y,Z,L)
      CALL ECRR(X,L,RX,LI)
      OUTPUT 'DIVISION'
      CALL DIVR(X,Y,Z,L,M,RZ,RY,LJ)
      CALL ECRR(X,L,RX,LI)
      CALL MULR(X,Y,Z,L,RY,LJ)
      OUTPUT 'MULTIPLICATION'
      CALL ECRR(X,L,RX,LI)
      END
```

```
      BLOCK DATA
      COMMON/ALGN/MN,LS,LP,NN,NI,NJ
      DATA MN,LS,LP,NN,NI,NJ/60,6,4,2147483647,108,105/
      END
```

Les résultats numériques obtenus sont :

```
+ ,2451E+2147450647
+ ,76532E+123457892
 SOMME
+ ,24510000000000000000000000000000000000000000000000000E+2147450647
 DIFFERENCE
+ ,24510000000000000000000000000000000000000000000000000E+2147450647
 DIVISION
+ ,3121020730A+2023992755
 ERREUR DANS S.P MUL:DEPASSEMENT DE CAPACITE SUR L EXPOSANT
*STOP* 0
**********************************************************************
+ ,1234567891234560E+18
+ ,1234567891234560A+10
 SOMME
+ ,1234567903580238912345 6A+18
 DIFFERENCE
+ ,12345678788888810876544A+18
 DIVISION
+ ,100000000000000A+9
 MULTIPLICATION
+ ,15241578780673483700609383936A+27
*STOP* 0
**********************************************************************
+ ,99999999999999999999E+3
+ ,99999999999999999999E+3
 SOMME
+ ,199999999999999999998E+4
 DIFFERENCE
+ ,00000000000000000000E+0
 DIVISION
+ ,10000000000000000000A+1
 MULTIPLICATION
+ ,9999999999999999999980000000000000000000001E+6
*STOP* 0
**********************************************************************
+ ,123456789E+6
+ ,12345E+3
 SOMME
+ ,123580239E+6
 DIFFERENCE
+ ,123333339E+6
 DIVISION
+ ,1000054994A+4
 MULTIPLICATION
+ ,1524074060205E+8
*STOP* 0
**********************************************************************
```

A N N E X E

ENSEMBLE DES SOUS-PROGRAMMES DE L'ARITHMETIQUE REELLE

```
      SUBROUTINE LONG(L,LI,LJ)
C** LONGUEUR TOTALE EN MOT MEMOIRE DU TABLEAU CONTENANT LE REEL.
      IMPLICIT INTEGER (A-Z)
      COMMON/BLOK/MM,LS,LP
      K=2
      IF(MOD(MM,LP).EQ.0) K=1
      L=MM/LP+K+LS
      LI=MM+15
      IF (MOD(LI,80).EQ.0) GO TO 1
      LI=(LI/80+1)*80
    1 LJ=2*L
      RETURN
      END
```

```
      SUBROUTINE LECR(X,L,RX,LI)
C** LECTURE D'UN REEL SUR CARTE
      IMPLICIT INTEGER(A-Z)
      DIMENSION X(L),RX(LI),EXP(4)
      COMMON/BLOK/MM,LS,LP,NN,NI,NJ
      DATA C0/1H-/,C1/1H0/,C2/Z01000000/,C3/1H /
      DATA EXP/1HE,1HA,1HT,1HX/
      K=MM/80
      IF(MOD(MM,80).NE.0) K=K+1
      LL=0
      DO 8 I=1,K
      L1=LL*80+1
      L2=(LL+1)*80
      READ(NJ,100) (RX(J),J=L1,L2)
      IF(RX(L2).EQ.C3) GO TO 4
      LL=LL+1
    8 CONTINUE
  100 FORMAT(80A1)
    4 I=3
      J=LS+2
    3 B=10**(LP-1)
      X(J)=0
      DO 1 K=1,LP
      X(J)=((RX(I)-C1)/C2)*B+X(J)
      B=B/10
      I=I+1
      IF(RX(I).LT.C1)GO TO 2
    1 CONTINUE
      J=J+1
      GO TO 3
    2 IF(I.GT.(MM+2)) GO TO 15
      X(LS+1)=1
      IF(RX(1).EQ.C0) X(LS+1)=-1
      SIG=1
      IF(RX(I+1).EQ.C0) SIG=-1
      K=I+2
      X(LS)=0
    9 X(LS)=X(LS)*10+(RX(K)-C1)/C2
      JJ=(RX(K)-C1)/C2
      K=K+1
      IF(RX(K).EQ.C3) GO TO 10
      GO TO 9
   10 X(LS)=SIG*X(LS)
      DO 5 K=1,4
      IF(RX(I).EQ.EXP(K)) GO TO 6
    5 CONTINUE
```

```
 6      X(1)=K
        DO 7 K=J+1,L
 7      X(K)=0
        X(4)=I-3
        X(5)=X(4)
        CALL CHNN(X,L)
        RETURN
 15     WRITE(NI,200)
 200    FORMAT('ERREUR : LA MENTISSE DEPASSE LE NOMBRE DE CHIFFRES MM')
        RETURN
        END
```

```
        SUBROUTINE ECRR(X,L,RX,LI)
C** ECRITURE D'UN REEL
        IMPLICIT INTEGER(A-Z)
        DIMENSION X(L),RX(LI),EXP(4)
        COMMON/BLOK/MM,LS,LP,NN,NI
        DATA C0/1H+/,C1/1H-/,C2/1H,/,C3/1H /,C4/Z01000000/,C5/1H0/
        DATA EXP/1HE,1HA,1HT,1HX/
        RX(1)=C0
        IF(X(LS+1).EQ.-1) RX(1)=C1
        RX(2)=C3
        RX(3)=C2
        I=4
        J=LS+2
 3      Z=X(J)
        B=10**(LP-1)
        DO 1 K=1,LP
        Y=Z/B
        RX(I)=Y*C4+C5
        IF((I-3).GE.X(4)) GO TO 2
        Z=MOD(Z,B)
        B=B/10
        I=I+1
 1      CONTINUE
        J=J+1
        GO TO 3
 2      DO 4 K=1,4
        IF(X(1).EQ.K) GO TO 5
 4      CONTINUE
 5      RX(I+1)=EXP(K)
        RX(I+2)=C0
        IF(X(LS).LT.0) RX(I+2)=C1
        Y=X(LS)
        IF(X(LS).LT.0) Y=-Y
        K=I+3
        IF(Y.NE.0) GO TO 7
        RX(K)=Y*C4+C5
        GO TO 8
 7      I=INT(LOG10(DFLOAT(Y))+1.E-6)
        I=10**I
 6      RX(K)=Y/I*C4+C5
        K=K+1
        Y=MOD(Y,I)
        I=I/10
        IF(I.NE.0) GO TO 6
        K=K-1
 8      WRITE(NI,100) (RX(I),I=1,K)
 100    FORMAT(132A1)
        RETURN
        END
```

```
      SUBROUTINE ADDR(X,Y,Z,L)
      IMPLICIT INTEGER(A-Z)
      DIMENSION X(L),Y(L),Z(L)
      COMMON/BLOK/MM,LS,LP,NN
      IF(Y(3).NE.0) GO TO 1
      DO 2 I=1,L
    2 X(I)=Z(I)
      RETURN
    1 IF(Z(3).NE.0) GO TO 3
      DO 4 I=1,L
    4 X(I)=Y(I)
      RETURN
    3 IF(Y(LS+1).NE.Z(LS+1)) GO TO 5
      IF(Y(LS).LT.Z(LS)) GO TO 6
      CALL SOMR(X,Y,Z,L)
      RETURN
    6 CALL SOMR(X,Z,Y,L)
      RETURN
    5 CALL COMR(Y,Z,L,N)
      GO TO (8,9,10) N
    8 CALL DIFR(X,Y,Z,L)
      RETURN
    9 CALL VFL(X,0,L)
      CALL FINM(X(1),Y(1),Z(1))
      X(4)=MAX0(Y(4),Z(4))
      X(5)=MIN0(Y(5),Z(5))
      RETURN
   10 CALL DIFR(X,Z,Y,L)
      RETURN
      END
```

```
      SUBROUTINE SOUR(X,Y,Z,L)
      IMPLICIT INTEGER(A-Z)
      DIMENSION X(L),Y(L),Z(L)
      COMMON/BLOK/MM,LS,LP,NN
      Z(LS+1)=-Z(LS+1)
      CALL ADDR(X,Y,Z,L)
      Z(LS+1)=-Z(LS+1)
      RETURN
      END
```

```
      SUBROUTINE MULR(X,Y,Z,L,RX,LJ)
      IMPLICIT INTEGER(A-B,D-Z)
      DIMENSION X(L),Y(L),Z(L),RX(LJ)
      COMMON/BLOK/MM,LS,LP,NN
      DO 15 I=1,L
   15 X(I)=0
      DO 1I=LS+2,Y(2)
      IF(Y(I).NE.0) GO TO 2
    1 CONTINUE
      GO TO 5
    2 DO 3 I=LS+2,Z(2)
      IF(Z(I).NE.0) GO TO 4
    3 CONTINUE
    5 CALL VFL(X,0,L)
      CALL FINM(X(1),Y(1),Z(1))
      X(4)=MAX0(Y(4),Z(4))
      X(5)=MIN0(Y(5),Z(5))
      RETURN
```

```
    4   CALL FINM(X(1),Y(1),Z(1))
        C=DABS(DFLOAT(Y(LS))+DFLOAT(Z(LS)))
        IF(C.GT.DFLOAT(NN)) GO TO 14
        IF(C.LT.DFLOAT(1/NN)) GO TO 14
        X(LS)=Y(LS)+Z(LS)
        X(LS+1)=Y(LS+1)*Z(LS+1)
        LY=Y(4)/LP
        IF(MOD(Y(4),LP).NE.0) LY=LY+1
        LZ=Z(4)/LP
        IF(MOD(Z(4),LP).NE.0) LZ=LZ+1
        LRX=LY+LZ+LS+1
        LY=LY+LS+1
        LZ=LZ+LS+1
        DO 7 I=1,LRX
    7   RX(I)=0
        B=10**LP
        IND=0
    9   RET=0
        LL=LRX-IND
        DO 8 I=LY,LS+2,-1
        A=RX(LL)+Y(I)*Z(LZ)+RET
        RX(LL)=MOD(A,B)
        RET=A/B
    8   LL=LL-1
        RX(LL)=RX(LL)+RET
        IND=IND+1
        LZ=LZ-1
        IF(LZ.GT.LS+1) GO TO 9
        RX(LL)=RET
        DO 6 J=LRX,LS+3,-1
        IF(RX(J).EQ.0) GO TO 6
        R=RX(J)/B
        RX(J)=MOD(RX(J),B)
        RX(J-1)=RX(J-1)+R
    6   CONTINUE
        CALL NORM(RX,LRX,DEC)
        X(LS)=X(LS)-DEC
        K=(L-LS-1)*LP
        X(5)=Y(5)+Z(5)-DEC
        IF(X(5).GT.K) X(5)=K
        X(4)=Y(4)+Z(4)-DEC
        IF(X(4).GT.K) X(4)=K
        KK=X(4)/LP+LS+1
        IF(MOD(X(4),LP).NE.0) KK=KK+1
        DO 10 I=LS+2,KK
   10   X(I)=RX(I)
        CALL CHNN(X,L)
        IF(RX(KK+1).NE.0) GO TO 11
        DO 12 I=KK+2,LRX
        IF(RX(I).NE.0) GO TO 13
   12   CONTINUE
        RETURN
   11   B=B/10
        ARO=RX(KK+1)/B
        IF(ARO.LT.5) GO TO 13
        CALL ARON(X,L,L,1)
   13   CALL FINM(X(1),X(1),2)
        RETURN
   14   OUTPUT'ERREUR DANS S.P MUL:DEPASSEMENT DE CAPACITE SUR L EXPOSANT'
        STOP
        END
```

```
      SUBROUTINE DIVR(X,Y,Z,L,M,RZ,RY,LJ)
      IMPLICIT INTEGER (A-B,D-Z)
      DIMENSION X(L),Y(L),Z(L),RZ(9,L),RY(LJ)
      COMMON/BLOK/MM,LS,LP,NN,NI
      IF(Y(3).NE.0) GO TO 1
      IF(Z(3).NE.0) GO TO 2
      WRITE(NI,100)
100   FORMAT(' ERREUR PROCEDURE (DIV):DIVISION INDETERMINEE 0/0')
      STOP
2     CALL VFL(X,0,L)
      CALL FINM(X(1),Y(1),Z(1))
      X(4)=X(5)=MAX0(Y(4),Z(4),M)
      RETURN
1     IF(Z(3).NE.0) GO TO 3
      WRITE(NI,110)
110   FORMAT('ERREUR PROCEDURE (DIV):DIVISION PAR ZERO')
      STOP
3     DO 4 I=1,9
      DO 4 J=1,L
4     RZ(I,J)=0
      DO 8 J=1,LJ
8     RY(J)=0
      IND=2
      DEB=Z(2)
      DO 5 J=LS+2,DEB
      RZ(1,IND)=Z(J)
5     IND=IND+1
      FIZ=IND-1
      B=10**LP
      DO 6 I=2,9
      RET=0
      DO 7 J=FIZ,1,-1
      A=RZ(1,J)+RZ(I-1,J)+RET
      RZ(I,J)=MOD(A,B)
7     RET=A/B
6     CONTINUE
      DEB=Y(2)
      IND=2
      DO 9 J=LS+2,DEB
      RY(IND)=Y(J)
9     IND=IND+1
      FIY=IND-1
      DO 25 I=1,L
25    X(I)=0
      CALL FINM(X(1),Y(1),Z(1))
      C=DABS(DFLOAT(Y(LS))-DFLOAT(Z(LS)))
      IF(C.GT.DFLOAT(NN)) GO TO 30
      IF(C.LT.DFLOAT(1/NN)) GO TO 14
      X(LS)=Y(LS)-Z(LS)
      X(LS+1)=Y(LS+1)*Z(LS+1)
      X(5)=X(4)=MAX0(Y(4),Z(4),M)
      DO 10 J=2,FIZ
      IF(RZ(1,J).GT.RY(J)) GO TO 11
      IF(RZ(1,J).LT.RY(J)) GO TO 12
10    CONTINUE
12    X(LS)=X(LS)+1
      GO TO 13
11    CALL MULR1(RY,10,L,1)
13    DEY=1
      FIX=(X(4)+1)/LP+LS+1
      IF(MOD(X(4),LP).NE.0) FIX=FIX+1
      IF(FIX.LE.L) GO TO 19
      X(4)=X(5)=(L-LS-1)*LP
      FIX=L
19    INDX=X(4)+1
      IX=LS+2
14    B=10**(LP-1)
```

```
      DO 15 K=1,LP
      INDX=INDX-1
      DO 16 IZ=1,9
      IRY=DEY
      DO 17 JZ=1,FIZ
      IF(RZ(IZ,JZ).GT.RY(IRY)) GO TO 18
      IF(RZ(IZ,JZ).LT.RY(IRY)) GO TO 16
17    IRY=IRY+1
      RX=IZ
      GO TO 20
18    RX=IZ-1
      GO TO 20
16    CONTINUE
      RX=9
20    IF(INDX.EQ.0) GO TO 21
      X(IX)=X(IX)+RX*B
      IF(RX.EQ.0) GO TO 23
      RET=0
      IZ=RX
      IRY=MAX0(FIY,FIZ)
      BB=10**LP
22    DIF=RY(IRY)-RZ(IZ,IRY)-RET
      RET=0
      IF(DIF.LT.0) RET=1
      RY(IRY)=DIF+BB*RET
      IRY=IRY-1
      IF(IRY.GE.1) GO TO 22
23    CALL MULR1(RY,10,L,1)
15    B=B/10
      IX=IX+1
      GO TO 14
21    CALL CHNN(X,L)
      CALL FINM(X(1),X(1),2)
      IF(RX.GE.5) GO TO 24
      RETURN
24    IND=MOD(X(4),LP)
      IF(IND.EQ.0) GO TO 26
      RES=10**(LP-IND)
      GO TO 27
26    RES=1
      IX=IX-1
27    CALL ARON(X,L,IX,RES)
      RETURN
30    OUTPUT'ERREUR DANS S.P DIV:DEPASSEMENT DE CAPACITE SUR L EXPOSANT'
      STOP
      END
```

```
      SUBROUTINE SOMR(X,Y,Z,L)
      IMPLICIT INTEGER(B-Z)
      DIMENSION X(L),Y(L),Z(L)
      COMMON/BLOK/MM,LS,LP,NN
      DO 1 I=1,L
1     X(I)=0
      A=DABS(DFLOAT(Y(LS))-DFLOAT(Z(LS)))
      IF(A.GT.DFLOAT(NN)) GO TO 14
      IF(A.LT.DFLOAT(1/NN)) GO TO 14
      DEC=Y(LS)-Z(LS)
      IF(DEC.NE.0) GO TO 2
      DO 3 K=1,L
3     X(K)=Z(K)
      GO TO 4
2     CALL DECA(X,Z,L,DEC)
4     X(4)=MAX0(X(4),Y(4))
```

```
      X(5)=MINO(X(5),Y(5))
      CALL FINM(X(1),Y(1),Z(1))
      RES=0
      P=10**LP
      DEB=MAX0(X(2),Y(2))
      DO 5 K=DEB,LS+2,-1
      S=Y(K)+X(K)+RES
      X(K)=MOD(S,P)
    5 RES=S/P
      IF(RES.EQ.0) GO TO 6
      CALL DECA(X,X,L,1)
      X(LS+2)=X(LS+2)+RES*P/10
    6 CALL CHNN(X,L)
      RETURN
   14 OUTPUT'ERREUR DANS S.P SOM:DEPASSEMENT DE CAPACITE SUR L EXPOSANT'
      STOP
      END
```

```
      SUBROUTINE DIFR(X,Y,Z,L)
      IMPLICIT INTEGER (A-B,D-Z)
      DIMENSION X(L),Y(L),Z(L)
      COMMON/BLOK/MM,LS,LP,NN
      DO 1 I=1,L
    1 X(I)=0
      C=DABS(DFLOAT(Y(LS))-DFLOAT(Z(LS)))
      IF(C.GT.DFLOAT(NN)) GO TO 14
      IF(C.LT.DFLOAT(1/NN)) GO TO 14
      DEC=Y(LS)-Z(LS)
      IF(DEC.NE.0) GO TO 2
      DO 3 K=1,L
    3 X(K)=Z(K)
      GO TO 4
    2 CALL DECA(X,Z,L,DEC)
    4 X(4)=MAX0(X(4),Y(4))
      X(5)=MINO(X(5),Y(5))
      CALL FINM(X(1),Y(1),Z(1))
      P=10**LP
      DEB=MAX0(X(2),Y(2))
      A=X(DEB)
      DO 5 K=DEB,LS+2,-1
      AY=Y(K)
      AX=A
      A=X(K-1)
      IF(AY.GE.AX) GO TO 5
      AY=AY+P
      A=A+1
    5 X(K)=AY-AX
      X(LS+1)=Y(LS+1)
      CALL NORM(X,L,D)
      X(4)=X(4)-D
      X(5)=X(5)-D
      X(LS)=X(LS)-D
      CALL CHNN(X,L)
      RETURN
   14 OUTPUT'ERREUR DANS S.P DIF:DEPASSEMENT DE CAPACITE SUR L EXPOSANT'
      STOP
      END
```

```
      SUBROUTINE COMR(X,Y,L,N)
      IMPLICIT INTEGER (A-Z)
      DIMENSION X(L),Y(L)
      COMMON/BLOK/MM,LS,LP
      IF(X(LS).NE.Y(LS)) GO TO 1
      DO 2 I=LS+2,L
      IF(X(I).NE.Y(I)) GO TO 3
    2 CONTINUE
      N=2
      RETURN
    3 IF(X(I).LT.Y(I)) GO TO 4
      N=1
      RETURN
    4 N=3
      RETURN
    1 IF(X(LS).LT.Y(LS)) GO TO 5
      N=1
      RETURN
    5 N=3
      RETURN
      END
```

```
      SUBROUTINE NORM(X,L,DEC)
      IMPLICIT INTEGER(A-Z)
      DIMENSION X(L)
      COMMON/BLOK/MM,LS,LP
      D2=0
      D1=LS+1
      REPEAT 1,FOR I=(LS+2,L)
      D1=D1+1
      IF(X(I).NE.0) GO TO 8
    1 CONTINUE
      IF(D1.LE.LS+2) GO TO 2
    8 D3=D1-LS-2
      REPEAT 6,FOR I=(D1,L)
    6 X(I-D3)=X(I)
      REPEAT 7,FOR J=(I-D3+1,L)
    7 X(J)=0
    2 B=10**(LP-1)
      S=X(LS+2)
    3 M=S/B
      IF(M.NE.0) GO TO 4
      D2=D2+1
      S=MOD(S,B)
      B=B/10
      GO TO 3
    4 B=10**D2
      CALL MULR1(X,B,L,LS+2)
      DEC=D2+D3*LP
      RETURN
      END
```

```
      SUBROUTINE DECA(X,Z,L,DEC)
C***DECALAGE DE LA MANTISSE DE Z DE DEC POSITION LE RESULTAT EST DANS X
      IMPLICIT INTEGER (A-B,D-Z)
      DIMENSION X(L),Z(L)
      COMMON/BLOK/MM,LS,LP,NN,NI
```

```
      K=(L-LS-1)*LP
      INDZ=Z(4)
      IF(K.LT.INDZ+DEC) INDZ=K-DEC
      IF(Z(3)+DEC.LE.K) GO TO 1
      IF(Z(1).LE.2) X(1)=2
      IF(Z(1).GE.3) X(1)=4
      GO TO 3
1     X(1)=Z(1)
3     IF(DEC.LT.K) GO TO 5
      R=Z(LS+2)
      DO 7 J=LS+2,L
7     X(J)=0
      IF(DEC.GT.K) GO TO 10
      GO TO 6
5     DEB=INDZ/LP+LS+2
      IF(MOD(INDZ,LP).NE.0) DEB=DEB+1
      DE=DEC/LP
      RES=MOD(DEC,LP)
      REPEAT2 ,FOR J=(DEB,LS+2,-1)
2     X(J+DE)=Z(J)
      REPEAT 4 ,FOR JJ=(LS+1+DE,LS+2,-1)
4     X(JJ)=0
      IF(RES.EQ.0) GO TO 6
      B=10**RES
      CALL DIVR1(X,B,R,L)
      B=10**(RES-1)
      GO TO 8
6     B=10**(LP-1)
8     R=R/B
      IF(R.LE.5) GO TO 10
      RET=1
      B=10**LP
      REPEAT 9,FOR J=(L,LS+2,-1)
      A=X(J)+RET
      X(J)=MOD(A,B)
      RET=A/B
      IF(RET.EQ.0) GO TO 10
9     CONTINUE
10    C=DABS(DFLOAT(Z(LS))+DFLOAT(DEC))
      IF(C.GT.DFLOAT(NN)) GO TO 14
      IF(C.LT.DFLOAT(1/NN)) GO TO 14
      X(LS)=Z(LS)+DEC
      X(LS+1)=Z(LS+1)
      X(4)=INDZ+DEC
      X(5)=Z(5)+DEC
      IF(X(5).GT.K) X(5)=K
      K=X(4)/LP+LS+1
      IF(MOD(X(4),LP).NE.0) K=K+1
12    IF(X(K).NE.0) GOTO 11
      K=K-1
      IF(K.GT.(LS+1))GOTO 12
      X(2)=7
      X(3)=0
      RETURN
11    X(2)=K
      KK=(K-LS-1)*LP
      B=10
      Y=X(K)
15    IF(MOD(Y,B).NE.0) GO TO 13
      KK=KK-1
      Y=Y/B
      GO TO 15
13    X(3)=KK
      RETURN
14    OUTPUT'ERREUR DANS S.P DEC:DEPASSEMENT DE CAPACITE SUR L EXPOSANT'
      STOP
      END
```

```
      SUBROUTINE MULR1(X,B,L,FIX)
      IMPLICIT INTEGER(A-Z)
      DIMENSION X(L)
      COMMON/BLOK/MM,LS,LP
      RES=0
      P=10**LP
      DO 1 K=L,FIX,-1
      M=X(K)
      M=M*B
      X(K)=MOD(M,P)+RES
    1 RES=M/P
      RETURN
      END
```

```
      SUBROUTINE DIVR1(X,C2,R,L)
      IMPLICIT INTEGER(A-Z)
      DIMENSION X(L)
      COMMON/BLOK/MM,LS,LP
      B=10**LP
      C1=0
      I=LS+2
    1 C1=C1*B+X(I)
      IF(C1.EQ.0) GO TO 2
      IF(C1.GE.C2) GO TO 3
      X(I)=0
      I=I+1
      IF(I.GT.L) GO TO 8
      C1=C1*B+X(I)
    3 X(I)=C1/C2
      C1=MOD(C1,C2)
    2 I=I+1
      IF(I.LE.L) GO TO 1
    8 R=C1
      RETURN
      END
```

```
      SUBROUTINE ARON(X,L,IX,RET)
C** RECHERCHE DU CHIFFRE ARO CONSICUTIF AU DERNIER CHIFFRE SIGNIFICATIF DE X
      IMPLICIT INTEGER(A-Z)
      DIMENSION X(L)
      COMMON/BLOK/MM,LS,LP,NN
      RES=RET
      B=10**LP
      DO 8  I=IX,LS+2,-1
      S=X(I)+RES
      X(I)=MOD(S,B)
      RES=S/B
      IF(RES.EQ.0) GO TO 1
    8 CONTINUE
      CALL DECA(X,X,L,1)
      X(LS+2)=X(LS+2)+RES*B/10
      RETURN
    1 CALL CHNN(X,L)
      RETURN
      END
```

```
      SUBROUTINE FINM(X1,Y1,Z1)
      IMPLICIT INTEGER(A-Z)
      IF(Y1.EQ.1) GO TO 1
      IF(Y1.NE.4) GO TO 2
      X1=4
      GO TO 5
    2 IF(Z1.LE.Y1) GO TO 3
      X1=4
      GO TO 5
    3 IF(Z1.EQ.Y1) GO TO 1
      IF(Z1.NE.1) GO TO 4
      X1=Y1
      GO TO 5
    4 X1=4
      GO TO 5
    1 X1=Z1
    5 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE VFL(X,N,L)
      IMPLICIT INTEGER (A-Z)
      DIMENSION X(L)
      COMMON/BLOK/MM,LS,LP
      IF(N.NE.0) GO TO 5
      DO 2 I=LS,L
    2 X(I)=0
      X(1)=1
      X(2)=7
      X(3)=0
      X(4)=X(5)=(L-LS-1)*LP
      X(LS+1)=1
      RETURN
    5 X(LS+1)=1
      IF(N.LT.0) X(LS+1)=-1
      N1=N
      IF(N.LT.0) N1=-N1
      I=INT(LOG10(DFLOAT(N1))+1.E-6)
      X(LS)=I+1
      X(4)=X(5)=(L-LS-1)*LP
      X(1)=1
      I=10**I
      J=LS+2
      B=N1
    4 A=10**(LP-1)
      X(J)=0
      DO 1 K=1,LP
      X(J)=B/I*A+X(J)
      B=MOD(B,I)
      I=I/10
      A=A/10
      IF(I.EQ.0) GO TO 3
    1 CONTINUE
      J=J+1
      GO TO 4
    3 DO 6 K=J+1,L
    6 X(K)=0
      CALL CHNN(X,L)
      RETURN
      END
```

```
      SUBROUTINE CHNN(X,L)
C** METTRE L'INDICE DU DERNIER MOT NON NUL DU REEL DANS X(3)
      IMPLICIT INTEGER (A-Z)
      DIMENSION X(L)
      COMMON/BLOK/MM,LS,LP
      K=X(4)/LP+LS+1
      IF(MOD(X(4),LP).NE.0) K=K+1
    2 IF(X(K).NE.0) GOTO1
      K=K-1
      IF(K.GT.(LS+1))GOTO 2
      X(2)=7
      X(3)=0
      X(LS)=0
      X(LS+1)=1
      RETURN
    1 X(2)=K
      KK=(K-LS-1)*LP
      B=10
      Y=X(K)
    4 IF(MOD(Y,B).NE.0) GO TO 3
      KK=KK-1
      Y=Y/B
      GO TO 4
    3 X(3)=KK
      RETURN
      END
```

CHAPITRE IV

APPLICATIONS ET RESULTATS NUMERIQUES

I - INTRODUCTION

Ce chapitre est consacré aux différentes expériences numériques utilisant les arithmétiques réelles et rationnelles étendues. Ces expériences ont été menées sur l'ordinateur IRIS 80 du centre de calcul de l'Université de Lille I.

Dans la première partie on donne un exemple où seule l'arithmétique rationnelle étendue fournit le résultat cherché : c'est le calcul des termes d'une suite de Choquet.

Dans la deuxième partie on donne une comparaison entre l'arithmétique de l'ordinateur, l'arithmétique réelle étendue et l'arithmétique rationnelle étendue, utilisées pour accélérer la convergence par l'ε-algorithme scalaire de la suite définie par $S_n = (\frac{3}{4})^n n^2$.

La troisième partie est consacrée à une comparaison identique mais portant sur le calcul de la fonction e^x par un développement en série entière pour x égal à -10, -15, -20, -25, -30.

II - ETUDE DES SUITES DE CHOQUET

Les suites de Choquet (A_n, R_n) sont définies par la récurrence :

A_1 est un entier positif

si A_n est impair $R_n = 1$

et $2A_{n+1} = 1 + 3A_n$

si A_n est pair $R_n = 0$

et $2A_{n+1} = 3A_n$

Les termes de telles suites sont des entiers ; il est donc possible de les calculer, pour n assez grand, que si on utilise l'arithmétique rationnelle étendue.

Pour cela on a écrit un programme de calcul utilisant les sous-programmes suivants du chapitre II :

INI, STO, DIVE, MUL, ADD, ECR

Les variables utilisées sont des T.E.P. de dimension L = 51, ce qui nous permet de travailler avec des entiers de 200 chiffres.

Le programme de calcul est le suivant :

```
C***  CALCUL DES SUITES DE CHOQUET
C***  A(1) EST UN ENTIER POSITIF
C***  SI A(N) EST IMPAIR 2*A(N+1)=1+3*A(N)
C***  SI A(N) EST   PAIR 2*A(N+1)=  3*A(N)
C***  SI A(N) EST IMPAIR R(N)=1
C***  SI A(N) EST   PAIR R(N)=0
C***  LE PROGRAMME CALCULE A(N) ET R(N)
      IMPLICIT INTEGER(A-Z)
      DIMENSION K(51),I(51),J(51),A(51),R(51),B(51),N1(51),N2(51)
      DIMENSION RA(240)
      COMMON/BLOK1/IBASE,LR,IN,KN,NM
      ITA=0
      N=1
      CALL INI(L,LR)
      CALL STO(1,K,L)
      CALL STO(2,I,L)
      CALL STO(3,J,L)
      CALL STO(4,A,L)
2     CALL DIVE(A,I,B,R,N1,N2,L)
      IF(N.EQ.0) GO TO 3
      IF(MOD(N,100).NE.0) GO TO 4
3     WRITE(IN,100) N
      CALL ECR(A,L,RA,LR)
      WRITE(IN,101) N
      CALL ECR(R,L,RA,LR)
      WRITE(IN,102)
4     CALL MUL(A,J,N1,L,ITA)
      IF(ITA.NE.0) GO TO 10
      DO 1 M=1,L
1     B(M)=N1(M)
      IF(R(L).EQ.1) CALL ADD(N1,K,B,L,ITA)
      IF(ITA.NE.0) GO TO 10
      N=N+1
      IF(N.GT.1000) GO TO 10
      CALL DIVE(B,I,A,R,N1,N2,L)
      GOTO 2
10    CONTINUE
100   FORMAT('A(',I5,')')
101   FORMAT('R(',I5,')')
102   FORMAT(20H********************)
      END
```

```
      BLOCK DATA
      COMMON/BLOK/NM,NI
      DATA NM,NI/2147483647,108/
      END
```

Les résultats numériques obtenus pour les 1000 premiers termes sont :

```
A(  100)
+ 1145396133350398418
R(  100)
+ 0
********************
A(  200)
+ 4656736007192203978674588454679479 01
R(  200)
+ 1
********************
A(  300)
+ 189324607455469898237985945827799500835400266464412982
R(  300)
+ 0
********************
A(  400)
+ 76972116655723738683179384916113486681124154037483165053900753590045
334
R(  400)
+ 0
********************
A(  500)
+ 31293674384928996674738129036090334603780165411090823218949459535716
765579129660761732666
R(  500)
+ 0
********************
A(  600)
+ 12722874419575842357521043355417269398187613727716461956738633541065
015978501361135065874214781350489115628
R(  600)
+ 0
********************
A(  700)
+ 51726268056554225624892719245230240116159209190214308509790732526862
59537582544926034213743862648260551574384795443948781827
R(  700)
+ 1
********************
A(  800)
+ 21029692450574875371943940353292059547817697715120966715389401662836
49469439669076520921736081983874359206657873726116262200288652122955 22
0582
R(  800)
+ 0
********************
A(  900)
+ 85499378381446545382195333994233785670860474538856178543783972886187
43918288981418155946974429101124705343308459768556731801113842610224 63
601610739945251928205
R(  900)
+ 1
********************
A( 1000)
+ 34760727953289832550915007062711658552894409350989328382543413984808
34097725648481080217587513676386754444064436341575383217611557667315 53
801872629042601615330734396254800962954
R( 1000)
+ 0
********************
*STOP* 0
**************************************************************************
```

Ce programme a exécuté 1000 fois le SP 'MUL', 2000 fois le SP 'DIVE' et en moyenne 500 fois le SP 'ADD'. 'MUL', 'DIVE' et 'ADD' représentent respectivement les algorithmes de la multiplication, de la division entière et de l'addition entre deux entiers. Le temps de calcul en minute a été de 0,256.

III - ETUDE DE L'ε-ALGORITHME

Pour montrer l'efficacité de l'arithmétique réelle et de l'arithmétique rationnelle étendue, nous les appliquons à l'accélération de la convergence de la suite de terme générale $S_n = (\frac{3}{4})^n\ n^2$ par l'ε-algorithme qui doit donner les résultats suivants :

$$\varepsilon_6^{(i)} = 0 \quad \forall\, i$$

Pour vérifier ce résultat on a écrit trois programmes utilisant respectivement l'arithmétique réelle de l'ordinateur, l'arithmétique réelle étendue et l'arithmétique rationnelle étendue.

<u>Programme avec l'arithmétique de l'ordinateur</u>

Le sous-programme de l'ε-algorithme scalaire est celui utilisé au chapitre I sous le nom APSIL. Le programme d'appel est écrit pour le calcul des termes $\varepsilon_6^{(i)}$, i = 0, 1, ..., 50. Ce programme est le suivant :

```
C************************************************************************
C********************L E-ALGORITHME VIRGULE FLOTTANTE POUR LA SERIE DE
C*********************TERME GENERAL EGAL A (3.0/4.0)**I*(I*I)
C************************************************************************
      DIMENSION D(50),P(50)
      DOUBLE PRECISION Z,P,D,ZE
      COMMON/BLOK/NM,NI
  100 FORMAT('LA VALEUR DU TERME DE LA SUITE DE RANG ',I2,' EST',D22.15)
  105 FORMAT(20H********************)
      ZE=1.D-15
      ICM=6
      IK=0
      MM=50
      DO 1 I=1,MM
      Z=(3.D0/4.D0)**I*(I*I)
      WRITE(NI,100) I,Z
      CALL APSIL(Z,ICM,IK,MM,ZE,P,D)
    1 WRITE(NI,105)
      END
```

```
      BLOCK DATA
      COMMON/BLOK1/IBASE,LM,IN,KN,NM
      DATA IBASE,LM,IN,KN,NM/10000,200,108,105,2147483647/
      END
```

Les résultats numériques obtenus sont les suivants :

```
LA VALEUR DU TERME DE LA SUITF DE RANG  1 EST  .750000000000000D+00
 EPSILON REEL VIRGULE FLOTTANTE
  EPSILON( 0, 0)=  .750000000000000D+00
**********************
LA VALEUR DU TERME DE LA SUITE DE RANG  2 EST  .225000000000000D+01
 EPSILON REEL VIRGULE FLOTTANTE
  EPSILON( 1, 0)=  .225000000000000D+01
**********************
LA VALEUR DU TERME DE LA SUITE DE RANG  3 EST  .379687500000000D+01
 EPSILON REEL VIRGULE FLOTTANTE
  EPSILON( 0, 2)= -.472500000000000D+02
**********************
LA VALEUR DU TERME DE LA SUITE DE RANG  4 EST  .506250000000000D+01
 EPSILON REEL VIRGULE FLOTTANTE
  EPSILON( 1, 2)=  .107578125000000D+02
**********************
LA VALEUR DU TERME DE LA SUITE DE RANG  5 EST  .593261718750000D+01
 EPSILON REEL VIRGULE FLOTTANTE
  EPSILON( 0, 4)=  .615189873417722D+01
**********************
LA VALEUR DU TERME DE LA SUITE DE RANG  6 EST  .640722656250000D+01
 EPSILON REEL VIRGULE FLOTTANTE
  EPSILON( 1, 4)=  .560769230769230D+01
**********************
LA VALEUR DU TERME DE LA SUITE DE RANG  7 EST  .654071044921875D+01
 EPSILON REEL VIRGULE FLOTTANTE
  EPSILON( 0, 6)=  .144595446727180D-11
**********************
LA VALEUR DU TERME DE LA SUITE DE RANG  8 EST  .640722656250000D+01
 EPSILON REEL VIRGULE FLOTTANTE
  EPSILON( 1, 6)= -.116684439888104D-11
**********************
LA VALEUR DU TERME DE LA SUITE DE RANG  9 EST  .608185958862305D+01
 EPSILON REEL VIRGULE FLOTTANTE
  EPSILON( 2, 6)= -.670130617663744D-12
**********************
LA VALEUR DU TERME DE LA SUITE DE RANG 10 EST  .563135147094727D+01
 EPSILON REEL VIRGULE FLOTTANTE
  EPSILON( 3, 6)=  .221134222044839D-11
**********************
LA VALEUR DU TERME DE LA SUITE DE RANG 11 EST  .511045145988464D+01
 EPSILON REEL VIRGULE FLOTTANTE
  EPSILON( 4, 6)= -.162514446344630D-11
**********************
LA VALEUR DU TERME DE LA SUITE DE RANG 12 EST  .456139469146729D+01
 EPSILON REEL VIRGULE FLOTTANTE
  EPSILON( 5, 6)=  .596411808828634D-12
**********************
LA VALEUR DU TERME DE LA SUITE DE RANG 13 EST  .401497761905193D+01
 EPSILON REEL VIRGULE FLOTTANTE
  EPSILON( 6, 6)= -.703062608131688D-12
```

Programme avec l'arithmétique réelle étendue

L'algorithme de l'ε-algorithme est celui présenté dans le chapitre I. Le sous-programme de cet algorithme EPSIL, en arithmétique réelle étendue, fait appel à l'ensemble des sous-programmes de cette arithmétique donné dans le chapitre III.

Les arguments de ce sous-programme sont :

T : Valeur du terme de la suite initiale

IK : Compteur à initialiser à zéro pour toute nouvelle suite

ICM : indice de la colonne maximum que l'on veut calculer ce paramètre ne doit pas dépasser MM1.

MM1 : taille maximum des tableaux représentant les diagonales P et D. C'est donc l'indice du dernier terme de la suite initiale traitée.

P : tableau de dimensions (L, MM1) représentant la diagonale calculée à l'étape précédente dans la table de l'ε-algorithme

D : tableau de dimensions (L, MM1) représentant la diagonale en cours de calcul dans la table de l'ε-algorithme.

Les arguments B, C, S, CONS, RY sont des tableaux de dimension L servant aux calculs internes au sous-programme ; Il en est de même pour les tableaux RX de dimension LJ et RZ de dimensions (9, LI)

L, LI et LJ sont les dimensions des différents tableaux intervenant comme paramètres du sous-programme et initialisés dans le programme d'appel par le sous-programme 'LØNG'.

Le sous-programme est donc le suivant :

```
      SUBROUTINE EPSIL(T,IK,ICM,MM1,B,C,S,CONS,D,P,RX,RZ,RY,L,LI,LJ)
      IMPLICIT INTEGER (A-Z)
      DIMENSION T(L),B(L),C(L),S(L),CONS(L),D(L,MM1),P(L,MM1)
      DIMENSION RZ(9,L),RY(LJ),RX(LI)
      COMMON/BLOK/MM,LS,LP,NN,NI
104   FORMAT(' DIMENSION INSUFFISANTES DANS EPSIL ')
105   FORMAT(' EPSILON(',I2,',',I2,')')
C     INITIALISATIONS
      IF(IK.NE.0) GO TO 1
      IK=1
      I1=0
      DO 7 I=1,L
      D(I,1)=T(I)
7     P(I,1)=T(I)
      M=1
      L1=M
      MI=M
      IF(2*(ICM/2).NE.ICM) ICM=ICM-1
      GO TO 66
1     IF(I1.EQ.1) RETURN
      M=M+1
      MI=MI+1
      IF(M.GT.(ICM+1)) M=ICM+1
      IF(M.GT.MM1) GO TO 6
      DO 8 I=1,L
```

```
 8        D(I,1)=T(I)
          CALL VFL(B,0,L)
          DO 2 J=2,M
          IF(J.EQ.2) GO TO 9
          DO 10 I=1,L
 10       B(I)=P(I,J-2)
 9        CALL SOUR(S,D(1,J-1),P(1,J-1),L)
          CALL VFL(CONS,1,L)
          CALL DIVR(C,CONS,S,L,MM,RZ,RY,LJ)
          CALL ADDR(D(1,J),B,C,L)
 2        CONTINUE
 C        OPERATIONS DE SORTIE
          L1=M
          IF(2*(M/2).EQ.M) L1=M-1
          DO 5 J=1,M
          DO 5 I=1,L
 5        P(I,J)=D(I,J)
 66       JJ=L1-1
          II=MI-JJ-1
          WRITE(NI,105) II,JJ
          CALL ECRR(D(1,L1),L,RX,LI)
          RETURN
 6        WRITE(NI,104)
          I1=1
          RETURN
          END
```

Le programme d'appel est écrit pour le calcul des termes $\varepsilon_6^{(i)}$, i = 0, 1, ..., 20. Il utilise des réels de 52 chiffres dans la mantisse. Ce programme est le suivant :

```
C*************************************************************************
C ** L E-ALGORITHME AVEC L ARITHMETIQUE REELLE ETENDUE,LA SUITE DE
C ** DEPART ETANT S(N)=(3.0/4.0)**I*(I*I)
C*************************************************************************
      IMPLICIT INTEGER (A-Z)
      DIMENSION TA(15),TB(15),TC(15),TD(15),TE(15),TF(15),TG(15),TI(15)
      DIMENSION T(15),RY(30),RX(80),RZ(9,15)
      DIMENSION IN1(15),IN2(15),IN3(15),IN4(15),D(15,21),P(15,21)
      DIMENSION S(15)
      DIMENSION TH(15)
      COMMON/BLOK/MM,LS,LP,NN,NI
100   FORMAT('LA VALEUR DU TERME DE LA SUITE DE RANG ',I2)
101   FORMAT(10X,20H********************)
      ICM=6
      MM1=21
      IK=0
      CALL LONG(L,LI,LJ)
      CALL VFL(TA,3,L)
      CALL VFL(TB,4,L)
      CALL VFL(S,1,L)
      CALL VFL(TE,1,L)
      CALL VFL(TG,1,L)
      DO 1 I=1,MM1-1
      CALL VFL(TC,I,L)
      CALL MULR(TD,TC,TC,L,RY,LJ)
      CALL MULR(TF,TA,TE,L,RY,LJ)
      CALL MULR(TH,TB,TG,L,RY,LJ)
      CALL MULR(TI,TF,TD,L,RY,LJ)
      CALL DIVR(T,TI,TH,L,MM,RZ,RY,LJ)
      WRITE(NI,100) I
      CALL ECRR(T,L,RX,LI)
```

```
      CALL EPSIL(T,IK,ICM,MM1,IN1,IN2,IN3,IN4,D,P,RX,RZ,RY,L,LI,LJ)
      DO 2 J=1,L
      TG(J)=TH(J)
    2 TE(J)=TF(J)
      WRITE(NI,101)
    1 CONTINUE
      END
```

```
      BLOCK DATA
      COMMON/BLOK/MM,LS,LP,NN,NI
      DATA MM,LS,LP,NN,NI/52,6,4,2147483647,108/
      END
```

Les résultats numériques obtenus sont les suivants :

```
LA VALEUR DU TERME DE LA SUITE DE RANG  1
+ ,7500000000000000000000000000000000000000A+0
 EPSILON( 0, 0)
+ ,7500000000000000000000000000000000000000A+0
          ********************
LA VALEUR DU TERME DE LA SUITE DE RANG  2
+ ,2250000000000000000000000000000000000000A+1
 EPSILON( 1, 0)
+ ,2250000000000000000000000000000000000000A+1
          ********************
LA VALEUR DU TERME DE LA SUITE DE RANG  3
+ ,3796875000000000000000000000000000000000A+1
 EPSILON( 0, 2)
- ,4725000000000000000000000000000000000000A+2
          ********************
LA VALEUR DU TERME DE LA SUITE DE RANG  4
+ ,5062500000000000000000000000000000000000A+1
 EPSILON( 1, 2)
+ ,1075781250000000000000000000000000000000A+2
          ********************
LA VALEUR DU TERME DE LA SUITE DE RANG  5
+ ,5932617187500000000000000000000000000000A+1
 EPSILON( 0, 4)
+ ,61518987341772151898734177215189873418A+1
          ********************
LA VALEUR DU TERME DE LA SUITE DE RANG  6
+ ,6407226562500000000000000000000000000000A+1
 EPSILON( 1, 4)
+ ,5607692307692307692307692307692307692329A+1
          ********************
LA VALEUR DU TERME DE LA SUITE DE RANG  7
+ ,6540710449218750000000000000000000000000A+1
 EPSILON( 0, 6)
- ,4592A-47
          ********************
LA VALEUR DU TERME DE LA SUITE DE RANG  8
+ ,6407226562500000000000000000000000000000A+1
 EPSILON( 1, 6)
+ ,43A-49
          ********************
LA VALEUR DU TERME DE LA SUITE DE RANG  9
+ ,6081859588623046875000000000000000000000A+1
 EPSILON( 2, 6)
+ ,1034A-47
          ********************
```

```
LA VALEUR DU TERME DE LA SUITE DE RANG 10
+ ,5631351470947265625000000000000000000000000000000000A+1
 EPSILON( 3, 6)
+ ,5039A-47
          *********************
LA VALEUR DU TERME DE LA SUITE DE RANG 11
+ ,5110451459884643554687500000000000000000000000000000A+1
 EPSILON( 4, 6)
- ,5553A-47
          *********************
LA VALEUR DU TERME DE LA SUITE DE RANG 12
+ ,4561394691467285156250000000000000000000000000000000A+1
 EPSILON( 5, 6)
+ ,2204A-47
          *********************
LA VALEUR DU TERME DE LA SUITE DE RANG 13
+ ,4014977619051933288574218750000000000000000000000000A+1
 EPSILON( 6, 6)
- ,7132A-48
          *********************
LA VALEUR DU TERME DE LA SUITE DE RANG 14
+ ,3492317810654640197753906250000000000000000000000000A+1
 EPSILON( 7, 6)
+ ,20621A-47
          *********************
LA VALEUR DU TERME DE LA SUITE DE RANG 15
+ ,3006778727285563945770263671875000000000000000000000A+1
 EPSILON( 8, 6)
- ,27890A-47
          *********************
LA VALEUR DU TERME DE LA SUITE DE RANG 16
+ ,2565784513950347900390625000000000000000000000000000A+1
 EPSILON( 9, 6)
+ ,24299A-47
          *********************
LA VALEUR DU TERME DE LA SUITE DE RANG 17
+ ,2172397630463819950819015502929687500000000000000000A+1
 EPSILON(10, 6)
- ,27140A-47
          *********************
LA VALEUR DU TERME DE LA SUITE DE RANG 18
+ ,1826618076825980097055435180664062500000000000000000A+1
 EPSILON(11, 6)
+ ,291971A-47
          *********************
LA VALEUR DU TERME DE LA SUITE DE RANG 19
+ ,1526410013273562071844935417175292968750000000000000A+1
 EPSILON(12, 6)
- ,277090A-47
          *********************
LA VALEUR DU TERME DE LA SUITE DE RANG 20
+ ,1268484775573597289621829986572265625000000000000000A+1
 EPSILON(13, 6)
+ ,220442A-47
          *********************
```

Programme avec l'arithmétique rationnelle étendue

On présente maintenant l'ensemble des sous-programmes de l'accélération de la convergence par l'ε-algorithme en arithmétique rationnelle. Pour cela, on utilise l'algorithme présenté au chapitre I avec des variables déclarées en T.E.P. et les opérations de l'arithémtique rationnelle étendue, dont les sous-programmes ont été donnés dans le chapitre II. De plus on a besoin du sous-programme CØNV (A, C, L) qui convertit le rationnel donné dans le tableau A de dimension L en un réel C défini avec une mantisse dont le nombre de chiffres est celui de la double précision de la machine.

Les arguments du sous-programme de l'ε-algorithme en arithmétique rationnelle, 'EPSIL', sont les suivants :

T1, T2 : numérateur et dénominateur du terme de la suite initiale

IK : compteur à initialiser à zéro pour toute nouvelle suite

ICM : indice de la colonne maximum que l'on veut calculer

MM : taille maximum des tableaux représentant les diagonales

DN, DD : tableaux de dimensions (L, MM) représentant le numérateur et le dénominateur de la diagonale calculée à l'étape précédente

CN, CD : même chose que DN, DD mais pour la diagonale en cours de calcul.

Les arguments INN, IDD, BN, BD, IN1, IN2, IN3, IN4, IN5, IN6, IN7 sont des tableaux de dimension L servant aux calculs internes au sous-programme ; il en est de même pour TEL qui est de dimension LR.

L et LR sont les dimensions des différents tableaux intervenant dans le sous-programme et initialisées dans le programme d'appel par le sous-programme 'INI'.

Les sous-programmes CØNV et EPSIL sont donc les suivants

```
      SUBROUTINE EPSIL(T1,T2,ICM,MM,IK,L,DN,DD,CN,CD,INN,IDD,BN,BD,IN1,I
     1N2,IN3,IN4,IN5,IN6,IN7,TEL,LR)
C***********************************************************************
C** SOUS-PROGRAMME DE L'EPSILON ALGORITHME.ON PART D'UN ELEMENT DE LA SUITE
C** DE DEPART EN RATIONNEL -TN/TD- ET ON CALCULE LA DIAGONALE MONTANTE A PARTIR
C** DE CET ELEMENT,-DN/DD-,EN UTILISANT LA DIAGONALE PRECEDENTE,-CN/CD-,TOUS LES
C** ENTIERS SONT DES TEP DE LONGUEUR -L-.
C** LES TABLEAUX -INN,IDD,BN,BD,IN1,IN2,IN3,IN4,IN5,IN6,IN7- SERVENT POUR LES
C** CALCULS INTERMEDIAIRES.
C** IK : PARAMETRE A METTRE A ZERO DANS LE PROGRAMME PRINCIPAL A CHAQUE FOIS QUE
C** L'ON CHANGE DE SUITE DE DEPART.
C** ICM : INDICE DE LA DERNIERE COLONNE QUE L'ON VEUT CALCULER.IL DOIT ETRE AU
C** PLUS EGAL A -MM-.
C** MM : INDICE DE CONTROLE DE LA DIMENSION DE LA TABLE EPSILON.IL DOIT ETRE
C** EGAL AU NOMBRE D'ELEMENTS MAXIMUM DE LA DERNIERE DIAGONALE QUE L'ON VEUT
C** CALCULER (C'EST LE DEUXIEME INDICE DES TABLEAUX -DN,DD,CN,CD-).
C***********************************************************************
```

```
      IMPLICIT INTEGER (A-O,S-W,Y-Z)
      IMPLICIT DOUBLE PRECISION (R,X)
      DIMENSION DN(L,MM),DD(L,MM),CN(L,MM),CD(L,MM),INN(L),IDD(L),BN(L)
      DIMENSION BD(L),IN1(L),IN2(L),IN3(L),IN4(L),IN5(L),IN6(L),IN7(L)
      DIMENSION T1(L),T2(L)
      DIMENSION TEL(LR)
      COMMON/BLOK1/IBASE,LM,IN,KN,NM,PR
  101 FORMAT('EPSILON(',I2,',',I2,')')
  102 FORMAT('---------------------------------=',D21.15)
  103 FORMAT('DIMENSION INSUFISANTE DANS LE TABLEAU EPSILON , AUGMENTER
     1 MM')
  104 FORMAT(' LE CALCUL DE EPSILON(',I2,',',I2,') EST IMPOSSIBLE',/,
     1'UN DEPASSEMENT DE CAPACITE SE PRODUIT AU TERME EPSILON(',I2,',',I
     12,')')
C***********************************************************************
C** INITIALISATION DU PREMIER ELEMENT EPSILON(0,0).
C***********************************************************************
      ITA=0
      IF(IK.NE.0) GO TO 1
      IK=1
      I1=0
      DO 40 JJK=1,L
      CN(JJK,1)=T1(JJK)
   40 CD(JJK,1)=T2(JJK)
      CALL CONV(CN(1,1),R1,L)
      CALL CONV(CD(1,1),R2,L)
      R=R1/R2
      M=1
      M1=M
      LL=M
      IF(2*(ICM/2).NE.ICM) ICM=ICM-1
      II=0
      JJ=0
      WRITE(IN,101) II,JJ
      CALL ECR(CN(1,1),L,TEL,LR)
      WRITE(IN,102) R
      CALL ECR(CD(1,1),L,TEL,LR)
      RETURN
C***********************************************************************
C** CALCUL DES ELEMENTS D'UNE DIAGONALE EN UTILISANT LA REGLE DU LOSANGE DE
C** L'EPSILON ALGORITHME.
C***********************************************************************
    1 IF(I1.EQ.1) RETURN
      M=M+1
      M1=M1+1
      IF(M.GT.(ICM+1)) M=ICM+1
      IF(M.GT.MM) GO TO 9
      DO 41 JJK=1,L
      DN(JJK,1)=T1(JJK)
   41 DD(JJK,1)=T2(JJK)
      DO 2 J=2,M
      DN(1,J)=1
      DD(1,J)=1
      DO 2 I=2,L
      DN(I,J)=0
    2 DD(I,J)=0
      DO 3 J=2,M
      DO 33 I=2,L
      BN(I)=0
   33 BD(I)=0
      BN(1)=1
      BD(1)=1
      BD(L)=1
      IF(J.EQ.2) GO TO 11
      DO 34 I=1,L
      BN(I)=CN(I,J-2)
   34 BD(I)=CD(I,J-2)
   11 CN(1,J-1)=-CN(1,J-1)
```

```
      CALL ADDFRA(IDD,INN,DN(1,J-1),DD(1,J-1),CN(1,J-1),CD(1,J-1),IN1,IN
     12,IN3,IN4,IN5,IN6,IN7,L,ITA)
      CN(1,J-1)=-CN(1,J-1)
      IF (ITA.NE.0) GO TO 20
      CALL ADDFRA(DN(1,J),DD(1,J),INN,IDD,BN,BD,IN1,IN2,IN3,IN4,IN5,IN6,
     1IN7,L,ITA)
      IF (ITA.NE.0) GO TO 20
 3    CONTINUE
C***********************************************************************
C** CONVERSION DU RESULTAT  EN REEL ET IMPRESSION DU DERNIER ELEMENT DE LA
C** DIAGONALE QUE L'ON VIENT DE CALCULER CORRESPONDANT A UNE COLONNE PAIRE.
C***********************************************************************
      LL=M
      IF(2*(M/2).EQ.M) LL=M-1
      IS=1
      IF(DN(1,LL).LT.0.AND.DD(1,LL).LT.0) IS=-1
      DN(1,LL)=IS*DN(1,LL)
      DD(1,LL)=IS*DD(1,LL)
      CALL CONV(DN(1,LL),R1,L)
      CALL CONV(DD(1,LL),R2,L)
      R=R1/R2
      JJ=LL-1
      II=M1-JJ-1
      WRITE(IN,101) II,JJ
      CALL ECR(DN(1,LL),L,TEL,LR)
      WRITE(IN,102) R
      CALL ECR(DD(1,LL),L,TEL,LR)
C***********************************************************************
C** PREPARATION DE LA DIAGONALE -CN/CD- POUR LE PROCHAIN APPELE DU
C** SOUS-PROGRAMME.
C***********************************************************************
      DO 6 J=1,M
      DO 6 I=1,L
      CN(I,J)=DN(I,J)
 6    CD(I,J)=DD(I,J)
      RETURN
C***********************************************************************
C** CAS OU IL Y A DEBORDEMENT DANS LES ELEMENTS DE LA DIAGONALE (TEST SUR -MM-).
C***********************************************************************
 9    WRITE(IN,103)
      I1=1
      RETURN
C***********************************************************************
C** DANS LE CAS OU IL Y A DEPASSEMENT DE CAPACITE ON IMPRIME UN MESSAGE
C** CORRESPONDANT EN LOCALISANT LE TERME OU CE DEPASSEMENT A EU LIEU.
C***********************************************************************
 20   LL=M
      IF(2*(M/2).EQ.M) LL=M-1
      JJ=LL-1
      II=M1-JJ-1
      II1=JJ-J+1
      JJ1=II+J-1
      WRITE(IN,104) II,JJ,II1,JJ1
      STOP
      END
```

```
      SUBROUTINE CONV(A,C,L)
C***********************************************************************
C** CONVERSION D'UN TEP -A(L)- EN UN REEL DOUBLE PRECISION -C-.
C**A(L) : TABLEAU CONTENANT L'ENTIER A CONVERTIR EN DOUBLE PRECISION
C** C : REEL DOUBLF PRECISION DANS LEWUEL ON CONVERTIT A(L)
C** D : REEL WU'ON DIVISE A CHAQUE ETAPE PAR 10 ET QUI PERMET DE REMPLIR -C-
C** CHIFFRE PAR CHIFFRE
C***********************************************************************
```

```
      IMPLICIT DOUBLE PRECISION (C,D)
      IMPLICIT LOGICAL (E)
      IMPLICIT INTEGER (A-B,F-Z)
      DIMENSION A(L)
      COMMON/BLOK1/IBASE,LM,IN
      D=1.D0
C************************************************
C** RECHERCHE DU PREMIER MOT NON NUL DANS -A(L)-.
C** SI TOUT A(L) EST NUL ,C=0
C************************************************
      DO 2 I=2,L
      IF(A(I).NE.0) GO TO 3
2     CONTINUE
      C=0
      RETURN
C******************************************************************************
C** CAS OU LE TEP -A(L)- EST NON NUL
C** VIDAGE DU PREMIER MOT NON NUL DE -A- DANS -C- EN COMMENCANT PAR LA RECHERCHE
C** DU PREMIER CHIFFRE NON NUL
C** B: EST UN BOULEEN QUI DEVIENT VRAI DES QUE L'ON RENCONTRE LE PREMIER CHIFFRE
C** NON NUL-G-,DANS LE PREMIER MOT NON NUL DE -A-,ET DANS CE CAS ON COMMENCE A
C**METTRE CES CHIFFRES DANS-C-,SINON ON DECREMENTE-K-DE 1 ET ON RECOMMENCE CETTE
C** RECHERCHE
C******************************************************************************
3     K=LOGBAS(IBASE)*(L-I+1)
      II=I
      C=0.D0
      E=.FALSE.
      X=A(I)
      BAS=IBASE/10
9     Q=X/BAS
      R=MOD(X,BAS)
      IF(Q.NE.0) GO TO 7
      IF(E) GO TO 6
      K=K-1
      GO TO 8
7     E=.TRUE.
6     D=D/10
      C=C+D*Q
8     BAS=BAS/10
      X=R
      IF(BAS.NE.0) GO TO 9
C******************************************************************************
C** VIDAGE DU RESTE DES MOTS DE-A(L)-DANS-C-PAR LA MEME TECHNIQUE QUE POUR LE
C** PREMIER MOT NON NUL
C******************************************************************************
      IF(I.GE.L) GO TO 11
      II=I+1
      DO 5 I=II,L
      X=A(I)
      BAS=IBASE/10
91    Q=X/BAS
      R=MOD(X,BAS)
      D=D/10
      C=C+D*Q
      BAS=BAS/10
      X=R
      IF(BAS.NE.0) GO TO 91
5     CONTINUE
C************************************************
C** NORMALISATION DU REEL -C-.
C************************************************
11    C=C*10**K
      IF(A(1).LT.0) C=-C
      RETURN
      END
```

Le programme d'appel est écrit pour le calcul des termes $\varepsilon_6^{(i)}$, i = 0, 1, ... 20 ; il utilise des rationnels pouvants comportés 200 chiffres. Ce programme est le suivant :

```
C*******************************************************************************
C** L'E-ALGORITHME EVEC UNE SUITE DE DEPART DONT LES TERMES SONT DONNES EN
C** RATIONNELLE LA SUITE ETANT (3.0/4.0)**I*(I*I)
C*******************************************************************************
      IMPLICIT INTEGER (A-Z)
      DIMENSION DN(51,21),DD(51,21),CN(51,21),CD(51,21),INN(51),IDD(51)
      DIMENSION BN(51),BD(51),IN1(51),IN2(51),IN3(51),IN4(51),IN5(51)
      DIMENSION IN6(51),IN7(51),TEL(240)
      DIMENSION TN(51),TD(51),A(51),AA(51),AN(51),AD(51),B(51),BB(51)
      DIMENSION EN(51),ED(51),C(51),IN8(51)
      COMMON/BLOK1/IBASE,LM,IN,KN,NM
100   FORMAT('RANG DE LA SUITE ',I2)
105   FORMAT(10X,20H********************)
      ITA=0
      IK=0
      ICM=6
      MM=20
      CALL INI(L,LM)
      CALL STO(1,AA,L)
      CALL STO(1,BB,L)
      CALL STO(1,ED,L)
      CALL STO(3,B,L)
      CALL STO(4,A,L)
      DO 3 I=1,MM
      WRITE(IN,100) I
      CALL STO(I,C,L)
      CALL MUL(C,C,EN,L,ITA)
      IF(ITA.NE.0) GO TO 1
      CALL MUL(B,BB,AN,L,ITA)
      IF(ITA.NE.0) GO TO 1
      CALL MUL(A,AA,AD,L,ITA)
      IF(ITA.NE.0) GO TO 1
      DO 2 J=1,L
      AA(J)=AD(J)
  2   BB(J)=AN(J)
      CALL MULFRA(TN,TD,AN,AD,EN,ED,IN1,IN2,IN3,IN4,IN5,IN6,IN7,IN8,L,
     1ITA)
      IF(ITA.NE.0) GO TO 1
      OUT PUT ' LA VALEUR DE LA SUITE'
      CALL ECR(TN,L,TEL,LR)
      OUT PUT '-------------------------------------'
      CALL ECR(TD,L,TEL,LR)
      CALL EPSIL(TN,TD,ICM,MM,IK,L,DN,DD,CN,CD,INN,IDD,BN,BD,IN1,IN2,IN3
     1,IN4,IN5,IN6,IN7,TEL,LR)
3     WRITE(IN,105)
 1    CONTINUE
      END
```

```
      BLOCK DATA
      COMMON/BLOK1/IBASE,LM,IN,KN,NM
      DATA IBASE,LM,IN,KN,NM/10000,200,108,105,2147483647/
      END
```

Les résultats numériques obtenus sont les suivants

```
RANG DE LA SUITE  1
  LA VALEUR DE LA SUITE
+ 3
 ------------------------------------
+ 4
EPSILON( 0, 0)
+ 3
-------------------------------= .750000000000000D+00
+ 4
          ********************
RANG DE LA SUITE  2
  LA VALEUR DE LA SUITE
+ 9
 ------------------------------------
+ 4
EPSILON( 1, 0)
+ 9
-------------------------------= .225000000000000D+01
+ 4
          ********************
RANG DE LA SUITE  3
  LA VALEUR DE LA SUITE
+ 243
 ------------------------------------
+ 64
EPSILON( 0, 2)
- 189
-------------------------------=-.472500000000000D+02
+ 4
          ********************
RANG DE LA SUITE  4
  LA VALEUR DE LA SUITE
+ 81
 ------------------------------------
+ 16
EPSILON( 1, 2)
+ 1377
-------------------------------= .107578125000000D+02
+ 128
          ********************
RANG DE LA SUITE  5
  LA VALEUR DE LA SUITE
+ 6075
 ------------------------------------
+ 1024
EPSILON( 0, 4)
+ 486
-------------------------------= .615189873417722D+01
+ 79
          ********************
RANG DE LA SUITE  6
  LA VALEUR DE LA SUITE
+ 6561
 ------------------------------------
+ 1024
EPSILON( 1, 4)
+ 729
-------------------------------= .560769230769231D+01
+ 130
          ********************
RANG DE LA SUITE  7
  LA VALEUR DE LA SUITE
+ 107163
 ------------------------------------
+ 16384
```

```
EPSILON( 0, 6)
+ 0
-------------------------------= .000000000000000D+00
+ 130
          ********************
RANG DE LA SUITE  8
  LA VALEUR DE LA SUITE
+ 6561
 -------------------------------------
+ 1024
EPSILON( 1, 6)
+ 0
-------------------------------= .000000000000000D+00
+ 440
          ********************
RANG DE LA SUITE  9
  LA VALEUR DE LA SUITE
+ 1594323
 -------------------------------------
+ 262144
EPSILON( 2, 6)
+ 0
-------------------------------= .000000000000000D+00
+ 1568
          ********************
RANG DE LA SUITE 10
  LA VALEUR DE LA SUITE
+ 1476225
 -------------------------------------
+ 262144
EPSILON( 3, 6)
+ 0
-------------------------------= .000000000000000D+00
+ 6016
          ********************
RANG DE LA SUITE 11
  LA VALEUR DE LA SUITE
+ 21434787
 -------------------------------------
+ 4194304
EPSILON( 4, 6)
+ 0
-------------------------------= .000000000000000D+00
+ 25088
          ********************
RANG DE LA SUITE 12
  LA VALEUR DE LA SUITE
+ 4782969
 -------------------------------------
+ 1048576
EPSILON( 5, 6)
+ 0
-------------------------------= .000000000000000D+00
+ 112640
          ********************
RANG DE LA SUITE 13
  LA VALEUR DE LA SUITE
+ 269440587
 -------------------------------------
+ 67108864
EPSILON( 6, 6)
+ 0
-------------------------------= .000000000000000D+00
+ 532480
          ********************
RANG DE LA SUITE 14
  LA VALEUR DE LA SUITE
+ 234365481
 -------------------------------------
+ 67108864
```

```
EPSILON( 7, 6)
+ 0
-------------------------------= .0000000000000000D+00
+ 2588672
         ********************
RANG DE LA SUITE 15
  LA VALEUR DE LA SUITE
+ 3228504075
 ----------------------------------
+ 1073741824
EPSILON( 8, 6)
+ 0
-------------------------------= .0000000000000000D+00
+ 12713984
         ********************
RANG DE LA SUITE 16
  LA VALEUR DE LA SUITE
+ 43046721
 ----------------------------------
+ 16777216
EPSILON( 9, 6)
+ 0
-------------------------------= .0000000000000000D+00
+ 62390272
         ********************
RANG DE LA SUITE 17
  LA VALEUR DE LA SUITE
+ 37321507107
 ----------------------------------
+ 17179869184
EPSILON(10, 6)
+ 0
-------------------------------= .0000000000000000D+00
+ 304087040
         ********************
RANG DE LA SUITE 18
  LA VALEUR DE LA SUITE
+ 31381059609
 ----------------------------------
+ 17179869184
EPSILON(11, 6)
+ 0
-------------------------------= .0000000000000000D+00
+ 1468006400
         ********************
RANG DE LA SUITE 19
  LA VALEUR DE LA SUITE
+ 419576389587
 ----------------------------------
+ 274877906944
EPSILON(12, 6)
+ 0
-------------------------------= .0000000000000000D+00
+ 7012876288
         ********************
RANG DE LA SUITE 20
  LA VALEUR DE LA SUITE
+ 87169610025
 ----------------------------------
+ 68719476736
EPSILON(13, 6)
+ 0
-------------------------------= .0000000000000000D+00
+ 33151778816
         ********************
```

Conclusion

D'après les résultats obtenus, l'arithmétique réelle étendue est plus précise que l'arithmétique de l'ordinateur puisqu'on obtient des résultats pour les termes $\varepsilon_6^{(i)}$ de l'ordre de 10^{-47} or l'arithmétique de l'ordinateur ne donne que des résultats de l'ordre de 10^{-12}. Par contre l'arithmétique rationnelle nous permet de trouver les résultats exacts puisque tous les termes $\varepsilon_6^{(i)}$ pour i = 0, 1, ..., 13 sont nuls. L'amélioration des résultats compense la dégradation des temps de calcul qui sont de :

Type d'arithmétique	temps en minute
arithmétique de l'ordinateur	0,010
arithmétique réelle	0,253
arithmétique rationnelle	0,522

IV - ETUDE DU DÉVELOPPEMENT EN SÉRIE ENTIÈRE DE LA FONCTION EXPONENTIELLE

Le développement en série entière de la fonction exponentielle est donné par :

$$e^x = \sum_{i=o}^{\infty} \frac{x^i}{i!}$$

posons $$S_n(x) = \sum_{i=o}^{\infty} \frac{x^i}{i!}$$

On va calculer la fonction e^x en utilisant l'arithmétique de l'ordinateur, l'arithmétique réelle étendue et l'arithmétique rationnelle étendue. Le critère d'arrêt dans le calcul du développement sera donné par :

$$|(x^n/_{n!}) \ / \ S_n(x)| \leq 10^{-14}$$

Ce choix est pleinement justifié lorsque x est négatif puisqu'alors

$$|e^{-x} - S_{n-1}(x)| \leq |\frac{x^n}{x!}|$$

Les tests numériques portent sur les valeurs de x suivantes :

-10, -15, -20, -25, -30

Arithmétique de l'ordinateur

Le programme qui permet de calculer le développement en série entière de e^x est le suivant :

```
C*********************************************************************************
C** CALCUL DE L EXPONENTIEL DE X PAR LE DEVELOPPEMENT EN SERIE ENTIERE
C**  L ARETHMETIQUE UTILISEE EST CELLE DE L ORDINATEUR EN REEL VIRGULE FLOTTANTE
C*********************************************************************************
      IMPLICIT INTEGER(A,X)
      IMPLICIT DOUBLE PRECISION(B-T)
      S=1.
      T=1.
      A=1
      READ(105,1) X
      Y=DEXP(DFLOAT(X))
      WRITE(108,200) X,Y
  200 FORMAT('X=',I4,/,'EXP(X)=',D22.15)
1     FORMAT(I3)
  5   T=T*DFLOAT(X)/A
      S=S+T
      R=T/S
      IF(A.EQ.-1) GO TO 7
      IF(MOD(A,10).NE.0) GO TO 6
7     WRITE(108,100) A,S,R
6     A=A+1
      IF(DABS(R).GT.1.E-14) GO TO 5
      WRITE(108,100) A,S,R
100   FORMAT('NOMBRE D ITERATION ',I4,/,' VALEUR DE L EXPONENTIEL',
     1D22.15,/,'PRECISION DU RESULTAT',D22.15)
      END
```

Les résultats obtenus sont ceux auxquels on s'attendait puisqu'on sait que pour x plus petit que -15 l'arithmétique de l'ordinateur ne donne plus le résultat cherché. Ces résultats sont les suivants :

```
X= -10
EXP(X)=  .4539992369245740-04
NOMBRE D ITERATION   10
 VALEUR DE L EXPONENTIEL  .1342587301587300+04
PRECISION DU RESULTAT  .2052553244873210+01
NOMBRE D ITERATION   20
 VALEUR DE L EXPONENTIEL  .1339686599569600+02
PRECISION DU RESULTAT  .3068118785865800+01
NOMBRE D ITERATION   30
 VALEUR DE L EXPONENTIEL  .9703415799136510-03
PRECISION DU RESULTAT  .3885217027545480+01
NOMBRE D ITERATION   40
 VALEUR DE L EXPONENTIEL  .4540234208017080-04
PRECISION DU RESULTAT  .2699458624764790-03
NOMBRE D ITERATION   50
 VALEUR DE L EXPONENTIEL  .4539992974576350-04
PRECISION DU RESULTAT  .7242190538719860-10
NOMBRE D ITERATION   57
 VALEUR DE L EXPONENTIEL  .4539992974522310-04
PRECISION DU RESULTAT  .3097961962369850-14
*STOP* 0
*****************************************************************************
X= -15
EXP(X)=  .3059023470086690-06
NOMBRE D ITERATION   10
 VALEUR DE L EXPONENTIEL  .9377044475446430+05
PRECISION DU RESULTAT  .1694663914274840+01
NOMBRE D ITERATION   20
 VALEUR DE L EXPONENTIEL  .5761283325879870+05
PRECISION DU RESULTAT  .2372364031862020+01
NOMBRE D ITERATION   30
 VALEUR DE L EXPONENTIEL  .2373976928574650+03
PRECISION DU RESULTAT  .3045097500397150+01
NOMBRE D ITERATION   40
 VALEUR DE L EXPONENTIEL  .3647345219310000-01
PRECISION DU RESULTAT  .3715595455849330+01
NOMBRE D ITERATION   50
 VALEUR DE L EXPONENTIEL  .7839810053671340-06
PRECISION DU RESULTAT  .2674130145129950+01
NOMBRE D ITERATION   60
 VALEUR DE L EXPONENTIEL  .3058794107546180-06
PRECISION DU RESULTAT  .1444609441882750-04
NOMBRE D ITERATION   70
 VALEUR DE L EXPONENTIEL  .3058785363781200-06
PRECISION DU RESULTAT  .5786741665121750-11
NOMBRE D ITERATION   76
 VALEUR DE L EXPONENTIEL  .3058785363778110-06
PRECISION DU RESULTAT -.2121698657926820-14
*STOP* 0
*****************************************************************************
X= -20
EXP(X)=  .2061153692167750-08
NOMBRE D ITERATION   10
 VALEUR DE L EXPONENTIEL  .1859623680776010+07
PRECISION DU RESULTAT  .1517441145597050+01
NOMBRE D ITERATION   20
 VALEUR DE L EXPONENTIEL  .2127721034254430+08
PRECISION DU RESULTAT  .2025632281109830+01
NOMBRE D ITERATION   30
 VALEUR DE L EXPONENTIEL  .1599694096422920+07
PRECISION DU RESULTAT  .2530479672378570+01
NOMBRE D ITERATION   40
 VALEUR DE L EXPONENTIEL  .4442034363124680+04
PRECISION DU RESULTAT  .3033701487574190+01
```

```
NOMBRE D ITERATION   50
 VALEUR DE L EXPONENTIEL  .104691697166443D+01
PRECISION DU RESULTAT  .353600337188641D+01
NOMBRE D ITERATION   60
 VALEUR DE L EXPONENTIEL  .343169269779526D-04
PRECISION DU RESULTAT  .403753685706037D+01
NOMBRE D ITERATION   70
 VALEUR DE L EXPONENTIEL  .187689474654812D-08
PRECISION DU RESULTAT  .525115384403353D+00
NOMBRE D ITERATION   80
 VALEUR DE L EXPONENTIEL  .165976153951205D-08
PRECISION DU RESULTAT  .101771513653481D-05
NOMBRE D ITERATION   90
 VALEUR DE L EXPONENTIEL  .165976120437106D-08
PRECISION DU RESULTAT  .502016753316262D-12
NOMBRE D ITERATION   94
 VALEUR DE L EXPONENTIEL  .165976120437091D-08
PRECISION DU RESULTAT -.515817449168820D-14
*STOP* 0
*****************************************************************************
 X= -25
 EXP(X)=  .138879437464046D-10
 NOMBRE D ITERATION   10
  VALEUR DE L EXPONENTIEL  .186134957973504D+08
 PRECISION DU RESULTAT  .141191680805411D+01
 NOMBRE D ITERATION   20
  VALEUR DE L EXPONENTIEL  .205602378984708D+10
 PRECISION DU RESULTAT  .181822414675676D+01
 NOMBRE D ITERATION   30
  VALEUR DE L EXPONENTIEL  .147150204620812D+10
 PRECISION DU RESULTAT  .222218041106418D+01
 NOMBRE D ITERATION   40
  VALEUR DE L EXPONENTIEL  .386225981208676D+08
 PRECISION DU RESULTAT  .262490622945983D+01
 NOMBRE D ITERATION   50
  VALEUR DE L EXPONENTIEL  .856894029516582D+05
 PRECISION DU RESULTAT  .302690258512990D+01
 NOMBRE D ITERATION   60
  VALEUR DE L EXPONENTIEL  .263712343537518D+02
 PRECISION DU RESULTAT  .342842941611421D+01
 NOMBRE D ITERATION   70
  VALEUR DE L EXPONENTIEL  .156420194597416D-02
 PRECISION DU RESULTAT  .382915326350024D+01
 NOMBRE D ITERATION   80
  VALEUR DE L EXPONENTIEL  .219679590335970D-06
 PRECISION DU RESULTAT  .435194362870429D+00
 NOMBRE D ITERATION   90
  VALEUR DE L EXPONENTIEL  .197081704232768D-06
 PRECISION DU RESULTAT  .222853100841258D-05
 NOMBRE D ITERATION  100
  VALEUR DE L EXPONENTIEL  .197081609401829D-06
 PRECISION DU RESULTAT  .338338260664125D-11
 NOMBRE D ITERATION  106
  VALEUR DE L EXPONENTIEL  .197081609401696D-06
 PRECISION DU RESULTAT -.285147617384443D-14
 *STOP* 0
*****************************************************************************
 X= -30
 EXP(X)=  .935762426747255D-13
 NOMBRE D ITERATION   10
  VALEUR DE L EXPONENTIEL  .121254849571429D+09
 PRECISION DU RESULTAT  .134199345313490D+01
 NOMBRE D ITERATION   20
  VALEUR DE L EXPONENTIEL  .852917122100321D+11
 PRECISION DU RESULTAT  .168032637647466D+01
```

```
NOMBRE D ITERATION   30
 VALEUR DE L EXPONENTIEL  .38484261253590D+12
PRECISION DU RESULTAT  .20169466571410D+01
NOMBRE D ITERATION   40
 VALEUR DE L EXPONENTIEL  .63336540117266D+11
PRECISION DU RESULTAT  .23526145852174D+01
NOMBRE D ITERATION   50
 VALEUR DE L EXPONENTIEL  .87822922927799D+09
PRECISION DU RESULTAT  .26876949561039D+01
NOMBRE D ITERATION   60
 VALEUR DE L EXPONENTIEL  .16655843003849D+07
PRECISION DU RESULTAT  .30223861901610D+01
NOMBRE D ITERATION   70
 VALEUR DE L EXPONENTIEL  .62252467446148D+03
PRECISION DU RESULTAT  .33568058613526D+01
NOMBRE D ITERATION   80
 VALEUR DE L EXPONENTIEL  .55969260710685D-01
PRECISION DU RESULTAT  .36899721186991D+01
NOMBRE D ITERATION   90
 VALEUR DE L EXPONENTIEL  .17479458117699D-04
PRECISION DU RESULTAT  .33608503110642D+00
NOMBRE D ITERATION  100
 VALEUR DE L EXPONENTIEL  .16019984288107D-04
PRECISION DU RESULTAT  .34471464253738D-05
NOMBRE D ITERATION  110
 VALEUR DE L EXPONENTIEL  .16019971619552D-04
PRECISION DU RESULTAT  .11960756029786D-10
NOMBRE D ITERATION  117
 VALEUR DE L EXPONENTIEL  .16019971619511D-04
PRECISION DU RESULTAT  .40813706249572D-14
*STOP* 0
******************************************************************************
```

<u>Arithmétique réelle étendue</u>

Le programme du calcul de e^x utilise l'ensemble des sous-programmes de l'arithmétique réelle étendue présentée au chapitre III. Les réels utilisés ont une mantisse de 60 chiffres. Le programme d'appel est le suivant :

```
C*****************************************************************************
C** CALCUL DE L EXPONENTIEL DE X PAR LE DEVELOPPEMENT EN SERIE ENTIERE
C** L ARITHMETIQUE UTILISEE EST L ARITHMETIQUE REELLE ETENDUE
C*****************************************************************************
      IMPLICIT INTEGER(A-Z)
      DIMENSION ER(30),X(30),S(30),T(30),A(30),Y(30),Z(30),RX(80)
      DIMENSION R(60),RZ(9,30),RY(60)
      COMMON/BLOK/MM,LS,LP,NN,NI,NJ
      CALL LONG(L,LI,LJ)
      J=1
      M=60
      READ(105,1) N
 1    FORMAT(I3)
      WRITE(NI,100) N,DEXP(DFLOAT(N))
100   FORMAT('EXP(',I3,')= ',D22.15,/,'********************')
      CALL LECR(ER,L,RX,LI)
      CALL VFL(X,N,L)
      CALL VFL(S,1,L)
      CALL VFL(T,1,L)
    5 CALL VFL(A,J,L)
      CALL DIVR(Y,X,A,L,M,RZ,RY,LJ)
      CALL MULR(Z,T,Y,L,R,LJ)
      DO 2 I=1,L
2     T(I)=Z(I)
      CALL ADDR(Z,S,T,L)
      DO 3 I=1,L
3     S(I)=Z(I)
      CALL DIVR(Z,T,S,L,M,RZ,RY,LJ)
      IF(MOD(J,10).NE.0) GO TO 4
      WRITE(NI,110) J
110   FORMAT('NOMBRE D ITERATION ',I4)
      WRITE(NI,200) N
200   FORMAT('LA VALEUR DE L EXPONENTIEL(',I4,') EST EGALE A')
      WRITE(NI,300)
300   FORMAT('LA PRECISION DU RESULTAT EST EGALE A ')
      CALL ECRR(Z,L,RX,LI)
      CALL ECRR(S,L,RX,LI)
4     J=J+1
      CALL COMR(Z,ER,L,N1)
      GO TO(5,6,6) N1
6     WRITE(NI,110) J
      WRITE(NI,200) N
      CALL ECRR(S,L,RX,LI)
      WRITE(NI,300)
      CALL ECRR(Z,L,RX,LI)
      END
```

```
      BLOCK DATA
      COMMON/BLOK/MM,LS,LP,NN,NI,NJ
      DATA MM,LS,LP,NN,NI,NJ/60,6,4,2147483647,108,105/
      END
```

Les résultats obtenus sont les suivants :

```
EXP(-10)=    .453999297624849D-04
********************
NOMBRE D ITERATION    10
LA VALEUR DE L EXPONENTIEL( -10) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,20525532448732146070854794830061727665265019106807645875 7799A+1
+ ,13425873015873015873015873015873015873015873015873016 1A+4
NOMBRE D ITERATION    20
LA VALEUR DE L EXPONENTIEL( -10) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,30681187858657917411715593625097511703089740629969926 1442443A+1
+ ,13396865995696041034514305564116001611337627485477138 6587563A+2
NOMBRE D ITERATION    30
LA VALEUR DE L EXPONENTIEL( -10) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,38852170274763634686262213776233819815609864904268883 6112556A+1
+ ,97034157993091446499644491281579269156726603595162363 318010A-3
NOMBRE D ITERATION    40
LA VALEUR DE L EXPONENTIEL( -10) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,26994586237384692330735779553099974417306421313677853 10341 69A-3
+ ,45402342097432605172373749618064401413527762646593669 5484520A-4
NOMBRE D ITERATION    50
LA VALEUR DE L EXPONENTIEL( -10) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,72421905359662837585131160482089116546297247267018442 1065351A-10
+ ,45399929763025296038488651071199597031482453954921531 8736630A-4
NOMBRE D ITERATION    57
LA VALEUR DE L EXPONENTIEL( -10) EST EGALE A
+ ,45399929762484872574095700167929000787149642276896078 7792247A-4
LA PRECISION DU RESULTAT EST EGALE A
+ ,30979619611919626405436142248771999095435923677527782 803964 6A-14
*STOP* 0
***************************************************************************
EXP(-15)=    .305902320501826D-06
********************
NOMBRE D ITERATION    10
LA VALEUR DE L EXPONENTIEL( -15) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,16946639142748375760459904943229731501946209504299946 2075643A+1
+ ,93770444754464285714285714285714285714285714285714286A+5
NOMBRE D ITERATION    20
LA VALEUR DE L EXPONENTIEL( -15) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,23723640318620151553736861121932856307951772694068257 2281470A+1
+ ,57612833258798712399184634748268096344168878445747087 940808A+5
NOMBRE D ITERATION    30
LA VALEUR DE L EXPONENTIEL( -15) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,30450975003968426451811929429521070159339983526828151 1058087A+1
+ ,23739769285748959451961705528654937742774380179728522 4620925A+3
NOMBRE D ITERATION    40
LA VALEUR DE L EXPONENTIEL( -15) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,37155954534264091489650497078109376457799110216040507 1371236A+1
+ ,36473452216884192060762236160403973914209338504625220 746254A-1
NOMBRE D ITERATION    50
LA VALEUR DE L EXPONENTIEL( -15) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,26740490208257938918754447582872039014165111088854379 3488353A+1
+ ,78400478949115040698264805678783702930158408461221948 924530A-6
```

```
NOMBRE D ITERATION   60
LA VALEUR DE L EXPONENTIEL( -15) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,144449712278738649088687119633721309602026252219095848852221A-4
+ ,305903194878632925687039946145831815730929040061976782161631A-6
NOMBRE D ITERATION   70
LA VALEUR DE L EXPONENTIEL( -15) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,578629174182214640037143901303363961007855311347189346246470A-11
+ ,305902320502135138934441460206525624728744293369163383428423A-6
NOMBRE D ITERATION   76
LA VALEUR DE L EXPONENTIEL( -15) EST EGALE A
+ ,305902320501825681203721160399071470045183243130618484927096A-6
LA PRECISION DU RESULTAT EST EGALE A
- ,212153369434002662599127395247063606729924178624348041692511A-14
*STOP* 0
*************************************************************************
EXP(-20)=    .206115362243856D-08
*********************
NOMBRE D ITERATION   10
LA VALEUR DE L EXPONENTIEL( -20) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,151744114559704868015781164091640330708960917788313559224244A+1
+ ,185962368077601410934744268077601410934744268077601410934744A+7
NOMBRE D ITERATION   20
LA VALEUR DE L EXPONENTIEL( -20) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,202563228110983421838446749447738667139008869554094704671427A+1
+ ,212772103425442991442230644084225022773281582978301171751390A+8
NOMBRE D ITERATION   30
LA VALEUR DE L EXPONENTIEL( -20) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,253047967237857065043860462759647273348393745231562085821782A+1
+ ,159969409642292840819821706548417333100762898991122583264048A+7
NOMBRE D ITERATION   40
LA VALEUR DE L EXPONENTIEL( -20) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,303370148757391620131634397481632405982104635725523890482148A+1
+ ,44420343631250907290303844701402614142099318617979240204794A+4
NOMBRE D ITERATION   50
LA VALEUR DE L EXPONENTIEL( -20) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,353600337053069262043599549932194546190829060589303405718193A+1
+ ,104691697206582172519553454225012748249033960628739156540163A+1
NOMBRE D ITERATION   60
LA VALEUR DE L EXPONENTIEL( -20) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,403748963203560500642969337461588924289553619643155177430959A+1
+ ,34317328370370852086555294015500240202658116594260646102791A-4
NOMBRE D ITERATION   70
LA VALEUR DE L EXPONENTIEL( -20) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,432599683492696679230120920408327000020166123108871701934351A+0
+ ,227828716461576819770424240855805268450839162878333303085844A-8
NOMBRE D ITERATION   80
LA VALEUR DE L EXPONENTIEL( -20) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,819523663231463110446278457431480966255008408064096573713120A-6
+ ,206115395757970008853979212876411970766353551366011541140743A-8
NOMBRE D ITERATION   90
LA VALEUR DE L EXPONENTIEL( -20) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,404253192012344516741089370191463664274038609077660392531467A-12
+ ,206115362243870820180949010372083191270792300035715765947405A-8
```

```
NOMBRE D ITERATION   94
LA VALEUR DE L EXPONENTIEL( -20) EST EGALE A
+ ,2061153622438556323675025604731365082561758652070671628429 81A-8
LA PRECISION DU RESULTAT EST EGALE A
- ,41536631784635349221999153424686510945811147637922686220789 3A-14
*STOP* 0
*************************************************************************
EXP(-25)=    .138879438649641D-10
********************
NOMBRE D ITERATION   10
LA VALEUR DE L EXPONENTIEL( -25) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,141191680805411211061917001221491975151338353199859140488402A+1
+ ,186134957973503637566137566137566137566137566137566137566139A+8
NOMBRE D ITERATION   20
LA VALEUR DE L EXPONENTIEL( -25) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,181822414675675708409463578370196144682500223084583370825235A+1
+ ,205602378984708412152396344512943782199892701273240493476998A+10
NOMBRE D ITERATION   30
LA VALEUR DE L EXPONENTIEL( -25) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,222218041106418165692096542458258108729058332765588922215481A+1
+ ,147150204620811990329275784082028956932010557566543968076386A+10
NOMBRE D ITERATION   40
LA VALEUR DE L EXPONENTIEL( -25) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,262490622945984291485457175600083787834830751587536957183068A+1
+ ,38622598120867503002514341337208736794560060861782421004457A+8
NOMBRE D ITERATION   50
LA VALEUR DE L EXPONENTIEL( -25) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,302690258513686433136049039985748387555544435896945745611087A+1
+ ,85689402951461437073067973062059663390876713157168767778290A+5
NOMBRE D ITERATION   60
LA VALEUR DE L EXPONENTIEL( -25) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,342842944173427826946897841116173379267315291274017510357920A+1
+ ,263712341566841937915877411778895079982920830944482975250053A+2
NOMBRE D ITERATION   70
LA VALEUR DE L EXPONENTIEL( -25) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,382963574441781622063662227927816570571017657243202396787352A+1
+ ,156400487825271174470186531145478943548296233572545952554156A-2
NOMBRE D ITERATION   80
LA VALEUR DE L EXPONENTIEL( -25) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,422801493618826183556764729000738801351694214506710292081546A+1
+ ,226118688781383881552570774705690105159718434767774191105320A-7
NOMBRE D ITERATION   90
LA VALEUR DE L EXPONENTIEL( -25) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,314102666359178693810737228199527502673370768213068077321432A-1
+ ,139827749367555997599705532371259521764670129072714774484684A-10
NOMBRE D ITERATION  100
LA VALEUR DE L EXPONENTIEL( -25) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,480130456646649595280132627962079519222665770896294330127102A-7
+ ,138879439974737640663824764452231710755665190034821354717348A-10
NOMBRE D ITERATION  110
LA VALEUR DE L EXPONENTIEL( -25) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,269057657415659251764886114885828128291595515831346832363773A-13
+ ,138879438649640893757559900113446462912288477326709229593945A-10
```

```
NOMBRE D ITERATION  112
LA VALEUR DE L EXPONENTIEL( -25) EST EGALE A
+ ,13887943864964005216800074979483564236876716731754045001244 8A-10
LA PRECISION DU RESULTAT EST EGALE A
- ,6059857149001371221077813034467275106372442400575241622429 32A-14
*STOP* 0
*************************************************************************
EXP(-30)=   .935762296884018D-13
********************
NOMBRE D ITERATION   10
LA VALEUR DE L EXPONENTIEL( -30) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,1341993453134900064268062788892495394944127047680839326712 66A+1
+ ,1212548495714285714285714285714285714285714285714285 72A+9
NOMBRE D ITERATION   20
LA VALEUR DE L EXPONENTIEL( -30) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,1680326376474655589412330128329385691003917041210895082414 22A+1
+ ,8529171221003217525508547160373805760802664827432629290214 5A+11
NOMBRE D ITERATION   30
LA VALEUR DE L EXPONENTIEL( -30) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,2016946657141051073919836882713031353558183978550909157519 70A+1
+ ,3848426125359078982868499424944262999227942476803411731908 78A+12
NOMBRE D ITERATION   40
LA VALEUR DE L EXPONENTIEL( -30) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,2352614585217424442260540612891028005350376590756155997267 21A+1
+ ,6333654011726656641586838055330222294982026982191580601229 3A+11
NOMBRE D ITERATION   50
LA VALEUR DE L EXPONENTIEL( -30) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,2687694956104005067649046825033405516863443133790916909678 65A+1
+ ,8782292292779837768969166417001690712946424066450001893698 6A+9
NOMBRE D ITERATION   60
LA VALEUR DE L EXPONENTIEL( -30) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,3022386190189725319141423599981712905613534133810426156155 97A+1
+ ,1685584300368905491613563475514643837524672364484977462365 00A+7
NOMBRE D ITERATION   70
LA VALEUR DE L EXPONENTIEL( -30) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,3356805947736268357149657165261387876696051456118087242264 99A+1
+ ,6225246584415179282158943590439048962767600754318692892750 1A+3
NOMBRE D ITERATION   80
LA VALEUR DE L EXPONENTIEL( -30) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,3691028594561736001304383708684498172203046117290641175592 00A+1
+ ,5595324073915949191907075755000026558297376890300949302486 3A-1
NOMBRE D ITERATION   90
LA VALEUR DE L EXPONENTIEL( -30) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,4025103250938641545740474203293324011570327734402259074383 25A+1
+ ,1459486591763925280826944930412291986572844552279301939451 19A-5
NOMBRE D ITERATION  100
LA VALEUR DE L EXPONENTIEL( -30) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,4327102772847900218921919667582562424200878029156334271355 63A+1
+ ,1276217239854213881462948173391759266726411143564017684083 00A-10
NOMBRE D ITERATION  110
LA VALEUR DE L EXPONENTIEL( -30) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,2046752733926692868003000008114989299174913048664356853715 30A-2
+ ,9361705933963401540098077930292788628712569054667219705516 05A-13
```

```
NOMBRE D ITERATION  120
LA VALEUR DE L EXPONENTIEL( -30) EST EGALE A
LA PRECISION DU RESULTAT EST EGALE A
+ ,28707222539955033785275005863515673494427005880388661113588 9A-8
+ ,935762297418422481055400442714238786385879480904025155254977A-13
NOMBRE D ITERATION  130
LA VALEUR DE L EXPONENTIEL( -30) EST EGALE A
+ ,935762296884016125739611402225401495528026769612416357841901A-13
LA PRECISION DU RESULTAT EST EGALE A
- ,759847628023963601599168864774384586103638732040078385923559A-14
*STOP* 0
***********************************************************************
```

Arithmétique rationnelle étendue

Le programme du calcul du développement en série entière de e^x utilise, en plus des sous-programmes de l'arithmétique rationnelle donnés au chapitre II, le sous-programme CØNV (A, C, K, K). Ce sous-programme transforme le rationnel donné dans le tableau A de dimension L en un nombre réel C et délivre dans K le nombre de chiffres du rationnel A. Ce sous-programme est le suivant :

```
      SUBROUTINE CONV(A,C,K,L)
C**********************************************************************************
C** CONVERSION D'UN TEP -A(L)- EN UN REEL DOUBLE PRECISION -C-.
C**A(L) : TABLEAU CONTENANT L'ENTIER A CONVERTIR EN DOUBLE PRECISION
C** C : REEL DOUBLE PRECISION DANS LEQUEL ON CONVERTIT A(L)
C** D : REEL QU'ON DIVISE A CHAQUE ETAPE PAR 10 ET QUI PERMET DE REMPLIR -C-
C** CHIFFRE PAR CHIFFRE
C**********************************************************************************
      IMPLICIT DOUBLE PRECISION (C,D)
      IMPLICIT LOGICAL (E)
      IMPLICIT INTEGER (A-B,F-Z)
      DIMENSION A(L)
      COMMON/BLOK1/IBASE,LM,IN
      D=1.D0
C*************************************************
C** RECHERCHE DU PREMIER MOT NON NUL DANS -A(L)-.
C** SI TOUT A(L) EST NUL ,C=0
C*************************************************
      DO 2 I=2,L
      IF(A(I).NE.0) GO TO 3
2     CONTINUE
      C=0
      K=0
      RETURN
C**********************************************************************************
C** CAS OU LE TEP -A(L)- EST NON NUL
C** VIDAGE DU PREMIER MOT NON NUL DE -A- DANS -C- EN COMMENCANT PAR LA RECHERCHE
C** DU PREMIER CHIFFRE NON NUL
C** B: EST UN BOULEEN QUI DEVIENT VRAI DES QUE L'ON RENCONTRE LE PREMIER CHIFFRE
C** NON NUL-Q-,DANS LE PREMIER MOT NON NUL DE -A-,ET DANS CE CAS ON COMMENCE A
C**METTRE CES CHIFFRES DANS-C-,SINON ON DECREMENTE-K-DE 1 ET ON RECOMMENCE CETTE
C** RECHERCHE
C**********************************************************************************
```

```
 3      K=LOGBAS(IBASE)*(L-I+1)
        II=I
        C=0.D0
        E=.FALSE.
        X=A(I)
        BAS=IBASE/10
 9      Q=X/BAS
        R=MOD(X,BAS)
        IF(Q.NE.0) GO TO 7
        IF(E) GO TO 6
        K=K-1
        GO TO 8
 7      E=.TRUE.
 6      D=D/10
        C=C+D*Q
 8      BAS=BAS/10
        X=R
        IF(BAS.NE.0) GO TO 9
C***********************************************************************************
C** VIDAGE DU RESTE DES MOTS DE-A(L)-DANS-C-PAR LA MEME TECHNIQUE QUE POUR LE
C** PREMIER MOT NON NUL
C***********************************************************************************
        IF(I.GE.L) GO TO 11
        II=I+1
        DO 5 I=II,L
        X=A(I)
        BAS=IBASE/10
 91     Q=X/BAS
        R=MOD(X,BAS)
        D=D/10
        C=C+D*Q
        BAS=BAS/10
        X=R
        IF(BAS.NE.0) GO TO 91
 5      CONTINUE
C****************************************************
C** NORMALISATION DU REEL -C-.
C****************************************************
 11     IF(A(1).LT.0) C=-C
        RETURN
        END
```

Le programme du calcul de e^x en rationnel utilise des rationnels de 212 chiffres. Ce programme est le suivant :

```
C******************************************************************************
C** CALCUL DE L EXPONENTIEL DE X PAR LE DEVELOPPEMENT EN SERIE ENTIERE
C** L ARITHMETIQUE ETILISEE EST L ARITHMETIQUE RATIONNELLE ETENDUE
C******************************************************************************
        IMPLICIT INTEGER(A-B,D-Z)
        IMPLICIT DOUBLE PRECISION(C)
        DIMENSION INA(54),IDA(54),NS(54),DS(54),NT(54),DT(54),A(54),X(54)
        DIMENSION N1(54),N2(54),N3(54),N4(54),N5(54),N6(54),N7(54),N8(54)
        DIMENSION TEL(240)
        COMMON/BLOK1/IBASE,LM,IN,KN
        ITA=0
        CALL INI(L,LR)
        J=1
        READ(KN,1) N
    1   FORMAT(I3)
        WRITE(108,200) N,DEXP(DFLOAT(N))
  200   FORMAT('X= ',I4,'EXP(X)= ',D22.15,/,'********************')
```

```
      CALL STO(N,X,L)
      CALL STO(1,NS,L)
      CALL STO(1,DS,L)
      CALL STO(1,NT,L)
      CALL STO(1,DT,L)
 5    CALL STO(J,A,L)
      CALL MULFRA(INA,IDA,NT,DT,X,A,N1,N2,N3,N4,N5,N6,N7,N8,L,ITA)
      IF(ITA.NE.0) GO TO 2
      DO 3 I=1,L
      NT(I)=INA(I)
 3    DT(I)=IDA(I)
      CALL ADDFRA(INA,IDA,NS,DS,NT,DT,N1,N2,N3,N4,N5,N6,N7,L,ITA)
      IF(ITA.NE.0) GO TO 2
      DO 4 I=1,L
      NS(I)=INA(I)
   4  DS(I)=IDA(I)
      CALL CONV(NT,C4,K1,L)
      CALL CONV(DT,C5,K2,L)
      C6=C4/C5
      K=K1-K2
      C6=C6*10**K
      CALL CONV(NS,C1,K3,L)
      CALL CONV(DS,C2,K4,L)
      C3=C1/C2
      K=K3-K4
      C3=C3*10**K
      IF(MOD(J,10).NE.0) GOTO 6
      WRITE(IN,220) J
 220  FORMAT('NOMBRE D ITERATION ',I4)
      WRITE(IN,230) N
 230  FORMAT('LA VALEUR DE L EXPONENTIEL(',I4,') EN RATIONNELLE')
      CALL ECR(NS,L,TEL,LR)
      CALL ECR(DS,L,TEL,LR)
      WRITE(IN,100) C3
 100  FORMAT('CETTE VALEUR EST EGALE EN REELLE A ',D22.15)
 6    J=J+1
      IF(DABS(C6/C3).GT.1.E-14) GO TO 5
 2    WRITE(IN,220) J
      WRITE(IN,230) N
      CALL ECR(NS,L,TEL,LR)
      CALL ECR(DS,L,TEL,LR)
      WRITE(IN,100) C3
      END
```

```
      BLOCK DATA
      COMMON/BLOK1/IBASE,LM,IN,KN
      DATA IBASE,LM,IN,KN/10000,212,108,105/
      END
```

Les résultats numériques obtenus sont les suivants :

```
X=  -10EXP(X)=    .453999297624849D-04
*********************
NOMBRE D ITERATION   10
LA VALEUR DE L EXPONENTIEL( -10) EN RATIONNELLE
+ 84583
+ 63
CETTE VALEUR EST EGALE EN REELLE A   .134258730158730D+04
NOMBRE D ITERATION   20
LA VALEUR DE L EXPONENTIEL( -10) EN RATIONNELLE
+ 198933485011
+ 14849255421
CETTE VALEUR EST EGALE EN REELLE A   .133968659956960D+02
NOMBRE D ITERATION   30
LA VALEUR DE L EXPONENTIEL( -10) EN RATIONNELLE
+ 4909246045519921l
+ 5059296795123883412l
CETTE VALEUR EST EGALE EN REELLE A   .970341579930914D-03
NOMBRE D ITERATION   40
LA VALEUR DE L EXPONENTIEL( -10) EN RATIONNELLE
+ 7666743632104801503027169
+ 16886229383612211676637564880 9
CETTE VALEUR EST EGALE EN REELLE A   .454023420974326D-04
NOMBRE D ITERATION   50
LA VALEUR DE L EXPONENTIEL( -10) EN RATIONNELLE
+ 30912693857768150013103748116930184 63
+ 680897393875356379511618560602404196 09243
CETTE VALEUR EST EGALE EN REELLE A   .453999297630253D-04
NOMBRE D ITERATION   57
LA VALEUR DE L EXPONENTIEL( -10) EN RATIONNELLE
+ 293578395463252801997640680440455345795371 5639
+ 646649448576557417121917430871440001043536521 02979
CETTE VALEUR EST EGALE EN REELLE A   .453999297624848D-04
*STOP* 0
*****************************************************************************
X=  -15EXP(X)=    .305902320501826D-06
*********************
NOMBRE D ITERATION   10
LA VALEUR DE L EXPONENTIEL( -15) EN RATIONNELLE
+ 168036637
+ 1792
CETTE VALEUR EST EGALE EN REELLE A   .937704447544643D+05
NOMBRE D ITERATION   20
LA VALEUR DE L EXPONENTIEL( -15) EN RATIONNELLE
+ 341817107714256 49
+ 593300291584
CETTE VALEUR EST EGALE EN REELLE A   .576128332587987D+05
NOMBRE D ITERATION   30
LA VALEUR DE L EXPONENTIEL( -15) EN RATIONNELLE
+ 168519038462050571152961
+ 7098596301995780341 76
CETTE VALEUR EST EGALE EN REELLE A   .237397692857490D+03
NOMBRE D ITERATION   40
LA VALEUR DE L EXPONENTIEL( -15) EN RATIONNELLE
+ 3932867503412468540159199890026 1
+ 107828222018201943879367972723097 6
CETTE VALEUR EST EGALE EN REELLE A   .364734522168842D-01
NOMBRE D ITERATION   50
LA VALEUR DE L EXPONENTIEL( -15) EN RATIONNELLE
+ 44461887185856060639859461207372128821 9
+ 5671124434675144554309615306755230368048087 04
CETTE VALEUR EST EGALE EN REELLE A   .784004789491150D-06
NOMBRE D ITERATION   60 .
LA VALEUR DE L EXPONENTIEL( -15) EN RATIONNELLE
+ 18229874101303841490071322068406816602541990776449289
+ 5959360479558424668259567532453435686337875113375015790 1824
CETTE VALEUR EST EGALE EN REELLE A   .305903194878632D-06
```

```
NOMBRE D ITERATION   70
LA VALEUR DE L EXPONENTIEL( -15) EN RATIONNELLE
+ 1295948032636456641984000413433397003487162736901496819782791745693З
+ 4236476632505346345619129759555335526876814107623177953070226926З357
411328
CETTE VALEUR EST EGALE EN REELLE A   .305902320502135D-06
NOMBRE D ITERATION   76
LA VALEUR DE L EXPONENTIEL( -15) EN RATIONNELLE
+ 4970505003329490258117525133470937845979359823294583732586342125З236
37311
+ 1624866720584365548091452266239372498649387036466008918476460975780б
628255563776
CETTE VALEUR EST EGALE EN REELLE A   .305902320501826D-06
*STOP* 0
**************************************************************************
X=  -20EXP(X)=   .206115362243856D-08
********************
NOMBRE D ITERATION   10
LA VALEUR DE L EXPONENTIEL( -20) EN RATIONNELLE
+ 1054406627
+ 567
CETTE VALEUR EST EGALE EN REELLE A   .185962368077601D+07
NOMBRE D ITERATION   20
LA VALEUR DE L EXPONENTIEL( -20) EN RATIONNELLE
+ 16628985843304379
+ 781539759
CETTE VALEUR EST EGALE EN REELLE A   .212772103425443D+08
NOMBRE D ITERATION   30
LA VALEUR DE L EXPONENTIEL( -20) EN RATIONNELLE
+ 809332721521111822З0927901
+ 5059296795123883412l
CETTE VALEUR EST EGALE EN REELLE A   .159969409642293D+07
NOMBRE D ITERATION   40
LA VALEUR DE L EXPONENTIEL( -20) EN RATIONNELLE
+ 675082900670562576704018329578186l
+ 1519760644525099050897380839281
CETTE VALEUR EST EGALE EN REELLE A   .444203436312509D+04
NOMBRE D ITERATION   50
LA VALEUR DE L EXPONENTIFL( -20) EN RATIONNELLE
+ 926695949248546473626767662726239289956779
+ 885166612037963293365104128783125454920159
CETTE VALEUR EST EGALE EN REELLE A   .104691697206582D+01
NOMBRE D ITERATION   60
LA VALEUR DE L EXPONENTIEL( -20) EN RATIONNELLE
+ 6492747322990234167818305933613718686234249405706l9
+ 18919734231397774565894772364179678694531895885941698799
CETTE VALEUR EST EGALE EN REELLE A   .343173283703708D-04
NOMBRE D ITERATION   70
LA VALEUR DE L EXPONENTIEL( -20) EN RATIONNELLE
+ 1211947257489963557908207391838666554575313183157566253337771
+ 53195544280492742732814476099574724161462571736731179050907071757539
1
CETTE VALEUR EST EGALE EN REELLE A   .227828716461576D-08
NOMBRE D ITERATION   80
LA VALEUR DE L EXPONENTIEL( -20) EN RATIONNELLE
+ 82548030705273495330185495439661350087276118315005940151558527493961
5491
+ 40049424935828236215989812259559282872983629276474537358477611138776
741244991151l
CETTE VALEUR EST EGALE EN REELLE A   .206115395757970D-08
NOMBRE D ITERATION   90
LA VALEUR DE L EXPONENTIEL( -20) EN RATIONNELLE
+ 83003505384752439600300847653034992543556569115529233358216576624352
0757809510976488992 21
+ 40270411909688058955012779209547625199169166169051971958720255616634
4257755191823212727442402558 41
CETTE VALEUR EST EGALE EN REELLE A   .206115362243871D-08
NOMBRE D ITERATION   94
LA VALEUR DE L EXPONENTIEL( -20) EN RATIONNELLE
+ 16156549319635487104511599769481560826361074262176864974394349842335З
5071986863502062605844 66429
```

```
+ 78385954078088709875342824603592356973930790256397972897929390355222
74342779033319653418393621559194809
CETTE VALEUR EST EGALE EN REELLE A    .2061153622438550-08
*STOP* 0
**************************************************************************
X=  -25EXP(X)=    .1388794386496410-10
********************
NOMBRE D ITERATION    10
LA VALEUR DE L EXPONENTIEL( -25) EN RATIONNELLE
+ 900595380659
+ 48384
CETTE VALEUR EST EGALE EN REELLE A    .1861349579735040+08
NOMBRE D ITERATION    20
LA VALEUR DE L EXPONENTIEL( -25) EN RATIONNELLE
+ 8003367051484667223102649
+ 3892643213082624
CETTE VALEUR EST EGALE EN REELLE A    .2056023789847080+10
NOMBRE D ITERATION    30
LA VALEUR DE L EXPONENTIEL( -25) EN RATIONNELLE
+ 384315200961176337780651868610983313
+ 261172046584311196195749888
CETTE VALEUR EST EGALE EN REELLE A    .1471502046208120+10
NOMBRE D ITERATION    40
LA VALEUR DE L EXPONENTIEL( -25) EN RATIONNELLE
+ 316363475682457916099804419051141964732655441839
+ 8191149510253404705587313137906104664064
CETTE VALEUR EST EGALE EN REELLE A    .3862259812086750+08
NOMBRE D ITERATION    50
LA VALEUR DE L EXPONENTIEL( -25) EN RATIONNELLE
+ 344350123602745765482830549610373466126723625913477819429967
+ 4018584701748967627956040098329073973760349027950395392
CETTE VALEUR EST EGALE EN REELLE A    .8568940295146140+05
NOMBRE D ITERATION    60
LA VALEUR DE L EXPONENTIEL( -25) EN RATIONNELLE
+ 11597477798937354942402231231573775455990115978647914834129348571820
38319
+ 4397758985533850362523733530945678069121407736329888285460956702572
544
CETTE VALEUR EST EGALE EN REELLE A    .2637123415668420+02
NOMBRE D ITERATION    70
LA VALEUR DE L EXPONENTIEL( -25) EN RATIONNELLE
+ 37205670002355638401968696878226762236406569575835566472638973517063
97035349498010281
+ 23788717362519600871134119817715659842199180912421895909113583657462
46169750652432941056
CETTE VALEUR EST EGALE EN REELLE A    .1564004878252710-02
NOMBRE D ITERATION    80
LA VALEUR DE L EXPONENTIEL( -25) EN RATIONNELLE
+ 84846532541005149892581890064440816713945628834284721721066422044084
245002130030044028934553223029
+ 37523007495871556377542149392662020576030048828714255687147556730687
73795717717627642521576657692345237504
CETTE VALEUR EST EGALE EN REELLE A    .2261186887813840-07
NOMBRE D ITERATION    90
LA VALEUR DE L EXPONENTIEL( -25) EN RATIONNELLE
+ 33513185781168105094432002730821432702830221416659240284807552894401
28233557049417495307138178810835317022962273
+ 23967478510345039135428155537372168752755375503446981669003107539635
6732853692803163860037714516326727225036500244887502848
CETTE VALEUR EST EGALE EN REELLE A    .1398277493675560-10
NOMBRE D ITERATION   100
LA VALEUR DE L EXPONENTIEL( -25) EN RATIONNELLE
```

```
+ 72483683023128030543298300415302587056686041596445768620445938020510
692184303228638661752354605122387441244018283815268482656468 83
+ 52191802498652898049261579244239312924668259807355852326960712378342
23473626722772776608241613989190321427069388230772241706388465192617 90
208
CETTE VALEUR EST EGALE EN REELLE A    .138879439974737D-10
NOMBRE D ITERATION  110
LA VALEUR DE L EXPONENTIEL( -25) EN RATIONNELLE
+ 87073612576441599822205552028879773087020986134919128445467627368319
75068324991472788392351382184763588761353696966244851005115865415824 81
5219685489
+ 62697267085091828762333104979547611311967228935617321464294548046160
30289865620318069410317126129420156251637940800836296617514141723718 36
70825420236538668646 4
CETTE VALEUR EST EGALE EN REELLE A    .138879438649641D-10
NOMBRE D ITERATION  112
LA VALEUR DE L EXPONENTIEL( -25) EN RATIONNELLE
+ 82153953465872151591028709447911494382479998008356950468125208728639
03668031711136985175900461561852887411331504151996284206685839162832 98
4811994938559
+ 59154871494784140437261284548203171272841080500754942801561906081552
24578488212770098488634208503107917423420397145589045858624592716328 27
9342378399317423386787 84
CETTE VALEUR EST EGALE EN REELLE A    .138879438649640D-10
*STOP* 0
************************************************************************
X=  -30EXP(X)=    .935762296884018D-13
*********************
NOMBRE D ITERATION   10
LA VALEUR DE L EXPONENTIEL( -30) EN RATIONNELLE
+ 848783947
+ 7
CETTE VALEUR EST EGALE EN REELLE A    .121254849571429D+09
NOMBRE D ITERATION   20
LA VALEUR DE L EXPONENTIEL( -30) EN RATIONNELLE
+ 27576772266884233
+ 323323
CETTE VALEUR EST EGALE EN REELLE A    .852917122100321D+11
NOMBRE D ITERATION   30
LA VALEUR DE L EXPONENTIEL( -30) EN RATIONNELLE
+ 140371114334042254434091
+ 364749406021
CETTE VALEUR EST EGALE EN REELLE A    .384842612535908D+12
NOMBRE D ITERATION   40
LA VALEUR DE L EXPONENTIEL( -30) EN RATIONNELLE
+ 24845454425774272769179603714310 9
+ 39227678650859868459 29
CETTE VALEUR EST EGALE EN REELLE A    .633365401172666D+11
NOMBRE D ITERATION   50
LA VALEUR DE L EXPONENTIEL( -30) EN RATIONNELLE
+ 79910456894931576058148872043835813964 1
+ 9099043191790387579109849693 21
CETTE VALEUR EST EGALE EN REELLE A    .878229229277984D+09
NOMBRE D ITERATION   60
LA VALEUR DE L EXPONENTIEL( -30) EN RATIONNELLE
+ 820014355611918149144722408067294919841476359 67
+ 4864667070916895294418671466369350657412 7
CETTE VALEUR EST EGALE EN REELLE A    .168558430036891D+07
NOMBRE D ITERATION   70
LA VALEUR DE L EXPONENTIEL( -30) EN RATIONNELLE
+ 17871115616038306348330865956073632784153229013956923722 1
+ 287074823040398156198926935393483755203483580099571551
CETTE VALEUR EST EGALE EN REELLE A    .622524658441518D+03
NOMBRE D ITERATION   80
LA VALEUR DE L EXPONENTIEL( -30) EN RATIONNELLE
+ 462826325700576287672699434724786197185415241539275909364233589 1
+ 8271662545126938198586249945875134660551191171026023047493445932 1
CETTE VALEUR EST EGALE EN REELLE A    .559532407391595D-01
```

```
NOMBRE D ITERATION   90
LA VALEUR DE L EXPONENTIEL( -30) EN RATIONNELLE
+ 85261493085310419395851566852285684962312682412681828553486479179059
33
+ 58418825884699615870812761154235144922560354740956595688249047728978
12286023
CETTE VALEUR EST EGALE EN REELLE A    .145948659176392D-05
NOMBRE D ITERATION  100
LA VALEUR DE L EXPONENTIEL( -30) EN RATIONNELLE
+ 15762221228707717171912811698153637328440546926797635047312667338126
93876729
+ 12350735232591187157565034046620738336717629024995532228509589689269
4392851198045763399
CETTE VALEUR EST EGALE EN REELLE A    .127621723985421D-10
NOMBRE D ITERATION  110
LA VALEUR DE L EXPONENTIEL( -30) EN RATIONNELLE
+ 12690434222659381057521332456220801931941028636850935332578204120998
95387778336060564723043
+ 13555685589973149843516652773924401248412729534904139947652529701667
987661818503269300797629457221610933
CETTE VALEUR EST EGALE EN REELLE A    .936170593396339D-13
NOMBRE D ITERATION  120
LA VALEUR DE L EXPONENTIEL( -30) EN RATIONNELLE
+ 39038607048314977688494742477024319627724979602024819656425016121889
74352199307153326439863943530828059
+ 41718508168168925483903645272695878351226021541115168743874241633925
206101609735964939174674397484981597313154300629
CETTE VALEUR EST EGALE EN REELLE A    .935762297418421D-13
NOMBRE D ITERATION  130
LA VALEUR DE L EXPONENTIEL( -30) EN RATIONNELLE
+ 15399902469730047591808827289245987361661599325850463473990603736152
441387657743455844216974745032099606819210687823
+ 16457066630072617617469585745800256959118805942012232979962150731473
268266910359201619908922229837670200920986644640322152290863
CETTE VALEUR EST EGALE EN REELLE A    .935762296884014D-13
*STOP* 0
**********************************************************************
```

Conclusion

L'exemple du calcul d'une série entière montre que l'utilisation de l'arithmétique rationnelle est peu performante car les entiers intervenants dans le calcul deviennent très vite assez grands ce qui nécessite des tableaux de dimension très grande et un temps de calcul très long. Le tableau suivant donne une comparaison du temps de calcul utilisé par les trois arithmétiques étudiées :

Type d'arithmétique	Temps de calcul en minute				
	e^{-10}	e^{-15}	e^{-20}	e^{-25}	e^{-30}
arithmétique de l'ordinateur	0,002	0,003	0,003	0,003	0,003
arithmétique réelle	0,268	0,334	0,429	0,499	0,614
arithmétique rationnelle	0,508	1,028	1,735	3,815	2,858

Les temps de calcul pour l'arithmétique de l'ordinateur ne sont donnés qu'à titre indicatif puisque les résultats obtenus sont faux.

CHAPITRE V

ARITHMETIQUE P-ADIQUE

I - INTRODUCTION

On étudie dans ce chapitre une nouvelle arithmétique exacte : l'arithmétique p-adique tronquée.

On commence par donner une construction algébrique du corps des nombres p-adiques ainsi que certaines de leurs propriétés. Ensuite on donne une définition axiomatique de ce corps ainsi que l'arithmétique p-adique.

Dans la deuxième partie on étudie l'ensemble des nombres p-adiques tronqués ainsi que l'arithmétique sur cet ensemble qui a été présenté en 1975 par les auteurs KRISHNAMURTHY, MAHADEVA RAO et SUBRAMANIAN [16].

Enfin on présente les problèmes que pose une telle arithmétique pour son utilisation pratique dans les algorithmes.

II - LE CORPS DES NOMBRES P-ADIQUES

II-1 DEFINITION ALGEBRIQUE [11]

On définit dans ce paragraphe les entiers p-adiques ainsi que les nombres p-adiques en étudiant certaines de leurs propriétés.

Notation : Soient x et r deux entiers et m un entier positif. Si m divise (r-x) on dit que r est congru x modulo m et on le note par $r \equiv x \bmod (m)$ ou par $r = |x|_m$ avec $r \in]0, m[$.

Définition 1 : Soit $\{x_n\}$ une suite d'entiers relatifs tels que x_o arbitraire.

$$x_n \equiv x_{n-1} \mod(p^n) \quad n = 1, 2, \ldots$$

l'objet mathématique défini par cette suite et noté $x = \{x_n\}$ s'appelle un entier p-adique. L'ensemble des entiers p-adiques sera noté par $\mathbb{Z}_p$.

Propriété 1 : Deux suites $\{x_n\}$ et $\{y_n\}$ d'entiers relatifs définissent le même entier p-adique si et seulement si

$$x_{n-1} \equiv y_{n-1} \mod (p^n) \quad n = 1, 2, \ldots$$

Pour tout élément x de $\mathbb{Z}$ on lui associe l'entier p-adique (x, x, ...) qu'on désigne également par x.

Exemple 1 : Soit les ensembles $\mathbb{Z}/_{p^n}\mathbb{Z}$, n = 1, 2, ..., définis par les classes d'équivalences modulo p^n.

En prenant p = 2 et en notant $\bar{x}$ la classe d'équivalence de x modulo p^n, on obtient :

$$\mathbb{Z}/2\mathbb{Z} = \{\bar{0}, \bar{1}\}$$
$$\mathbb{Z}/4\mathbb{Z} = \{\bar{0}, \bar{1}, \bar{2}, \bar{3}\}$$
$$\mathbb{Z}/8\mathbb{Z} = \{\bar{0}, \bar{1}, \bar{2}, \bar{3}, \bar{4}, \bar{5}, \bar{6}, \bar{7}\}$$
$$\mathbb{Z}/16\mathbb{Z} = \{\overline{0}, \overline{1}, \overline{2}, \overline{3}, \overline{4}, \overline{5}, \overline{6}, \overline{7}, \overline{8}, \overline{9}, \overline{10}, \overline{11}, \overline{12}, \overline{13}, \overline{14}, \overline{15}\}$$

et ceci pour n allant jusqu'à l'infini.

D'après la définition 1 la suite (1, 3, 7, 15, ...) est un entier p-adique puisque la relation de congruence est vérifiée entre deux termes successifs :

$$3 \equiv 1 \mod (2)$$
$$7 \equiv 3 \mod (4)$$
$$15 \equiv 7 \mod (8)$$

D'après la propriété 1 en choisissant d'autres représentants des classes d'équivalence $\{\overline{1}, \overline{3}, \overline{7}, \overline{15}, ...\}$ on trouve le même entier p-adique. Prenons par exemple la suite (-9, 11, 39, -17, ...), la propriété 1 est vérifiée puisque

$$1 \equiv -9 \mod (2)$$
$$3 \equiv 11 \mod (4)$$
$$7 \equiv 39 \mod (8)$$
$$15 \equiv -17 \mod (16)$$

on en déduit que les deux suites (1, 3, 7, 15, ...) et (-9, 11, 39, -17,...) définissent le même entier p-adique.

Définition 2 : On appelle respectivement somme et produit des entiers p-adiques $x = \{x_n\}$ et $y = \{y_n\}$ les entiers p-adiques définis par les suites $\{x_n + y_n\}$ et $\{x_n\, y_n\}$.

Muni de ces deux lois l'ensemble $\mathbb{Z}_p$ est un anneau commutatif.

On dit que x est divisible par y dans $\mathbb{Z}_p$ s'il existe un entier p-adique z tel que x = yz. y et z s'appellent les diviseurs de x.

Un diviseur du nombre 1 s'appelle une unité, d'où :

Propriété 2 : L'entier p-adique $x = \{x_o, x_1, x_2, \ldots\}$ est une unité si et seulement si $x_o \not\equiv 0 \bmod (p)$.

Il résulte de cette propriété que l'entier p-adique $x = \{x, x, \ldots\}$ est une unité si et seulement si $x \not\equiv 0 \bmod (p)$, et dans ce cas x^{-1}, appartient à $\mathbb{Z}_p$. Donc tout nombre rationnel dont le dénominateur n'est pas divisible par p appartient à $\mathbb{Z}_p$.

Théorème 1 : Tout entier p-adique non nul peut s'écrire de façon unique

$$x = p^n \, U$$

aven $n \in \mathbb{N}$ et U unité de $\mathbb{Z}_p$.

Démonstration :

Si x est une unité la démonstration est terminée.

Supposons donc que x ne soit pas une unité c'est à dire que $x_o \equiv 0 \bmod (p)$. D'après la propriété 1 la suite $\{0, x_1, x_2, x_3, \ldots\}$ définit le même entier p-adique. Il en est de même de la suite $\{0, 0, x_2, x_3, \ldots\}$ si $x_1 \equiv 0 \bmod (p^2)$ et ainsi de suite. Puisque $x \neq 0$ les congruences $x_k \equiv 0 \bmod (p^{k+1})$ ne peuvent pas avoir lieu pour tout k. Soit donc n le plus petit entier tel que $x_n \not\equiv 0 \bmod (p^{n+1})$ on a

$$x_{n+k} \equiv x_{n-1} \equiv 0 \bmod (p^n) \text{ et ceci } \forall k$$

et donc

$$U_k = \frac{x_{n+k}}{p^n} \text{ est un entier.}$$

Or $p^n U_k - p^n U_{k-1} = x_{n+k} - x_{n+k-1} \equiv 0 \mod (p^{n+k})$

et donc $U_k \equiv U_{k-1} \bmod (p^k) \quad \forall k \geq 0$

on en déduit que $U = \{U_k\}$ est un entier p-adique et que $U_o = \frac{x_n}{p^n} \not\equiv 0 \bmod (p)$.

Par conséquent, U est une unité et la congruence $p^n U_k = x_{n+k} \equiv x_k \bmod (p^{k+1})$ entraine que $x = p^n U$.

Montrons maintenant l'unicité. Pour cela on suppose que l'on ait également $x_n = p^k V$. Alors d'après la propriété 1 on a :

$$p^k V_{i-1} = p^n U_{i-1} \bmod (p^i) \text{ pour } i = 1, 2, \ldots$$

Comme U et V sont des unités les U_i et V_i ne sont pas divisibles par p. Posons $i = n+1$ dans la congruence précédente ; il vient :

$$p^k \mathcal{V}_n \equiv p^n \mathcal{U}_n \not\equiv 0 \mod (p^{n+1})$$

et donc $k \leq n$. Par symétrie $n \leq k$ et donc $k = n$. La congruence précédente s'écrit alors :

$$\mathcal{V}_{i-1} \equiv \mathcal{U}_{i-1} \mod (p^{i-n}) \quad i = 1, 2, \ldots$$

ou encore

$$\mathcal{V}_{i+n-1} \equiv \mathcal{U}_{i+n-1} \text{ pod } (p^i) \quad i = 1, 2, \ldots$$

mais $\mathcal{V}_{i+n-1} \equiv \mathcal{V}_{i-1} \mod (p^i)$ et $\mathcal{U}_{i+n-1} \equiv \mathcal{U}_{i-1} \mod (p^i)$

et donc $\mathcal{U}_{i-1} \equiv \mathcal{V}_{i-1} \mod (p^i)$ $i = 1, 2, \ldots$

ce qui prouve que $\mathcal{U} = \mathcal{V}$.

Propriété 3 : L'anneau $\mathbb{Z}_p$ est intègre

Démonstration:

Soient x et y deux entiers p-adiques non nuls. D'après la propriété 2 on a $x = p^n u$ et $y = p^k v$ où u et v sont des unités de $\mathbb{Z}_p$. Si $xy = 0$ alors $p^{n+k}\, \mathcal{U}\mathcal{V} = 0$ et donc en multipliant par u^{-1} et v^{-1} on trouve $p^{n+k} = 0$ ce qui est impossible.

Puisque $\mathbb{Z}_p$ est intègre on peut construire par symétrisation, son corps des fractions et on a :

Définition 3 : Le corps des fractions de $\mathbb{Z}_p$ s'appelle le corps des nombres p-adiques. On le désigne par $\mathbb{Q}_p$.

Théorème 2 : Tout nombre p-adique non nul peut s'écrire de façon unique :

$$x = p^n u$$

avec $n \in \mathbb{Z}$ et u unité de $\mathbb{Z}_p$.

Démonstration :

Soit $x = \frac{a}{b}$ avec $a \in \mathbb{Z}_p$ et $b \in \mathbb{Z}_p$. D'après le théorème 1 on a, de façon unique, $a = p^m\, a'$ et $b = p^k\, b'$ où a' et b' sont des unités de $\mathbb{Z}_p$ d'où $x = p^{m-k} \frac{a'}{b'} = p^{m-k}\, a'(b')^{-1}$, $a'(b')^{-1} \in \mathbb{Z}_p$ et est une unité puisque $a'(b')^{-1}$ $(a')^{-1}\, b' = 1$.

II-2 REPRESENTATION CANONIQUE DES NOMBRES p-ADIQUES

D'après la propriété 1 un entier p-adique peut être défini par plusieurs suites d'entiers rationnels qui se déduisent les uns des autres par congruence. Parmi toutes ces suites nous allons en caractériser une et cette caractérisation sera à la base du développement de Hensel d'un nombre p-adique.

Soit l'entier p-adique $x = (x_o, x_1, x_2, \dots)$. Prenons y_n le plus petit entier positif ou nul congru à x_n modulo (p^{n+1}). On obtient ainsi la suite (y_n) vérifiant :

$$y_n \equiv y_{n-1} \mod (p^n) \quad n = 1, 2, \dots$$

$$0 \leq y_n < p^{n+1} \quad n = 0, 1, 2, \dots$$

Cette suite est la suite canonique représentant le nombre p-adique x.
Elle est définie par :

$$y_o = a_o$$

$$y_n = y_{n-1} + a_n\, p^n \quad n = 1, 2, \dots$$

$$\text{et} \quad a_n \in [0, p[$$

Toute suite canonique est donc de la forme :

$$\{a_o, a_o + a_1 p, a_o + a_1 p + a_2 p^2, \dots\} \text{ avec } 0 \leq a_i < p.$$

On écrit symboliquement l'entier p-adique x :

$$x = \sum_{i=o}^{\infty} a_i\, p^i$$

Propriété 4 : Tout nombre p-adique non nul peut s'écrire de façon unique.

$$x = p^n \sum_{i=o} a_i\, p^i$$

avec n $\mathbb{Z}$ et $0 \leq a_i < p$.

Cette série s'appelle le développement de Hensel du nombre p-adique x.

On va donner maintenant l'algorithme qui permet de calculer les coefficients a_i du développement de Hensel, appelé algorithme de Hensel.

II-3 ALGORITHME DE HENSEL [11, 13]

Avant de présenter l'algorithme nous avons besoin d'un certain nombre de résultats :

Lemme 1 : Quel que soit $x = a/b \in \mathbb{Z}_p$, il existe des entiers uniques c et d tels que :

$$\frac{a}{b} = c + \frac{d}{b} p \quad \text{avec } 0 \leq c \leq p-1$$

Démonstration :

Puisque $x \in \mathbb{Z}_p$ alors p ne divise pas b. D'autre part, puisque p est premier, le p.g.c.d. de b et p est égal à 1. Alors d'après le théorème de Bezout, il existe des entiers n et m tels que :

$$b_n + p_m = 1$$

et donc $\quad b(an) + p(am) = a$

$\forall k \in \mathbb{Z}$ on a :

$$b(an-kp) + p\,(am - kb) = a$$

d'autre part on sait que $\forall$ an $\exists$ k tel que :

$$kp \leq an < (k+1)\,p$$

On peut donc choisir l'entier k de telle sorte que :

$$0 \leq an - kp \leq p-1$$

avec ce choix de k, posons :

$$c = an - kp \text{ et } d = am + kb$$

d'où :

$$bc + pd = a$$

puisque $b \neq 0$ on a encore

$$\frac{a}{b} = c + \frac{d}{b} p$$

Montrons maintenant l'unicité de c et d. On suppose qu'il existe c' et d' possédant la même propriété, c'est à dire

$$bc' + pd' = a$$

en faisant la différence on a :

$$b(c'-c) = p(d-d')$$

ou encore

$$\frac{b}{p}(c'-c) = d-d'$$

puisque d - d' est entier et que p ne divise pas b, il s'en suit que p divise (c'-c).
D'autre part $0 \leq c \leq p-1$ et $0 \leq c' \leq p-1$ donc $-(p-1) \leq c'-c \leq p-1$ et par conséquent c' = c et d' = d.

Dans le lemme 1, on peut remplacer p par p^n avec $n \geq 1$. Le p.g.c.d. de b et p^n est toujours égal à 1 et donc on a le :

Lemme 2 : Quel que soit $x = a/b \in \mathbb{Z}_p$, il existe des entiers uniques c_n et d_n tels que $\forall\, n \geq 1$:

$$\frac{a}{b} = c_n + \frac{d_n}{b}\, p^n \qquad 0 \leq c_n \leq p^n - 1$$

Nous pouvons maintenant énoncer la propriété fondamentale :

Lemme 3 : Quel que soit $x = a/b \in \mathbb{Z}_p$, il existe des entiers uniques $a_o,\ldots,a_{n-1}$ et d_n, tels que $\forall\, n \geq 1$:

$$\frac{a}{b} = a_o + a_1 p + \ldots + a_{n-1}\, p^{n-1} + \frac{d_n}{b}\, p^n \text{ avec } 0 \leq a_i \leq p-1$$

Démonstration :

D'après le lemme 2 on a également

$$\frac{a}{b} = c_{n+1} + \frac{d_{n+1}}{b}\, p^{n+1} \qquad 0 \leq c_{n+1} \leq p^{n+1} - 1$$

donc $$c_n - c_{n+1} = \frac{p^n}{b}\,(pd_{n+1} - d_n)$$

ou encore

$$\frac{b}{p^n}\,(c_n - c_{n+1}) = p\, d_{n+1} - d_n$$

puisque p ne divise pas b, il en est de même de p^n et, puisque $p\, d_{n+1} - d_n$ est un entier, p^n divise $c_n - c_{n+1}$.

Il existe donc un entier a_n tel que :

$$c_{n+1} = c_n + a_n\, p^n$$

mais $0 \leq c_n \leq p^n-1$ et $0 \leq c_{n+1} \leq p^{n+1} - 1$ et donc on a :

$$0 \leq a_n \leq p-1 \quad \forall\, n$$

en posant $c_1 = a_o$ on a donc $c_n = a_o + a_1 p + \ldots + a_{n-1}\, p^{n-1}$

d'où la fin de la démonstration en utilisant le lemme 2.

L'obtention pratique du développement de Hensel d'un nombre rationnel repose sur le théorème 2 et l'utilisation récursive du lemme 1.

Soit r un nombre rationnel ; l'algorithme pour obtenir le développement de Hensel est le suivant :

[E1] trouver n tel que

$r = p^n \frac{a}{b}$ où p ne divise ni a ni b.

[E2] trouver x et y entiers tels que

$$bx + py = 1$$

ce qui revient à résoudre la congruence

$$bx \equiv 1 \mod (p)$$

cette résolution s'effectue à l'aide de l'algorithme d'Euclide et du théorème de Bezout [13].

[E3] On pose $k = 0$ et $d_o = a$

[E4] On cherche j tel que

$$0 \leq a_k = d_k x - jp \leq p-1$$

on pose $d_{k+1} = d_k y + jb$

[E5] Si $d_{k+1} \neq 0$ on pose $k = k+1$ et on va à [E4]

si $d_{k+1} = 0$ on prend $a_i = 0$ pour $i > k$ et on arrête l'algorithme.

Dans le cas où r est un entier ou une fraction de base c'est à dire $r = a$ ou $r = \frac{a}{p^n}$ le développement de Hensel est identique au développement de a dans le système à base p, il est donc fini.

<u>Notations</u>

Deux types de notation sont utilisés pour représenter un nombre p-adique x.

<u>1er type</u> :

si $n = 0$ $\quad . a_o a_1 \ldots a_n \ldots \quad \ldots (H_p)$

si $n > 0$ $\quad . 00 \ldots 0\, a_o a_1 \ldots a_n \ldots (H_p)$

avec (n-1) zéros après la virgule.

si n 0 $\quad a_n \ldots a_{-1} . a_o a_1 \ldots a_{-n} \ldots (H_p)$

2ème type : On utilise une notation virgule flottante normalisée c'est à dire avec une mantisse et un exposant (m_x, e_x) [15]

si $n = 0$ $\quad m_x = .a_o\ a_1 \ldots a_n \ldots,\ e_x = 0$

si $n > 0$ $\quad m_x = .a_n\ a_{n+1} \ldots\ldots\ldots,\ e_x = -n$

si $n < 0$ $\quad m_x = .a_{-n}\ a_{-n+1} \ldots\ldots,\ e_x = n$

II-4 DEFINITION AXIOMATIQUE [11]

Soient x un nombre rationnel et p un nombre premier. Il est toujours possible de trouver des entiers relatifs a, b et α tel que x s'écrit sous la forme :

$x = p^{\alpha} \frac{a}{b}$ avec p ne divisant ni a ni b.

On pose pour la suite la fonction $|.|_p$ définie par :

$|x|_p = C^{\alpha}$ où C est un nombre réel tel que $C \in]0, 1[$

pour x = 0 nous posons

$|0|_p = 0$

Propriété 5 : L'application $x \to |x|_p$ de $\mathbb{Q} \to \mathbb{R}$ vérifie les propriétés d'une valeur absolue qui sont :

- $|x|_p = 0$ si et seulement si $x = 0$
- $|xy|_p = |x|_p\ |y|_p$
- $|x+y|_p < |x|_p + |y|_p$

de plus elle vérifie :

$|x+y|_p \leq \text{Max}\ (|x|_p, |y|_p)$

Une telle application est appelée valeur absolue ultramétrique ou non archimédienne.

Définition 4 :

Le complété de l'ensemble $\mathbb{Q}$ des nombres rationnels muni de la distance $d(x, y) = |x-y|_p$ est appelé le corps des nombres p-adiques. On le note $\mathbb{Q}_p$. La distance d(x, y) est appelée la distance p-adique.

Définition 5 : On dit qu'une suite $\{x_n\}$ de nombres p-adiques converge vers $x \in \mathbb{Q}_p$ si :

$$\lim_{n \to \infty} d(x_n, x) = 0$$

Propriété 6 : la suite $(p^n)_{n \geq 1}$ converge vers 0 dans $\mathbb{Q}_p$.

Démonstration :

D'après la définition de la distance p-adique on a :

$$d(0, p^n) = |p^n|_p = c^n$$

et comme $c \in]0, 1[$ alors $c^n \to 0$ pour $n \to \infty$.

Propriété 7 : Le développement de Hensel de tout nombre p-adique converge dans $\mathbb{Q}_p$.

Propriété 8 [12] : La série infinie $1 + p^2 + p^4 + p^6 + \ldots$ converge vers $1/(1-p^2)$ dans l'ensemble $\mathbb{Q}_p$ muni de la distance p-adique.

On voit l'analogie avec les nombres réels pour lesquels le développement en base p converge dans $\mathbb{R}$. Cependant l'analogie s'arrête là car x_n (la $n^{\text{ième}}$ somme partielle du développement de Hensel de x) ne fournit pas une valeur approchée de x comme c'est le cas dans $\mathbb{R}$ pour le développement en base p.

II-5 EXEMPLE DE CALCUL DU DEVELOPPEMENT DE HENSEL

On donne dans ce paragraphe un exemple de calcul des coefficients du développement de Hensel par l'algorithme de Hensel.

Soit $r = 3/8$ et $p = 5$. On voit que pour l'étape 1 de l'algorithme $n = 0$. Passons maintenant à l'étape 2.

On commence par faire les divisions successives de l'algorithme d'Euclide.

Divisons b par p on trouve :

$$8 = 5q_o + r_o \text{ avec } q_o = 1 \text{ et } r_o = 3$$

Divisons maintenant p par r_o ; on a :

$$5 = 3q_1 + r_1 \quad \text{avec} \quad q_1 \text{ et } r_1 = 2$$

Divisons r_o par r_1 ; on a :

$$3 = 2q_2 + r_2 \quad \text{avec} \quad q_2 = 1 \text{ et } r_2 = 1$$

Divisons r_1 par r_2 ; on a :

$$2 = 1q_3 + r_3 \text{ avec } q_3 = 2 \text{ et } r_3 = 0$$

on conclut que $r_2 = 1$ est le p.g.c.d. de b et p. Utilisons maintenant la démonstration du théorème de Bezout pour construire x et y.

Cette démonstration consiste à construire les suites x_i, y_i en utilisant les suites q_i et r_i de l'algorithme d'Euclide de la façon suivante :

$$x_o = 1 \qquad y_o = -q_o$$

$$x_1 = -x_o q_o \qquad y_1 = 1 - y_o q_1$$

les autres termes sont donnés par :

$$x_i = x_{i-2} - x_{i-1}\, q_i \quad \text{et} \quad y_i = y_{i-2} - y_{i-1}\, q_i$$

pour i = 3, 4, ..., k-1. L'étape (k-1) étant l'avant dernière étape dans les divisions successives de l'algorithme d'Euclide.

Pour l'exemple qui nous intéresse on a :

$$x_o = 1 \qquad y_o = -1$$

$$x_1 = -1 \qquad y_1 = 2$$

$$x_2 = 2 \qquad y_2 = -3$$

et donc x = 2 et y = 3 et on a bien :

$$bx + py = 1$$

A l'étape 3 on pose k = 0 et $d_o = 3$.

Faisons l'étape 4 :

$$0 \le a_o = 2\times 3 - 1\times 5 \le 4$$

par conséquent : $a_o = 1$, j = 1 et d_1

on a :

$$0 \le a_1 = 2\times(-1) - (-1)\times 5 \le 4$$

par conséquent : $a_1 = 3$, j = -1 et $d_2 = -5$

on a : $\quad 0 \le a_2 = (-5)\times 2 - (-2)\times 5 \le 4$

par conséquent : $a_2 = 0$, j = -2 et $d_3 = -1$

on voit donc que l'on aura, par un argument de périodicité :

$$a_3 = 3,\ d_4 = -5$$

$a_4 = 0,\ d_5 = -1$ et ainsi de suite. Finalement :

$$\frac{3}{8} = .1303030\ \ldots$$

vérifions que le développement de Hensel converge vers 3/8 dans $\mathbb{Q}_5$. On a :

$$1 + 3p + 3p^3 + 3p^5 + \ldots = 1 + 3p\ (1+p^2 + p^4 + \ldots)$$

or d'après la propriété 8 la série $(1 + p^2 + p^4 + \ldots)$ converge vers $1/(1-p^2)$ et par conséquent le développement de Hensel précédent converge vers :

$$1 + 3p/(1-p^2) = 1 + \frac{3\times5}{1-25} = \frac{3}{8}$$

III - ARITHMÉTIQUE P-ADIQUE

On va maintenant étudier les opérations arithmétiques élémentaires sur les nombres p-adiques. Ces opérations sont définies à partir du développement de Hensel des nombres p-adiques de la même manière que les opérations arithmétiques en base p sont définies à partir du développement en série des nombres réels en base p.

Addition

On considère les deux nombres p-adiques suivants :

$$a = p^n(a_o + a_1p + \ldots) \qquad a_o \neq 0$$

$$p = p^m(b_o + b_1p + \ldots) \qquad b_o \neq 0$$

on suppose, pour simplifier, que n = m. Si par exemple n > m on écrira :

$$a = p^m(0 + 0p + \ldots + 0p^{n-m-1} + a_op^{n-m} + a_1p^{n-m+1} + \ldots)$$

Cette opération est analogue à celle que l'on effectue en pase p en plaçant les chiffres correspondants aux mêmes puissances de p les uns en dessous des autres.

Effectuons maintenant l'addition c = a+b. On a :

$$c = a+b = p^n[(a_o+b_o)+(a_1+b_1)p + \ldots] = p^n(c_o+c_1p + \ldots)$$

Le calcul des coefficients peut se formaliser par les règles suivantes :

$$r_o = 0$$

pour i = 0, 1, 2, ... on a :

$$c_i = a_i + b_i + r_i - p$$

$$r_{i+1} = \begin{cases} 1 & \text{si } a_i + b_i + r_i \geq p \\ 0 & \text{si } a_i + b_i + r_i < p \end{cases}$$

Donnons un exemple avec p = 5 :

$$\frac{3}{8} = .1303030 \ldots$$

$$-\frac{22}{8} = .1333333 \ldots$$

effectuons l'addition

	. 1303030 ...
	. 1333333 ...
report	0010101
	. 2141414 ...

et on a bien la suite $(2 + p + 4p^2 + p^3 + 4p^4 + \ldots)$ qui converge vers

$$2 + \frac{p}{1-p^2} + \frac{4p^2}{1-p^2} = -\frac{19}{8} = \frac{3}{8} - \frac{22}{8}$$

<u>Complémentation et soustraction</u>

L'opération de complémentation, consiste à obtenir le développement de Hensel de (-a) à partir du développement de a. En d'autres termes il faut trouver le développement de (-a) tel que :

$$a + (-1) = 0$$

Posons : $a = p^n (a_o + a_1 p + \ldots)$

le calcul des coefficients de $-a = p^n (b_o + b_1 p + \ldots)$ est donné par les règles suivantes :

$$b_o = p - a_o$$

$$b_i = p - a_i - 1 \qquad i = 1, 2, \ldots$$

prenons un exemple avec p = 5 :

$$\frac{3}{8} = .\ 1303030 \ldots$$

on a donc :

$$-\frac{3}{8} = .\ 4141414$$

en effet :

$$,4141414\ \ldots = 4 + \frac{p}{1-p^2} + \frac{4p^2}{1-p^2} = -\frac{3}{8}$$

La soustraction revient donc à effectuer d'abord la complémentation du nombre à soustraire suivie d'une addition.

<u>Multiplication</u>

On considère les deux p-adiques suivants :

$$a = p^n\ (a_o + a_1\ p \ldots)$$

$$p = p^m\ (b_o + b_1\ p \ldots)$$

Posons $c = ab = p^{n+m}\ (c_o + c_1\ p + \ldots)$. Les coefficients c_i sont calculés par les formules suivantes :

$$r_o = 0$$

pour $i = 0, 1, \ldots$ trouver $r_{i+1} \in \mathbb{N}$ et $0 \leq c_i < p-1$ tels que :

$$r_i + a_o\ b_i + a_1\ b_{i-1} + \ldots + a_{i-1}\ b_1 + a_i\ b_o = r_{i+1}\ p + c_i$$

donnons un exemple numérique avec $p = 5$:

$$\frac{3}{8} = .\ 1303030 \ldots$$

$$-\frac{4}{3} = .\ 2131313$$

On peut disposer la multiplication entre nombres p-adiques comme on le fait pour la multiplication entre nombres réels ; la seule différence est que l'on travaille de gauche vers la droite :

$$\begin{array}{lr}
 & .\ 13030\ \ldots\ldots \\
 & .\ 21313\ \ldots\ldots \\
\hline
 & 26060606\ \ldots \\
 & 1303030\ \ldots \\
 & 390909\ \ldots \\
 & 13030\ \ldots \\
 & 3909\ \ldots \\
 & \ldots\ldots\ldots \\
 & \ldots\ldots \\
\text{Report} & 00113\ \ldots\ldots \\
\hline
 & .\ 22222\ \ldots\ldots
\end{array}$$

en effet $.\ 2222 \ldots = \frac{2}{1-p} = -\frac{1}{2} = (-\frac{4}{3})(\frac{3}{8})$

Inversion et division

L'opération d'inversion consiste à obtenir le développement de Hensel de a^{-1} à partir du développement de a. En d'autres termes il faut trouver le développement de a^{-1} tel que :

$$a \times a^{-1} = 1$$

Posons : $a = p^n(a_o + a_1 p + \dots)$ $\quad a_o \neq 0$

$$a^{-1} = p^n(b_o + b_1 p + \dots)$$

en effectuant la multiplication aa^{-1} on doit avoir :

$$c_o = 1 \text{ et } c_i = 0 \text{ pour } i > 0$$

il faut donc trouver les b_i et les r_i satisfaisants :

$$a_o\, b_o = 1 + r_1\, p$$

$$r_i + a_o\, b_i + \dots + a_i\, b_o = r_{i+1}\, p \quad i = 1, 2, \dots$$

avec $0 \leq b_i \leq p-1$ et $r_i \in \mathbb{N}$

Reprenons l'exemple 3/8 = ,1303030 ...

on a :

$b_o = 1 + r_1 p$	donc $b_o = 1$ et $r_1 = 0$
$b_1 + 3 = r_2 p$	donc $b_1 = 2$ et $r_2 = 1$
$1 + b_2 + 6 = r_3 p$	donc $b_2 = 3$ et $r_3 = 2$
$2 + b_2 + 9 + 3 = r_4 p$	donc $b_3 = 1$ et $r_4 = 3$
$3 + b_4 + 3 + 6 = r_5 p$	donc $b_4 = 3$ et $r_5 = 3$
$3 + b_5 + 9 + 9 + 3 = r_6 p$	donc $b_5 = 1$ et $r_6 = 5$

et ainsi de suite. On a donc :

$$(3/8)^{-1} = .123131\ \dots = 1 + 2p + \frac{3p^2}{1-p^2} + \frac{p^3}{1-p^2} = \frac{8}{3}$$

La division revient à effectuer d'abord une inversion suivie d'une multiplication.

III - LES NOMBRES P-ADIQUES TRONQUES [14, 15, 16]

On a vu au paragraphe II-3 que seuls les entiers et les fractions de base admettent un développement de Hensel fini. Pour les autres rationnels si on tronque ce developpement infini, la série tronquée représente un entier qui n'a rien de commun avec le rationnel de départ au sens de la métrique usuelle. Il n'est donc pas possible d'exploiter directement la représentation tronquée.

C'est en 1975 que KRISHNAMURTHY, MAHADEVA RAO et SUBRAMANIAN ont proposé une représentation tronquée des nombres p-adiques qu'on va reprendre dans ce paragraphe en donnant les limites de son utilisation dans le calcul numérique.

Soit le nombre rationnel α, et soit son développement de Hensel représenté par $a_{-n} \dots a_{-1}, a_o\, a_1\, a_2 \dots$ on appelle code de Hensel le développement fini suivant $H(p, r, \alpha) = a_{-n} \dots\, a_{-1}.a_o\, a_1\, a_2 \dots a_m$ et répondant aux conditions :

- $r = n + m + 1$
- le numérateur et le dénominateur de α doivent appartenir à un intervalle $[-N, +N]$
- le rang r doit être pair.

Théorème [4] : Soit p un nombre premier et soit r un entier positif. Tout nombre rationnel $\alpha = \frac{a}{b}$ tel que $0 \leq |a| \leq N$, $0 \leq |b| \leq N$ a un code $H(p, r, \alpha)$ unique pour $N = \left[\sqrt{\frac{p^r-1}{2}}\right]$

III-1 CONVERSION DES RATIONNELS EN p-ADIQUES TRONQUES

On présente dans ce paragraphe la conversion d'un certain nombre de rationnels en p-adiques représentés par leurs codes de Hensel. Pour les exemples on prend p = 5 et r = 8, ce qui donne un intervalle de définition des entiers égal à [-441, 441]

Cas d'un entier positif

Le code de Hensel d'un entier positif s'écrit sous la forme :

$$. a_o\, a_1 \dots a_{\frac{r}{2}-1}\ \underbrace{0 \dots\dots 0}_{\frac{r}{2}\text{ termes}}$$

<u>exemple</u> : $p = 5 \quad r = 8 \quad N = 441$

$H(5, 8, 199) = .44210000$

$H(5, 8, 17) = .23000000$

<u>Cas d'un entier négatif</u>

Soit $. a_o a_1 \ldots a_{\frac{r}{2}-1} 0 \ldots 0$ le code de Hensel d'un entier positif.

Le code de Hensel de son opposé s'écrit :

$$b_o b_1 \ldots b_{\frac{r}{2}-1} \underbrace{(p-1)(p-1) \ldots (p-1)}_{\frac{r}{2} \text{ termes}}$$

avec $b_o = p - a_o$

$b_i = p - 1 - a_i$ pour $i = 1, \ldots, \frac{r}{2} - 1$

<u>Exemple</u>

$H(5, 8, -199) = .10234444$

$H(5, 8, -17) = .31444444$

<u>Cas d'une fraction de base</u>

Une fraction de base est un rationnel $\frac{a}{b}$ avec $b = p^n$. Dans ce cas le code de Hensel représentant $\frac{a}{b}$ est le même que celui de a avec un décalage de n positions du point p-adique.

<u>Exemple</u>

$H(5, 8, \frac{199}{125}) = H(5, 8, 199 \times 5^{-3}) = 442.10000$

<u>Cas d'un rationnel quelconque</u>

On prend dans ce cas la troncature du développement de Hensel à r digits.

<u>Exemple</u>

On prend $p = 5$ et $r = 4$, ce qui donne $N = 17$

$H(5, 4, \frac{2}{3}) = .4131$

$H(5, 4, \frac{10}{3}) = .0413$

III-2 PROBLEME POSE PAR LA REPRESENTATION EN CODE DE HENSEL

Définition : On appelle ensemble des N-Fractions de Farey l'ensemble des rationnels $\frac{a}{b}$ tel que $0 \leq a \leq b \leq N$; son cardinal est égal à $\frac{3N^2}{\pi^2} \approx \frac{N^2}{3}$

On déduit de cette définition que le cardinal de l'ensemble des rationnels défini dans l'intervalle $[-N, N]$ est égal à $\frac{4N^2}{3}$. D'autre part le code de Hensel permet de représenter plus de $(p^r-1) \times \frac{r}{2}$ rationnels différents.
Or d'après le théorème de caractérisation des p-adiques N doit être égal à la partie entière de $\sqrt{\frac{p^r-1}{2}}$. Donc dès que $r \geq 4$ on peut représenter plus de nombres p-adiques que de rationnels pour p et r donnés.

Ceci montre qu'il existe dans la représentation des p-adiques par le code de Hensel tout un ensemble de nombres qui représentent des rationnels n'appartenant pas à l'intervalle $[-N, +N]$.

Exemple :
Le code de Hensel H (5, 8, α) = .4443000 représente l'entier 499 qui n'appartient pas à l'intervalle [-441, 441]

III-3 CONVERSION D'UN p-ADIQUE TRONQUE EN UN RATIONNEL

On vient de voir qu'un rationnel $\frac{a}{b}$ s'écrit d'une façon unique dans la représentation p-adique par :

$$\frac{a}{b} = \sum_{i=0}^{\infty} a_i\, p^i \quad \text{avec les } a_i \text{ des entiers compris entre 0 et p-1}$$

Cette représentation p-adique s'écrit symboliquement par le code de Hensel par :

$$H(p,\ r,\ \tfrac{a}{b}) = .\ a_0\ a_1 \ \dots\ a_{r-1}$$

Pour pouvoir convertir cette représentation en rationnel KRISHNAMURTHY et al. proposent de prendre r un entier pair, soit $r = 2k$, et de pondérer les coefficients a_i par les fonctions $\omega(i)$ définies par :

$$\omega(i) = p^i \text{ pour } i = 0, 1, \dots, k-1$$

$$\omega(i) = - \frac{p^i}{p^k-1} \text{ pour } i = k, \dots, 2k-1$$

en prenant $W = - C_o - C_1 p - \ldots - C_{k-1}\, p^{k-1} + (C_o - C_k)p^k + \ldots + (C_{k-1} - C_{2k-1})p^{2k-1}$

les auteurs montrent que le rationnel $\frac{a}{b}$ vérifie une équation diophantienne de la forme :

$$a(p^h - 1) - b\, W \equiv 0 \pmod{p^{2k}}$$

ou $\quad a(p^k - 1) - b\, W = K\, p^{2k}$ où K est un entier.

Brézinski [17] a donné la justification théorique d'un tel choix.

En effet, considérons la série infinie :

$$f(p) = \frac{a}{b} = a_o + a_1 p + \ldots + a_n\, p^n + \ldots$$

son approximant de type Padé avec $v(x) = x^{k-1}$ comme polynôme générateur est égal à :

$$(2k-1/k)_f(p) = C_o + C_1 p + \ldots + C_k p^{k-1} + p^k (k-1/k)_{f_{k-1}}(p)$$

avec $f_{k-1}(p) = C_k + C_{k+1}\, p + C_{k+2}\, p^2 + \ldots$

ce qui donne $(k-1/k)_{f_{k-1}}(p) = \dfrac{C_k + C_{k+1}\, p + \ldots + C_{2k-1} p^{2k-1}}{1 - p^k}$

et on obtient :

$$(2k-1/k)_f(p) = C_o + C_1 p + \ldots + C_{k-1} p^{k-1} - C_k \frac{p^k}{p^k - 1} \ldots - C_{2k-1} \frac{p^{2k-1}}{p^k - 1}$$

Donc l'approximant de type Padé $(2k-1/k)_f(p)$ est égal au $H(p, r, \frac{a}{b})$ code de $\frac{a}{b}$.

D'autre part on sait que $f(p) - (2k-1/k)_f(p) = \dfrac{p^{2k}}{\tilde{v}(p)}\; c^{(k)}\left(\dfrac{v(x)}{1-xp}\right)$

avec $\tilde{v}(x) = x^k\, v(x^{-1})$ et $c^{(k)}(x^i) = C_{k+i}$.

Soit W le numérateur de $(2k-1/k)_f(p)$. On a :

$$W = C_o - C_1 p - \ldots - C_{k-1}\, p^{k-1} + (C_o - C_k)p^k + \ldots + (C_{k-1} - C_{2k-1})p^{2k-1}$$

ce qui donne :

$$(2k-1/k)_f(p) = \frac{W}{p^k - 1} = - \frac{W}{\tilde{v}(p)}$$

en remplaçant $f(p)$ par $\frac{a}{b}$ on obtient :

$$a(p^k - 1) - b\, W = -b\, p^{2k}\, c^{(k)}\left(\frac{v(x)}{1-xp}\right)$$

Comme le membre de gauche est un entier celui de droite l'est aussi et l'on peut écrire l'équation sous la forme :

$$a(p^k - 1) - b\, W = K\, p^{2k}$$

ou

$$a(p^k-1) - b\,W \equiv 0 \quad (\text{mod } p^{2k})$$

Exemples

On prend p = 5 et r = 4

1°) Soit le p-adique .4200

$$\frac{W}{p^2-1} = 4 + 2 \times 5 = 14$$

l'équation diophantienne est : 24a - b 336 ≡ 0 mod (625) le résultat est :

a = 14   b = 1

2°) Soit le p-adique .1241

$$\frac{W}{p^2-1} = 1 + 2 \times 5 - \frac{4\times 25}{24} - \frac{1\times 125}{24} = \frac{13}{8}$$

l'équation diophantienne à résoudre est : 8a - 13b ≡ 0 mod (625) la solution est : a = 13, b = 8.

3°) Soit le p-adique .4201

$$\frac{W}{p^2-1} = 4 + 2 \times 5 - \frac{0\times 25}{24} - \frac{1\times 125}{24} = \frac{211}{24}$$

l'équation diophantienne à résoudre est :

$$24\,a - 211\,b \equiv 0 \quad \text{mod } (625)$$

la solution est : a = 1, b = 9

III-4 ARITHMETIQUE p-ADIQUE TRONQUEE [14, 15, 16]

On va maintenant étudier les opérations arithmétiques élémentaires sur les nombres p-adiques tronqués. Ces opérations sont définies à partir du code de Hensel des nombres p-adiques de la même manière que les opérations arithmétiques en base p sont définies à partir du développement en série des nombres réels en base p.

L'addition et la multiplication de deux nombres p-adiques s'obtiennent donc à partir de l'addition et la multiplication des codes de Hensel correspondants. Comme dans la numérotation en base p, les chiffres a_i du code de Hensel sont compris entre 0 et p-1. Ceci implique donc une opération de report ou de retenue d'un digit sur l'autre comme pour la numération en base p. La différence réside dans le fait que les opérations sur les codes de Hensel s'effectuent de la gauche vers la droite et que le calcul s'arrête dès qu'on obtient r digits dans le résultat.

Addition

L'addition s'effectue donc de gauche vers la droite, modulo p. Les calculs sont arrêtés à r chiffres dans le résultat.
Montrons l'addition sur un exemple.
Prenons p = 5 et r = 4 ce qui nous donne un intervalle de définition égal à [-17, 17].
Soit les deux p-adiques représentés par :

$$H(5,\ 4,\ \tfrac{2}{3}) = .4131$$

$$H(5,\ 4,\ \tfrac{1}{5}) = 1.000$$

on dispose les calculs de la façon suivante :

$$\begin{array}{r} .4131 \\ 1.000 \\ \hline 1.413 \end{array}$$

on vérifie que 1.413 représente le rationnel $\frac{13}{15}$.
en effet $\frac{2}{3} + \frac{1}{5} = \frac{13}{15}$

Dépassement de capacité dans l'addition

Le dépassement de capacité se produit quand le résultat de l'opération n'appartient plus à l'intervalle [-N, N]. Deux cas différents peuvent se présenter :

1er cas : Bien qu'il y ait dépassement de capacité dans le résultat, sa conversion en rationnel reste possible.

Exemple : $H(5,\ 4,\ \tfrac{7}{12}) = .1202$

$H(5,\ 4,\ \tfrac{1}{5}) = 1.000$

faisons l'addition

$$\begin{array}{r} .1202 \\ 1.000 \\ \hline 1.120 \end{array}$$

Convertissons le p-adique .1120 en rationnel

$$\frac{W}{p^2-1} = 1 + 5 - \frac{2\times 25}{24} = \frac{47}{12}$$

ce qui donne pour le p-adique 1.120 la valeur rationnelle $\frac{47}{12} \times 5^{-1} = \frac{47}{60}$
qui est bien le résultat de l'opération $\frac{7}{12} + \frac{1}{5}$

2ème cas : Soient les deux p-adiques :

$$H(5, 4, \frac{7}{12}) = .1202$$

$$H(5, 4, \frac{17}{11}) = .2411$$

effectuons l'addition :

$$\begin{array}{r} .1202 \\ .2411 \\ \hline .3123 \end{array}$$

Pour convertir en rationnel le p-adique qu'on vient de trouver on doit résoudre l'équation diophantienne :

$$24a - 233b \equiv 0 \bmod (625)$$

Or le résultat de l'addition en rationnel est égal à 281/132 et ne vérifie pas cette équation.

Il n'est donc pas possible de savoir à l'avance si un p-adique représente un rationnel défini dans l'intervalle [-N, +N] et s'il est possible de le convertir pour trouver le rationnel cherché.

<u>Complémentation et soustraction</u>

Soit le p-adique représenté par son code de Hensel général

$\alpha = a_{-n} \ldots a_{-1} \,.\, a_o \ldots a_m$ avec $(1 + n + m)$ pair

l'opération de complémentation consiste à obtenir le p-adique $(-\alpha)$ dont le code de Hensel s'écrit :

$-\alpha = b_{-n} \ldots b_{-1} \,.\, b_o \ldots b_m$

par les règles suivantes :

<u>1er cas</u> : si $a_i \neq 0$ pour $i = -n, \ldots, 0, \ldots, m$

alors $b_{-n} = p - a_{-n}$

et $b_i = p - 1 - a_i$ pour $i = -n + 1, \ldots, 0, \ldots, m$

2ème cas : s'il existe j, $-n < j < m$ tel que $a_i = 0$ pour $-n \leq i \leq j$

alors $b_i = 0$ pour $-n \leq i \leq j$

et $b_{j+1} = p - a_{j+1}$

$b_i - p - 1 - a_i$ pour $j + 1 < i \leq m$

Exemple

Soit le p-adique représenté par son code de Hensel :

$H(5, 4, \frac{5}{4}) = .0433$

son complément est défini par :

$H(5, 4, -\frac{5}{4}) = .0111$

La soustraction revient à effectuer d'abord la complémentation du nombre à soustraire suivie d'une addition.

Exemple

Soit à effectuer l'opération $\frac{4}{7} - \frac{3}{2} = -\frac{13}{14}$ en arithmétique p-adique avec $p = 5$ et $r = 4$

on a $H(5, 4, \frac{4}{7}) = .2423$

et $H(5, 4, +\frac{3}{2}) = .4222$

le complément de $H(5, 4, \frac{3}{2})$ est défini par $H(5, 4, -\frac{3}{2}) = .1222$

on exécute l'addition :

$$\begin{array}{r} .2423 \\ .1222 \\ \hline .3101 \end{array}$$

et on vérifie que .3101 représente bien le rationnel $-\frac{13}{14}$ puisqu'il est solution de l'équation diophantienne :

$$24a - 67\,b \equiv 0 \mod (625)$$

Multiplication

Dans l'état actuel de la recherche seule l'utilisation de la notation virgule flottante des p-adiques permet d'effectuer la multiplication. Soient les deux p-adiques α et β définis par

$\alpha = (m_\alpha, e_\alpha)$ avec $m_\alpha = .a_{-n} \ldots a_o \ldots a_m$, $e_\alpha = n$ et tel que $n+m+1 = r$

$\beta = (m_\beta, e_\beta)$ avec $m_\beta = .a_{-k} \ldots b_o \ldots b_s$, $e_\beta = k$ et tel que $k+s+1 = r$

On commence par calculer les produits partiels entre les mantisses :

$$P_{ij} = b_i\, a_j \quad \text{pour} \quad j = -n, \ldots, m$$

ainsi que $P_i = \sum_{j=-n}^{m} P_{ij}\ \Delta(n+j)$

$\Delta(x)$ réprésentant le décalage de x digits vers la droite la mantisse du résultat est obtenue par $m_{\alpha\beta} = \sum_{i=-k}^{s} P_i\ \Delta(k+i)$.

Toutes les opérations sur les digits a_i, b_i se font modulo p et dans le cas où il y a une retenue elle est reportée sur le digit de droite.

L'exposant $e_{\alpha\beta}$ est donné par l'addition de e_α et e_β.

Pour obtenir le résultat définitif on tronque le résultat à r digits.

Exemple

Effectuons l'opération $\frac{7}{15} \times \frac{15}{13} = \frac{7}{13}$

les codes de Hensel des opérandes sont donnés par :

$$H(5,\ 4,\ \alpha) = H(5,\ 4,\ \tfrac{7}{15}) = (.4313,\ -1)$$

$$H(5,\ 4,\ \beta) = H(5,\ 4,\ \tfrac{15}{13}) = (.1143,\ +1)$$

posons la multiplication en p-adique.

m_α : .4313

m_β : .1143

calculons le premier produit partiel P_o

$$\begin{array}{ll} P_{oo}: & 4 \\ P_{01}: & \ \ 3 \\ P_{02}: & \ \ \ \ 1 \\ P_{03}: & \underline{\ \ \ \ \ \ 3} \\ P_o: & 4313 \end{array}$$

le second produit partiel P_1 est donné par :

$$\begin{array}{ll}
P_{10} : & 4 \\
P_{11} : & \ \ 3 \\
P_{12} : & \ \ \ \ 1 \\
P_{13} : & \underline{\ \ \ \ \ \ 3} \\
P_1 \ : & 4313
\end{array}$$

de la même manière on calcule P_2 et P_3 et on trouve :

$$\begin{array}{ll}
P_o & : \ 4313 \\
P_1 & : \ \ \ 4313 \\
P_2 & : \ \ \ \ \ 10231 \\
P_3 & : \ \ \ \ \ \ \ \underline{21002} \\
 & \ \ \ 42122412
\end{array}$$

l'exposant est égal à 1 - 1 = 0

en tronquant la mantisse à 4 digits on trouve le résultat définitif : (.4212,0)

En effet, on vérifie que $H(5, 4, \frac{7}{13}) = (.4212, 0)$

Effectuons la multiplication sans utiliser la représentation virgule flottante.

$\frac{7}{15}$ est codé par 4.313

$\frac{15}{13}$ est codé par .0114

posons la multiplication

$$\begin{array}{l}
\ \ 4.313 \\
\underline{.0\ 114} \\
\ \ 0\ 000 \\
\ \ \ \ \ 4313 \\
\ \ \ \ \ \ \ 4313 \\
\ \ \ \ \ \ \ \ \ \underline{10232} \\
\ \ 0\ 4210142
\end{array}$$

en plaçant convenablement le point p-adique et en tronquant le résultat à 4 digits on trouve .4210 qui est différent du résultat exact .4212

Division

Pour les mêmes raisons que la multiplication, la division ne peut se faire que si on représente les p-adiques par une mantisse et un exposant.

Soit le dividente $\alpha = (m_\alpha, e_\alpha) = (.a_o \dots a_n, e_\alpha)$

et le diviseur $\beta = (m_\beta, e_\beta) = (.b_o \dots b_n, e_\beta)$.

Notons par γ le quotient $\gamma = \frac{\alpha}{\beta} = (m_\gamma, e_\gamma) = (.q_o \dots q_n, e_\gamma)$.

La division dans le système p-adique est très simple car elle est déterministe contrairement à la division dans les systèmes numériques habituels où elle s'effectue par tatonnement. L'algorithme de la division consiste en une inversion suivie d'une multiplication. Les étapes de l'algorithme du calcul de la mantisse m_γ sont les suivantes :

on pose $R_o = \alpha$

on calcule b_o^{-1} inverse de b_o modulo p, en résolvant l'équation $|b_o \times b_o^{-1}|_p = 1$ qu'on peut écrire sous la forme $b_o \times b_o^{-1} \equiv 1 \bmod (p)$, la solution d'une telle équation est donnée par l'algorithme d'Euclide et la démonstration du théorème de Bezout.

On calcule pour i variant de 0 à n les termes q_i et R_{i+1} par :

$q_i = R_{ii} \times b_o^{-1} \bmod (p)$ R_{ii} étant le $i^{ème}$ digit de R_i

$R_{i+1} = R_i + q_i \times \bar{\beta} \times \Delta(i)$ $\bar{\beta}$ est le complément de β et $\Delta(i)$ est le décalage à droite de i digits.

L'exposant e_γ est égal à $e_\gamma = e_\alpha - e_\beta$

Exemple

Soit à effectuer la division $\gamma = \frac{\alpha}{\beta}$ pour $\alpha = \frac{2}{15}$ et $\beta = \frac{4}{15}$

les codes de Hensel de ces deux rationnels s'écrivent :

$H(5, 4, \frac{2}{15}) = (.4131, -1)$

$H(5, 4, \frac{4}{15}) = (.3313, -1)$

l'inverse de b_o est égal à $b_o^{-1} = 2$.

Le complément de $\frac{4}{15}$ est égal en p-adique à .2131 posons les différentes étapes du calcul du quotient.

i	0	1	2	3	
R_o	4	1	3	1	=> $q_o = 3$
$q_o\ \bar{\beta}\ \Delta(0)$	1	4	4	4	
R_1	0	1	3	1	=> $q_1 = 2$
$q_1\ \bar{\beta}\ \Delta(1)$		4	2	1	
R_2		0	1	3	=> $q_2 = 2$
$q_2\ \bar{\beta}\ \Delta(2)$			4	2	
R_3			0	1	=> $q_3 = 2$

l'exposant e_γ est donné par $e_\gamma = -1 - (-1) = 0$
le résultat de la division est donc (.32222, 0) qui représente bien le rationnel $\frac{1}{2}$.

III-5 PROBLEME POSE PAR L'ARITHMETIQUE p-ADIQUE TRONQUEE

Comme on vient de le voir il n'est pas possible d'effectuer correctement la multiplication ou la division si le premier digit de l'un des opérandes est nul. Ce qui nous oblige à utiliser la notation virgule flottante normalisée. Or l'addition s'effectue en utilisant la notation sans la virgule flottante.

Dans une expression arithmétique il est donc possible que le premier digit d'un résultat intermédiaire, dû à une addition, soit nul, ceci empêcherait la poursuite du calcul de l'expression dans l'arithmétique p-adique si celle-ci comprend une multiplication ou une division.

Exemple
On veut effectuer l'expression $(\frac{2}{16} + \frac{3}{16}) : (\frac{2}{17} + \frac{3}{17})$ dans l'arithmétique p-adique. Les codes de Hensel des opérandes sont :

$H(5,\ 4,\ \frac{2}{16}) = .2414$

$H(5,\ 4,\ \frac{3}{16}) = .3104$

$H(5,\ 4,\ \frac{2}{17}) = .1132$

$H(5,\ 4,\ \frac{3}{17}) = .4121$

Effectuons les deux additions en p-adiques

$$\begin{array}{r} .2414 \\ .3104 \\ \hline .0123 \end{array}$$

le p-adique .123 représente bien le rationnel $\frac{5}{16}$

$$\begin{array}{r} .1132 \\ .4121 \\ \hline .0304 \end{array}$$

le p-adique .304 représente bien le rationnel $\frac{5}{17}$.

Par contre, il est impossible d'effectuer la division puisque les premiers digits des opérandes sont nuls. Comme il est impossible de représenter les résultats intermédiaires en virgule flottante normalisée, la seule solution pour continuer les calculs est de convertir en rationnels les résultats intermédiaires dont le premier digit est nul, et de convertir ces rationnels en p-adiques représentés en virgule flottante normalisée.
Appliquons cette règle à l'exemple qu'on vient de voir.
La conversion en rationnel de .0123 donne $\frac{5}{16}$ et celle de .304 donne $\frac{5}{17}$. On convertit maintenant en p-adique, représenté en virgule flottante, ces mêmes rationnels on trouve :

$$H(5,\ 4,\ \frac{5}{16}) = (.1234,\ 1)$$

$$H(5,\ 4,\ \frac{5}{17}) = (.3043,\ 1)$$

effectuons la division p-adique des mantisses :

$\alpha = .1234$

$\beta = .3043$

on a donc $\bar{\beta} = .2401$ et $b_o^{-1} = |3|_5^{-1} = 2$

i	1	2	3	4		
R_o	1	2	3	4	=>	$q_o = 2$
$q_o\ \bar{\beta}\ \Delta(0)$	4	3	1	2		
R_1	0	1	0	2	=>	$q_1 = 2$
$q_1\ \bar{\beta}\ \Delta(1)$		4	3	1		
R_2		0	4	3	=>	$q_2 = 3$
$q_2\ \bar{\beta}\ \Delta(1)$			1	3		
R_3			0	2	=>	$q_3 = 4$

Le résultat final est donc (.2234, 0), et il représente bien le rationnel $\frac{17}{16}$.

V - CONCLUSION

Les problèmes qu'on a exposés pour l'addition p-adique ainsi que le calcul des expressions arithmétiques font que cette arithmétique reste à améliorer. De plus il serait souhaitable de pouvoir résoudre les problèmes suivants :

- recherche d'algorithmes simples pour la conversion des p-adiques en rationnels
- détection du signe et de la grandeur du code de Hensel directement sans le convertir explicitement en rationnel
- détection du dépassement de capacité.

BIBLIOGRAPHIE

[1] M. LA PORTE et J. VIGNES
Algorithmes numériques. Analyse et mise en oeuvre. Technip, Paris.

[2] R.E. MOORE
Interval analysis. Prentice Hall, 1966.

[3] K. NICKEL
Interval Analysis, in "The State of the art in numerical analysis", G. Jacobs ed.

[4] U.W. KULISH, W.L. MIRANKER
Computer arithmetic in theory and practice. Academic Press, New-York, 1981.

[5] T. GREGORY
On residue arithmetic with rational operands. University of Tennessee ; Knoxville, 1980.

[6] P. HENRICI
Applied and Computational complex. Analysis. Volume 2, Addison-Wesley Reading.

[7] D.E. KNUTH
The art of computer programming. Volume 2. Addison-Wesley. 1969.

[8] J. DUMONTET
Etude de la preuve des programmes numériques. Thèse d'Etat, Paris, 1980.

[9] R.P. BRENT
A fortran multiple-precision arithmetic Package. ACM transaction on Mathematical software, vol 4, n° 1, March 1978, p. 57-70.

[10] C. BREZINSKI
Algorithmes d'accélération de la convergence. Technip, Paris 1978.

[11] J.P. SERRE
Cours d'arithmétique. Presses Universitaires de France, Paris, 1970.

[12] G. BACHMAN
Introduction to p-adic numbers and valuation theory. Academic Press, New-York, 1964.

[13] C. PISOT, M. ZAMANSKY
Mathématiques Générales. Dunod, Paris, 1966.

[14] T. MAHADEVARAO
Finite field computational techniques for exact solution of numerical problems. Bangalore 1975.

[15] R.T. GREGORY
The use of finite segment p-adic arithmetic for exact computation BIT 18 (1978) p. 282-300.

[16] KRISHNAMURTHY, MAHADEVARAO, SUBRAMANIA
Finite segment p-adic number systems with applications to exact computation. Proc. Indian Acad. Sci. Vol 81 A, n° 2, 1975 page 58-79.

[17] C. BREZINSKI
Padé-type Approximation and general orthogonal polynomials. Birkhäuser Verlag, 1980.