USTL
FLANDRES ARTOIS

UNIVERSITE DES SCIENCES ET TECHNIQUES
DE LILLE FLANDRES ARTOIS

N° d'ordre : 362

THÈSE

Nouveau Régime

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE FLANDRES ARTOIS

pour obtenir le titre de

DOCTEUR EN MATHEMATIQUES APPLIQUEES

par

Anne-Marie LITOVSKY

Accélération de la convergence des ensembles synchronisables

Soutenue le 8 juin 1989 devant la Commission d'Examen :

Président et rapporteur :	C. BREZINSKI
Rapporteur :	J. DELLA-DORA
Examinateurs :	J.P. DELAHAYE
	B. GERMAIN-BONNE
	P. MOREL
	P. SABLONNIERE

Université des Sciences et Techniques de Lille Flandres-Artois
Laboratoire d'Analyse Numérique et d'Optimisation - Bât M3 - 1e étage
U.F.R. d'I.E.E.A.
59655 VILLENEUVE D'ASCQ - CEDEX - FRANCE
Tél. 20.43.46.71

UNIVERSITE DES SCIENCES
ET TECHNIQUES DE LILLE
FLANDRES ARTOIS

DOYENS HONORAIRES DE L'ANCIENNE FACULTE DES SCIENCES

M.H. LEFEBVRE, M. PARREAU.

PROFESSEURS HONORAIRES DES ANCIENNES FACULTES DE DROIT ET SCIENCES ECONOMIQUES, DES SCIENCES ET DES LETTRES

MM. ARNOULT, BONTE, BROCHARD, CHAPPELON, CHAUDRON, CORDONNIER, DECUYPER, DEHEUVELS, DEHORS, DION, FAUVEL, FLEURY, GERMAIN, GLACET, GONTIER, KOURGANOFF, LAMOTTE, LASSERRE, LELONG, LHOMME, LIEBAERT, MARTINOT-LAGARDE, MAZET, MICHEL, PEREZ, ROIG, ROSEAU, ROUELLE, SCHILTZ, SAVARD, ZAMANSKI, Mes BEAUJEU, LELONG.

PROFESSEUR EMERITE

M. A. LEBRUN

ANCIENS PRESIDENTS DE L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

MM. M. PAREAU, J. LOMBARD, M. MIGEON, J. CORTOIS.

PRESIDENT DE L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE FLANDRES ARTOIS

M. A. DUBRULLE.

PROFESSEURS - CLASSE EXCEPTIONNELLE

M. CONSTANT Eugène	Electronique
M. FOURET René	Physique du solide
M. GABILLARD Robert	Electronique
M. MONTREUIL Jean	Biochimie
M. PARREAU Michel	Analyse
M. TRIDOT Gabriel	Chimie Appliquée

PROFESSEURS - 1ère CLASSE

M. BACCHUS Pierre	Astronomie
M. BIAYS Pierre	Géographie
M. BILLARD Jean	Physique du Solide
M. BOILLY Bénoni	Biologie
M. BONNELLE Jean-Pierre	Chimie-Physique
M. BOSCQ Denis	Probabilitéss
M. BOUGHON Pierre	Algèbre
M. BOURIQUET Robert	Biologie Végétale
M. BREZINSKI Claude	Analyse Numérique

M. BRIDOUX Michel	Chimie-Physique
M. CELET Paul	Géologie Générale
M. CHAMLEY Hervé	Géotechnique
M. COEURE Gérard	Analyse
M. CORDONNIER Vincent	Informatique
M. DAUCHET Max	Informatique
M. DEBOURSE Jean-Pierre	Gestion des Entreprises
M. DHAINAUT André	Biologie Animale
M. DOUKHAN Jean-Claude	Physique du Solide
M. DYMENT Arthur	Mécanique
M. ESCAIG Bertrand	Physique du Solide
M. FAURE Robert	Mécanique
M. FOCT Jacques	Métallurgie
M. FRONTIER Serge	Ecologie Numérique
M. GRANELLE Jean-Jacques	Sciences Economiques
M. GRUSON Laurent	Algèbre
M. GUILLAUME Jean	Microbiologie
M. HECTOR Joseph	Géométrie
M. LABLACHE-COMBIER Alain	Chimie Organique
M. LACOSTE Louis	Biologie Végétale
M. LAVEINE Jean-Pierre	Paléontologie
M. LEHMANN Daniel	Géométrie
Mme LENOBLE Jacqueline	Physique Atomique et Moléculaire
M. LEROY Jean-Marie	Spectrochimie
M. LHOMME Jean	Chimie Organique Biologique
M. LOMBARD Jacques	Sociologie
M. LOUCHEUX Claude	Chimie Physique
M. LUCQUIN Michel	Chimie Physique
M. MACKE Bruno	Physique Moléculaire et Rayonnements Atmosph.
M. MIGEON Michel	E.U.D.I.L.
M. PAQUET Jacques	Géologie Générale
M. PETIT Francis	Chimie Organique
M. POUZET Pierre	Modélisation - calcul Scientifique
M. PROUVOST Jean	Minéralogie
M. RACZY Ladislas	Electronique
M. SALMER Georges	Electronique
M. SCHAMPS Joel	Spectroscopie Moléculaire
M. SEGUIER Guy	Electrotechnique
M. SIMON Michel	Sociologie
Melle SPIK Geneviève	Biochimie
M. STANKIEWICZ François	Sciences Economiques
M. TILLIEU Jacques	Physique Théorique
M. TOULOTTE Jean-Marc	Automatique
M. VIDAL Pierre	Automatique
M. ZEYTOUNIAN Radyadour	Mécanique

PROFESSEURS - 2ème CLASSE

M. ALLAMANDO Etienne	Composants Electroniques
M. ANDRIES Jean-Claude	Biologie des organismes
M. ANTOINE Philippe	Analyse
M. BART André	Biologie animale
M. BASSERY Louis	Génie des Procédés et Réactions Chimiques

Mme BATTIAU Yvonne	Géographie
M. BEGUIN Paul	Mécanique
M. BELLET Jean	Physique Atomique et Moléculaire
M. BERTRAND Hugues	Sciences Economiques et Sociales
M. BERZIN Robert	Analyse
M.BKOUCHE Rudolphe	Algèbre
M. BODARD Marcel	Biologie Végétale
M. BOIS Pierre	Mécanique
M. BOISSIER Daniel	Génie Civil
M. BOIVIN Jean-Claude	Spectroscopie
M. BOUQUELET Stéphane	Biologie Appliquée aux enzymes
M. BOUQUIN Henri	Gestion
M. BRASSELET Jean-Paul	Géométrie et Topologie
M. BRUYELLE Pierre	Géographie
M. CAPURON Alfred	Biologie Animale
M. CATTEAU Jean-Pierre	Chimie Organique
M. CAYATTE Jean-Louis	Sciences Economiques
M. CHAPOTON Alain	Electronique
M. CHARET Pierre	Biochimie Structurale
M. CHIVE Maurice	Composants Electroniques Optiques
M. COMYN Gérard	Informatique Théorique
M. COQUERY Jean-Marie	Psychophysiologie
M. CORIAT Benjamin	Sciences Economiques et Sociales
Mme CORSIN Paule	Paléontologie
M. CORTOIS Jean	Physique Nucléaire et Corpusculaire
M. COUTURIER Daniel	Chimie Organique
M. CRAMPON Norbert	Tectolique Géodynamique
M. CROSNIER Yves	Electronique
M. CURGY Jean-Jacques	Biologie
Melle DACHARRY Monique	Géographie
M. DEBRABANT Pierre	Géologie Appliquée
M. DEGAUQUE Pierre	Electronique
M. DEJAEGER Roger	Electrochimie et Cinétique
M. DELAHAYE Jean-Paul	Informatique
M. DELORME Pierre	Physioliogie Animale
M. DELORME Robert	Sciences Economiques
M. DEMUNTER Paul	Sociologie
M. DENEL Jacques	Informatique
M. DE PARIS Jean Claude	Analyse
M. DEPREZ Gilbert	Physique du Solide - Cristallographie
M. DERIEUX Jean-Claude	Microbiologie
Melle DESSAUX Odile	Spectroscopie de la réactivité Chimique
M. DEVRAINNE Pierre	Chimie Minérale
Mme DHAINAUT Nicole	Biologie Animale
M. DHAMELINCOURT Paul	Chimie Physique
M. DORMARD Serge	Sciences Economiques
M. DUBOIS Henri	Spectroscopie Hertzienne
M. DUBRULLE Alain	Spectroscopie Hertzienne
M. DUBUS Jean-Paul	Spectrométrie des Solides
M. DUPONT Christophe	Vie de la firme (I.A.E.)
Mme EVRARD Micheline	Génie des procédés et réactions chimiques
M. FAKIR Sabah	Algèbre
M. FAUQUAMBERGUE Renaud	Composants électroniques

M. FONTAINE Hubert	Dynamique des cristaux
M. FOUQUART Yves	Optique atmosphérique
M. FOURNET Bernard	Biochimie Sturcturale
M. GAMBLIN André	Géographie urbaine, industrielle et démog.
M. GLORIEUX Pierre	Physique moléculaire et rayonnements Atmos.
M. GOBLOT Rémi	Algèbre
M. GOSSELIN Gabriel	Sociologie
M. GOUDMAND Pierre	Chimie Physique
M. GOURIEROUX Christian	Probabilités et Statistiques
M. GREGORY Pierre	I.A.E.
M. GREMY Jean-Paul	Sociologie
M. GREVET Patrice	Sciences Economiques
M. GRIMBLOT Jean	Chimie Organique
M. GUILBAULT Pierre	Physiologie animale
M. HENRY Jean-Pierre	Génie Mécanique
M. HERMAN Maurice	Physique spatiale
M. HOUDART René	Physique atomique
M. JACOB Gérard	Informatique
M. JACOB Pierre	Probabilités et Statistiques
M. Jean Raymond	Biologie des populations végétales
M. JOFFRE Patrick	Vie de la firme (I.A.E.)
M. JOURNEL Gérard	Spectroscopie hertzienne
M. KREMBEL Jean	Biochimie
M. LANGRAND Claude	Probabilités et statistiques
M. LATTEUX Michel	Informatique
Mme LECLERCQ Ginette	Catalyse
M. LEFEBVRE Jacques	Physique
M. LEFEBVRE Christian	Pétrologie
Melle LEGRAND Denise	Algèbre
Melle LEGRAND Solange	Algèbre
M. LEGRAND Pierre	Chimie
Mme LEHMANN Josiane	Analyse
M. LEMAIRE Jean	Spectroscopie hertzienne
M. LE MAROIS Henri	Vie de la firme (I.A.E.)
M. LEROY Yves	Composants électroniques
M. LESENNE Jacques	Systèmes électroniques
M. LHENAFF René	Géographie
M. LOCQUENEUX Robert	Physique théorique
M. LOSFELD Joseph	Informatique
M. LOUAGE Francis	Electronique
M. MAHIEU Jean-Marie	Optique-Physique atomique
M. MAIZIERES Christian	Automatique
M. MAURISSON Patrick	Sciences Economiques et Sociales
M. MESMACQUE Gérard	Génie Mécanique
M. MESSELYN Jean	Physique atomique et moléculaire
M. MONTEL Marc	Physique du solide
M. MORCELLET Michel	Chimie Organique
M. MORTREUX André	Chimie Organique
Mme MOUNIER Yvonne	Physiologie des structures contractiles
Mme MOUYART-TASSIN Annie Françoise	Informatique
M. NICOLE Jacques	Spectrochimie
M. NOTELET Françis	Systèmes électroniques
M. PARSY Fernand	Mécanique

M. PECQUE Marcel	Chimie organique
M. PERROT Pierre	Chimie appliquée
M. STEEN Jean-Pierre	Informatique

C'est grâce à Monsieur le Professeur C. BREZINSKI que j'ai découvert et apprécié l'accélération de la convergence. Aussi je le remercie tout particulièrement de l'honneur qu'il me fait en jugeant mon travail et en présidant ce jury.

Monsieur le Professeur J. DELLA DORA a accepté de s'intéresser à ce travail et de le juger, je lui en suis très reconnaissante.

De nombreux résultats de cette thèse ont pour origine des travaux de Monsieur le Professeur J.P. DELAHAYE. Je le remercie d'être revenu à l'Analyse Numérique en acceptant de faire partie de ce jury.

Mes remerciements vont aussi à Monsieur le Professeur P. MOREL qui, par sa présence, donne une coloration bordelaise à ce jury.

Etant étudiante, j'ai beaucoup apprécié les enseignements de Monsieur le Professeur P. SABLONNIERE. Je le remercie chaleureusement pour ses encouragements sympathiques et pour avoir accepté de participer à ce jury.

Enfin je voudrais exprimer ma plus sincère et chaleureuse gratitude à Monsieur B. GERMAIN-BONNE. C'est avec beaucoup de compétence et de gentillesse qu'il m'a guidée et épaulée au cours de ce travail. Il n'aura ménagé ni son temps ni son ardeur et aura toujours su répondre à mes interrogations et dissiper mes doutes.

à Igor

ACCELERATION DE LA CONVERGENCE

DES

ENSEMBLES SYNCHRONISABLES

Anne-Marie LITOVSKY

SOMMAIRE

INTRODUCTION

En analyse numérique, l'accélération de la convergence des suites numériques se propose de construire algorithmiquement, à partir d'une suite (S_n) convergente, une suite (A_n) convergeant plus vite que (S_n) vers la même limite S^*, c'est-à-dire vérifiant le critère d'accélération suivant :

$$\lim_{n \to +\infty} \frac{A_n - S^*}{S_n - S^*} = 0.$$

Notons que ce critère d'accélération de la convergence suppose que l'on ne considère que des suites de l'ensemble C^*, c'est-à-dire des suites qui sont, à partir d'un certain rang, toujours différentes de leur limite.

Une information purement numérique ne suffit pas pour accélérer la convergence d'une suite et de nombreuses conditions suffisantes ont été trouvées (concernant notamment la façon de converger) qui permettent de construire des "procédés d'accélération" pour des familles de suites : un procédé d'accélération d'une famille E de suites de C^* est une transformation algorithmique définie sur E qui transforme toute suite (S_n) de E en une suite qui converge plus vite que (S_n).

Dans [5], B. Germain-Bonne et C. Kowalewski définissent les familles de suites "synchronisables" : synchroniser une famille E incluse dans C^*, c'est trouver une transformation algorithmique définie sur E qui transforme toute suite (S_n) de E en une suite synchrone avec (S_n), c'est-à-dire qui converge vers la même limite et qui vérifie le critère de synchronisation suivant :

$$\lim_{n \to +\infty} \frac{T_n - S^*}{S_n - S^*} = \tau, \text{ avec } \tau \neq 1.$$

Cela signifie simplement que la limite du rapport $(T_n-S^*)/(S_n-S^*)$ existe et est différente de 1, mais on n'a à priori aucune information sur la valeur de cette limite. (Notons qu'étant donnée la terminologie "synchronisation", il peut sembler étonnant d'éliminer la valeur $\tau = 1$. Mais cela se justifie par le fait que, si l'on admettait $\tau = 1$, tout ensemble de suites serait synchronisable par le procédé défini par : $T_n = S_n$!).

Il est clair que, si l'on possède un procédé d'accélération d'une famille E, ce procédé synchronise à fortiori E. Le résultat principal de ce travail consiste à démontrer la réciproque, c'est-à-dire que tout ensemble synchronisable est accélérable. Plus précisément nous construisons de manière algorithmique un procédé d'accélération de E (que nous appelons le ACCES-algorithme) à partir d'un procédé de synchronisation de la famille E. En particulier, dans le cas des suites à convergence linéaire synchronisées par le procédé $T_n = S_{n-1}$, le ACCES-algorithme redonne exactement le δ^2 d'Aitken !

Pour la construction du ACCES-algorithme, la difficulté réside dans le fait que, pour toute suite (S_n) de E, la valeur τ (qui dépend de (S_n)) est inconnue. Si cette valeur était connue, il suffirait de considérer le procédé qui transforme (S_n) en la suite (A_n) définie par $A_n = (\tau S_n - T_n)/(\tau-1)$. L'idée du ACCES-algorithme consiste à déterminer une estimation de τ. Pour cela, il a été déterminant, pour toute suite (S_n) de E et pour tout entier n suffisamment grand, de savoir placer S_n par rapport à la limite S^*, c'est-à-dire de savoir répondre à la question "quel est le signe de S_n-S^* ?". Nous montrons que, dès qu'un ensemble est synchronisable, nous savons répondre à cette question.

Notre travail se divise en deux chapitres :

le chapitre I est consacré à l'étude de la détermination du signe de S_n-S^* qui nous a semblé intéressante en elle-même (sans hypothèse de synchronisation).

Dans le chapitre II, nous montrons l'équivalence entre synchronisation et accélération de la convergence grâce à la construction du ACCES-algorithme.

CHAPITRE I.

PROBLEME

DE LA DECIDABILITE ASYMPTOTIQUE

DE LA QUESTION Q "QUEL EST LE SIGNE DE $S_n - S^*$?"

SUR DES SOUS-ENSEMBLES DE C^*

INTRODUCTION

L'étude de l'accélération de la convergence et de la synchronisation des suites numériques convergentes suppose que l'on ne considère que des suites qui sont, à partir d'un certain rang, toujours différentes de leur limite. Nous appelons C^* l'ensemble de ces suites.

La possibilité de répondre à la question Q "quel est le signe de S_n-S^* ?" où S^* est la limite - de valeur inconnue ! - d'une suite quelconque d'un ensemble synchronisable est une étape importante pour montrer l'équivalence entre synchronisation et accélération de la convergence (cf. chapitre II). Dans ce chapitre, nous étudions cette question sans l'hypothèse de synchronisation.

Etant donné un sous-ensemble E de C^*, s'il existe une transformation algorithmique normale τ telle que, pour toute suite (S_n) de E, la suite $\tau((S_n))$ donne à partir d'un certain rang la réponse à la question Q, nous dirons que la question Q est asymptotiquement décidable sur E (asymptotiquement indécidable dans le cas contraire).

Sans aucun renseignement sur une suite (S_n) de C^*, il est clair intuitivement que la donnée de S_0, ..., S_n ne nous permet pas de situer S^* par rapport à S_n, c'est-à-dire que l'on ne peut pas répondre à la question Q sur C^*. En revanche, un renseignement tel que "la suite est monotone" ou "la suite est alternée" permet, à partir des deux premiers termes de la suite de situer S^* par rapport à S_n pour tout entier n et donc donne naissance à un algorithme de décidabilité de la question Q. De manière plus générale, quels renseignements permettent de bâtir un algorithme de décidabilité ? Ce premier chapitre apporte une réponse partielle à cette question. Volontairement nous avons défini des ensembles de suites par des propriétés portant uniquement sur la position d'un nombre fini de termes de la suite par rapport à sa limite, ainsi les propriétés "la suite est monotone" ou "la suite est alternée" apparaissent naturellement comme des cas particuliers.

Sur certains ensembles la question Q est asymptotiquement décidable, sur d'autres elle est asymptotiquement indécidable. De plus, si la question Q est asymptotiquement décidable (resp. indécidable) sur un ensemble, alors elle est asymptotiquement décidable sur tout sous-ensemble (resp. indécidable sur tout sur-ensemble). Nous allons donc chercher à prouver soit la décidabilité asymptotique de la question Q sur des sous-ensembles de C* aussi "grands" que possible, soit la non décidabilité asymptotique de la question Q sur des sous-ensembles de C* aussi "petits" que possible, pour essayer de définir une frontière entre le décidable et l'indécidable.

Nous définissons dans la partie B deux ensembles C1 et C2 qui forment une partition de C* : C1 est l'ensemble des suites dont les termes, à partir d'un certain rang, sont tous d'un même côté de la limite; C2 est son complémentaire. Nous montrons que la question Q est asymptotiquement indécidable sur chacun des ensembles C1 et C2 (et donc à fortiori sur C*).

Nous prouvons ensuite des résultats de décidabilité pour des sous-ensembles de C1 (partie C), tandis que, à une exception près - l'ensemble **ALT** des suites asymptotiquement alternées - nous ne prouvons que des résultats d'indécidabilité sur des sous-ensembles de C2 (partie D). Dans la partie E, nous complétons ces résultats pour des sous-ensembles de C* "à cheval" sur C1 et C2.

A. DEFINITIONS

Donnons pour commencer quelques définitions qui, pour l'essentiel, sont dues à J.P. Delahaye [2].

Si R^N désigne l'ensemble des suites numériques, une transformation de suites τ dans R^N est une fonction de R^N dans R^N. Notons D_τ l'ensemble de définition de τ et $\tau(S_n)$ la suite transformée par τ de la suite (S_n).

Lorsqu'il existe un algorithme A tel que, pour toute suite $(S_n) \in D_\tau$, le n-ième terme de la suite $\tau(S_n)$ est calculé par A à partir des n premiers termes de la suite (S_n), τ sera appelée une transformation algorithmique normale ou plus brièvement une transformation algorithmique.

Soit $E \subset R^N$. Une question Q sur E est une transformation de suites définie sur E à valeurs dans REP^N, où REP désigne l'ensemble des réponses possibles.

Q est asymptotiquement décidable sur E si et seulement si il existe une transformation algorithmique normale τ définie sur E telle que :

$$\forall\ (S_n) \in E,\ \exists\ N,\ \forall\ n \geq N : Q((S_n))_n = \tau((S_n))_n.$$

Nous dirons que la suite des réponses (à la question Q) données par τ est asymptotiquement exacte ou exacte à partir d'un certain rang.

Notons C^* le sous-ensemble de R^N constitué par les suites convergentes dont les termes sont tous différents de la limite à partir d'un certain rang. Plus précisément, si (S_n) désigne une suite convergente de limite S^*, nous avons:

$$(S_n) \in C^* \iff \exists\ N,\ \forall\ n \geq N : S_n \neq S^*.$$

Nous utiliserons ici la notion de décidabilité asymptotique sur des sous-ensembles E de C^* pour la question Q "quel est le signe de S_n-S^* ?".

Et donc dire que la question Q est asymptotiquement décidable sur E signifie qu'il existe τ telle que, pour toute suite (S_n) de E, la suite (R_n) des réponses fournies par τ vérifie :

1) $\forall\ n \in N,\ R_n \in \{-1,+1\}$,
2) $\exists\ N,\ \forall\ n \geq N : R_n\ (S_n-S^*) > 0$.

Autrement dit R_n donne le signe de S_n-S^* pour tout $n \geq N$.

Tout algorithme permettant le calcul de la suite des réponses sera appelé algorithme de décidabilité (asymptotique).

Montrer que la question Q est asymptotiquement indécidable sur E revient à prouver qu'il n'existe pas d'algorithme de décidabilité de Q sur E. Et nous serons amené à utiliser le lemme suivant :

<u>Lemme 0.</u>

**Soit E $\subset$ C* et A un algorithme de décidabilité sur E.
Si (U_n) et (V_n) sont deux suites de E telles que :
$\exists$ m, $\forall$ n $\leq$ m : $U_n = V_n$,
alors A donne, jusqu'à l'indice m, la même réponse pour les deux suites (U_n) et (V_n).**

B. DEUX ENSEMBLES SUR LESQUELS LA QUESTION Q EST ASYMPTOTIQUEMENT INDECIDABLE

Commençons par définir deux sous-ensembles de C^*, $C1$ et $C2$, qui forment une partition de C^*.

Soit (S_n) une suite de C^*, et S^* sa limite.

- ou bien, à partir d'un certain rang, tous les termes de la suite sont d'un même côté de S^* : on dira que la suite appartient à $C1$,

- ou bien la suite a une infinité de termes inférieurs à S^* et une infinité de termes supérieurs à S^* : on dira que la suite appartient à $C2$.

Nous avons donc les équivalences suivantes, en notant $e_n = S_n - S^*$:

$$(S_n) \in C1 \iff \text{(i)} \quad \exists N, \forall n > N : e_n e_{n-1} > 0$$

$$(S_n) \in C2 \iff \begin{array}{c} (S_n) \in C^* \\ \text{et} \\ \text{(ii)} \quad \forall N, \exists n > N : e_n e_{n-1} < 0 \end{array}$$

Remarquons que la condition (i) implique que $(S_n) \in C^*$, alors que la condition (ii) n'entraîne pas que $(S_n) \in C^*$.

Nous montrons, dans cette première partie, que la question Q est asymptotiquement indécidable sur les ensembles $C1$ et $C2$, ce qui entraîne, à fortiori, que Q est asymptotiquement indécidable sur C^*.

1. ENSEMBLE C1

Une suite (S_n) de $C1$ de limite S^* peut avoir, à partir d'un certain rang, tous ses termes inférieurs à S^* et, dans ce cas, l'erreur $e_n = S_n - S^*$ est asymptotiquement négative : nous dirons que la suite appartient à $C1^-$. Sinon elle a, à partir d'un certain rang, tous ses termes supérieurs à S^*, l'erreur e_n étant alors asymptotiquement positive : nous dirons que la suite appartient à $C1^+$.

$$(S_n) \in C1^- \iff \exists N, \forall n \geq N : S_n - S^* < 0.$$

$(S_n) \in C1^+ \iff \exists N, \forall n \geq N : S_n - S^* > 0.$

Théorème 1.

La question "quel est le signe de S_n-S^* ?" est asymptotiquement indécidable sur l'ensemble C1.

PREUVE

Faisons un raisonnement par l'absurde.
Supposons que la question "quel est le signe de S_n-S^* ?" soit asymptotiquement décidable sur C1, c'est-à-dire qu'il existe un algorithme, noté Φ, qui transforme toute suite (S_n) de C1 en une suite (R_n) telle que :

$\forall n \geq 0, R_n \in \{-1,+1\}$ et

$\exists N, \forall n \geq N : R_n (S_n-S^*) > 0.$

Or, pour toute suite (S_n) de C1, le signe de l'erreur $e_n = S_n-S^*$ est asymptotiquement constant.
Donc, l'algorithme Φ fournit, pour toute suite (S_n) de C1, une suite (R_n) telle que :

$\exists N, \forall n > N : R_n = R_{n-1}.$

et, en particulier, l'algorithme Φ fournit, pour toute suite (S_n) de $C1^+$, une suite (R_n) telle que :

(1) $\exists N, \forall n \geq N : R_n = +1$

Nous allons alors construire une suite (S_n) de $C1^+$ strictement positive, convergeant vers zéro, formée d'une succession de débuts de suites monotones croissantes (donc appartenant à $C1^-$) convergeant vers des valeurs de plus en plus petites ($1/2, 1/4, \ldots, 1/2^p, \ldots$), si bien que la suite (R_n) des réponses fournies par l'algorithme Φ appliqué à la suite (S_n) ne vérifiera pas la propriété (1), c'est-à-dire comportera une infinité de réponses fausses.

Construction des suites monotones croissantes :

Par récurrence sur l'entier p, $p > 0$, nous construisons une suite de suites $(T_{p,n})$ asymptotiquement monotones croissantes (donc appartenant à $C1^-$), telles que :

$\forall p > 0, \lim_{n\to+\infty} T_{p,n} = 1/2^p$

La suite $(T_{1,n})$ est définie par :

$T_{1,0} = 1$

$T_{1,1} = 1/4$

$T_{1,n} = (T_{1,n-1} + 1/2)/2$ pour tout $n > 1$

La suite $(T_{1,n})$ appartient à $C1^-$; elle converge vers 1/2 en croissant.
Désignons par $R_{1,n}$ la réponse fournie à l'étape n par l'algorithme Φ appliqué à la suite $(T_{1,n})$, alors :

il existe un entier $\beta(1) > 1$ tel que : $R_{1,\beta(1)} = -1$

La suite $(T_{2,n})$ est alors définie par :

$T_{2,n} = T_{1,n}$ pour tout $n \leq \beta(1)$

$T_{2,\beta(1)+1} = 1/8$

$T_{2,n} = (T_{2,n-1} + 1/4)/2$ pour tout $n > \beta(1)+1$

La suite $(T_{2,n})$ appartient à $C1^-$; elle converge vers 1/4 en croissant.

Désignons par $R_{2,n}$ la réponse fournie à l'étape n par l'algorithme Φ appliqué à la suite $(T_{2,n})$, alors :

il existe un entier $\beta(2) > \beta(1)$ tel que : $R_{2,\beta(2)} = -1$

Supposons maintenant la suite $(T_{p-1,n})$ construite. La suite $(T_{p,n})$ sera alors définie par :

$T_{p,n} = T_{p-1,n}$ pour tout $n \leq \beta(p-1)$

$T_{p,\beta(p-1)+1} = 1/2^{p+1}$

$T_{p,n} = (T_{p,n-1} + 1/2^p)/2$ pour tout $n > \beta(p-1)+1$

La suite $(T_{p,n})$ appartient à $C1^-$; elle converge vers $1/2^p$ en croissant.

Désignons par $R_{p,n}$ la réponse fournie à l'étape n par l'algorithme Φ appliqué à la suite $(T_{p,n})$. On a :

il existe un entier $\beta(p) > \beta(p-1)$ tel que :

$R_{p,\beta(p)} = -1$

Construction de la suite (S_n) :

Posons pour tout entier $p > 0$ et pour tout entier $n \leq \beta(p)$

$S_n = T_{p,n}$

La suite (S_n) ainsi construite vérifie :

$\forall\, n \geq 0,\ S_n > 0$

$\forall\, p > 0,\ \forall\, n \geq \beta(p) : 0 < S_n < 1/2^p$

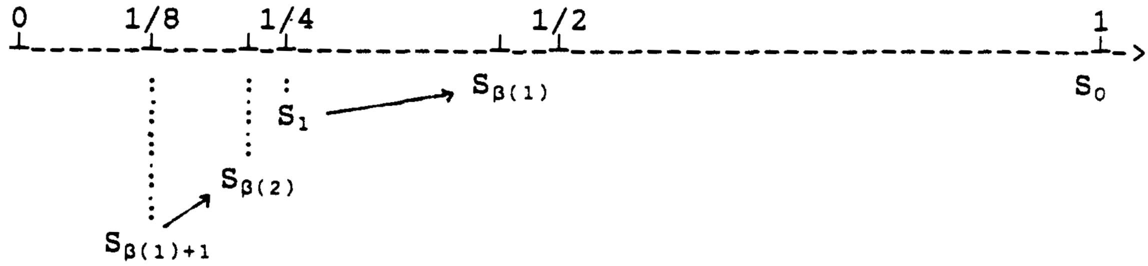
Donc la suite (S_n) converge vers zéro et appartient à $C1^+$.

Si on applique l'algorithme Φ à la suite (S_n) et si on désigne par R_n la réponse fournie à l'étape n, on a d'après le lemme 0, pour tout entier $p > 0$:

$R_{\beta(p)} = R_{p,\beta(p)} = -1$

et donc la suite (R_n) des réponses fournies par l'algorithme Φ appliqué à la suite (S_n) ainsi construite ne vérifie pas la propriété (1).■

Illustration de la construction de la suite (S_n) :



<u>Corollaire 1.</u>

> La question "quel est le signe de $S_n - S^*$?" est asymptotiquement indécidable sur l'ensemble C^*.

<u>2. ENSEMBLE C2</u>

<u>Théorème 2.</u>

> La question "quel est le signe de $S_n - S^*$?" est asymptotiquement indécidable sur l'ensemble C2.

PREUVE

Comme pour la proposition précédente, faisons un raisonnement par l'absurde.
Supposons que la question "quel est le signe de $S_n - S^*$?" soit asymptotiquement décidable sur C2.
Alors il existe un algorithme Φ qui transforme toute suite (S_n) de C2 en une suite (R_n) telle que :

$\forall\, n \geq 0,\ R_n \in \{-1,+1\}$ et

(1) $\quad \exists\, N,\ \forall\, n \geq N : R_n\,(S_n - S^*) > 0.$

Nous allons alors construire une suite (S_n) de C2, de telle sorte que, pour les termes de rang pair, l'algorithme Φ ne donne jamais la bonne réponse. Ainsi nous aurons :

$\forall\, p > 0,\ R_{2p}\,(S_{2p} - S^*) < 0$

ce qui contredit la propriété (1).

La construction de (S_n) utilise une suite de suites $(T_{p,n})$ $(p > 0)$ asymptotiquement alternées (donc appartenant à C2) que l'on construit par récurrence sur l'entier p.

Construction des suites $(T_{p,n})$ asymptotiquement alternées :

La suite $(T_{1,n})$ est définie par :

$T_{1,0} = -2$

$T_{1,1} = +2$
$T_{1,2} = 0$
$T_{1,n} = (-1)^n/n$ pour tout $n > 2$

Elle est asymptotiquement alternée et elle converge vers zéro.
Notons $R_{1,n}$ la réponse fournie à l'étape n par l'algorithme Φ appliqué à la suite $(T_{1,n})$. La valeur de $R_{1,2}$ va nous permettre de définir la suite $(T_{2,n})$.

La suite $(T_{2,n})$ est définie par :

$T_{2,n} = T_{1,n}$ pour tout $n \leq 2$
$T_{2,3} = +1$ si $R_{1,2} = +1$
$\quad\quad -1$ si $R_{1,2} = -1$
$T_{2,4} = (T_{2,2} + T_{2,3})/2$
$T_{2,n} = (-1)^n/n$ pour tout $n > 4$

Illustration de la construction de $T_{2,n}$, $n \leq 4$:

a) lorsque $R_{1,2} = -1$: dans ce cas l'algorithme Φ indique que la limite de la suite $(T_{1,n})$ est à gauche de $T_{1,2}$.

```
 T1,0                      T1,2                       T1,1
--+-----------+-----------+-----+-----+-----------+------->
 -2          -1           0           +1          +2
 T2,0                     T2,2  T2,4  T2,3        T2,1
```

b) lorsque $R_{1,2} = +1$: dans ce cas l'algorithme Φ indique que la limite de la suite $(T_{1,n})$ est à droite de $T_{1,2}$.

```
 T1,0                      T1,2                       T1,1
--+-----------+-----+-----+-----------+-----------+------->
 -2          -1           0           +1          +2
 T2,0        T2,3  T2,4   T2,2                    T2,1
```

Supposons maintenant que l'on ait construit la suite $(T_{p,n})$.
Elle est telle que $T_{p,2p}$ est le milieu du segment $\langle T_{p,2p-1}, T_{p,2p-2}\rangle$.
La réponse $R_{p,2p}$ fournie à l'étape 2p par l'algorithme Φ appliqué à la suite $(T_{p,n})$ va nous permettre de définir la suite $(T_{p+1,n})$.
Notons a et b, avec $a < b$, les milieux des segments $\langle T_{p,2p-2}, T_{p,2p}\rangle$ et $\langle T_{p,2p-1}, T_{p,2p}\rangle$.

La suite $(T_{p+1,n})$ est définie par :

$T_{p+1,n} = T_{p,n}$ pour tout $n \leq 2p$
$T_{p+1,2p+1} = b$ si $R_{p,2p} = +1$
$\quad\quad = a$ si $R_{p,2p} = -1$

$T_{p+1,2p+2} = (T_{p+1,2p} + T_{p+1,2p+1})/2$
$T_{p+1,n} = (-1)^n/n$ pour tout $n > 2p+2$

remarque : pour tout entier $p > 0$, le choix de $T_{p,n}$ pour tout $n > 2p$ n'a aucune importance, il suffit que la suite $(T_{p,n})$ appartienne à C2.

Illustration de la construction de $T_{p+1,n}$ pour $n \le 2p+2$:

a) lorsque $R_{p,2p} = -1$: dans ce cas l'algorithme Φ indique que la limite de la suite $(T_{p,n})$ est à gauche de $T_{p,2p}$.

```
Tp,2p-2            a              Tp,2p           b              Tp,2p-1
--┴-----------┴-----------┴-----┴-----┴-----------┴------->
Tp+1,2p-2                  Tp+1,2p   :     :              Tp+1,2p-1
                                     :     :
                              Tp+1,2p+2  Tp+1,2p+1
```

b) lorsque $R_{p,2p} = +1$:
l'algorithme Φ indique que la limite de la suite $(T_{p,n})$ est à droite de $T_{p,2p}$.

```
Tp,2p-2            a              Tp,2p           b              Tp,2p-1
--┴-----------┴-----┴-----┴-----------┴-----------┴------->
Tp+1,2p-2          :     :  Tp+1,2p                          Tp+1,2p-1
                   :     :
          Tp+1,2p+1  Tp+1,2p+2
```

Construction de la suite (S_n) :

posons pour tout entier $p > 0$ et pour tout entier $n \le 2p$:
$S_n = T_{p,n}$

La suite (S_n) ainsi construite est convergente (théorème des intervalles emboités [9]) de limite S^* avec :
$\forall\ p \ge 0,\ S^* \in \langle S_{2p}, S_{2p+1} \rangle$
Donc la suite (S_n) appartient à C2.

Illustration de la construction de la suite (S_n) :

```
--┴-----------┴-----┴-----┴-----------┴-----------┴------->
S2p-2        S2p+1  S2p+2  S2p                         S2p-1

--┴-----------┴-----------┴-----┴-----┴-----------┴------->
S2p-2                    S2p   S2p+2  S2p+1            S2p-1
```

Si on applique l'algorithme Φ à la suite (S_n) et si on désigne par R_n la réponse fournie à l'étape n, on a d'après le lemme 0 :

$\forall\ p \geq 0,\ R_{2p} = R_{p,2p}$

et par construction de la suite (S_n), on a :

$\forall\ p \geq 0,\ R_{2p}\ (S_{2p}-S^*) < 0$

Donc la suite (S_n) ne vérifie pas la propriété (1).■

C^* est naturellement partitionné en deux sous-ensembles C1 et C2 dont les prototypes sont respectivement les suites monotones et les suites alternées. Les démonstrations des théorèmes 1 et 2 utilisent d'ailleurs de telles suites. Mais la situation n'est pas réellement symétrique entre C1 et C2. D'ores et déjà deux différences sont apparues :
- d'une part, au niveau de la définition, la condition (i) qui définit C1 entraîne que C1 est inclus dans C^*, alors que la seule condition (ii) ne définirait pas C2 comme sous-ensemble de C^*,
- d'autre part, la partition naturelle de C1 en $C1^+$ et $C1^-$, qui s'est avérée utile dans la preuve du théorème 1, n'a pas d'équivalent dans C2.

Nous verrons d'autres différences dans la suite.

C. ETUDE DE SOUS-ENSEMBLES DE C1

Dans cette deuxième partie nous étudions la décidabilité asymptotique de la question Q "quel est le signe de S_n-S^* ?" sur des sous-ensembles de l'ensemble C1. Rappelons que :

$$(S_n) \in C1 \iff \exists N, \forall n > N : (S_n-S^*)(S_{n-1}-S^*) > 0$$

Nous avons déjà défini deux sous-ensembles de C1, $C1^-$ et $C1^+$, qui forment une partition de C1. Une suite de $C1^-$ (resp. $C1^+$) est caractérisée par le fait que la différence S_n-S^* est asymptotiquement négative (resp. positive).

$$(S_n) \in C1^- \iff \exists N, \forall n \geq N : S_n-S^* < 0.$$

$$(S_n) \in C1^+ \iff \exists N, \forall n \geq N : S_n-S^* > 0.$$

La question Q est évidemment asymptotiquement décidable sur chacun des ensembles $C1^-$ et $C1^+$: pour l'ensemble $C1^-$ l'algorithme de décidabilité propose à chaque étape n la réponse $R_n = -1$ tandis que, pour l'ensemble $C1^+$, l'algorithme de décidabilité propose à chaque étape n la réponse $R_n = +1$.

Nous nous intéresserons alors à la construction de sous-ensembles de C1 inclus ni dans $C1^-$ ni dans $C1^+$ puis nous étudierons la décidabilité asymptotique de la question Q sur ces ensembles. Remarquons qu'étudier la décidabilité asymptotique de la question Q sur un sous-ensemble E de C1 revient, pour toute suite de E, à décider "asymptotiquement" si elle appartient à $C1^-$ ou à $C1^+$.

1. DECIDABILITE ASYMPTOTIQUE DE LA QUESTION Q SUR L'ENSEMBLE $C1^•$

Considérons le sous-ensemble de C1, noté $C1^•$, formé par les suites (S_n) dont tous les termes sont d'un même côté de la limite S^* (c'est-à-dire pour lesquels l'erreur $e_n = S_n-S^*$ a un signe constant pour tout $n \geq 0$).

$$(S_n) \in C1^• \iff \forall n > 0, \quad e_n\, e_{n-1} > 0.$$

Posons, pour toute suite (S_n) :
$U_n = \min (S_0,\ldots,S_n)$
$V_n = \max (S_0,\ldots,S_n)$.

La suite (U_n) est monotone décroissante et la suite (V_n) est monotone croissante.

<u>Lemme 1.</u>

Pour toute suite (S_n) de $C1^{\bullet}$, nous avons les équivalences suivantes :

a) $(S_n) \in C1^- \Leftrightarrow$ la suite (U_n) est stationnaire mais la suite (V_n) est non stationnaire.

b) $(S_n) \in C1^+ \Leftrightarrow$ la suite (V_n) est stationnaire mais la suite (U_n) est non stationnaire.

PREUVE

Faisons par exemple la preuve de a).
Pour toute suite (S_n) convergente, nous avons :

$$(S_n) \in C1^{\bullet} \cap C1^- \Leftrightarrow \forall n \geq 0,\ S_n - S^* < 0$$

alors comme $S_0 < S^*$ et que la suite (S_n) est convergente :

$$\exists N,\ \forall n > N : S_0 \leq S_n < S^*$$

et donc :

$$\forall n > N,\ S_0 \leq \min(S_{N+1}, \ldots, S_n)$$

or :

$$U_N = \min(S_0, \ldots, S_N) \leq S_0$$

d'où :

$$\begin{aligned} \forall n \geq N,\ U_n &= \min(S_0, \ldots, S_N, S_{N+1}, \ldots, S_n) \\ &= \min(U_N, S_{N+1}, \ldots, S_n) \\ &= U_N \end{aligned}$$

et la suite (U_n) est bien stationnaire.

Supposons que la suite (V_n) soit elle aussi stationnaire c'est-à-dire :

$$\exists N,\ \forall n \geq N : V_n = V_N$$

la suite (S_n) appartenant à $C1^{\bullet} \cap C1^-$, nous avons :

$$V_N < S^*$$

d'autre part, pour tout $n \geq N$:

$$S_n \leq V_n = V_N < S^*$$

et, comme la suite (S_n) converge vers S^*, nous obtenons :

$$S^* \leq V_N < S^*$$

ce qui est absurde.
donc la suite (V_n) n'est pas stationnaire.■

Ce lemme nous permet alors de construire un algorithme de décidabilité de la question Q sur $C1^{\bullet}$.

description de l'algorithme

```
| R0 <- +1;    {initialisation arbitraire}
| n <- 1;
|
| Répéter à l'infini
|
|    Un <- min (Un-1,Sn);
|    Vn <- max (Vn-1,Sn);
|    Si (Un = Un-1 et Vn = Vn-1)  alors  Rn <- Rn-1;
|    Si Un < Un-1  alors  Rn <- +1;
|    Si Vn > Vn-1  alors  Rn <- -1;
|    n <- n+1
|
| fin du répéter.
```

PREUVE DE L'ALGORITHME :

Soit (S_n) une suite de $C1^{\bullet} \cap C1^-$.

D'après le lemme 1 :

$\exists\ N, \forall\ n > N : U_n = U_{n-1}$

et :

$\exists\ N' \geq N : V_{N'} > V_{N'-1}$

alors l'algorithme donne, à l'étape N', la réponse :

$R_{N'} = -1$

montrons par récurrence que :

$\forall\ n \geq N' : R_n = -1$

c'est vrai pour $n = N'$.

supposons que, pour $n > N'$, $R_{n-1} = -1$

à l'étape n, deux cas se présentent :

ou bien $V_n = V_{n-1}$, et l'algorithme donne alors la réponse :

$R_n = R_{n-1}$

donc $R_n = -1$

ou bien $V_n > V_{n-1}$, et l'algorithme donne la réponse :

$R_n = -1$

d'où le résultat.

Le cas $(S_n) \in C1^{\bullet} \cap C1^+$ est analogue.■

Nous pouvons alors énoncer :

<u>Théorème 3.</u>

‖ La question "quel est le signe de S_n-S* ?" est asymptotiquement décidable sur l'ensemble $C1^{\bullet}$.

2. DECIDABILITE ASYMPTOTIQUE DE LA QUESTION Q SUR L'ENSEMBLE MON

Un sous-ensemble classique de C1 est l'ensemble - noté **MON** - des suites asymptotiquement monotones, pour lesquelles, à partir d'un certain rang, la différence entre deux termes consécutifs, $\delta S_n = S_{n+1} - S_n$, a un signe constant. Ce qui se traduit par :

$$(S_n) \in \mathbf{MON} \iff \exists N : \forall n \geq N,\ \delta S_n\, \delta S_{n+1} \geq 0$$

Lorsque, pour tout n, $n \geq N$, δS_n est positif, la suite est asymptotiquement monotone croissante et donc appartient à $C1^-$. Et, lorsque, pour tout n, $n \geq N$, δS_n est négatif, la suite est asymptotiquement monotone décroissante et donc appartient à $C1^+$.

Donc, pour toute suite (S_n) de **MON** et pour tout entier n assez grand, l'erreur $e_n = S_n - S^*$ a le même signe que $S_{n-1} - S_n$, i.e. $-\delta S_{n-1}$. Cela nous permet de proposer un algorithme de décidabilité asymptotique de la question Q "quel est le signe de $S_n - S^*$?" sur **MON** :

description de l'algorithme

```
| n <- 1
|
| Répéter à l'infini
|
|    Si  δS(n-1) ≠ 0  alors R(n) = signe (-δS(n-1))
|                     sinon R(n) <- R(n-1);
|    n <- n+1
|
| fin du répéter.
```

Et nous énonçons :

Proposition 1.

> La question Q est asymptotiquement décidable sur l'ensemble **MON**.

3. DEFINITIONS DES SUITES PROGRESSIVES

Nous commençons par donner une autre définition de **MON** qui utilise la notion d'indice de progression.

Si $(S_n) \in$ **MON**, pour tout $n \geq N$, S_{n+1} est plus proche que S_n de la limite : nous dirons que S_{n+1} est meilleur que S_n, ou que S_n est moins bon que S_{n+1}.

De manière plus générale, nous dirons :

DEFINITION 1 :

> Soient S_n et S_p deux termes d'une suite (S_n) de C^* de limite S^*.
> S_p est meilleur que S_n, ou bien S_n est moins bon que S_p, si et seulement si :
> 1) S_n et S_p sont d'un même côté de S^*
> 2) et l'erreur $e_p = S_p - S^*$ est inférieure ou égale, en valeur absolue, à l'erreur $e_n = S_n - S^*$.

Autrement dit, S_p est meilleur que S_n si et seulement si les erreurs e_p et e_n vérifient :

1) $e_p\, e_n > 0$.
2) $|e_p| \leq |e_n|$

En utilisant la notation :

$<S_n,S^*< \; = [S_n,S^*[$, si $S_n < S^*$

$\qquad\qquad]S^*,S_n]$, si $S_n > S^*$,

nous pouvons aussi écrire :

S_p est meilleur que S_n si et seulement si $S_p \in \; <S_n,S^*<$

DEFINITION 2 :

> Soit (S_n) une suite de C^* de limite S^*.
> On appelle indice de progression l'entier $\mu^+(n)$ défini par :
> $\mu^+(n) = \min (\{j > 0 \;/\; S_{n+j}$ est meilleur que $S_n\} \cup \{0\})$

Autrement dit :

$$\mu^+(n) = \min (\{j > 0 \;/\; S_{n+j} \in \; <S_n,S^*< \} \cup \{0\}).$$

Comme les suites de C^* sont convergentes, pour toute suite (S_n) de C^*, $\mu^+(n)$ est strictement positif pour tout entier n suffisamment grand.

L'ensemble **MON** apparaît donc comme l'ensemble des suites dont l'indice de progression est constant, égal à 1 à partir d'un certain rang.

$$(S_n) \in \mathbf{MON} \iff \exists N : \forall n \geq N,\ \mu^+(n) = 1$$

Cette définition peut être généralisée : pour tout entier q, $q \geq 1$, notons **q.PROG** l'ensemble des suites q-progressives pour lesquelles, à partir d'un certain rang, l'indice de progression est compris entre 1 et q. En d'autres termes une suite (S_n) q-progressive est telle que, pour tout n suffisamment grand, parmi les termes $S_{n+1}, S_{n+2}, \dots, S_{n+q}$, il en existe un meilleur que S_n.

DEFINITIONS 3 :

Soit q un entier non nul et (S_n) une suite convergente de limite S^*.

$(S_n) \in q.\mathrm{PROG} \iff \exists N, \forall n \geq N : 1 \leq \mu^+(n) \leq q$

notons : $*.\mathrm{PROG} = \bigcup_{q=1}^{+\infty} q.\mathrm{PROG}$

De façon plus détaillée, une suite (S_n) de l'ensemble q.PROG est caractérisée par :

$$\exists N, \forall n \geq N, \exists j \text{ tel que : } 1 \leq j \leq q \quad \text{et} \quad S_{n+j} \in \langle S_n, S^* \langle$$

Une suite de l'ensemble *.PROG sera appelée une suite progressive.

Remarquons que *.PROG est strictement inclus dans C^*, comme le montre l'exemple suivant.

EXEMPLE 0 :

Posons pour tout $n \geq 0$, $\theta(n) = n(n+1)/2$.

Soit (S_n) la suite définie par :

$S_{\theta(n)} = (-1)^{n+1}\, n^{-1}$,

$S_{\theta(n)+j} = S_{\theta(n)}$, pour tout j, $1 \leq j \leq n$

```
                                 0
---┴-----┴-----┴--------┴-------┴-----------┴-------┴--->
 S0     S3    Sθ(2n)            Sθ(2n+1)      S6      S1
      = S4  = Sθ(2n)+1        = Sθ(2n+1)+1    :     = S2
      = S5    :                 :           = S9
            = Sθ(2n)+2n       = Sθ(2n+1)+2n+1
```

Pour tout entier $n \geq 0$ et tout entier j, $0 \leq j < n$, on a :

$\mu^+(\theta(n)+n) = \theta(n+2) - (\theta(n)+n) = n+3$,

$\mu^+(\theta(n)+j) = 1$.

ce qui prouve que la suite (S_n) n'appartient à aucun ensemble q.PROG, donc n'appartient pas à *.PROG.■

Par définition de q.PROG, nous avons immédiatement :

<u>Proposition 2.</u>

1.PROG = MON

$\forall q \geq 1$, $q.\mathrm{PROG} \subset (q+1).\mathrm{PROG}$

Mais pour tout entier q, q≥2, q.PROG est un sous-ensemble de C* qui n'est inclus ni dans C1 ni dans C2. L'intersection de q.PROG avec C1 (resp. C2) sera notée q.PROG1 (resp. q.PROG2). De même l'intersection de *.PROG avec C1 (resp. C2) sera notée *.PROG1 (resp. *.PROG2).

DEFINITIONS 4 :

$$q.PROG1 = q.PROG \cap C1$$

$$*.PROG1 = *.PROG \cap C1 = \bigcup_{q=1}^{+\infty} q.PROG1$$

$$q.PROG2 = q.PROG \cap C2$$

$$*.PROG2 = *.PROG \cap C2 = \bigcup_{q=1}^{+\infty} q.PROG2$$

Dans cette troisième partie C où nous étudions les sous-ensembles de C1, nous étudierons uniquement q.PROG1 et *.PROG1, l'étude de q.PROG2 et *.PROG2 sera faite dans la partie D.

Nous avons les équivalences suivantes :

$$(S_n) \in q.PROG1 \cap C1^- \iff \exists N, \forall n \geq N, \exists j \text{ tel que } 1 \leq j \leq q \text{ et } S_n \leq S_{n+j} < S^*$$

$$(S_n) \in q.PROG1 \cap C1^+ \iff \exists N, \forall n \geq N, \exists j \text{ tel que } 1 \leq j \leq q \text{ et } S^* < S_{n+j} \leq S_n$$

Et nous avons : 1.PROG1 = MON

4. DEFINITIONS DES SUITES REGRESSIVES

Pour définir l'indice de progression $\mu^+(n)$, nous avons cherché, parmi les termes de la suite d'indice supérieur à n, ceux qui étaient "meilleurs" que S_n. Nous allons maintenant définir un indice de régression $\mu^-(n)$ en regardant, parmi les termes de la suite d'indice inférieur à n, ceux qui sont "moins bons" que S_n.

DEFINITION 5 :

Soit (S_n) une suite de C* de limite S*.
On appelle indice de régression l'entier $\mu^-(n)$ défini par :
$\mu^-(n) = \min (\{j > 0 \;/\; S_{n-j} \text{ est moins bon que } S_n\} \cup \{0\})$.

Autrement dit :

$\mu^-(n) = \min(\{j > 0 \ / \ S_n \in \langle S_{n-j}, S^*\langle \ \} \cup \{0\}).$

De même que pour $\mu^+(n)$, $\mu^-(n)$ est non nul pour tout entier n suffisamment grand.

Nous définissons alors, pour tout entier q, $q \geq 1$, l'ensemble q.REG des suites q-régressives pour lesquelles, à partir d'un certain rang, l'indice de régression est inférieur ou égal à q. En d'autres termes une suite (S_n) q-régressive est telle que, pour tout n suffisamment grand, parmi les termes S_{n-1}, S_{n-2}, ..., S_{n-q}, il en existe un moins bon que S_n.

DEFINITION 6 :

Soit q un entier non nul et (S_n) une suite convergente de limite S^*.

$(S_n) \in$ q.REG $\Leftrightarrow \exists N, \forall n \geq N : \mu^-(n) \leq q$

Notons $*.\mathrm{REG} = \bigcup_{q=1}^{+\infty} q.\mathrm{REG}$

De façon plus détaillée, une suite (S_n) de l'ensemble q.REG est caractérisée par :

$$\exists N, \forall n \geq N, \exists j \text{ tel que : } 1 \leq j \leq q \quad \text{et} \quad S_n \in \langle S_{n-j}, S^*\langle$$

Une suite de l'ensemble *.REG sera appelée une suite régressive.

Comme pour *.PROG, *.REG est strictement inclus dans C^*. Il suffit de reprendre l'exemple 0 dans lequel, pour tout $n > 0$, on a : $\mu^-(\theta(n+1)) = \theta(n+1) - (\theta(n-1)+n-1) = n+2$. La suite considérée n'appartenant ni à *.PROG, ni à *.REG, cela montre donc aussi que l'ensemble *.PROG U *.REG est strictement inclus dans C^*.

Nous avons immédiatement :

<u>Proposition 3.</u>

1.REG = MON

$\forall q \geq 1$, q.REG $\subset$ (q+1).REG

D'autre part, comme pour les suites q-progressives, pour tout entier q, $q \geq 2$, q.REG est un sous-ensemble de C^* qui n'est inclus ni dans C1 ni dans C2. Et nous pouvons définir les intersections de q.REG et *.REG avec C1 et C2.

DEFINITION 7 :

$$\left|\begin{array}{l} q.\mathbf{REG1} = q.\mathbf{REG} \cap \mathbf{C1} \\[6pt] *.\mathbf{REG1} = *.\mathbf{REG} \cap \mathbf{C1} = \bigcup_{q=1}^{+\infty} q.\mathbf{REG1} \\[6pt] q.\mathbf{REG2} = q.\mathbf{REG} \cap \mathbf{C2} \\[6pt] *.\mathbf{REG2} = *.\mathbf{REG} \cap \mathbf{C2} = \bigcup_{q=1}^{+\infty} q.\mathbf{REG2} \end{array}\right.$$

Nous laissons pour le paragraphe suivant l'étude de q.**REG2** et *.**REG2**.

Pour q.**REG1**, nous avons les équivalences suivantes :

$(S_n) \in q.\mathbf{REG1} \cap \mathbf{C1}^- \iff \exists N,\ \forall n \geq N,\ \exists j$ tel que $1 \leq j \leq q$ et $S_{n-j} \leq S_n < S^*$

$(S_n) \in q.\mathbf{REG1} \cap \mathbf{C1}^+ \iff \exists N,\ \forall n \geq N,\ \exists j$ tel que $1 \leq j \leq q$ et $S^* < S_n \leq S_{n-j}$

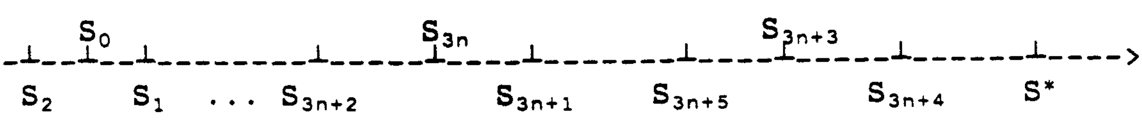
Et nous avons : **1.REG1 = MON**.

Les ensembles **1.REG1** et **1.PROG1** sont donc tous les deux égaux à **MON**. Une question se pose donc tout naturellement : pour $q \geq 2$, les ensembles q.**PROG1** et q.**REG1** sont-ils égaux, ou tout au moins comparables pour la relation d'inclusion ? Leur intersection est non vide, puisqu'ils contiennent tous les deux **MON**, mais aucun des deux n'est inclus dans l'autre, comme le montrent les exemples suivants.

EXEMPLE 1 :

Soit (S_n) une suite convergente de limite S^* telle que :

$$\forall n \geq 0,\ S_{3n+2} < S_{3n} < S_{3n+1} < S_{3n+5} < S^*$$



Pour tout entier $n \geq 0$, nous avons :

$\mu^+(3n) = 1$
$\mu^+(3n+1) = 2$
$\mu^+(3n+2) = 1$

et, pour tout entier $n > 0$, nous avons :

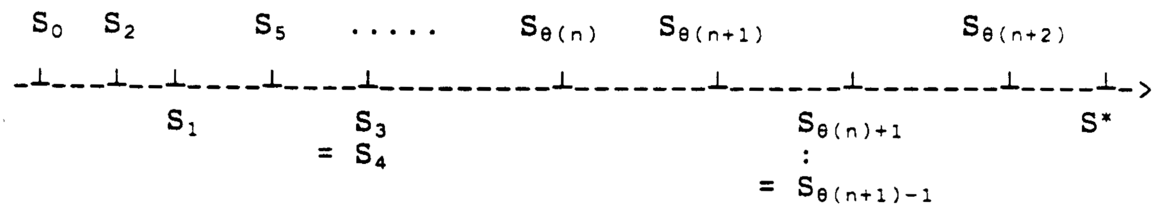
$\mu^-(3n) = 1$
$\mu^-(3n+1) = 1$
$\mu^-(3n+2) = 3$

Donc la suite (S_n) appartient à **2.PROG1** et à **3.REG1**, mais n'appartient pas à **2.REG1**.■

EXEMPLE 2 :

Posons, pour tout $n > 0$, $\theta(n) = (n-1)(n+2)/2$.
Soit (S_n) une suite convergente de limite S^* telle que, pour tout entier $n > 0$ et pour tous les entiers j et k compris strictement entre $\theta(n)$ et $\theta(n+1)$, on ait :

$$S_{\theta(n)} < S_{\theta(n+1)} < S_j = S_k < S_{\theta(n+2)} < S^*.$$



Pour tout entier $n > 0$, nous avons :

$$\mu^+(\theta(n)-1) = 2$$

et pour tout entier j tel que $j+1 \notin \theta(N^*)$:

$$\mu^+(j) = 1$$

D'autre part, pour tout entier $n > 1$, nous avons :

$$\mu^-(\theta(n)) = \theta(n) - \theta(n-1) = n$$

et pour tout entier $j \notin \theta(N^*)$:

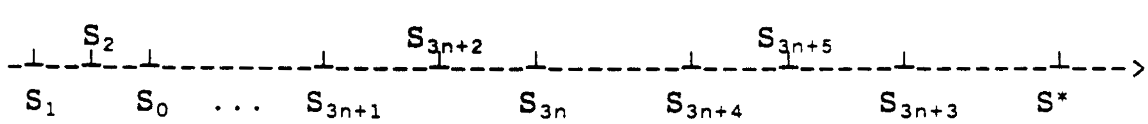
$$\mu^-(j) = 1$$

Donc la suite (S_n) appartient à **2.PROG1** mais n'appartient à aucun ensemble **q.REG1**, avec $q \geq 1$, donc n'appartient pas à ***.REG1**.■

EXEMPLE 3 :

Soit (S_n) une suite convergente de limite S^* telle que :

$$\forall\ n \geq 0,\ S_{3n+1} < S_{3n+2} < S_{3n} < S_{3n+4} < S^*$$



Pour tout entier $n \geq 0$, nous avons :

$$\mu^+(3n) = 3$$
$$\mu^+(3n+1) = 1$$
$$\mu^+(3n+2) = 1$$

et, pour tout entier $n > 0$, nous avons :

$$\mu^-(3n) = 1$$
$$\mu^-(3n+1) = 2$$
$$\mu^-(3n+2) = 1$$

Donc la suite (S_n) appartient à **2.REG1** et à **3.PROG1**, mais n'appartient pas à **2.PROG1**.■

EXEMPLE 4 :

Posons, pour tout $n > 0$, $\theta(n) = (n-1)(n+2)/2$.
Soit (S_n) une suite convergente de limite S^* telle que, pour tout entier $n > 1$ et pour tous les entiers j et k compris strictement entre $\theta(n)$ et $\theta(n+1)$, on ait :

$$S_{\theta(n-1)} < S_j = S_k < S_{\theta(n)} < S_{\theta(n+1)} < S^*.$$

S_0 S_2 S_5 ... $S_{\theta(n-1)}$ $S_{\theta(n)}$ $S_{\theta(n+1)}$

-⊥---⊥-⊥---⊥----⊥----⊥------⊥------⊥------⊥-----⊥----⊥->

S_1 $S_3 = S_4$ $S_6 = S_7 = S_8$ $S_{\theta(n)+1} : = S_{\theta(n+1)-1}$ S^*

Pour tout entier n > 1, nous avons :

$\mu^-(\theta(n)+1) = 2$

et pour tout entier j tel que $j-1 \notin \theta(N^*)$:

$\mu^-(j) = 1$

D'autre part, pour tout entier $n \geq 1$, nous avons :

$\mu^+(\theta(n)) = \theta(n+1) - \theta(n) = n+1$

et pour tout entier $j \notin \theta(N^*)$:

$\mu^+(j) = 1$

Donc la suite (S_n) appartient à 2.REG1 mais n'appartient à aucun ensemble q.PROG1, avec $q \geq 1$, donc n'appartient pas à *.PROG1.■

En résumé nous pouvons énoncer :

Proposition 4.

‖ a) 1.PROG1 = 1.REG1 = MON
‖ b) Pour tout entier $q \geq 1$: q.PROG1 $\cap$ q.REG1 $\neq \emptyset$

Proposition 5.

‖ Pour tout entier $q \geq 2$:
‖ a) q.PROG1 $\not\subset$ q.REG1
‖ b) q.PROG1 $\not\subset$ *.REG1
‖ c) q.REG1 $\not\subset$ q.PROG1
‖ d) q.REG1 $\not\subset$ *.PROG1

Nous allons voir maintenant que, pour tout entier q, $q \geq 1$, Q est asymptotiquement décidable sur chacun des ensembles q.PROG1 et q.REG1, puis nous étendrons ces résultats aux ensembles *.PROG1 et *.REG1; mais cela ne résulte nullement - et nous en verrons plus loin un exemple - du fait que *.PROG1 et *.REG1 sont réunions de sous-ensembles sur lesquels Q est asymptotiquement décidable.

5. DECIDABILITE ASYMPTOTIQUE DE LA QUESTION Q SUR LES ENSEMBLES q.PROG1, q.REG1

Donnons d'abord une autre caractérisation de l'ensemble q.PROG1. Nous utiliserons les notations suivantes :

Pour toute suite (S_n), $U_{n,q}$ (resp. $V_{n,q}$) désigne le plus petit élément (resp. le plus grand élément) de l'ensemble contenant les q termes de la suite d'indice inférieur ou égal à n, c'est-à-dire :

$$U_{n,q} = \min (S_n, S_{n-1}, \ldots, S_{n-q+1}).$$

$$V_{n,q} = \max (S_n, S_{n-1}, \ldots, S_{n-q+1}).$$

Lemme 2.

Pour toute suite (S_n) convergente de limite S^*, nous avons les équivalences suivantes :

$(S_n) \in q.PROG1 \cap C1^- \iff \exists N, \forall n \geq N : S_n \leq V_{n+q,q} < S^*$

$(S_n) \in q.PROG1 \cap C1^+ \iff \exists N, \forall n \geq N : S^* < U_{n+q,q} \leq S_n$

PREUVE

$(S_n) \in q.PROG1 \cap C1^-$

$\iff \exists N, \forall n \geq N, \exists j$ tel que : $1 \leq j \leq q$ et $S_n \leq S_{n+j} < S^*$

$\iff \exists N, \forall n \geq N : S_n \leq \max (S_{n+1}, \ldots, S_{n+q}) = V_{n+q,q} < S^*$

La deuxième équivalence se démontre de façon analogue.■

D'autre part, nous pouvons énoncer :

Lemme 3.

Toute suite (S_n) de C^* de limite S^* vérifie la propriété

$\forall N, \exists n \geq N$ tel que : $\forall p > n, |S_p - S^*| < |S_n - S^*|$

PREUVE

Faisons une démonstration par l'absurde.

Supposons qu'il existe un entier $\alpha(0)$ tel que :

$\forall n \geq \alpha(0), \exists \delta(n) > n \ / \ |e_{\delta(n)}| \geq |e_n|$

On peut toujours supposer : $e_{\alpha(0)} \neq 0$.

De plus, en posant $\alpha(1) = \delta(\alpha(0))$, on a :

$\alpha(1) > \alpha(0)$ et

$|e_{\alpha(1)}| \geq |e_{\alpha(0)}|$

On construit alors par récurrence une suite strictement croissante d'entiers $(\alpha(k))$ telle que, pour tout $k \geq 0$:

$\alpha(k+1) = \delta(\alpha(k))$

et $|e_{\alpha(k+1)}| \geq |e_{\alpha(k)}| \geq |e_{\alpha(0)}| > 0$
ce qui contredit le fait que la suite (S_n) converge.■

Corollaire 1.

a) Toute suite (S_n) de $C1^-$ vérifie :
$\forall N, \exists n \geq N : S_n < \inf \{ S_p, p > n \}$.
b) Toute suite (S_n) de $C1^+$ vérifie :
$\forall N, \exists n \geq N : S_n > \sup \{ S_p, p > n \}$.

PREUVE
D'après le lemme 3, on a :
$\forall N, \exists n \geq N : S_n \leq \inf \{ S_p, p > n \}$
et comme la suite (S_n) converge vers S^*, on a :
$S_n \neq \inf \{ S_p, p > n \}$. ■

Le lemme 2 et le corollaire 1 nous permettent alors de construire un algorithme de décidabilité de la question Q sur q.PROG1, que l'on appellera le q.prog-algorithme.

Comme q.PROG1 est inclus dans C1 et que $C1 = C1^- \cup C1^+$, nous avons :

$$q.PROG1 = (q.PROG1 \cap C1^-) \cup (q.PROG1 \cap C1^+).$$

Pour toute suite (S_n) de q.PROG1, la suite (R_n) des réponses fournies par l'algorithme devra donc vérifier :
- $\exists N_0, \forall n \geq N_0 : R_n = -1$ si $(S_n) \in q.PROG1 \cap C1^-$
- $\exists N_0, \forall n \geq N_0 : R_n = +1$ si $(S_n) \in q.PROG1 \cap C1^+$.

description du q.prog-algorithme

```
| n <- q;
|
| répéter à l'infini
|
|    (a) tant que S(n-q) ≤ V(n,q) faire
|                     R(n) <- -1;
|                     n <- n+1
|                     fin du tant que (a);
|
|    (b) tant que S(n-q) ≥ U(n,q) faire
|                     R(n) <- +1;
|                     n <- n+1
|                     fin du tant que (b)
| fin du répéter.
```

PREUVE DU q.PROG-ALGORITHME :

Etant donnée une suite (S_n), notons :

$C_a(n)$ la proposition : $S_{n-q} \leq V_{n,q}$

$C_b(n)$ la proposition : $S_{n-q} \geq U_{n,q}$

La proposition $C_a(n)$ caractérisera les suites de q.PROG1 $\cap$ $C1^-$.

La proposition $C_b(n)$ caractérisera les suites de q.PROG1 $\cap$ $C1^+$.

Fait 1 : **pour tout entier n, on a** : $C_a(n)$ ou $C_b(n)$.

En conséquence, lors du déroulement de l'algorithme, quelque soit l'entier N, il existera un instant t tel que la valeur de n à l'instant t sera égale à N. Nous dirons "la valeur de N est atteinte".

- Si (S_n) une suite de q.PROG1 $\cap$ $C1^-$, nous allons montrer qu'à partir d'un certain rang, l'algorithme boucle dans le tant que (a).

(S_n) vérifie les deux faits suivants :

Fait 2 : $\exists$ N, $\forall$ n $\geq$ N on a $C_a(n)$.
ce fait résulte du lemme 2.

Fait 3 : $\forall$ N', $\exists$ m $\geq$ N' tel que l'on ait non $C_b(m)$.
ce fait résulte du corollaire 1.

D'après le fait 1, la valeur N sera atteinte.
Et pour tout n $\geq$ N, nous serons après toute incrémentation n $\leftarrow$ n+1, dans l'un des deux cas suivants :

1er cas : $R_{n-1} = -1$.
on était donc dans le tant que (a).
alors d'après le fait 2, pour tout n' $\geq$ n, on aura $C_a(n')$, i.e. on reste dans le tant que (a).
Par suite on aura bien : $\forall$ n' $\geq$ n, $R_{n'} = -1$.

2ème cas : $R_{n-1} = +1$.
on était donc dans le tant que (b).
alors d'après le fait 3, il existe m $\geq$ n tel que l'on ait non $C_b(m)$, i.e. on ne peut pas boucler dans le tant que (b).
On va alors tester la condition $C_a(m)$ qui est vraie car m $\geq$ N et on se ramène au 1er cas.

- Si (S_n) est une suite de q.PROG1 $\cap$ $C1^+$, on montre de la même façon que l'algorithme boucle dans le tant que (b).■

Et donc, nous avons le résultat :

Théorème 4.

Pour tout entier q non nul, la question "quel est le signe de $S_n - S^*$?" est asymptotiquement décidable sur l'ensemble q.PROG1.

Etudions maintenant les ensembles q.REG1.

En reprenant les notations précédentes, nous avons :

$U_{n-1,q} = \min (S_{n-1}, S_{n-2}, \ldots, S_{n-q})$
$V_{n-1,q} = \max (S_{n-1}, S_{n-2}, \ldots, S_{n-q})$

Lemme 4.

Pour toute suite (S_n) convergente, nous avons les deux équivalences suivantes :
$(S_n) \in$ q.REG1 $\cap$ $C1^-$ $\iff \exists N, \forall n \geq N : U_{n-1,q} \leq S_n < S^*$
$(S_n) \in$ q.REG1 $\cap$ $C1^+$ $\iff \exists N, \forall n \geq N : S^* < S_n \leq V_{n-1,q}$

PREUVE

$(S_n) \in$ q.REG1 $\cap$ $C1^-$
$\iff \exists N, \forall n \geq N, \exists j$ tel que : $1 \leq j \leq q$ et $S_{n-j} \leq S_n < S^*$
$\iff \exists N, \forall n \geq N, \min (S_{n-1}, \ldots, S_{n-q}) = U_{n-1,q} \leq S_n < S^*$.
idem pour la deuxième équivalence.■

Or, nous pouvons énoncer le lemme suivant :

Lemme 5.

Soit q un entier quelconque non nul.
a) toute suite (S_n) de $C1^-$ vérifie :
$\forall N, \exists n \geq N : V_{n-1,q} < S_n$
b) toute suite (S_n) de $C1^+$ vérifie :
$\forall N, \exists n \geq N : S_n < U_{n-1,q}$

PREUVE

Faisons par exemple la démonstration dans le cas où (S_n) est une suite de $C1^-$ de limite S^*.
Raisonnons par l'absurde. Supposons que (S_n) vérifie :
il existe $N \geq q$ tel que :
(1) $\forall n \geq N-q : S_n < S^*$
(2) $\forall n \geq N : S_n \leq V_{n-1,q}$
alors on a pour tout $n > N$:
$V_{n-1,q} = \max (S_n, V_{n-1,q})$

$= \max(S_n, S_{n-1}, \ldots, S_{n-q+1}, S_{n-q})$
$\geq \max(S_n, S_{n-1}, \ldots, S_{n-q+1}) = V_{n,q}$

Or, d'après (1) : $V_{N,q} < S^*$
donc : $\forall\, n \geq N,\ S_n \leq V_{N,q} < S^*$
ce qui contredit le fait que la suite (S_n) converge vers S^*.■

remarque :
Notons, qu'à la différence du lemme 2, ce résultat n'est pas vrai pour toute suite de C^*. Il suffit de considérer une suite (S_n) alternée telle que les deux sous-suites (S_{2p}) et (S_{2p+1}) sont monotones, l'une croissante et l'autre décroissante.
Alors, pour tout entier q non nul et tout entier n, on a :
$S_n \in [U_{n-1,q}\ ,\ V_{n-1,q}]$.

S^*

$S_{2p-2q}\ \ldots\ S_{2p-2}\quad S_{2p}\qquad S_{2p-1}\quad S_{2p-3}\ \ldots\ S_{2p-2q-1}$

Les lemmes 4 et 5 nous permettent alors de construire un algorithme de décidabilité de la question Q sur q.REG1, que l'on appellera le q.reg-algorithme. A chaque étape n, $n \geq q$, il utilise, lui aussi, q+1 termes consécutifs de la suite, à savoir $S_n, S_{n-1}, \ldots, S_{n-q}$.

Comme une suite de **q.REG1** appartient soit à $C1^-$, soit à $C1^+$, la suite (R_n) des réponses fournies par l'algorithme devra vérifier :
- $\exists\, N_0,\ \forall\, n \geq N_0 : R_n = -1$ si $(S_n) \in$ **q.REG1** $\cap\, C1^-$
- $\exists\, N_0,\ \forall\, n \geq N_0 : R_n = +1$ si $(S_n) \in$ **q.REG1** $\cap\, C1^+$.

DESCRIPTION DU q.REG-ALGORITHME

| n <- q;
|
| répéter à l'infini
|
| (a) tant que $S_n \geq U_{n-1,q}$ faire
| R_n <- -1;
| n <- n+1
| **fin du tant que** (a);
|
| (b) tant que $S_n \leq V_{n-1,q}$ faire
| R_n <- +1;
| n <- n+1
| **fin du tant que** (b)
| fin du répéter.

PREUVE DU q.REG-ALGORITHME :

Etant donnée une suite (S_n), notons :

$C_a(n)$ la proposition : $S_n \geq U_{n-1,q}$

$C_b(n)$ la proposition : $S_n \leq V_{n-1,q}$

La proposition $C_a(n)$ caractérisera les suites de q.REG1 $\cap$ $C1^-$.

La proposition $C_b(n)$ caractérisera les suites de q.REG1 $\cap$ $C1^+$.

Fait 1 : pour tout entier n, on a : $C_a(n)$ ou $C_b(n)$.

En conséquence, lors du déroulement de l'algorithme, quelque soit l'entier N, il existera un instant t tel que la valeur de n à l'instant t sera égale à N. Nous dirons "la valeur de N est atteinte".

- Si (S_n) une suite de q.REG1 $\cap$ $C1^-$, nous allons montrer qu'à partir d'un certain rang, l'algorithme boucle dans le tant que (a).

(S_n) vérifie les deux faits suivants :

Fait 2 : $\exists$ N, $\forall$ n $\geq$ N on a $C_a(n)$.
ce fait résulte du lemme 4.

Fait 3 : $\forall$ N', $\exists$ m $\geq$ N' tel que l'on ait non $C_b(m)$.
ce fait résulte du lemme 5.

D'après le fait 1, la valeur N sera atteinte.
Et pour tout n $\geq$ N, nous serons après toute incrémentation n $\leftarrow$ n+1, dans l'un des deux cas suivants :

1er cas : $R_{n-1} = -1$.
on était donc dans le tant que (a).
alors d'après le fait 2, pour tout n' $\geq$ n, on aura $C_a(n')$, i.e. on reste dans le tant que (a).
Par suite on aura bien : $\forall$ n' $\geq$ n, $R_{n'} = -1$.

2ème cas : $R_{n-1} = +1$.
on était donc dans le tant que (b).
alors d'après le fait 3, il existe m $\geq$ n tel que l'on ait non $C_b(m)$, i.e. on ne peut pas boucler dans le tant que (b).
On va alors tester la condition $C_a(m)$ qui est vraie car m $\geq$ N et on se ramène au 1er cas.

- Si (S_n) est une suite de q.REG1 $\cap$ $C1^+$, on montre de la même façon que l'algorithme boucle dans le tant que (b).■

Et donc, nous avons le résultat :

Théorème 5.

Pour tout entier q non nul, la question "quel est le signe de S_n-S^* ?" est asymptotiquement décidable sur l'ensemble q.REG1.

6. DECIDABILITE ASYMPTOTIQUE DE LA QUESTION Q SUR LES ENSEMBLES *.PROG1 ET *.REG1

En utilisant les mêmes idées, nous allons montrer que, de manière plus générale, la question Q "quel est le signe de S_n-S^* ?" est asymptotiquement décidable sur *.PROG1.

Lemme 6.

a) Soit (S_n) une suite de r.PROG1 $\cap$ $C1^-$ et $\tilde{N} \geq r$ tels que : $\forall$ n $\geq \tilde{N}$, $S_{n-r} \leq V_{n,r} < S^*$.
Alors, pour tout entier q $\geq$ r et tout entier n $\geq \tilde{N}$+q-r, on a : $S_{n-q} \leq V_{n,q} < S^*$.
b) Soit (S_n) une suite de r.PROG1 $\cap$ $C1^+$ et $\tilde{N} \geq r$ tels que $\forall$ n $\geq \tilde{N}$, $S^* < U_{n,r} \leq S_{n-r}$.
Alors, pour tout entier q $\geq$ r et tout entier n $\geq \tilde{N}$+q-r, on a : $S^* < U_{n,q} \leq S_{n-q}$.

PREUVE

Faisons par exemple la démonstration de (a).
Raisonnons par récurrence sur l'entier q.
La propriété est vraie pour q = r.
Supposons que :
$\forall n \geq \tilde{N}+q-r,\ S_{n-q} \leq V_{n,q}$
Montrons que :
$\forall n \geq \tilde{N}+(q+1)-r,\ S_{n-(q+1)} \leq V_{n,q+1}$
Soit n $\geq \tilde{N}$+q+1-r, alors on a :
$n-1 \geq \tilde{N}+q-r$
et d'après l'hypothèse de récurrence
$S_{n-1-q} \leq V_{n-1,q}$
Or : $V_{n-1,q} = \max (S_{n-1}, \ldots, S_{n-q})$
$\leq \max (S_n, S_{n-1}, \ldots, S_{n-q})$
d'où : $V_{n-1,q} \leq V_{n,q+1}$
et on a donc bien :
$S_{n-(q+1)} \leq V_{n,q+1}$.■

Décrivons maintenant le prog-algorithme : algorithme de décidabilité de la question Q "quel est le signe de $S_n - S^*$?" sur l'ensemble *.PROG1.

description du prog-algorithme

```
| q <- 0; n <- 0; nq <- 0;
|
| répéter à l'infini
|    q <- q+1; n <- n+2; nq <- n  {nq servira uniquement à la
|                                    preuve de l'algorithme}
|    (a) tant que S_{n-q} ≤ V_{n,q} faire
|                     R_n <- -1;
|                     n <- n+1
|                     fin du tant que (a);
|
|    (b) tant que S_{n-q} ≥ U_{n,q} faire
|                     R_n <- +1;
|                     n <- n+1
|                     fin du tant que (b)
| fin du répéter.
```

PREUVE DU PROG-ALGORITHME :

Etant donnée une suite (S_n), notons :

A(q,n) la proposition : $S_{n-q} \le V_{n,q}$

B(q,n) la proposition : $S_{n-q} \ge U_{n,q}$

La proposition A(q,n) caractérisera les suites de q.PROG1 $\cap$ $C1^-$.

La proposition B(q,n) caractérisera les suites de q.PROG1 $\cap$ $C1^+$.

Fait 1 : tout au long de l'exécution de l'algorithme, on a $n \ge nq \ge 2q$

- Soit (S_n) une suite de *.PROG1 $\cap$ $C1^-$.

Soit r le plus petit entier tel que $(S_n) \in$ r.PROG1.

(S_n) vérifie les trois faits suivants :

Fait 2 : $\exists\ \tilde{N} \ge r,\ \forall\ q \ge r,\ \forall\ n \ge \tilde{N}+q-r$, on a : A(q,n).
ce fait résulte du lemme 2 et du lemme 6.

Fait 3 : $\forall\ q \ge 1,\ \forall\ n,\ \exists\ n' \ge n$ tel que l'on ait non B(q,n').
ce fait découle du corollaire 1 du lemme 3.

On ne pourra donc pas boucler dans le tant que (b).

Fait 4 : $\forall$ q < r, $\forall$ n, $\exists$ n' $\geq$ n tel que l'on ait non A(q,n').
cela résulte du choix de r tel que $(S_n) \notin$ q.PROG1 pour q < r.

Tant que l'on n'a pas atteint la valeur r, on ne peut pas boucler dans le tant que (a).

d'après les faits 3 et 4, q atteindra la valeur r.

1er cas : $\exists$ q, avec r $\leq$ q < $\tilde{N}$, tel que :
$\forall$ n $\geq$ nq, A(q,n).
alors on boucle dans le tant que (a) et on a bien :
$\forall$ n $\geq$ nq, $R_n = -1$.

2ème cas : $\forall$ q, avec r $\leq$ q < $\tilde{N}$, il existe n $\geq$ nq tel que l'on ait non A(q,n).
Alors, d'après le fait 3, q atteindra la valeur $\tilde{N}$.
et d'après le fait 1, pour tout n $\geq$ nq, on aura :
n $\geq$ 2q $\geq$ $\tilde{N}$+q $\geq$ $\tilde{N}$+q-r.
Donc, d'après le fait 2, pour tout n $\geq$ nq, on a : A(q,n').
On boucle donc dans le tant que (a).
Et on a bien : $\forall$ n $\geq$ nq, $R_n = -1$.

- La démonstration est analogue pour une une suite de *.PROG1 $\cap$ $C1^+$.■

Et donc, nous énonçons :

Théorème 6.

La question "quel est le signe de S_n-S* ?" est asymptotiquement décidable sur l'ensemble *.PROG1.

Etudions maintenant la décidabilité de la question Q sur l'ensemble *.REG1.

Lemme 7.

a) Soit (S_n) une suite de r.REG1 $\cap$ $C1^-$ et $\tilde{N} \geq r$ tels que:
$\forall$ n $\geq \tilde{N}$, $U_{n-1,r} \leq S_n < S^*$.
Alors, pour tout entier q $\geq$ r et tout entier n $\geq$ $\tilde{N}$+q-r,
on a : $U_{n-1,q} \leq S_n < S^*$.
b) Soit (S_n) une suite de r.REG1 $\cap$ $C1^+$ et $\tilde{N} \geq r$ tels que:
$\forall$ n $\geq \tilde{N}$, $S^* < S_n \leq V_{n-1,r}$.
Alors, pour tout entier q $\geq$ r et tout entier n $\geq$ $\tilde{N}$+q-r,
on a : $S^* < S_n \leq V_{n-1,q}$.

PREUVE

Faisons par exemple la démonstration de (a)
Raisonnons par récurrence sur l'entier q.
La propriété est vraie pour q = r.
Supposons que :

$\forall\ n \geq \tilde{N}+q-r,\ U_{n-1,q} \leq S_n < S^*$

Montrons que :

$\forall\ n \geq \tilde{N}+(q+1)-r,\ U_{n-1,q+1} \leq S_n < S^*$

Soit $n \geq \tilde{N}+q+1-r$, alors on a :

$$\begin{aligned} U_{n-1,q+1} &= \min\ (S_{n-1}, S_{n-2}, \ldots, S_{n-q}, S_{n-(q+1)}) \\ &\leq \min\ (S_{n-1}, S_{n-2}, \ldots, S_{n-q}) \end{aligned}$$

d'où : $U_{n-1,q+1} \leq U_{n-1,q}$

Et comme $n-1 \geq \tilde{N}+q-r$, on a bien :

$U_{n-1,q+1} \leq S_{n-1} < S^*$.■

L'algorithme de décidabilité de la question Q sur l'ensemble *.REG1 est appelé le reg-algorithme. Nous le décrivons :

description du reg-algorithme

```
| q <- 0; n <- 0; nq <- 0;
|
| répéter à l'infini
|    q <- q+1; n <- n+2; nq <- n;  {nq servira uniquement à la
|                                    preuve de l'algorithme}
|    (a) tant que Sn ≥ Un-1,q faire
|                    Rn <- -1;
|                    n <- n+1
|                    fin du tant que (a);
|
|    (b) tant que Sn ≤ Vn-1,q faire
|                    Rn <- +1;
|                    n <- n+1
|                    fin du tant que (b)
| fin du répéter.
```

PREUVE DU REG-ALGORITHME :

Etant donnée une suite (S_n), notons :

$A(q,n)$ la proposition : $S_n \geq U_{n-1,q}$

$B(q,n)$ la proposition : $S_n \leq V_{n-1,q}$

La proposition $A(q,n)$ caractérisera les suites de q.REG1 $\cap$ $C1^-$.

La proposition $B(q,n)$ caractérisera les suites de q.REG1 $\cap$ $C1^+$.

Fait 1 : tout au long de l'exécution de l'algorithme, on a

$n \geq nq \geq 2q$

- Soit (S_n) une suite de *.REG1 $\cap$ $C1^-$.
Soit r le plus petit entier tel que $(S_n) \in$ r.REG1.

(S_n) vérifie les trois faits suivants :

Fait 2 : $\exists\ \tilde{N} \geq r$, $\forall\ q \geq r$, $\forall\ n \geq \tilde{N}+q-r$, on a : $A(q,n)$.
ce fait résulte du lemme 4 et du lemme 7.

Fait 3 : $\forall\ q \geq 1$, $\forall\ n$, $\exists\ n' \geq n$ tel que l'on ait non $B(q,n')$.
ce fait découle du lemme 5.

On ne pourra donc pas boucler dans le tant que (b).

Fait 4 : $\forall\ q < r$, $\forall\ n$, $\exists\ n' \geq n$ tel que l'on ait non $C_a(n')$.
cela résulte du choix de r tel que $(S_n) \notin$ q.REG1 pour $q < r$.

Tant que l'on n'a pas atteint la valeur r, on ne peut pas boucler dans le tant que (a).

d'après les faits 3 et 4, q atteindra la valeur r.

1er cas : $\exists\ q$, avec $r \leq q < \tilde{N}$, tel que :
$\forall\ n \geq nq$, $A(q,n)$.
alors on boucle dans le tant que (a) et on a bien :
$\forall\ n \geq nq$, $R_n = -1$.

2ème cas : $\forall\ q$, avec $r \leq q < \tilde{N}$, il existe $n \geq nq$ tel que l'on ait non $A(q,n)$.
Alors, d'après le fait 3, q atteindra la valeur $\tilde{N}$.
et d'après le fait 1, pour tout $n \geq nq$, on aura :
$n \geq 2q \geq \tilde{N}+q \geq \tilde{N}+q-r$.
Donc, d'après le fait 2, pour tout $n \geq nq$, on a : $A(q,n')$.
On boucle donc dans le tant que (a).
Et on a bien : $\forall\ n \geq nq$, $R_n = -1$.

- La démonstration est analogue pour une une suite de *.REG1 $\cap$ $C1^+$.■

D'où :

Théorème 7.

La question "quel est le signe de S_n-S^* ?" est asymptotiquement décidable sur l'ensemble *.REG1.

La question Q est donc asymptotiquement décidable sur tous les sous-ensembles stricts de C1 envisagés, la raison intuitive pouvant être que, comme C1 est partitionné en $C1^+$ et $C1^-$, la suite des réponses à la question Q est asymptotiquement constante, égale à +1 (pour toute suite de $C1^+$) ou égale à -1 (pour toute suite de $C1^-$).

Nous avons proposé deux généralisations possibles de la monotonie : les suites progressives et les suites régressives. Dans C1, elles donnent les mêmes résultats de décidabilité de la question Q. Nous verrons qu'il n'en est pas de même dans C^*.

D. ETUDE DE SOUS-ENSEMBLES DE C2

Rappelons qu'une suite (S_n) de C2 est une suite de C^*, de limite S^* telle que :

$$\forall N, \exists n > N : (S_n - S^*)(S_{n-1} - S^*) < 0$$

Nous avons prouvé dans la partie B que la question Q "quel est le signe de $S_n - S^*$?" est asymptotiquement indécidable sur C2. Existe-t-il alors des sous-ensembles de C2 sur lesquels la question Q soit asymptotiquement décidable ?

1. DECIDABILITE ASYMPTOTIQUE DE LA QUESTION Q SUR L'ENSEMBLE ALT

ALT désigne le sous-ensemble de C2 contenant les suites (S_n), de limite S^*, asymptotiquement alternées, c'est-à-dire telles que, pour tout entier n suffisamment grand, les erreurs $e_n = S_n - S^*$ et $e_{n-1} = S_{n-1} - S^*$ sont de signe opposé.

$$(S_n) \in \textbf{ALT} \iff \exists N, \forall n > N : e_n \, e_{n-1} < 0$$

Autrement dit, pour tout n suffisamment grand, la limite S^* appartient au segment d'extrémités S_n et S_{n-1}. Ce qui peut encore s'écrire, en utilisant la notation :

$$>S_n, S_{n-1}< \; = \begin{cases}]S_n, S_{n-1}[& \text{si} \quad S_n < S_{n-1} \\ ]S_{n-1}, S_n[& \text{si} \quad S_{n-1} < S_n \end{cases}$$

$$(S_n) \in \textbf{ALT} \iff \exists N, \forall n > N : S^* \in \; >S_n, S_{n-1}<$$

Donc, pour toute suite (S_n) de **ALT** et pour tout entier n assez grand, S_{n-1} et S^* se trouvent d'un même côté par rapport à S_n. Cela nous permet de proposer un algorithme de décidabilité de la question Q "quel est le signe de $S_n - S^*$?" sur l'ensemble **ALT**, qui utilise, à chaque étape n, $n \geq 1$, deux termes consécutifs de la suite : S_n et S_{n-1}.

description de l'algorithme

```
| n <- 1;
|
| Répéter à l'infini
|
|    Si  S_{n-1} < S_n    alors  R_n <- +1;
|                         sinon  R_n <- -1
|    n <- n+1
|
| fin du répéter.
```

Nous énonçons :

Proposition 6.

La question "quel est le signe de S_n-S^* ?" est asymptotiquement décidable sur l'ensemble **ALT**.

Un sous-ensemble particulier de **ALT** est le sous-ensemble, noté OSC, des suites oscillantes pour lesquelles les deux sous-suites (S_{2p}) et (S_{2p+1}) sont asymptotiquement monotones, l'une croissante et l'autre décroissante.

$$(S_n) \in \text{OSC} \Leftrightarrow (S_n) \in \textbf{ALT} \text{ et } \exists N, \forall n > N : |e_n| \leq |e_{n-2}|.$$

Nous allons maintenant construire de nouveaux sous-ensembles de C2, pour lesquels nous éudierons par la suite le problème de la décidabilité asymptotique de la question "quel est le signe de S_n-S^* ?".

2. DEFINITIONS DES SUITES q-ALTERNEES

Afin de généraliser la notion de suites alternées, nous allons définir deux indices d'alernance, $\alpha^+(n)$ et $\alpha^-(n)$ et des ensembles associés à ces indices.

DEFINITION 8 :

Soit (S_n) une suite de **C***, de limite S^* et $e_n = S_n - S^*$.
On appelle **indices d'alternance** les entiers $\alpha^+(n)$ et $\alpha^-(n)$ définis par :
$\alpha^+(n) = \min (\{j > 0 \ / \ e_n \, e_{n+j} < 0\} \cup \{0\})$,
$\alpha^-(n) = \min (\{j > 0 \ / \ e_n \, e_{n-j} < 0\} \cup \{0\})$.

Pour toute suite de C2, $\alpha^-(n)$ est non nul à partir d'un certain rang tandis que $\alpha^+(n)$ est non nul pour tout entier n.

L'ensemble **ALT** apparaît donc comme l'ensemble des suites de C2 pour lesquelles les indices d'alternance sont, à partir d'un certain rang, tous les deux constants et égaux à 1.

$(S_n) \in$ **ALT** $\iff \exists N, \forall n \geq N : \alpha^+(n) = 1$
$\iff \exists N, \forall n \geq N : \alpha^-(n) = 1$

Nous pouvons généraliser cette définition et construire à priori deux types de sous-ensembles de C2 : les ensembles $q.$**ALT**$^+$ et $q.$**ALT**$^-$, où q est un entier non nul. $q.$**ALT**$^+$ est l'ensemble des suites pour lesquelles, à partir d'un certain rang, $\alpha^+(n)$ est compris entre 1 et q ; et $q.$**ALT**$^-$ est l'ensemble des suites pour lesquelles, à partir d'un certain rang $\alpha^-(n)$ est compris entre 1 et q.

DEFINITION 9 :

Soit q un entier non nul et (S_n) une suite convergente de limite S^*.
$(S_n) \in q.$**ALT**$^+ \iff \exists N, \forall n \geq N : 1 \leq \alpha^+(n) \leq q$
$(S_n) \in q.$**ALT**$^- \iff \exists N, \forall n \geq N : 1 \leq \alpha^-(n) \leq q$

De façon plus détaillée, une suite (S_n) de $q.$**ALT**$^+$ est caractérisée par :

$$\exists N, \forall n \geq N, \exists j \text{ tel que : } 1 \leq j \leq q \quad \text{et} \quad e_n\, e_{n+j} < 0$$

Et une suite (S_n) de $q.$**ALT**$^-$ est caractérisée par :

$$\exists N, \forall n \geq N, \exists j \text{ tel que : } 1 \leq j \leq q \quad \text{et} \quad e_n\, e_{n-j} < 0$$

Or nous avons :

Proposition 7.

Pour tout entier $q \in$ **N***, $q.$**ALT**$^+ = q.$**ALT**$^-$.

PREUVE

Montrons par exemple l'inclusion $q.$**ALT**$^+ \subset q.$**ALT**$^-$.
Raisonnons par l'absurde.
Soit $(S_n) \in q.$**ALT**$^-$. Suposons que $(S_n) \notin q.$**ALT**$^+$.
Alors (S_n) vérifie :
(1) $\exists N, \forall n \geq N, \exists j$, tel que : $1 \leq j \leq q$ et $e_n\, e_{n-j} < 0$
(2) $\forall \tilde{N}, \exists n \geq \tilde{N}$ tel que : pour tout j avec $1 \leq j \leq q$ on a $e_n\, e_{n+j} > 0$

en particulier il existe $n \geq N$ tel que, pour tout j avec $1 \leq j \leq q$, on a : $e_n\, e_{n+j} > 0$.
cela entraîne que, pour tout j avec $1 \leq j \leq q$, on a :

$e_{n+q}\ e_{n+q-j} > 0$

ce qui contredit (1).■

Nous noterons donc plus simplement par q.ALT cet ensemble, et une suite de q.ALT sera appelée une suite q-alternée. Une suite (S_n) q-alternée est donc telle que, pour tout entier n suffisamment grand, la limite S^* appartient au plus petit segment contenant S_n ainsi que les q termes suivants, et aussi au plus petit segment contenant S_n ainsi que les q termes précédents.

Il résulte de la définition de q.ALT :

Proposition 8.

‖ 1.ALT = ALT
‖ $\forall$ q ≥ 1, q.ALT ⊂ (q+1).ALT

*.ALT désignera l'union de tous les ensembles q.ALT, pour q ≥ 1.

3. SUITES PROGRESSIVES ET REGRESSIVES DE C2

Nous reprenons ici les notions d'indice de progression et de régression introduites dans le paragraphe B.3 qui nous ont permis de définir, pour tout entier q, q≥1, les ensembles q.PROG (resp. *.PROG) des suites q-progressives (resp. progressives) et les ensembles q.REG (resp. *.REG) des suites q-régressives (resp. régressives). Rappelons les définitions :

$$\mu^+(n) = \min \{j,\ j>0 \ \text{ et } \ S_{n+j} \in \langle S_n, S^* \langle \ \}$$

$$(S_n) \in q.PROG \iff \exists N,\ \forall n \geq N : \mu^+(n) \leq q$$

$$\mu^-(n) = \min \{j,\ j>0 \ \text{ et } \ S_n \in \langle S_{n-j}, S^* \langle \ \}$$

$$(S_n) \in q.REG \iff \exists N,\ \forall n \geq N : \mu^-(n) \leq q$$

$$*.PROG = \bigcup_{q=1}^{+\infty} q.PROG$$

$$*.REG = \bigcup_{q=1}^{+\infty} q.REG$$

Nous avions aussi défini les intersections de ces ensembles avec C2. Les notations sont les suivantes :

DEFINITION 10 :

$q.PROG2 = q.PROG \cap C2$

$q.REG2 = q.REG \cap C2$

$$*.PROG2 = *.PROG \cap C2 = \bigcup_{q=1}^{+\infty} q.PROG2$$

$$*.REG2 = *.REG \cap C2 = \bigcup_{q=1}^{+\infty} q.REG2$$

Il résulte des définitions que :

<u>Proposition 9.</u>

$\forall\, q \geq 1,\ q.PROG2 \subset (q+1).PROG2$

$\forall\, q \geq 1,\ q.REG2 \subset (q+1).REG2$

D'autre part, nous avons :

<u>Proposition 10.</u>

a) $1.PROG2 = 1.REG2 = \emptyset$

b) $2.PROG2 = 2.REG2 = OSC$

c) pour tout $q \geq 2$, $q.PROG2 \cap q.REG2 \neq \emptyset$

PREUVE

a) $1.PROG2 = 1.PROG \cap C2 = MON \cap C2 = \emptyset$

b) Il suffit de montrer que : $2.PROG2 \subset OSC$
$2.REG2 \subset OSC$

- Soit $(S_n) \in 2.PROG2$.

(S_n) vérifie :

(1) $\exists\, N,\ \forall\, n \geq N,\ \exists\, j$ tel que : $1 \leq j \leq 2$ et $S_{n+j} \in\ <S_n, S^*<$

Comme $(S_n) \in C2$, il existe $\tilde{N} \geq N$ tel que :

$S_{\tilde{N}} < S^* < S_{\tilde{N}+1}$

Alors, d'après (1), on a nécessairement :

$S_{\tilde{N}} \leq S_{\tilde{N}+2} < S^* < S_{\tilde{N}+3} \leq S_{\tilde{N}+1}$

Et par récurrence on obtient pour tout $p \geq 0$

$S_{\tilde{N}+2p} \leq S_{\tilde{N}+2p+2} < S^* < S_{\tilde{N}+2p+3} \leq S_{\tilde{N}+2p+1}$

ce qui prouve que $(S_n) \in OSC$.

- Soit maintenant $(S_n) \in 2.REG2$.

(S_n) vérifie :

(2) $\exists\, N,\ \forall\, n \geq N,\ \exists\, j$ tel que : $1 \leq j \leq 2$ et $S_n \in\ <S_{n-j}, S^*<$

Montrons par l'absurde que $(S_n) \in ALT$.

Supposons que : $\forall\, \tilde{N},\ \exists\, n \geq \tilde{N}$ tel que $e_n\, e_{n+1} > 0$

en particulier il existe $n \geq N$ tel que $e_n\, e_{n+1} > 0$,

c'est-à-dire S_n et S_{n+1} sont situés d'un même côté par rapport à la limite.

Désignons par n' le plus petit entier tel que :

$n' \geq n+2$

et $e_{n+1}\, e_{n'} < 0$, i.e. S_{n+1} et $S_{n'}$ sont situés de part et d'autre de la limite.

n' existe car $(S_n) \in$ **C2**.

On a alors : $e_{n'}\, e_{n'-1} < 0$ et $e_{n'}\, e_{n'-2} < 0$

c'est-à-dire $S_{n'-1}$ et $S_{n'}$ d'une part, $S_{n'-2}$ et $S_{n'}$ d'autre part, sont situés de part et d'autre de la limite.

ce qui contredit la propriété (2).

Donc $(S_n) \in$ q.**ALT**

Il résulte alors de la propriété (2) que $(S_n) \in$ OSC.

c) Pour tout entier $q \geq 2$, on a, d'après la proposition 9 :

2.**PROG2** $\subset$ q.**PROG2**

2.**REG2** $\subset$ q.**REG2**

et comme : 2.**PROG2** = 2.**REG2** = OSC,

on a l'inclusion :

OSC $\subset$ q.**PROG2** $\cap$ q.**REG2**.■

Proposition 11.

Pour tout entier $q \geq 1$, on a les inclusions :

a) q.**PROG2** $\subset$ q.**ALT**

b) q.**REG2** $\subset$ q.**ALT**

PREUVE

a) Soit $(S_n) \in$ q.**PROG2**.

(S_n) vérifie :

(1) $\exists$ N, $\forall$ $n \geq N$, $\exists$ j tel que : $1 \leq j \leq q$ et $S_{n+j} \in \,]S_n, S^*[$

Nous allons construire par récurrence une suite strictement croissante d'entiers $(\beta(i))_{i\geq 0}$ telle que :

$e_{\beta(i)}\, e_{\beta(i)+1} < 0$

$\beta(i+1) - \beta(i) \leq q$

Puis nous montrerons que, pour tout $i \geq 0$ et pour tout n avec $\beta(i) \leq n < \beta(i+1)$, il existe h tel que :

$1 \leq h \leq q$ et $e_n\, e_{n+h} < 0$

ce qui prouvera que $(S_n) \in$ q.**ALT**.

Par définition de **C2**, il existe un entier $\beta(0) \geq N$ tel que

$e_{\beta(0)}\, e_{\beta(0)+1} < 0$

Supposons que l'on ait construit $\beta(i)$ tel que :

$e_{\beta(i)}\, e_{\beta(i)+1} < 0$

Alors, d'après (1), il existe j avec $2 \leq j \leq q$ tel que :

$e_{\beta(i)}\, e_{\beta(i)+j} > 0$

Et il existe k avec $2 \leq k \leq q+1$ tel que :

$e_{\beta(i)+1}\, e_{\beta(i)+k} > 0$

Donc il existe un entier $\beta(i+1)$ tel que :

$\beta(i)+2 \le \beta(i+1) \le \beta(i)+q$

et $e_{\beta(i+1)}\ e_{\beta(i+1)+1} < 0$

Et on a bien : $\beta(i+1) - \beta(i) \le q$

Par construction, on a : $e_{\beta(i)}\ e_{\beta(i)+1} < 0$

D'autre part, pour tout n tel que : $\beta(i) < n < \beta(i+1)$

- ou bien : $e_n\ e_{\beta(i+1)} < 0$
  alors, en posant $h = \beta(i+1)-n$
  on a bien : $e_n\ e_{n+h} < 0$ avec $1 \le h < q$
- ou bien : $e_n\ e_{\beta(i)+1} > 0$
  on a alors : $e_n\ e_{\beta(i+1)+1} < 0$
  et, en posant : $h = \beta(i+1)+1-n$
  on obtient : $e_n\ e_{n+h} < 0$ avec $1 < h \le q$.

b) Soit $(S_n) \in$ q.**REG2**.

(S_n) vérifie :

(2) $\exists$ N, $\forall$ n $\ge$ N, $\exists$ j tel que : $1 \le j \le q$ et $S_n \in \left<S_{n-j}, S^*\right<$

Nous allons construire par récurrence une suite strictement croissante d'entiers $(\beta(i))_{i \ge 0}$ telle que :

$e_{\beta(i)}\ e_{\beta(i)+1} < 0$

$\beta(i+1)-\beta(i) \le q$

Puis nous montrerons que, pour tout $i \ge 0$ et pour tout n avec $\beta(i) < n \le \beta(i+1)$, il existe h tel que :

$1 \le h \le q$ et $e_n\ e_{n-h} < 0$

ce qui prouvera que $(S_n) \in$ q.**ALT**.

Comme $(S_n) \in$ C2, il existe un entier $\beta(0) \ge N$ tel que :

$e_{\beta(0)}\ e_{\beta(0)+1} < 0$

Supposons que l'on ait construit $\beta(i)$ tel que :

$e_{\beta(i)}\ e_{\beta(i)+1} < 0$

Et considérons le terme $S_{\beta(i)+q+1}$

- ou bien : $e_{\beta(i)+1}\ e_{\beta(i)+q+1} < 0$

alors il existe un entier $\beta(i+1)$ tel que :

$\beta(i)+1 \le \beta(i+1) \le \beta(i)+q$

$e_{\beta(i+1)}\ e_{\beta(i+1)+1} < 0$

et on a bien : $\beta(i+1)-\beta(i) \le q$

- ou bien : $e_{\beta(i)+1}\ e_{\beta(i)+q+1} > 0$

alors montrons par l'absurde qu'il existe k avec $2 \le k \le q$ tel que :

(3) $e_{\beta(i)+k}\ e_{\beta(i)+q+1} < 0$

Supposons que, pour tout k avec $2 \le k \le q$, on ait :

$e_{\beta(i)+k}\ e_{\beta(i)+q+1} > 0$

tous les termes $S_{\beta(i)+1}, S_{\beta(i)+2}, \ldots, S_{\beta(i)+q}, S_{\beta(i)+q+1}$ seraient donc d'un même côté par rapport à la limite.

alors d'après (2) on aurait :

$\forall$ n > $\beta(i)+1$, $e_{\beta(i)+1}\ e_n > 0$

ce qui contredit le fait que $(S_n) \in$ C2.

Il résulte de (3) qu'il existe une entier $\beta(i+1)$ tel que :

$\beta(i)+k \leq \beta(i+1) \leq \beta(i)+q$
$e_{\beta(i+1)}\ e_{\beta(i+1)+1} < 0$

et on a bien : $\beta(i+1)-\beta(i) \leq q$.

Par construction, on a : $e_{\beta(i)}\ e_{\beta(i)+1} < 0$
D'autre part, pour tout n tel que : $\beta(i)+1 < n \leq \beta(i+1)$
- ou bien : $e_n\ e_{\beta(i)} < 0$
  alors, en posant $h = n-\beta(i)$
  on a bien : $e_n\ e_{n-h} < 0$ avec $1<h\leq q$
- ou bien : $e_n\ e_{\beta(i)} > 0$
  on a alors : $e_n\ e_{\beta(i)+1} < 0$
  et, en posant : $h = n-(\beta(i)+1)$
  on obtient : $e_n\ e_{n-h} < 0$ avec $1\leq h<q$.■

Enfin les ensembles q.PROG2 et q.REG2 sont incomparables pour la relation d'inclusion, comme le montrent les exemples suivants.

EXEMPLE 1 :

Soit (S_n) une suite convergente de limite S^* telle que :

$\forall\ n \geq 1,\ S_{5n-4} < S_{5n} < S_{5n-2} < S_{5n+1} < S^*$
et $\qquad S^* < S_{5n+2} < S_{5n-1} < S_{5n-3}$

$S_{5n} \qquad S_{5n+5} \qquad S^* \qquad S_{5n+4} \qquad S_{5n-1}$

$S_{5n-4} \qquad S_{5n-2} \qquad S_{5n+1} \qquad S_{5n+3} \qquad S_{5n+2} \qquad S_{5n-3}$

Pour tout entier $n > 0$, nous avons :
- d'une part :
  $\mu^+(5n) = 1$
  $\mu^+(5n+1) = 2$
  $\mu^+(5n+2) = 2$
  $\mu^+(5n+3) = 3$
  $\mu^+(5n+4) = 3$

ce qui prouve que la suite (S_n) appartient à 3.**PROG2**
- d'autre part :
  $\mu^-(5n) = 4$
  $\mu^-(5n+1) = 1$
  $\mu^-(5n+2) = 3$
  $\mu^-(5n+3) = 2$
  $\mu^-(5n+4) = 2$

ce qui prouve que la suite (S_n) appartient à 4.**REG2**, mais n'appartient pas à 3.**REG2**.■

EXEMPLE 2 :

Posons, pour tout entier n, $\theta(n) = n^2$.
Notons que :
$\theta(n)+1 = \theta(n+1)-2n$

et $\theta(n)+2 = \theta(n+1)-(2(n-1)+1)$.
Soit (S_n) une suite convergente de limite nulle vérifiant les inégalités :

$$\forall\ i,\ 1 \le i \le n-1,\quad S_{\theta(n)} < S_{\theta(n)-2i} < S_{\theta(n+1)} < 0,$$
$$\forall\ i,\ 0 \le i \le n-1,\quad 0 < S_{\theta(n+1)-1} < S_{\theta(n)-(2i+1)}.$$

$S_{\theta(n)}$ $S_{\theta(n+1)}$ 0

$S_{\theta(n)-2}$: $= S_{\theta(n)-2i}$: $= S_{\theta(n)-2(n-1)}$ $S_{\theta(n+1)-1}$ $S_{\theta(n)-1}$: $= S_{\theta(n)-(2i+1)}$: $= S_{\theta(n)-(2(n-1)+1)}$

Pour tout entier $n \ge 2$, on a :
$\mu^+(\theta(n)) = 1$, car $S_{\theta(n)} < S_{\theta(n+1)} < S_{\theta(n)+1} < 0$
$\mu^+(\theta(n)-2) = 3$, car $S_{\theta(n)} < S_{\theta(n)-2} < S_{\theta(n)+1} < 0 < S_{\theta(n)-1}$
$\mu^+(\theta(n)-1) = 3$, car $S_{\theta(n)} < S_{\theta(n)+1} < 0 < S_{\theta(n)+2} < S_{\theta(n)-1}$
$\mu^+(\theta(n)-2i) = 2$, pour tout $i\ /\ 1 < i \le n-1$
$\mu^+(\theta(n)-(2i+1)) = 2$, pour tout $i\ /\ 0 < i \le n-1$.
Ce qui prouve que la suite appartient à 3.PROG2.
Par contre, elle n'appartient à aucun ensemble q.REG2, avec $q \ge 1$ et donc n'appartient pas à *.REG2. En effet il suffit de remarquer que pour tout $n > 0$, on a :

$$\mu^-(\theta(n)) = \theta(n)-\theta(n-1) = 2n-1.\blacksquare$$

EXEMPLE 3 :
Soit (S_n) une suite convergente de limite S^* telle que :

$$\forall\ n \ge 1,\ S_{5n+2} < S_{5n} < S_{5n+4} < S_{5n+7} < S^*$$
$$\text{et}\qquad S^* < S_{5n+6} < S_{5n+3} < S_{5n+1}$$

S_{5n-5} S_{5n} S^* S_{5n+8} S_{5n+3}

S_{5n-3} S_{5n-1} S_{5n+2} S_{5n+4} S_{5n+6} S_{5n+1}

Pour tout entier $n > 0$, nous avons :
- d'une part :
  - $\mu^-(5n) = 1$
  - $\mu^-(5n+1) = 3$
  - $\mu^-(5n+2) = 3$
  - $\mu^-(5n+3) = 2$
  - $\mu^-(5n+4) = 2$

ce qui prouve que la suite (S_n) appartient à 3.REG2
- d'autre part :
  - $\mu^+(5n) = 4$
  - $\mu^+(5n+1) = 2$
  - $\mu^+(5n+2) = 2$
  - $\mu^+(5n+3) = 3$
  - $\mu^+(5n+4) = 1$

ce qui prouve que la suite (S_n) appartient à 4.PROG2, mais n'appartient pas à 3.PROG2.■

EXEMPLE 4 :

Posons, pour tout entier n, $\theta(n) = n^2$.

Notons que :

$$\theta(n)-1 = \theta(n-1)+2(n-1)$$

et $\theta(n)-2 = \theta(n-1)+2(n-2)+1$.

Soit (S_n) une suite convergente de limite nulle vérifiant les inégalités :

$$\forall\, i,\ 1 \le i \le n, \quad S_{\theta(n-1)} < S_{\theta(n)+2i} < S_{\theta(n)} < 0,$$
$$\forall\, i,\ 0 \le i \le n-1, \quad 0 < S_{\theta(n+1)+1} < S_{\theta(n)+2i+1}.$$

$S_{\theta(n-1)}$ $S_{\theta(n)}$ 0

--⊥--------⊥-------⊥-----⊥------⊥----------⊥----------->

$S_{\theta(n)+2}$ $S_{\theta(n+1)+1}$ $S_{\theta(n)+1}$

$:$ $:$

$= S_{\theta(n)+2i}$ $= S_{\theta(n)+(2i+1)}$

$:$ $:$

$= S_{\theta(n)+2n}$ $= S_{\theta(n)+(2(n-1)+1)}$

Pour tout entier $n \ge 1$, on a :

$\mu^-(\theta(n)) = 1$,

car $S_{\theta(n)-1} < S_{\theta(n)} < 0$.

$\mu^-(\theta(n)+2) = 3$,

car $S_{\theta(n)-1} < S_{\theta(n)+2} < S_{\theta(n)} < 0 < S_{\theta(n)+1}$.

$\mu^-(\theta(n)+1) = 3$,

car $S_{\theta(n)-1} < S_{\theta(n)} < 0 < S_{\theta(n)+1} < S_{\theta(n)-2}$.

$\mu^-(\theta(n)+2i) = 2, \quad \forall\, i,\ 1 < i \le n$

$\mu^-(\theta(n)+(2i+1)) = 2, \quad \forall\, i,\ 0 < i \le n-1$.

Ce qui prouve que la suite appartient à 3.REG2.

Par contre, elle n'appartient à aucun ensemble q.PROG2, avec $q \ge 1$ et donc n'appartient pas à *.PROG2.

En effet il suffit de remarquer que, pour tout entier $n > 0$, on a :

$$\mu^+(\theta(n)) = \theta(n+1)-\theta(n) = 2n+1.■$$

En résumé nous énonçons :

<u>Proposition 12.</u>

Pour tout entier $q \ge 3$:

a) q.PROG2 $\not\subset$ q.REG2

b) q.PROG2 $\not\subset$ *.REG2

c) q.REG2 $\not\subset$ q.PROG2

d) q.REG2 $\not\subset$ *.PROG2

Nous allons maintenant étudier la décidabilité asymptotique de la question Q "quel est le signe de S_n-S^* ?"

sur chacun des ensembles introduits. Puisque 2.PROG2 et 2.REG2 sont tous les deux égaux au sous-ensemble OSC de ALT et que 1.ALT est égal à ALT, la décidabilité asymptotique de la question Q ne se pose plus que pour les ensembles q.ALT avec $q \geq 2$ ainsi que pour les ensembles q.PROG2 et q.REG2 avec $q \geq 3$. Nous allons prouver que la question Q est asymptotiquement indécidable sur chacun de ces ensembles.

4. INDECIDABILITE ASYMPTOTIQUE DE LA QUESTION Q SUR :

a) l'ensemble q.PROG2, $q \geq 3$

Théorème 8.

Pour tout entier $q \geq 3$, la question "quel est le signe de S_n-S^* ?" est asymptotiquement indécidable sur l'ensemble q.PROG2.

PREUVE

Comme 3.PROG2 $\subset$ q.PROG2, pour tout entier $q \geq 3$, il suffit de prouver que la question Q "quel est le signe de S_n-S^* ?" est asymptotiquement indécidable sur 3.PROG2.
La démonstration est identique à celle du théorème 2. Il suffit de remarquer que :
- d'une part, les suites $(T_{p,n})$ que l'on construit sont dans l'ensemble OSC donc à fortiori dans 3.PROG2,
- d'autre part, la suite (S_n) que l'on construit, pour laquelle l'algorithme Φ donne une infinité de réponses fausses appartient aussi à 3.PROG2. En effet, pour tout entier $p > 0$, nous avons deux cas de figures possibles :

- 1er cas : $S_{2p} \in <S_{2p-2},S^*<$ et $S_{2p+1} \in < S_{2p-1},S^*<$

```
                                        S*
--⊥-----------⊥-----------⊥-------⊥---⊥-----------⊥------->
S2p-2                     S2p         S2p+1       S2p-1
```

Dans ce cas, nous avons :

$\mu^+(2p-2) = 2$
$\mu^+(2p-1) = 2$

- 2ème cas : $S_{2p} \in <S_{2p-2},S^*<$ et $S_{2p+1} \in < S_{2p-1},S^*<$

```
                        S*
--⊥-----------⊥------⊥----⊥-----------⊥-----------⊥------->
S2p-2         S2p+1       S2p                     S2p-1
```

Et dans ce cas, nous avons :

$\mu^+(2p-2) = 3$
$\mu^+(2p-1) = 1$ ■

Comme l'ensemble q.PROG2 est inclus dans le sous-ensemble q.PROG de C^*, nous déduisons immédiatement le corollaire suivant :

Corollaire 2.

> Pour tout entier $q \geq 3$, la question "quel est le signe de $S_n - S^*$?" est asymptotiquement indécidable sur l'ensemble q.PROG .

b) l'ensemble q.REG2, $q \geq 3$

Théorème 9.

> Pour tout entier $q \geq 3$, la question "quel est le signe de $S_n - S^*$?" est asymptotiquement indécidable sur l'ensemble q.REG2.

PREUVE

Comme 3.REG2 $\subset$ q.REG2, pour tout entier $q \geq 3$, il suffit de prouver que la question Q "quel est le signe de $S_n - S^*$?" est asymptotiquement indécidable sur 3.REG2.
La démonstration est elle aussi identique à celle du théorème 2. Il suffit de remarquer que :
- d'une part, les suites $(T_{p,n})$ que l'on construit sont dans l'ensemble OSC donc à fortiori dans 3.REG2,
- d'autre part, la suite (S_n) que l'on construit, appartient aussi à 3.REG2. En effet, reprenons les deux cas possibles (voir les illustrations dans la preuve du théorème 8) :
- 1er cas : $S_{2p} \in \langle S_{2p-2}, S^* \langle$ et $S_{2p+1} \in \langle S_{2p-1}, S^* \langle$
nous avons alors :
$\mu^-(2p) = 2$
$\mu^-(2p+1) = 2$
- 2ème cas : $S_{2p} \in \langle S_{2p-2}, S^* \langle$ et $S_{2p+1} \in \langle S_{2p-1}, S^* \langle$
Et dans ce cas, nous avons :
$\mu^-(2p) = 1$
$\mu^-(2p+1) = 3$ ■

Comme l'ensemble q.REG2 est inclus dans le sous-ensemble q.REG de C^*, nous pouvons énoncer le corollaire suivant :

Corollaire 3.

Pour tout entier $q \geq 3$, la question "quel est le signe de $S_n - S^*$?" est asymptotiquement indécidable sur l'ensemble q.REG .

c) l'ensemble q.ALT, $q \geq 2$

Théorème 10.

Pour tout entier $q \geq 2$, la question "quel est le signe de $S_n - S^*$?" est asymptotiquement indécidable sur l'ensemble q.ALT.

PREUVE

Comme 2.ALT $\subset$ q.ALT, pour tout entier $q \geq 2$, il suffit de prouver que la question Q "quel est le signe de $S_n - S^*$?" est asymptotiquement indécidable sur 2.ALT.
La démonstration est encore une fois identique à celle du théorème 2. Il suffit de remarquer que :
- d'une part, les suites $(T_{p,n})$ que l'on construit sont dans l'ensemble OSC donc à fortiori dans 2.ALT,
- d'autre part, la suite (S_n) que l'on construit, appartient aussi à 2.ALT. En effet, reprenons les deux cas possibles (voir les illustrations dans la preuve du théorème 8) :
- 1er cas : $S_{2p} \in \langle S_{2p-2}, S^* \langle$ et $S_{2p+1} \in \langle S_{2p-1}, S^* \langle$
nous avons alors :

$$\alpha^-(2p) = 1$$
$$\alpha^-(2p+1) = 1$$

- 2ème cas : $S_{2p} \in \langle S_{2p-2}, S^* \langle$ et $S_{2p+1} \in \langle S_{2p-1}, S^* \langle$
Et dans ce cas, nous avons :

$$\alpha^-(2p) = 2$$
$$\alpha^-(2p+1) = 1 \blacksquare$$

Contrairement au cas des suites monotones, il y a une seule généralisation de la notion de suites alternées et elle ne fournit que des résultats d'indécidabilité de la question Q. En effet, on est seulement capable de dire : la limite appartient à un segment contenant au moins 3 termes consécutifs de la suite; mais on ne peut pas situer cette limite par rapport à un de ces termes autre que les deux extrémités.

E. ETUDE DE SOUS-ENSEMBLES DE C*

Dans les paragraphes précédents, nous avons étudié la décidabilité de la question Q "quel est le signe de S_n-S* ?" sur des sous-ensembles de C1 ou de C2. Il reste maintenant à compléter ces résultats pour des sous-ensembles de C* inclus ni dans C1 ni dans C2.

Nous avions introduit deux types de sous-ensembles de C* : les ensembles q.PROG des suites q-progressives et les ensembles q.REG des suites q-régressives, où q est un entier non nul.

Les théorèmes 8 et 9 nous ont déjà permis d'affirmer que la question Q était asymptotiquement indécidable sur chacun des ensembles q.PROG et q.REG lorsque q≥3. Il nous reste donc à faire l'étude de 2.PROG et 2.REG. D'après la proposition 10, nous avons :

2.PROG = 2.PROG1 U OSC
2.REG = 2.REG1 U OSC

Donc chacun des ensembles 2.PROG et 2.REG est la réunion de deux ensembles sur lesquelles la question Q est asymptotiquement décidable (cf. théorèmes 4, 5, proposition 6).

Nous montrons ici que la question Q est asymptotiquement décidable sur l'ensemble 2.PROG tandis qu'elle est asymptotiquement indécidable sur l'ensemble 2.REG. Ce résultat est intéressant car il montre que la question Q peut être asymptotiquement décidable sur deux ensembles sans l'être sur leur réunion, et il met en évidence une différence entre les ensembles q.PROG et q.REG .

1. DECIDABILITE ASYMPTOTIQUE DE LA QUESTION Q SUR L'ENSEMBLE 2.PROG

Théorème 11.

La question "quel est le signe de S_n-S* ?" est asymptotiquement décidable sur l'ensemble 2.PROG.

Nous avons : 2.PROG = 2.PROG1 U OSC, où 2.PROG1 désigne l'ensemble des suites 2-progressives de C1 et OSC désigne le

sous-ensemble de C2 formé par les suites oscillantes. Et comme $C1 = C1^- \cup C1^+$, nous avons :
$2.PROG = (2.PROG1 \cap C1^-) \cup (2.PROG1 \cap C1^+) \cup OSC$.

Nous devons donc construire un algorithme tel que, pour toute suite (S_n) de 2.PROG, la suite (R_n) des réponses fournies par l'algorithme vérifie :
- $\exists\, N_0, \forall\, n \geq N_0 : R_n = -1$ si $(S_n) \in 2.PROG1 \cap C1^-$,
- $\exists\, N_0, \forall\, n \geq N_0 : R_n = +1$ si $(S_n) \in 2.PROG1 \cap C1^+$,
- $\exists\, N_0, \forall\, n \geq N_0 : R_n = (-1)^n$ si $(S_n) \in OSC$.

Dans l'algorithme nous utiliserons une variable REP qui pourra prendre trois valeurs, OSC, $C1^-$ ou $C1^+$, suivant que l'algorithme considère - jusqu'à preuve du contraire - que la suite (S_n) appartient à OSC, $C1^-$ ou $C1^+$. Cette variable servira uniquement pour la preuve de l'algorithme.

Nous noterons C'(n) la proposition : $S_{n-2} \leq S_n < S_{n-1} \leq S_{n-3}$
et C"(n) la proposition : $S_{n-3} \leq S_{n-1} < S_n \leq S_{n-2}$

description de l'algorithme

```
| n <- 2;
| Répéter à l'infini
|    (a) tant que S(n-2) ≤ max (S(n-1),S(n))
|                    {la suite appartient à C1-}
|                    faire
|                    REP <- C1-;
|                    R(n) <- -1;
|                    n <- n+1
|                    fin du tant que (a);
|
|    (b) tant que S(n-2) ≥ min (S(n-1),S(n))
|                    {la suite appartient à C1+}
|                    faire
|                    REP <- C1+;
|                    R(n) <- +1;
|                    n <- n+1
|                    fin du tant que (b);
|
|    (c) tant que (C'(n) et C"(n-1)) ou (C'(n-1) et C"(n))
|                    {la suite appartient à OSC}
|                    faire
|                    REP <- OSC;
|                    Si C'(n) et C"(n-1)    alors R(n) <- -1;
|                                           sinon R(n) <- +1;
|                    n <- n+1
|                    fin du tant que (c)
| fin du répéter.
```

PREUVE DE L'ALGORITHME

Etant donnée une suite (S_n), notons :
$C_a(n)$ la proposition : $S_{n-2} \leq \max (S_{n-1}, S_n)$
$C_b(n)$ la proposition : $S_{n-2} \geq \min (S_{n-1}, S_n)$
$C_c(n)$ la proposition : (C'(n) et C"(n-1)) ou (C'(n-1) et C"(n)).
La proposition $C_a(n)$ caractérisera les suites de $C1^- \cap$ 2.PROG1.
La proposition $C_b(n)$ caractérisera les suites de $C1^+ \cap$ 2.PROG1.
La proposition $C_c(n)$ caractérisera les suites de OSC.

Fait 1 : pour tout entier n, on a $C_a(n)$ ou $C_b(n)$.

En conséquence, lors du déroulement de l'algorithme, quelque soit l'entier N, il existera un instant t tel que la valeur de n à l'instant t sera égale à N. Nous dirons "la valeur de N est atteinte".

- Si (S_n) est une suite de 2.PROG1 $\cap$ $C1^-$, nous allons montrer qu'à partir d'un certain rang, l'algorithme boucle dans le tant que (a), i.e. la valeur de la variable REP sera asymptotiquement constante égale à $C1^-$.

(S_n) vérifie les trois faits suivants :

Fait 2 : $\exists$ N, $\forall$ n $\geq$ N : on a $C_a(n)$
ce fait résulte du lemme 1.

Fait 3 : $\forall$ N', $\exists$ m $\geq$ N' tel que l'on ait non $C_b(m)$
ce fait résulte du corollaire 1 du lemme 3.

Fait 4 : $\forall$ N', $\exists$ m $\geq$ N' tel que l'on ait non $C_c(m)$

Preuve du fait 4 :
Raisonnons par l'absurde.
Supposons qu'il existe un entier N' tel que
(i) $\forall$ n $\geq$ N', $C_c(n)$
On peut supposer que l'on a C'(N').
Montrons tout d'abord par récurrence que,
(ii) $\forall$ n $\geq$ N', n $\equiv$ N' (mod 2) => C'(n)
(la congruence sera désormais simplement notée n $\equiv$ N')

C'est vrai pour n = N'.
supposons que l'on ait C'(n), avec n $\equiv$ N',
et montrons que l'on a C'(n+2).
C'(n) => non C"(n)
or $C_c(n+1)$ et non C"(n) => C"(n+1)
donc on a : non C'(n+1)

or $C_c(n+2)$ et non $C'(n+1)$ => $C'(n+2)$.
Alors d'après la propriété (ii) et le fait 2, nous avons pour tout entier $n \geq N'$ tel que $n \equiv N'$:

$$S_{n-2} \leq S_n < S_{n-1} \leq S_{n-3} \leq \max (S_n, S_{n-1}) = S_{n-1}$$

donc : $S_{n-3} = S_{n-1}$
Par conséquent, la sous-suite $(S_{n-1})_{n \equiv N'}$ serait stationnaire, ce qui est contraire à $(S_n) \in C^*$.
{fin de la preuve du fait 4} ∎

D'après le fait 1, la valeur N sera atteinte.
Et pour tout $n \geq N$, nous serons après toute incrémentation $n \leftarrow n+1$, dans l'un des trois cas suivants :

1er cas : REP = $C1^-$
on était donc dans le tant que (a).
alors d'après le fait 2, pour tout $m \geq n$, on aura :

$C_a(m)$ et REP = $C1^-$,

i.e. on reste dans le tant que (a).
Par suite on aura bien : $\forall\ m \geq n,\ R_m = -1$.

2ème cas : REP = $C1^+$
on était donc dans le tant que (b).
alors d'après le fait 3, il existe $m \geq n$ tel que l'on ait non $C_b(m)$,
i.e. on ne peut pas boucler dans le tant que (b).
On va alors tester la condition $C_c(m)$ et on est ramené au 3éme cas.

3ème cas : REP = OSC ou (REP = $C1^+$ et non $C_b(n)$)
c'est-à-dire on était dans le tant que (c) ou bien on vient de sortir du tant que (b).
alors, d'après le fait 4, il existe $m \geq n$ tel que l'on ait non $C_c(m)$;
i.e. on ne peut pas boucler dans le tant que (c).
on va alors tester la condition $C_a(m)$ qui est vraie (car $n \geq N$) et on se ramène au 1er cas.

- Si (S_n) est une suite de 2.PROG1 $\cap$ $C1^+$, on montre de façon analogue qu'à partir d'un certain rang, l'algorithme boucle dans le tant que (b), i.e. la valeur de la variable REP sera asymptotiquement constante égale à $C1^+$.

- Enfin si (S_n) est une suite de OSC, nous allons montrer qu'à partir d'un certain rang, l'algorithme boucle dans le tant que (c), i.e. la valeur de la variable REP sera asymptotiquement constante égale à OSC.

(S_n) vérifie les trois faits suivants :

Fait 5 : il existe N tel que, pour tout $n \geq N$:

$n \equiv N \Rightarrow C'(n)$ et $C''(n-1)$

$n \equiv N+1 \Rightarrow C''(n)$ et $C'(n-1)$

pour tout $n > N$, on aura donc $C_c(n)$.
ce fait résulte de la définition des suites oscillantes.

Fait 6 : $\forall N'$, $\exists\, m \geq N'$ tel que non $C_a(m)$
car la sous-suite $(S_n)_{n\equiv N}$ converge vers S^* en croissant et appartient à C^*.

Fait 7 : $\forall N'$, $\exists\, m \geq N'$ tel que non $C_b(m)$
car la sous-suite $(S_n)_{n\equiv N+1}$ converge vers S^* en décroissant et appartient à C^*.

La valeur N+1 sera atteinte d'après le fait 1.
Et pour tout $n \geq N$, nous serons après toute incrémentation $n \leftarrow n+1$, dans l'un des trois cas suivants :

1er cas : REP = OSC
on était donc dans le tant que (c).
alors d'après le fait 5, pour tout $m \geq n$, on aura :

$C_c(m)$ et REP = OSC.

Donc on reste dans le tant que (c) et plus précisément :
si $m \equiv N$, alors on aura : ($C'(m)$ et $C''(m-1)$)
et donc : $R_m = -1$
si $m \equiv N+1$, alors on aura : ($C''(m)$ et $C'(m-1)$)
et donc : $R_m = +1$.

2ème cas : REP = $C1^-$
on était donc dans le tant que (a).
alors, d'après le fait 6, il existe $m \geq n$ tel que l'on ait non $C_a(m)$
i.e. on ne peut pas boucler dans le tant que (a).
On va alors tester la condition $C_b(m)$ et on est ramené au 3éme cas.

3ème cas : REP = $C1^+$ ou (REP = $C1^-$ et non $C_a(n)$)
c'est-à-dire on était dans le tant que (b) ou bien on vient de sortir du tant que (a).
alors, d'après le fait 7, il existe $m \geq n$ tel que l'on ait non $C_b(m)$,
i.e. on ne peut pas boucler dans le tant que (b).
on va alors tester la condition $C_c(m)$, qui est vrai (car $m > N$) et on se ramène au 1er cas.∎

2. INDECIDABILITE ASYMPTOTIQUE DE LA QUESTION Q SUR L'ENSEMBLE 2.REG

Théorème 12.

La question "quel est le signe de $S_n - S^*$?" est asymptotiquement indécidable sur l'ensemble 2.REG.

PREUVE

Rappelons que 2.REG = 2.REG1 U OSC, où 2.REG1 contient toutes les suites 2-régressives de C1 et où OSC désigne le sous-ensemble de C2 formé par les suites oscillantes.
Faisons un raisonnement par l'absurde.
Supposons que la question "quel est le signe de $S_n - S^*$?" soit asymptotiquement décidable sur 2.REG, c'est-à-dire qu'il existe un algorithme, noté Φ, qui transforme toute suite (S_n) de 2.REG en une suite (R_n) telle que :

$\forall n \geq 0,\ R_n \in \{-1,+1\}$ et

$\exists N,\ \forall n \geq N : R_n\ (S_n - S^*) > 0.$

En particulier, l'algorithme Φ fournit, pour toute suite (S_n) de 2.REG1 $\cap$ C1$^+$, une suite (R_n) telle que :

(1) $\exists N,\ \forall n \geq N : R_n = +1$

Nous allons alors construire une suite (S_n) de 2.REG1 strictement positive, convergeant vers zéro, formée d'une succession de débuts de suites oscillantes convergeant vers des valeurs de plus en plus petites $(1, 1/4, \ldots, 1/2^{2p}, \ldots)$, de telle sorte que la suite (R_n) des réponses fournies par l'algorithme Φ appliqué à la suite (S_n) ne vérifiera pas la propriété (1), c'est-à-dire comportera une infinité de réponses fausses.

Construction des suites oscillantes :
Par récurrence sur l'entier p, nous construisons une suite de suites $(T_{p,n})$ oscillantes telles que :

$\forall p \geq 0,\ \lim\limits_{n \to +\infty} T_{p,n} = 1/2^{2p}$

La suite $(T_{0,n})$ est définie par :

$T_{0,1} = 1/2$

$T_{0,n} = 1 - (T_{0,n-1} - 1)/2$ pour tout $n > 1$

La suite $(T_{0,n})$ appartient à OSC; elle converge vers 1
Désignons par $R_{0,n}$ la réponse fournie à l'étape n par l'algorithme Φ appliqué à la suite $(T_{0,n})$, alors :

il existe un entier $\beta(0) > 1$ tel que : $R_{0,\beta(0)} = -1$

Supposons maintenant la suite $(T_{p-1,n})$ construite, avec $p>1$. La suite $(T_{p,n})$ sera alors définie par :

$\forall\, n \le \beta(p),\ T_{p,n} = T_{p-1,n}$

$T_{p,\beta(p)+1} = 1/2^{2p+1}$

$\forall\, n > \beta(p-1)+1,\ \ T_{p,n} = 1/2^{2p} - (T_{p,n-1} - 1/2^{2p})/2$

La suite $(T_{p,n})$ appartient à OSC; elle converge vers $1/2^{2p}$.
Désignons par $R_{p,n}$ la réponse fournie à l'étape n par l'algorithme Φ appliqué à la suite $(T_{p,n})$. On a :

il existe un entier $\beta(p) > \beta(p-1)$ tel que :

$R_{p,\beta(p)} = -1$

Construction de la suite (S_n) :
Posons pour tout entier $p \ge 0$ et pour tout entier $n \le \beta(p)$

$S_n = T_{p,n}$

La suite (S_n) ainsi construite vérifie :

$\forall\, n > 0,\ S_n > 0$

$\forall\, p > 0,\ \ \mu^-(\beta(p)) = 1$ si $\beta(p)-\beta(p-1)$ est impair
2 si $\beta(p)-\beta(p-1)$ est pair

$\mu^-(\beta(p)+2i) = 2$, pour tout $i > 0$

$\mu^-(\beta(p)+2i+1) = 1$, pour tout $i \ge 0$

D'autre part :

$\forall\, p \ge 0,\ \forall\, n \ge \beta(p) : 0 < S_n < 1/2^{2p}$

La suite (S_n) converge donc vers zéro, et appartient à 2.REG1 $\cap$ $C1^+$.

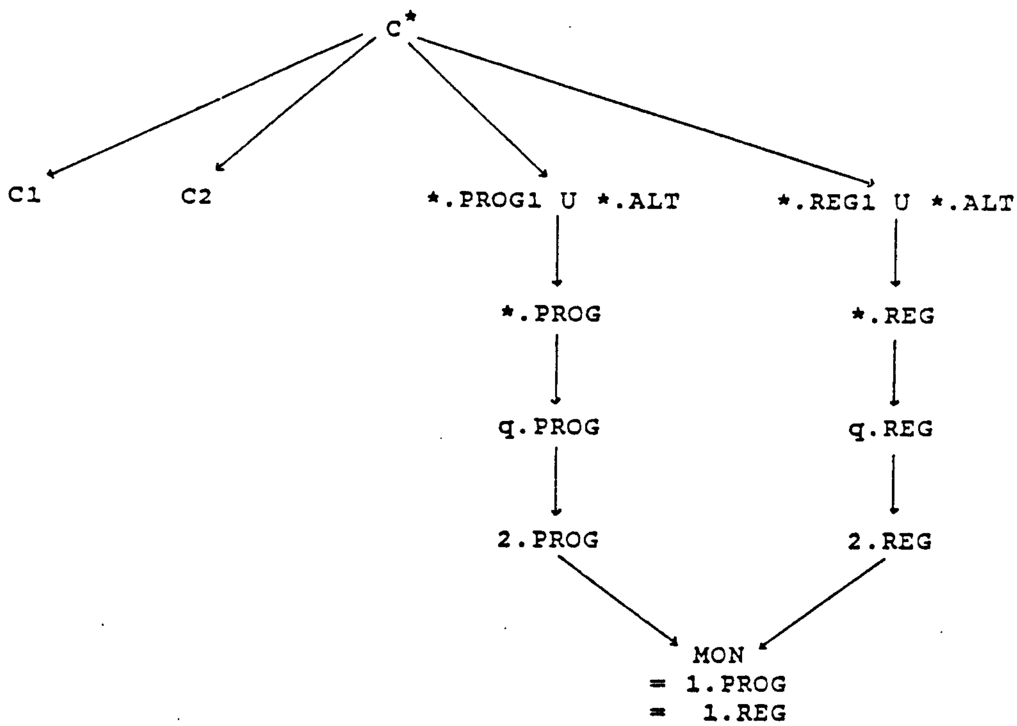
Si on applique l'algorithme Φ à la suite (S_n) et si on désigne par R_n la réponse fournie à l'étape n, on a d'après le lemme 0, pour tout entier $p \ge 0$: $R_{\beta(p)} = R_{p,\beta(p)} = -1$, et donc la suite (R_n) des réponses fournies par l'algorithme Φ appliqué à la suite (S_n) ainsi construite ne vérifie pas la propriété (1).∎

A l'issue de ce premier chapitre, plusieurs faits méritent d'être relevés. Les divers sous-ensembles de C^* ont été définis uniquement par des propriétés faisant intervenir, pour toute suite, sa limite et des termes consécutifs (appartenance ou non à certains segments).

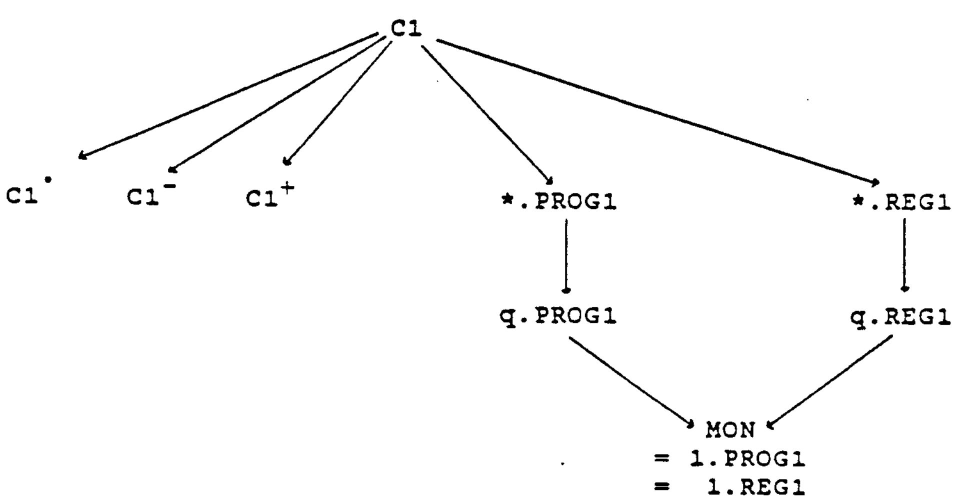
Sur C1, qui est défini par une propriété asymptotique, on n'a que des résultats de décidabilité de la question Q (sauf sur C1 lui-même !). En revanche, sur C2, qui n'est pas défini par une propriété asymptotique, on n'a (hormis sur ALT) que des résultats d'indécidabilité (même sur C2 !!).

En outre, il apparaît ici une différence entre les suites progressives et les suites régressives qui semble difficile à justifier : la question Q est asymptotiquement décidable sur 2.PROG, elle ne l'est pas sur 2.REG.

F. TABLEAUX RECAPITULATIFS



Sous-ensembles de C*



Sous-ensembles de C1

$$
\begin{tikzcd}
 & \mathbf{C2} \arrow[d] & \\
 & \mathbf{*.ALT} \arrow[dl] \arrow[d] \arrow[dr] & \\
\mathbf{*.PROG2} \arrow[d] & \mathbf{q.ALT} \arrow[dl] \arrow[d] \arrow[dr] & \mathbf{*.REG2} \arrow[d] \\
\mathbf{q.PROG2} \arrow[dr] & \mathbf{ALT} \arrow[d] & \mathbf{q.REG2} \arrow[dl] \\
 & \begin{array}{c} \mathbf{OSC} \\ = \mathbf{2.PROG2} \\ = \mathbf{2.REG2} \end{array} \arrow[d] & \\
 & \begin{array}{c} \emptyset \\ = \mathbf{1.PROG2} \\ = \mathbf{1.REG2} \end{array} &
\end{tikzcd}
$$

<u>Sous-ensembles de C2</u>

CHAPITRE II.

EQUIVALENCE

ENTRE

SYNCHRONISATION

ET

ACCELERATION DE LA CONVERGENCE

A. INTRODUCTION

1. ACCELERATION DE LA CONVERGENCE

DEFINITION :

Soient (S_n) et (A_n) deux suites convergentes de limites respectives S^* et A^*.
(A_n) converge plus vite que (S_n) si et seulement si :

$$\lim_{n \to +\infty} \frac{A_n - A^*}{S_n - S^*} = 0$$

Pour qu'il existe une suite (A_n) qui converge plus vite que (S_n), il est nécessaire que $(S_n) \in C^*$.

DEFINITION :

Soit E un sous-ensemble de C^* et PA une transformation algorithmique normale définie sur E.
PA est un procédé d'accélération de E si et seulement si, pour toute suite (S_n) de E de transformée (A_n) par PA,
(a) $\lim\limits_{n \to +\infty} A_n = \lim\limits_{n \to +\infty} S_n$
(a') (A_n) converge plus vite que (S_n).

DEFINITION :

Soit E un sous-ensemble de C^*.
E est accélérable s'il existe un procédé d'accélération PA de E.

Nous dirons alors que l'ensemble E est accélérable par le procédé PA.

2. SYNCHRONISATION

DEFINITION :

Soient (S_n) et (T_n) deux suites convergentes de limites respectives S^* et T^*.
(T_n) est **synchrone avec** (S_n) si, et seulement si, il existe un réel τ tel que $\tau \neq 1$ et

$$\lim_{n \to +\infty} \frac{T_n - T^*}{S_n - S^*} = \tau$$

Comme précédemmment, l'existence d'une suite (T_n) synchrone avec (S_n), suppose que $(S_n) \in C^*$.

remarque :
La valeur $\tau = 0$ étant permise, la relation 'est synchrone avec' n'est pas symétrique (!).

DEFINITION :

Soit E un sous-ensemble de C^* et PS une transformation algorithmique normale définie sur E.
PS est un **procédé de synchronisation** de E si et seulement si pour toute suite (S_n) de E de transformée (T_n) par PS
(b) $\lim_{n \to +\infty} T_n = \lim_{n \to +\infty} S_n$
(b') (T_n) est synchrone avec (S_n).

DEFINITION :

Soit E un sous-ensemble de C^*.
E est **synchronisable** s'il existe un procédé de synchronisation PS de l'ensemble E.

Nous dirons alors que E est synchronisable par le procédé PS.

3. PRINCIPE DU ACCES-ALGORITHME

Par définition de l'accélération de la convergence et de la synchronisation, tout ensemble accélérable est à fortiori synchronisable. L'objet de ce chapitre est de montrer la réciproque, c'est-à-dire que tout ensemble synchronisable est accélérable.

Etant donné un ensemble E synchronisable, nous allons, à partir d'un procédé de synchronisation PS de E, construire un procédé d'accélération de E, appelé ACCES-algorithme (pour (ACcélération de la Convergence d'un Ensemble Synchronisable). Le principe est le suivant :

Pour toute suite (S_n) de E, notons (T_n) la suite transformée de (S_n) par le procédé PS, et τ la limite - de valeur inconnue ! - du rapport $((T_n-S^*)/(S_n-S^*))$. Nous construisons par un procédé algorithmique une suite (Γ_n) qui aura pour limite τ. Il sera facile alors de vérifier que le procédé qui à (S_n) associe la suite (A_n) définie par :

$$A_n = S_n - \frac{T_n - S_n}{\Gamma_n - 1}$$

est bien un procédé d'accélération de E.

La difficulté réside en fait dans la construction de la suite (Γ_n). Pour construire cette suite (Γ_n), la détermination du signe de S_n-S^* a été nécessaire, c'est pourquoi dans un premier temps, nous montrons que la question Q "quel est le signe de S_n-S^* ?" est asymptotiquement décidable sur tout ensemble synchronisable.

B. LA QUESTION Q "QUEL EST LE SIGNE DE S_n-S^* ?" EST ASYMPTOTIQUEMENT DECIDABLE SUR TOUT SOUS-ENSEMBLE DE C* SYNCHRONISABLE

1. PRINCIPE DU DAQES-ALGORITHME

Soit E un sous-ensemble de C*, synchronisable par le procédé de synchronisation PS. Nous montrons ici que la question Q "quel est le signe de S_n-S^* ?" est asymptotiquement décidable sur E. Plus précisément, nous allons définir une transformation algorithmique normale qui transforme toute suite (S_n) de E en une suite (R_n) (la suite des réponses à la question Q) de telle sorte que $R_n \in \{-1,+1\}$ et que R_n soit asymptotiquement égal au signe de S_n-S^*, c'est-à-dire :

$$\exists N, \forall n \geq N : R_n (S_n-S^*) > 0.$$

L'algorithme proposé sera appelé le DAQES-algorithme (pour Décidabilité Asymptotique de la question Q "quel est le signe de S_n-S^* ?" sur un Ensemble Synchronisable).

Etant donnée une suite (S_n) de E, nous désignons par (T_n) la suite transformée de (S_n) par le procédé de synchronisation PS. Définissons la suite (τ_n) des rapports de synchronisation de (S_n) par le procédé PS :

DEFINITION :

Soient E un sous-ensemble de C* synchronisable par le procédé PS et (S_n) une suite de E.
La suite (τ_n) des rapports de synchronisation de (S_n) par le procédé PS est définie par :

$$\tau_n = \frac{T_n-S^*}{S_n-S^*} \quad \text{si } S_n \neq S^*$$

$$\tau_n = \infty \quad \text{si } S_n = S^*$$

Le DAQES-algorithme va utiliser l'égalité fondamentale suivante qui lie les termes S_n, T_n et τ_n.

<u>Lemme 8.</u>

Soient E un sous-ensemble de C^* synchronisable par le procédé PS.
Soit (S_n) une suite de E et $N^* \in N$ tel que :

$$\forall\, n \geq N^*, \quad S_n \neq S^*$$

Alors nous avons l'égalité :

$$\forall\, n \geq N^*, \quad T_n - S_n = (\tau_n - 1)(S_n - S^*).$$

Cette égalité vient du fait que :

$$\forall\, n \geq N^*, \quad \tau_n = \frac{T_n - S^*}{S_n - S^*}$$

$T_n - S_n$ étant connu, pour évaluer le signe de $S_n - S^*$, il suffit donc de connaître le signe de $\tau_n - 1$.

Or τ_n ne peut être calculé par son expression en fonction de T_n, S_n et S^*, puisque, évidemment, on ne connaît pas S^*.

Mais par définition du procédé de synchronisation, nous avons :

$$\lim_{n \to +\infty} \tau_n = \tau \neq 1,$$

Il existe donc un réel strictement positif α et un entier $p \geq N^*$, - α et p dépendent de (S_n) - tels que :

(1) $\quad \forall\, n \geq p, \quad \tau_n - 1 \leq -\alpha$

ou

(2) $\quad \forall\, n \geq p, \quad \tau_n - 1 \geq \alpha.$

Nous montrons que l'on peut construire un algorithme qui, pour toute suite (S_n) de E, détermine asymptotiquement une valeur α et un entier p tels que la suite (S_n) vérifie (1) ou (2). De plus l'algorithme pourra déterminer (toujours asymptotiquement) laquelle des deux conditions (1) ou (2) la suite (S_n) vérifie. Ce qui permettra donc asymptotiquement de déterminer de façon exacte le signe de $\tau_n - 1$ et par conséquent celui de $S_n - S^*$.

L'algorithme utilisera d'autres suites construites à partir de (S_n) et (T_n), le procédé de construction de ces suites étant aussi un procédé de synchronisation de l'ensemble E. Nous définirons ces procédés dans le paragraphe B.2 . Nous utiliserons aussi certains sous-ensembles de E, notés $E^-(p,\alpha)$ et $E^+(p,\alpha)$, que nous allons définir et étudier dans le paragraphe B.3 . Nous décrirons alors dans le paragraphe B.4 l'algorithme de décidabilité de la question Q "quel est le signe de $S_n - S^*$?" sur tout ensemble E synchronisable. Et, pour finir, nous donnerons dans le paragraphe B.5 d'autres algorithmes de décidabilité de la question Q sur E lorsque E est un sous-ensemble particulier de C^* inclus soit dans C2, soit dans *.PROG1 U *.ALT, soit dans *.REG1 U *.ALT.

2. FAMILLE DE PROCEDES DE SYNCHRONISATION

Soit E un sous-ensemble de C^* et PS une transformation algorithmique normale définie sur E. Pour tout réel a, on peut définir le procédé PS_a, associé à PS, par :

$$PS_a = a\ PS + (1-a)\ Id_E$$

où Id_E est la transformation algorithmique qui, à toute suite (S_n) de E, associe (S_n) elle-même.

remarque 1 :
1) Pour a = 0, on retrouve la transformation identique, i.e. :
$PS_0 = Id_E$.
2) Pour a = 1, on retrouve le procédé initial, i.e. :
$PS_1 = PS$.

remarque 2 :
Pour tous réels a et b, on a : $(PS_a)_b = PS_{ab}$.
En effet :

$$\begin{aligned}(PS_a)_b &= b\ PS_a + (1-b)\ Id_E \\ &= b\ [a\ PS + (1-a)\ Id_E] + (1-b)\ Id_E \\ &= ab\ PS + (1-ab)\ Id_E \\ &= PS_{ab}.\end{aligned}$$

Etant donnée une suite (S_n) de E, nous noterons (T_n) la suite transformée de (S_n) par le procédé PS et $(T_a(n))$ la suite transformée de (S_n) par le procédé PS_a. Nous avons :

$$\forall\ n \geq 0, \quad T_a(n) = a\ T_n + (1-a)\ S_n.$$

remarque 3 :
Pour tout réel a, les suites $(T_a(n))$ et $(T_{-a}(n))$ transformées de la suite (S_n) par les procédés PS_a et PS_{-a} vérifient :
$\forall\ n \geq 0$, S_n est le milieu du segment $<T_a(n),T_{-a}(n)>$.

```
------------┴-------------┴-------------┴----------------->
          Ta(n)           Sn          T-a(n)
```

Enfin, lorsque PS est un procédé de synchronisation de l'ensemble E, nous avons le résultat suivant :

<u>Proposition 13.</u>

Soit E un sous-ensemble de C^* synchronisable par le procédé PS. Alors, pour tout réel a non nul, le procédé PS_a est aussi un procédé de synchronisation de E.

PREUVE

Soit (S_n) une suite de E de limite S^*, (T_n) la suite transformée de (S_n) par le procédé PS et $(T_a(n))$ la suite transformée de (S_n) par le procédé PS_a.
D'une part nous avons :

$$\lim_{n\to+\infty} T_a(n) = S^*$$

D'autre part la suite $(\tau_a(n))$ des rapports de synchronisation de (S_n) par le procédé PS_a admet une limite $\tau_a \neq 1$.
En effet :
Soit $N^* \in N$ tel que : $\forall\, n \geq N^*,\ S_n \neq S^*$.
Alors pour tout $n \geq N^*$, on a :

$$\tau_a(n)-1 = \frac{T_a(n)-S_n}{S_n-S^*} = \frac{a(T_n-S_n)}{S_n-S^*} = a\ (\tau_n-1)$$

Et donc la suite $(\tau_a(n))$ converge vers $\tau_a = 1+a(\tau-1)$, avec $\tau_a \neq 1$ car $a \neq 0$ et $\tau \neq 1$.
Ce qui prouve que $(T_a(n))$ est synchrone avec (S_n).■

Etant donné un ensemble E inclus dans C^* et PS un procédé de synchronisation de E, nous pouvons donc définir une famille de procédés de synchronisation de l'ensemble E associés au procédé PS.

DEFINITION :

Soit E un sous-ensemble de C^* synchronisable par le procédé PS.
Posons : $F(PS) = \{ PS_a,\ a \in R^* \}$
$F(PS)$ est appelée famille des procédés de synchronisation de E associés à PS.

<u>3. ENSEMBLES $E^-(p,\alpha)$ ET $E^+(p,\alpha)$</u>

DEFINITIONS :

Soit E un ensemble synchronisable.
Soient un réel $\alpha > 0$ et p un entier.
Pour toute suite (S_n) de E, nous posons :
$(S_n) \in E^-(p,\alpha) \iff$ pour tout $n \geq p$, $\tau_n \leq 1-\alpha$ ou $\tau_n = \infty$
$(S_n) \in E^+(p,\alpha) \iff$ pour tout $n \geq p$, $\tau_n \geq 1+\alpha$ ou $\tau_n = \infty$

Commençons par donner quelques propriétés d'inclusion immédiates sur ces ensembles.

Lemme 9.

Soit E un ensemble synchronisable.
Pour tous réels α et α' tels que $0 < \alpha' < \alpha$ et pour tout entier p, on a les inclusions suivantes :

(i) $E^-(p,\alpha) \subset E^-(p+1,\alpha)$
$E^-(p,\alpha) \subset E^-(p,\alpha')$

(ii) $E^+(p,\alpha) \subset E^+(p+1,\alpha)$
$E^+(p,\alpha) \subset E^+(p,\alpha')$

PREUVE

Montrons par exemple les inclusions (i).
Soit $(S_n) \in E^-(p,\alpha)$. Alors :
$$\forall n \geq p,\ \tau_n \leq 1-\alpha \text{ ou } \tau_n = \infty$$
à fortiori on a :
$$\forall n \geq p+1,\quad \tau_n \leq 1-\alpha \quad \text{ou} \quad \tau_n = \infty$$
et $\forall n \geq p, \quad \tau_n \leq 1-\alpha' \quad \text{ou} \quad \tau_n = \infty$
Donc $(S_n) \in E^-(p+1,\alpha)$ et $(S_n) \in E^-(p,\alpha')$.■

L'introduction des ensembles $E^-(p,\alpha)$ et $E^+(p,\alpha)$ est justifiée par les deux lemmes suivants.

Lemme 10.

Soit E un ensemble synchronisable.
Pour toute suite (a_k) strictement décroissante de limite nulle, on a :
$$E = \bigcup_{k\geq 0} [E^-(k,a_k) \cup E^+(k,a_k)]$$

PREUVE

Nous avons vu dans le paragraphe B.1 que, pour toute suite (S_n) de E, il existe un entier p et un réel $\alpha > 0$ tels que (S_n) vérifie l'une des deux propriétés suivantes :

(1) $\forall n \geq p, \quad \tau_n-1 \leq \alpha$

(2) $\forall n \geq p, \quad \tau_n-1 \geq \alpha$

Cela peut se traduire par :
$$E = \bigcup_{p\geq 0,\alpha>0} [E^-(p,\alpha) \cup E^+(p,\alpha)]$$
Il suffit donc de montrer que, pour tout $\alpha > 0$ et tout entier p, il existe k tel que :
$$E^-(p,\alpha) \subset E^-(k,a_k)$$
$$E^+(p,\alpha) \subset E^+(k,a_k)$$
Comme la suite (a_k) converge vers 0 en décroissant, il existe un entier $k \geq p$ tel que :

$0 < a_k < \alpha$

Donc d'après le lemme 9 :

$E^-(p,\alpha) \subset E^-(k,\alpha) \subset E^-(k,a_k)$

De même pour l'autre inclusion.■

remarque :
Ce résultat reste valable si (a_k) est seulement une suite de réels strictement positifs de limite nulle.
Mais lorsque (a_k) est décroissante, pour tout entier k, nous avons les inclusions :

$E^-(k,a_k) \subset E^-(k+1,a_{k+1})$

$E^+(k,a_k) \subset E^+(k+1,a_{k+1})$.

<u>Lemme 11.</u>

Soit E un sous-ensemble de C^* synchronisable.
Soient (S_n) une suite de E et $N^* \in N$ tel que :

$\forall\, n \geq N^*,\ S_n-S^* \neq 0.$

Pour tout réel $\alpha > 0$ et tout entier p, on a :
a) Si $(S_n) \in E^-(p,\alpha)$, alors :

$\forall\, n \geq \max(p,N^*), \quad (S_n-S^*)(T_n-S_n) < 0$

b) Si $(S_n) \in E^+(p,\alpha)$, alors :

$\forall\, n \geq \max(p,N^*), \quad (S_n-S^*)(T_n-S_n) > 0$

PREUVE

Montrons par exemple a).

Si $(S_n) \in E^-(p,\alpha)$, alors on a, pour tout $n \geq \max(p,N^*)$:

$\tau_n-1 \leq -\alpha < 0$

Et d'après le lemme 8 :

$T_n-S_n = (\tau_n-1)(S_n-S^*)$

On a donc :

$\forall\, n \geq \max(p,N^*),\ (T_n-S_n)(S_n-S^*) < 0.$■

Compte tenu de ces résultats, il s'agit donc de construire un algorithme qui utilisera une suite donnée (a_k) strictement décroissante de limite nulle (par exemple $a_k = 1/2^k$) et qui, pour toute suite (S_n) de E, déterminera asymptotiquement un entier $h \geq 0$ - qui dépend de (S_n) - tels que $(S_n) \in E^-(h,a_h) \cup E^+(h,a_h)$ et ensuite déterminera asymptotiquement auquel des deux ensembles $E^-(h,a_h)$ ou bien $E^+(h,a_h)$ la suite (S_n) appartient effectivement. L'algorithme saura alors - en vertu du lemme 11 - répondre à la question Q "quel est le signe de S_n-S^* ?" et la réponse sera asymptotiquement exacte. La réponse sera :

- R_n est du signe de T_n-S_n si $(S_n) \in E^+(h,a_h)$,

- R_n est du signe opposé à celui de T_n-S_n si $(S_n) \in E^-(h,a_h)$.

Les résultats que nous allons maintenant énoncer seront utilisés dans la preuve de l'algorithme pour montrer que l'entier h peut être effectivement déterminé et que les réponses proposées par l'algorithme seront bien asymptotiquement exactes.

Commençons par donner, pour tout réel $\alpha > 0$, une caractérisation de $\tau_n \le 1-\alpha$ (resp. $\tau_n \ge 1+\alpha$), qui utilisera le procédé $PS_{-1/\alpha}$ (resp. $PS_{1/\alpha}$) défini dans le paragraphe B.2, transformant (S_n) en $(T_{-1/\alpha}(n))$ (resp. $(T_{1/\alpha}(n))$).

<u>Lemme 12.</u>

Soit E un sous-ensemble de C* synchronisable par le procédé PS et soit un réel $\alpha > 0$.
Pour toute suite (S_n) de E et pour tout entier n, on a :
(a) $\tau_n \ge 1+\alpha \iff S^* \in\ ]S_n, T_{-1/\alpha}(n)[$
(b) $\tau_n \le 1-\alpha \iff S^* \in\ ]S_n, T_{1/\alpha}(n)[$

PREUVE

$$\tau_n - 1 \ge \alpha$$

$$\iff \frac{T_n - S_n}{S_n - S^*} \ge \alpha$$

$$\iff \frac{T_n - (1+\alpha)\, S_n + \alpha\, S^*}{S_n - S^*} \ge 0$$

$$\iff \frac{-1/\alpha\; T_n + (1-(-1/\alpha))\, S_n - S^*}{S_n - S^*} \le 0$$

$$\iff \frac{T_{-1/\alpha}(n) - S^*}{S_n - S^*} \le 0$$

$$\iff S^* \in\ ]S_n, T_{-1/\alpha}(n)[$$

ce qui prouve (a).

De même :

$$\tau_n - 1 \le -\alpha$$

$$\iff \frac{T_{1/\alpha}(n) - S^*}{S_n - S^*} \le 0$$

$$\iff S^* \in\ ]S_n, T_{1/\alpha}(n)[$$

ce qui prouve (b).■

Proposition 14.

Soient E un ensemble synchronisable par le procédé PS et (S_n) une suite de E.
Pour tout réel $\alpha>0$ et pour tout entier p, les conditions suivantes sont équivalentes :

(i) $(S_n) \notin E^-(p,\alpha)$

(ii) $\exists\ n \geq p,\ \bigcap_{j=p}^{n} \langle S_j, T_{1/\alpha}(j)\rangle = \emptyset$

PREUVE

non(i) => non(ii)

Soit $(S_n) \in E^-(p,\alpha)$. Alors on a :

$\forall\ n \geq p,\ \tau_n \leq 1-\alpha$ ou $\tau_n = \infty$

c'est-à-dire :

$\forall\ n \geq p,\ \tau_n \leq 1-\alpha$ ou $S_n = S^*$

Donc d'après le lemme 12, on a :

$\forall\ n \geq p,\ S^* \in \langle S_n, T_{1/\alpha}(n)\rangle$

Donc :

$\forall\ n \geq p,\ \bigcap_{j=p}^{n} \langle S_j, T_{1/\alpha}(j)\rangle \neq \emptyset$.

(i) => (ii)

Soit $(S_n) \notin E^-(p,\alpha)$.

Alors il existe un entier m > p tel que :

$\tau_m > 1-\alpha$ et τ_m fini.

On a donc d'après le lemme 12 :

$S^* \notin \langle S_m, T_{1/\alpha}(m)\rangle$

```
         T1/α(n)      Sn
---]-----⊥---⊥--⊥------[-------[------------]------------->
 S*-δ        S*       S*+δ     Sm           T1/α(m)
```

Il existe un réel $\delta > 0$ tel que les intervalles $]S^*-\delta, S^*+\delta[$ et $\langle S_n, T_{1/\alpha}(n)\rangle$ soient disjoints.

Comme les suites (S_n) et $(T_{1/\alpha}(n))$ sont convergentes et de limite S^*, il existe un entier n > m tel que :

$S_n \in\ ]S^*-\delta, S^*+\delta[$

et $T_{1/\alpha}(n) \in\ ]S^*-\delta, S^*+\delta[$

Par conséquent :

$\langle S_m, T_{1/\alpha}(m)\rangle \cap \langle S_n, T_{1/\alpha}(n)\rangle = \emptyset$

Donc : $\bigcap_{j=p}^{n} \langle S_j, T_{1/\alpha}(j)\rangle = \emptyset$. ■

<u>Proposition 15.</u>

Soient E un ensemble synchronisable par le procédé PS et (S_n) une suite de E.
Pour tout réel $\alpha>0$ et pour tout entier p, les conditions suivantes sont équivalentes :

(i) $(S_n) \notin E^{+}(p,\alpha)$

(ii) $\exists\, n \geq p, \quad \bigcap_{j=p}^{n} \langle S_j, T_{-1/\alpha}(j)\rangle = \emptyset$

PREUVE

La démonstration est analogue à celle de la proposition 14.■

Des deux propositions précédentes, nous pouvons déduire le corollaire :

<u>Corollaire 5.</u>

Soient E un ensemble synchronisable par le procédé PS et (S_n) une suite de E.
Pour tout réel $\alpha>0$ et pour tout entier p, les conditions suivantes sont équivalentes :

(i) $(S_n) \notin E^{-}(p,\alpha) \cup E^{+}(p,\alpha)$

(ii) il existe m > p tel que :

$$\bigcap_{j=p}^{m} \langle S_j, T_{1/\alpha}(j)\rangle = \emptyset$$

$$\bigcap_{j=p}^{m} \langle S_j, T_{-1/\alpha}(j)\rangle = \emptyset$$

PREUVE

L'implication (ii) => (i) est immédiate d'après les propositions 14 et 15.
Réciproquement, si $(S_n) \notin E^{-}(p,\alpha) \cup E^{+}(p,\alpha)$, alors :

$(S_n) \notin E^{-}(p,\alpha)$

et $(S_n) \notin E^{+}(p,\alpha)$.

Donc il existe un entier N' > p tel que :

$$\bigcap_{j=p}^{N'} \langle S_j, T_{1/\alpha}(j)\rangle = \emptyset.$$

Et il existe un entier N" > p tel que :

$$\bigcap_{j=p}^{N''} \langle S_j, T_{1/\alpha}(j)\rangle = \emptyset.$$

En posant m = max(N',N"), on a bien (ii).■

Les trois résultats précédents signifient en fait que les questions suivantes sont semi-décidables :
Q1- "Est-ce que (S_n) n'appartient pas à $E^-(p,\alpha)$?"
Q2- "Est-ce que (S_n) n'appartient pas à $E^+(p,\alpha)$?"
Q3- "Est-ce que (S_n) n'appartient pas à $E^-(p,\alpha) \cup E^+(p,\alpha)$?"
c'est-à-dire que l'on peut construire un algorithme qui, pour toute suite (S_n) n'appartenant pas à l'ensemble $E^-(p,\alpha)$ (resp. $E^+(p,\alpha)$, $E^-(p,\alpha) \cup E^+(p,\alpha)$), donne la bonne réponse et, pour toutes les autres, boucle.

Par exemple, l'algorithme de semi-décidabilité de la question Q1 s'écrit, en utilisant la notation $A(p,\alpha,n)$ pour la proposition :

$$\bigcap_{j=p}^{n} \langle S_j, T_{-1/\alpha}(j)\rangle \neq \emptyset$$

```
| n <- 0;
| Répéter à l'infini
|    Si non A(p,α,n)  alors  la suite (Sn) n'appartient pas
|                            à E-(p,α);
|    n <- n+1
| fin du répéter.
```

4. DESCRIPTION DU DAQES-ALGORITHME

Tous les résultats précédents nous permettent maintenant de proposer le DAQES-algorithme et d'énoncer :

Théorème 13.

> La question Q "quel est le signe de $S_n - S^*$?" est asymptotiquement décidable sur tout ensemble synchronisable.

Pour la description du DAQES-algorithme sur un ensemble synchronisable E par un procédé PS, d'après le lemme 10, nous utiliserons une suite auxiliaire (a_k) strictement décroissante de limite nulle (par exemple $a_k = 2^{-k}$) et nous noterons, pour toute suite (S_n) de E :

$A(k,n)$ la proposition : $\bigcap_{j=k}^{n} \langle S_j, T_{-1/a_k}(j)\rangle \neq \emptyset$

$B(k,n)$ la proposition : $\bigcap_{j=k}^{n} \langle S_j, T_{1/a_k}(j)\rangle \neq \emptyset$

description de l'algorithme :

```
| k <- 0; n <- 0;
|
| Répéter à l'infini
|
|    Si  A(k,n)        alors   R_n <- signe(T_n-S_n);
|                              {(S_n) ∈ E+(k,a_k)}
|    Si  B(k,n)        alors   R_n <- - signe(T_n-S_n);
|                              {(S_n) ∈ E-(k,a_k)}
|    Si non A(k,n) et non B(k,n) alors
|                      k <- n+1;
|                      R_n <- R_{n-1};
|                      {(S_n) ∉ E-(k,a_k) U E+(k,a_k)}
|    n <- n+1
|
| fin du répéter.
```

PREUVE DE L'ALGORITHME

Notons $C(k,n)$ la proposition : non $A(k,n)$ et non $B(k,n)$.

Remarquons que, lors du déroulement de l'algorithme, quelque soit l'entier N, il existera un instant t tel que la valeur de n à l'instant t sera égale à N. D'autre part, à chaque étape n, on a $n \geq k$.

- Si (S_n) est une suite de $\bigcup_{k \geq 0} E^+(k,a_k)$,

La suite (a_k) est strictement décroissante. Nous avons donc :

$\forall\, k \geq 0,\ E^+(k,a_k) \subset E^+(k+1,a_{k+1})$

notons k_0 le plus petit entier tel que $(S_n) \in E^+(k_0,a_{k0})$.

(S_n) vérifie les trois faits suivants :

Fait 1 : $\forall\, k < k_0,\ \forall\, N',\ \exists\, m \geq N'$ tel que l'on ait $C(k,n)$.
ce fait résulte du choix de k_0 tel que $(S_n) \notin E^+(k,a_k)$ pour tout $k < k_0$ et du corollaire 5.

Fait 2 : $\forall\, k \geq k_0,\ \forall\, n \geq k$, on a : $A(k,n)$ et non $C(k,n)$.
ce fait résulte de la proposition 15 et du corollaire 5.

Fait 3 : $\forall\, k \geq k_0,\ \exists\, m_k \geq k$ tel que l'on ait :
$\forall\, n \geq m_k$, non $B(k,n)$.
ce fait résulte de la proposition 14.

D'après les trois faits précédents, k atteindra une valeur $h \geq k_0$, et pour tout $n \geq h$, on aura :

$A(h,n)$ et non $C(h,n)$,
et donc la valeur de k ne sera plus modifiée par la suite.
De plus, pour tout $n \geq m_h$, on a non $B(h,n)$.
Ce qui entraîne :

$\forall\, n \geq m_h$, R_n est du signe de T_n-S_n

et donc, d'après le lemme 11 :

$\forall\, n \geq \max(m_h,N^*)$, $R_n\ (S_n-S^*) > 0$

- La démonstration est analogue si $(S_n) \in \bigcup_{k\geq 0} E^-(k,a_k)$

Cela prouve bien que l'algorithme proposé est asymptotiquement exact sur E pour la question Q "quel est le signe de S_n-S^* ?". ■

5. AUTRES VERSIONS DU DAQES-ALGORITHME SUR DES SOUS-ENSEMBLES DE C*

Nous avons jusqu'à maintenant donné deux types d'algorithmes de décidabilité de la question Q "quel est le signe de S_n-S^* ?" sur des sous-ensembles E de C* :

Dans le chapitre I, nous avions vu que, lorsque les suites de E possèdent toutes une même propriété (être monotones, être progressives, être alternées, etc...), cette seule propriété nous permettait de proposer des algorithmes n'utilisant pas de suites auxiliaires.

Le DAQES-algorithme, lui, fonctionne sur tout sous-ensemble E de C* synchronisable et il est basé uniquement sur un procédé auxiliaire : un procédé de synchronisation PS qui fournit, pour toute suite (S_n) de E, une suite (T_n) synchrone avec (S_n).

Nous allons voir ici que, pour les sous-ensembles E de C* sur lesquels nous avons montré dans le chapitre I que la question Q était asymptotiquement indécidable, l'ajout de l'hypothèse synchronisable nous permet de proposer d'autres algorithmes que le DAQES-algorithme. Nous envisagerons les trois cas suivants :

a) E ⊂ C2
b) E ⊂ *.PROG1 ∪ *.ALT
c) E ⊂ *.REG1 ∪ *.ALT.

Ces ensembles sont en fait les "plus grands" sous-ensembles de C* introduits dans le chapitre I sur lesquels la question Q "quel est le signe de S_n-S^* ?" est asymptotiquement indécidable. Remarquons enfin que les inclusions envisagées sont nécessairement strictes, puisque tout sous-ensemble de C^* sur lequel la question Q est asymptotiquement indécidable est non synchronisable (cf. théorème 13).

a) $E \subset C2$

Lemme 13.

Soit E un sous-ensemble de C^* synchronisable par le procédé PS et (S_n) une suite de E.
Alors il existe un entier N tel que, pour tout $n \geq N$:
(1) $T_n - S_n \neq 0$
(2) $(T_n-S_n)(T_{n-1}-S_{n-1}) < 0 \iff (S_n-S^*)(S_{n-1}-S^*) < 0$

PREUVE

Comme la suite $(S_n) \in C^*$, il existe un entier N^* tel que, pour tout $n \geq N^*$, on ait :

$S_n \neq S^*$.

D'après le lemme 8, on a:

$\forall n \geq N^*,\ T_n-S_n = (\tau_n-1)(S_n-S^*)$

Il suffit alors de remarquer que, comme (τ_n-1) converge vers $\tau-1 \neq 0$, il existe $N \geq N^*$ tel que :

$\forall n \geq N,\ (\tau_n-1)(\tau_{n-1}-1) > 0.$ ■

Or par définition de l'ensemble C2, on a, pour toute suite (S_n) de C2 : $\forall N, \exists n \geq N : (S_n-S^*)(S_{n-1}-S^*) < 0$. Ce qui permet d'énoncer :

Lemme 14.

Soit E un sous-ensemble de C2 synchronisable par le procédé PS et (S_n) une suite de E.
Alors, pour tout entier n, il existe $m \geq n$ tel que l'on ait $(T_m-S_m)(T_{m-1}-S_{m-1}) < 0$.

Autrement dit, étant donnés un sous-ensemble E de C2 et (S_n) une suite de E, les changements de signe de S_n-S^* (qui sont en nombre infini) sont indiqués par les changements de signe de T_n-S_n, ce qui nous permet de proposer un nouvel algorithme de décidabilité de la question Q "quel est le signe de S_n-S^* ?", qui fonctionne sur tout sous-ensemble de C2 synchronisable.

DESCRIPTION DE L'ALGORITHME :

```
| R0 <- +1;   {initialisation arbitraire}
| n <- 1;
|
| Répéter à l'infini
|
|    Rn <- Rn-1;
|    Si (Tn-Sn)(Tn-1-Sn-1) < 0  alors  Rn <- signe (Sn-Sn-1);
|    n <- n+1
|
| fin du répéter.
```

PREUVE DE L'ALGORITHME

D'après les lemmes 13 et 14, il existe $m \geq N$ tel que l'on ait :

$(T_m-S_m)(T_{m-1}-S_{m-1}) < 0$

et $(S_m-S^*)(S_{m-1}-S^*) < 0$.

On aura alors : $R_m = \text{signe}(S_m-S_{m-1}) = \text{signe}(S_m-S^*)$

d'où : $R_m\ (S_m-S^*) > 0$

Montrons par récurrence que : $\forall\ n \geq m,\ R_n\ (S_n-S^*) > 0$

Supposons que, pour $n \geq m$, on ait : $R_n\ (S_n-S^*) > 0$.

ou bien : $(T_{n+1}-S_{n+1})(T_n-S_n) > 0$

ce qui entraîne : $(S_{n+1}-S^*)(S_n-S^*) > 0$

Et comme $R_{n+1} = R_n$ on a bien $R_{n+1}\ (S_{n+1}-S^*) > 0$.

ou bien : $(T_{n+1}-S_{n+1})(T_n-S_n) < 0$

ce qui entraîne : $(S_{n+1}-S^*)(S_n-S^*) < 0$

Et comme : $R_{n+1} = \text{signe}(S_{n+1}-S_n) = \text{signe}(S_{n+1}-S^*)$

on a bien : $R_{n+1}\ (S_{n+1}-S^*) > 0$.■

Remarquons que, lorsque E est un sous-ensemble de C1, pour toute suite (S_n) de E et pour tout entier n suffisamment grand, on a $(S_n-S^*)(S_{n-1}-S^*) > 0$ et donc aussi d'après le lemme 13 $(T_n-S_n)(T_{n-1}-S_{n-1}) > 0$. Mais, contrairement au cas où E est inclus dans C2, cela ne nous donne aucun renseignement pour connaître effectivement le signe de S_n-S^*.

b) $E \subset *.PROG1 \cup *.ALT$

Nous reprenons ici le prog-algorithme donné dans la partie I.C.6, qui fonctionne sur l'ensemble *.PROG1 des suites progressives de C1 et le complétons pour l'ensemble *.ALT. Rappelons que *.ALT est un sous-ensemble de C2, union des ensembles de suites q-alternées c'est-à-dire des ensembles q.ALT définis par :

$$(S_n) \in q.ALT \iff \exists\, N,\ \forall\, n \geq N,\ \exists\, j \text{ tel que : } 1 \leq j \leq q \text{ et } (S_n - S^*)(S_{n-j} - S^*) < 0.$$

Donc, pour n suffisamment grand, l'une au moins des quantités $S_{n-j}-S^*$, pour $1 \leq j \leq q$, sera de signe opposé à celui de S_n-S^*, ce qui - compte tenu de l'égalité $T_n-S_n = (\tau_n-1)(S_n-S^*)$ et du fait que τ_n-1 est de signe constant à partir d'un certain rang - sera indiqué par un changement de signe entre T_n-S_n et une des quantités $T_{n-j}-S_{n-j}$, pour $1 \leq j \leq q$.

En revanche, lorsque $(S_n) \in$ *.PROG1, toutes les quantités T_n-S_n ont asymptotiquement le même signe.

Nous allons donc utiliser une procédure "recherche" qui utilise les termes $S_n, \ldots, S_{n-q}, T_n, \ldots, T_{n-q}$ et détermine si les quantités $(T_{n-j}-S_{n-j})$ avec $1 \leq j \leq q$ sont toutes du même signe que la quantité (T_n-S_n) :
- si c'est le cas, le booléen changement_de_signe prendra la valeur faux (ce qui laissera supposer que $(S_n) \in$ q.PROG1),
- sinon le booléen changement_de_signe prendra la valeur vrai (ce qui laissera supposer que $(S_n) \in$ q.ALT) et k sera tel que $(T_n-S_n)(T_{n-k}-S_{n-k}) < 0$.

DESCRIPTION DE LA PROCEDURE

recherche (q, n, changement_de_signe, k)

```
| changement_de_signe <- faux;
| j <- 0;
|
| tant que (j<q et non changement_de_signe) faire
|    j <- j+1;
|    changement_de_signe <- (Tn-Sn)(Tn-j-Sn-j) < 0
| fin du tant que;
|
| k <- j.
```

DESCRIPTION DE L'ALGORITHME :

q <- 0; n <- 0;

Répéter à l'infini

q <- q+1; n <- n+2;
recherche (q,n,changement_de_signe,k)

(a) tant que ($S_{n-q} \leq V_{n,q}$ et non changement_de_signe)
faire
R_n <- -1;
n <- n+1;
recherche (q,n,changement_de_signe,k)
fin du tant que (a);

(b) tant que ($S_{n-q} \geq U_{n,q}$ et non changement_de_signe)
faire
R_n <- +1;
n <- n+1;
recherche (q,n,changement_de_signe,k)
fin du tant que (b);

(c) tant que changement_de_signe
faire
R_n <- signe(S_n-S_{n-k});
n <- n+1;
recherche (q,n,changement_de_signe,k)
fin du tant que (c);

fin du répéter.

PREUVE DE L'ALGORITHME

Etant donnée une suite (S_n), notons :
P(q,n) la proposition :
il existe j, $1 \leq j \leq q$, tel que $(T_n-S_n)(T_{n-j}-S_{n-j}) < 0$
Lorsque l'on appelle la procédure recherche avec les paramètres q et n, le booléen changement_de_signe prend la valeur vraie si et seulement si on a P(q,n).
Notons aussi :
A(q,n) la proposition :
$S_{n-q} \leq V_{n,q}$ et non P(q,n)
B(q,n) la proposition :
$S_{n-q} \geq U_{n,q}$ et non P(q,n)
La proposition P(q,n) caractérisera les suites de q.ALT,
La proposition A(q,n) caractérisera les suites de q.PROG1 $\cap$ $C1^-$.
La proposition B(q,n) caractérisera les suites de q.PROG1 $\cap$ $C1^+$.

Remarquons que, lors du déroulement de l'algorithme, quelque soit l'entier N, il existera un instant t tel que la valeur de n à l'instant t sera égale à N.

- Si (S_n) est une suite de *.ALT, notons r le plus petit entier tel que $(S_n) \in$ r.ALT. (S_n) vérifie :

Fait 1 : $\forall q \geq 1, \forall N', \exists m \geq N'$ tel que l'on ait $P(q,m)$

Cela provient de l'inclusion *.ALT $\subset$ C2 et du lemme 13.

On ne pourra donc boucler ni dans le tant que (a) ni dans le tant que (b).

Fait 2 : $\forall q < r, \forall N', \exists m \geq N'$ tel que l'on ait :
non $P(q,m)$

Cela résulte du choix de r tel que $(S_n) \notin$ q.ALT pour $q < r$ et du lemme 13.

Tant que l'on n'a pas atteint la valeur r, on ne peut donc pas boucler dans le tant que (c).

Fait 3 : $\exists N, \forall n \geq N, \forall q \geq r$: on a $P(q,n)$.

Cela résulte de la définition de r.ALT.

D'après les faits 1 et 2, q atteindra la valeur r à un instant t'.
De plus n atteindra la valeur N à un instant t".
Alors à tout instant t ≥ max(t',t"), on a :

$$q \geq r \quad \text{et} \quad n \geq N.$$

Donc d'après le fait 2, on boucle dans le tant que (c).
Et on a :

$$(T_n-S_n)(T_{n-k}-S_{n-k}) < 0.$$

Par suite, pour tout n suffisamment grand, on a :

$$(S_n-S^*)(S_{n-k}-S^*) < 0$$

D'où :

$$\text{signe}(S_n-S^*) = \text{signe}(S_n-S_{n-k})$$

et donc :

$$R_n\ (S_n-S^*) > 0.$$

- Si (S_n) est une suite de *.PROG1, nous avons le fait suivant :

Fait 4 : $\forall q \geq 1, \exists N, \forall n \geq N$:on a non $P(q,n)$.

Cela montre que l'on ne boucle pas dans le tant que (c) et que, pour tout n suffisamment grand, la condition A(q,n) (resp. B(n,q)) est vraie si et seulement si $S_{n-q} \leq V_{n,q}$ (resp. $S_{n-q} \geq U_{n,q}$).
La suite de la preuve est donc identique à celle du prog-algorithme.∎

c) E c *.REG1 U *.ALT

L'algorithme de décidabilité de la question Q "quel est le signe de S_n-S^* ?" sur l'ensemble *.REG1 U *.ALT reprend le reg-algorithme énoncé dans la partie I.C.6 ainsi que la partie concernant *.ALT donnée dans le paragraphe pécédent. Nous donnons cet algorithme sans démonstration.

DESCRIPTION DE L'ALGORITHME :

```
| q <- 0; n <- 0;
|
| Répéter à l'infini
|
|    q <- q+1; n <- n+2;
|    recherche (q,n,changement_de_signe,k)
|
|    (a) tant que (S_n ≥ U_{n-1,q} et non changement_de_signe)
|                    faire
|                    R_n <- -1;
|                    n <- n+1;
|                    recherche (q,n,changement_de_signe,k)
|                    fin du tant que (a);
|
|    (b) tant que (S_n ≤ V_{n-1,q} et non changement_de_signe)
|                    faire
|                    R_n <- +1;
|                    n <- n+1;
|                    recherche (q,n,changement_de_signe,k)
|                    fin du tant que (b);
|
|    (c) tant que changement_de_signe
|                    faire
|                    R_n <- signe(S_n-S_{n-k});
|                    n <- n+1;
|                    recherche (q,n,changement_de_signe,k)
|                    fin du tant que (c);
| fin du répéter.
```

Remarquons que le DAQES-algorithme est basé sur l'hypothèse $\tau \neq 1$, ce qui donne une deuxième justification (celle-ci à posteriori) de l'élimination de la valeur $\tau = 1$ dans la définition de la synchronisation.

D'autre part, la preuve du DAQES-algorithme, qui utilise des intersections de segments, semble difficilement généralisable à des suites d'éléments n'appartenant pas à un ensemble ordonné.

C. LE ACCES-ALGORITHME

1. DESCRIPTION DE L'ALGORITHME

Soit E un ensemble synchronisable par le procédé de synchronisation PS.

Etant donnée une suite (S_n) de E, nous noterons (T_n) la suite transformée de (S_n) par le procédé PS, et (τ_n) la suite des rapports de synchronisation de (S_n) par PS. Rappelons que, par définition du procédé de synchronisation PS :

. $\tau_n = \dfrac{T_n - S^*}{S_n - S^*}$, pour tout entier n tel que $S_n \neq S^*$.

. il existe un réel τ, $\tau \neq 1$, tel que : $\lim_{n \to +\infty} \tau_n = \tau$.

Dans cette partie, nous allons proposer un procédé d'accélération de la convergence de l'ensemble E, le ACCES-algorithme (ACcélération de la Convergence d'un Ensemble Synchronisable). Cet algorithme est basé sur la proposition suivante :

Proposition 16.

Soient E un sous-ensemble de C^* synchronisable par le procédé PS et (S_n) une suite de E de limite S^*.
Pour toute suite (Γ_n) telle que :
(i) $\lim_{n \to +\infty} \Gamma_n = \lim_{n \to +\infty} \tau_n = \tau$

La suite (A_n) définie par :

$$A_n = S_n - \frac{T_n - S_n}{\Gamma_n - 1} \quad \text{si } \Gamma_n \neq 1 \text{ (sinon } A_n = 0\text{)}$$

converge vers S^* plus vite que (S_n).

PREUVE

Comme $\lim_{n \to +\infty} T_n = \lim_{n \to +\infty} S_n = S^*$ et $\lim_{n \to +\infty} \Gamma_n - 1 = \tau - 1 \neq 0$,
la suite (A_n) converge elle aussi vers S^*.
D'autre part, pour tout entier $n \geq 0$, on a :

$$A_n - S^* = S_n - S^* - \frac{(T_n - S^*) - (S_n - S^*)}{\Gamma_n - 1}$$

Comme $(S_n) \in C^*$, il existe un entier N^* tel que :

$\forall\ n \geq N^*,\ S_n \neq S^*$.

Alors, pour tout $n \geq N^*$, on obtient :

$$\frac{A_n-S^*}{S_n-S^*} = 1 - \frac{\dfrac{T_n-S^*}{S_n-S^*} - 1}{\Gamma_n-1}$$

c'est-à-dire :

$$\frac{A_n-S^*}{S_n-S^*} = 1 - \frac{\tau_n-1}{\Gamma_n-1}$$

D'où : $\dfrac{A_n-S^*}{S_n-S^*}$ converge vers zéro.■

remarques :

- Si τ est connu, il suffit de poser $\Gamma_n = \tau$, pour tout $n \geq 0$, et on obtient le procédé d'accélération :

$$A_n = \frac{\tau\ S_n - T_n}{\tau - 1}$$

- Si l'on sait que (τ_n) converge vers 0, le procédé PS est en fait un procédé d'accélération de la convergence de l'ensemble E. En posant $\Gamma_n = 0$, pour tout $n \geq 0$, on obtient : $A_n = T_n$, si bien que le procédé que l'on "construit" n'est autre que le procédé PS lui-même.
- Si l'on a $\Gamma_n = \tau_n$, pour tout $n \geq 0$, on obtient le procédé d'accélération : $A_n = S^*$, qui constitue bien la meilleure façon d'accélérer une suite ! Il est d'ailleurs normal de retrouver $A_n = S^*$, puisque l'équation $\Gamma_n = \tau_n$ permet effectivement de déterminer S^*.

En vertu de la proposition 16, pour définir un procédé d'accélération de la convergence de l'ensemble E, il nous suffit donc de construire une suite (Γ_n) de limite τ, la construction devant se faire à l'aide d'une transformation algorithmique normale, c'est-à-dire que, pour construire le terme de rang n, Γ_n, l'algorithme devra utiliser au plus les termes S_0, S_1, ..., S_n, il pourra bien sûr utiliser tout autre quantité construite à partir de ces termes.

Pour construire (Γ_n), nous utiliserons :

- la suite (R_n) transformée de la suite (S_n) par le DAQES-algorithme. Rappelons que R_n est asymptotiquement égal au signe de S_n-S^*, c'est-à-dire qu'il existe un entier N tel que : $\forall\ n \geq N,\ R_n\ (S_n-S^*) > 0$.

- et éventuellement la suite (θ_n) transformée de (S_n) par le procédé de synchronisation PS_{-1} (rappelons que θ_n est défini par $\theta_n = 2\ S_n - T_n$, c'est-à-dire que θ_n est le symétrique de T_n par rapport à S_n).

Pour tout entier n, Γ_n sera défini par :

$$\Gamma_n = \frac{T_n - T_{p(n)}}{S_n - S_{p(n)}} \quad \text{si } S_n \neq S_{p(n)} \text{ (sinon } \Gamma_n = 1)$$

où p(n) est un entier, appelé prédécesseur de n, vérifiant l'inégalité $-1 \leq p(n) < n$, et choisi de telle sorte que asymptotiquement p(n) soit le plus grand entier $j \geq 0$, tel que d'une part S_j n'appartienne pas au segment $]T_j, \theta_j[$ et d'autre part S_j et S_n soient d'un même côté par rapport à S^*.

Algorithme de calcul de p(n), pour un entier n donné :

| **Si** $R_n(T_n-S_n) > 0$
| **alors** $p(n) \leftarrow \max(\{0 \leq j < n \ / \ (T_n - S_j)R_n \leq 0\} \cup \{-1\})$
|
| **Si** $R_n(T_n-S_n) < 0$
| **alors** $p(n) \leftarrow \max(\{0 \leq j < n \ / \ (\theta_n - S_j)R_n \leq 0\} \cup \{-1\})$
|
| **Si** $R_n(T_n-S_n) = 0$
| **alors** $p(n) \leftarrow -1$

DEFINITION :
| p(n) sera appelé le **prédecesseur** de n.
| $S_{p(n)}$ sera appelé le **prédecesseur** de S_n.

Donnons des schémas illustrant la construction de p(n). Sur ces schémas, volontairement nous ne faisons pas figurer la limite S^* de la suite (S_n), mais nous indiquons par une flèche la position supposée de S^* par rapport à S_n, fournie par R_n (cette position sera exacte à partir d'un certain rang). Nous noterons :

$U_n = \min(S_0, \ldots, S_n)$
$V_n = \max(S_0, \ldots, S_n)$

Illustration de la construction de la suite $(S_{p(n)})$:

a) lorsque $R_n(T_n-S_n) > 0$:

- Si $R_n = +1$, p(n) est égal au plus grand entier j, s'il existe, tel que $0 \leq j < n$ et $T_n \leq S_j$, sinon $p(n) = -1$. Donc pour que p(n) soit différent de -1, il faut et il suffit que $T_n \leq V_n$.

```
---------------------⊥----------⊥-------⊥------------⊥-->
                  <- Sn         Tn      Sp(n)        Vn
```

figure 1.

- Si R_n = -1, p(n) est égal au plus grand entier j, s'il existe, tel que $0 \le j < n$ et $T_n \ge S_j$, sinon p(n) = -1. Donc pour que p(n) soit différent de -1, il faut et il suffit que $U_n \le T_n$.

```
--⊥----⊥-----⊥----------⊥------------------------------------>
  Un   Sp(n) Tn         Sn  ->
```

figure 2.

b) lorsque $R_n(T_n-S_n) < 0$:

- Si R_n = +1, p(n) est égal au plus grand entier j, s'il existe, tel que $0 \le j < n$ et $\theta_n \le S_j$, sinon p(n) = -1. Donc pour que p(n) soit différent de -1, il faut et il suffit que $\theta_n \le V_n$.

```
-----------⊥----------⊥----------⊥-----------⊥-------⊥--->
           Tn         Sn         θn          Sp(n)   Vn
                   <-
```

figure 3.

- Si R_n = -1, p(n) est égal au plus grand entier j, s'il existe, tel que $0 \le j < n$ et $\theta_n \ge S_j$, sinon p(n) = -1. Donc pour que p(n) soit différent de -1, il faut et il suffit que $U_n \le \theta_n$.

```
--⊥----⊥------⊥---------⊥----------⊥---------------------->
  Un   Sp(n)  θn        Sn         Tn
                          ->
```

figure 4.

Donnons maintenant un lemme qui montre que p(n) prendra au plus un nombre fini de fois la valeur -1, et donc que l'on se trouvera, pour toute valeur de n suffisamment grande, dans l'un des quatre cas de figure donnés ci-dessus.

Lemme 15.

> Soit E un ensemble synchronisable par le procédé PS.
> Pour toute suite (S_n) de E, il existe un entier N'
> tel que :
> $$\forall\ n \geq N',\ 0 \leq p(n) < n$$

PREUVE

Par construction de p(n), on a $p(n) < n$ pour tout $n \geq 0$. Il suffit donc de prouver que $p(n) \geq 0$ au-delà d'un indice N'.

- Supposons que $\lim_{n\to+\infty} \tau_n = \tau > 1$

Soit $N^* \in \mathbb{N}$ tel que :

$$\forall\ n \geq N^*,\ S_n \neq S^*.$$

Alors, pour tout $n \geq N^*$, on a :

$$R_n(T_n-S_n) = (\tau_n-1)\ R_n(S_n-S^*).$$

Comme $\lim_{n\to+\infty} \tau_n-1 = \tau-1 > 0$,

et comme la réponse R_n fournie par le DAQES-algorithme est asymptotiquement égale au signe de S_n-S^*, il existe un entier N tel que, pour tout $n \geq N$, on ait :

$$\tau_n-1 > 0 \quad \text{et} \quad R_n(S_n-S^*) > 0$$

et donc :

$$R_n(T_n-S_n) > 0$$

La construction de p(n), pour $n \geq N$, se fera alors uniquement à l'aide de la suite (T_n) et pour montrer que $p(n) \geq 0$, il suffit de prouver que l'une des deux conditions suivantes est vérifiée :

$$A'(n) : U_n \leq T_n < S_n < S^*$$
$$A''(n) : S^* < S_n < T_n \leq V_n$$

Nous avons donné dans le chapitre précédent une partition de C^* en trois sous-ensembles :

$$C^* = C1^- \cup C1^+ \cup C2$$

Rappelons-en les définitions :

$$(S_n) \in C1^- \iff \exists N_1,\ \forall n \geq N_1 : S_n-S^* < 0,$$

$$(S_n) \in C1^+ \iff \exists N_1,\ \forall n \geq N_1 : S_n-S^* > 0,$$

$$(S_n) \in C2 \iff \begin{array}{c} (S_n) \in C^* \\ \text{et} \\ \forall N_1,\ \exists n > N_1 \text{ tel que } (S_n-S^*)(S_{n-1}-S^*) < 0 \end{array}$$

. Si $(S_n) \in C1^-$, pour tout $n \geq N$, on a :

$$T_n < S_n < S^*$$

Comme la suite (S_n) converge vers S^*, la suite (U_n) est stationnaire et comme (T_n) converge aussi vers S^*, il existe un entier $N' \geq N$ tel que, pour tout $n \geq N'$, on ait $A'(n)$.

. Si $(S_n) \in C1^+$, pour tout $n \geq N$, on a :

$$S^* < S_n < T_n$$

C'est alors la suite (V_n) qui est stationnaire et comme (T_n) converge vers S^*, il existe un entier $N' \geq N$ tel que, pour tout $n \geq N'$, on ait $A''(n)$.

. Si $(S_n) \in C2$, pour tout $n \geq N$, on a :

$$T_n < S_n < S^* \quad \text{si} \quad R_n = -1$$
$$S^* < S_n < T_n \quad \text{si} \quad R_n = +1$$

Dans ce cas, les deux suites (U_n) et (V_n) sont stationnaires
et comme (T_n) converge vers S^*, il existe un entier $N' \geq N$ tel que, pour tout $n \geq N'$, on ait : $U_n \leq T_n \leq V_n$.
On a donc :

soit $R_n = -1$ et $A'(n)$,
soit $R_n = +1$ et $A''(n)$.

- Supposons que $\lim\limits_{n\to+\infty} \tau_n = \tau < 1$

Comme $\lim\limits_{n\to+\infty} \tau_n-1 = \tau-1 > 0$,
il existe un entier N tel que, pour tout $n \geq N$, on ait :

$$\tau_n-1 < 0 \quad \text{et} \quad R_n(S_n-S^*) > 0$$

et donc :

$$R_n(T_n-S_n) < 0$$

La construction de $p(n)$, pour $n \geq N$, se fera alors uniquement à l'aide de la suite (θ_n) et pour montrer que $p(n) \geq 0$, il suffit de prouver que l'une des deux conditions suivantes est vérifiée :

$B'(n)$: $U_n \leq \theta_n < S_n < S^*$
$B''(n)$: $S^* < S_n < \theta_n \leq V_n$

On est ramené au cas précédent en remplaçant T_n par θ_n.■

remarque 1 :
Nous avons vu dans la preuve précédente que :
- ou bien $\tau > 1$, et alors pour tout $n \geq N'$, on a $R_n(T_n-S_n) > 0$ si bien que la construction de $p(n)$ (pour $n \geq N'$) se fait uniquement à l'aide du procédé de synchronisation PS (cf. figure 1 ou 2).
- ou bien $\tau < 1$, et alors pour tout $n \geq N'$, on a $R_n(T_n-S_n) < 0$ si bien que la construction de $p(n)$ (pour $n \geq N'$) se fait uniquement à l'aide du procédé de synchronisation PS_{-1} (cf. figure 3 ou 4).
Or, la suite $(\tau_{-1}(n))$ des rapports de synchronisation de (S_n) par PS_{-1} vérifie :

. $\lim\limits_{n \to +\infty} \tau_{-1}(n) = 2-\tau > 1$

. pour tout $n \geq N'$: $R_n(\theta_n - S_n) > 0$.

On peut donc considérer que l'on s'est ramené, moyennant le choix du procédé de synchronisation, à un cas "standard", pour lequel la limite τ de la suite des rapports de synchronisation vérifie : $\tau > 1$.

remarque 2 :
La suite $(S_{p(n)})$ n'est pas forcément une suite extraite de la suite (S_n), comme le montre l'exemple suivant.

EXEMPLE :

Soient (S_n) et (T_n) deux suites négatives de limite nulle telles que (T_n) est synchrone avec (S_n) et pour tout entier $k \geq 0$:

$T_{3k+2} < S_{3k+2} < T_{3k} < S_{3k} < T_{3k+1} < S_{3k+1} < T_{3k+5} < S_{3k+5}$

T_{3k-1} T_{3k-3} T_{3k-2} T_{3k+2} T_{3k} T_{3k+1}

```
-┴---┴--┴---┴---┴----┴---┴----┴----┴----┴--┴----┴----┴-->
```

S_{3k-1} S_{3k-3} S_{3k-2} S_{3k+2} S_{3k} S_{3k+1} 0

On a donc, pour tout $n \geq 0$, $T_n < S_n$ et on aura alors, pour tout entier n suffisamment grand, $R_n = -1$. Donc p(n) sera défini par : p(n) est le plus grand entier $j < n$ tel que $S_j \leq T_n$.

On a donc, pour tout $k \geq 0$:

$p(3k) = 3k-1$

$p(3k+1) = 3k$

$p(3k+2) = 3k-1$

et la suite (p(n)) n'est pas croissante.■

Lemme 16.

Soit E un ensemble synchronisable par le procédé PS.
Pour toute suite (S_n) de E, nous avons :

(a) $\lim\limits_{n \to +\infty} S_{p(n)} = S^*$

(b) $\lim\limits_{n \to +\infty} T_{p(n)} = S^*$

(c) $\lim\limits_{n \to +\infty} \tau_{p(n)} = \tau$

(d) Il existe un réel B et un entier $\tilde{N}$ tel que :

$$\forall\, n \geq \tilde{N},\quad 0 < \frac{S_n - S^*}{S_{p(n)} - S^*} < B < 1$$

PREUVE

- Supposons que : $\lim\limits_{n \to +\infty} \tau_n = \tau > 1$

La construction de $p(n)$ se fera en utilisant le procédé de synchronisation PS qui transforme (S_n) en (T_n).

Les suites (S_n) et (T_n) convergent vers S^* et la suite (τ_n) converge vers τ, ce qui se traduit par :
pour tout réel $\alpha > 0$, il existe un entier M tel que, pour tout $n \geq M$, on ait :

$|S_n - S^*| < \alpha$
et $|T_n - S^*| < \alpha$
et $|\tau_n - \tau| < \alpha$.

. Si $(S_n) \in C2$, on peut toujours supposer que :
$S_M < S^* < S_{M+1}$
Alors, comme (T_n) converge vers S^*, il existe $M' \geq M$ tel que pour tout $n \geq M'$, on ait :
$S_M \leq T_n \leq S_{M+1}$.

. Si $(S_n) \in C1^-$, on peut toujours supposer que :
$S_M < S^*$
Alors, comme (T_n) est aussi une suite de $C1^-$ qui converge vers S^*, il existe $M' \geq M$ tel que pour tout $n \geq M'$, on ait :
$S_M \leq T_n < S^*$

. Si $(S_n) \in C1^+$, on peut toujours supposer que :
$S^* < S_M$
De même que précédemment, il existe $M' \geq M$ tel que pour tout $n \geq M'$, on ait :
$S^* < T_n \leq S_M$

Il s'ensuit que, dans tous les cas et pour tout $n \geq M'$, on a : $p(n) \geq M$
et donc :

$|S_{p(n)} - S^*| < \alpha$
et $|T_{p(n)} - S^*| < \alpha$
et $|\tau_{p(n)} - \tau| < \alpha$.

Ce qui prouve (a), (b) et (c).

Soit un réel τ' tel que : $1 < \tau' < \tau$.
Il existe un entier $\tilde{N}$ tel que, pour tout $n \geq \tilde{N}$, on ait d'une part : $1 < \tau' < \tau_n$

d'autre part :
. ou bien $R_n = -1$ et $S_{p(n)} \leq T_n < S_n < S^*$

--⊥-----⊥----⊥-----------⊥-------------⊥--------------------->

U_n $S_{p(n)}$ T_n S_n S^*

. ou bien $R_n = +1$ et $S^* < S_n < T_n \leq S_{p(n)}$

```
-----------┴-----------┴----------┴-------┴------------┴-->
           S*          Sn         Tn      Sp(n)        Vn
```

On a donc, pour tout $n \geq \tilde{N}$:

$$0 < \frac{S_n - S^*}{S_{p(n)} - S^*} < \frac{S_n - S^*}{T_n - S^*} = \frac{1}{\tau_n} < \frac{1}{\tau'} < 1$$

ce qui prouve (d) (on pose ici $B = 1/\tau'$).

- Supposons maintenant que : $\lim_{n \to +\infty} \tau_n = \tau < 1$,

La construction de $p(n)$ se fera en utilisant le procédé de synchronisation PS_{-1} qui transforme (S_n) en (θ_n).
La suite $(\tau_{-1}(n))$ des rapports de synchronisation de (S_n) par le procédé PS_{-1} vérifie :
$\lim_{n \to +\infty} \tau_{-1}(n) = 2-\tau > 1$.
La démonstration est alors identique à la précédente, il suffit de remplacer T_n par θ_n, τ_n par $\tau_{-1}(n)$ et τ par $2-\tau$.■

Afin que Γ_n soit défini pour tout entier n, nous posons (de façon arbitraire) $S_{-1} = 0$ et $T_{-1} = 0$.

<u>Proposition 17.</u>

Soit E un ensemble synchronisable par le procédé PS.
Pour toute suite (S_n) de E, la suite (Γ_n) définie par :

$$\Gamma_n = \frac{T_n - T_{p(n)}}{S_n - S_{p(n)}} \quad \text{si } S_n \neq S_{p(n)} \text{ (sinon } \Gamma_n = 1\text{)}$$

vérifie :

$$\lim_{n \to +\infty} \Gamma_n = \tau$$

PREUVE

il existe un entier N_0 tel que :

$$\forall\ n \geq N_0,\ T_n = \frac{T_n - T_{p(n)}}{S_n - S_{p(n)}}$$

En effet, si $\tau > 1$, à partir d'un certain rang , on a :
ou bien $T_{p(n)} < S_{p(n)} \leq T_n < S_n < S^*$
ou bien $S^* < S_n < T_n \leq S_{p(n)} < T_{p(n)}$
et si $\tau < 1$, à partir d'un certain rang , on a :
ou bien $\theta_{p(n)} < S_{p(n)} \leq \theta_n < S_n < S^*$
ou bien $S^* < S_n < \theta_n \leq S_{p(n)} < \theta_{p(n)}$
et donc $S_n \neq S_{p(n)}$.

Il s'ensuit que, pour tout $n \geq N_0$, on a :

$$\Gamma_n = \frac{T_n - S^* - (T_{p(n)} - S^*)}{S_n - S^* - (S_{p(n)} - S^*)}$$

Posons :

$$z_n = \frac{S_n - S^*}{S_{p(n)} - S^*}$$

En divisant dans l'expression de Γ_n numérateur et dénominateur par $S_{p(n)}-S^*$, on obtient :

$$\Gamma_n = \frac{\tau_n\ z_n - \tau_{p(n)}}{z_n - 1} = \tau_n + \frac{\tau_n - \tau_{p(n)}}{z_n - 1}$$

D'après le lemme précédent, $\dfrac{1}{z_n-1}$ est borné

et $\tau_n - \tau_{p(n)}$ converge vers 0.

Par suite, on a bien :

$$\lim_{n \to +\infty} \Gamma_n = \lim_{n \to +\infty} \tau_n = \tau. \blacksquare$$

D'après les résultats précédents (théorème 13, propositions 16 et 17), nous pouvons maintenant décrire le ACCES-algorithme et énoncer :

<u>Théorème 14.</u>

|| Pour qu'un sous-ensemble de C^* soit accélérable, il faut et il suffit qu'il soit synchronisable.

Description du ACCES-algorithme :

```
| Répéter à l'infini
|
|    calcul de Tn {par le procédé PS}
|
|    calcul de Rn {cf. le DAQES-algorithme}
|
|    calcul de p(n)
|
|    calcul de Γn
|
|    calcul de An
|
|    n <- n+1
|
| fin du répéter
```

Il faut signaler ici le cas où E est égal à l'ensemble LIN des suites à convergence linéaire. Pour toute suite (S_n)

de **LIN** de limite S^*, il existe un réel K, avec $|K| \le 1$ et $K(K-1) \neq 0$ tel que :

$$\lim_{n \to +\infty} \frac{S_{n+1}-S^*}{S_n-S^*} = K.$$

Il est clair que **LIN** est synchronisable par le procédé de synchronisation défini par : $\forall\, n > 0,\ T_n = S_{n-1}$.

Ainsi, pour tout $n > 0$, le prédécesseur de n, $p(n)$, est égal à $n-1$ et le ACCES-algorithme donne

$$A_n = S_n - \frac{S_{n-1}-S_n}{\dfrac{S_{n-1}-S_{n-2}}{S_n-S_{n-1}} - 1} = \frac{S_n\, S_{n-2} - (S_{n-1})^2}{S_n + S_{n-2} - 2\, S_{n-1}}$$

On retrouve donc le δ^2 d'Aitken !

2. FAMILLE DE PROCEDES D'ACCELERATION

Etant donné un ensemble **E** synchronisable par un procédé PS, le procédé d'accélération de **E** que nous avons construit - le ACCES-algorithme - dépend du procédé PS. Rappelons que, pour toute suite (S_n) de **E**, la suite (A_n) construite par le ACCES-algorithme converge plus vite que (S_n), et qu'elle est donnée (sauf peut-être pour un nombre fini de termes) par la formule :

$$A_n = S_n - \frac{T_n-S_n}{\Gamma_n-1}$$

où (T_n) est la suite transformée de (S_n) par PS

et $\Gamma_n = \dfrac{T_n - T_{p(n)}}{S_n - S_{p(n)}}$

et $S_{p(n)}$ est le prédécesseur de S_n.

Or nous avons remarqué au paragraphe B.2 que, à partir d'un procédé de synchronisation PS, on pouvait définir une famille F(PS) de procédés de synchronisation notés PS_a, où a est un réel non nul quelconque. Comme tout procédé de synchronisation nous permet de définir un procédé d'accélération, il existe donc une famille de procédés d'accélération associée à la famille F(PS). Notons $(A_a(n))$ la suite convergeant plus vite que (S_n), obtenue avec le procédé PS_a, elle est définie par :

$$A_a(n) = S_n - \frac{\theta_a(n)-S_n}{\Gamma_a(n)-1}$$

où $(\theta_a(n))$ est la suite transformée de (S_n) par PS_a

$$\text{et } \Gamma_a(n) = \frac{\theta_a(n) - \theta_a(pa(n))}{S_n - S_{pa(n)}}$$

et $S_{pa(n)}$ est le prédécesseur de S_n obtenu à partir de PS_a.

Dans ce qui précède, on accélère toutes les suites d'un ensemble E synchronisable avec un même procédé d'accélération. Mais on pourrait aussi, pour toute suite de E, choisir un procédé d'accélération particulier en fonction de certains critères d'"efficacité". C'est ce que nous faisons dans le paragraphe suivant en nous intéressant à l'écart entre n et son prédécesseur p(n).

3. CHOIX DU PROCEDE D'ACCELERATION

Etant donnés un ensemble E synchronisable et une suite (S_n) de E, la construction de la suite $(S_{p(n)})$ des prédécesseurs de S_n dépend du procédé de synchronisation choisi. Nous allons donner ici une condition nécessaire et suffisante sur l'ensemble E pour que, pour toute suite (S_n) de E, on puisse choisir un procédé de synchronisation tel que l'écart entre n et son prédécesseur p(n) (calculé à partir de ce procédé) soit majoré par un entier q indépendamment de n. Ainsi la recherche du prédécesseur p(n) se fera uniquement parmi les termes $S_{n-1}, \ldots, S_{n-q}$.

Commençons par donner deux définitions.

DEFINITION :

| Soit E un sous-ensemble de C^* synchronisable et PS un procédé de synchronisation de E.
| Etant donnée une suite (S_n) de E, la suite des prédécesseurs pour PS est à délai borné q si et seulement si
| $$\exists N, \forall n \geq N : \; n - p(n) \leq q.$$

DEFINITION :

| Soit E un sous-ensemble de C^* synchronisable.
| E est à délai borné q si, et seulement si, pour toute suite (S_n) de E, il existe un procédé de synchronisation PS_a de E, dans la famille F(PS), tel que la suite des prédécesseurs pour PS_a soit à délai borné q.

Le problème initial se formule alors de la façon suivante : donner une condition nécessaire et suffisante pour qu'un ensemble E synchronisable soit à délai borné q.

Nous utiliserons la notation :

$$M_{n-1,q} = \min (S_{n-1},\ldots,S_{n-q}) \quad \text{si } S_n < S^*,$$
$$= \max (S_{n-1},\ldots,S_{n-q}) \quad \text{si } S_n > S^*.$$

```
-----┴-------------┴--------┴------------------------------->
   M(n-1,q)          S(n)     S*

--------------------┴--------┴-----------------┴------------>
                    S*       S(n)              M(n-1,q)
```

Et nous considèrerons le sous-ensemble q.BORN de C^* défini par :

$$(S_n) \in q.\text{BORN} \iff \exists A > 0,\ \exists N,\ \forall n \geq N : \frac{M_{n-1,q}-S^*}{S_n-S^*} > 1+A$$

Remarquons que q.BORN est un sous-ensemble de l'ensemble q.REG des suites q-régressives.

<u>Proposition 18.</u>

> Soit E un sous-ensemble de C^* synchronisable.
> E est à délai borné q si et seulement si E est inclus dans q.BORN.

PREUVE

Supposons que E soit à délai borné q.
Soient (S_n) une suite de E, et PS un procédé de synchronisation de E pour lequel la suite $(S_{p(n)})$ est à délai borné q.

- Etudions le cas où la suite (T_n) transformée par le procédé PS vérifie :

$$\lim_{n\to+\infty} \tau_n = \tau > 1$$

Alors il existe un réel τ' tel que $1 < \tau' < \tau$ et il existe un entier N tel que, pour tout $n \geq N$, on ait :

a) $n - p(n) \leq q$ (car E est à délai borné q)

b) $S_{p(n)} \leq T_n < S_n < S^*$ ou $S^* < S_n < T_n \leq S_{p(n)}$
(par construction du prédécesseur $S_{p(n)}$ de S_n - voir paragraphe C.2)

c) $1 < \tau' < \tau_n$ (car $\lim_{n\to+\infty} \tau_n = \tau > 1$)

De a) et b) il résulte que :

$$M_{n-1,q} \leq S_{p(n)} < S^* \quad \text{ou} \quad M_{n-1,q} \geq S_{p(n)} > S^*$$

D'où les deux cas de figures ci-dessous :

------⊥---------⊥-------------⊥----------⊥---------⊥------>
$M_{n-1,q}$ $S_{p(n)}$ T_n S_n S^*

------⊥---------⊥-------------⊥----------⊥---------⊥------>
S^* S_n T_n $S_{p(n)}$ $M_{n-1,q}$

On a donc :

$$\frac{M_{n-1,q}-S^*}{S_n-S^*} \geq \frac{T_n-S^*}{S_n-S^*} = \tau_n > \tau' > 1$$

En posant $A = \tau'-1$, cela montre que $(S_n) \in$ q.BORN.

- Lorsque $\lim_{n\to+\infty} \tau_n = \tau < 1$, la démonstration est semblable, il suffit de remplacer T_n par son symétrique θ_n par rapport à S_n.

Réciproquement, soit (S_n) une suite de q.BORN.

La condition $\dfrac{M_{n-1,q}-S^*}{S_n-S^*} > 1$ entraîne :

ou bien $M_{n-1,q} < S_n < S^*$

ou bien $M_{n-1,q} > S_n > S^*$

Pour prouver que E est à délai borné q, il suffit alors de prouver que l'on peut choisir un procédé de synchronisation PS_a transformant (S_n) en $(\theta_a(n))$ tel que, pour tout n suffisamment grand, on ait :

$\theta_a(n) \in \langle M_{n-1,q}, S_n \langle$

------⊥---------------------------⊥----------⊥---------⊥------>
$M_{n-1,q}$ $\theta_a(n)$ S_n S^*

------⊥---------⊥-------------⊥--------------------⊥------>
S^* S_n $\theta_a(n)$ $M_{n-1,q}$

Or $\theta_a(n) \in \langle M_{n-1,q}, S_n \langle$ équivaut à :

$$\frac{M_{n-1,q}-S^*}{S_n-S^*} \geq \tau_a(n) > 1 \quad \text{où } \tau_a(n) = \frac{\theta_a(n)-S^*}{S_n-S^*}$$

et on a vu que : $\tau_a(n) = 1 + a\ (\tau_n-1)$.

Donc il suffit de choisir a tel que, pour tout entier n suffisamment grand, on ait :

$A \geq a\ (\tau_n-1) > 0.$

Comme τ_n-1 est asymptotiquement du signe de $\tau-1$, nous aurons nécessairement :

$a\ (\tau-1) > 0.$

Et comme τ_n-1 a pour limite $\tau-1$, il existe un réel τ'' avec $\tau \in \rangle 1, \tau'' \langle$ tel que pour tout n suffisamment grand on ait :

$|\tau_n-1| < |\tau''-1|.$

Par suite, il suffira de choisir un réel a vérifiant les deux conditions :

(C_1) $a\ (\tau-1) > 0$

(C_2) $|a| < \dfrac{A}{|\tau''-1|}$ ■

Donnons alors un algorithme qui permet, pour une suite (S_n) donnée d'un sous-ensemble E synchronisable de q.BORN, de construire asymptotiquement un procédé PS_a pour lequel la suite des prédécesseurs est à délai borné q.

D'après le lemme 11, pour toute suite (S_n) de E, le signe de $R_n\ (T_n-S_n)$ est asymptotiquement constant, égal au signe de τ-1. Cela nous fournit donc un moyen pour choisir le signe du réel a. D'autre part, si (b_k) désigne une suite arbitraire décroissante de limite nulle (par exemple $b_k = 1/2^k$), en posant $a_k = \pm\ b_k$ (suivant le signe de $R_n(T_n-S_n)$), nous savons qu'il existe un entier k tel que le réel a_k vérifie les deux conditions (C_1) et (C_2) énoncées ci-dessus si bien que la suite des prédécesseurs est à délai borné q pour PS_{a_k} (remarquons que, si le réel a_k convient, il en est de même du réel a_{k+1}).

Par ailleurs, pour savoir si un procédé PS_{a_k} convient, nous devons, pour tout entier n, rechercher le prédécesseur de S_n. Cette recherche se fera uniquement parmi les q termes précédant S_n : $S_{n-1}, \ldots, S_{n-q}$, et si aucun de ces termes ne convient nous choisirons S_{-1} comme prédécesseur de S_n (voir paragraphe C.2). Nous utiliserons alors un booléen procédé_délai_borné qui prendra la valeur faux lorsque le procédé PS_{a_k} ne conviendra pas, c'est-à-dire lorsque nous n'aurons pas trouvé de prédécesseur de S_n parmi les q termes $S_{n-1}, \ldots, S_{n-q}$, auquel cas on incrémentera la valeur de k.

algorithme de choix du procédé PS_{a_k} et de calcul de $pa_k(n)$, lorsque E est à délai borné q :

```
| k <- 0; n <- q;
| procédé_délai_borné <- vrai;
|
| Répéter à l'infini
|
|    Si Rn (Tn-Sn) > 0 et procédé_délai_borné alors
|        pak(n) <- max ({n-q≤j<n / (θak(n)-Sj)Rn ≤ 0} U {-1});
|        procédé_délai_borné <- (p(n) ≠ -1);
|        finsi;
|
|    Si Rn (Tn-Sn) < 0 et procédé_délai_borné alors
|       pak(n) <- max ({n-q≤j<n / (θ-ak(n)-Sj)Rn ≤ 0} U {-1});
|       procédé_délai_borné <- (p(n) ≠ -1);
|       finsi;
|
|    Si Rn (Tn-Sn) = 0 et procédé_délai_borné alors
|           pak(n) <- -1;
|           finsi;
|
|    Si  non procédé_délai_borné  alors
|           k <- k+1;
|           procédé_délai_borné <- vrai;
|           finsi;
|
|    n <- n+1;
|
| Fin du répéter.
```

Nous envisageons maintenant deux cas particuliers, lorsque q = 1 ou q = 2. Nous noterons APO-1 l'ensemble des suites (S_n) convergentes de limite S^* qui vérifie la propriété suivante (introduite dans [1]) : il existe un entier N et deux réels a et b, avec $a < 1 < b$, tels que :

$$\forall n \geq N, \frac{S_{n+1}-S^*}{S_n-S^*} \notin [a,b].$$

Corollaire 6.

Soit E un sous-ensemble de C^* synchronisable.
Alors les conditions suivantes sont équivalentes :
(i) E est à délai borné 1,
(ii) E est inclus dans MON ∩ APO-1

PREUVE

Il suffit de prouver que 1.BORN = MON ∩ APO-1

Pour toute suite (S_n), lorsque q = 1, nous avons :

$M_{n-1,1} = S_{n-1}$

Par conséquent :

$$(S_n) \in 1.\text{BORN} \Leftrightarrow \exists A > 0, \exists N, \forall n \geq N : \frac{S_{n-1}-S^*}{S_n-S^*} > 1+A > 1$$

Et il est immédiat que cela équivaut à :

$(S_n) \in$ MON ∩ APO-1. ■

Corollaire 7.

Soit E un sous-ensemble de C2 synchronisable.
Alors les conditions suivantes sont équivalentes :
(i) E est à délai borné 2,
(ii) E est inclus dans OSC et pour toute suite (S_n) de E, les deux sous-suites (S_{2p}) et (S_{2p+1}) appartiennent à APO-1.

PREUVE

Soit E ⊂ C2, à délai borné 2.

Comme : 2.BORN ⊂ 2.REG

On a : E ⊂ 2.REG ∩ C2 = 2.REG2 = OSC

(cf. proposition 10 du chapitre 1).

Et, pour tout n suffisamment grand, on a alors :

$M_{n-1,2} = S_{n-2}$

ce qui prouve que les deux sous-suites (S_{2p}) et (S_{2p+1}) appartiennent à APO-1.

La réciproque est immédiate.■

remarque :

La condition (ii) du corollaire 7 n'est pas équivalente à E ⊂ OSC ∩ APO-1, comme le montre l'exemple suivant.

EXEMPLE :

Soit (S_n) la suite définie, pour tout entier $p \geq 0$, par :

$S_{4p} = S_{4p+2} = 1/2^p$

$S_{4p+1} = S_{4p+3} = - 1/2^p$

- $(S_n) \in$ OSC et donc $(S_n) \in$ APO-1,
- mais (S_n) ne vérifie pas les conditions (ii) du corollaire 7, puisque pour tout entier $p \geq 0$, on a :

$S_{4p+2} / S_{4p} = 1$

$S_{4p+3} / S_{4p+1} = 1$. ■

Enfin nous proposons une généralisation de la proposition 18. Dans ce qui précède, nous cherchions des ensembles E pour lesquels un entier q était valable pour toutes les suites de E. Nous cherchons maintenant à caractériser les ensembles synchronisables tels que, pour chaque suite, il existe un entier q "convenable". Pour cela, nous utiliserons les deux définitions suivantes.

DEFINITION :

Soit E un sous-ensemble de C^* synchronisable.
E est à délai borné si, et seulement si, pour toute suite (S_n) de E, il existe :
- un entier q et
- un procédé de synchronisation PS_a

tels que la suite des prédécesseurs pour PS_a soit à délai borné q.

DEFINITION :

$$*.BORN = \bigcup_{q\geq 1} q.BORN$$

<u>Proposition 19.</u>

Soit E un sous-ensemble de C^* synchronisable.
E est à délai borné si et seulement si E est inclus dans *.BORN.

L'algorithme que l'on propose dans ce cas diffère du précédent uniquement par la valeur de q non fixée au départ. La variable q sera initialisée à 1 et sera incrémentée de 1 en même temps que k chaque fois que le booléen procédé_délai_borné prendra la valeur faux. Nous ne réécrirons pas l'algorithme plus en détail.

Notons que la propriété d'appartenance aux ensembles q.BORN ou *.BORN, qui permet de choisir "au mieux" un procédé d'accélération pour toute suite d'un ensemble synchronisable par un procédé PS, ne fait pas intervenir de procédé de synchronisation.

A la fin de ce travail, une question se pose : un ensemble synchronisable peut-il être accéléré par un procédé distinct du ACCES-algorithme et de tous les procédés associés au ACCES-algorithme ?

D. EXEMPLES DE SYNCHRONISATION

Pour terminer, nous présentons une famille de suites pour laquelle un procédé de synchronisation se présente, nous semble-t-il, de façon plus naturelle qu'un procédé d'accélération.

Il arrive fréquemment que l'on se donne des suites numériques par une relation de récurrence entre deux termes consécutifs de la suite. Nous étudions dans cette dernière partie la possibilité de synchroniser, et donc d'accélèrer de telles suites.

Si (S_n) désigne une suite convergente définie par la donnée de son premier terme S_0 et par la relation de récurrence $S_{n+1} = f(S_n)$, sa limite S^* vérifie alors (sous l'hypothèse que f est continue) la relation $S^* = f(S^*)$, et nous avons :

$$e_{n+1} = S_{n+1}-S^* = f(S^*+e_n) - f(S^*)$$

Ce qui, en supposant maintenant que f est suffisamment dérivable dans un voisinage de S^*, fournit entre les erreurs e_n et e_{n+1} une relation du type :

$$\text{(R)} \qquad e_{n+1} = K\, e_n - \alpha_n\, e_n{}^k$$

avec $K = f'(S^*)$,

$k \geq 2$,

et $\lim\limits_{n\to+\infty} \alpha_n = \alpha = - f^{(k)}(S^*)/k! \neq 0$.

($f^{(k)}(S^*)$ est la première dérivée d'ordre ≥ 2 non nulle en S^*).

Comme le rapport e_{n+1}/e_n converge vers le réel K, K vérifie nécessairement : $|K| \leq 1$ (sinon la suite (S_n) ne pourrait converger). Nous supposerons ici $K \neq 0$: le cas $K = 0$ correspondant aux suites à convergence superlinéaire n'est pas intéressant ici puisque ce sont des suites non accélérables [2] (et donc non synchronisables).

Ainsi les suites convergentes vérifiant une relation du type (R) avec $K \neq 0$ sont :
- soit à convergence logarithmique (lorsque $K = 1$),
- soit à convergence linéaire (lorsque $K \neq 1$).

Nous savons déjà que les suites à convergence linéaire sont synchronisables par le procédé de synchronisation défini par : $T_n = S_{n-1}$, et accélérables par le δ^2 d'Aitken. Il nous reste à étudier l'ensemble des suites à convergence logarithmique vérifiant une relation du type (R) (ce que nous faisons dans le paragraphe 1). Et dans le paragraphe 2, nous nous intéresserons à la synchronisation de la réunion des deux ensembles précédents (notons que l'union de deux ensembles synchronisables n'est pas nécessairement synchronisable).

1. ETUDE DES SUITES CONVERGENTES DU TYPE : $e_{n+1} = e_n - \alpha_n e_n^k$

Nous allons montrer que toute suite convergente à convergence logarithmique vérifiant une relation du type (R) : $e_{n+1} = e_n - \alpha_n e_n^k$ est synchronisable par le procédé de synchronisation défini par : $T_n = S_{[n/2]}$, où $[n/2]$ désigne la partie entière du réel $n/2$.

Remarquons tout d'abord que, pour une suite (S_n) convergente vérifiant une relation du type $e_{n+1} = e_n - \alpha_n e_n^k$, certaines conditions doivent être satisfaites (sinon il ne pourrait y avoir convergence), à savoir :

1) Dans le cas $\alpha > 0$ et k pair, on a nécessairement $e_0 > 0$.

2) Dans le cas $\alpha < 0$ et k pair, on a nécessairement $e_0 < 0$ et on peut se ramener au cas 1) en considérant la suite (X_n) définie par $X_n = -S_n$ de limite $X^* = -S^*$. En effet, on a :

$X_0 - X^* > 0$

et $X_{n+1} = X_n - \beta_n X_n^k$,

avec $\beta_n = -\alpha_n$, de limite $\beta = -\alpha > 0$.

3) Dans le cas $\alpha < 0$ et k impair, le seul cas de convergence est celui de la suite constante égale à S^* !

4) Dans le cas $\alpha > 0$ et k impair, si $e_0 < 0$, en considérant la suite (X_n) définie par : $X_n = - S_n$ de limite $X^* = -S^*$, on se ramène au cas $X_0 - X^* > 0$.

Si bien que pour chaque résultat nous pourrons nous contenter de faire la démonstration lorsque (S_n) vérifie :

$$e_{n+1} = e_n - \alpha_n e_n^k$$
$$\text{avec } \lim_{n \to +\infty} \alpha_n = \alpha > 0 \quad \text{et} \quad S_0 > S^*.$$

Notons aussi que, dans tous les cas de convergence, les suites sont monotones, décroissantes si $S_0 > S^*$, croissantes si $S_0 < S^*$.

Nous pouvons énoncer :

Lemme 17.

Soit (S_n) une suite convergente de limite S^*, dont les erreurs $e_n = S_n - S^*$ vérifient :

$$e_{n+1} = e_n - \alpha_n e_n^k, \text{ avec } \lim_{n\to+\infty} \alpha_n = \alpha \neq 0.$$

Alors on peut écrire e_n sous la forme :

$$e_n = \delta_n n^{-h}, \text{ avec } h = (k-1)^{-1} \text{ et } \lim_{n\to+\infty} \delta_n = \delta \neq 0.$$

PREUVE

D'après ce qui précéde, nous ferons la démonstration uniquement dans le cas : $\alpha > 0$ et $S_0 > S^*$.
Considérons la suite (y_n) définie pour tout $n > 0$ par :

$$y_n = n^{-h} \text{ avec } h = (k-1)^{-1}.$$

Posons $\delta_n = e_n/y_n$, pour tout $n > 0$.
Il s'agit de montrer que la suite (δ_n) admet une limite non nulle δ.
Il est facile de vérifier que y_{n+1} peut s'exprimer en fonction de y_n de la façon suivante :

$$y_{n+1} = y_n - \beta_n y_n^k, \text{ avec } \lim_{n\to+\infty} \beta_n = h.$$

Or nous avons :

Fait : Soient (x_n) et (y_n) deux suites convergentes de limite nulle vérifiant :

$$x_0 > 0, \quad y_0 > 0$$

$$x_{n+1} = x_n - \alpha_n x_n^k, \quad \text{avec } \lim_{n\to+\infty} \alpha_n = \alpha > 0$$

$$y_{n+1} = y_n - \beta_n y_n^k, \quad \text{avec } \lim_{n\to+\infty} \beta_n = \beta > 0$$

$$\text{alors } \lim_{n\to+\infty} \frac{x_n}{y_n} = (\beta/\alpha)^h \quad \text{avec } h = (k-1)^{-1}.$$

Les suites (e_n) et (y_n) vérifiant les hypothèses du fait, on obtient :

$$\lim_{n\to+\infty} \frac{e_n}{y_n} = \lim_{n\to+\infty} \delta_n = (h/\alpha)^h \neq 0.$$

D'où le résultat.■

PREUVE DU FAIT:

1) Nous commençons par faire la preuve dans le cas particulier où (x_n) et (y_n) sont deux suites convergentes de limite nulle vérifiant :

$$x_0 > 0 \quad \text{et} \quad \forall\, n \geq 0, \quad x_{n+1} = x_n - x_n^k,$$

$$y_0 > 0 \quad \text{et} \quad \forall\, n \geq 0, \quad y_{n+1} = y_n - y_n^k.$$

Il s'agit alors de montrer que :

$$\lim_{n\to+\infty} \frac{x_n}{y_n} = 1.$$

Nous savons que les suites (x_n) et (y_n) sont décroissantes de limite nulle.
Posons $A = k^{-h}$ avec $h = (k-1)^{-1}$.
Alors il existe un entier N tel que :

$$0 < x_N < A,$$

et il existe un entier m tel que :

$$0 < y_m \leq x_N.$$

Soit n le plus grand des entiers j tels que $y_m \leq x_j$.
Nous avons donc $n \geq N$ et :

$$0 < x_{n+1} < y_m \leq x_n < A.$$

Montrons alors par récurrence que, pour tout entier $j \geq 0$, on a :

$$(1) \qquad x_{n+j+1} \leq y_{m+j} \leq x_{n+j} < A.$$

- C'est vrai pour $j = 0$.
- Supposons que l'on ait :

$$x_{n+j+1} \leq y_{m+j} \leq x_{n+j} < A.$$

Alors, compte tenu des relations de récurrence entre :

- x_{n+j+1} et x_{n+j} d'une part,
- y_{m+j+1} et y_{m+j} d'autre part,

on obtient :

$$x_{n+j+1} - y_{m+j+1} = x_{n+j} - y_{m+j} - ((x_{n+j})^k - (y_{m+j})^k),$$

et le membre de droite est bien positif puisque la fonction $f(x) = x-x^k$ est croissante sur l'intervalle $[0,A]$.
De même on a :

$$y_{m+j+1} - x_{n+j+2} = y_{m+j} - x_{n+j+1} - ((y_{m+j})^k) - (x_{n+j+1})^k)$$

et donc : $y_{m+j+1} - x_{n+j+2} \geq 0$.

Divisons alors tous les membres de l'inégalité (1) par x_{n+j}

$$\frac{x_{n+j+1}}{x_{n+j}} \leq \frac{y_{m+j}}{x_{n+j}} \leq 1,$$

et faisons tendre j vers $+\infty$, on obtient alors :

$$\lim_{j\to+\infty} \frac{y_{m+j}}{x_{n+j}} = 1.$$

Et comme pour tout $n > m$ on a :

$$\frac{y_{n+j}}{x_{n+j}} = \frac{y_{n+j}}{y_{n+j-1}} \cdots\cdots \frac{y_{m+j+1}}{y_{m+j}} \frac{y_{m+j}}{x_{n+j}},$$

et que la suite (y_n) est à convergence logarithmique, on en déduit que :

$$\lim_{j\to+\infty} \frac{y_{n+j}}{x_{n+j}} = 1.$$

Soit encore :

$$\lim_{n\to+\infty} \frac{y_n}{x_n} = 1.$$

2) De ce premier cas nous pouvons alors déduire que si (x_n) et (y_n) sont deux suites convergentes de limite nulle vérifiant :

$$x_0 > 0 \quad \text{et} \quad \forall\, n \geq 0, \quad x_{n+1} = x_n - \alpha\, x_n^k \quad (\alpha > 0),$$
$$y_0 > 0 \quad \text{et} \quad \forall\, n \geq 0, \quad y_{n+1} = y_n - \beta\, y_n^k \quad (\beta > 0).$$

alors $\lim\limits_{n\to+\infty} \dfrac{x_n}{y_n} = (\beta/\alpha)^h$ avec $h = (k-1)^{-1}$.

En effet il suffit de considérer les suites (X_n) et (Y_n) définies pour tout entier n par :

$$X_n = \alpha^h\, x_n$$
$$Y_n = \beta^h\, y_n$$

Ces suites vérifient les relations de récurrence :

$$X_{n+1} = X_n - X_n^k$$
$$Y_{n+1} = Y_n - Y_n^k$$

En effet (pour la suite (Y_n), le calcul est semblable) :

$$\begin{aligned} X_{n+1} &= \alpha^h\, x_{n+1} \\ &= \alpha^h\, (x_n - \alpha\, x_n^k) \\ &= X_n - (\alpha^{(h+1)/k}\, x_n)^k \\ &= X_n - (\alpha^h\, x_n)^k \\ &= X_n - X_n^k \end{aligned}$$

D'après l'étude faite en 1), on obtient :

$$\lim_{n\to+\infty} \frac{X_n}{Y_n} = 1$$

L'égalité suivante nous donne alors immédiatement le résultat :

$$\frac{x_n}{y_n} = (\beta/\alpha)^h\, \frac{X_n}{Y_n}.$$

3) Nous pouvons maintenant démontrer le fait dans le cas général.

Posons $h = (k-1)^{-1}$ et considérons les fonctions Φ et θ définies par :

$$\Phi(\delta) = ((\beta+\delta)/(\alpha-\delta))^h,$$
$$\theta(\delta) = ((\beta-\delta)/(\alpha+\delta))^h.$$

Comme Φ et θ tendent vers $(\beta/\alpha)^h$ lorsque δ tend vers 0, pour tout réel $\sigma > 0$, il existe un réel $\delta < 1$ tel que :

(1) $\quad |\Phi(\delta) - (\beta/\alpha)^h| < \sigma/2,$

(2) $\quad |\theta(\delta) - (\beta/\alpha)^h| < \sigma/2.$

δ étant maintenant fixé, considérons les suites $({}^+x_n)$, $({}^-x_n)$, $({}^+y_n)$ et $({}^-y_n)$ définies comme suit :

$${}^+x_0 = {}^-x_0 = x_0$$
$${}^+x_{n+1} = {}^+x_n - (\alpha-\delta)\, ({}^+x_n)^k$$
$${}^-x_{n+1} = {}^-x_n - (\alpha+\delta)\, ({}^-x_n)^k$$

$${}^+y_0 = {}^-y_0 = y_0$$

$${}^{+}y_{n+1} = {}^{+}y_n - (\beta-\delta)\,({}^{+}y_n)^k$$
$${}^{-}y_{n+1} = {}^{-}y_n - (\beta+\delta)\,({}^{-}y_n)^k$$

Comme les suites (α_n) et (β_n) convergent respectivement vers α et β, il existe un entier N' tel que pour tout $n \geq N'$ on ait :

$$\alpha-\delta < \alpha_n < \alpha+\delta$$
$$\beta-\delta < \beta_n < \beta+\delta$$

Alors pour tout $n \geq N'$, on a :

$${}^{-}x_n < x_n < {}^{+}x_n$$
$${}^{-}y_n < y_n < {}^{+}y_n$$

Et donc aussi :

$$(3) \qquad \frac{{}^{-}x_n}{{}^{+}y_n} < \frac{x_n}{y_n} < \frac{{}^{+}x_n}{{}^{-}y_n}$$

D'après ce qui a été démontré au 2), on a :

$$\lim_{n\to+\infty} \frac{{}^{+}x_n}{{}^{-}y_n} = ((\beta+\delta)/(\alpha-\delta))^h = \Phi(\delta)$$

$$\lim_{n\to+\infty} \frac{{}^{-}x_n}{{}^{+}y_n} = ((\beta-\delta)/(\alpha+\delta))^h = \theta(\delta)$$

Donc il existe un entier $N'' \geq N'$ tel que pour tout $n \geq N''$ on ait :

$$\Phi(\delta) - \sigma/2 < \frac{{}^{+}x_n}{{}^{-}y_n} < \Phi(\delta) + \sigma/2$$

$$\Phi(\delta) - \sigma/2 < \frac{{}^{-}x_n}{{}^{+}y_n} < \theta(\delta) + \sigma/2$$

Et compte tenu de l'inégalité (3) et du choix de δ (inégalités (1) et (2)), on obtient finalement, pour tout $n \geq N''$:

$$(\beta/\alpha)^h - \sigma < \frac{x_n}{y_n} < (\beta/\alpha)^h + \sigma$$

ce qui prouve la convergence de la suite (x_n/y_n) vers $(\beta/\alpha)^h$.■

Dès lors nous avons :

Proposition 20.

Soit (S_n) une suite convergente de limite S^*, dont les erreurs $e_n = S_n - S^*$ vérifient :

$$e_{n+1} = e_n - \alpha_n e_n^k, \text{ avec } \lim_{n\to+\infty} \alpha_n = \alpha \neq 0$$

Soit (T_n) la suite définie, pour tout entier $n \geq 0$, par :

$$T_n = S_{[n/2]}$$

Alors on a :

(1) (T_n) est synchrone avec (S_n),

(2) $$\lim_{n\to+\infty} \frac{T_n - S^*}{S_n - S^*} = 2^h \quad \text{avec } h = (k-1)^{-1}.$$

PREUVE

Soit (S_n) une suite vérifiant les hypothèses précédentes.
D'après le lemme 17, nous avons :

$$\frac{T_n - S^*}{S_n - S^*} = \frac{e_{[n/2]}}{e_n} = \frac{\delta_{[n/2]}\,[n/2]^{-h}}{\delta_n \quad n^{-h}}$$

Comme les suites (δ_n) et $(\delta_{[n/2]})$ convergent vers $\delta \neq 0$ et que la suite $(n/[n/2])$ converge vers 2, le rapport $(T_n - S^*)/(S_n - S^*)$ converge bien vers 2^h, ce qui prouve que (T_n) est synchrone avec (S_n).■

2. ETUDE DES SUITES CONVERGENTES DU TYPE : $e_{n+1} = K\, e_n - \alpha_n e_n^k$, AVEC $|K| \leq 1$ ET $K \neq 0$.

Les suites que nous considérons maintenant sont des suites (S_n) convergentes dont les erreurs $e_n = S_n - S^*$ vérifient une relation du type (R) : $e_{n+1} = K\, e_n - \alpha_n e_n^k$, où K désigne un réel non nul tel que $|K| \leq 1$. Elles sont de deux types :

- soit à convergence linéaire ($K \neq 1$) et nous savons les synchroniser par $T_n = S_{n-1}$,

- soit à convergence logarithmique ($K = 1$) et nous savons les synchroniser par $T_n = S_{[n/2]}$.

Il s'agit ici de trouver un "moyen algorithmique" pour décider asymptotiquement à quelle catégorie appartient une suite (S_n) donnée. Pour cela nous utiliserons les deux suites (A_n) et (B_n) définies par :

$$A_n = \frac{S_n - S_{[n/2]}}{S_{[n/2]} - S_{[n/4]}}$$

$$B_n = \frac{S_n - S_{n-1}}{S_{n-1} - S_{n-2}}$$

Sur les suites (A_n) et (B_n) nous avons les résultats suivants, selon que la suite (S_n) est à convergence logarithmique (proposition 21) ou à convergence linéaire (proposition 22) :

Proposition 21.

Soit (S_n) une suite convergente de limite S^*, dont les erreurs $e_n = S_n - S^*$ vérifient :

$$e_{n+1} = e_n - \alpha_n e_n^k, \text{ avec } \lim_{n\to+\infty} \alpha_n = \alpha \neq 0$$

Alors : $\lim_{n\to+\infty} A_n = 2^{-h}$ **où** $h = (k-1)^{-1}$,

et $\lim_{n\to+\infty} B_n = 1.$

PREUVE

Par définition de A_n, nous pouvons écrire :

$$A_n = \frac{e_n - e_{[n/2]}}{e_{[n/2]} - e_{[n/4]}}$$

ou encore, d'après le lemme 17 :

$$A_n = \frac{\delta_n n^{-h} - \delta_{[n/2]} [n/2])^{-h}}{\delta_{[n/2]} [n/2]^{-h} - \delta_{[n/4]} [n/4])^{-h}}$$

$$A_n = \frac{\delta_n}{\delta_{[n/2]}} \; \frac{n^{-h}}{([n/2])^{-h}} \; \frac{1 - \dfrac{\delta_{[n/2]}}{\delta_n} \dfrac{([n/2])^{-h}}{n^{-h}}}{1 - \dfrac{\delta_{[n/4]}}{\delta_{[n/2]}} \dfrac{([n/4])^{-h}}{([n/2])^{-h}}}$$

D'après le lemme 17, les suites (δ_n), $(\delta_{[n/2]})$, et $(\delta_{[n/4]})$ convergent vers une limite non nulle δ.

D'autre part les suites $(n^{-h}/[n/2]^{-h})$ et $([n/2]^{-h}/[n/4]^{-h})$ tendent vers 2^{-h}.

Il s'ensuit que la suite (A_n) converge vers 2^{-h}.

Pour la suite (B_n), il suffit de remarquer que B_n peut s'écrire sous la forme suivante dans laquelle (e_{n-1}/e_{n-2}) converge vers 1 (suite à convergence logarithmique).

$$B_n = \frac{e_n - e_{n-1}}{e_{n-1} - e_{n-2}} = \frac{- \alpha_{n-1} (e_{n-1})^k}{- \alpha_{n-2} (e_{n-2})^k} . \blacksquare$$

<u>Proposition 22.</u>

Soit (S_n) une suite convergente de limite S^*, dont les erreurs $e_n = S_n - S^*$ vérifient :

$$e_{n+1} = K\, e_n - \alpha_n\, e_n^{k}, \text{ avec } \lim_{n \to +\infty} \alpha_n = \alpha \neq 0 \text{ et } K \neq 1.$$

Alors : $\lim_{n \to +\infty} B_n = K$,

et lorsque $0 < K < 1$, on a :

$$\lim_{n \to +\infty} A_n = 0.$$

PREUVE

Pour (B_n), c'est un résultat classique (voir par exemple [1]).

Par définition de A_n, nous pouvons écrire :

$$A_n = \frac{e_n - e_{[n/2]}}{e_{[n/2]} - e_{[n/4]}}$$

$$A_n = \frac{e_{[n/2]}}{e_n} \; \frac{\dfrac{e_n}{e_{[n/2]}} - 1}{\dfrac{e_{[n/2]}}{e_{[n/4]}} - 1}$$

Il suffit de montrer que :

$$\lim_{n \to +\infty} \frac{e_{2n}}{e_n} = 0$$

Comme : $\lim_{n \to +\infty} \dfrac{e_{n+1}}{e_n} = K > 0$,

les erreurs e_n ont toutes un même signe à partir d'un certain rang et on peut toujours supposer qu'elles sont positives, quitte à changer α_n en $-\alpha_n$.

D'autre part, comme :

$$e_{n+1} = e_n\, K\, (1 - \alpha_n K^{-1} e_n^{k-1}).$$

on peut écrire :

$$0 \le \frac{e_{2n}}{e_n} = K^n \prod_{i=n}^{2n-1} (1 - \alpha_i K^{-1} e_i^{k-1}).$$

Envisageons alors les deux cas suivants selon le signe de α.

1er cas : $\alpha > 0$

alors il existe un entier N tel que :

$$\forall\, i \ge N,\ \alpha_i > 0 \quad \text{et} \quad 1 - \alpha_i K^{-1} e_i^{k-1} < 1.$$

Il en résulte que pour tout $n \ge N$ on a :

$$0 \le \frac{e_{2n}}{e_n} < K^n$$

Ce qui prouve la convergence de (e_{2n}/e_n) vers 0.

2ème cas : $\alpha < 0$.
alors il existe un entier N tel que :
$\forall\ i \geq N,\ \alpha_i < 0$ et $1 - \alpha_i\ K^{-1}\ e_i^{k-1} > 1$.
Pour tout $n \geq N$, nous avons alors :

$$\prod_{i=n}^{2n-1} (1 - \alpha_i\ K^{-1}\ e_i^{k-1}) < \prod_{i=N}^{+\infty} (1 - \alpha_i\ K^{-1}\ e_i^{k-1}),$$

et il suffit de montrer que le produit infini converge. Nous savons qu'il converge si et seulement si la série suivante converge :

$$\sum_{i=N}^{+\infty} \ln(1 - \alpha_i\ K^{-1}\ e_i^{k-1}).$$

C'est une série à termes positifs dont le terme général est équivalent à :

$$- \alpha\ K^{-1}\ e_i^{k-1}$$

Or comme la suite (e_{n+1}/e_n) converge vers K, avec $0<K<1$, la série à termes positifs $\Sigma\ e_n$ converge et, pour tout entier $q \geq 1$, la série $\Sigma\ e_n^q$ converge aussi.
D'où le résultat.■

Posons $g(x) = \min(|x|, |1-x|)$. D'après les deux propositions précédentes, nous remarquons que :

- la propriété "B_n converge vers une limite négative" caractérise les suites à convergence linéaire avec $-1 \leq K < 0$.
- la propriété "$g(A_n)$ converge vers 0" caractérise les suites à convergence linéaire avec $0 < K < 1$,
- tandis que la propriété "$g(B_n)$ converge vers 0" caractérise les suites à convergence logarithmique.

Nous pouvons alors énoncer.

<u>Proposition 23.</u>

L'ensemble des suites (S_n) convergentes de limite S^*, dont les erreurs $e_n = S_n - S^*$ vérifient :

$$e_{n+1} = K\ e_n - \alpha_n\ e_n^k, \text{ avec } \lim_{n\to+\infty} \alpha_n = \alpha \neq 0$$

$$\text{et } |K| \leq 1 \text{ et } K \neq 0$$

est synchronisable.

description de l'algorithme de synchronisation

```
| Répéter à l'infini
|
|    calcul de Bn;
|    Si Bn < 0 alors Tn <- Sn-1
|                   {convergence linéaire avec K < 0};
|             sinon calcul de An, g(An), g(Bn);
|                   Si g(An) < g(Bn) alors Tn <- Sn-1
|                          {convergence linéaire avec K > 0}
|                                  sinon Tn <- S[n/2];
|                          {convergence logarithmique}
|             finsi;
|    n <- n+1;
|
| fin du répéter.
```

CONCLUSION

Dans ce travail, nous avons établi un résultat théorique : la condition "E est une famille de suites accélérable" est équivalente à une autre condition à priori plus facile à démontrer (car moins exigente) "E est une famille de suites synchronisable". Pour cela, nous avons construit un procédé d'accélération qui fonctionne sur tout ensemble synchronisable : le ACCES-algorithme.

Ce résultat pourrait permettre de démontrer plus facilement que certaines familles de suites sont accélérables : les familles pour lesquelles un procédé de synchronisation se présenterait de façon plus "naturelle" qu'un procédé d'accélération.

Une application immédiate du ACCES-algorithme a été de retrouver le δ^2 d'Aitken dans le cas particulier des suites à convergence linéaire, qui sont facilement synchronisables. Une question se pose alors : peut-on retrouver d'autres algorithmes classiques à partir du ACCES-algorithme, et ce, à partir de quels procédés de synchronisation ?

Les applications numériques n'ont pas été abordées ici et cette étude reste donc à envisager. Elle permettrait de comparer le ACCES-algorithme avec d'autres algorithmes classiques d'accélération de la convergence.

Chapitre I.

A. DEFINITIONS

Lemme 0.
Soit $E \subset C^*$ et A un algorithme de décidabilité sur E.
Si (U_n) et (V_n) sont deux suites de E telles que :
$\exists\, m, \forall\, n \leq m : U_n = V_n,$
alors A donne, jusqu'à l'indice m, la même réponse pour les deux suites (U_n) et (V_n).

B. DEUX ENSEMBLES SUR LESQUELS LA QUESTION Q EST ASYMPTOTIQUEMENT INDECIDABLE

1. Ensemble C1

Théorème 1.
La question "quel est le signe de S_n-S^* ?" est asymptotiquement indécidable sur l'ensemble C1.

Corollaire 1.
La question "quel est le signe de S_n-S^* ?" est asymptotiquement indécidable sur l'ensemble C^*.

2. Ensemble C2

Théorème 2.
La question "quel est le signe de S_n-S^* ?" est asymptotiquement indécidable sur l'ensemble C2.

C. ETUDE DE SOUS-ENSEMBLES DE C1

1. Décidabilité asymptotique de la question Q sur l'ensemble $C1^{\circ}$

Lemme 1.
Pour toute suite (S_n) de $C1^{\bullet}$, nous avons les équivalences suivantes :
a) $(S_n) \in C1^- \Leftrightarrow$ la suite (U_n) est stationnaire mais la suite (V_n) est non stationnaire.
b) $(S_n) \in C1^+ \Leftrightarrow$ la suite (V_n) est stationnaire mais la suite (U_n) est non stationnaire.

Théorème 3
La question "quel est le signe de S_n-S^* ?" est asymptotiquement décidable sur l'ensemble $C1^{\bullet}$.

2. Décidabilité asymptotique de la question Q sur l'ensemble MON

Proposition 1.
La question Q est asymptotiquement décidable sur l'ensemble MON.

3. Définitions des suites progressives

Proposition 2.
1.PROG = MON
$\forall$ q $\geq$ 1, q.PROG $\subset$ (q+1).PROG

4. Définitions des suites régressives

Proposition 3.
1.REG = MON
$\forall$ q $\geq$ 1, q.REG $\subset$ (q+1).REG

Proposition 4.
a) 1.PROG1 = 1.REG1 = MON
b) Pour tout entier q $\geq$ 1 : q.PROG1 $\cap$ q.REG1 $\neq \emptyset$

Proposition 5.
Pour tout entier q $\geq$ 2 :
a) q.PROG1 $\not\subset$ q.REG1
b) q.PROG1 $\not\subset$ *.REG1
c) q.REG1 $\not\subset$ q.PROG1
d) q.REG1 $\not\subset$ *.PROG1

5. Décidabilité asymptotique de la question Q sur les ensembles q.PROG1, q.REG1

Lemme 2.
Pour toute suite (S_n) convergente de limite S^*, nous avons les équivalences suivantes :
$(S_n) \in$ q.PROG1 $\cap$ $C1^-$ <=> $\exists N, \forall n \geq N : S_n \leq V_{n+q,q} < S^*$
$(S_n) \in$ q.PROG1 $\cap$ $C1^+$ <=> $\exists N, \forall n \geq N : S^* < U_{n+q,q} \leq S_n$

Lemme 3.
Toute suite (S_n) de C^* de limite S^* vérifie la propriété :
$\forall N, \exists n \geq N$ tel que : $\forall p > n, |S_p - S^*| < |S_n - S^*|$

Corollaire 2.
a) Toute suite (S_n) de $C1^-$ vérifie :
$\forall N, \exists n \geq N : S_n < \inf \{ S_p, p > n \}$.
b) Toute suite (S_n) de $C1^+$ vérifie :
$\forall N, \exists n \geq N : S_n > \sup \{ S_p, p > n \}$.

Théorème 4.
Pour tout entier q non nul, la question "quel est le signe de $S_n - S^*$?" est asymptotiquement décidable sur l'ensemble q.PROG1.

Lemme 4.

Pour toute suite (S_n) convergente, nous avons les deux équivalences suivantes :

$(S_n) \in q.REG1 \cap C1^- \iff \exists N, \forall n \geq N : U_{n-1,q} \leq S_n < S^*$

$(S_n) \in q.REG1 \cap C1^+ \iff \exists N, \forall n \geq N : S^* < S_n \leq V_{n-1,q}$

Lemme 5.

Soit q un entier quelconque non nul.

a) toute suite (S_n) de $C1^-$ vérifie :

$\forall N, \exists n \geq N : V_{n-1,q} < S_n$

b) toute suite (S_n) de $C1^+$ vérifie :

$\forall N, \exists n \geq N : S_n < U_{n-1,q}$

Théorème 5.

Pour tout entier q non nul, la question "quel est le signe de S_n-S^* ?" est asymptotiquement décidable sur l'ensemble q.REG1.

6. Décidabilité asymptotique de la question Q sur les ensembles *.PROG1 et *.REG1

Lemme 6.

a) Soit (S_n) une suite de $r.PROG1 \cap C1^-$ et $\tilde{N} \geq r$ tels que:

$\forall n \geq \tilde{N}, S_{n-r} \leq V_{n,r} < S^*$.

Alors, pour tout entier $q \geq r$ et tout entier $n \geq \tilde{N}+q-r$, on a : $S_{n-q} \leq V_{n,q} < S^*$.

b) Soit (S_n) une suite de $r.PROG1 \cap C1^+$ et $\tilde{N} \geq r$ tels que:

$\forall n \geq \tilde{N}, S^* < U_{n,r} \leq S_{n-r}$.

Alors, pour tout entier $q \geq r$ et tout entier $n \geq \tilde{N}+q-r$, on a : $S^* < U_{n,q} \leq S_{n-q}$.

Théorème 6.

La question "quel est le signe de S_n-S^* ?" est asymptotiquement décidable sur l'ensemble *.PROG1.

Lemme 7.

a) Soit (S_n) une suite de $r.REG1 \cap C1^-$ et $\tilde{N} \geq r$ tels que:

$\forall n \geq \tilde{N}, U_{n-1,r} \leq S_n < S^*$.

Alors, pour tout entier $q \geq r$ et tout entier $n \geq \tilde{N}+q-r$, on a : $U_{n-1,q} \leq S_n < S^*$.

b) Soit (S_n) une suite de $r.REG1 \cap C1^+$ et $\tilde{N} \geq r$ tels que:

$\forall n \geq \tilde{N}, S^* < S_n \leq V_{n-1,r}$.

Alors, pour tout entier $q \geq r$ et tout entier $n \geq \tilde{N}+q-r$, on a : $S^* < S_n \leq V_{n-1,q}$.

Théorème 7.

La question "quel est le signe de S_n-S^* ?" est asymptotiquement décidable sur l'ensemble *.REG1.

D. ETUDE DE SOUS-ENSEMBLES DE C2

1. Décidabilité asymptotique de la question Q sur l'ensemble ALT

Proposition 6.
La question "quel est le signe de S_n-S* ?" est asymptotiquement décidable sur l'ensemble ALT.

2. Définitions des suites q-alternées

Proposition 7.
Pour tout entier $q \in N^*$, $q.ALT^+ = q.ALT^-$.

Proposition 8.
1.ALT = ALT
$\forall\ q \geq 1$, q.ALT $\subset$ (q+1).ALT

3. Suites progressives et régressives de C2

Proposition 9.
$\forall\ q \geq 1$, q.PROG2 $\subset$ (q+1).PROG2
$\forall\ q \geq 1$, q.REG2 $\subset$ (q+1).REG2

Proposition 10.
a) 1.PROG2 = 1.REG2 = $\emptyset$
b) 2.PROG2 = 2.REG2 = OSC
c) pour tout $q \geq 2$, q.PROG2 $\cap$ q.REG2 $\neq \emptyset$

Proposition 11.
Pour tout entier $q \geq 1$, on a les inclusions :
a) q.PROG2 $\subset$ q.ALT
b) q.REG2 $\subset$ q.ALT

Proposition 12.
Pour tout entier $q \geq 3$:
a) q.PROG2 $\not\subset$ q.REG2
b) q.PROG2 $\not\subset$ *.REG2
c) q.REG2 $\not\subset$ q.PROG2
d) q.REG2 $\not\subset$ *.PROG2

4. Indécidabilité asymptotique de la question Q sur :

a) l'ensemble q.PROG2, $q \geq 3$

Théorème 8.
Pour tout entier $q \geq 3$, la question "quel est le signe de S_n-S* ?" est asymptotiquement indécidable sur l'ensemble q.PROG2.

Corollaire 3.

Pour tout entier $q \geq 3$, la question "quel est le signe de S_n-S* ?" est asymptotiquement indécidable sur l'ensemble q.PROG .

b) l'ensemble q.REG2, $q \geq 3$

Théorème 9.

Pour tout entier $q \geq 3$, la question "quel est le signe de S_n-S* ?" est asymptotiquement indécidable sur l'ensemble q.REG2.

Corollaire 4.

Pour tout entier $q \geq 3$, la question "quel est le signe de S_n-S* ?" est asymptotiquement indécidable sur l'ensemble q.REG .

c) l'ensemble q.ALT, $q \geq 2$

Théorème 10.

Pour tout entier $q \geq 2$, la question "quel est le signe de S_n-S* ?" est asymptotiquement indécidable sur l'ensemble q.ALT.

E. ETUDE DE SOUS-ENSEMBLES DE C*

1. Décidabilité asymptotique de la question Q sur l'ensemble 2.PROG

Théorème 11.

La question "quel est le signe de S_n-S* ?" est asymptotiquement décidable sur l'ensemble 2.PROG.

2. Indécidabilité asymptotique de la question Q sur l'ensemble 2.REG

Théorème 12.

La question "quel est le signe de S_n-S* ?" est asymptotiquement indécidable sur l'ensemble 2.REG.

F. TABLEAUX RECAPITULATIFS

Chapitre II.

A. INTRODUCTION

B. LA QUESTION Q "QUEL EST LE SIGNE DE S_n-S^* ?" EST ASYMPTOTIQUEMENT DECIDABLE SUR TOUT SOUS-ENSEMBLE DE C^* SYNCHRONISABLE

1. Principe de l'algorithme

Lemme 8.
Soient E un sous-ensemble de C^* synchronisable par le procédé PS.
Soit (S_n) une suite de E et $N^* \in N$ tel que :
$\forall\ n \geq N^*, \quad S_n \neq S^*$
Alors nous avons l'égalité :
$\forall\ n \geq N^*, \quad T_n - S_n = (\tau_n - 1)(S_n - S^*)$.

2. Famille de procédés de synchronisation

Proposition 13.
Soit E un sous-ensemble de C^* synchronisable par le procédé PS. Alors, pour tout réel a non nul, le procédé PS_a est aussi un procédé de synchronisation de E.

3. Ensembles $E^-(p,\alpha)$ et $E^+(p,\alpha)$

Lemme 9.
Soit E un ensemble synchronisable.
Pour tous réels α et α' tels que $0 < \alpha' < \alpha$ et pour tout entier p, on a les inclusions suivantes :

(i) $E^-(p,\alpha) \subset E^-(p+1,\alpha)$
$E^-(p,\alpha) \subset E^-(p,\alpha')$

(ii) $E^+(p,\alpha) \subset E^+(p+1,\alpha)$
$E^+(p,\alpha) \subset E^+(p,\alpha')$

Lemme 10.
Soit E un ensemble synchronisable.
Pour toute suite (a_k) strictement décroissante de limite nulle, on a :

$$E = \bigcup_{k \geq 0} [E^-(k,a_k) \cup E^+(k,a_k)]$$

Lemme 11.

Soit E un sous-ensemble de C^* synchronisable.
Soient (S_n) une suite de E et $N^* \in N$ tel que :
$\forall n \geq N^*, S_n - S^* \neq 0$.
Pour tout réel $\alpha > 0$ et tout entier p, on a :
a) Si $(S_n) \in E^-(p,\alpha)$, alors :
$\forall n \geq \max(p,N^*), \quad (S_n-S^*)(T_n-S_n) < 0$
b) Si $(S_n) \in E^+(p,\alpha)$, alors :
$\forall n \geq \max(p,N^*), \quad (S_n-S^*)(T_n-S_n) > 0$

Lemme 12.

Soit E un sous-ensemble de C^* synchronisable par le procédé PS et soit un réel $\alpha > 0$.
Pour toute suite (S_n) de E et pour tout entier n, on a :
(a) $\tau_n \geq 1+\alpha \quad \Leftrightarrow \quad S^* \in\,]S_n, T_{-1/\alpha}(n)[$
(b) $\tau_n \leq 1-\alpha \quad \Leftrightarrow \quad S^* \in\,]S_n, T_{1/\alpha}(n)[$

Proposition 14.

Soient E un ensemble synchronisable par le procédé PS et (S_n) une suite de E.
Pour tout réel $\alpha>0$ et pour tout entier p, les conditions suivantes sont équivalentes :
(i) $(S_n) \notin E^-(p,\alpha)$
(ii) $\exists\, n \geq p, \quad \bigcap_{j=p}^{n} \langle S_j, T_{1/\alpha}(j)\rangle = \emptyset$

Proposition 15.

Soient E un ensemble synchronisable par le procédé PS et (S_n) une suite de E.
Pour tout réel $\alpha>0$ et pour tout entier p, les conditions suivantes sont équivalentes :
(i) $(S_n) \notin E^+(p,\alpha)$
(ii) $\exists\, n \geq p, \quad \bigcap_{j=p}^{n} \langle S_j, T_{-1/\alpha}(j)\rangle = \emptyset$

Corollaire 5.

Soient E un ensemble synchronisable par le procédé PS et (S_n) une suite de E.
Pour tout réel $\alpha>0$ et pour tout entier p, les conditions suivantes sont équivalentes :
(i) $(S_n) \notin E^-(p,\alpha) \cup E^+(p,\alpha)$
(ii) il existe $m > p$ tel que :

$$\bigcap_{j=p}^{m} \langle S_j, T_{1/\alpha}(j)\rangle = \emptyset$$

$$\bigcap_{j=p}^{m} \langle S_j, T_{-1/\alpha}(j)\rangle = \emptyset$$

4. Description du DAQES-algorithme

Théorème 13.
La question Q "quel est le signe de S_n-S^* ?" est asymptotiquement décidable sur tout ensemble synchronisable.

5. Autres versions du DAQES-algorithme sur des sous-ensembles de C*

a) E c C2

Lemme 13.
Soit E un sous-ensemble de C* synchronisable par le procédé PS et (S_n) une suite de E.
Alors il existe un entier N tel que, pour tout $n \geq N$:
(1) $T_n-S_n \neq 0$
(2) $(T_n-S_n)(T_{n-1}-S_{n-1}) < 0 \Leftrightarrow (S_n-S^*)(S_{n-1}-S^*) < 0$

Lemme 14.
Soit E un sous-ensemble de C2 synchronisable par le procédé PS et (S_n) une suite de E.
Alors, pour tout entier n, il existe $m \geq n$ tel que l'on ait $(T_m-S_m)(T_{m-1}-S_{m-1}) < 0$.

b) *.PROG1 U *.ALT

c) *.REG1 U *.ALT

C. LE ACCES-ALGORITHME

1. Description de l'algorithme

Proposition 16.
Soient E un sous-ensemble de C* synchronisable par le procédé PS et (S_n) une suite de E de limite S*.
Pour toute suite (Γ_n) telle que :
(i) $\lim_{n\to+\infty} \Gamma_n = \lim_{n\to+\infty} \tau_n = \tau$
La suite (A_n) définie par :

$$A_n = S_n - \frac{T_n - S_n}{\Gamma_n - 1} \quad \text{si } \Gamma_n \neq 1 \text{ (sinon } A_n = 0)$$

converge vers S* plus vite que (S_n).

Lemme 15.
Soit E un ensemble synchronisable par le procédé PS.
Pour toute suite (S_n) de E, il existe un entier N' tel que
$\forall n \geq N', 0 \leq p(n) < n$

Lemme 16.

Soit E un ensemble synchronisable par le procédé PS.
Pour toute suite (S_n) de E, nous avons :

(a) $\lim_{n\to+\infty} S_{p(n)} = S^*$

(b) $\lim_{n\to+\infty} T_{p(n)} = S^*$

(c) $\lim_{n\to+\infty} \tau_{p(n)} = \tau$

(d) Il existe un réel B et un entier Ñ tel que :

$$\forall\, n \geq \tilde{N},\quad 0 < \frac{S_n - S^*}{S_{p(n)} - S^*} < B < 1$$

Proposition 17.

Soit E un ensemble synchronisable par le procédé PS.
Pour toute suite (S_n) de E, la suite (Γ_n) définie par :

$$\Gamma_n = \frac{T_n - T_{p(n)}}{S_n - S_{p(n)}} \quad \text{si } S_n \neq S_{p(n)} \text{ (sinon } \Gamma_n = 1)$$

vérifie :

$$\lim_{n\to+\infty} \Gamma_n = \tau$$

Théorème 14.

Pour qu'un sous-ensemble de C^* soit accélérable, il faut et il suffit qu'il soit synchronisable.

2. Famille de procédés d'accélération

3. Choix du procédé d'accélération

Proposition 18.

Soit E un sous-ensemble de C^* synchronisable.
E est à délai borné q si et seulement si E est inclus dans q.BORN.

Corollaire 6.

Soit E un sous-ensemble de C^* synchronisable.
Alors les conditions suivantes sont équivalentes :

(i) E est à délai borné 1,
(ii) E est inclus dans MON ∩ APO-1

Corollaire 7.

Soit E un sous-ensemble de C2 synchronisable.
Alors les conditions suivantes sont équivalentes :

(i) E est à délai borné 2,
(ii) E est inclus dans OSC et pour toute suite (S_n) de E, les deux sous-suites (S_{2p}) et (S_{2p+1}) appartiennent à APO-1.

Proposition 19.

Soit E un sous-ensemble de C^* synchronisable.
E est à délai borné si et seulement si E est inclus dans $*$.BORN.

D. EXEMPLES DE SYNCHRONISATION

1. Etude des suites convergentes du type : $e_{n+1} = e_n - \alpha_n e_n^k$

Lemme 17.

Soit (S_n) une suite convergente de limite S^*, dont les erreurs $e_n = S_n - S^*$ vérifient :

$$e_{n+1} = e_n - \alpha_n e_n^k, \text{ avec } \lim_{n\to+\infty} \alpha_n = \alpha \neq 0.$$

Alors on peut écrire e_n sous la forme :

$$e_n = \delta_n n^{-h}, \text{ avec } h = (k-1)^{-1} \text{ et } \lim_{n\to+\infty} \delta_n = \delta \neq 0.$$

Proposition 20.

Soit (S_n) une suite convergente de limite S^*, dont les erreurs $e_n = S_n - S^*$ vérifient :

$$e_{n+1} = e_n - \alpha_n e_n^k, \text{ avec } \lim_{n\to+\infty} \alpha_n = \alpha \neq 0$$

Soit (T_n) la suite définie, pour tout entier $n \geq 0$, par :

$$T_n = S_{[n/2]}$$

Alors on a :

(1) (T_n) est synchrone avec (S_n),

(2) $$\lim_{n\to+\infty} \frac{T_n - S^*}{S_n - S^*} = 2^h \quad \text{avec } h = (k-1)^{-1}.$$

2. Etude des suites convergentes du type : $e_{n+1} = K e_n - \alpha_n e_n^k$, avec $|K| \leq 1$ et $K \neq 0$.

Proposition 21.

Soit (S_n) une suite convergente de limite S^*, dont les erreurs $e_n = S_n - S^*$ vérifient :

$$e_{n+1} = e_n - \alpha_n e_n^k, \text{ avec } \lim_{n\to+\infty} \alpha_n = \alpha \neq 0$$

Alors : $$\lim_{n\to+\infty} A_n = 2^{-h} \quad \text{où } h = (k-1)^{-1},$$

et $$\lim_{n\to+\infty} B_n = 1.$$

Proposition 22.

Soit (S_n) une suite convergente de limite S^*, dont les erreurs $e_n = S_n - S^*$ vérifient :

$$e_{n+1} = K\, e_n - \alpha_n\, e_n^k, \text{ avec } \lim_{n \to +\infty} \alpha_n = \alpha \neq 0 \text{ et } K \neq 1.$$

Alors : $\lim_{n \to +\infty} B_n = K,$

et lorsque $0 < K < 1$, on a :

$$\lim_{n \to +\infty} A_n = 0.$$

Proposition 23.

L'ensemble des suites (S_n) convergentes de limite S^*, dont les erreurs $e_n = S_n - S^*$ vérifient :

$$e_{n+1} = K\, e_n - \alpha_n\, e_n^k, \text{ avec } \lim_{n \to +\infty} \alpha_n = \alpha \neq 0$$

$$\text{et} \quad |K| \leq 1 \text{ et } K \neq 0$$

est synchronisable.

BIBLIOGRAPHIE

[1] C. BREZINSKI, Accélération de la convergence en Analyse Numérique, Lecture Notes in Mathematics, 584, Springer-Verlag, Heidelberg, 1977.

[2] J.P. DELAHAYE, Sequence transformations, Springer Series in Computational Mathematics, 11, Springer-Verlag, New York, (1988).

[3] J.P. DELAHAYE, B. GERMAIN-BONNE, Résultats négatifs en accélération de la convergence, Numer. Math., 35, 1980, pp. 443-457.

[4] B. GERMAIN-BONNE, Estimation de la limite de suites et formalisation des procédés d'accélération de convergence, thèse, Lille, 1978.

[5] B. GERMAIN-BONNE, C. KOWALEWSKI, Accélération de la convergence par utilisation d'une transformation auxiliaire. Lille, publication interne, ANO 77 (1982).

[6] B. GERMAIN-BONNE, Accélération de la convergence par utilisation d'une transformation auxiliaire. Lille, publications internes, ANO 81,88 (1982).

[7] B. GERMAIN-BONNE, A.M. LITOVSKY, Sufficient information for accelerating the convergence, actes du 12ème congrès IMACS, Paris 1988.

[8] C. KOWALEWSKI, Possibilités d'accélération de la convergence logarithmique, thèse de 3ème cycle, Lille, 1981.

[9] J.LELONG-FERRAND, J.M. ARNAUDIES, Cours de Mathématiques, tome 2, analyse, Ed. Dunod.

[10] SENHADJI, Transformations de suites par composition de méthodes. thèse de 3ème cycle, Lille, 1984.

RESUME

Etant données deux suites numériques convergentes, l'une converge plus vite que l'autre (resp. est synchrone avec l'autre) si et seulement si la suite des rapports des erreurs converge vers zéro (resp. admet une limite différente de un). Une famille **E** de suites numériques convergentes est accélérable (resp. synchronisable) par un procédé algorithmique si et seulement si, pour toute suite de **E**, la suite transformée par ce procédé converge plus vite que la suite initiale (resp. est synchrone avec la suite initiale).

Pour une famille de suites, être synchronisable semble à priori moins contraignant qu'être accélérable. Nous prouvons en fait que ces deux notions sont équivalentes. Plus précisément, nous construisons algorithmiquement, à partir de tout procédé de synchronisation d'une famille **E**, un procédé d'accélération de **E** appelé le ACCES-algorithme.

Une étape importante du ACCES-algorithme consiste en la détermination, à partir d'un certain rang, de la position de chaque terme de la suite à accélérer par rapport à sa limite. Cela nous a conduit à chercher différents types d'informations sur des ensembles de suites qui permettent de décider asymptotiquement du signe de l'erreur.

MOTS-CLES

Accélération de la convergence
Décidabilité asymptotique
Suite numérique
Synchronisation
Transformation algorithmique