

**MINISTRE DE L' ENSEIGNEMENT SUPERIEURE ET DE
 LA RECHERCHE SCIENTIFIQUE
 ECOLE NATIONALE POLYTECHNIQUE
 DEPARTEMENT ELECTRONIQUE**



**PROJET DE FIN D'ETUDE POUR
 L'OBTENTION DU DIPLOME D'INGENIEUR D'ETAT EN ELECTRONIQUE**

Thème

**COMMANDE ET ORIENTATION D'UN PANNEAU
 SOLAIRE A L' AIDE DU MICRO-CONTRÔLEUR
 68HC11**

**Proposé par : Mr. LARBES
 Mr. HADDADI**

**Réalisé par :
 MEKHAZNI Mohamed**

Promotion 2001

Remerciements

*Je tiens par le biais de ce modeste travail à remercier tous ceux
qui ont participé de près ou de loin à la réalisation de ce
mémoire.*

*Je tiens à remercier spécialement mon promoteur Mr. L'ARBES
et Mr. HADDADI.*

*Je tiens également à exprimer mes sincères reconnaissances à
madame HAMMAMI.*

Dédicace

Je dédie ce modeste travail à :

Mes chères parents.

Mes frères et sœurs.

Tous mes amis.

Mohamed

ملخص :

في هذه المذكرة لنهاية الدراسة قمنا بدراسة التحكم للوح الطاقة الشمسية باستعمال الميكروتحكمي 68HC11 , هذا الأخير متكون من دوائر و أجزاء إلكترونية كاملة و كافية تسمح له بالاشتغال كجزء إلكتروني واحد بمفرده و تنفيذ عدة برامج مخزونة في ذاكرته الميتة أو المبرمجة, المسماة أيضا ذاكرة المعطيات. أن التحكم في لوح الطاقة الشمسية يستلزم توظيف توابع رياضية مركبة من دوال مثلثية معقدة يصعب تمثيلها في اللغة الرقمية و لا حتى خزنها في ذاكرة المعطيات للميكروتحكمي 68HC11 لضخامة البرامج المثلثة لهذه التوابع, فكان لازما خلق طريقة أخرى للتحكم بتوابع رياضية بسيطة من جهة و من جهة أخرى السماح بالاستغلال المقتصد لذاكرة المعطيات, و فعلا تمكنا من إيجاد طريقة للتحكم جد فعالة, تتسم بالسهولة من ناحية البرمجة الرقمية, و مقتصدة لفضاء الذاكرة.

Résumé :

Dans ce mémoire de fin d'étude on a réalisé la commande d'orientation d'un panneau solaire en utilisant le micro-contrôleur 68HC11, ce dernier possède une architecture interne complète qui lui attribut la qualité d'être autonome et la possibilité d'exécuter des programmes stockés dans ça mémoire morte ou ça mémoire programmable, dite encore mémoire de données.

La commande d'orientation d'un panneau solaire exige l'utilisation des fonctions trigonométriques qui ne peuvent être exprimées en langage assembleur 68HC11, et même les implantées en mémoire sous forme de séries de fonction à cause du volume important d'espace mémoire nécessaire. A cet effet il a fallut trouver une autre expression de la commande capable de réunir à la fois la simplicité des formules mathématiques et la réduction d'espace mémoire utilisé.

Abstract:

In this memory we have discuss the order of orientation of a solar panel using the micro-controller 68HC11. This last possesses a complete internal architecture that it attribute the quality to be autonomous, and the possibility to execute programs stocked in it read-only memory or programmable memory, so-called memory of data.

The order of orientation of a solar panel require the use of trigonometric function, that cannot be expressed in Assembler 68HC11 language, and even implanted in memory under shape of polynomials functions, because of the important volume of memory space necessary. To this effect it was necessary to find another expression of the order able at a time to simplify the mathematical formulas and reduce the consumption of space memory.

Mots clés :

Angle optimal
Approximation
Assembleur
Autonomie
Binaire
Codage
Commande
Déclinaison
EEPROM
Energie solaire
Espace mémoire
Hexadécimal
Implantation
Interpolation
Mantisse
Microcontrôleur
Microprocesseur
Orientation
Panneau solaire
Polynômes
68HC811E2

Sommaire

INTRODUCTION GENERALE

Chapitre I

ETUDE DU MICROCONTROLEUR 68HC11

Introduction	03
II- Description du microcontrôleur 68HC11	04
II-1-1 Synoptique interne	04
II-2-2 Description des connexions du 68HC11	06
II-1-3 Les différents modes de fonctionnements	09
II-2 L'unité centrale	11
II-3 La mémoire	24
I-3-1 La RAM	25
I-3-2 La ROM	25
I-3-3 L' EEPROM	27
▪ Effacement de l' EEPROM :	
▪ programmation de l' EEPROM	

Chapitre II

PHOTOPILES ET PANNEAUX SOLAIRES

I- Principe de fonctionnement	30
II- Conversion photovoltaïque	31
II-1 Absorption des photons	32
II-2 Collecte des porteurs	32
III- Rendement d'une photopile	33
IV- Caractéristiques théoriques	33
V- Domaines d'utilisation	34
V-1 Applications terrestres sans concentration	34
V-2 Applications terrestres avec concentration	35
V-3 Applications spatiales	36

Chapitre III

Calcul et Adaptation Des Angles D'orientations d'un Panneau Solaire pour l'implantation sur EEPROM

PARTIE I

I- Calcul des angles d'orientations d'un panneau solaire	39
II- Adaptation des angles d'orientations (Approximation)	41
II-1 Nécessité de l'approximation	41
▪ Utilisation de l'expression	41
▪ implantation des valeurs des angles d'orientations directe sur EEPROM	41
▪ Approximation par des polynômes	42
II-2 Les erreurs	48

PARTIE II

Introduction	51
I- Application des algorithmes d'approximations :	51
I-1 Programme d'approximation <i>APPROXIM</i> (,)	51
I-2 Approximation de la courbe de la latitude d' Alger 36.7°	54
I-4 Approximation des courbes des latitudes comprises entre 36° et 22°	57
I-4-2 Construction des relations	58
I-4-3 Substitution de la matrice $G_0[365,2]$	59
I-4-4 Erreur commise avec l'approximation P_g	59
I-4-5 Substitution des matrices d'applications G_i	60
II- Construction de l'application G_s relative au deux partie Pseudo-symétrique de la référence	64
III- Les angles d'orientation des latitudes exprimées avec les degrés et les minutes	65
III-3 Généralisation de l'interpolation	67
IV Algorithme de génération de toutes les angles d'orientation à partir de Mg	68
Conclusion	71

Chapitre IV

IMPLANTATION SUR MICROCONTROLEUR 68HC11

Introduction	73
II- Choix du microcontrôleur	73
III Implantation des données sur EEPROM	73
III-1 Stockage des données	73
II-2 Codage des nombres signés avec virgule flottante	74

Conversion hexadécimale des chiffres avec virgule flottante	
II-1 Implantation de la matrice Mg	76
Programmes de la génération des angles d'orientations à partir de \$Mg	77
Sous programme « calcul et ajustement de la position du jour »	77
Sous programme « multiplication signée avec virgule flottante »	78
Sous programme « addition signée avec virgule flottante »	78
Sous programme « interpolation »	79
Sous programme « calcul de la forme polynomiale de second ordre »	79
Programme global	80
Mise en œuvre du 68HC11	81
Conclusion	81

Conclusion générale

ANNEXE I

« Les principales villes d'Algérie avec les latitudes correspondantes »

ANNEXE II

« La famille 68HC11Ex »

ANNEXE III

« Le programme Approxim(,) »

Bibliographie

Introduction générale

Le développement des applications à base d'un microcontrôleur est devenu l'une des applications de l'actualité les plus répondue dans de vastes domaines.

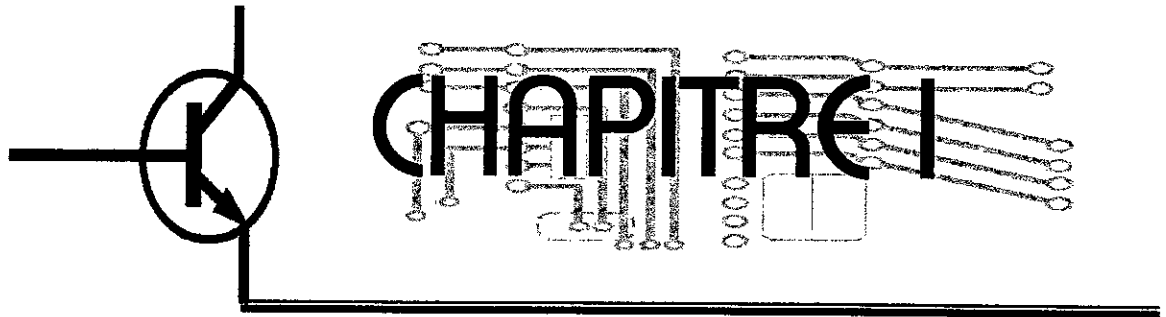
Malgré que ces composantes datent de plusieurs années, ils connaissent un grand essor , que ce soit en applications industrielles ou en applications domestiques.

La photovoltaïque est la plus jeune des énergies renouvelables, non épuisable, disponible en permanence. L'idée de la commande d'un panneau solaire s'avère fort bonne et exploitable sur des horizons à la fois techniques et économique.

Afin d'optimiser le rendement des cellules photovoltaïque, il a fallut concevoir des systèmes de commande qui peuvent orienter les cellules en question, et ceci constitue l'objet principale de notre travail.

Dans ce sens ; nous allons présenter notre travail comme suit :

Une présentation du micro-contrôleur utilisé (68HC11) débutera notre document, par la suite vient une brève description des photopiles. Juste après vient le calcul des angles d'orientation du panneau solaire. L'implantation de la commande sur micro-contrôleur constituera la finalité de notre travail. Et ce document sera clôturé par une conclusion générale.



LE Micro-contrôleur 68HC11

- Introduction
- Description du Microcontrôleur 68HC11
 - Synoptique interne
 - L'unité centrale
 - Reset et interruptions
 - La mémoire
- Conclusion

Introduction :

Le concept d'une puce généraliste, programmable à volonté sur le principe d'un ordinateur, s'est développé vers 1969; c'était l'invention du microprocesseur.

Les progrès technologiques rapides, augmentation du nombre de transistors intégrés et du nombre d'instructions exécutées par seconde. Ont fait que ce composant pénètre chaque jour d'avantage tous les domaines du monde industriel et du monde domestique.

Les descendants de ces premiers microprocesseurs sont présents dans plusieurs domaines de notre vie, Bien souvent, après intégration de nombreuses fonctions permettant de répondre à un maximum de besoins, ils sont dédiés à un usage unique. Ce sont les Microcontrôleurs.

Dans notre étude nous allons utiliser un microcontrôleur reconnue par ses qualités dans plusieurs applications, c'est le 68HC11 de MOTOROLA. Bien qu'il ait été mis sur le marché depuis déjà quelques années, le 68HC11 est resté relativement méconnu du grand public pendant assez longtemps et on ne doit l'intérêt croissant porté à cette famille de circuit que de puis sa découverte dans un nombre sans cesse plus important d'appareils les plus divers. Il faut reconnaître que cette présence n'est pas vraiment surprenante car le 68HC11 facilite énormément l'utilisation même dans les applications complexes, grâce à la puissance et de ces instructions et ces ressources internes ainsi son interfaçage complet : ports parallèles, convertisseurs analogiques-numériques, temporisateurs et liaisons série synchrone et asynchrone.

En effet l'appellation 68HC11 ne caractérise pas un seul circuit, mais toute une famille de circuits construit autour du même noyau de base, la famille 6800 des fameux microprocesseurs de MOTOROLA, bien que ce dernier ne soit pas un véritable microprocesseur à 16 bits, mais la particularité de son architecture interne lui permet de réaliser un certain nombre d'opérations arithmétiques et logiques sur 16 bits ce qui accroît encore sa souplesse d'emploi et sa puissance logicielle.

Le microcontrôleur 68HC11 est décliné en de très nombreuses versions, afin que chaque utilisateur trouve un modèle adéquat pour son application. dans notre projet nous allons travailler sur la version dite E2 ou plus exactement 68HC811E2, disponible au laboratoire de l'école (ENP). Dans ce chapitre nous allons faire une étude générale de ce microcontrôleur.

I- Description du microcontrôleur 68HC11:

I-1 Synoptique interne[1] :

Avant de décortiquer le contenu du microcontrôleur 68HC11 ce qui sera fait le long de ce chapitre, nous proposons de découvrir en figure I.1 ce qui est contenu dans le noyau de base de cette famille .

Autour de l'unité centrale 68HC11 proprement dite qui, est dérivée du microprocesseur 6801, on trouve d'abord de la mémoire.

Elle se subdivise au maximum en trois blocs distincts dont la taille et la présence varient selon les références exactes du circuit. La RAM ou mémoire vive, de 256 octets au moins est toujours présente. la ROM ou mémoire morte peut être présente ou absente. Lorsqu'elle est présente, elle peut être du type ROM programmable par masque, c'est-à-dire lors de la fabrication du circuit, ou du type EPROM c'est-à-dire programmable électriquement, dans ce dernier cas, cette mémoire EPROM peut être non effaçable; elle s'appelle alors OTPROM pour One Time PROM ,ROM programmable une fois. Elle peut aussi être de type EEPROM, programmable et effaçable électriquement.

Dans tous les cas, cette mémoire destinée au programme ne doit pas être confondue avec la petite EEPROM visible sur ce synoptique qui est, elle, destinée aux données.

L'unité centrale est entourée d'un certain nombre de ports parallèles baptisés port A à port E qui peuvent être bidirectionnels ou unidirectionnels selon le cas, certaines lignes de ces ports sont également partagées avec d'autres ressources internes et ne sont donc pas nécessairement accessibles directement en permanence.

Des entrées/sorties séries sont aussi disponibles et peuvent fonctionner en mode synchrone ou asynchrone selon que l'on utilise SPI ou la SCI.(Voir figureI-1).

Un timer ou plus exactement un sous-ensemble timer est également disponible. Il comporte plusieurs timers très évolués ainsi qu'un accumulateur d'impulsions, une horloge temps réel et "un chien de garde" ou COP (pour Computer Operating Properly) destiné à surveiller le fonctionnement du microcontrôleur.

Un convertisseur analogique-numérique à huit entrées, est intégré ainsi que toute la logique nécessaire, tant au traitement des interruptions qu'à la génération d'horloge. La mise en œuvre matérielle est ainsi fort simple puisque, dans de nombreux cas, un quartz et une cellule RC pour le Reset sont les seules composantes externes indispensables pour le pilotage de cette logique. Remarquez aussi, en partie basse de la figure I-1. Que le circuit peut fonctionner selon deux modes: un mode "single chip" ou circuit seul et un mode "expanded" ou étendu qui lui permet alors d'adresser des circuits externes comme

n'importe quel microprocesseur ordinaire. Dans ce dernier cas, les lignes des ports B et C se transforment en lignes d'adresses et de données.

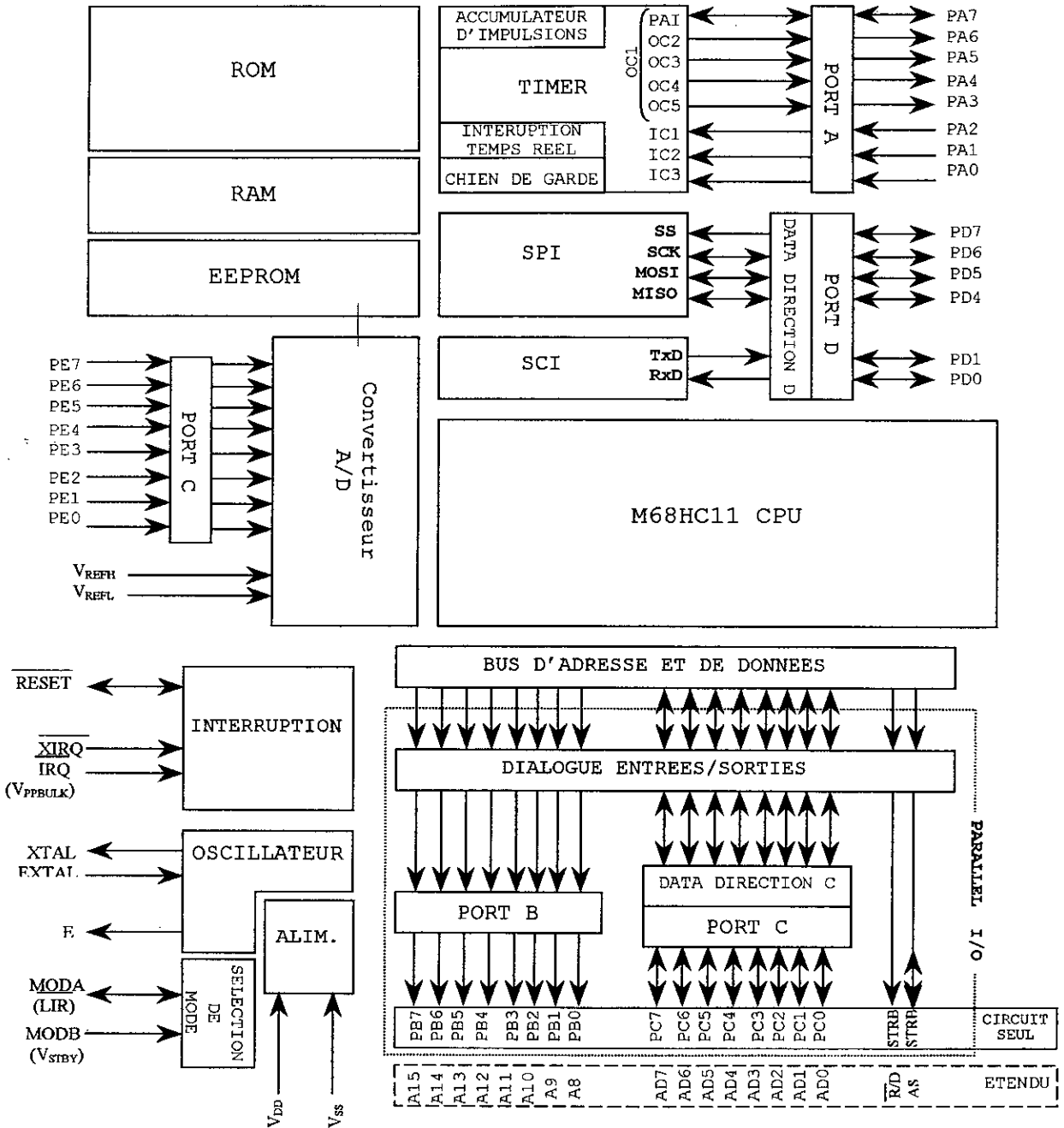


Fig. I-1: Synoptique interne du 68HC11

I-2 Description des connexions du 68HC11:

□ VDD et VSS

Ce sont des connexions d'alimentation du circuit. Le circuit est réalisé en technologie HCMOS, il est alimenté avec une tension de 5 V, ce qui le rend compatible avec les circuits CMOS et TTL.

□ MODA et MODB

Ces deux signaux permettent de choisir (pendant la durée du Reset) le mode de fonctionnement du micro-contrôleur. En dehors de cette phase de Reset la ligne MODA devient LIR (LIR : Load Instruction Register ie chargement du registre d'instruction) et peut être employé pour synchroniser un appareil de mesure durant la mise au point d'un programme. LIR passe au niveau logique 0 pendant le premier cycle d'horloge E de l'exécution d'une instruction.

MODB peut servir à alimenter la RAM interne dès que la tension VDD devient inférieure à $V_{stby} - 0.7$ volts. La commutation se fait automatiquement grâce à une logique interne.

□ EXTAL et XTAL

Ils permettent la connexion du quartz de l'horloge ou d'une horloge externe. Généralement on utilise un quartz de 8 MHz mais on peut descendre cette fréquence en dessous à condition de faire attention à la limite fixée par le surveillant d'horloge. La fréquence de l'horloge du bus est égale au quart de la fréquence d'horloge. Ces deux connexions peuvent aussi être utilisées comme entrée d'horloge (horloge externe) : EXTAL, ou comme sortie d'horloge.

□ Signal E

Sortie de l'horloge bus, cette ligne est surtout utilisée en mode étendu, pour dialoguer correctement avec certains périphériques.

□ VREFL et VREFH

Ces deux connexions permettent d'appliquer les tensions de référence utiles au convertisseur. Ces tensions ne doivent pas dépasser la tension d'alimentation du 68HC11 pour ne pas endommager celui-ci et l'écart entre VREFL et VREFH ne doit pas descendre en dessous de 2.5 V pour garder une bonne précision au convertisseur.

Ces entrées doivent être très soigneusement découplées.

□ **IRQbarre et XIRQbarre**

Sont des entrées d'interruptions externes. IRQbarre est une entrée d'interruption masquable alors que XIRQbarre est non masquable.

□ **RESET**

C'est la broche Reset ou remise à l'état initial mais, contrairement à de nombreux circuits, c'est une ligne bidirectionnelle. Utilisée en entrée et forcée à la masse, elle provoque une ré-initialisation du circuit. Si par contre un événement interne anormal survient tel que par exemple la détection par le chien de garde ou COP d'une anomalie, ce dernier provoque un Reset interne du 68HC11 mais fait aussi générer un niveau bas à cette patte afin que cette information puisse, le cas échéant, être exploitée par des boîtes externes. Afin d'éviter l'altération du contenu de l'EEPROM mais aussi de certains registres réalisés selon la même technologie, il est fortement conseillé de piloter cette broche par un circuit détectant automatiquement toute baisse de VDD en dessous de la valeur limite autorisée.

□ **STRA et STRB**

Sont respectivement AS et R/Wbarre en mode étendu. En mode circuit seul, ces lignes peuvent être utilisées pour divers protocoles de dialogue via les ports B et C. STRA est sensible à un front et permet de mémoriser dans un registre spécial associé au port C les données appliquées sur celles des lignes de PC0 à PC7 qui ont été placées en entrées. Le sens du front actif est programmable par logiciel. Si elle n'est pas utilisée pour provoquer cette mémorisation, STRA peut toujours servir à provoquer une interruption lors de la réception d'un front actif. En mode circuit seul, STRA doit être ramener au VDD par une résistance de rappel de 4.7 à 10 K Ω si elle n'est utilisée. STRB quant à elle est une sortie indiquant lorsque des données valides sont présentes sur les lignes des port C et D qui ont été placées en sortie (fonction strobe). Si STRB n'est pas utilisée en tant que telle, elle peut servir de sortie à usage général mais sa manipulation est moins simple que celle d'une classique ligne de port.

□ **PE0 à PE7**

Sont les entrées du port parallèles E. Ces lignes sont unidirectionnelles et ne fonctionnent qu'en entrée. De plus elles sont partagées avec les entrées du convertisseur analogique-digital mais, malgré cela il est possible de les utiliser simultanément dans les

deux modes. En effet, lors d'une entrée de donnée numérique, le buffer interne correspondant n'est mis en fonction que pour un très court instant ce qui peut introduire une légère perturbation uniquement si cela se produit pendant une phase d'échantillonnage de l'entrée correspondante.

□ PD0 à PD5

Sont les entrées-sorties parallèles du port D qui sont communes avec les lignes des ports série de la SPI et de la SCI. La figure I-1, précise l'affectation relative de ces pattes aux différentes lignes spécifiques de ces interfaces. Il est évident qu'ici aussi toute simultanéité de fonction est impossible sauf quelques cas particuliers. Ainsi, même avec la SPI en fonction, la ligne SSbarre peut être utilisée comme sortie sous réserve que l'interface fonctionne en mode maître. Ces six lignes pouvant être configurées pour fonctionner en « OU » câblé, il est nécessaire soit de désactiver ce mode par logiciel, soit de prendre la précaution de ramener ces lignes par des résistances de rappel au VDD.

□ PC0 à PC7

Sont les entrées-sorties parallèles à usage général en mode circuit seul. En mode étendu, elles deviennent les lignes où circulent les données et les huit bits de poids faibles des adresses en mode multiplexé. La ligne AS (Adress Strobe) dont le rôle est alors tenu par la patte STRA indique alors si le port C véhicule des adresses (AS au niveau haut) ou si ce sont des données (AS au niveau bas). Le rôle de la ligne lecture, écriture est tenu quand à lui par la patte STRB.

□ PB0 à PB7

Sont des sorties parallèles en mode circuit seul et véhiculent les 8 bits de poids forts des lignes d'adresses en mode étendu. Les remarques faites ci-avant pour le port C s'appliquent aussi ici. Il est évident que, si le circuit est utilisé en mode circuit étendu, les sorties PB0 à PB7 et les entrées sorties PC0 à PC7 sont perdues. Il est toute fois possible de les récupérer en plaçant sur le bus externe du 68HC11 un boîtier appelé 68Hc24 ou PRU (pour Port Replacement Unit) qui présente alors le même aspect, vu du logiciel, que les ports B et C perdus.

□ PA0 à PA7

Sont les entrées (PA0, PA1, PA2), les sorties (PA3, PA4, PA5, Pa6) et l'entrée/sortie (PA7) du port A. Ces lignes sont partagées avec celle du timer ce qui justifie le fait que certaines d'entre-elles soient unidirectionnelles. En effet, PA0 à PA2. sont aussi les entrées de capture IC3 à IC1 du timer ; PA3 à PA 6 sont aussi les sorties de comparaison OC1 à OC5 du timer tandis que PA7 peut être configurée comme entrée de capture ou sortie de comparaison. Si le timer à accumulation des impulsions est utilisé, c'est également PA7 qui lui sert d'entrée. Il est évident si une des pattes de sortie de comparaison est validée pour cette fonction au niveau du timer, elle ne peut être utilisée en tant que patte de sortie parallèle à usage général.

I-3 Les différents modes de fonctionnements du 68HC11 :

Comme nous l'avons vu dans le schéma synoptique du microcontrôleur, les deux pattes MODA et MODB peuvent sélectionner les modes de fonctionnements de ce dernier. En effet le microcontrôleur peut fonctionner selon les modes suivant :

□ Mode *single chip* ou circuit seul :

C'est le mode microcontrôleur proprement dit. Le circuit remplit seul toutes les fonctions, sans l'utilisation de quelconques circuits logiques externes.

□ Mode étendu ou multiplexé

Le microcontrôleur utilise des ports d'entrées/sorties parallèles comme des bus d'adresses et de données, cela lui permet ainsi d'adresser 64ko. Le bus d'adresses est constitué du port B pour les bits de poids fort et du port C pour les bits de poids faible, ce dernier étant multiplexé pour obtenir aussi le bus de données. L'espace adressable comprend les adresses que l'on retrouve en mode mono-chip.

Le signal R/W est obtenu à partir de la ligne STRB et le signal de commande de multiplexage est obtenu à partir de la ligne STRA.

□ Le mode spécial test :

Ce mode est réservé au fabricant pour tester le bon fonctionnement du microcontrôleur en fin de chaîne de fabrication.

□ Le mode spécial Bootstrap :

Ce mode charge en RAM un programme par la liaison et lance son exécution. Ce mode de fonctionnement permet à l'utilisateur de brancher le microcontrôleur avec la

liaison série asynchrone de celui-ci sur un ordinateur et ainsi grâce au dialogue établi, de mettre au point et de charger un logiciel dans ce microcontrôleur. Dans celui-ci le fabricant a placé un logiciel en ROM (de \$BF40 à \$BFFF) appelé *chargeur (boot loader)* et son vecteur Reset. Au moment du Reset, le microcontrôleur lit les deux connexions MODA et MODB, si celles-ci sont toutes les deux à 0, il exécute le programme boot-loader. Celui-ci initialise la liaison série pour établir la liaison avec l'ordinateur, une fois cette liaison établie, l'ordinateur envoie le logiciel à tester dans la RAM du microcontrôleur (256 octets maximum pour 68HC811E2) puis lance celui-ci à partir de l'adresse 0. bien entendu, pour pouvoir fonctionner avec des branchements et des interruptions il est nécessaire de prévoir aussi dans la RAM un emplacement pour la PILE et un emplacement pour les vecteurs d'interruptions. Ceux-ci sont alors appelés pseudo-vecteurs (tableau I-1).

ADRESSE	ORIGINE
00C4-C6	SCI
00C7-C9	SPI
00CA-CC	ACC. IMPULSION ENTREE
00CD-CF	DEBORDEMENT ACC. IMPULSION
00D0-D2	DEBORDEMENT TIMER
00D3-D5	TOC5
00D6-D8	TOC4
00D9-DB	TOC3
00DC-DE	TOC2
00DF-E1	TOC1
00E2-E4	TIC3
00E5-E7	TIC2
00E8-EA	TIC1
00EB-ED	INTERRUPTION TEMPS Réel
00EF-F0	IRQ
00F1-F3	XIRQ
00F4-F6	SWI
00F7-F9	Code opération illégal
00FA-FC	COP

00FD-FF	Surveillance d'horloge
---------	------------------------

Tableau I-1

I-4 L'UNITE CENTRALE:

L'unité centrale du 68HC11 est un modèle 8 bits disposant de fonctionnalités particulières lui permettant d'exécuter avec un maximum de souplesse certaines instructions, y compris arithmétiques et logiques, sur 16 bits. Elle est issue de l'unité centrale 6801 mais a été enrichie de nouvelles instructions.

L'unité centrale des 68HC11 est capable d'exécuter, au niveau code objet, tout programme écrit pour un 6800 ou un 6801 ce qui peut être intéressant pour faire évoluer rapidement une application existante. De ce fait, les programme sources 6800 et 6801 peuvent évidemment être soumis eux aussi à un assembleur 68HC11 pour donner du code exécutable. Quatre-vingt-dix nouvelles instructions étant cependant disponibles par rapport au jeu d'instructions initiales de ces deux microprocesseurs, il est toutefois évident qu'il sera préférable de réécrire tout ou une partie d'une application existante pour bénéficier pleinement de la puissance du 68HC11.

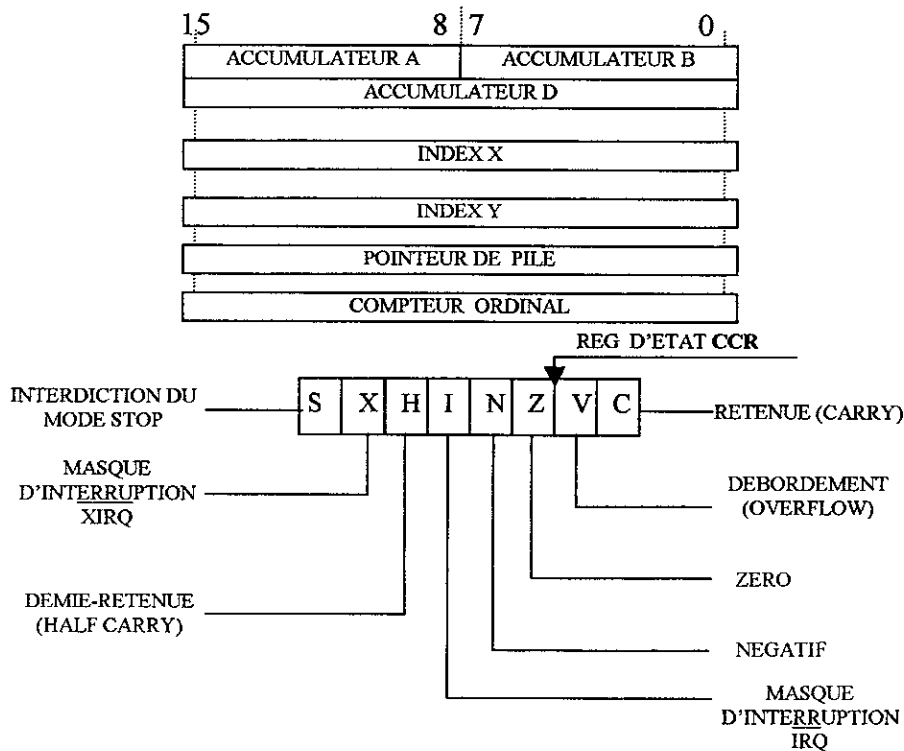


Fig.I-2 Les registres de l'unité centrale

I-4-2 Registres de l'unité centrale :

La figure ci avant montre les différents registres de l'unité centrale du 68HC11, le registre CCR (registre d'état) est illustré d'une manière plus détaillée en raisons de l'importance de ce dernier.

1- l'accumulateur D :

Il est constitué par la mise bout à bout des accumulateurs A et B. en conséquence toute modification de l'accumulateur D affecte le contenu de l'accumulateur A ou/et de l'accumulateur B. il permet de faire des opérations arithmétiques et logiques sur 16 bits.

2- Les registres X et Y : registres d'index

Ce nom vient du fait qu'ils sont utilisés principalement pour contenir un index (repère d'adresse). On peut aussi les utiliser pour stocker temporairement des informations sur 16 Bits ou encore comme compteur.

Le mode d'adressage appelé indexé permet de reconstituer une adresse effective en additionnant le contenu du registre d'index avec un offset sur 8 bits.

3- Le pointeur de pile : Stack Pointer(SP)

C'est un registre 16 bits qui contient la première adresse libre de la zone mémoire définie pour contenir la PILE.

La pile est une zone mémoire RAM, utilisée par l'unité centrale elle-même ou par le programmeur pour sauvegarder momentanément certaines informations.

La position de la pile est déterminée par le programmeur, il lui suffit de mettre l'adresse dans le pointeur de pile, toutefois avec précaution car elle est utilisée par l'unité centrale.

4-le compteur de programme PC :

Le compteur de programme est un registre de 16 bits qui contient l'adresse de la prochaine instruction à exécuter. Après l'opération RESET, le PC est chargé avec l'adresse de début du programme.

5- Le registre d'état C.C.R : Code Condition Register

Le registre d'état (C.C.R) est un registre qui renseigne sur l'état de l'unité centrale après chaque opération effectuée. Il contient 8 indicateurs (drapeaux : *flags*), chacun ayant une signification bien particulière. La figure *fig.I-2* montre les différents bits du registre, comme on le peut voir.

Les bits indicateurs S et X sont spécifiques à l'unité centrale du 68HC11, tandis que les autres bits se trouvent avec les mêmes significations dans tous les microprocesseur 8 bits de la gamme.

□ **C (Carry) : retenue**

Ce bit est mis à 1 automatiquement quand une opération arithmétique génère une retenue ou un emprunt. Il peut servir de lien dans des opérations de décalage ou de rotation à droite ou à gauche.

Le bit C peut aussi servir d'indicateur d'erreur lors d'une multiplication, d'une division ou d'une addition en mode binaire non signé.

□ **V : *overflow* ou débordement**

Ce bit est un indicateur de dépassement de capacité qui passe à 1 lorsque l'opération arithmétique, en mode binaire signé, donne un résultat supérieur au contenu maximum de l'accumulateur.

□ **Z : *zero***

Ce bit est mis à 1 chaque fois que le contenu du registre concerné par l'opération arithmétique ou logique ou encore par une manipulation de donnée devient nul. Autrement il est mis à 0.

□ **N : négatif**

Ce bit est mis à 1 si le résultat de l'opération arithmétique, logique ou manipulation de donnée est négatif, c'est-à-dire si le bit de poids fort est à 1, (en binaire signé le bit de poids fort représente le signe du nombre, 0 nombre positif, 1 nombre négatif).

□ **I : masque ou indicateur d'interruption**

En tant qu'indicateur d'interruption, ce bit passe à 1 dès qu'une interruption de programme est prise en compte, c'est-à-dire quand les registres sont sauvegardés dans la pile et avant que le vecteur d'interruption ne soit recherché.

Si ce bit est à 1, il interdit toute interruption masquable, toutefois si le périphérique qui a demandé l'interruption le permet, celle-ci est mémorisée et est traitée ultérieurement quand I repasse à 0.

Ce bit est mis à 1 par le reset et peut être remis à 0 par une instruction appropriée.

Le bit I est normalement à 0 avant la prise en compte d'une demande d'interruption, donc après le retour de celle-ci provoqué par l'instruction RTI, le bit doit reprendre sa valeur d'origine, ie 0.

□ **H : (*Half Carry*) indicateur de demie-retnue**

Ce bit est mis à 1 quand une retenue apparaît entre les bits 3 et 4 d'une opération ADD, ABA ou ABC. Autrement ce bit est à 0. il permet d'effectuer des opérations arithmétiques sur des nombres BCD sans avoir à les convertir en binaire naturel (sur 4 bits). Le résultat est ensuite exploité par l'instruction DAA (conversion en BCD) qui peut alors effectuer un report sur les 4 bits, dans ce cas le bit H passe à 1.

□ **X : masque d'interruption de XIRQ**

Le fonctionnement est similaire à celui de IRQ mais est spécifique à une demande d'interruption XIRQ du micro-contrôleur 68HC11. Après un reset, ce bit est mis à 1 et peut être mis à 0 par une instruction appropriée (TAP). Comme pour le bit I, X reprend sa valeur initiale au retour d'une interruption. Il ne peut être mis à 1 par programmation.

□ **S : stop disable ou masque de stop**

S permet quand il est mis à 1, d'empêcher l'instruction STOP d'être effective. Dans ce cas celle-ci est considérée comme une instruction NOP (non opération).

L'instruction STOP arrête le micro-contrôleur donc le met en mode faible consommation. Si l'instruction STOP est rencontrée alors que S est à 1, elle n'a aucun effet.

II- RESET et INTERRUPTIONS

II-1 LE RESET

Après avoir placé le programme en mémoire on peut se demander comment faire pour lancer et ainsi mettre en route l'application

L'opération reset permet ce démarrage, elle consiste à placer un niveau logique 0 sur la connexion appelée reset du boîtier du micro-contrôleur. Une fois cette tension détectée par l'unité centrale, une procédure d'initiation interne est déclenchée. Celle-ci va aller chercher dans la case mémoire \$FFFE et \$FFFF l'adresse de départ du logiciel et va la placer dans le compteur de programme. Cette position en haut de l'espace mémoire est

spécifique à la gamme 68xx de chez Motorola. Les deux cases mémoires contenant cette adresse constituent le vecteur de reset. Une fois le compteur de programme chargé, le programme peut alors démarrer.

II-1-2 Initiation au moment du reset :

La connexion reset est en réalité une entrée/sortie (ligne bidirectionnelle). Le reset initialise un certain nombre de registres afin que les programmes de l'utilisateur puissent bien les configurer. Les bits du registre d'état I, X et S sont mis à 1, donc toutes les interruptions sont masquées et le mode stop est interdit au moment du reset, le programme peut alors en toute sécurité initialiser son pointeur de pile.

Le registre INIT est chargé avec 01, donc la RAM est placée à partir de \$0000 et les registres à partir de \$1000.

Le déclenchement de l'opération reset provoque la mise à l'état bas (en sortie) de la ligne reset. Ceci permet éventuellement de déclencher le reset sur des périphériques externes.

II-1-3 Les quatre sources de reset :

1- POR (power on reset) à la mise sous tension

Ce type de reset est destiné à initialiser le microcontrôleur. Une transition positive de Vdd génère le POR qui est utilisé seulement pour l'initiation au démarrage, un délai de 4096 cycles d'horloge interne permet à l'oscillateur de se stabiliser. Si la connexion est encore à l'état 0 après ces 4096 cycles, le micro-contrôleur reste en condition de reset jusqu'à ce que la connexion passe à l'état logique 1.

POR ne peut être utilisé pour détecter des micro-coupures d'alimentation. Le 68HC11 a besoin d'un circuit détecteur de niveau qui maintienne la connexion à l'état bas tant que la tension d'alimentation Vdd n'ait pas atteint le minimum requis.

Le vecteur de ce type de reset se trouve en \$FFFE \$FFFF.

3- Reset extérieur :

Le micro-contrôleur distingue un reset interne d'un reset externe en détectant si la connexion repasse à l'état 1 en un temps inférieur à deux cycles d'horloge E après un reset interne. Quand une condition de reset est détectée, la connexion est mise à 0 par un circuit interne au micro-contrôleur pendant 4 cycles d'horloge E puis est relâchée. Deux cycles plus tard elle est échantillonnée, si la connexion est restée à 0, le micro-contrôleur en déduit qu'il s'agit d'un reset externe. Dans le cas contraire, cela veut dire que le reset est

d'origine interne, provoqué par le COP (surveillance du logiciel) ou par la surveillance d'horloge (CM).

C'est pour cela qu'il ne faut pas mettre de circuit RC sur la connexion reset car la constante de charge de celui-ci peut induire en erreur le micro-contrôleur.

4- Surveillance de fonctionnement (chien de garde) : le COP (Computer Operating proprely) :

Un compteur interne est décrémenté en permanence. Quand le logiciel fonctionne normalement, il recharge ce compteur avec un délai programmé bien précis l'empêchant ainsi d'arriver à 0. si le logiciel est perturbé, le rechargement n'a plus lieu en temps voulu et un niveau 0 est placé pendant 4 cycles d'horloge sur la connexion reset.

Le bit NOCOP du registre CONFIG permet la mise en service du COP. Le délai de rechargement est programmable avec les bits CR0 et CR1 du registre OPTION (adresse \$1039).

Le vecteur associé se trouve en \$FFFA \$FFFB.

5- Surveillance d'horloge : clock monitor (CM)

La surveillance d'horloge CM déclenche un reset si la période du signal d'horloge devient supérieure à une valeur déterminée par la constante du temps d'un circuit RC interne (5 μ s). Pratiquement ceci interdit de faire descendre la fréquence d'horloge en dessous de 200 Khz. Cette surveillance complète le COP car celui-ci ne peut détecter d'erreur d'horloge.

Le bit CME du registre OPTION permet la mise en service de la surveillance d'horloge.

Remarque:

L'instruction STOP arrête l'horloge et par conséquent n'est pas compatible avec la surveillance d'horloge. Il faut donc mettre le bit CME du registre OPTION à 0, avant d'utiliser le stop.

Le vecteur associé se trouve en \$FFFC \$FFFD

II-2 LES INTERRUPTIONS :

Une interruption comme son nom l'indique interrompt un programme en cours pour faire exécuter à l'unité centrale un autre programme (suite d'instructions appelée aussi sous-programme). Celui-ci se termine par une instruction de retour d'interruption RTI qui



permet à l'unité centrale de reprendre le programme principal, dans les même conditions, c'est-à-dire celles qui existaient dans l'UC au moment de l'interruption.

II-2-1 Origine d'une interruption :

Sur le boîtier du micro-contrôleur il existe des connexions d'entrées actives au niveau logique 0 et qui ont la possibilité de déclencher des interruptions dites externes (IRQ, XIRQ). Dans un micro-contrôleur les interfaces qui sont intégrées, ont la possibilité aussi d'effectuer des demandes d'interruption.

Il existe aussi une instruction SWI (SoftWare Interrupt) qui déclenche une demande d'interruption.

Les circuits externes doivent maintenir le niveau pendant un temps suffisant pour que le micro-contrôleur puisse prendre en compte l'instruction.

II-2-2 Opération effectuée lors d'une interruption :

La demande étant généralement provoquée par un événement extérieur, elle n'est pas synchrone avec le déroulement de l'instruction en cours, par conséquent, l'instruction en cours est d'abord complètement terminée.

Puis les étapes suivantes se suivent :

- Tous les registres sont sauvegardés dans la pile dans ordre précis ;
- Le pointeur de pile est décrémenté du nombre d'octets chargés ;
- Le masque d'interruption correspondant au registre d'état est mis à 1 ; interdisant ainsi la prise en compte d'une nouvelle demande d'interruption ;
- Le contenu du vecteur d'interruption de même mon (adresse de départ de sous-programme d'interruption) est chargé dans le PC (compteur programme) puis le sous programme d'interruption est effectué.

quand le sous-programme d'interruption est terminé :

(celui-ci se termine obligatoirement par l'instruction RTI)

l'instruction RTI permet :

- A l'UC de récupérer tous ces registres à partir de la pile, donc l'état initial des bits I et X ;
- Au pointeur de pile d'être incrémenté du nombre d'octets correspondant et au compteur de programme de se charger avec l'adresse de l'instruction de programme principal qu'il devait exécuter au moment de la demande de l'interruption.

II-2-3 Hiérarchie des interruptions :

Nous avons vu que les demandes d'interruptions peuvent avoir plusieurs origines et qu'il existe deux bits masques d'interruptions (I et X) dans le registre d'état. Compte-tenu du nombre important d'origines possibles de demande d'interruptions, il est nécessaire de prévoir des priorités car en cas d'arrivée simultanée de plusieurs demandes, c'est celle qui est de priorité la plus élevée qui sera exécutée en premier puis ensuite celle de priorité immédiatement inférieure, et ainsi de suite.

Ces priorités sont définies par le constructeur selon l'ordre ci-dessous.

II-2-4 Interruption non masquables (dans l'ordre de priorité)

- POR ou reset externe
- CM surveillance d'horloge
- COP chien de garde
- XIRQ
- Interruption pour code illégal
- Interruption par programme SWI

II-2-5 Interruption masquables

L'ordre ici n'est pas précisé, car dans le cas du micro-contrôleur 68HC11 le constructeur a prévu la possibilité pour le programmeur de modifier cette hiérarchie grâce au registre HPRIO (\$103C). Celui-ci, grâce au bits PSEL0 à PSEL3. Déterminant la source de l'interruption que l'on veut plus prioritaire après XIRQ.

- débordement timer
- accumulateur d'impulsion plein
- front d'entrée sur l'accumulateur d'impulsion
- fin de transmission série synchrone
- transmission série asynchrone
- IRQ
- Interruption par l'horloge temps réel
- Entrée de capteur 1
- Entrée de capteur 2
- Entrée de capteur 3
- Entrée de capteur 4

- Entrée de capteur 4 ou sortie de comparaison 5

II-3 Les vecteurs d'interruptions :

Les vecteurs d'interruption qui contiennent les adresses de départ des sous-programmes sont placés en haut de l'espace adressable. Le programmeur doit donc comme pour le reset remplir ces vecteurs avant la mise en service de l'application.

II-3-1 Remarques importantes (spécificité du micro-contrôleur):

a- Interruption pour code illégal :

Le micro-contrôleur est capable de détecter un code machine incorrecte ou inconnu quand le décodeur d'instruction ne peut générer une séquence connue de micro-instruction. Dans ce il génère une interruption dont le vecteur se trouve en \$FFF8 \$FFF9.

L'utilisateur a donc intérêt à prévoir un sous-programme pour palier cette éventualité ou à défaut de provoquer un reset (ce qui ne résoudra pas le problème !) pour éviter un plantage pour les périphériques pilotés.

b- La fonction STOP et WAIT :

L'instruction STOP place le micro-contrôleur en état de plus faible consommation (100 μ A) si le bit S du registre d'état est à 0. si le bit S du registre d'état est à 1, la fonction STOP est considérée comme un NOP (Non Operation) sans effet.

En état STOP les horloges sont arrêtées donc tout les processus aussi. Le contenu de la RAM est maintenu, et les entrées/sorties sont inchangées. A la remise en route un délai est nécessaire pour que la fréquence d'horloge puisse se stabiliser (voir le bit DLY du registre OPTION).

Pour quitter cette fonction on peut utiliser les lignes d'interruptions IRQ, XIRQ ou reset.

Pour la fonction WAIT (attente d'interruption),cette instruction stoppe l'exécution du programme après avoir compilé tous les registres. Dans ce cas, la consommation du circuit dépend du nombre d'horloges en service. L'oscillateur à quartz reste actif. Pour sortir de cet état on peut utiliser IRQ, XIRQ ou une interruption d'origine interne.

III- Les registres du micro-contrôleur 68HC11 :

Le micro-contrôleur 68HC11 contient 64 registres distincts, ces registres sont spécialement conçus pour des différentes tâches de configuration et fonctionnement de ce dernier.

Dans ce qui suit nous allons voir les registres les plus importants, ceux du fonctionnement du micro-contrôleur, la configuration et l'emplacement des mémoires, les vecteurs d'interruption ainsi que la protection.

III-1 Le registre config \$103F

Le registre config comporte en réalité deux registres distincts, l'un en EEPROM qui conserve les informations en permanence et l'autre est un registre de travail normal en RAM.

Après un reset, le registre en EEPROM est recopié dans le registre de travail, une fois le micro-contrôleur en route ; il devient alors possible de modifier le registre en EEPROM par programme.

Ce registre peut différer d'une version de micro-contrôleur à une autre car les structures peuvent être différentes.

Le registre config est protégé par le bit PTCN mi à 1 du registre BPRTsauf pour les modèles A. Sa modification nécessite la même procédure de programmation et effacement que l'EEPROM.

<i>config</i>	EE3	EE2	EE1	EE0	NOSEC	NECOP	ROMON	EEON
---------------	-----	-----	-----	-----	-------	-------	-------	------

- **EE3, EE2, EE1, EE0** détermine l'emplacement de l'EEPROM, celle-ci peut être placée n'importe où sur les 64 ko de l'espace adressable et ceci par secteur de 4 ko , si par exemple : Si EEn (0000) l'EEPROM va de \$0800 jusqu'à \$0FFF
Si EEn (1111) l'EEPROM va de \$F800 jusqu'à \$FFFF
(cas du 68HC811E2)
- **NOSEC : eeprom security disable**

Ce bit est normalement à 1 lors de la fabrication du circuit, ce qui veut dire que la sécurité est enlevée, toutefois on peut demander au fabricant de mettre cette sécurité en

service. Sécurité veut dire que le circuit ne peut être employé qu'en mode circuit seul ou en mode bootstrap, ceci interdit la relecture du logiciel qui ne peut se faire qu'en mode étendu.

□ **NOCOP :**

Ce bit permet d'inhiber le système de surveillance de fonctionnement du logiciel (COP) appelé chien de garde.

□ **ROMON : Rom On**

Ce bit est spécifique aux modèles A, pour le modèle 68HC811E2, ce bit est forcé à 1, dans ce cas la ROM est validée, dans le cas contraire (bit à 0) cette mémoire est invalidée. En mode circuit seul l'état de ce bit est ignoré et il est considéré comme étant à 1 puisque l'appel à une ROM externe est impossible dans ce cas.

□ **EEON EEPROM ENABLE**

Ce bit est normalement à 1, si ce bit est à 0, la zone mémoire EEPROM est retirée de la carte mémoire.

III-2 Le registre INIT \$103D :

Ce registre permet de modifier par pas de 4 ko, les emplacements des registres et de la RAM. Il est protégé car il ne peut être écrit que pendant les 64 cycles d'horloge E qui suivent un RESET.

INIT \$103D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0
Au RESET	0	0	0	0	0	0	0	1

- Si RAMn (0000) la RAM va de \$0000 jusqu'à \$00FF
- RAMn (1000) la RAM va de \$F1000 jusqu'à \$10FF
- Si REGn (0001) les registres vont de \$1000 à \$103F
- REGn (1111) les registres vont de \$F000 à \$F03F

On peut remarquer qu'il y'a un risque de chevauchement des registres sur la zone RAM. Que l'on se rassure, les registres sont prioritaires sur la RAM, et les octets communs sont dévalidés sur la zone RAM .

III-3 Le registre OPTION \$1039

Ce registre permet la mise en service d'un certain nombre de fonction. Certains de ces bits sont protégés (IRQE, DLY, CR1 et CR0) et ne peuvent être modifiés que pendant les 64 premiers cycles d'horloge E qui suivent un RESET.

OPTION \$1039	ADPU	CSEL	IRQE	DLY	CME	0	CR1	CR0
Au RESET	0	0	0	1	0	0	0	0

□ **ADPU (A/D power Up)**

Ce bit mis à 1 met la mise en service de la pompe de charge du convertisseur analogique/numérique. Il faut attendre 100 µs avant d'utiliser le CAN.

□ **CSEL (Clock Select) : selection d'horloge**

Permet de sélectionner l'horloge de la pompe de charge du convertisseur A/D et du système d'écriture en EEPROM. Doit mis à 1 si la fréquence d'horloge est trop faible pour la pompe de charge.

Si CSEL =0, l'horloge est le signal E

Si CSEL =1, l'horloge est fixée par un circuit RC interne .

□ **IRQE (IRQ Enable)**

Ce bit détermine le signal actif de l'entrée de demande d'interruption IRQ.

Si IRQE =0, l'entrée est sensible à un niveau logique 0

Si IRQE =1, l'entrée est sensible à un front descendant .

□ **DLY (enable oscillator startup DeLaY) délai de mise en route de l'oscillateur.**

L'instruction STOP permet d'arrêter l'oscillateur du micro-contrôleur, à la remise en route on peut déterminer un délai d'attente pour s'affranchir du temps de stabilité de l'oscillateur.

Si DLY est à 1, le temps d'attente est de 4064 cycles d'horloge E.

Si DLY est à 0, la remise en route se fait après 4 cycles d'hologe E.

□ **CME (clock Monitor enable) mise en service de la surveillance d'horloge**



L'horloge du micro-contrôleur peut être surveillée en permanence, ce bit permet la mise en service ou non de cette surveillance.

Si CME est à 1 la surveillance est en service

Si CME est à 0 la surveillance est arrêtée

- **CR[0 :1] Cop timer Rate select : programmation de la période d'horloge du chien de garde**

Le COP est un chien de garde qui surveille le bon déroulement d'un programme, ces deux bits déterminent le facteur de division de l'horloge E déjà pré-divisé par 2^{15} .

III-4 Le Registre BPROT \$1035

Ce registre permet la protection du registre CONFIG et la protection par bloc de 512 octets de l'EEPROM.

BPROT \$1035	0	0	0	PTCON	BPRT3	BPRT2	BPRT1	BPRT0
Au RESET	0	0	0	1	1	1	1	1

- **PTCON**

Si PTCN est à 1, le registre CONFIG ne peut être effacé ou programmé.

Si PTCN est à 0 le registre CONFIG peut être programmé et effacé normalement.

- **BPRT[3 :0]**

Ces 4 bits sélectionnent le paquet de 512 octets qui doivent être protégés.

Si le bit est à 1, la protection est en service.

BPRT0 protège de \$x800 à \$x9FF, BPRT1 protège de \$xA00 à \$xBFF etc.

III-5 Le registre HPRIO

Le registre HPRIO contient 4 bits qui sont relatifs aux interruptions, les autres bits sont utilisés pour la configuration du fonctionnement du micro-contrôleur.

HPRIO \$103C	RBOOT	SMOD	MDA	IRV	PSEL3	PSEL2	PSEL1	PSEL0
Au RESET	0	0	0	0	0	1	0	1

□ **PSEL[0 :3]**

Détermine la source de l'interruption que l'on veut la plus prioritaire après XIRQ.

□ **IRV (*Internal Read Visibility*)**

Si ce bit est à 1, les données sont disponibles sur le bus externe lors d'une lecture. S'il est à 0 cette fonction est invalide. Ce bit ne peut être écrit que si SMOD (voir ci-après) est à 1. Il est forcé à 0 dans le cas contraire.

□ **MDA (*MoDe A select*)**

Si ce bit est à 1 le circuit est en mode étendu ou en mode spécial test, s'il est à 0, le circuit est mode circuit seul ou bootstrap. Ce bit ne peut être écrit que lorsque SMOD est à 1.

□ **SMOD : *Special MODe***

Si ce bit est à 1, le circuit est dans l'un des deux modes spéciaux. Si le bit est à 0, le circuit est dans l'un des modes normaux.

□ **RBOOT : *Read BOOTstrap***

Si ce bit est à 1 la lecture de la ROM de bootstrap est validée des adresses \$BF40 à \$BFFF. Si ce dernier est à 0, la ROM de bootstrap est dévalidée et non présente dans l'espace adressable.

IV- La mémoire:

Jusqu'à trois types de mémoires différentes peuvent coexister dans un 68HC11; mais nous allons donner plus d'importance à la mémoire celle du type EEPROM, et à ces divers modes de fonctionnement.

Un 68HC11 dispose d'une zone de mémoire vive ou RAM dont la taille varie selon la référence exacte du boîtier de 256 octets pour le 68HC11A8 (et 68HC811E2) jusqu'à 1.25Ko pour 68HC11M2.

Il dispose également d'une zone de mémoire morte ou ROM qui est programmable par masque, c'est-à-dire lors de la fabrication du circuit, sur les versions 68HC11 sans chiffre entre le C et le 11. Cette ROM interne peut être désactivée volontairement par l'utilisateur du circuit au moyen d'un bit du registre CONFIG ou être désactivée en usine par MOTOROLA comme le cas pour le 68HC11A0 par exemple. Dans un cas comme

l'autre, l'utilisation du mode étendu avec mémoire de programme externe est alors indispensable pour que l'application puisse fonctionner.

Les 68HC11 disposent aussi d'une mémoire EEPROM qui ne doit pas être confondue avec celle dont nous parlerons dans ce qui suit dans le cas des 68HC11. Il s'agit en effet d'une EEPROM de petite taille que l'on appelle habituellement EEPROM de données même si en peut éventuellement y stocker du programme. Cette mémoire a essentiellement pour rôle, de mémoriser des paramètres de configuration, sélectionnés par l'utilisateur et qui doivent être conservés même en absence de toute alimentation.

IV-1 La RAM:

La RAM du 68HC11 est du type statique et ne nécessite donc aucune circuiterie externe de rafraîchissement de même qu'elle n'impose aucune fréquence minimum d'horloge. Cette RAM peut être déplacée par bloc de 4 Ko dans tout l'espace mémoire adressable grâce aux quatre bits prévus à cet effet dans le registre INIT. Suite à un Reset, cette RAM est placée par défaut à partir de l'adresse \$0000 ce qui rend ainsi utilisable direct par le mode d'adressage direct dont l'étendue d'action n'est que de 256 octets.

Malgré la présence de l'EEPROM de données, il peut être nécessaire de sauvegarder le contenu de la RAM d'une période de fonctionnement de l'application à une autre. Pour cela deux méthodes totalement différentes sont disponibles.

La première, qui est aussi la plus simple, passe par l'utilisation de l'instruction STOP. Cette instruction arrête l'horloge du 68HC11 et suspend donc toute activité interne tout en préservant le contenu de la RAM et des registres. Comme le 68HC11 est réalisé en technologie CMOS, l'essentiel de sa consommation se produit lors des transitions des signaux logiques. L'horloge étant arrêtée, ces transitions absentes et le boîtier ne consomme plus ainsi que quelques μA .

Si cette solution n'est plus utilisable, ce qui est le cas généralement dans les applications où d'autres circuits, beaucoup plus gourmands en énergie, imposent d'arrêter l'alimentation. Il faut alors faire appel à la patte MODB/ V_{stby} . Dans ces conditions, il est possible d'appliquer sur cette patte une tension qui ne sera utilisé que pour maintenir le contenu de la RAM interne.

Une circuiterie interne au 68HC11 commute alors automatiquement l'alimentation de la RAM de VDD à V_{stby} lorsque VDD devient inférieur d'environ 0.7 à V_{stby} .

IV-2 La ROM :

Comme nous l'avons dit auparavant, la ROM peut être ou non présente dans le 68HC11 selon le type exact de boîtier retenu. Lorsqu'elle est présente, elle peut toujours être désactivée par un bit du registre CONFIG. Dans ce cas elle n'occupe plus aucun *emplacement* dans l'espace mémoire adressable.

Cette ROM qui est par défaut de type programmable par masque, c'est-à-dire réservée à des productions en grande série vu les frais de masquage qu'elle implique, peut se transformer en EPROM c'est-à-dire en ROM programmable électriquement mais non effaçable (OTPROM) sur les 68HC711 ou en EEPROM c'est-à-dire en mémoire programmable et effaçable électriquement sur les 68HC811.

Ces dernières versions de boîtier, sont évidemment réservées aux phases de développement d'une application tandis que les versions OTPROM sont destinées à des fabrications en petite série où les frais de masquage d'une version ROM ne peut être amortis.

Ces deux versions, OTPROM et EEPROM, se programment soit sur des programmeurs universels du commerce supportant le 68HC11, soit avec des montages spécifiques.

Cette ROM ou EPROM ou EEPROM, est évidemment utilisée pour le stockage des programmes et des vecteurs de Reset et d'interruptions placés à ses adresses les plus hautes. Si elle est absente de l'intérieur du 68HC11 ce dernier doit impérativement fonctionner en mode étendu avec cette mémoire en externe.

Cette ROM est toujours complétée par un petit morceau de ROM supplémentaire de 192 octets implantée de \$BF40 à \$BFFF qui contient le bootloader.

IV-3 L'EEPROM :

L'EEPROM de petite taille ou EEPROM de données qui équipe la majorité des versions de 68HC11 peut être utilisée comme de la ROM c'est-à-dire qu'elle peut être lue sans aucune restriction ni précaution particulière. Par contre, elle ne peut évidemment être écrite ou programmée que moyennant une procédure bien précise. De la même façon elle peut être effacée, octet par octet, ligne part ligne ou d'un seul coup grâce là aussi à des commandes protégées pour des raisons de sécurité.

Aucune haute tension spéciale n'est à prévoir sur le 68HC11 pour effacer ou programmer cette mémoire, un circuit interne à pompe de charge génère en effet cette tension à partir de l'alimentation normale VDD sans aucune composante externe.

Ce circuit peut être piloté par deux sources d'horloge distinctes : l'horloge bus système E ou une horloge générée localement par une cellule R-C interne au circuit. Lorsque la fréquence de l'horloge E est supérieure ou égale à 1.5 Mhz environ c'est elle qui doit être utilisée alors que, pour des fréquences inférieures, il faut faire appel à l'horloge R-C. La sélection se fait au moyen d'un bit du registre OPTION.

	-	CSEL	-	-	-	-	-	OPTION
Reset	0	0	0	1	0	0	0	0

- Bit 6 ou bit CSEL pour Clock SElect. Lorsque ce bit est à 0, l'horloge E est utilisée, alors que sa mise à 1 provoque l'utilisation de l'horloge à cellule RC. Il faut mentionner que ce bit agit aussi sur le même type de processeur pour le convertisseur analogique-numérique.

La gestion des fonctions particulières de l'EEPROM est assurée par le registre PPROG. Ce registre comprend un certain nombre de bits qui doivent impérativement être manipulés de façon décrite ci-dessous si l'on veut pouvoir effacer ou programmer l'EEPROM.

Le registre CONFIG, est également constitué au moins en partie d'un registre en EEPROM. Bien qu'il soit indépendant de l'espace mémoire de l'EEPROM proprement dite, il est également contrôlé par le registre PPROG et se programme exactement comme l'EEPROM elle-même. Cependant, pour des raisons de sécurité, il ne peut être effacé que par le mode effacement global et cet effacement ainsi que toute modification de son contenu ne peuvent avoir lieu que lorsque le 68HC11 est placé en mode **bootstrap**

	ODD	EVEN	0	BYTE	ROW	ERASE	EELAT	EEPGM	PPROG
Reset	0	0	0	0	0	0	0	0	0

La signification des bits du registre PPROG est suivante :

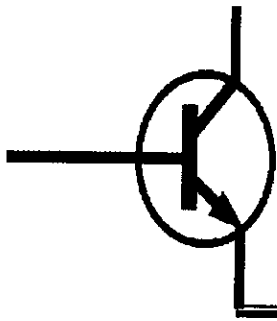
- Bit 0 ou bit EEGM pour EEprom ProGraMming voltage enable. S'il est à 1 ce bit autorise la génération interne de la tension de programmation VPP indispensable pour les opérations de programmation et effacement. Une logique interne interdit de mettre ce bit à 1 si le bit EELAT n'a pas été mis à 1 avant.

- Bit 1 ou bit EELAT pour Éeprom Latch control. Lorsque ce bit est à 0, l'EEPROM se comporte comme une mémoire ordinaire et peut être lue sans restriction. Lorsque ce bit est mis à 1, l'EEPROM se comporte comme si elle avait été enlevée de l'espace adressable du 68HC11 et mise sur un programmeur. Les données et les adresses qui lui sont appliquées sont mémorisées de façon à rester stables le temps nécessaire à leur programmation dans la mémoire. L'accès en lecture à l'EEPROM avec ce bit à 1 est impossible. De ce fait, un programme qui écrit dans l'EEPROM ou qui l'efface ne peut être exécutée depuis l'EEPROM elle-même.
- Bit 2 ou bit ERASE pour effacement. S'il est à 0 ce bit place l'EEPROM en mode lecture ou écriture, s'il est à 1 il configure la mémoire pour l'effacement.
- Bit 3 ou bit ROW pour ligne et bit 4 ou bit BYTE pour octet. Ces deux bits sélectionnent le mode d'effacement de la mémoire conformément aux indications du tableau I-1. Ces bits n'ont aucune signification hors le mode d'effacement programmé par mise à 1 du bit ERASE.

BYTE	ROW	Type d'effacement
0	0	Effacement complet
0	1	Effacement Ligne (16 octets)
1	0	Effacement d'un octet
1	1	Effacement d'un octet

Tableau I-1 : Sélection des différentes modes grâce au registre PPROG

- Bit 6 ou bit EVEN pour pair et bit 7 ou bit ODD pour impair. Ces deux bits ne sont utilisés en principe qu'en phase de test par MOTOROLA. Ils permettent de programmer en une seule opération tout le côté pair de la mémoire (si EVEN est à 1) ou tout le côté impair (si ODD est à 1) avec la même donnée. Comme le circuit interne de pompe de charge qui génère VPP est incapable de fournir le courant nécessaire à une telle opération, il faut faire appel à une haute tension de programmation interne (20V typique mais peut varier selon le type exacte de 68HC11 utilisé) appliquée sur la patte IRQ dont l'autre nom est VPPBULK. MOTOROLA déconseille l'usage de cette fonction hormis ses propres applications de test en fabrication.



CHAPITRE II

PHOTOPILES SOLAIRES

- Principe de fonctionnement
- Conversion photovoltaïque
- Rendement d'une photopile
- Caractéristiques théoriques
- Domaines d'utilisation



PHOTOPILES SOLAIRES[2]

I- Principe de fonctionnement

Les photopiles sont des composants électroniques à semi-conducteurs qui, lorsqu'ils sont éclairés par le rayonnement solaire, développent une force électromotrice capable de débiter un courant dans un circuit extérieur. Leur fiabilité et leur faisabilité ont été largement prouvées depuis les années 1960, puisque la plupart des satellites artificiels utilisent ces dispositifs pour tirer directement leur énergie du rayonnement solaire (qui atteint $1,35 \text{ kW/m}^2$ hors de l'atmosphère). Sur terre, l'absorption préférentielle par l'atmosphère de certaines longueurs d'onde modifie quelque peu le spectre solaire, mais l'énergie fournie est toujours importante, non polluante, inépuisable et assez uniformément répartie à la surface du globe. En effet, l'énergie annuelle globale ne varie que d'un facteur 2 des régions nordiques aux régions les plus favorisées, souvent tropicales et désertiques.

Les spécialistes du photovoltaïque utilisent comme unité de référence la puissance crête (maximale) débitée par une photopile à $25 \text{ }^\circ\text{C}$ sous une puissance lumineuse incidente fixée, pour les tests d'évaluation, à 1 kW/m^2 . La distribution spectrale du rayonnement solaire dépend de l'épaisseur de l'atmosphère à traverser (Air Mass: AM). On utilisera, par exemple, AM0 (hors atmosphère) pour les applications spatiales et AM1, ou AM1,5, pour les applications terrestres pour lesquelles la photopile est placée au niveau de la mer dans des conditions d'éclairement qui correspondent à un ciel parfaitement clair et dégagé, sous un soleil de midi au zénith (90°) pour AM1 ou d'inclinaison (U 420) pour AM1,5 (c'est-à-dire pour un rayonnement traversant 1 ou 1,5 fois l'épaisseur de la couche atmosphérique). Plus le nombre d'unités de masse atmosphérique sera élevé, plus le spectre solaire se décalera vers le proche infrarouge.

En revanche, l'utilisateur sera évidemment plus intéressé par le prix du kilowattheure, seule unité qui lui permette de comparer le prix de l'énergie solaire à celui de l'énergie fournie par le réseau ou par d'autres sources. Le passage de la puissance crête (disponible à midi) à l'énergie (journalière ou annuelle) disponible n'est pas simple, puisqu'il faudra tenir compte de la durée de l'ensoleillement local.

II- Conversion photovoltaïque

La structure la plus simple d'une cellule photovoltaïque comporte une jonction entre deux zones dopées différemment d'un même matériau (homojonctions p-n) ou entre deux matériaux (hétérostructures). La moins épaisse, ou celle de largeur de bande interdite la plus grande, est soumise au flux lumineux.

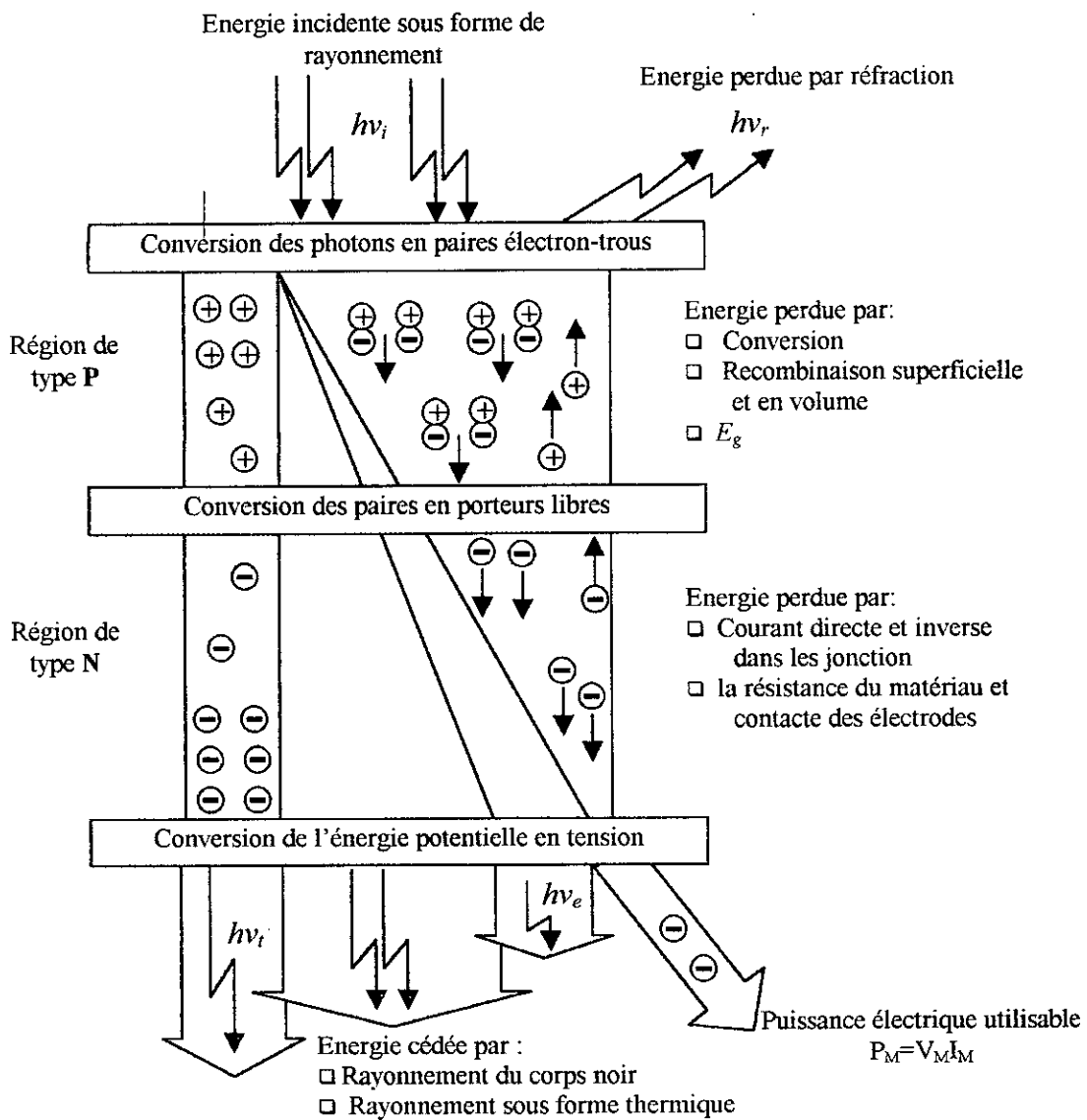


Fig II-1 : Conversion photovoltaïque

Chacune des régions est reliée à une électrode métallique au moyen d'un contact ohmique de faible résistance. Le principe de fonctionnement, relativement simple, peut être décomposé en deux parties (figure II-1) l'absorption de photons et la collecte des porteurs de charge créés.



II-1 Absorption des photons

Soit une cellule illuminée par un flux de photons d'intensité $J(h\nu)$. Selon leur énergie, les photons sont réfléchis (R = coefficient de réflexion du matériau), ou absorbés (α = coefficient d'absorption), ou transmis. Les photons absorbés, seuls utiles à la génération de l'effet photovoltaïque, sont ceux dont l'énergie $h\nu$ est supérieure ou égale à la largeur de bande interdite du matériau ($E_g = 1,1 \text{ eV}$ dans le silicium). On définit ainsi une grandeur appelée rendement d'absorption η_a :

$$\eta_a = E_g \frac{\int_{E_g}^{\infty} (1-R) e^{-\alpha d} J(h\nu) d(h\nu)}{\int_{E_g}^{\infty} h\nu J(h\nu) d(h\nu)}$$

où d est l'épaisseur de la cellule. Dans le cas du silicium, η_a est de l'ordre de 40 %.

L'énergie des photons absorbés sert partiellement à briser une liaison de valence, qui crée, de ce fait, une paire électron-trou susceptible de se mouvoir. L'énergie excédentaire est rapidement cédée au réseau cristallin sous forme de chaleur (phonons).

II-2 Collecte des porteurs

Les paires ainsi créées principalement au voisinage de la surface éclairée vont diffuser vers la face arrière sous l'action d'une force qui est liée à leur gradient de concentration.

Un grand nombre d'entre elles se recombinent en chemin, suivant un phénomène inverse de celui de l'absorption, et sont perdues pour le processus de conversion. Les paires qui atteignent la zone de charge d'espace sont alors séparées: le champ électrique de la jonction accélère les électrons vers la région n et les trous vers la région p. L'accumulation de ces charges dans des régions où elles sont majoritaires donne naissance à une différence de potentiel aux bornes de la cellule en circuit ouvert (V_{oc}). Si les bornes sont fermées sur une résistance d'utilisation (R_c), un courant I_c s'établira dans le circuit ainsi constitué. Le courant maximal I_{sc} est obtenu quand on court-circuite les bornes de la cellule ($R_c = 0$). La cellule se comporte donc comme un générateur de puissance électrique.



III- Rendement d'une photopile

La puissance fournie au circuit extérieur par la cellule éclairée dépend de la valeur de la résistance de la charge R_c ; on définit la quantité:

$$f_c = \frac{V_M I_M}{V_{sc} I_{sc}}$$

dite facteur de courbe. Cette quantité, toujours inférieure à l'unité (généralement de l'ordre de 0,85), est importante pour l'efficacité de conversion, le rendement de la photopile étant défini par le rapport:

$$\eta = \frac{V_M I_M}{\text{puissance totale incidente}}$$

Pour une énergie totale incidente de 1 kW/m^2 , le rendement d'une photopile silicium de structure simple (à jonction p-n comme celle décrite ci-dessus) aura un rendement qui ne dépassera pas les 20 %. Les rendements théoriques de structures plus complexes (un minimum de pertes électriques en volume et en surface) peuvent atteindre 25 à 27 %, suivant l'estimation des pertes électriques inéluctables.

IV- Caractéristiques théoriques

Le circuit équivalent simplifié d'une photopile peut être décrit comme un générateur de courant égal à I_{sc} , shunté par une diode qui a les caractéristiques de la cellule à l'obscurité. Dans ces conditions, l'expression générale du courant est:

$$I = I_{sc} - I_0 \left[\exp \left[\frac{q(V - IR_s)}{kT} \right] - 1 \right] + \frac{V - IR_s}{R_{sh}}$$

où I_0 représente le courant de saturation de la diode ($I_0 = 10^{-12}$ à $10^{-13} \text{ A} \cdot \text{cm}^{-2}$ pour une diode au silicium).

Dans le cas où R_{sh} est très grand et R_s négligeable, on obtient une expression approchée de la tension:

$$V = \frac{kT}{q} \ln \left(1 + \frac{I_{sc} - I}{I_0} \right)$$

avec $I_0 = 10^{-12} \text{ A} \cdot \text{cm}^{-2}$, $kT/q = 0,025 \text{ V}$ et $I_{sc} = 4 \times 10^{-2} \text{ A} \cdot \text{cm}^{-2}$. On trouve



alors $V_{oc} = 0,6 \text{ V}$.

Il faut noter que la diminution des pertes de recombinaison de surface dans les structures les plus sophistiquées, permet de réduire le courant de saturation I_0 à des valeurs de l'ordre de $10^{-13} \text{ A} \cdot \text{cm}^{-2}$ et de porter la tension V_{oc} à presque 700 mV.

V- Domaines d'utilisation

On peut diviser les applications de l'énergie photovoltaïque en trois grands domaines.

V-1 Applications terrestres sans concentration

Le coût d'une extension d'un réseau centralisé d'électricité augmente en fonction de la distance, et sa rentabilité dépendra beaucoup de la puissance à raccorder. Dans les zones rurales des pays en voie de développement ou à faible densité de population, l'électricité photovoltaïque apparaît de plus en plus comme la solution la plus satisfaisante pour les petites quantités d'énergie. L'électrification des populations rurales dans les pays en développement ne progresse que très lentement. En raison d'un pouvoir d'achat limité des abonnés, certains des réseaux déjà installés ne sont utilisés qu'à 10 % de leur capacité, alors que les pertes en ligne peuvent dépasser 15 %. C'est ainsi qu'au Moyen-Orient et en Asie le pourcentage d'électrification des zones rurales est passé de 15 à environ 20 % en quinze ans et est resté totalement inchangé à 4 % pour les pays subsahariens.

Les besoins de ces populations peuvent parfaitement être couverts par des photopiles pour alimenter des lampes, des pompes à eau, des téléviseurs et des réfrigérateurs. L'installation de minicentrales de recharge d'accumulateurs est à développer, d'autant que leur usage y est déjà très répandu. La rentabilité d'une installation photovoltaïque par rapport à un petit groupe Diesel est déjà assurée. On compte un peu plus de 3 000 pompes solaires installées et un millier de réfrigérateurs. C'est encore très peu par rapport aux besoins.

En résumé, grâce à des aides de la Banque mondiale et à des coopérations, le minimum d'électricité indispensable à la vie "moderne" pourra être apporté par les photopiles et contribuer ainsi à stopper l'immigration des populations des zones rurales vers des villes surpeuplées.

Dans les pays développés, de petits convertisseurs d'énergie photovoltaïque peuvent être utilisés pour fournir certains services dans le domaine des loisirs avec le caravanning et la plaisance, dans les nombreux petits systèmes où l'on peut remplacer les



piles chimiques lampes de jardin, systèmes de veille électronique, de surveillance. L'exemple typique est la calculette de poche, pour laquelle l'énergie est de plus en plus souvent fournie par de petites barrettes de photopiles à base de silicium amorphe qui fonctionnent sous les néons. Ce type d'utilisation s'est fortement développé (plus de 100 millions de calculettes sont sur le marché), bien que le rendement de conversion n'est que de quelques pourcents et que le prix du kilowattheure est au moins 100 fois supérieur à celui du réseau.

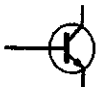
D'autres applications sont économiquement rentables dans le domaine public par exemple, l'équipement de refuges de montagne, de systèmes de communication ou des relais hertziens (plus de 75 000 installés dans le monde). Dans les zones isolées, et à titre expérimental, un certain nombre de maisons ont été équipées avec des panneaux photovoltaïques. On en compte environ 30 000, qui se partagent pour moitié entre l'Europe et les Etats-Unis, dont plus de 4 000 dans les départements et territoires d'outre-mer.

Enfin, la Communauté européenne, l'A.D.E.M.E., le D.O.E. américain, le M.I.T.I. japonais et certaines compagnies d'électricité ont mis en place de petites centrales photovoltaïques pour stimuler le marché mondial, qui a été en progression constante jusqu'en 1992 et qui stagne depuis lors à un niveau qui se situe aux alentours de 50 MWc/an. Ces centrales, qui contribuent à la mise au point de générateurs dans des conditions réelles d'utilisation, produisent de l'électricité à l'aide de panneaux plans qui couvrent une surface proportionnelle à la puissance crête installée. On peut citer, à titre d'exemple et par ordre d'importance, celles de Kaw en Guyane de 35 kWc et de Poomia en Corse de 44 kWc, les installations de Delphes en Grèce (300 kWc) et de Kobern-Gaundorf en Allemagne (340 kWc), ainsi que celles de Saijo et de Lugo, toutes deux de 1 MWc et respectivement installées au Japon et aux Etats-Unis. Enfin, la plus importante de ces centrales est celle de Carina Plain, de 8 MWc, de la Southern California Edison Co., qui a installé les panneaux photovoltaïques sur de grands héliostats qui suivent le Soleil.

En plus d'un suivi des mouvements apparents du Soleil, on peut focaliser son rayonnement sur des photopiles de plus petites dimensions, qui devront être plus sophistiquées et plus performantes.

V-2 Applications terrestres avec concentration

Cette technique de production d'électricité qui utilise la concentration du rayonnement solaire n'est intéressante que dans des régions à faible nébulosité, c'est-à-



dire essentiellement dans celles qui entourent le bassin méditerranéen et, en général, dans toutes les zones tropicales du type saharien, où l'ensoleillement direct est très important.

L'intérêt économique d'un tel dispositif dépendra du rapport qui existe entre le prix de la photopile et le coût du système mécanique de suivi et de l'optique de concentration du Soleil. En fait, la plupart des études ont montré que le rendement de conversion devra nettement dépasser 20 % à des températures de fonctionnement relativement élevées pour que cette utilisation soit intéressante.

Plusieurs méthodes ont été proposées pour focaliser le rayonnement solaire d'un facteur compris entre quelques unités et 2 000. Pour les plus faibles valeurs de celui-ci, on préférera des systèmes fixes à miroir réfléchissant la lumière sur le panneau photovoltaïque ou des suivis qui suivent un seul axe (hauteur ou azimut du Soleil). Le choix des cellules dépendra également du taux de concentration envisagé. Pour toutes celles qui sont inférieures à 100, on peut faire appel au silicium; les cellules ne diffèrent de celles qui sont employées sans concentration que par une optimisation des structures et des différentes résistances, afin de permettre l'évacuation des densités de courant plus élevées qui y circulent.

Pour une concentration plus importante, trois voies principales ont été explorées: l'emploi d'une cellule GaAs, capable de donner des rendements importants même à des températures élevées; la dispersion par un prisme du rayonnement solaire focalisé de manière à séparer différents domaines de longueurs d'onde; dans chaque zone est placée une cellule optimisée pour ce type de photons. Des rendements très élevés ont été atteints par cette méthode; enfin, la mise en place, dans le foyer du concentrateur, d'une cellule à très haut rendement (plus de 35 %), qui est formée de plusieurs matériaux épitaxiés les uns sur les autres et comportant plusieurs jonctions de façon à récupérer l'ensemble des photons du spectre solaire (il s'agit, en fait, d'intégrer le système précédent). □terme, il devrait être possible d'atteindre des rendements de 50 % avec des structures comportant trois matériaux. Mais de grands problèmes se posent pour l'intégration de plusieurs matériaux dont la maille cristalline est différente et, surtout, pour le nombre des jonctions à préparer (une douzaine).

V- Applications spatiales

Depuis le lancement du premier satellite Vanguard (États-Unis) en 1959, alimenté en électricité par des photopiles, les critères du choix du matériau le mieux adapté aux applications spatiales sont assez proches de ceux qui sont nécessaires à la concentration du point de vue des performances, avec, en plus, le problème du rapport rendement/ poids, de sorte qu'on a conservé relativement longtemps des photopiles à base de silicium



monocristallin amincies afin d'en réduire le poids, avant de se tourner vers de nouveaux matériaux en couches minces, par exemple les tellurures et les sulfures de cadmium (qui n'ont eu guère de succès en raison de leur faible durée de vie) et, plus généralement, vers des alliages composés.



CHAPITRE III

Calcul et Adaptation Des Angles D'orientations d'un Panneau Solaire pour l'implantation sur €EPROM

Partie I

- Calcul des angles d'orientations d'un panneau solaire
- Adaptation des angles d'orientations à l'implantation
 - Nécessité de l'approximation
 - Les erreurs

Partie II

- Application des algorithmes d'approximations

Conclusion

PARTIE I

CALCUL D' ANGES D' ORIENTATIONS D'UN PANNEAU SOLAIRE

Introduction :

Dans ce chapitre nous allons voir la commande d'orientation d'un panneau solaire par PC, cette commande consiste au calcul de l'angle optimal d'orientation à l'aide des formules basées sur le principe « maximisation de puissance absorbées ». A cet effet, un programme nommé « **Solangle** » exécuté sur PC, réalisé sous forme visuelle permet cette commande.

Dans ce qui suit, nous allons essayer de traduire les données renvoyer par ce programme, afin de les adapter pour implantation sur EEPROM du micro-contrôleur.

I- Calcul des angles d'orientations d'un panneau solaire :

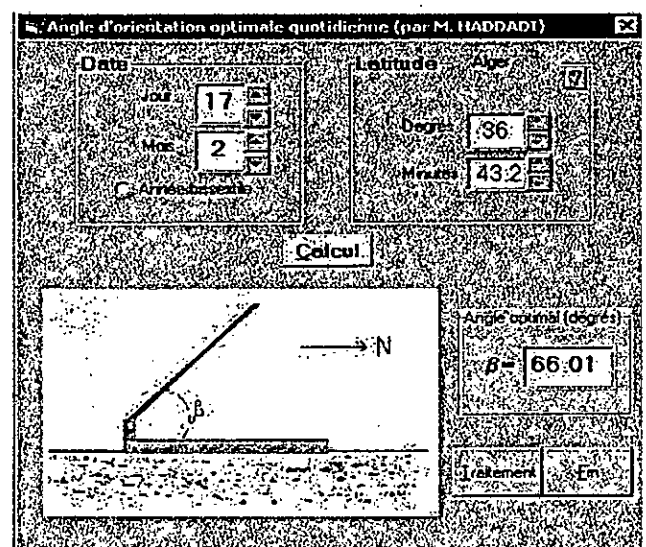
Les angles d'orientations optimales d'un panneau solaire sont définie comme les angle pour lesquels,la puissance absorbée par le panneau solaire est maximale.

La figure ci-dessous représente la première interface d'exécution du programme **Solangle** de monsieur **HADADI** qui permet le calcul des angles optimaux d'orientation d'un panneau solaire.

L'interface comprend des zones de saisit de la latitude et la date, ainsi qu'une liste prédéfinie des viles de l'Algérie avec leurs latitudes correspondantes.

Un bouton **Calcul** permettant l'exécution du programme. Deux autres boutons **Traitement** et **Fin** figurent, un pour quitter le programme, l'autre permet un traitement par Ms Excel. Ce dernier charge des fenêtres d'Excel, contenant les angles d'orientation d'une année entière, logés dans un tableau comme présenté dans la figure *fig.III-2*, une autre fenêtre illustre le graphe correspondant au tableau indiqué.

La figure *fig.III-3* représente la courbe des angles d'orientations optimales de la latitude



FigIII-1: Interface du programme qui calcul les angles d'orientation optimales



36.7° de la ville d'Alger.

Le calcul des angles optimal d'orientations est donnés par la formule :

$$Ang_{optimale} = \xi - Atg(-\pi/180 * tg \xi * tg \delta)$$

Où ξ : la latitude, δ : déclinaison du soleil

La déclinaison du soleil donnée par la formule :

$$\delta = 23.45 \sin[360(284+n)/365]$$

Où n présente la position du jours .

Calcul de l'angle d'orientation optimale
Laboratoire d'Energie Solaire, ENP

Résultats

Latitude 36,72
Année 2001

N°	Date	Angle	Déclinaison
1	01-janv	65,9472427	-23,0326246
2	02-janv	65,8722115	-22,9575494
3	03-janv	65,7893777	-22,8747927
4	04-janv	65,6987269	-22,7843767
5	05-janv	65,6002449	-22,686328
6	06-janv	65,4939186	-22,580677
7	07-janv	65,3797352	-22,4674582
8	08-janv	65,2576829	-22,3467103
9	09-janv	65,1277505	-22,2184759
364	30-déc	66,0739594	-23,1596614
365	31-déc	66,0144867	-23,1

Fig. III-2: une partie du tableau Excel généré par le programme Solangle

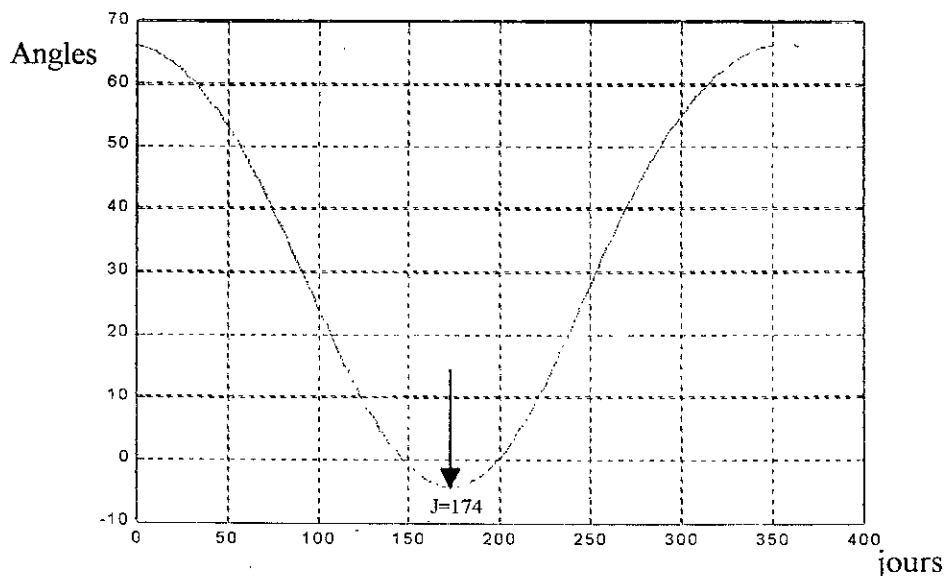
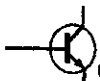


Fig. III-3 Courbe représentant les angles d'orientation d'une année entière de la latitude d'Alger 36.7°



II- Adaptation des angles d'orientations (Approximation):

II-1 Nécessité de l'approximation

▪ L'utilisation de l'expression :

L'expression qui donne les angles d'orientation en fonction de la latitude est une expression contenant des fonctions non intégrées dans le microcontrôleur comme le *Sin* et *tg*.

De telles fonctions ne peuvent être exprimées par l'assembleur 68HC11, pour les utiliser il faut les écrire sous une autre forme de sorte que l'on peut les exprimer par programme. Une solution est généralement adoptée pour ce genre de problème, cette solution consiste à remplacer ces fonctions par leurs développements limités en série de Taylor (DL).

Cette solution est faisable, car les opérations d'addition, soustraction et multiplication ainsi que la division sur 8bit sont disponible sur 68HC11, ce qui fait que les polynômes et les fractions rationnelles seules, peuvent être exprimés par programme assembleur.

L'inconvénient de cette méthode, réside dans le développement limités lui même, car ces DL sont écrits autour d'un point bien spécifier, ce qui fait qu'on ne peut pas faire des calculs avec précision en éloignant du ce point, de plus cette précision dépend de l'ordre du DL, plus l'ordre est grand la précision est bonne, et cela nécessite des multiplication de 32 bit pour seulement un ordre de 3, pour des ordres supérieurs, il faut faire pour chaque résultat de multiplication une factorisation en parties de 8 bits afin de continuer d'utiliser l'instruction MUL (Multiplication de deux nombres de 8 bits) du 68HC11.

De telles opérations augmentent la tailles des programmes et les calculs devient longs est très complexes, et avec une précision médiocre.

▪ Implantation des valeurs des angles d'orientations directe sur EEPROM

le chiffre $65,1277505^\circ$ ou soit encore $65,12^\circ$ à un centième près ne peut être stocké sur la mémoire du microcontrôleur (EEPROM) sur cette forme, mais il doit être codé en binaire de manière à pouvoir reconnaître la virgule, Pour cela il nous faut au moins 2 octets (16 bits) un octet pour la partie entière (8 bit peut stocké les entiers entre 0 à 256) et le deuxième pour la partie décimale, et de cette façon en aura pour chaque jours un angle qui occupe 2 octets de mémoire EEPROM, ce qui fait que pour une latitude donnée : $(365 \text{ jours}) * (2 \text{ octets})$ soit 730 octets. Cela juste pour une seule latitude, si on compte les latitudes de 36° à 22° qui balayent la surface du pays, on aura la totalité de l'espace mémoire nécessaire égale à :

$(365 \text{ jours}) * (2 \text{ octets}) * ((36-22)+1) = 10950 \text{ octets}$ soit environ 10.7 Ko de mémoire EEPROM.



Sachant que le microcontrôleur 68HC811E2 ne dispose que de 2Ko de mémoire EEPROM, ces valeurs ne peuvent être stockées sur cette EEPROM. Une solution est adoptée, et celle d'ajouter une EEPROM extérieure en plus et utiliser le microcontrôleur en mode étendu. Cette solution est faisable car les ports D et C de 16 bits peuvent être utilisés dans ce cas comme bus d'adresses et de données (voir chapitre I) et peuvent adresser jusqu'à 64 ko de mémoire extérieure.

Cette solution n'est pas économique du fait d'utilisation des puces mémoires EEPROM en plus, qui revient chère et encombrant, ainsi que cette perte d'espace en mémoire empêche le microcontrôleur à être utilisé en d'autres applications.

Conclusion :

Les deux solutions apportées, que soit l'implantation de la formule ou les l'implantation des valeurs des angles d'orientations directement en EEPROM, ne sont pas très intéressantes, pour cela on a adopté une solution qui permet à la fois de résoudre le problème d'espace mémoire et la complication des programmes .

II-2 L' Approximation polynomiale :

L'approximation par des polynômes est une des idées les plus anciennes de l'analyse numérique et reste l'une des plus appliquées. Un polynôme $p(x)$ est utilisé comme substitut à une fonction $y(x)$ pour plusieurs raisons. Peut être la plus importante étant que les polynômes sont faciles à calculer puisque n'interviennent que des puissances entières. De plus, leurs dérivées et intégrales peuvent être exprimées sans difficultés et sont également des polynômes. Dans ces conditions, il n'est pas difficile de comprendre pourquoi les polynômes sont considérés comme produit puissant de substitution des fonctions .

Dans notre cas les polynômes à moyen ordre, ou à variable moins de 16 bits, peuvent être programmés sans difficulté en Assembleur, car il ne représente que les multiplications et additions, disponibles sur l'assembleur 68HC11.

II-2-1 Critères d'approximation :

La différence $y(x)-p(x)$ représente l'erreur d'approximation et l'idée centrale est, bien entendu, de maintenir cette différence aussi petite que possible ou tout au moins raisonnablement petite. La simplicité de l'utilisation des polynômes permet d'approcher cet objectif selon des manières ou algorithmes différents, parmi les quelles nous allons examiner, l'interpolation, l'osculation, les moindres carrés, les mini-max et l'utilisation des systèmes surdéterminés et opté pour la méthode la plus adéquate.

**II-2-2 Critères et choix de l'algorithme d'approximation :**

Dans notre cas on est confronté à deux problèmes :

- ❑ Problème d'espace mémoire
- ❑ Problème de difficulté et complexité des calcul pour des nombres représentés sur 16 bits, avec des puissances supérieures à deux.

Et cela tout en gardant la précision avec la moindre l'erreur possible avec le polynôme d'approximation .

Les coefficients des polynômes ne doivent pas dépasser une taille de 16 bit . L'ordre du polynôme d'approximation aussi doit être contrôlé pour ne pas provoquer des calculs lourds et des programmes volumineux.

II-2-3 Méthodes d'approximation :**II-2-3-1 L'interpolation[3] :**

Le polynôme d'interpolation est un polynôme qui coïncide avec $y(x)$ en un certain point, il peut être exprimé sous une variété de formes différentes, toutes équivalentes essentiellement à la formule de *Newton* , mais chacune d'elles étant adaptée à des circonstances différentes.

Lagrange, Taylor, Gauss, Stirling et d'autres noms, ont énoncé plusieurs formules d'interpolations polynomiales, en examinant de près ces formules on distingue la chose suivante :

Toutes les formules d'interpolations donnent des polynômes qui se particularisent par le fait que le degré de ces derniers est dans le même ordre de grandeur, que le nombre de points à approximer. Dans notre cas un nombre de 365 points est approximé par des polynômes de degré de l'ordre de cents est loin d'être utile pour notre étude.

Cette méthode n'est pas adaptée pour notre cas.

II-2-3-2 L'osculatio[n] [3] :

Les polynômes osculatoires ne coïncident pas seulement avec une fonction donnée en un argument spécifié, ce qui est l'idée de base de l'interpolation, mais ont des dérivées jusqu'à un certain ordre qui coïncident également avec des dérivées de la fonction donnée. Ainsi dans le cas d'osculatio[n] la plus simple, nous avons :

$$p(x_i) = y(x_i) \text{ et } p'(x_i) = y'(x_i).$$

pour $i=1..,n$. dans le langage de la géométrie, cela signifie que nos deux courbes sont tangentes en ces $n+1$ points. Des osculation d'ordre supérieur requerraient $p''(x_i) = y''(x_i)$,



et ainsi de suite. Dans de tel cas, nous dirons que les courbes correspondantes possèdent des contacts d'ordre supérieur.

La formule d'*Hermite*, donne le polynôme osculateur sous l'expression :

$$P(x) = \sum_{i=0, n} U_i(x)y_i + \sum_{i=0, n} V_i(x)y_i'$$

avec $U_i(x) = [1 - 2L_i(x_i)(x - x_i)][L_i(x)]^2$ et $V_i(x) = (x - x_i)[L_i(x_i)]^2$

Où $L_i(x)$ est le multiplicateur de *Lagrange*.

Ce polynôme est représenté par au moins $2n + 1$ coefficients, ceci-dit au moins un ordre de deux fois le nombre des points à approximer.

Le même problème rencontré avec l'interpolation se présente ici. On ne peut pas dans ces deux cas contrôler le degré du polynôme d'approximation, cette méthode ne concorde pas donc avec nos objectifs.

Pour soulever ce problème on va voir les autre méthode, mais cette fois avec des algorithmes où on peut contrôler l'ordre des polynômes.

II-2-3-3 Moindre carrée[3] :

L'idée fondamentale qui consiste à choisir une approximation polynomiale $p(x)$ pour une fonction donnée $y(x)$ de manière à minimiser les carrés des erreurs (en un certain sens) a été développée tout d'abord par Gauss. Il existe plusieurs façons de faire qui dépendent de l'ensemble d'arguments impliqués et de la mesure choisie de l'erreur.

Quand les données sont discrètes, nous pouvons minimiser la somme

$$S = \sum_{i=0}^N [y_i - a_0 - a_1x_i - \dots - a_mx_i^m]^2$$

Pour les données spécifiées, x_i , y_i et $m < N$. la condition $m < N$ rend improbable que le polynôme $p(x) = a_0 + a_1x + \dots + a_mx^m$ coïncide en tous les points N données. Aussi, il est probable que S pourra être rendu nul. L'idée de Gauss était de rendre S aussi petit que possible. Les techniques classiques d'analyse conduisent alors à des équations, qui permettent de déterminer les coefficients a_j , ces équations sont

$$\begin{cases} s_0a_0 + s_1a_1 + \dots + s_ma_m = t_0 \\ s_1a_1 + s_2a_2 + \dots + s_{m+1}a_m = t_1 \\ \dots \\ s_ma_0 + s_{m+1}a_1 + \dots + s_{2m}a_m = t_m \end{cases} \dots (I)$$



Où $s_k = \sum_{i=0}^N x_i^k$, $t_k = \sum_{i=0}^N y_i x_i^k$. Ce système d'équations linéaires détermine les a_j de manière

unique et les valeurs de a_j ainsi obtenues rendent la valeur de S aussi minimum que possible.

Algorithme :

L'algorithme de cette méthode consiste à construire une matrice S et un vecteur T , dont les éléments sont définies par la formule de s_k , et t_k .

moindre carré :

lire $x[i, i=0,..n], y[i, i=0,..n], m$

pour $k=0,.. 2m$ faire

$s=0$

$t=0$

pour $i=0,.. m$ faire

$s=s+x(i)^k$

$t=t+y(i)x(i)^k$

Fait

$S(k)=s$

Si $k \leq m$ alors

$T(k)=t$

fin

fait

%construction de la matrice S :

pour $i=0,..m$ faire

pour $j=0,..m$ faire

$S(i, j)=S(j+i)$

fait

fait

% solution de système :

$coeff = inverse(S)*T$

fin.

II-2-2-4 Polynôme Mini-Max[3] :

L'idée de base de l'approximation mini-max à l'aide d'un polynôme peut être illustré dans le cas des données discrète x_i, y_i où $i=1, \dots, N$. soit $p(x)$ un polynôme de degré n ou inférieur, soit $h_i = p(x_i) - y_i$ la différence qu'il présente en nos points donnés, et soit H , la plus grande

**II-2-2-5 Utilisation des systèmes surdéterminés [4]:**

Un systèmes surdéterminés d'équation linéaire prend la forme

$$Va = b \dots \text{ (III)}$$

Où la matrice V possède plus de lignes que de colonnes. Généralement il n'existe pas de vecteurs solution x , aussi l'équation ainsi écrite n'a pas de sens. Les systèmes surdéterminés apparaissent dans le cas de l'approximation par des polynômes où le nombre de points (x_i, y_i) est supérieurs au degré des polynômes. La méthode d'interpolation décrite précédemment consiste à résoudre le système linéaire (I), avec un nombre de points égale au coefficients du polynôme, dans le cas contraire (système III, $N > m$) en aura un système surdéterminé de la forme

$$Va = \begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^m \\ 1 & x_2 & x_2^2 & \dots & x_2^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \dots & x_{n-1}^m \\ 1 & x_n & x_n^2 & \dots & x_n^m \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_m \end{pmatrix}$$

la matrice $V(N, m)$ est nommée matrice de Vandermonde, cette matrice définit la fonction $g(i, j) = x_i^{j-1}$ $i = 1, \dots, N, j = 1, \dots, m$, le système tel qu'il est ne peut être résolu, car on ne peut inverser la matrice de V , (matrice $N \times m$).

solution des systèmes surdéterminer :

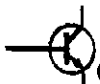
il existe plusieurs méthodes pour la résolution de tel système, on propose ici une de ces méthodes. Pour ça on donne un exemple, pour bien comprendre la méthode.

Soit le système $Ax=b$ où A est une matrice 3×2 .

$$\begin{cases} 3x + 5y = 13 \\ -x + 3y = 2 \\ x - 4y = -7 \end{cases} \implies A = \begin{pmatrix} 3 & 5 \\ -1 & 3 \\ 1 & -4 \end{pmatrix}$$

Soit tA la transposé de A . La matrice produit ${}^tA * A$ est donnée par :

$${}^tA * A = \begin{pmatrix} 3 & -1 & 1 \\ 5 & 3 & -4 \end{pmatrix} \begin{pmatrix} 3 & 5 \\ -1 & 3 \\ 1 & -4 \end{pmatrix} = \begin{pmatrix} 11 & 16 \\ 16 & 40 \end{pmatrix}$$



Le résultat est une matrice carrée, si ${}^tA^*A$ est inversible la solution de $Ax=b$ est donnée par :

$$({}^tA^*A)X = {}^tA^*b \implies X = \text{inverse}({}^tA^*A) * {}^tA^*b$$

La solution sera directement déduit :

$$({}^tA^*A)^{-1} * {}^tA^*b = \begin{pmatrix} 11 & 16 \\ 16 & 40 \end{pmatrix}^{-1} \begin{pmatrix} 3 & -1 & 1 \\ 5 & 3 & -4 \end{pmatrix} \begin{pmatrix} 13 \\ 2 \\ -7 \end{pmatrix} = \begin{pmatrix} 11 & 16 \\ 16 & 40 \end{pmatrix}^{-1} \begin{pmatrix} 44 \\ 43 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Cas général

Dans le cas général le système (I) peut être transformé en un système d'équations linéaire carré par la multiplication des deux côté de (I) par la transposé de la matrice de Vandermonde, en aura ainsi la matrice carrée ${}^tV^*V$. mais le problème n'est pas encore résolu, dans l'exemple, précédent on a inversé ${}^tA^*A$, la question qui se pose « petons inverser la matrice ${}^tV^*V$? » pour répondre à cette question nous allons voir le théorème sur les déterminant des matrice particulières :

Théorème : [5]

« Si A est une matrice n^*m à coefficients dans \mathfrak{R} $n > m$ alors ;

$$\det({}^tA^*A) = \sum M^2$$

où M parcourt tous les mineurs $\begin{pmatrix} m \\ n \end{pmatrix}$ d'ordre m de la matrice A »

on applique ce théorème a notre cas en peut avoir les résultats suivants :

la matrice ${}^tV^*V$ est inversible seulement et seulement si $\det({}^tV^*V)$ soit différent de zéro donc il suffit d'avoir un mineur non nulle pour remplir la condition, et puisque la matrice V contient des vecteurs linéairement indépendant, (caractéristique des polynômes puisque $(x_i, y_i) \neq (x_j, y_j)$ pour $j \neq i$), le déterminant de ${}^tV^*V$ ne peut être nulle, d'ou la possibilité d'inversement.

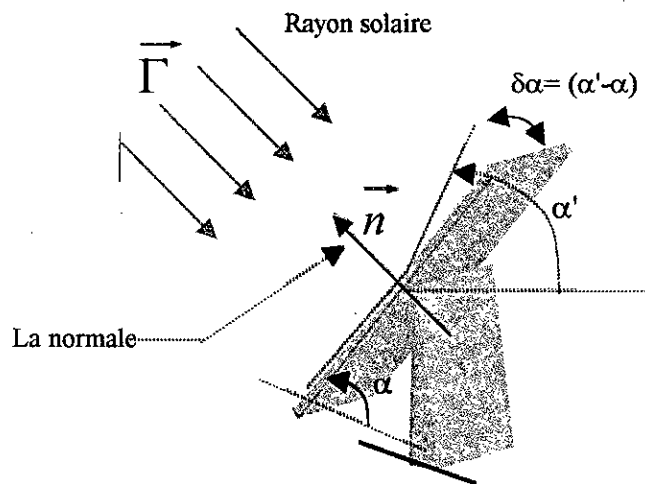
III- Les erreurs :

Dans les paragraphes ci-dessus, on parlait de l'approximation, des critères et choix des l'algorithmes les mieux adaptés pour notre étude. L'introduction de l'erreur d'approximation



n'est encore pas mit en évidence jusque ici. La précision d'une approximation est un facteur très important, qui doit être considéré et manipulé avec une grand soin .

Donc la première des choses à définir dans de tel cas, est bien entendu, la précision, est cela revient à poser la question suivante « à quelle précision peut on apprécie une approximation ? ». autrement dit, quelle est l'erreur permise dans une pareille opération. La figure (III-4) donne un schéma qui peut nous aider à répondre à cette question.



FigIII-4: Erreur d'orientation d'un panneau solaire

L'angle d'orientation optimale α est définie de manière à avoir la perpendicularité entre la surface du panneau solaire, et les rayons solaire, afin d'avoir un flux maximale de puissance.

On désigne par $\vec{\Gamma}$ le champs de la lumière du jour, et par \vec{n} le vecteur normale à la surface du panneau solaire. La condition « flux maximal » ce traduit par l'écriture mathématique :

$$\frac{\vec{\Gamma} \cdot \vec{n}}{\|\vec{\Gamma}\| \|\vec{n}\|} = 1$$

cette écriture a la même signification que $\cos(\vec{\Gamma}, \vec{n}) = 1$.

une erreur d'orientation d'un angle α' va donner naissance à un vecteur \vec{n}' perpendiculaire à la surface du panneau et non colinéaire avec les rayons solaire. On aura donc un rapport produit scalaire sur module des vecteur inférieure à 1, le flux de puissance va donc diminuer d'un rapport égale à : $\cos(\alpha - \alpha')$ avec

$$\cos(\alpha - \alpha') = \cos(\vec{\Gamma}, \vec{n}') < 1$$



Si on calcul la différence de flux du passage d'un angle optimal α à un angle α' , la variation de ce dernier sera donné par :

$$\text{flux}_{\text{optimal}}(\alpha) - \text{flux}(\alpha') = \text{flux}_{\text{optimal}} (1 - \cos(\alpha - \alpha'))$$

ce qui signifie que, une erreur d'orientation du panneau de $(\alpha - \alpha')^\circ$, va causé une diminution du flux de puissance avec un rapport de $[1 - \cos(\alpha - \alpha')]$. En fait cette valeur traduit l'erreur relative commise sur la puissance, incitée par une erreur d'orientation de $(\alpha - \alpha')^\circ$.

Si on note α_{err} la quantité $\alpha - \alpha'$, (l'erreur d'orientation), alors l'erreur relative commise sur le flux de puissance est :

$$\text{Err}_{\text{relative}} = (1 - \cos(\alpha_{\text{err}}))$$

Cette relation nous permet de voir quelle est l'erreur sur l'angle d'orientation qu'on peut se permettre, on se basons sur l'erreur relative de puissance. Par exemple, si on peut admettre une erreur relative (perte) de puissance de un millième 1/1000, l'erreur commise sur l'orientation du panneau est :

$$\text{ArcCos}(1 - \text{Err}_{\text{relative}}) = \text{ArcCos}(1 - 1/1000) = 0.9991^\circ$$

D'une autre façon, une erreur d'orientation d'environ 1° , va causer une perte de 1/1000 de puissance absorbée.

Conclusion :

Un intervalle d'erreur de $\pm 1^\circ$ sur l'orientation est suffisamment tolérable. On peut donc définir l'erreur admissible par l'approximation polynomiale par $\pm 1^\circ$.

PARTIE II

APPLICATION DES ALGORITHMES D'APPROXIMATIONS

Introduction :

Dans cette section nous allons entamer la partie programmation. Les différentes méthodes que nous avons présenté dans la partie précédente, seront traduits en langage Matlab afin de tirer profit de ce dernier, le choix du langage Matlab est bien justifier, ces ressource et ça souplesse en terme de matrices et vecteurs, ainsi que ça prise en charge des interfaces graphique et boites dialogues, le rendit parmi les langage le plus utilisé dans le domaine d'ingénierie.

I- Application des algorithmes d'approximations :

I-1 Programme d'approximation *APPROXIM (,)*:

Les algorithmes d'approximations que nous avons choisi, pour la programmation, dans le but d'approximer les données, angles d'orientations, on était sélectionné selon des critères. En effet ces critères expriment deux choses, une erreur minimale, et possibilité de contrôler le degré du polynôme employé dans l'approximation, ces méthodes ainsi sélectionnées, nous les avons traduit vers le langage *Matlab*, comme déjà mentionné, sous forme de sous-programmes, appartenant à un programme principal nommé *APPROXIM (,)*.

Le programme principal est on réalité, conçu sous forme d'une fonction, elle-même fait appel à d'autres fonctions constituant les trois méthodes ; moindre carrée, mini-max et système surdéterminé. Nous avons aussi associé aux programmes, des outils de dialogues avec l'utilisateur. L'utilisateur peut choisir la méthode d'approximation, le degré du polynôme d'approximation, il peut choisir aussi de voir les différentes courbes ; erreur, courbes réelle courbe polynomiale, en plus des boites de dialogues et figures, des message d'erreur complète ces derniers afin d'assurer une bonne utilisation des programme .

Le programme *APPROXIM (,)*, peut être exécuté a partir de la fenêtre de commande Matlab, ou appelé d'un programme Matlab, les données à approximer doivent être placées comme vecteurs x, y sous la syntaxe *APPROXIM (x ,y)*.

Après la saisie des vecteurs x et y et validation d'exécution, une boîte de dialogue s'affiche, contenant trois boutons permettant le choix de la méthode d'approximation. Voir Fig. III-5



Fig III-5 : Boite de dialogue pour le choix de la méthode

Une fois un choix effectué, une autre boîte de dialogue s'affiche afin de permettre à l'utilisateur la saisie d'un entier « n » déterminant le degré du polynôme d'approximation, par défaut : un deux « 2 » est affiché à l'apparition de la boîte, l'entier saisi peut prendre n'importe quelle valeur entière supérieure ou égale à « un », Voir Fig. III-6.

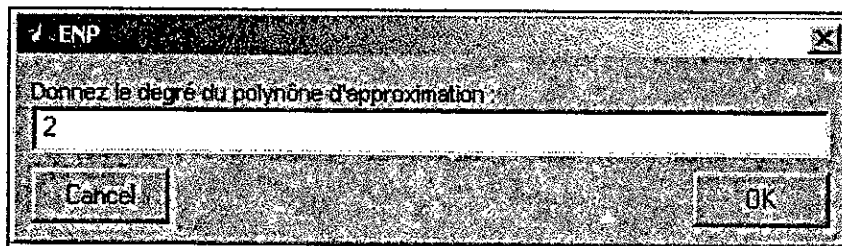
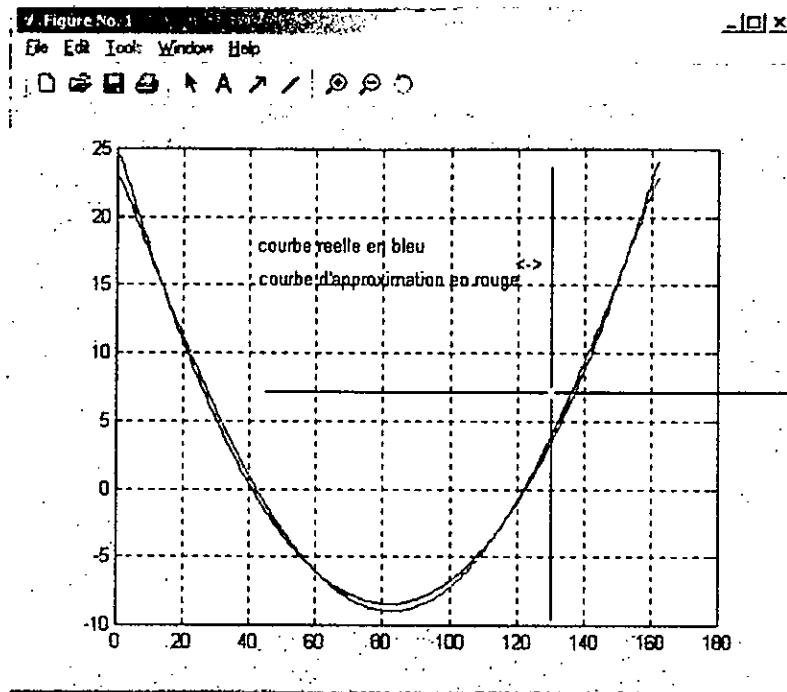
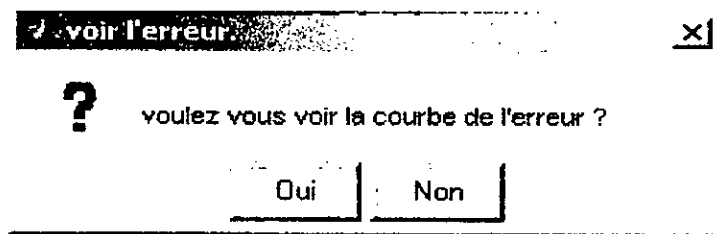


Fig III-6 : Boite de dialogue pour le degré du polynôme d'approximation

A ce niveau le programme *APPROXIM* renvoie les coefficients du polynôme de degré n substituant les données x, y , et sauvegarde ces coefficients dans un répertoire créé dans la racine « C:\Res », sous forme d'un fichier texte « *Resultat.txt* », juste après une fenêtre s'imprime sur écran montrant la courbe représentant à la fois les données (x, y) en bleu, et la courbe du polynôme d'approximation en rouge, Voir Fig. III-7. L'utilisateur peut constater aussi l'apparition de deux lignes accompagnant le pointeur de la souris, en cliquant sur le bouton gauche, un texte s'affiche « *la courbe réelle est en bleue, courbe d'approximation en rouge* », un autre clic fait apparaître une autre boîte de dialogue, cette dernière demande la confirmation pour l'affichage de la figure 'courbe d'erreur' commise avec l'approximation Voir Fig. III-8.

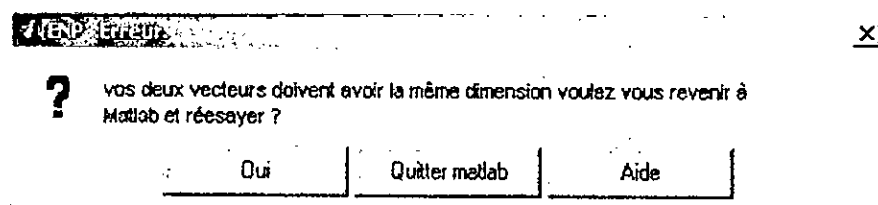


FigIII-7 : Figure affichée par programme représentant les courbe réelle et approximée.



FigIII-8 : Boite de dialogue « Voir l'erreur »

Une confirmation de Oui, réaffiche sur écran la courbe d'erreur commise par l'approximation. Un fichier d'aide est fourni avec le programme *APPROXIM*, ce fichier texte est prévu pour d'éventuelles erreurs, que se soit de manipulation de programme, mauvaise syntaxe ou encore des entrées de données erronées. La figure *Fig. III-9* présente la boite de dialogue affichée dans une telle situation.



FigII-9 : boite de dialogue « fausse manipulation »

**I-2 Approximation de la courbe de la latitude d' Alger 36.7° :**

La figure *Fig. III-3* donne les angles d'orientation de la latitude d'Alger 36.7°. Pour approximer cette courbe on utilise une des méthodes du programme *APPROXIM*.

Soit [angle(i), i=1,365] le vecteur contenant les angles qui correspond à cette courbe.

Tout d'abord nous allons procéder par une décomposition de la courbe en intervalles car l'approximation de la courbe toute entière va conduire à des erreurs importantes, pour les diminuer il faut augmenter l'ordre du polynôme, et cela va nous amener par ailleurs à manipuler des nombres au delà de 16 bits ce qui rend difficile la programmation en *assembleur*. En effet ces intervalles sont choisis de manière à avoir une fonction d'erreur minimales avec des polynômes de second ordre.

Les intervalles choisis sont :

J=1, 70

J=70, 110

J=110, 145

J=145, 174

J=174, 200

J=200, 230

J=230, 265

J=265, 290

J=290, 365

Avec la méthode *surdet* on va essayer d'approximer la première partie pour $j=1,70$

Après exécution du programme, on reçoit les coefficients du polynôme $p(x)$ ainsi que l'erreur commise. Les coefficients fournis du polynôme sont :

$$[-0.0038 \quad -0.0702 \quad 66.0668]$$

soit

$$p(x) = -0.0038x^2 - 0.0702x + 66.0668$$

la figure *Fig. III-10* représente simultanément la courbe réelle $y(x)$ et la courbe $p(x)$;

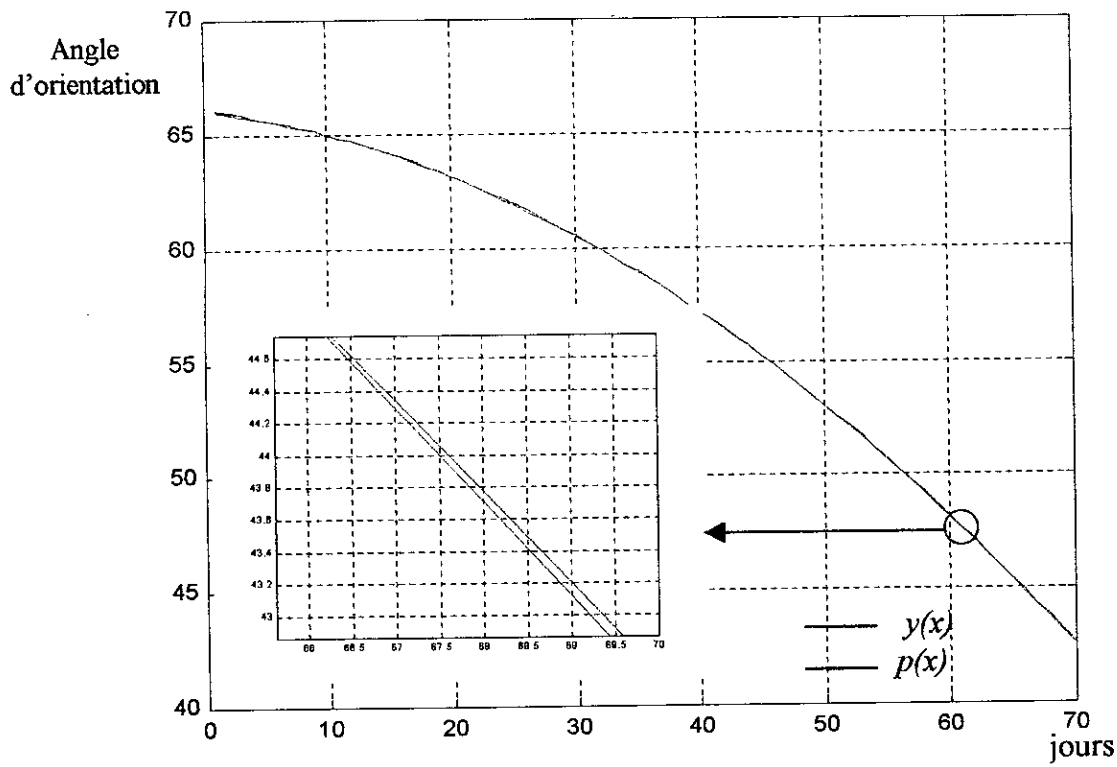


Fig III-10 : courbes $p(x)$ et $y(x)$

L'erreur (en degré) commise est représentée par la figure -FigIII-12 :

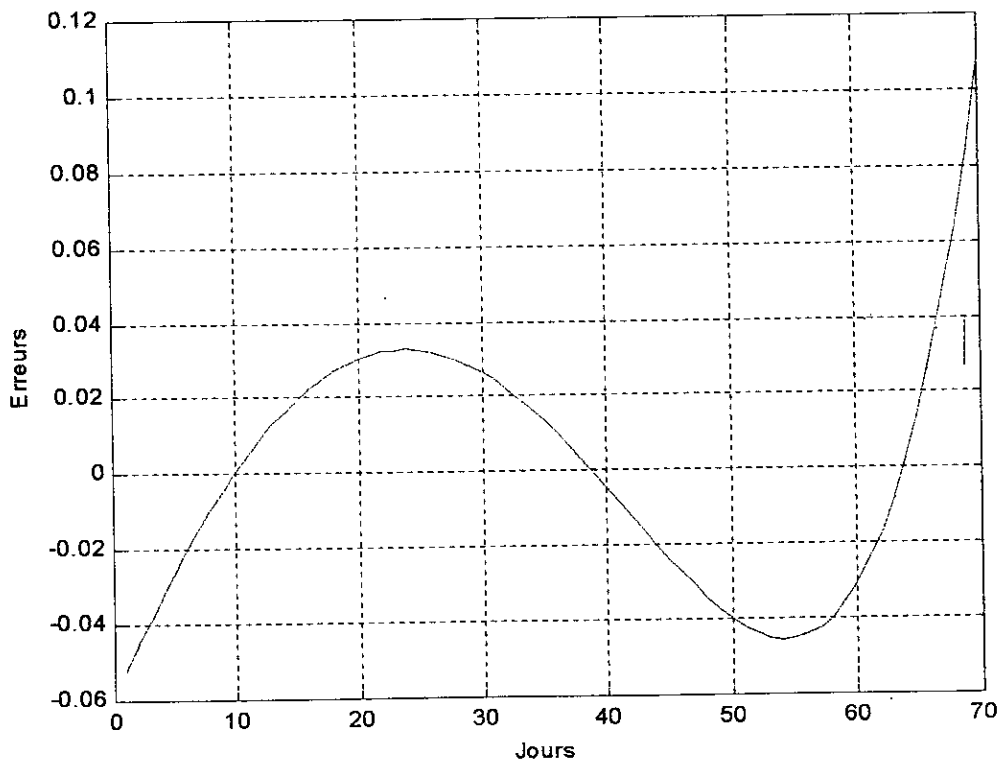
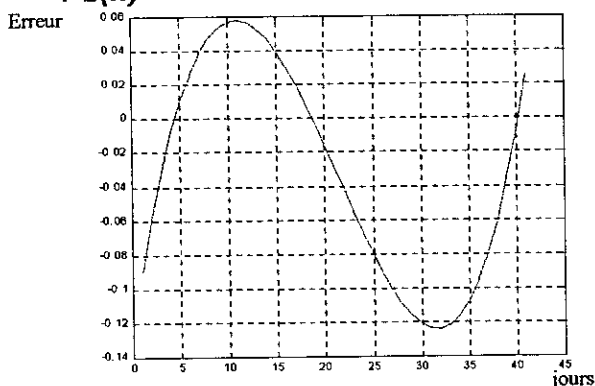


Fig III-11 : Erreur commise par l'approximation

L'erreur commise par l'approximation $p(x)$ est comprise entre $[-0.06, 0.1]$ environ, ce qui donne une erreur relative variant entre 10^{-3} à $2,2 \cdot 10^{-3}$. On peut dire que l'approximation des données (x_i, y_i) est très bonne.

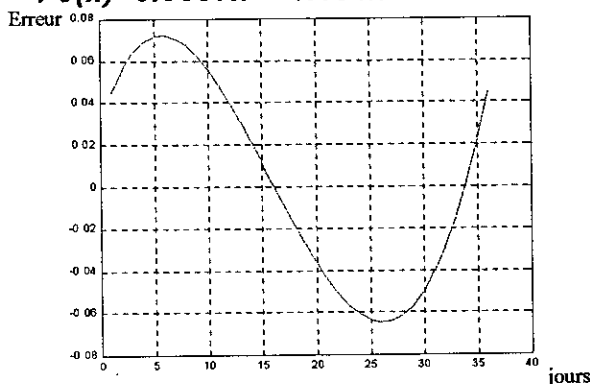
On va procéder de la même manière pour toutes les parties qui restent, les figures ci-après donnent les polynômes de chaque approximations ainsi que les erreurs associées.

$$P2(x) = -0.0004x^2 - 05552x + 83.5434$$



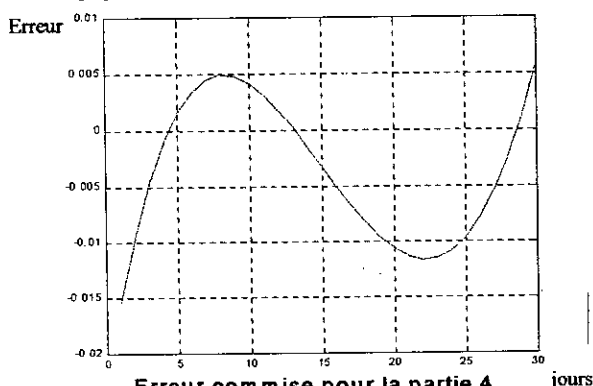
Erreur commise pour la partie 2

$$P3(x) = 0.0037x^2 - 1.434x + 130.5827$$



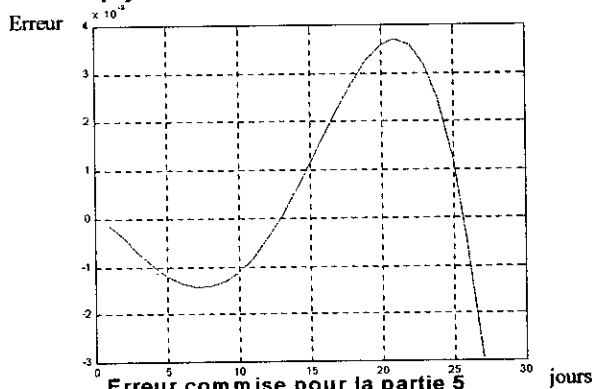
Erreur commise pour la partie 3

$$P4(x) = 0.0059x^2 - 2.054x + 174.2884$$



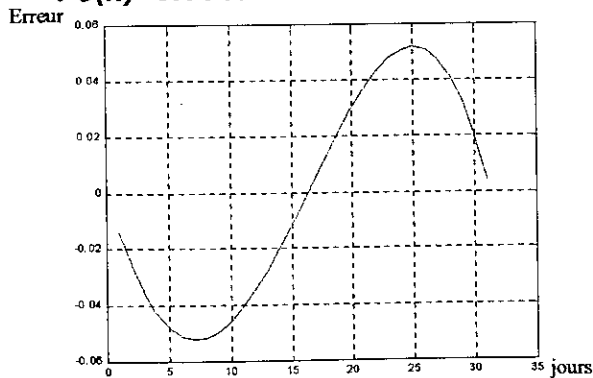
Erreur commise pour la partie 4

$$P5(x) = 0.0063x^2 - 2.188x + 185.5$$



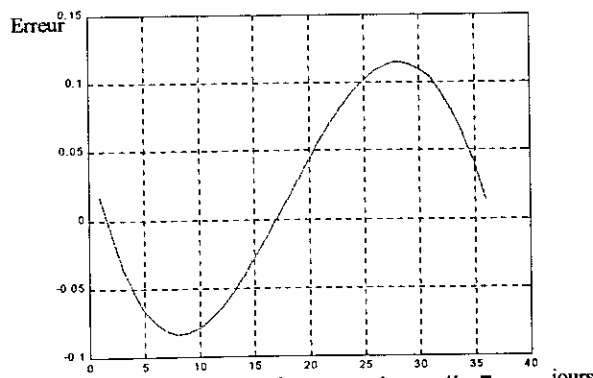
Erreur commise pour la partie 5

$$P6(x) = 0.0047x^2 - 1.5375x + 119.411$$

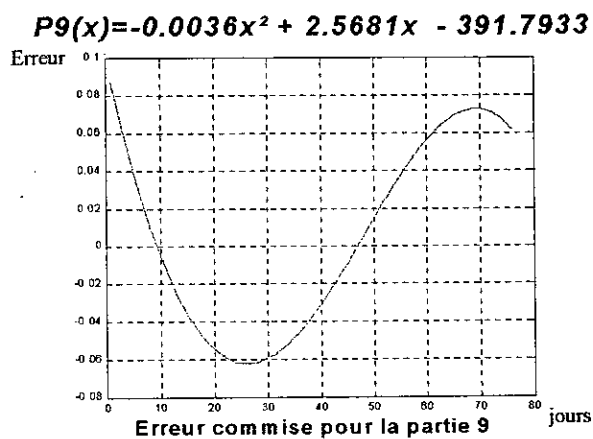
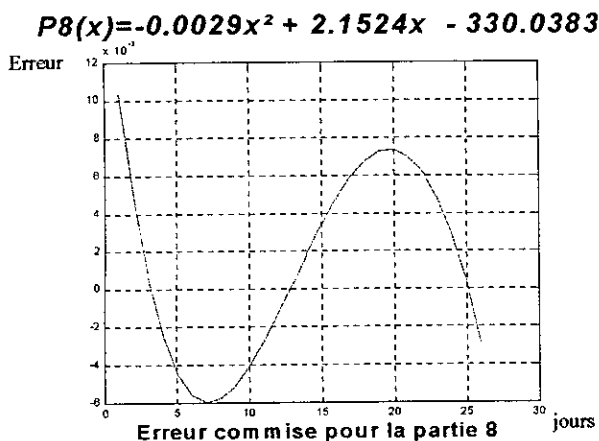


Erreur commise pour la partie 6

$$P7(x) = 0.6373x - 132.1762$$



Erreur commise pour la partie 7



Résultats :

Les polynômes d'approximation des angles d'orientation de la latitude d'Alger sont :

$P1(x) = -0.0038x^2 - 0.0702x + 66.0668$	$x = 1, 70$
$P2(x) = -0.0004x^2 - 0.5552x + 83.5434$	$x = 70, 110$
$P3(x) = 0.0037x^2 - 1.434x + 130.5827$	$x = 110, 145$
$P4(x) = 0.0059x^2 - 2.054x + 174.2884$	$x = 145, 174$
$P5(x) = 0.0063x^2 - 2.188x + 185.5$	$x = 174, 200$
$P6(x) = 0.0047x^2 - 1.5375x + 119.411$	$x = 200, 230$
$P7(x) = 0.6373x - 132.1762$	$x = 230, 265$
$P8(x) = -0.0029x^2 + 2.1524x - 330.0383$	$x = 265, 290$
$P9(x) = -0.0036x^2 + 2.5681x - 391.7933$	$x = 290, 365$

I-4 Approximation des courbes des latitudes comprises entre 36° et 22° :

Dans la section précédente on a approximé la courbe qui correspond à la latitude d'Alger 36.7° avec neuf polynômes, ces polynômes ont donné une très bonne précision. Il nous reste encore les latitudes couvrant le territoire Algérien, à partir de 36° à 22° extrême Sud (Tamanrasset), pour cela on va utiliser la même méthode. Mais cette fois on est confronté au problème de grand nombre de polynômes fourni pour chaque courbe, si on a neuf polynômes en moyenne pour chaque latitude, on aura alors (9*14), soient 126 polynômes distincts.

Afin d'éviter ce problème et de minimiser le nombre de polynômes on propose la solution suivante :

On cherche des relations entre les courbes d'angles d'orientation des différentes latitudes avec une courbe d'angle d'orientation qu'on appellera *la référence*. Une fois ces relations établies, toutes les courbes des différentes latitudes seront déduites à partir de la *référence*.

I-4-2 Construction des relations :

Pour rechercher des relations entre les courbes de différentes latitudes et la référence nous allons tout d'abord définir les ensembles suivants :

- Ang36.7 l'ensemble des valeurs des angles d'orientation qui correspond à la latitude d'Alger (36.7°), en quelque sorte, Ang36.7 constitue un vecteur de 365 éléments contenant les angles d'orientations de cette latitude.
- Ang36, Ang35, à Ang22, les ensembles des valeurs des angles d'orientation qui correspondent aux latitudes 36°, 35° à 22° respectivement. Ces ensembles constituent des vecteurs de 365 éléments.

On définit l'ensemble Ang36.7 comme l'ensemble de référence (*la référence*). Et on va construire des applications notées G_i , ces applications sont définies dans l'ensemble Ang36.7 vers les ensembles Ang36, Ang35, ... Ang22 respectivement, de façon que chaque G_i est l'application qui va faire correspondre chaque valeur du vecteurs Ang36.7 (d'un jour j) avec la valeur du même indice j (même jour) d'une autre latitude suivant l'indice de l'application G .

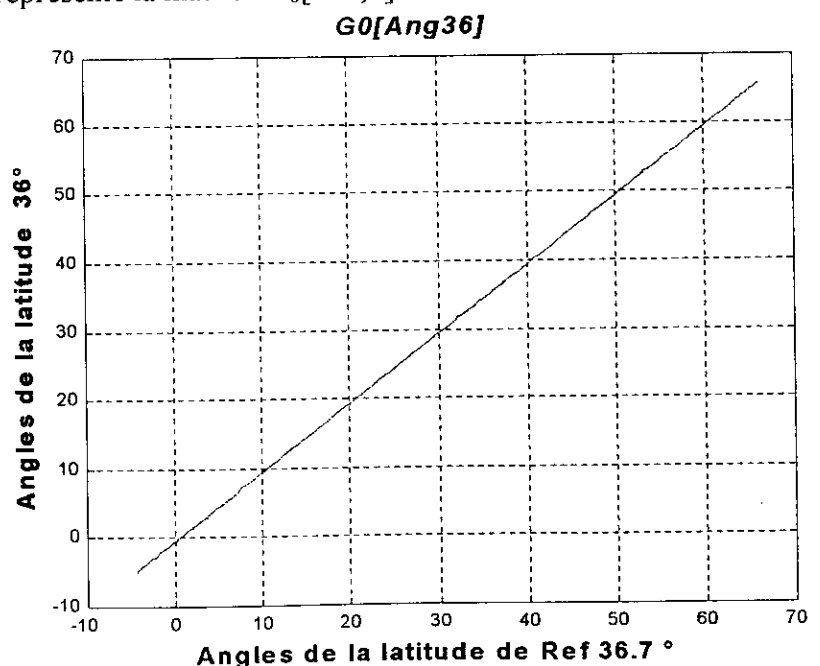
Soit G_0 l'application définie de la référence vers Ang36, donc on peut écrire :

$$\begin{aligned}
 G_0 : \text{Ang36.7} &\longrightarrow \text{Ang36} \\
 \text{Ang36.7}(j) &\longrightarrow \text{Ang36}(j) ; j=1 \text{ à } 365
 \end{aligned}$$

Définie ainsi, l'application G_0 sera représentée par une matrice de deux colonnes et 365 lignes, soit $G_0[365,2]$.

La figure Fig. III-21 est la courbe qui représente la matrice $G_0[365,2]$.

FigIII-21 : courbe $G_0[\text{Ang36}(j)]$



I-4-3 Substitution de la matrice $G_0[365,2]$:

L'application G_0 est représentée par la matrice $G_0[365,2]$, les éléments de cette matrice peuvent être substitués ou approximatés par un polynôme P_g , ce polynôme est défini de la manière :

$$P_g[Ang36(j)] = Ang36.7(j) \quad j=1,365$$

L'utilisation du programme d'approximation $Approxim(x=Ang36(j), y=Ang36.7(j))$ donne le résultat :

$$P_g(x) = 0.0001x^2 + 0.9907x - 0.5437 \quad x \in Ang36$$

Avec une autre écriture :

Pour $j=1,365$

$$Ang36(j) \approx P_g[Ang36.7(j)] = 0.0001[Ang36.7(j)]^2 + 0.9907[Ang36.7(j)] - 0.5437$$

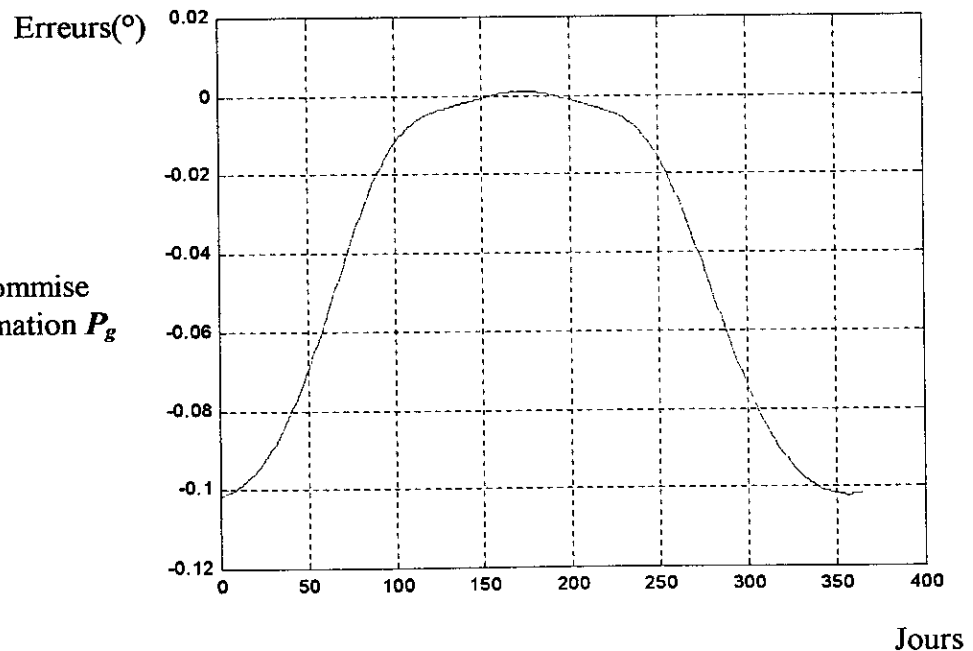
Le polynôme P_g nous permet donc d'avoir tout les angles de la latitude 36° à partir de ceux de la référence .

I-4-4 Erreur commise avec l'approximation P_g :

L'erreur commise avec l'approximation P_g est calculée par la différence ;

$$j=1,365 \quad Erreur(j) = Ang36(j) - P_g[Ang36.7(j)]$$

La Fig.III-22 représente l'erreur (en degré) de l'approximation.



FigIII-14 : Erreur commise avec l'approximation P_g

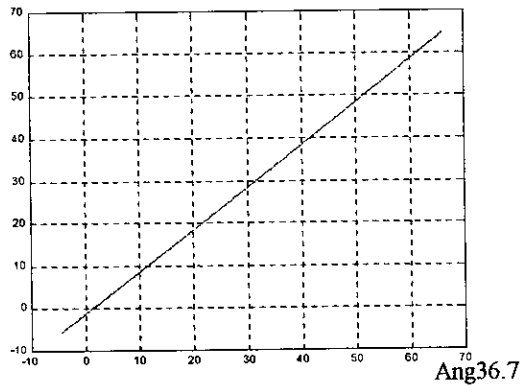
Résultat :

Le polynôme P_g donne une relation liant Ang36 avec la référence, ce qui nous permet d'en déduire Ang36 à partir de la référence. L'erreur commise est très faible, on peut dire que l'approximation de G_θ nous a fourni une relation assez précise entre la référence et la latitude 36°.

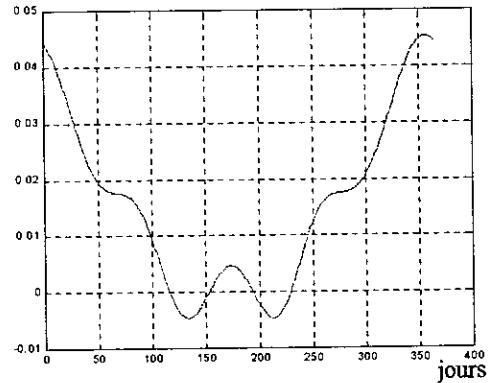
I-4-5 Substitution des matrices d'applications G_i :

Les applications G_i , définies à partir de la référence vers les ensembles des latitudes 36° à 22°, ci-avant exposées ; peuvent être également représentées sous forme matricielle $G_i[365,2]$, ces matrices sont soumises au même algorithme d'approximation, les résultats sont données en figures ci-après.

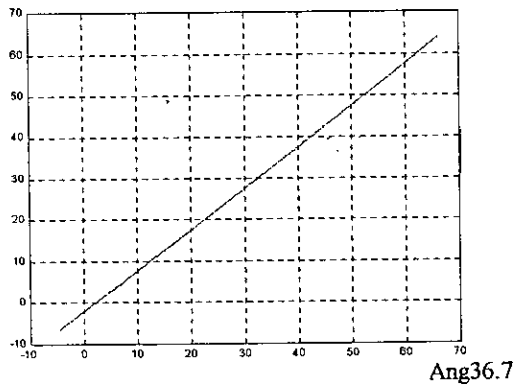
Ang35, $P(x)=0.0003x^2 + 0.9782x - 1.3073$



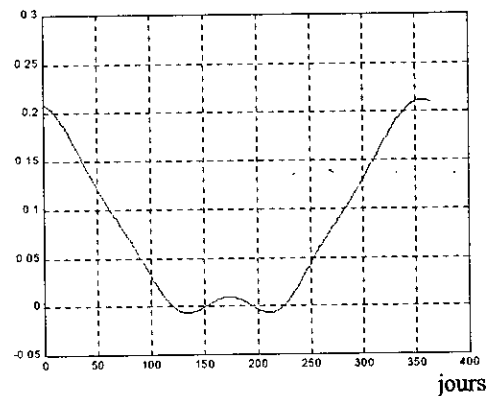
Err35, $P(x)=0.0005x^2 + 0.9661x - 2.08$



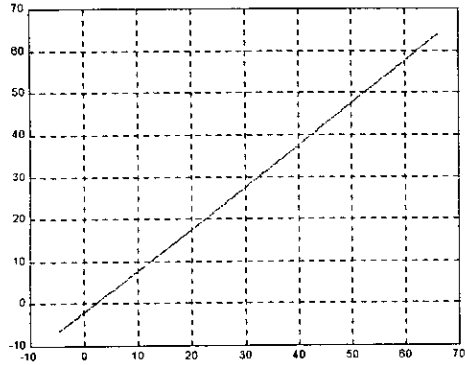
Ang34, $P(x)= 0.0005x^2 + 0.9661x - 2.08$



Err34,

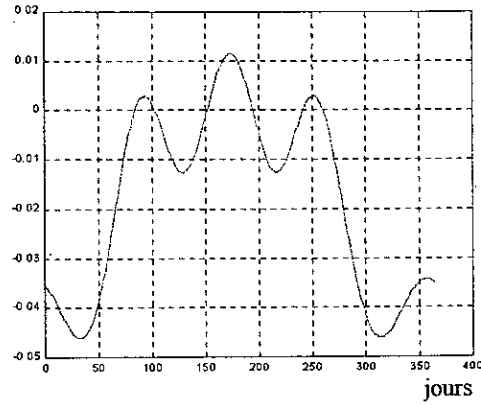


Ang33, $P(x) = 0.0006x^2 + 0.9545x - 2.8613$

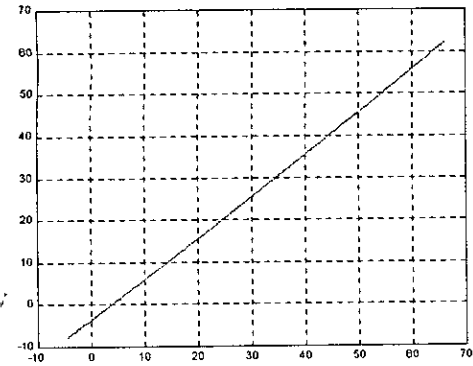


Ang36.7

ERR33

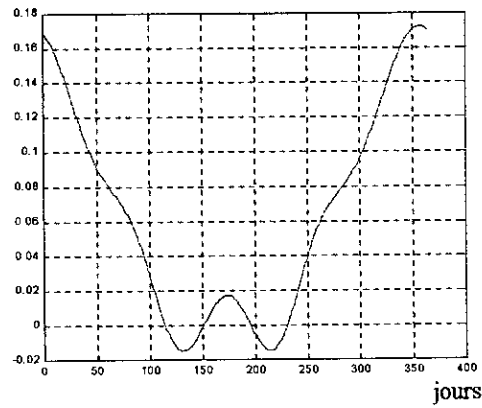


Ang32, $P(x) = 0.0008x^2 + 0.9432x - 3.6504$

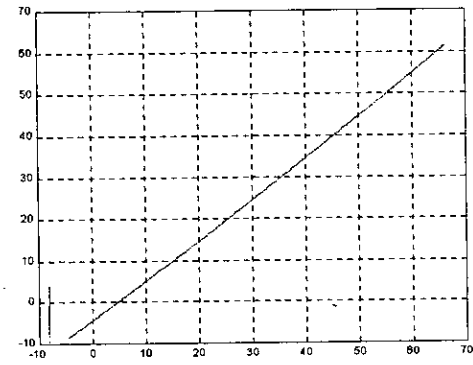


Ang36.7

Err32

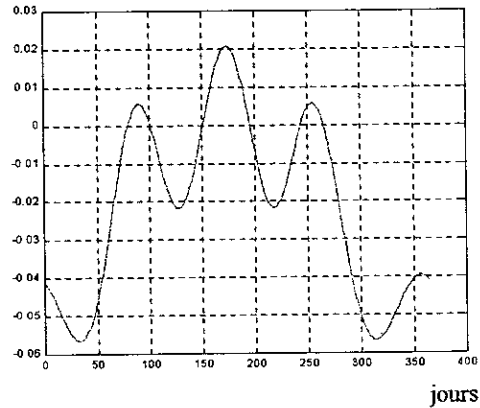


Ang31, $P(x) = 0.0009x^2 + 0.9323x - 4.4469$

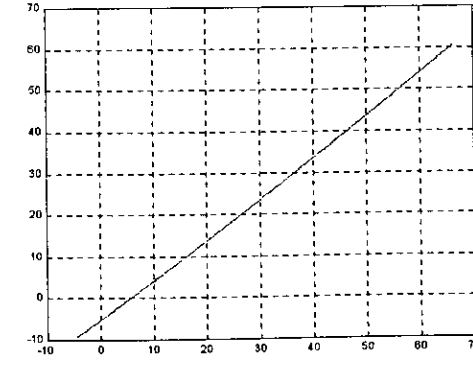


Ang36.7

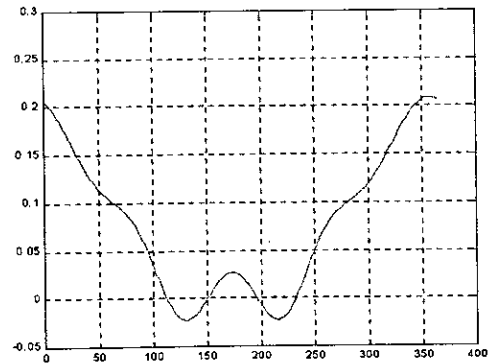
Err31



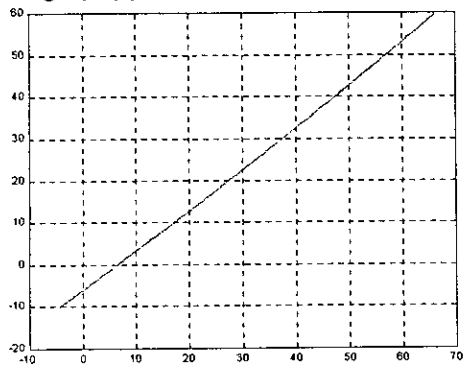
Ang30, $P(x) = 0.0011x^2 + 0.9218x - 5.2502$



Err30

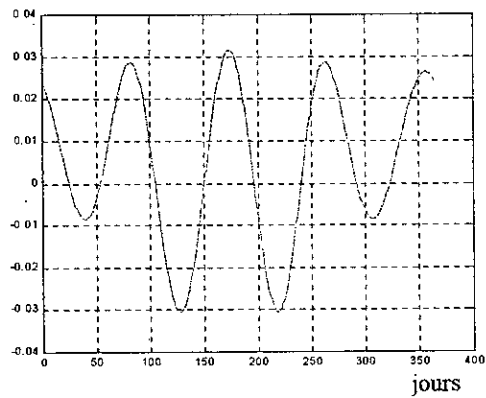


Ang29, $P(x) = 0.0012x^2 + 0.9115x - 6.0599$

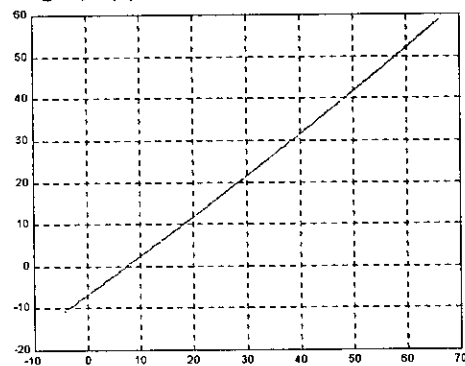


Ang36.7

Err29

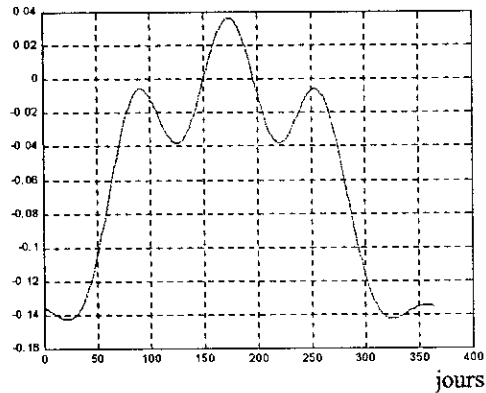


Ang28, $P(x) = 0.0013x^2 + 0.9016x - 6.8756$

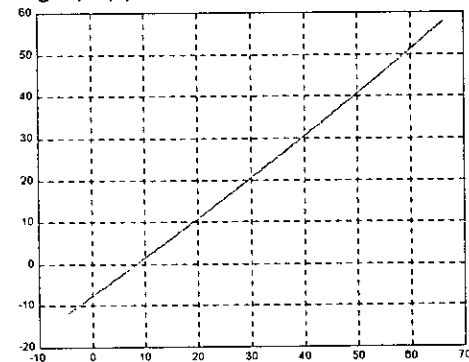


Ang36.7

Err28

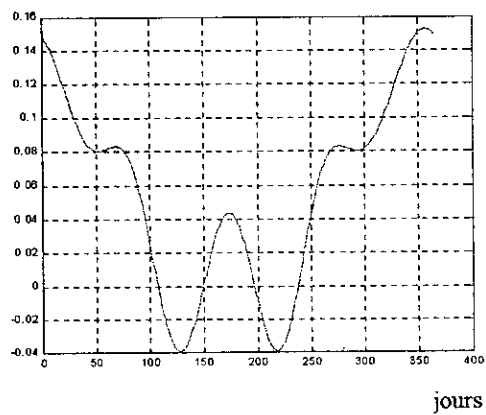


Ang27, $P(x) = 0.0015x^2 + 0.8919x - 7.6968$

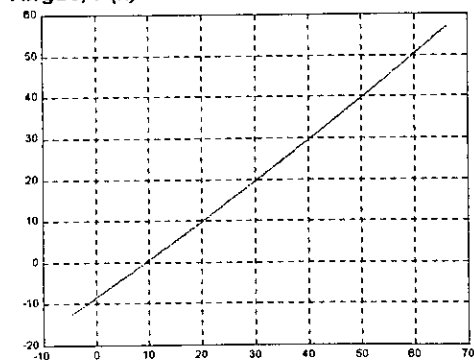


Ang36.7

Err27

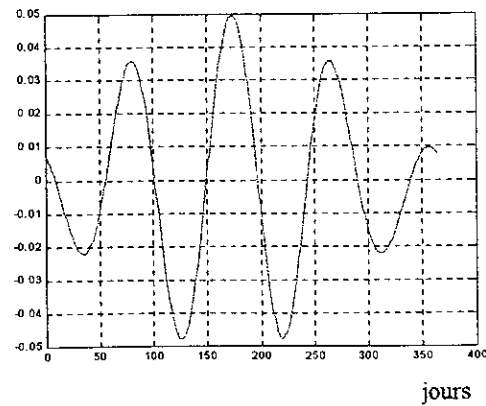


Ang26, $P(x) = 0.0016x^2 + 0.8824x - 8.5233$

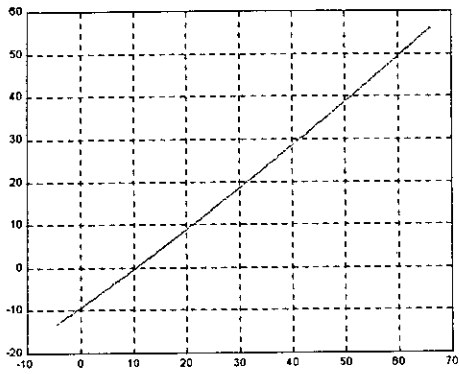


Ang36.7

Err26

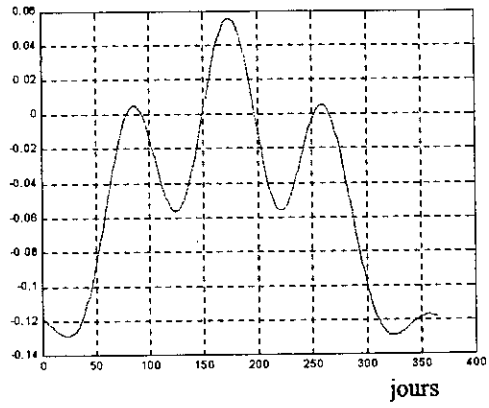


Ang25, $P(x) = 0.0017x^2 + 0.8732x - 9.3546$

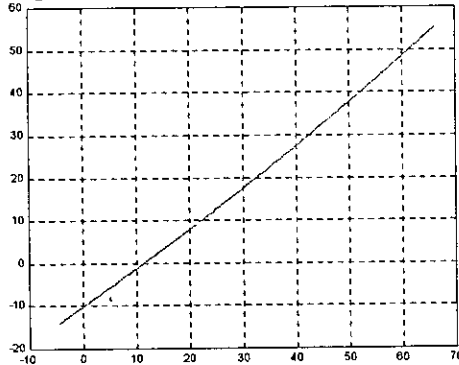


Ang36.7

Err25

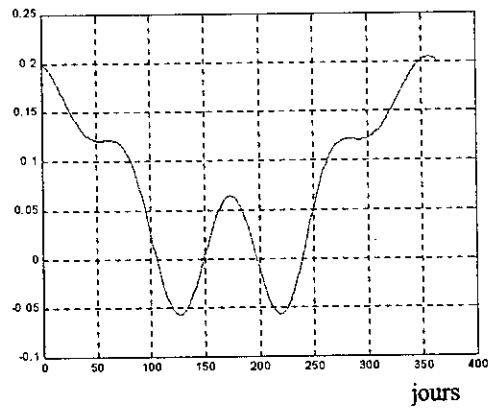


Ang24, $P(x) = 0.0019x^2 + 0.8642x - 10.1905$

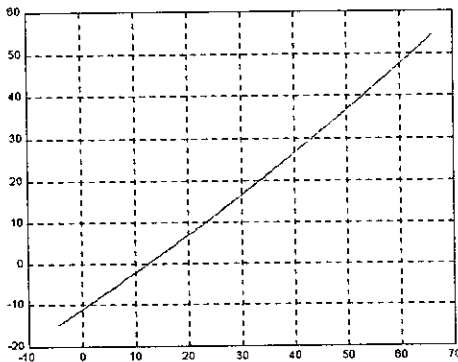


Ang36.7

Err24

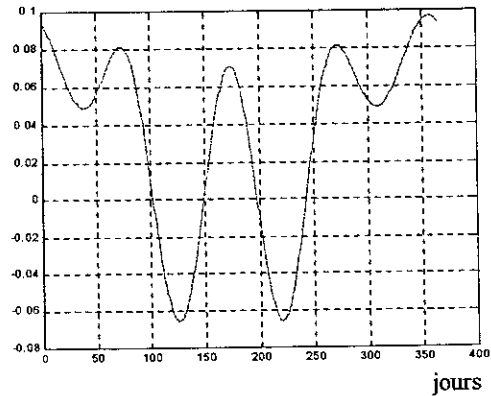


Ang23, $P(x) = 0.002x^2 + 0.8554x - 11.0307$

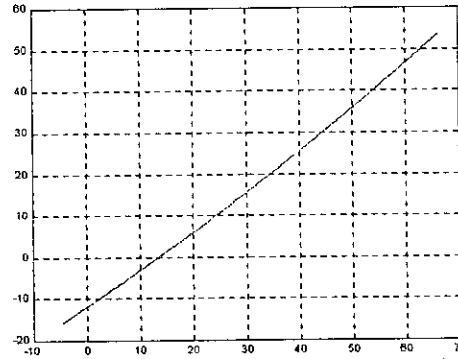


Ang36.7

Err23

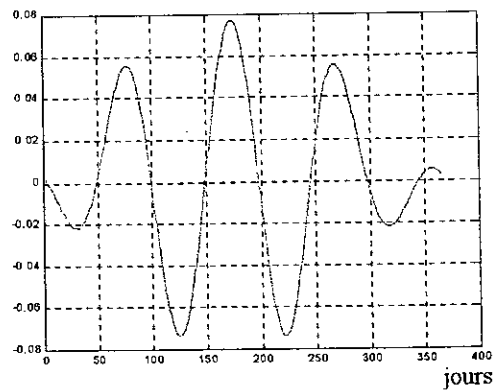


Ang22, $P(x) = 0.0021x^2 + 0.8469x - 11.8750$



Ang36.7

Err22





II- Construction de l'application G_s relative au deux partie Pseudo-symétrique de la référence :

On examinant de près l'aspect de la figure de référence (fig.III-3), la première chose à constater, est la l'allure symétrique de cette dernière par rapport à la droite verticale passante par le point $j=174$. On peut penser qu'il peut y avoir une relation liant ces deux partie. Pour cela on va utiliser les mêmes démarches employées pour la construction des applications G_i .

Donc on peut définir l'application G_s dans l'ensemble $\{Ang36.7(j), j=1, 174 \text{ et } j=357, 365\}$, vert l'ensemble $\{Ang36.7(j), j=174, 365-8\}$, de manière que :

$$G_s(Ang36.7(j))=Ang36.7(365-8-j), j=1,174+9$$

Les chiffres +9 et -8 sont introduites avec l'indice j , dans le but d'ajuster la courbe de référence, pour que la symétrie de la courbe soit adéquate avec les indices j et l'application G_s . autrement dit, le point 174 de la symétrie sera déplacé vers le milieu de la courbe c'est a dire $174+8=182$, et les points d'indice supérieur à $2*174$, soit 348 seront partagés entre les deux parties symétriques.

L'application du programme d'approximation donne le polynôme substituant l'application G_s :

$$P_s(x) = - 0.0016x^2 + 1.097x + 0.4695$$

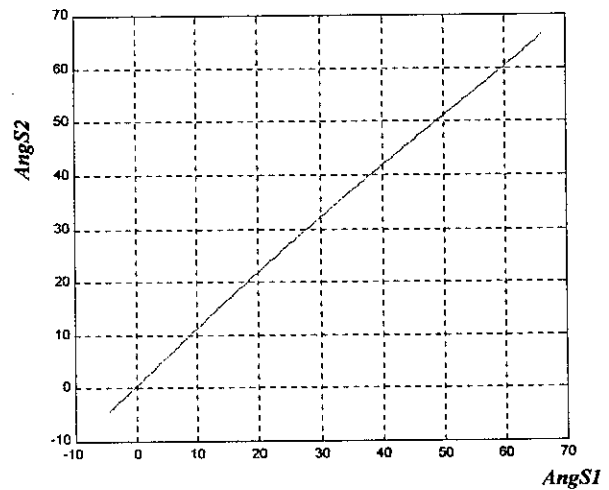
Les figures fig.III-42-43 représentent la courbe G_s et la courbe de l'erreur commise par l'approximation.

On peut conclure que cette approximation à donner une réduction du nombre du polynômes qui substituent la référence.

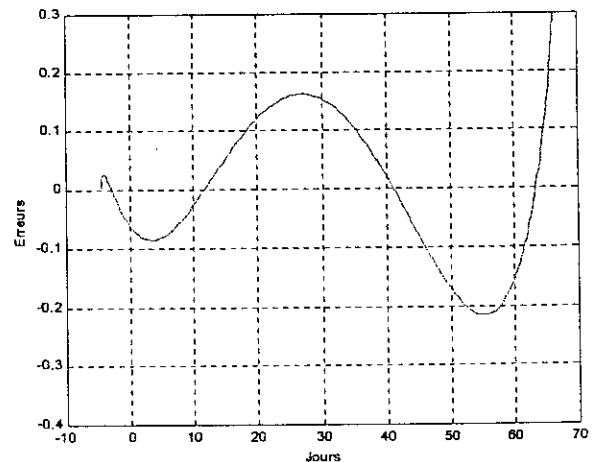
Remarque :

Le polynôme PI approximant la partie $j=1,70$ de la référence devient pour $j=-9,1,70$:

$$PI(x)=-0.0038x^2 -0.0064x + 66.326$$



FigIII-42 : Courbe de G_s



FigIII-43 : Courbe d'erreur



III- Les angles d'orientation des latitudes exprimées avec les degrés et les minutes :

Dans ce paragraphe nous allons voir les angles d'orientation qui correspondent aux latitudes non entières. Les latitudes non entières sont simplement les latitudes exprimées avec les degrés et les fractions du degré, les minutes ; par exemple la ville de Ghardaïa correspond à la latitude 32° et 24 minute, ou encore 32.4° (24 minute = $24/60^\circ = 0.4^\circ$).

Avec les polynômes précédent on peut calculer toutes les valeurs d'angles d'orientation avec une très grande précision, a condition que les angles calculés correspond à des latitudes de valeurs entières en degré. Pour les latitudes comportant des minutes, le calcul sera différent, pour ça nous allons suivre les étapes suivantes :

Soit la latitude n degré et m minute

- On calcul les angles d'orientations correspondant à la latitude n degré, en utilisant les polynômes d'approximations.
- On calcul les angles d'orientations correspondant à la latitude $n+1$ degré, avec la même méthode.
- les angles d'orientations de la latitude n degré et m minute seront déduites par l'interpolation en utilisant les minutes m comme poids d'interpolation entre les deux latitudes n et $n+1$, de la manière :

$$Ang_{lat}(n,m) = (1-m/60) * Ang_{lat}(n,0) + (m/60) * Ang_{lat}(n+1,0)$$

Avec $Ang_{lat}(n,m)$ le vecteur comprenant les angles d'orientations de la latitude n degré et m minute, et ainsi de suite.

L'idée de l'interpolation linéaire en utilisant les minutes comme poids ne vient pas de l'hasard. La figure *fig.III-44* représente la courbe des angles d'orientation le long d'une année pour toute les latitudes couvrant l' Algérie, cette courbe est représentée en trois dimension (*angle, latitude, jours*), cette représentation est invoquée afin de montrer les variations des angles d'orientation en fonction des latitudes, ces variations peut être vues comme des variations linaires sur chaque axe des jours.

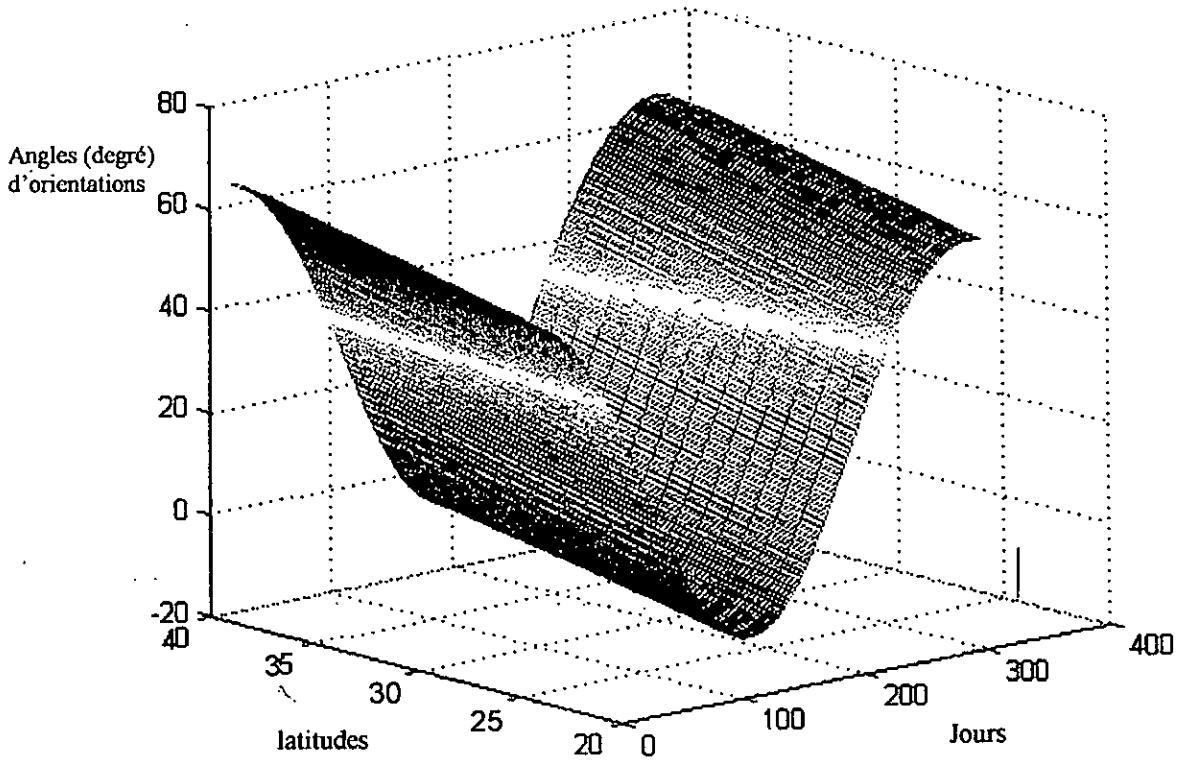


Fig.III-44: les angles d'orientations en fonction des jours et des latitudes

III-2 Les erreurs :

L'erreur commise par l'usage de l'interpolation est donnée par la figure fig.III-45 ci-dessous. On remarque que l'erreur est très faible, on peut donc utiliser l'interpolation pour la déduction des valeurs des latitudes exprimées par les degrés et minute.

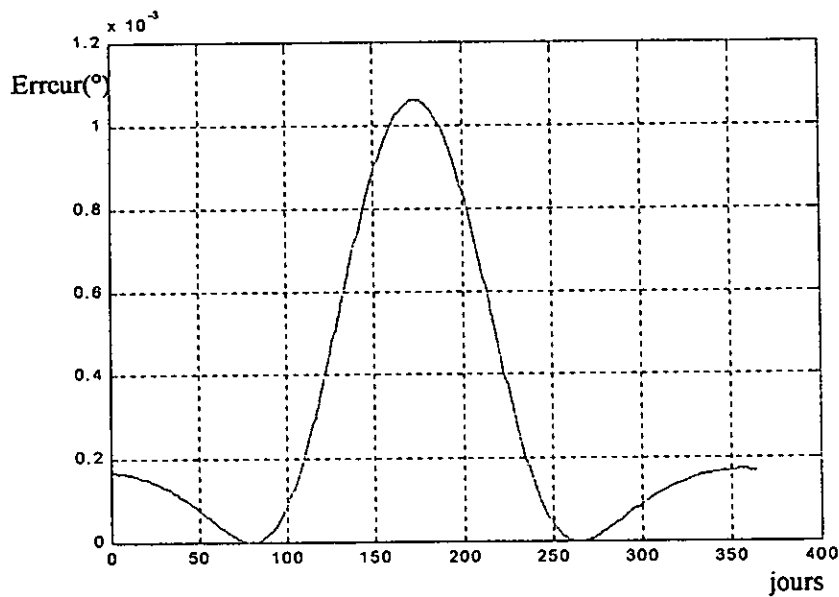


Fig.III-45 : Erreur commise avec l'interpolation

III-3 Généralisation de l'interpolation :

La figure III-44 représente tout les angles d'orientations durant une année, en parcourant toutes les latitudes d' Algérie de 22° à 37°. La justification de l'interpolation entre deux latitudes successives, peut être étendue pour une interpolation plus généralisée sur toutes les latitudes.

En effet cette interpolation consiste à déduire les valeurs angles d'orientations des latitudes comprises entre les latitudes 36.7° d'Alger, et 22° de Tamnrasset, qui vont être pris comme extrémités de l'interpolation, et la différence entre valeurs des latitudes sera pris comme poids de l'interpolation. On supposant x la variable latitude, on peut exprimer le vecteur d'angle correspondant à cette latitude x par :

$$Ang(x) = [(36.7-x)/(36.7-22)] * Ang(22^\circ) + [1 - (36.7-x)/(36.7-22)] * Ang(36.7^\circ)$$

L'erreur commise avec des valeurs de latitudes interpolés et valeurs réelles est donnée par la courbe fig.III-46 en 3 dimensions (jour, latitude- x , erreur)

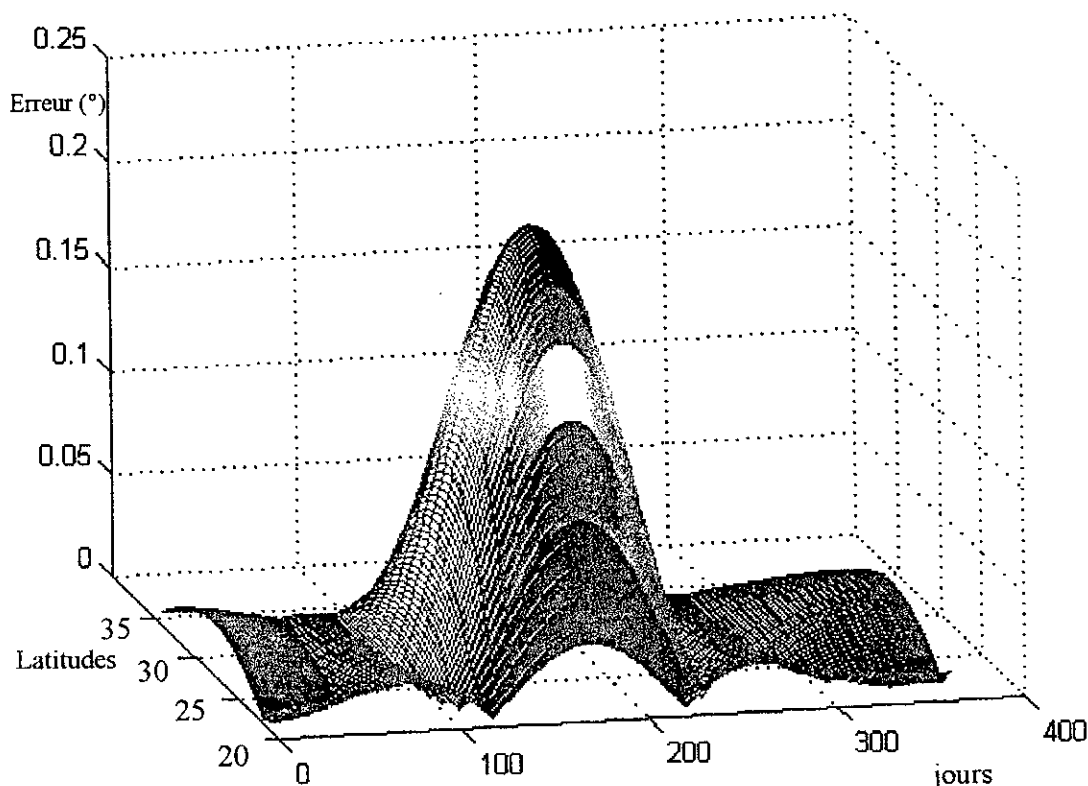


fig.III 46 : Erreur commise avec l'interpolation



L'erreur représentée dans la figure est inférieure à 0.2 degré, ce qui fait que l'interpolation est bien réussite, de ce fait on peut réduire les polynômes représentant les latitudes comprises entre 36.7° et 22°.

Résultats :

Dans tous les paragraphes précédents on voulait approximer les valeurs d'angles d'orientation dans le but de les exprimer sous forme de polynômes, pouvant générer ces angles.

Avec l'interpolation on a put diminuer le nombre des polynômes, de manière à éliminer tous les polynômes représentant les latitudes comprises entre 36.7° et 22°. Les coefficients des différents polynômes sont rangés dans une matrice M_g , qu'ont appellera *matrice génératrice d'angle d'orientation*, cette matrice est donnée par :

$M_g =$	-0.0038	-0.0064	66.32	}	<i>La référence</i>
	-0.0004	-0.5552	83.54		
	0.0037	-1.434	130.58		
	0.0059	-2.054	174.28		
	-0.0016 1.0967 0.4695			<i>Coefficient générant la symétrie</i>	
	0.0021 0.8469 -11.8750				
	<i>Latitude 22°</i>				

IV Algorithme de génération de toutes les angles d'orientation à partir de M_g :

La matrice génératrice M_g contient des coefficients de polynôme de second ordre, cette matrice donne toutes les angles d'orientations d'un panneau solaire à n'importe quelle point dans le territoire de l'Algérie. Les angles fournis sont assez proche des angles optimales d'orientation, l'algorithme suivant donne les étapes pour extraire les valeurs d'angles à partir de M_g .

%génération de la première latitude -référence

j=1

faire tant que j < (70+9)

*Ang(i) = $M_g(1,1) * j^2 + M_g(1,2) * j + M_g(1,3)$*

J=j+1

Fait



faire tant que j < (110+9)

$$\text{Ang}(i) = \text{Mg}(2,1) * j^2 + \text{Mg}(2,2) * j + \text{Mg}(2,3)$$

$$J = j + 1$$

Fait

faire tant que j < (145+9)

$$\text{Ang}(i) = \text{Mg}(3,1) * j^2 + \text{Mg}(3,2) * j + \text{Mg}(3,3)$$

$$J = j + 1$$

Fait

faire tant que j < (174+9)

$$\text{Ang}(i) = \text{Mg}(4,1) * j^2 + \text{Mg}(4,2) * j + \text{Mg}(4,3)$$

$$J = j + 1$$

Fait

%génération de la partie symétrique

pour j=1, 174+9

faire

$$\text{Angsym}(i) = \text{Mg}(5,1) * [\text{Ang}(j)]^2 + \text{Mg}(5,2) * [\text{Ang}(j)] + \text{Mg}(5,3)$$

fait

%Arrangement des valeurs de Ang(i) et Angsym(i)

$$\text{Ang}36.7^\circ = [\text{Ang}(i) ; \text{Angsym}(i+174)]_{i=1,174}$$

%génération de la latitude 22°

pour j=1, 365

faire

$$\text{ang}22(i) = \text{Mg}(6,1) * [\text{Ang}36.7(j)]^2 + \text{Mg}(6,2) * [\text{Ang}36.7(j)] + \text{Mg}(6,3)$$

fait

% déduction des angles des latitudes entres 22° et 36.7°

pour lat=36,23,-1 faire

$$x = \text{lat}$$

pour i=1, 365 faire

$$\text{Interpol}(\text{lat}, i) = ((x-22)/(36.7-22)) * \text{Ang}(i) + ((36.7-x)/(36.7-22)) * \text{ang}22(i)$$

$$\text{Err}(\text{lat}, i) = \text{Interpol}(\text{lat}, i) - y_m[((\text{lat}-1) * 365) + i];$$

$$\text{Erpuissance}(\text{lat}, i) = 1 - \cos(\text{err}(\text{lat}, i) * \pi / 180);$$

Fait

Fait

%fin

L'algorithme est traduit avec Matlab, les résultats globaux obtenus sont représentés sur les figures *fig.III-46* et *fig.III-47*.

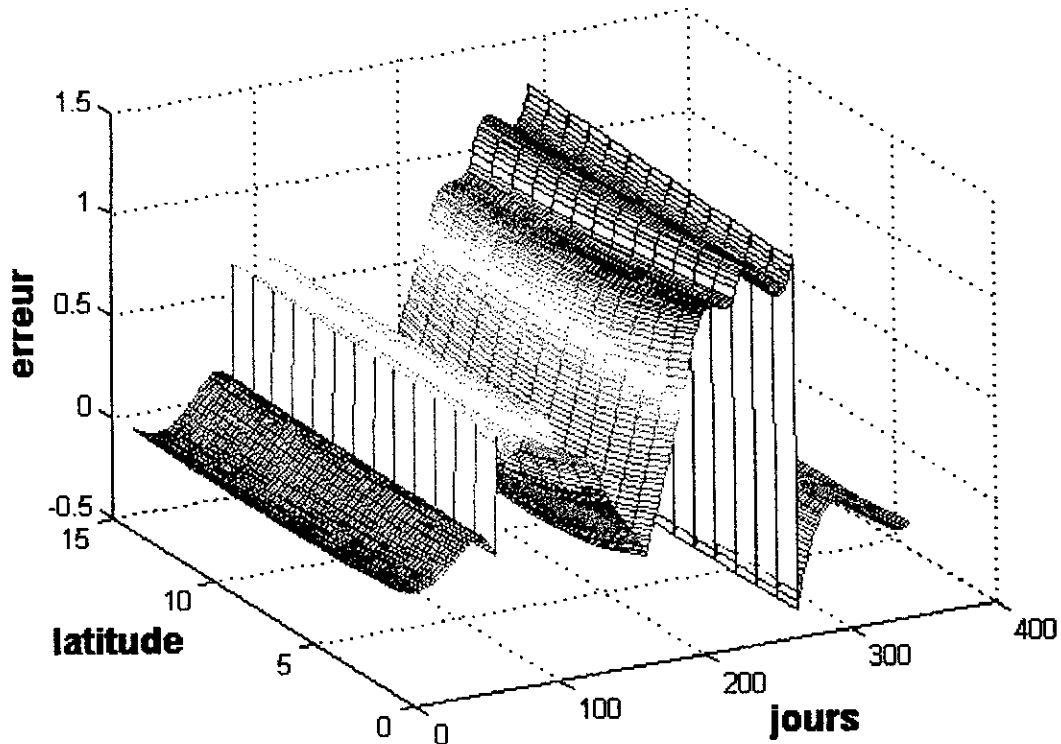


Fig.III-47 Erreur (degré) totale commise par la substitution

4

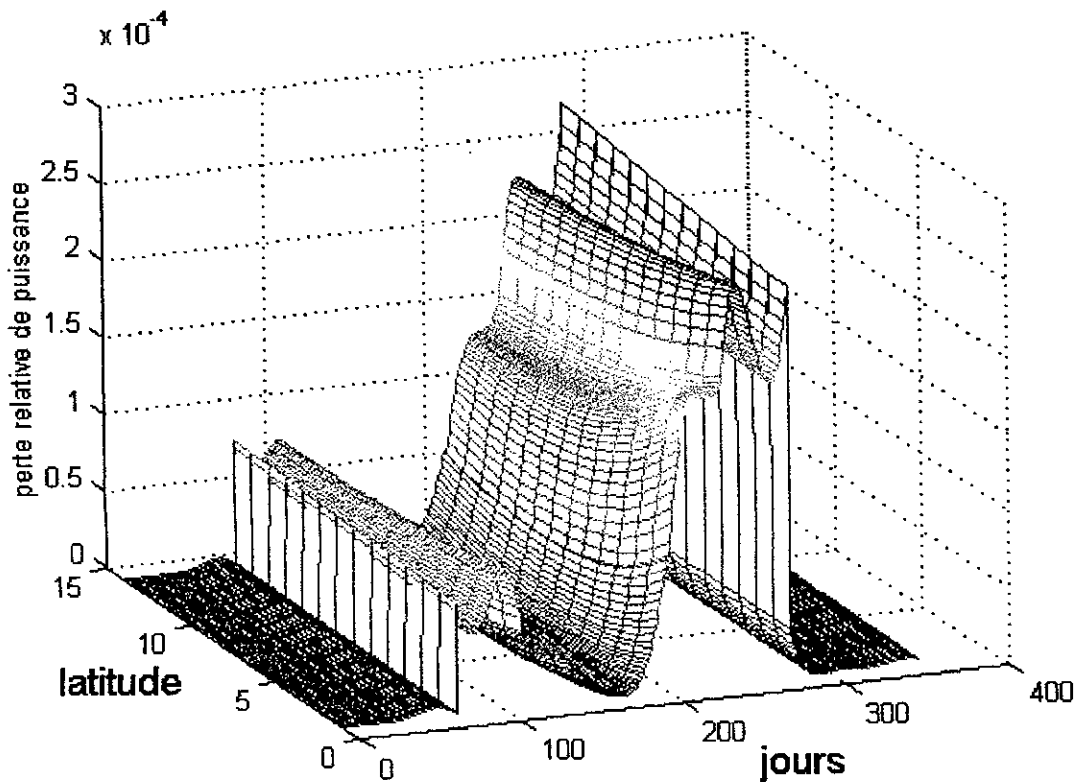


Fig.III-48 : Perte relative en puissance



les deux figures *fig.III-47* et *fig.III-48* représentent respectivement l'erreur commise dans l'évaluation des angles d'orientation en degré, et la perte relative en puissance du panneau solaire avec ces valeurs d'angles d'orientation.

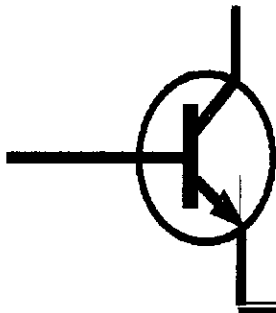
La perte relative en puissance est de l'ordre des millièmes, cette perte insignifiante de puissance nous amène à dire que la génération des angles d'orientations par la matrice Mg est parfaite.

Conclusion :

Les problèmes rencontrés lors de l'implantation de la commande d'orientation d'un panneau solaire tel, la taille des données à implantées dans la EEPROM, la complexité des programmes employant les formules géométriques pour calcul des angles optimales, nous a conduit à la conception du programme *APPROXIM(,)* donnant une substitution des valeurs d'angles d'orientations en polynôme de second ordre.

La représentation polynomiales des angles d'orientation est la formulation des relations entre angles et latitudes, nous a mené à une forme condensée des données, cette forme comprend des coefficients, donnant à la fois les polynômes d'approximations, et les argument des relations liant les différentes parties des données, cette forme condensée de donnée est exprimée sous la forme d'une matrice de six lignes et trois colonnes, cette matrice est nommée « *Matrice génératrice des angles d'orientation* » Mg .

La solution apportée par la matrice génératrice des angles d'orientation Mg , est très pratique, car elle assure une bonne commande, et une taille mémoire d'EEPROM très réduite, de plus il est clair que la matrice génératrice des angles d'orientation Mg , donne une très bonne approche des angles optimales et présente ainsi une parfaite commande d'orientation d'un panneau solaire.



CHAPITRE IV

Implantation sur Micro-contrôleur 68HC11

- Introduction
- Choix du microcontrôleur
- Implantation des données sur EEPROM
- conclusion



Introduction :

Le développement d'une application à base de microcontrôleur, quels que soient son type et sa marque, diffère quelque peu de celui d'une application à base de microprocesseur à vocation plus généraliste.

Dans le présent chapitre nous allons exposer la programmation du microcontrôleur 68HC11, ou à terme plus précis, programmation de l'EEPROM de données.

L'implantation sur 68HC11 des programmes d'extraction des angles d'orientations à partir de la matrice génératrice Mg , est une tâche assez simple comparée à la difficulté de la mise au point des programmes eux-mêmes. L'implantation de la matrice Mg seule, nécessite un codage des coefficients en hexadécimale, y compris un codage de signe, la mantisse et de la virgule flottante.

II- Choix du microcontrôleur [6] :

Le choix du microcontrôleur est évidemment dicté en tout premier lieu par l'adaptation de son architecture interne aux besoins de notre application. Ce choix est aussi conditionné par ces fonctions et ses capacités de ce dernier à travailler, et de s'adapter avec d'autres éléments qui peuvent être utilisés par notre application.

Le microcontrôleur utilisé (disponible à l'école ENP), est le 68HC811E2. Cette appellation, indique les caractéristiques de ce microcontrôleur, en effet l'appellation complète des microcontrôleurs de la famille 68HC11, est plus longue, le nécessaire pour décrire au maximum ces caractéristiques. (voir annexe).

Le 68HC811E2 dispose de 256 octets de RAM, à partir de l'adresse \$0000, 2048 octets d'EEPROM à partir de \$f800. Les registres des interfaces sont placés en temps normal de \$1000 à \$103f. Il est possible aussi de modifier l'emplacement des mémoires et des registres grâce à un registre de configuration.

III Implantation des données sur EEPROM :

III-1 Stockage des données [7] :

La capacité mémoire d'un microcontrôleur, ou d'un dispositif à microprocesseur est par construction finie. Si x est un nombre réel, il est donc nécessaire de présenter x sous forme approchée. La notation la plus utilisée à l'heure actuelle est la représentation avec virgule flottante :

$$x \approx \pm m b^p$$

Où b est la base de numération, m la mantisse, et p l'exposant.



Les calculs internes sont généralement effectués en base $b=2$, même si les résultats affichés sont finalement en base de 10.

La mantisse m est un nombre écrit avec virgule fixe et possédant un nombre maximum N de chiffres significatifs (imposé par le choix de la taille des emplacements mémoires alloués au type réel) : suivant les machines m s'écrira :

$$m = a_0 a_1 a_2 \dots a_{k_1} , v_0 v_1 v_2 \dots v_{k_2} = a_0 a_1 a_2 \dots a_{k_1} + 0 , v_0 v_1 v_2 \dots v_{k_2}$$

$$\sum_{i=1}^{k_1} a_i b^{-i} + \sum_{i=1}^{k_2} v_i b^i \quad k_1 + k_2 = N$$

Dans notre cas les arguments à implanter, les coefficients de la matrice Mg (voir Chp.III), s'écrivent de la même manière, chaque coefficient de cette matrice contient jusqu'à 4 chiffres après la virgule, et un signe alternant entre négatif et positif.

Un codage de tous les coefficients de Mg est indispensable, les coefficients doivent être soumis à des transformations permettant à la fois ; la reconnaissance des signes et les parties décimale et entière des nombres, ce qui conduit à l'introduction des octets d'entête pour chaque nombre donnant des informations sur signe et l'exposant, afin de faciliter les opérations de multiplication, addition et plus essentiellement le stockage en mémoire.

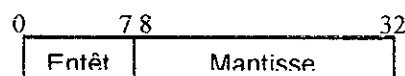
II-2 Codage des Nombres signés avec virgule flottante :

Le 68HC11 dispose d'instruction lui permettant d'effectuer quasiment toutes les opérations arithmétiques courantes sur des entiers codés sur 16 bits ce qui convient à un grand nombre d'applications. Il est parfois nécessaire d'utiliser une précision supérieure, soit de faire des calculs, avec des nombres au-delà de 16 bits, ou encore des manipulations des signes et virgule flottante.

Le codage que nous avons apporté, consiste à exprimer les nombres de la manière :

1 octet d'entête : contient l'exposant et le signe (un 1 pour négatif et 0 pour positif).

2 ou 3 octets (selon la précision) : contient la mantisse



donc si par exemple $x = -120,254$, il sera codé comme suite :

octets d'entête = 13 (1: négatif ; 3: trois chiffres après la virgule)

mantisse = 120254

Le nombre prend la forme en mémoire $x_{codé} = 13120256$.

Si le nombre était positif alors x s'écrit $x_{codé} = 03120256$.



Une représentation des nombres, de cette manière permet à la fois la reconnaissance de l'emplacement de la virgule (exposant) et le signe, et ainsi la mantisse entière. Cette représentation permet la reconstruction des nombres d'une manière exacte.

Conversion hexadécimale des nombres avec virgule flottante :

Dans l'exemple précédent, nous avons codé le nombre x en une forme contenant, un octet d'entête et 3 autres octets portant la mantisse en décimal. La conversion des nombres codés de cette manière en base hexadécimale n'est pas évidente.

La conversion des nombres décimaux entiers en hexadécimal est une opération ordinaire (division successive par 16), mais quand il s'agit des nombres ayant des virgule flottante l'algorithme de conversion change. A titre d'exemple le nombre 5 en décimal reste le même en hexadécimal \$5, mais le nombre 0,5 devient en hexadécimal \$0,8. pour mettre ça au clair on peut dire ; 0.5 en décimal veut dire *un demi de 1*, se qui correspond en hexadécimal avec la même écriture *un demie de 1*, la chose qu'il faut faire attention et savoir c'est que le 1 du décimal et le 1 du hexadécimal, ne se subdivisent pas de la même manière, c'est-à-dire, le 1 décimal se subdivise en 10 dixièmes, de 0,1 0,2 à 0,9, mais en hexadécimal le 1 se subdivise en 16 seizième, de 0,1 à 0,A... 0,F, se qui fait que un demi de 1 en décimal = 0.5 et un demi de \$1 en hexadécimal est 0,8.

D'une manière plus générale, un nombre décimal avec virgule flottante est écrit sous la forme :

$$a_0 a_1 \dots a_{k1}, b_0 b_1 \dots b_{k2} = a_0 * 10^{k1} + a_1 * 10^{k1-1} + \dots + a_{k1} * 10^0 + b_0 * 10^{-1} + \dots + b_{k2} * 10^{-k2} \quad \dots(1)$$

et un nombre en hexadécimal avec virgule flottante est écrit de la même manière, mais on remplace la base 10 par 16 et on écrit :

$$\$u_0 u_1 \dots u_{k1}, \$v_0 v_1 \dots v_{k2} = \$u_0 * 16^{k1} + \$u_1 * 16^{k1-1} + \dots + \$u_{k1} * 16^0 + \$v_0 * 16^{-1} + \dots + \$v_{k2} * 16^{-k2} \quad \dots(2)$$

donc si on veut convertir le premier nombre (décimal), en hexadécimal, il suffit de l'écrire sous la forme (2), c'est-à-dire en puissance positive et négative de 16. Procédons ainsi on aura :

$$a_0 a_1 \dots a_{k1}, b_0 b_1 \dots b_{k2} = [(a_0 * 10^{k1}) / 16^{k1}] * 16^{k1} + [(a_1 * 10^{k1-1}) / 16^{k1-1}] * 16^{k1-1} + \dots + [(a_0 * 10^0) / 16^0] * 16^0 + [(b_0 * 10^{-1}) / 16^{-1}] * 16^{-1} + \dots + [(b_0 * 10^{-k2}) / 16^{-k2}] * 16^{-k2} .$$



cette écriture donne une idée sur la conversion décimal-hexadécimal, et si on veut programmer la conversion, l'algorithme effectue les opérations suivantes :

Division successive par 16 de la partie entière et multiplication successive par 16 de la partie décimale, et extraction de la partie entière du résultat de chaque opération. Par exemple le nombre 168.254 sera convertit comme suit :

$$168 = 160 + 8 = 10 * 16 + 8 = \$A8$$

$$0.254 = 4 * 16^{-1} + 0.064 = 4 * 16^{-1} + 1 * 16^{-2} + 0.024 = 4 * 16^{-1} + 1 * 16^{-2} + 0 * 16^{-3} + 6 * 16^{-4} + 2 * 16^{-5} + \dots$$

donc le nombre 168.254 = \$A8.410624C... à 7 chiffres après la virgule.

On peut aussi coder le résultat en ajoutant un octet d'entête indiquant le signe et l'exposant, d'où l'écriture ;

$$\$A8.410624C = \$07A8410624C, (0 \text{ pour positif et } 7 \text{ pour } 7 \text{ chiffres après la virgule}).$$

Résultat :

Cette méthode donne la conversion décimal-hexadécimal en virgule flottante, de plus si on combine le codage des nombres exposé ci-avant on aura des résultats exprimés sous une forme facile à manipuler et stocker dans la mémoire.

II-1 Implantation de la matrice Mg :

La matrice Mg, est la matrice contenant les coefficients des polynômes générant la commande d'orientation d'un panneau solaire. Tout ses éléments sont en décimal signé avec virgule flottante. L'implantation de cette matrice nécessite la conversion en hexadécimal ainsi que son codage.

La matrice sur ça forme décimale est :

$$Mg = \begin{pmatrix} -0.0038 & -0.0064 & 66.32 \\ -0.0004 & -0.5552 & 83.54 \\ 0.0037 & -1.434 & 130.58 \\ 0.0059 & -2.054 & 174.28 \\ -0.0016 & 1.0967 & 0.4695 \\ 0.0021 & 0.8469 & -11.8750 \end{pmatrix}$$



Après conversion, on récupère la nouvelle matrice \$Mg\$;

$$Mg = \begin{pmatrix} f4f9 & f3 1A & 02D \\ f41A & f38E1 & 02528A \\ 04f2 & f316f1 & 028294 \\ 0416E & f320DD & 02AE47 \\ f468 & 03118C & 0278 \\ 04 89 & 03 D8C & f11BE \end{pmatrix}$$

Maintenant la matrice \$Mg\$ est sous une forme adéquate pour l'implantation sur EEPROM.

Programmes de la génération des angles d'orientations à partir de \$Mg\$:

La génération des angles d'orientations à partir de \$Mg\$, nécessite des programmes qui se chargent à la fois de la gestion des signes et de la virgule flottante, ainsi que l'extraction des relations portées par la matrice \$Mg\$.

Nous avons mis au point 5 sous programme en assembleur 68HC11, ces sous programme sont :

- ❑ Sous programme « calcul et ajustement de la position du jour »
- ❑ Sous programme « multiplication signée avec virgule flottante »
- ❑ Sous programme « addition signée avec virgule flottante »
- ❑ Sous programme « interpolation »
- ❑ Sous programme « calcul de la forme polynomiale de second ordre »

Ces sous programme sont gérés par un programme principal qui s'occupe des variables d'entées « jours, mois, latitude » et génère à la fin, une valeur correspondant à l'angles d'orientation pour la latitude donnée, à la date spécifier.

Sous programme « calcul et ajustement de la position du jour » :

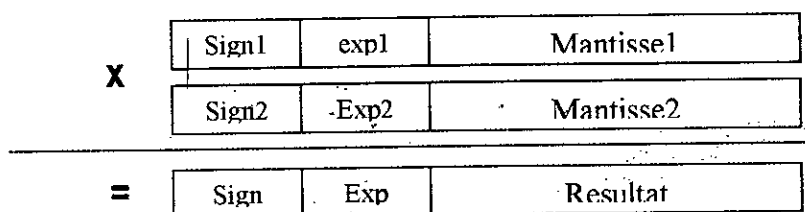
Ce programme calcul la position du jour \$J\$ à partir de la date *jour ; mois*, il consiste donc à donner un entier allant de 1 (1^{er} janvier) à 365 (31 décembre) le long de l'année. En plus il génère une autre variable de test indiquant la ligne de la matrice \$Mg\$ qui s'accorde avec cette position du jour (subdivision de la courbe de référence en intervalles), et positionne le bit le plus fort de cette variable à 1 si la symétrie intervient (\$J > 174\$), où \$J\$ prendra la valeur symétrique par apport à 174 (si \$J=178\$ alors \$j\$ reçoit 170).

**Sous programme « multiplication signée avec virgule flottante »**

Ce programme considère que tous les nombres manipulés (ex. coefficient de la matrice Mg) sont en hexadécimal et codés sur 3 octets ou 24 bits, 8 bits d'entête ; identifie le signe et l'exposant, et 16 bits pour la mantisse.

Dans ce programme apparaît mieux l'intérêt du codage des nombres que nous avons introduit. La multiplication des nombres codés ainsi, devient facile et moins compliquées.

La multiplication consiste à additionner les exposants des deux nombres, et faire un OU EXCLUSIF entre signes du multiplicateur et multipliant. Le schéma ci-dessous montre ces opérations.



$$\text{Résultat} = \text{Mantisse1} * \text{Mantisse2}$$

$$\text{Sign} = \text{XOR}(\text{sign1}, \text{sign2})$$

$$\text{Exp} = \text{exp1} + \text{exp2}$$

Il faut aussi signaler deux choses; l'une, est que la multiplication $\text{Mantisse1} * \text{Mantisse2}$ est une multiplication de 16 bits, et cela sous-entend un résultat de 16 à 32 bits, Le programme dans ce cas (plus de 16 bits) élimine les bits hauts nuls de l'octet du poids fort, et les bits (de 4 à 8bits) faibles du poids le plus faible et arrange l'exposant afin de réduire le résultat à 16 bits. La seconde ; et pour les mêmes raisons que la première, une procédure de subdivision des mantisses de 16 bits, en segments de 8 bits, afin de pouvoir manipuler les instructions de multiplications.

Sous programme « addition signée avec virgule flottante » :

Le sous programme addition signée avec virgule flottante, donne l'impression de simplicité, mais en réalité il est assez compliqué que le programme de multiplication.

Pour faire une addition signée avec virgule flottante, il faut tenir compte de tous les paramètres, que se soit le signe, l'exposant et la mantisse elle-même. Si par exemple on veut additionner deux nombres avec des exposants différents, on aura un problème d'alignement des bits à additionner, ce qui corrompt les résultats si l'addition s'effectue directement sur les mantisses (voir *fig.IV-2*). De même, les signes peuvent transformer l'addition en soustraction, et le signe du résultat va s'alterner entre le signe du premier nombre et le deuxième, car la



La mantisse seule ne peut fournir une information sur le plus grand nombre, afin de prendre son signe.

$$\begin{array}{r}
 \$10.54f \\
 + \$0.2574 \\
 \hline
 \$01,7A64 =
 \end{array}
 \xrightarrow{\text{Sur 68HC11}}
 \begin{array}{r}
 \boxed{0} \quad \boxed{3} \quad \boxed{1054f} \\
 \boxed{0} \quad \boxed{4} \quad \boxed{2574} \\
 \hline
 \boxed{*} \quad \boxed{*} \quad \boxed{12AC3} =
 \end{array}$$

Fig. IV-2 :Erreur d'addition des nombres ayant des exposants différents

Pour pallier ces problèmes, nous avons conçu le sous programme addition signée avec virgule flottante, de façon à faire une addition tenant compte de toutes les variantes de cette dernière, l'idée du programme, est de mettre les deux nombres à un exposant identique, c'est-à-dire faire des opérations de décalage à gauche et incrémentation de l'exposant associé à la mantisse décalée jusqu'à ce que les exposants s'identifient. L'intérêt de ces opérations, est de permettre la comparaison entre les deux mantisses, d'une manière sûre, afin de pouvoir décider sur l'opération d'addition ou soustraction, et signe du résultat. Ainsi dans l'exemple de la *figIV-2*, un décalage du premier nombre va donner un exposant identique = 4, est la mantisse sera décalée vers la gauche (voir *fig.IV-3*).

$$\begin{array}{r}
 \$10.54f \\
 + \$0.2574 \\
 \hline
 \$10,7A64 =
 \end{array}
 \xrightarrow{\text{Sur 68HC11}}
 \begin{array}{r}
 \boxed{0} \quad \boxed{3+1=4} \quad \boxed{1054f0} \leftarrow \\
 \boxed{0} \quad \boxed{4} \quad \boxed{2574} \\
 \hline
 \boxed{0} \quad \boxed{4} \quad \boxed{107A64} =
 \end{array}$$

Fig. IV-2 :Erreur d'addition des nombres ayant des exposants différents

Sous programme « interpolation » :

Le sous programme interpolation comme son nom l'indique, fait l'interpolation des données acquises à partir de Mg, des angles d'orientations des latitudes de référence et celle de 22°.

Sous programme « calcul de la forme polynomiale de second ordre » :

Ce programme nommé *Spcalcul* est en réalité composé des deux sous programmes *additions* et *multiplication*, mais il reste comme un programme très intéressant, car il permet



de la forme d'un polynôme de second ordre D'une autre façon ; pour une variable J , et des coefficients a_1 , a_2 et a_3 , il donne la constante ; $const = a_1 * J^2 + a_2 * J + a_3$,

Le sous programme calcul de la forme polynomiale de second ordre, donne plus de souplesse lors de l'extraction des angles d'orientation à partir de la matrice Mg , est grâce à ça indépendance totale de la matrice Mg (au contraire des sous programme interpolation et position de jours), il présente de grands avantages, même pour l'utilisation en d'autre application, quand peut voir plus loin.

Programme global :

Le programme global est le programme à implanter dans l'EEPROM du 68HC11, il comporte tout les sous programmes déjà cités, ainsi que la matrice Mg . La figure *fig.IV-4* représente l'algorithme général du programme « commande d'orientation d'un panneau solaire ».

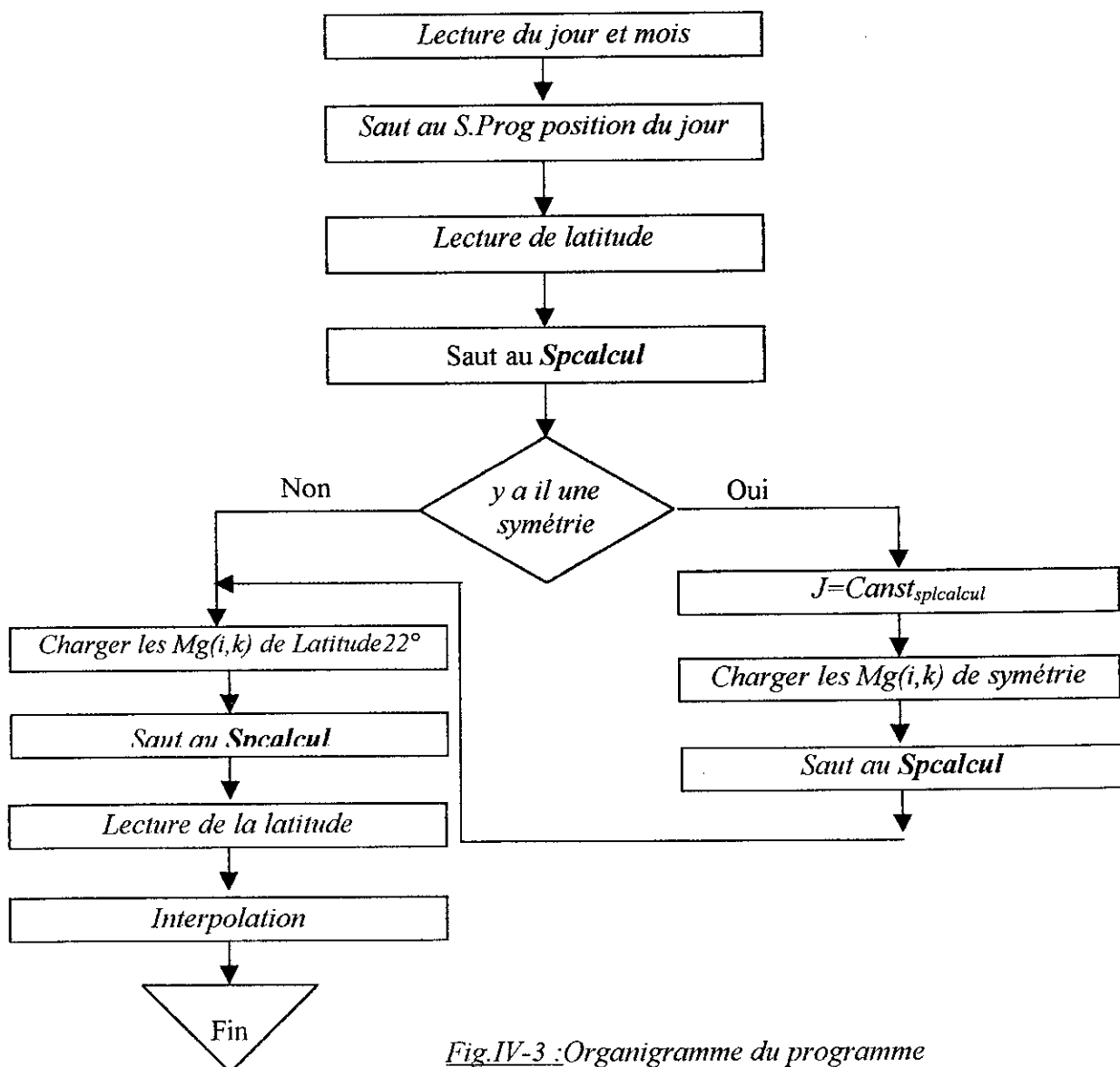


Fig.IV-3 : Organigramme du programme global



Mise en œuvre du 68HC11:

Le programme global présenté par l'organigramme *figIV-3*, doit être complètement implanté dans l'EEPROM du 68HC11. une telle opération nécessite toute une configuration des registres internes, et la rédaction d'autre programme qui pilote la gestion de ces derniers, ainsi que la gestion de l'EEPROM, la configuration du microcontrôleur en mode Boutstrap est nécessaire (ChapI), afin d'insérer ces programme dans la mémoire à travers la liaison série RS232 du PC.

Le PCPUG11 est un logiciel distribué gratuitement par Motorola, permet d'effectuer toutes les opérations d'initiation et de chargement sur le 68HC11, implanté dans un ordinateur, le logiciel peut grâce à une liaison série avec le microcontrôleur, la communication gérant la mise au point du programme et son implantation en mémoire.

Malheureusement le programme de la génération des angles d'orientation d'un panneau solaire, n'a pas été implanté sur microcontrôleur à cause des problèmes rencontrés lors de la simulation de la carte 68HC811E2, et cela est dû principalement aux variables intermédiaires de programme chargées en temps réels en RAM, cette dernière, la RAM ; débute de l'adresse \$00, en revanche la matrice *Mg* est stockée en EEPROM à partir de l'adresse \$f800 ce qui fait qu'on aura à faire à des calculs réalisés dans la RAM avec des arguments dans la EEPROM, ce qui signifie que la distance mémoire entre les arguments (coefficients de la matrice *Mg*) et les résultats de calcul est égale à la distance RAM-EEPROM soit de \$f800. Cette distance entraîne des adressages inhérents supérieur à 6 bits dont l'assembleur 68HC11 l'interdit, donc les étiquettes de branchement utilisées dans le programme donnent des adresses longues non supportés par 68HC11, mais on peut dans tout les cas refaire le programme de sorte à réduire la distance des branchements ou de reconfigurer le microcontrôleur de façon à arranger la RAM et la EEPROM dans le Map-mémoire de manière adjacente, et peut être ainsi le problème d'implémentation sera réglé.

Conclusion :

Les programmes de calcul des angles d'orientations d'un panneau solaire ont été mis au point avec un grand soin, le codage des nombres hexadécimales signés avec virgule flottante est choisit de manière à permettre une grande souplesse lors de la programmation en assembleur 68HC11.

Des programmes sont aussi réalisés à usage général soit la multiplication signée avec virgule flottante ou addition signée avec virgule flottante, peuvent être utilisés en d'autre applications.



A défaut de temps les programmes n'ont pas été introduit sur EEPROM, ce qui nous laisse le temps dans le futur de les voir encore et essayer de les implanter.

Conclusion générale

Dans le but de la commande l'orientation d'un panneau solaire par microcontrôleur 68HC11 qui ne dispose que de fonctionnalités très réduites à l'échelle des calculateurs ou encore des PCs, nous avons réalisé plusieurs programmes, afin de surmonter le problème d'adaptation entre microcontrôleur et la commande du panneau.

Les programmes réalisés le long de ce projet, se partagent en programmes exécutés sur PC, et programmes en langage assembleur, pour le microcontrôleur.

Les étapes suivies, du calcul d'angles d'orientations à l'approximation polynomiale, suivi de la construction de matrice génératrice et terminée par les programmes assembleur extracteur de la commande à partir de la matrice déjà nommée M_g , et spécialement le programme *s calcul*, permettant la génération de la forme polynomiale de second ordre, peut servir non seulement à la commande d'un panneau solaire, mais aussi pour tout les systèmes ayant une entrée et une sortie, par là notre procédé acquière un aspect plutôt général et ainsi le système du panneau solaire se réduit à un cas particulier, pour le quel les résultats obtenus se révèlent excellents.

Tel la multiplication, l'addition, la soustraction et la division les programme
La commande d'orientation d'un panneau solaire constitue un sujet d'actualité besoin
d'intégré ces applications dans notre pays.

Annexe I

Les principales villes d'Algérie avec les
latitudes correspondantes

Les principales villes d' Algérie avec les latitudes correspondantes :

Site	Degré	Minute
Adrar	27	49.2
Ain sefra	32	45.6
Alger	36	43.2
Annaba	36	49.8
Aoulef	27	3.6
Ain salah	27	12
Batna	35	33
Beni saf	35	18
Biskra	34	48
B.B.Arredj	36	4.2
Béchar	31	38.4
Bejaïa	36	43.2
B.O.Dris	28	6
Beni Abbès	30	7.8
Chlef	36	13.2
Canstantine	36	16.8
Djelfa	34	40.8
Djanet	23	33
El Oued	33	30
El Bayadh	33	40.2
El Golea	30	34.2
Ghardaïa	32	24
H.Messaoud	31	40.2
In-Amenas	28	3
Illizi	26	30
M'sila	35	40.8
Mostganem	35	48.6
Maghnia	34	48.6
Oran	35	37.8
Ouargla	31	54
Skikda	36	52.8
Sétif	36	10.8
Saida	34	51.6
Tamanrasset	22	46.8
Tindouf	27	40.2
Tebessa	35	28.8
Tlemcen	34	57
Timimoun	29	15
Tougourt	33	7.2

Annexe II

La famille 680HCl | Ex

Famille du 68HC11EX

Référence	ROM (octets)	RAM (octets)	EPROM (octets)	EEPROM (octets)	Timer	E/S (Circuit seul)	E/S (Etendu)	Série	A/D	PWM	Boitier	Tempé- rature	OTP	Vitesse bus maxi.
HC11E0	x	512	x	x	16-bits, 3/4IC, 4/50 C RTI, pulse acc.	38	22	SPI SCI	8-ch 8-bit	x	52PLCC 64QFP	M,V,C	711E9	3MHz
HC11E1	x	512	x	512	16-bits, 3/4IC, 4/50 C RTI, pulse acc.	38	22	SPI SCI	8-ch 8-bit	x	52PLCC 64QFP	M,V,C	711E9	3MHz
HC811E2	x	256	x	2048	16-bits, 3/4IC, 4/50 C RTI, pulse acc.	38	22	SPI SCI	8-ch 8-bit	x	52PLCC	M,V,C	X	2MHz
HC11E8	12k	512	x	x	16-bits, 3/4IC, 4/50 C RTI, pulse acc.	38	22	SPI SCI	8-ch 8-bit	x	52PLCC 64QFP	M,V,C	711E9	3MHz
HC11E9	12k	512	x	512	16-bits, 3/4IC, 4/50 C RTI, pulse acc.	38	22	SPI SCI	8-ch 8-bit	x	52PLCC 64QFP	M,V,C	711E9	3MHz
HC711E9	x	512	12k	512	16-bits, 3/4IC, 4/50 C RTI, pulse acc.	38	22	SPI SCI	8-ch 8-bit	x	52PLCC 52PLCC 64QFP	M,V,C	711E9	3MHz
HC11EA9	12k	x	x	512	16-bits, 3/4IC, 4/50 C RTI, pulse acc.,PLL	38	22	SPI SCI	8-ch 8-bit	x	52PLCC 64 PDIP	C	711E9	2MHz
HC711EA9	x	12k	12k	512	16-bits, 3/4IC, 4/50 C RTI, pulse acc.,PLL	38	22	SCI+	8-ch 8-bit	x	52PLCC	C	711E9	2MHz
HC11E18	x	x	x	512	16-bits, 3/4IC, 4/50 C RTI, pulse acc.	38	22	SCI+	8-ch 8-bit	x	52PLCC 64QFP	C	X	3MHz
HC11E20	20k	x	x	512	16-bits, 3/4IC, 4/50 C RTI, pulse acc.	38	22	SPI SCI+	8-ch 8-bit	x	52PLCC 64QFP	M,V,C	711E20	3MHz
HC711E20	x	20k	20k	512	16-bits, 3/4IC, 4/50 C RTI, pulse acc.	38	22	SPI SCI+	8-ch 8-bit	x	52PLCC 64QFP	C M,V,C	711E20	2MHz
HC11E32	32k	x	x	512	16-bits, 3/4IC, 4/50 C RTI, pulse acc.	38	22	SPI SCI+	8-ch 8-bit	x	52PLCC	C	711E32	4MHz
HC711E32	x	32k	32k	512	16-bits, 3/4IC, 4/50 C RTI, pulse acc.	38	22	SPI SCI+	8-ch 8-bit	x	52PLCC	C	711E32	4MHz

Annexe III

LE PROGRAMME APPROXIM(X,Y)

Le programme *Approxim(x,y)* :

C'est un programme qui fait l'approximation des points $(y(i),x(i))$ par un polynôme de degré 'n' au choix. Les algorithmes d'approximations choisis, dans le but d'approximer les données, angles d'orientations, on était sélectionné selon des critères. En effet ces critères expriment deux choses, une erreur minimale, et possibilité de contrôler le degré du polynôme employé dans l'approximation, ces méthodes ainsi sélectionnées, sont traduites vers le langage *Matlab*, sous forme de sous-programmes, appartenant à un programme principal nommé *APPROXIM(,)*.

Le programme principal est en réalité, conçu sous forme d'une fonction, elle-même fait appel à d'autres fonctions constituant les trois méthodes ; moindre carrée, mini-max et système surdéterminé. Le programme est aussi doté, des outils de dialogues avec les utilisateurs ; l'utilisateur peut choisir la méthode d'approximation, le degré du polynôme d'approximation, il peut choisir également la visualisation des différentes courbes ; erreur, courbes réelle courbe polynomiale, en plus des boites de dialogues et figures, des messages d'erreur complète ces derniers afin d'assurer une bonne interaction avec le programme.

DESCRIPTION

$p = \text{Approxim}(x,y,n)$ trouve les coefficients d'un polynôme $p(x)$ de degré n qui substitue parfaitement les données, $p(x(i))$ à $y(i)$, dans sens des moindres carrés, système surdéterminer ou au sens de l'erreur Min-Max. Le résultat p est un vecteur ligne de $n+1$ élément, ce vecteur contient les coefficients polynomiaux avec les puissances descendantes.

LA SYNTAXE

Le programme *Approxim* peut être utilisé sous deux formes, soit directement sur la fenêtre de command Matlab, et dans ce cas il faut procéder de la manière :

Insertion des données :

$X=[x_0 \ x_1 \ x_2 \ \dots \ x_n]$ ←

$Y=[y_0 \ y_1 \ y_2 \ \dots \ y_n]$ ←

Puis tapez :

Approxim(X,Y) ←

Et puis, le programme se charge automatiquement du dialogue avec l'utilisateur en affichant des messages et boites de dialogues pour le choix de la puissance du polynôme d'approximation ainsi que l'algorithme voulu.

La deuxième forme, d'utilisation possible du programme est l'emploi de *Approxim(.)* comme fonction interne de Matlab c'est-à-dire, l'insertion de la ligne $p = \text{Approxim}(X,Y,n)$ directement dans un autre programme dont on a besoin de substituer des données (x,y) avec un polynôme, dans ce cas on n'aura plus le dialogue qui permet à l'utilisateur le choix des algorithmes et la visualisation, c'est pour ça qu'on a introduit le troisième arguments n à la place de deux, (n indique le degré du polynôme d'approximation), et ôté toute procédure d'affichage et gestion d'erreur. De plus le programme *Approxim(.)* est décomposé en ces trois principaux Algorithmes, et l'écriture exacte à la place de $p = \text{Approxim}(X,Y,n)$ est : $p = \text{Minmax}(X,Y,n)$, $p = \text{Moindre}(X,Y,n)$ ou $p = \text{Surdet}(X,Y,n)$, selon l'algorithme.

BIBLIOGRAPHIE

- [1]. CHRISTIAN TAVERNIER.
« Micro-contrôleur 68HC11 (Version UVPR0M & EEPROM)
Description »
Edition Dunod, Paris, 1997
- [2]. Photopiles solaires
Article de SIFFERT PAUL ; Docteur ès Sciences, directeur de
laboratoire Phase (UPR du CNRS). Encyclopédie *Universalis*.
- [3]. FRANCIS SCHEILD
« Analyse numérique, cours et problèmes ».
Copyright McGraw-Hill, New York, 1986.
- [4]. Matlab Book Reference (source PDF)
http://www.mathworks.com/access/helpdesk/help/pdf_doc/matlab/refbook.pdf
- [5]. A.KOSTRIKIN
« Introduction à l'algèbre».
Edition MIR, MOSCOU, 1977.
- [6]. BERNARD BEGHYN
« Micro-contrôleur 68HC11 »
Edition Hermès , Paris, 1997
- [7]. JEAN-PIERRE DEMAILLY
« Analyse numérique et équation différentielles »
Edition Dunod 1999.
- [8]. DARDAN
« Le Microprocesseur 6809 »
- [9]. PAUL HOROWITZ ET WINFIELD HILL.
« Traitement d'Electronique Analogique et numérique »
Volume 2 : technique numérique et analogique
1^{er} Edition, Publitr0nic/Elector, novembre 1996.

[10]. MOHAND MOKHTARI et ABDELHALIM MESBAH.
« Apprendre à maîtriser MATLAB »
Edition 1996.

[11]. CHRISTIAN TAVERNIER.
« Micro-contrôleur 68HC11 (Version UVROM & EEPROM)
Applications »
Edition Dunod, Paris, 1997