M2AN - MODÉLISATION MATHÉMATIQUE ET ANALYSE NUMÉRIQUE

J. VUILLEMIN

Contribution à la résolution numérique des équations de Laplace et de la chaleur

M2AN - Modélisation mathématique et analyse numérique, tome 27, n° 5 (1993), p. 591-611

<http://www.numdam.org/item?id=M2AN_1993__27_5_591_0>

MATHEMATICAL MODELLING AND NUMERICAL ANALYSIS
MODÉLISATION MATHÉMATIQUE ET ANALYSE NUMÉRIQUE
(Vol. 27, n° 5, 1993, p. 591 à 611)

CONTRIBUTION A LA RÉSOLUTION NUMÉRIQUE DES ÉQUATIONS DE LAPLACE ET DE LA CHALEUR (*)

par J. VUILLEMIN (1)

Communiqué par R. TEMAM

Résumé. — *Nous adaptons les techniques classiques de résolution par différences finies des équations de la chaleur et de Laplace, au traitement par des opérateurs matériels spécialisés. Le parallélisme du calcul est obtenu par un pipe-line régulier, dont la profondeur arbitraire s'adapte directement au volume de matériel disponible.*

Une implantation de la méthode, sur technologie PAM (mémoire active programmable [BRV 89], [BRV 92]*), utilise un pipe-line de* 128 *opérateurs* 24 *bits, avec une fréquence d'horloge de 20 MHz. Ceci permet de calculer* 5 G (2) *opérations utiles par seconde, addition ou décalage en virgule fixe. Les résultats obtenus en virgule fixe étant, pour ce type de problème, égaux à ceux obtenus en virgule flottante, cette réalisation dépasse, en valeur absolue, les puissances de calculs précédemment publiées (*[McB 88] *et* [McB & al 91]*), tous types de matériels confondus. Un ordinateur séquentiel classique devra, quant à lui, exécuter* 25 G *instructions par seconde pour reproduire le même calcul. Si enfin on compare, en francs par opération par seconde, la résolution des mêmes équations sur super-ordinateur scalaire et ordinateur massivement parallèle, la technique proposée permet une économie de deux ordres de grandeur sur le prix de ce calcul.*

Abstract. — Contribution to the numerical solution of the Laplace and heat equations by specialized hardware. *We adapt the classical finite difference method to compute solutions of the Heat and Laplace equations with help from special purpose hardware. Parallelism is achieved through a pipe-line, whose depth can be adapted to available silicon area.*

An implementation of the method on PAM *(Programmable Active Memory) technology runs at* 20 MHz, *with a pipe depth of* 128 *operators. This lets us compute* 5G *additions and shifts per second, with a* 24 *bits fixed-point data format. Since it is easy to show that fixed-point gives the same results as floating-point for these two problems, this exceeds computing performances previously reported (*[McB 88] *and* [McB & al 91]*) for super-computers.*

A serial computer will need to execute 25 G *instructions per second, to reproduce the same computation. The price of solving the Heat and Laplace equations, in $ per operation per second, is two order of magnitude lower with the proposed PAM implementation, than with super-computers.*

(*) Manuscrit reçu le 6 mai 1992 et sous forme révisée le 16 septembre 1992.

(1) Digital Equipment Corporation, Paris Research Laboratory, 85, avenue Victor Hugo, 92563 Rueil-Malmaison Cedex, France.

(2) $1\ T = 10^{12}$, $1\ G = 10^9$, $1\ M = 10^6$, $1\ K = 10^3$.

M² AN Modélisation mathématique et Analyse numérique 0764-583X/93/05/591/21/$ 4.00

1. INTRODUCTION

L'analyse numérique, au sens général de [DL 88], est le domaine privilégié d'applications des super-ordinateurs. Pour de multiples raisons :

- La simulation numérique du monde physique est l'une des clés des technologies modernes ; de la chimie moléculaire au nucléaire, de l'espace à l'électronique.
- Dans beaucoup de ces domaines, la puissance de calcul aujourd'hui disponible est dérisoire, en regard de l'ampleur des problèmes à traiter ; de la simulation d'une aile à celle de l'avion tout entier ; de la recherche d'une nappe de pétrole à la cartographie souterraine du globe.
- De par la nature même des phénomènes physiques sous-jacents, nombre de ces problèmes offrent des possibilités, sans limite conceptuelle, au parallélisme des calculs ; de la simulation des galaxies à celle des circuits intégrés, de l'équation de Schrödinger à celles de Maxwell et Laplace.
- Au vu de l'importance scientifique, financière ou stratégique de certaines applications, les agents économiques, états et industrie, consacrent des sommes considérables à la mise au point et à l'exploitation des super-ordinateurs.

1.1. Super-ordinateurs ou circuits spécifiques ?

Pour arriver aux performances des super-ordinateurs, du Gops en 1990 au Tops en l'an 2000, on suit, pour l'instant, deux voies principales :

1. les semi-conducteurs (ECL ou AsGa) ultra-rapides ;
2. les structures massivement parallèles.

Elles sont largement antagonistes. Les technologies rapides sont, aujourd'hui, intrinsèquement *chaudes* par leur consommation électrique. Les machines résultantes sont donc volumineuses, et à faible parallélisme. Les structures massivement parallèles utilisent des technologies beaucoup plus lentes. Elles doivent de plus consacrer, en l'état courant de l'art, une partie importante de leurs ressources à la gestion même du parallélisme. C'est pourquoi il leur faut *beaucoup* de processeurs, pour simplement égaler les performances des machines séquentielles les plus rapides. Dans les deux cas, les super-ordinateurs résultants sont d'un usage complexe, et coûtent très cher.

Il existe pourtant une troisième voie, rendue possible par l'avènement des circuits à haute densité d'intégration (VLSI). Ceux-ci permettent, en théorie, de réaliser des matériels spécifiques, dont la structure physique est directement calquée sur la nature algorithmique de chaque application :

• Avantages : dans un volume et pour un prix comparable à ceux d'un micro-ordinateur, on dispose d'une puissance de calcul comparable à celle d'un super-ordinateur, voire nettement supérieure dans bien des cas.

• Inconvénients : une telle machine n'est capable d'exécuter qu'une classe très particulière d'applications, voire une seule. Le coût et le temps de réalisation du prototype initial sont élevés, sans commune mesure avec celui de la première mise au point d'un logiciel pour super-ordinateur.

Ces inconvénients, combinés aux difficultés que rencontre l'analyste numéricien devant formuler ses problèmes en termes utilisables par un électronicien, cantonnent, pour le moment, notre troisième voie aux études académiques, sans retour expérimental. Ceci, en dépit de son impact économique, potentiellement considérable.

1.2. Circuits programmables

La technologie PAM (mémoires actives programmables) change les données du problème. Il s'agit de composants dont on peut changer dynamiquement, plusieurs centaines de fois par seconde, la configuration : structure de connexion et tables de vérité des fonctions booléennes. On consultera les références [GK 89], [X 87] et [K 89] pour des implantations et des applications de cette technologie. Nous décrivons les réalisations et applications de $DecPeRLe_0$, une PAM de 50 K portes logiques en [BRV 89], et $DecPeRLe_1$, une PAM de 250 K portes en [BRV 91]. Ces études montrent que :

1. Tout circuit digital synchrone est réalisable, sur une PAM de taille suffisante. Une PAM de taille infinie est au calcul parallèle ce qu'une machine de Turing est au calcul séquentiel.

2. Si on compare, à technologie égale, les performances de deux réalisations du même algorithme parallèle, par un circuit spécifique et par PAM :

• La période d'horloge est comparable, car invariablement déterminée par des considérations externes (vitesse des mémoires ou du Bus) ;

• Le circuit spécifique est plus petit, d'environ deux ordres de grandeur, que la réalisation PAM, soit une carte ($\approx 100\ cm^2$) pour un composant ($\approx 1\ cm^2$) ;

• Le temps de mise au point d'une application PAM est comparable à celui de l'écriture d'un logiciel très optimisé, quelques semaines, en regard des mois nécessaires à la réalisation d'un circuit spécifique ;

• Le coût matériel initial de la solution PAM est nul ([1]), celui d'un circuit

([1]) A condition, bien entendu, de disposer de cette technologie.

spécifique est de quelques centaines de milliers de francs. Le circuit spécifique ne devient rentable qu'à partir de plusieurs centaines de systèmes réalisés.

1.3. Contribution

Munis de cette technologie PAM, nous tentons, depuis trois ans, d'en évaluer l'impact scientifique, en particulier pour l'analyse numérique. Sous l'impulsion de J. P. Quadrat, nous avons choisi de résoudre l'équation de Laplace, en deux dimensions :

$$\frac{\delta^2 F}{\delta x^2} + \frac{\delta^2 F}{\delta y^2} = 0 .$$

C'est l'un des problèmes de base en analyse numérique. Un premier essai de 40 K portes, pour 100 Mops, réalisé sur $DecP\,eRLe_0$ par F. Rocheteau, nous apporte les enseignements suivants :

- Le schéma aux différences finies, en deux dimensions :

$$4\,F_{n+1}(x, y) = F_n(x+1, y) + F_n(x-1, y) + F_n(x, y+1) + F_n(x, y-1) \quad (1)$$

est à abandonner, en faveur du schéma, en une dimension, du paragraphe 4.3. Ce dernier divise la bande passante des mémoires par deux. Sa topologie linéaire réduit notablement la surface de l'opérateur atomique, en augmentant d'autant la profondeur du pipe-line. Son utilisation, par la méthode des directions alternées, permet de résoudre à la fois l'équation de la chaleur et celle de Laplace, en dimension arbitraire, et non seulement deux.

- L'utilisation d'une arithmétique entière, plutôt que flottante, est la clé des performances de ce type de matériel ; la précision initialement choisie de 16 bits n'est cependant pas suffisante, pour les tailles de grilles compatibles avec la vitesse du traitement ; il faut passer à 32 bits de précision ou plus (*cf.* § 4.1).

- Le prix à payer, pour simplifier l'opérateur atomique, est un pas de temps très petit, fixé par l'équation (4). Ceci est bon pour l'équation de la chaleur, mais trop lent pour celle de Laplace. On accélère la convergence, dans ce cas, par l'emploi des techniques multi-grilles du paragraphe 6.3.

Nous justifions ici les décisions d'implantation, et décrivons les résultats obtenus. Les paragraphes 4.4.1 (plomberie numérique), et 6.3 (multi-dimensions), sont les seuls où nous nous éloignons de méthodes bien connues, en analyse numérique ; pour tous les autres usages de techniques classiques (et largement élémentaires), nous renvoyons le lecteur aux ouvrages de base, principalement [DL 88].

2. ÉQUATION DE LA CHALEUR

On se propose de résoudre numériquement l'équation de diffusion de la chaleur :

$$\frac{\delta T}{\delta t} - \Delta T = 0 , \tag{1}$$

valable dans un domaine $\Omega \subseteq C_k$ intérieur au cube $C_k = [0, 1[^k$ de l'espace R^k de dimension $k > 0$ (typiquement $k = 2, 3$). Dans l'équation (1), le symbole Δ représente l'opérateur de Laplace (on dit aussi laplacien) :

$$\Delta T = \frac{\delta^2 T}{\delta x_1^2} + \cdots + \frac{\delta^2 T}{\delta x_k^2} .$$

Les températures initiales, au temps $t = 0$, sont données, en tout point $(x_1, \ldots, x_k) \in C_k$ du cube, par la condition de Cauchy :

$$T(0, x_1, \ldots, x_k) = T_0(x_1, \ldots, x_k) . \tag{2}$$

On suppose aussi connues et fixes à tout instant, les températures en tout point $(x_1, \ldots, x_k) \in \Gamma$ du complémentaire de Ω dans $C_k = \Omega \cup \Gamma$, par la condition de type Dirichlet :

$$\forall t \geqslant 0 : \quad T(t, x_1, \ldots, x_k) = T_0(x_1, \ldots, x_k) . \tag{3}$$

On conviendra, sans perdre en généralité pour la suite, que Γ est composé d'un ensemble *fini* de sources ponctuelles de chaleur (représentées mathématiquement par des masses de Dirac, au sens des distributions). On étend enfin les solutions de (1, 2, 3) à tout point de l'espace R^k par *périodicité,* soit pour $1 \leqslant j \leqslant k$:

$$T[x_j + 1] = T[x_j] .$$

Dautray, Lions [DL 88] et Feynman [FLS 63] décrivent quelques-unes des multiples applications du calcul de (1, 2, 3) en thermodynamique, neutronique, cinétique chimique, hydrodynamique, ...

Remarque : En l'absence de la condition (3), soit $\Gamma = \emptyset$, $\Omega = C_k$, les solutions de (1, 2) sont telles que :

$$\frac{\delta}{\delta t} \int_{C_k} T(t, x_1, \ldots, x_k)\, dx_1 \ldots dx_k = 0 ,$$

pour $t \geqslant 0$, ce qui exprime la conservation de la quantité initiale de chaleur.

3. RÉSOLUTION PAR DIFFÉRENCES FINIES

On se fixe un pas d'espace $\Delta x = \Delta x_1 = \cdots = \Delta x_k$, identique dans chaque dimension, et un pas de temps Δt tel que :

$$\Delta t = \frac{k}{4} \Delta x^2 . \tag{4}$$

On calcule les valeurs approchées $G_n(i_1, \ldots, i_k)$ des solutions $T(t_n, x_i, \ldots, x_{i_k})$ de (1, 2, 3), prises en $t_n = n\,\Delta t$, $x_i = i\,\Delta x$, par un schéma (explicite) aux différences finies, que l'on peut écrire (méthode des directions alternées) :

$$G_{n+1}(i_1, \ldots, i_k) = \mathscr{H}_1 \ldots \mathscr{H}_k\, G_n(i_1, \ldots, i_k)\,. \tag{5}$$

La relation (4) permet de donner à chacun des opérateurs $\mathscr{H}_j$, pour $1 \leqslant j \leqslant k$, la forme particulièrement simple :

$$\mathscr{H}_j\, G_n(i_1, \ldots, i_k) = \quad \text{si } (x_1, \ldots, x_k) \in \Gamma \quad \text{alors} \quad G_n(i_1, \ldots, i_k)$$
$$\text{sinon} \quad \frac{1}{4}\,(G[i_j + 1] + 2\,G + G[i_j - 1])\,. \tag{6}$$

La formule (5) ne dépend pas du choix du système d'axes, puisque les opérateurs (6) commutent tous, deux à deux :

$$\mathscr{H}_j\, \mathscr{H}_\ell = \mathscr{H}_\ell\, \mathscr{H}_j\,.$$

Cette propriété, dont nous ferons bon usage, traduit le caractère *isotrope* de la diffusion de la chaleur, dans le monde physique.

Remarque : Quand $\Gamma = \emptyset$, le schéma (5) conserve la quantité initiale de chaleur, pour tout $n \geqslant 0$:

$$\sum_{x_1, \ldots, x_k} G_n(x_1, \ldots, x_k) = \sum_{x_1, \ldots, x_k} G_0(x_1, \ldots, x_k)\,. \tag{7}$$

4. STRUCTURE MATÉRIELLE

Dans ce qui suit, nous décrivons une structure matérielle permettant de tirer le meilleur parti des technologies de circuits VLSI, dans le calcul des solutions de l'équation de la chaleur. Afin d'alléger les notations, on se place en dimension $k = 2$; la généralisation à $k > 2$ est traitée au paragraphe 4.5. On se propose donc de résoudre numériquement le problème :

$$\begin{cases} T(0, x, y) = T_0(x, y) & \text{pour } 0 \leqslant x, y < 1\,, \\ \dfrac{\delta}{\delta t}\,T(t, x, y) = 0 & \text{pour } x, y \in \Gamma,\ t > 0\,, \\ \dfrac{\delta T}{\delta t} = \dfrac{\delta^2 T}{\delta x^2} + \dfrac{\delta^2 T}{\delta y^2} & \text{pour } x, y \notin \Gamma,\ t > 0\,, \\ T(t, x, y) = T(t, x + 1, y) = T(t, x, y + 1) & \text{pour } t \geqslant 0\,. \end{cases} \tag{8}$$

On utilise pour cela le schéma aux différences finies :

$$\begin{cases} G_{n+1}(i,j) &= \mathscr{H}_x \mathscr{H}_y G_n(i,j), \\ \mathscr{H}_x G(i,j) &= \dfrac{1}{4}(G(i+1,j) + 2G + G(i-1,j)), \quad \text{pour } i,j \notin \Gamma, \\ \mathscr{H}_y G(i,j) &= \dfrac{1}{4}(G(i,j+1) + 2G + G(i,j-1)), \quad \text{pour } i,j \notin \Gamma, \\ G(i,j) &= \mathscr{H}_x G(i,j) = \mathscr{H}_y G(i,j), \quad \text{pour } i,j \in \Gamma. \end{cases} \tag{9}$$

4.1. Températures discrètes

Soient $I = \inf T_0$ *et* $S = \sup T_0$ les valeurs extrêmes des températures initiales dans le carré C_2. Par le principe du maximum, la solution de (8) satisfait

$$I \leqslant T(t, x, y) \leqslant S,$$

pour tout $t \geqslant 0$ et $0 \leqslant x, y < 1$; ceci est évident si l'on se reporte à l'interprétation physique du problème. On représente alors les *températures discrètes* $G_n(i, j)$ par des entiers naturels, avec la correspondance :

$$T(n\,\Delta t, i\,\Delta x, j\,\Delta y) = I + \frac{S - I}{2^b - 1} G_n(i, j).$$

Le nombre b de bits dans l'écriture binaire de ces entiers est tel que :

$$2^{-b} < \sup |T_0(i\,\Delta x + \varepsilon, j\,\Delta y + \epsilon) - T_0(i\,\Delta x, j\,\Delta y)| / T_0(i\,\Delta x, j\,\Delta y).$$

De cette façon, les erreurs dues à l'arrondi numérique sont inférieures à celles inhérentes à la discrétisation des températures initiales. Avec ce choix, les décalages liés aux exposants, dans une représentation des températures par des nombres flottants, sont factorisés une fois pour toutes, à la présentation initiale du problème. Tous les calculs ultérieurs sont faits sur la mantisse entière, ceci en vue de simplifier, sans perte de précision numérique, la réalisation matérielle de l'opérateur $\mathscr{H}$, qui suit.

4.2. Représentation du problème

La grille de travail est de taille $m \times m$, l'entier $m = 2^a$ étant une puissance de deux. Les pas correspondants d'espace et de temps sont donnés par :

$$\Delta x = \Delta y = 2^{-a}, \quad \Delta t = 2^{-(2a+1)}.$$

A l'instant $t_n = n\,\Delta t$, l'état du système est stocké dans une mémoire composée de $m \times m$ mots contigus, de $b + 1$ bits chacun. Aux b bits

représentant la température discrète, nous en ajoutons un, qui vaut 1 si le point correspondant (i, j) de la grille est une condition limite $(i\ \Delta x, j\ \Delta y) \in \Gamma$, sinon 0. En convenant de le placer en tête des poids faibles de l'écriture binaire, on réduit le test d'appartenance à Γ, à un test de parité, et on trouve, à l'adresse $i + mj$ du tableau M, la quantité :

$$M(i + mj) = \begin{cases} 1 + 2\,G_n(i, j) & \text{si } (i, j) \in \Gamma\,, \\ 2\,G_n(i, j) & \text{si } (i, j) \notin \Gamma\,. \end{cases}$$

La mémoire M est dotée d'un contrôleur, générant des suites d'adresses, qui permettent de parcourir son contenu, de deux manières :

1. suivant l'axe des x, dans un balayage horizontal, correspondant à l'ordre lexicographique $(i, j) <_x (i', j')$ si $i < i'$, ou $i = i'$ et $j < j'$;

2. suivant l'axe des y, dans un balayage vertical, correspondant à l'ordre lexicographique dual $(i, j) <_y (i', j')$ si $j < j'$, ou $j = j'$ et $i < i'$.

Le contrôleur mémoire est composé de deux compteurs binaires sur $2\,a$ bits, et d'un multiplexeur :

- Le compteur horizontal C_x génère les $2\,a$ bits d'adresse dans l'ordre $b_0 \ldots b_{a-1}\, b_a \ldots b_{2a-1}$;
- Le compteur vertical C_y inverse les a bits de poids fort, avec ceux de poids faible, dans l'ordre $b_a \ldots b_{2a-1}\, b_0 \ldots b_{a-1}$;
- Le multiplexeur connecte les adresses de la mémoire, à l'un ou l'autre des deux compteurs, suivant la nature de la passe.

Remarque : Avec un tel contrôleur, la structure topologique de l'espace de travail est naturellement celle d'un tore à deux dimensions. En remplaçant la condition de périodicité :

$$T(t, x, y) = T(t, x + 1, y) = T(t, x, y + 1)\,,$$

par une condition de Neuman sur les bords du carré :

$$\frac{\delta}{\delta x} T(t, x, b) = \frac{\delta}{\delta y} T(t, b, y), \quad \text{pour } b = 0 \text{ ou } b = 1\,,$$

on retrouve la topologie usuelle du plan euclidien. Les modifications correspondantes, dans le contrôle des compteurs C_x et C_y, sont les suivantes :

1. lors d'un balayage horizontal, la valeur de C_x doit être tenue constante (en fixant son incrément à zéro), pendant un nombre de cycles égal à la profondeur p du pipe-line des données (*cf.* § 4.3), *avant* ($C_x = -1 \pmod m$), et *après* ($C_x = 0 \pmod m$) chaque retour de ligne ;

2. lors d'un balayage vertical, la valeur de C_y doit être tenue constante, pendant p cycles, *avant* ($C_y = -1 \pmod m$), et *après* ($C_y = 0 \pmod m$) chaque retour de colonne.

4.3. Traitement à la chaîne

Pour résoudre les équations (8), nous nous dotons de :

- Une mémoire $\mathcal{M}_s$, munie de son générateur d'adresses, servant de source des données. Elle représente l'état du système au temps t_n.
- Une mémoire $\mathcal{M}_d$, servant de destination aux résultats du calcul. Elle est identique à $\mathcal{M}_s$, et contiendra, en fin de passe, l'état du système au temps t_{n+1}. La transition $n \mapsto n+1$ inverse les rôles de $\mathcal{M}_s$ et $\mathcal{M}_d$. On peut, sans obstacle théorique, représenter $\mathcal{M}_s$ et $\mathcal{M}_d$ dans la même mémoire physique ; ceci divise cependant la bande passante du traitement, par un facteur deux, puisqu'il faut lire et écrire une mémoire à chaque pas de calcul.
- Un opérateur $\mathcal{H}$, dont on trouvera la description au paragraphe 4.4 ; son objet est de transformer les données provenant de $\mathcal{M}_s$ en données à destination de $\mathcal{M}_d$.
- Un contrôle externe, qui établit le sens du transfert $\mathcal{M}_s \mapsto \mathcal{M}_d$, et la nature de la passe : horizontale ou verticale, avec ou sans multi-dimension (*cf.* § 6.3).

Voici, sous ces hypothèses, le schéma d'une itération (9), composée d'une passe horizontale suivie d'une passe verticale à travers tout l'espace :

$$\boxed{M} \mapsto \boxed{H}_x \mapsto \boxed{M} \mapsto \boxed{H}_y \mapsto \boxed{M}.$$

Il faut répéter n fois cette opération, afin d'obtenir $(\mathcal{H}_x \mathcal{H}_y)^n G_0 \Rightarrow G_n$. Par isotropie de la diffusion, nous avons :

$$(\mathcal{H}_x \mathcal{H}_y)^p G_r = \mathcal{H}_x^p \mathcal{H}_y^p G_r = G_{r+p}.$$

Ainsi, le calcul de p passes :

$$\left.\begin{array}{l} \boxed{M} \mapsto \boxed{H}_x \mapsto \boxed{M} \mapsto \boxed{H}_y \mapsto \boxed{M} \mapsto \\ \mapsto \boxed{M} \mapsto \boxed{H}_x \mapsto \boxed{M} \mapsto \boxed{H}_y \mapsto \boxed{M} \end{array}\right\} p,$$

donne le même résultat qu'une seule passe de p traitements $\mathcal{H}$ à la chaîne :

$$\boxed{M} \mapsto \overbrace{\boxed{H}_x \ldots \boxed{H}_x}^{p} \mapsto \boxed{M} \mapsto \overbrace{\boxed{H}_y \ldots \boxed{H}_y}^{p} \mapsto \boxed{M}.$$

Ceci justifie d'utiliser le schéma :

$$G_{n+p}(i, j) = \mathscr{H}_x^p \mathscr{H}_y^p G_n(i, j), \tag{10}$$

qui cumule p passes en une, avec *exactement* la même demande de *bande passante mémoire* que le schéma (9), pour une seule passe. En choisissant la plus grande valeur de p réalisable dans une technologie donnée, on adapte le calcul de chaque itération au volume de logique disponible, en déroulant la boucle interne du calcul, autant que faire se peut. La structure d'ensemble du système est donc :

$$\boxed{\begin{matrix}\mathscr{M}_s \\ \mathscr{C}_s\end{matrix}} \mapsto \overbrace{\boxed{H} \mapsto \cdots \mapsto \boxed{H}}^{p} \mapsto \boxed{\begin{matrix}\mathscr{M}_d \\ \mathscr{C}_d\end{matrix}} \tag{11}$$

Remarque : Le pipe-line des données introduit un retard de p cycles entre les adresses des sources $\mathscr{M}_s$, et celles des destinations $\mathscr{M}_d$; on compense en retardant de p cycles la mise en route des compteurs de $\mathscr{M}_d$, afin de maintenir invariantes les relations : $\mathscr{C}_x^d = \mathscr{C}_x^s + p$, $\mathscr{C}_y^d = \mathscr{C}_y^s + p$.

4.4. Opérateur atomique

L'ensemble du système (11) est synchronisé par une horloge digitale τ, de période $\Delta\tau$. Comme une étape du schéma (10) correspond à deux parcours complets des mémoires (en x et en y), on a la relation suivante entre le pas de calcul $\Delta\tau$ et le pas de temps Δt :

$$m^2 \Delta\tau = p \Delta t .$$

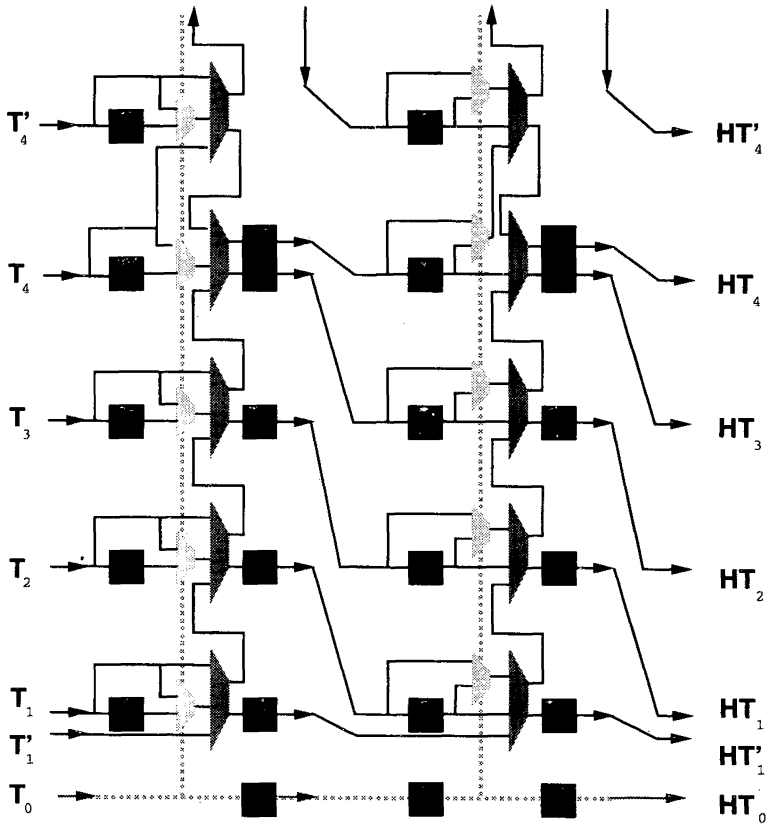
Dans une passe horizontale (resp. verticale), les entrées des opérateurs $\mathscr{K}$ arrivent dans l'ordre $<_x$ (resp. $<_y$) ; chaque opérateur $\mathscr{H}$ calcule un pas de diffusion *en une dimension,* suivant l'axe déterminé par l'ordre d'arrivée de ses entrées. On décompose $\mathscr{H} = \mathscr{H}^+ \mathscr{H}^- = \mathscr{H}^- \mathscr{H}^+$ en produit d'un opérateur $\mathscr{H}^+$ de diffusion *en avant,* par un opérateur $\mathscr{H}^-$ de diffusion *en arrière.*

• $\mathscr{H}^+$ comprend un registre $\boxed{\rho}$ et un circuit combinatoire $\boxed{\mathscr{A}^+}$, connectés comme suit :

$$G(v+1) \mapsto \boxed{\rho} \mapsto \begin{matrix} G(v) \mapsto \\ G(v+1) \mapsto \end{matrix} \boxed{\mathscr{A}^+} \mapsto \mathscr{H}^+ G(v) .$$

• $\mathscr{H}^-$ comprend un registre $\boxed{\rho}$ et un circuit combinatoire $\boxed{A}$, connectés comme suit :

$$G(v) \mapsto \boxed{\rho} \mapsto \begin{matrix} G(v-1) \mapsto \\ G(v) \mapsto \end{matrix} \boxed{A^-} \mapsto \mathscr{H}^- G(v) .$$



Dans ces schémas, les carrés sont des registres synchrones 1 bit (flip-flop), les petits losanges des multiplexeurs, et les grands des additionneurs binaires complets. La température d'entrée est $T = \sum_{i \geq 0} (T_i + T'_i)\, 2^i$, et celle de sortie :

$$\mathscr{H}T = \sum_{i \geq 0} (HT_i + HT'_i)\, 2^i .$$

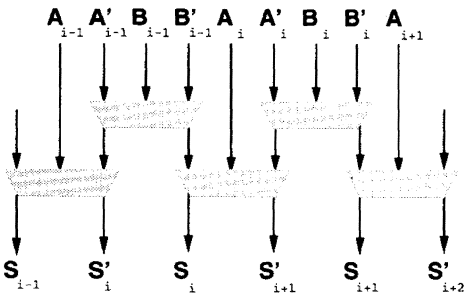
Figure 1. — Schémas de $\mathscr{H} = \mathscr{H}^- \mathscr{H}^+$, sur 4 bits.

Si $E = (E_1\, E_2 \ldots E_k \ldots)$ représente la suite des valeurs d'entrées d'un registre

$$E \mapsto \boxed{\rho} \mapsto S ,$$

la suite des sorties $S = (0\, E_1\, E_2 \ldots E_{k-1} \ldots)$ correspondantes est la même, à un retard d'une période $\Delta\tau$ près. Les circuits $\mathscr{A}^-$ et $\mathscr{A}^+$ sont identiques, après permutation des entrées :

$$\begin{cases} \mathscr{A}(v_0, v_1) = \mathscr{A}^+(v_0, v_1) = \mathscr{A}^-(v_1, v_0) , & \\ \mathscr{A}(v_0, v_1) = \dfrac{1}{2}(v_0 + v_1) , & \text{si} \quad v_0 \notin \Gamma , \\ \mathscr{A}(v_0, v_1) = v_0 , & \text{si} \quad v_0 \in \Gamma . \end{cases}$$



Ici, les losanges représentent des additionneurs binaires complets. Les températures d'entrée sont $T_a = \sum_{i \geq 0} (A_i + A'_i)\, 2^i$, $T_b = \sum_{i \geq 0} (B_i + B'_i)\, 2^i$, et celle de sortie

$$T_s = T_a + T_b = \sum_{i \geq 0} (S_i + S'_i)\, 2^i.$$

Figure 2. — Addition sans retenues, sur 2 bits.

4.4.1. *Plomberie numérique*

Afin de fixer la structure de l'opérateur $\mathscr{A}$, il convient de régler la question d'arrondi liée à la division par deux ci-dessus. Le choix naturel :

$$\mathscr{A}(v_0, v_1) = \left[\frac{v_0 + v_1}{2} \right], \quad \text{si} \quad v_0 \notin \Gamma,$$

amène des *pertes numériques de chaleur* en violation de l'invariant (7). Cette question, qui n'est pas abordée dans notre bibliographie, a pourtant une incidence significative sur les résultats numériques, dès que la taille m^2 de la grille devient grande, en regard du nombre b de bits représentant les températures. La solution la plus simple à ce problème est d'interdir le partage du bit de poids faible des températures (de poids 2 puisque le bit de parité représente l'appartenance à Γ). Ceci est réalisé par la formule arithmétique :

$$\begin{cases} \mathscr{A}(v_0, v_1) = v_0 \text{ si } v_0 \cdot | \cdot 2 = 1, \\ \mathscr{A}(v_0, v_1) = 2(v_0 \div 4 + v_1 \div 4 + (v_0 \cdot | \cdot 4) \div 2), \text{ sinon}. \end{cases} \tag{12}$$

Ici, $n \cdot | \cdot d$ désigne le reste de la division entière de n par d, et $n \div d$ le quotient. La formule (12) restaure l'invariant (7), de la quantité initiale de chaleur.

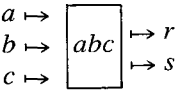
4.4.2. *Additions sans retenues*

Le pas de calcul $\Delta\tau$ est déterminé par le composant le plus lent (chemin critique) de l'ensemble du système synchrone (11). Notre objectif est de

régler $\Delta\tau$ sur la vitesse nominale des mémoires. Nous montrons en [V 91] comment réaliser des compteurs binaires capables d'opérer à la vitesse intrinsèque (toggle rate) de la technologie de réalisation, pour un nombre de bits *arbitraire*. Ceci résout la question des contrôleurs d'adresse.

Pour régler celle des additionneurs $\mathscr{A}$, on utilise la technique classique (carry save adder) des représentations *binaires redondantes*. A l'intérieur du chemin des données, chaque température T est représentée par la somme $T = R + S$ de deux nombres binaires R et S. L'addition de deux températures redondantes $T + T' = R + R' + S + S'$ se calcule alors bit à bit, *sans propagation de retenue,* comme dans les schémas de la Figure 2, qui sont réalisés exclusivement à partir d'additionneurs binaires complets $\boxed{\text{abc}}$.

Le circuit combinatoire



calcule l'unique solution binaire $a, b, c, r, s \in \{0, 1\}$ de l'équation :

$$a + b + c = s + 2\,r\,.$$

Avec cette technique, le délai critique de l'opérateur $\mathscr{H}$ est double de celui de $\boxed{abc}$, indépendamment du nombre de bits b ; on le rend facilement inférieur à celui des mémoires. L'inconvénient de la méthode est de doubler la surface nécessaire à la réalisation de $\mathscr{H}$, en comparaison d'une propagation combinatoire des retenues sur b bits (voir la remarque ci-dessous).

En fin du chemin des données, après les p opérateurs $\mathscr{H}$, on doit ajouter un unique additionneur combinatoire (voir par exemple [GV 82] pour une structure de délai $2 \log_2 b$), qui convertit la représentation redondante $R + S$ en un entier binaire unique $T = R + S$, pour écriture dans la mémoire $\mathscr{M}_d$. On rendra ce circuit plus rapide que les mémoires, en coupant son chemin critique par un nombre suffisant de registres binaires (au plus $\log_2 b$). La surface totale de cet unique élément est négligeable devant celle des p opérateurs $\mathscr{H}$, pour p suffisamment grand ; le nombre r de registres, mis en série dans cet additionneur, augmente d'autant la longueur totale du pipe-line, qui devient $p + r$.

Remarque : En pratique, il est inutile de pousser la représentation redondante jusqu'au niveau du bit. Découpons, en effet, $b = cd$ en d blocs de c bits ; la propagation des retenues est combinatoire, sur c bits, à l'intérieur de chaque bloc ; d bits redondants suffisent alors à casser les chaînes de retenues, en choisissant pour c la valeur maximale compatible avec le débit des mémoires. Cette remarque réduit la surface totale d'un tel opérateur par un facteur proche de deux, sans perte en vitesse pour l'ensemble du système. Les schémas de la figure 1 montrent une réalisation, sur $c = 4$ bits ; ils illustrent aussi la réalisation de la formule (12).

4.5. Autres dimensions

Pour appliquer notre système à la résolution de l'équation de la chaleur (1, 2, 3) en $k > 2$ dimensions, il suffit d'adapter les contrôleurs des mémoires ; le chemin des données (formé de p opérateurs $\mathscr{H}$) reste inchangé. Chaque contrôleur comprend alors k compteurs sur $a = ka'$ bits. Ainsi, en $k = 3$ dimensions, on arrange les 3 a bits de chaque compteur dans l'ordre :

- $b_0 \ldots b_{a'-1}\, b_{a'} \ldots b_{2a'-1} \ldots b_{2a'} \ldots b_{3a'-1}$ pour $\mathscr{C}_x$;
- $b_{a'} \ldots b_{2a'-1}\, b_{2a'} \ldots b_{3a'-1}\, b_0 \ldots b_{a'-1}$ pour $\mathscr{C}_y$;
- $b_{2a'} \ldots b_{3a'-1}\, b_0 \ldots b_{a'-1}\, b_{a'} \ldots b_{2a'-1}$ pour $\mathscr{C}_z$.

On obtiendra, bien entendu, exactement le même résultat en échangeant arbitrairement les a' bits de poids fort de l'un de ces compteurs, avec les a' bits du milieu. La généralisation à $k > 3$ est directe, bien que peu réaliste, vu les tailles présentes des mémoires ; la méthode des éléments finis prend ici le pas sur celle des différences finies.

5. EQUATION DE LAPLACE

Le cas *statique* de l'équation de la chaleur, dite équation de Laplace (aussi de d'Alembert), est particulièrement intéressant, par le nombre de lois de la physique qui se réduisent à l'équation :

$$\Delta U = 0\,, \tag{13}$$

sujette à la condition de Dirichlet :

$$U(x_1, \ldots, x_k) = U_0(x_1, \ldots, x_k)\,, \tag{14}$$

valable en tout point $(x_1, \ldots, x_k) \in \Gamma$ d'un ensemble discret Γ. Nous résolvons (13, 14) en prenant le *cas limite,* pour $t \mapsto \infty$, de l'équation de la chaleur (1, 2, 3). Soit, en effet, $T(t, x_1, \ldots, x_k)$ une solution à l'équation (1) de la chaleur, obtenue à partir d'un état initial (2) *arbitraire,* compatible avec la condition de Dirichlet (3), que l'on prend identique à (14). La fonction

$$U(x_1, \ldots, x_k) = T(\infty, x_1, \ldots, x_k)$$

est solution des équations (13, 14) ; cette solution est *unique,* indépendamment de l'état initial, pourvu que $\Gamma \neq \emptyset$ soit non vide.

Dautray, Lions [DL 88] et Feynman [FLS 63] décrivent quelques applications du calcul de (13, 14) en électrostatique, électromagnétisme, optique, élasticité, mécanique, hydrostatique, etc.

6. VITESSE DE CONVERGENCE

L'itération (5) $\mathscr{H}_1 \ldots \mathscr{H}_k G_t \Rightarrow G_{t+1}$ converge, pour $t \mapsto \infty$, vers une solution $F = G_\infty$ indépendante de l'état initial G_0, pourvu qu'il soit compatible avec la condition de Dirichlet (14), supposée non vide. Le choix de l'état initial affecte cependant la *vitesse de convergence,* i.e. le nombre minimal $T = T(\varepsilon)$ de pas d'itération ($t \mapsto t + 1$) nécessaire pour obtenir une solution G_T de (13, 14), approchée à $\varepsilon > 0$ près :

$$|\Delta F| \approx |G_{T+1} - G_T| < \varepsilon .$$

En général, le processus devient numériquement stable, à $\varepsilon = \Delta x$ près, après un nombre de pas d'itération (5) de l'ordre de $m = \Delta x^{-1}$. Le nombre total d'opérations nécessaires à la résolution du problème (13, 14) de Laplace, en prenant la limite de l'évolution des équations (1, 2, 3) de la chaleur, est donc proportionnel à m^{k+1}. Les techniques qui suivent permettent de réduire ce temps de convergence, par un choix approprié de l'état initial.

On se limite ici, pour simplifier les notations, au problème de Laplace en deux dimensions, sur une grille de taille m^2. La complexité du processus itératif est $O(m^3)$, dans ce cas.

6.1. Multi-grilles

L'idée générale est de choisir, pour état initial sur une grille fine $m \times m$, une injection de la solution au même problème de Laplace, calculée récursivement sur une grille grossière $m' \times m'$, avec $m' < m$. La description la plus simple de cet algorithme vient en prenant $m = 2m'$:

1. Les m^2 données initiales $G_0(i, j)$ sont projetées sur la grille $m'^2 = \dfrac{m^2}{4}$, suivant la formule :

$$G'_0(i', j') = \frac{G_0(i, j) + G_0(i+1, j) + G_0(i, j+1) + G_0(i+1, j+1)}{4} ,$$

avec $i = 2i'$ et $j = 2j'$. On décide de l'appartenance $(i', j') \in \Gamma'$, aux points de Dirichlet de la grille grossière, par : $(2i', 2j') \in \Gamma$ ou $(2i'+1, 2j') \in \Gamma$ ou $(2i', 2j'+1) \in \Gamma$, ou enfin $(2i'+1, 2j'+1) \in \Gamma$.

2. On résout récursivement le problème de Laplace, sur les m'^2 points de la grille grossière. La récursion se termine par un calcul itératif, quand la taille de grille est suffisamment petite : $m < m_0$.

3. On injecte la solution $G'_\infty(i', j')$ ainsi obtenue pour la grille m'^2, sur la grille fine :

$$\begin{cases} G_1(i, j) = G'_\infty(i \div 2, j \div 2) & \text{pour } i, j \notin \Gamma , \\ G_1(i, j) = G_0(i, j) & \text{pour } i, j \in \Gamma . \end{cases}$$

4. On termine le calcul itérativement, en appliquant le schéma (9) jusqu'à la convergence, à partir de l'état initial $G_1(i, j)$ calculé à l'étape précédente.

Une analyse simple montre que le nombre total d'opérations $\frac{4}{3}m^2 = m^2 + \frac{1}{4}m^2 + \frac{1}{16}m^2 + \cdots$ de cet algorithme, est proportionnel à la taille m^2 du problème initial.

6.2. Multi-échelles

L'algorithme multi-grilles s'adapte mal au matériel présenté en Section 4 :

- Le temps de projection (étape 1) et d'injection (étape 3) est égal à celui $m^2 \Delta\tau$ d'un parcours complet de la mémoire ; ces opérations nécessitent, de plus, un matériel spécifique.
- La résolution récursive (étape 2) nécessite $\log_2 m$ compteurs d'adresse, nombre qui croît arbitrairement avec la taille m de la grille.

L'algorithme multi-échelles (parallel superconvergent multigrid de [FMcB 88]) élimine la projection et l'injection du multi-grilles, en résolvant récursivement quatre problèmes de Laplace, au lieu d'un seul pour le multi-grilles, sur les grilles grossières (0, 0), (0, 1), (1, 0) et (1, 1), obtenues en considérant (i, j) (mod 2).

Cette méthode converge plus vite (voir [McB *et al.* 90]) que le multi-grilles. Chaque étape récursive traite un volume de données égal à celui m^2 du problème initial, pour un total proportionnel à $m^2 \log_2 m$ opérations. Cet algorithme est donc plus lent que le précédent, sur des machines séquentielles, ou à faible parallélisme. Il se montre pourtant plus rapide, sur des machines hautement parallèles, qui tirent parti des avantages que le multi-échelles possède sur le multi-grilles :

- nombre total de passes plus faible, dû à sa meilleure convergence ;
- simplicité du contrôle, en particulier lors des transitions entre passes ;
- volume constant des calculs pour chaque passe, qui permet d'exploiter un parallélisme arbitraire.

Il ne reste plus, pour adapter l'algorithme multi-échelles au matériel décrit en Section 4, qu'à en éliminer la récursivité.

6.3. Multi-dimensions

Si on élimine la récursion, le multi-échelles revient à résoudre, d'abord, un problème de Laplace sur le parallélépipède $\frac{m}{2} \times \frac{m}{2} \times 4$, en trois dimensions ; le résultat de ce calcul est alors injecté comme valeur initiale d'une résolution itérative sur le carré m^2. C'est la structure de l'algorithme multi-dimensions qui suit, après optimisation du choix des dimensions, dans la première étape.

Pour le décrire, remarquons qu'il existe une *bijection naturelle* entre le carré m^2 et le cube en quatre dimensions $\sqrt{m^4}$. En posant $m = 2^{2^a}$, donc $r = \sqrt{m} = 2^a$, le point générique $(i, j) \in \mathscr{C}_2$ du carré m^2 est mis en correspondance avec celui $(u, v, w, x) \in \mathscr{C}_4$ du cube r^4 par la formule :

$$i = u + rv\ , \quad j = w + rx\ ;$$

et réciproquement par :

$$u = i \cdot | \cdot r,\ v = i \div r,\ w = j \cdot | \cdot r,\ x = j \div r\ .$$

Le problème de Laplace, en dimension deux, se résout alors en deux passes :

1. On calcule itérativement l'évolution de l'équation de la chaleur en *quatre* dimentions, pour un nombre d'étapes de l'ordre de $\sqrt{m}$; le cube r^4 de départ est en correspondance naturelle avec la grille initiale m^2, pour $m = 2^{2^a}$, $r = \sqrt{m} = 2^a$.

2. On calcule itérativement l'évolution de l'équation de la chaleur en *deux* dimensions, à partir de l'état calculé à l'étape précédente, par la correspondance naturelle.

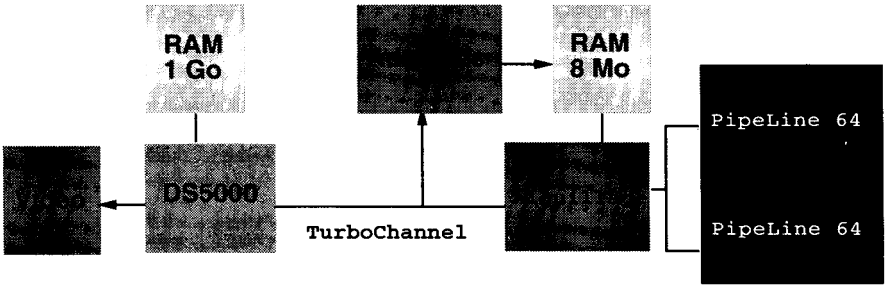


Figure 3. — Structure du système.

Nous conjecturons que le nombre de passes nécessaires à l'étape 2 du multi-dimensions est de l'ordre de $\sqrt{m}$, ce qui donnerait à cet algorithme une complexité en $O(m^2 \sqrt{m})$. Cette hypothèse est en accord avec les essais expérimentaux, ainsi que les analyses que nous avons pu mener, dans quelques cas particuliers.

Remarque : On peut, bien entendu, itérer plusieurs fois la technique multi-dimensions ; à la limite, on retrouve le multi-échelles. Ceci n'est cependant pas nécessaire, vu les tailles de grilles réalisables, avec les mémoires actuelles.

7. RÉALISATIONS CONCRÈTES

Nous présentons ici quelques caractéristiques des matériels que nous avons réalisés, au moyen de la technologie PAM. Si l'utilisation de *composants configurables* s'est révélée indispensable dans la mise au point des compromis logiciel/matériel auxquels nous sommes arrivés, les résultats de cette étude sont directement applicables aux technologies, plus classiques, de circuits prédiffusés (ASIC gate array) et à façon (full custom VLSI). L'utilisation de circuits spécifiques permettra de réduire, à performance constante, le volume et le prix unitaire du matériel nécessaire.

L'ensemble du système (hors mémoires) correspond à un prédiffusé de 180 K portes, ou un circuit à façon de 800 K transistors, doté de 256 pattes de communication.

Les températures sont représentées sur 25 bits, soit 24 bits utiles, et un bit de parité pour décider de l'appartenance à Γ. Ce choix permet des largeurs m de grilles jusqu'à $m = 2^{24} \approx 8\,M$, suffisant en pratique.

7.1. Structure d'ensemble

Le système de la Figure 3, se compose de :

- Un ordinateur hôte (DS5000, station 25 MIPS) qui contrôle l'ensemble (nature et sens des passes). Il permet aussi la visualisation en temps réel (de 5 à 20 images $512 \times 512 \times 8$ par seconde) de l'état interne du système. En mode 3D, les RAM de l'hôte servent de mémoire de données, ce qui permet de traiter des cubes de 256^3 échantillons.
- L'hôte est relié à la PAM par un lien TurboChannel, à haut débit (100 M octets par seconde).
- La partie PAM, réalisée sur $DecPeRLe_1$, comprend :

— 4 M octets de RAM, utilisés en mode 2 D pour stocker quatre images de 512^2 échantillons.

— Un générateur d'adresses, qui comprend les 8 compteurs nécessaires au multi-échelle, en 2 D.

— Un chemin des données, composé de deux pipe-lines de 64 opérateurs $\mathscr{H}$ chacun. Chaque élément $\mathscr{H}$ opère sur des entiers 25 bits, organisés en 6 blocs de 4 bits, avec un bit de redondance par bloc ; le bit de parité (appartenance à Γ) est dupliqué, par raison de symétrie, pour un total de 32 bits par opérateur. Cet ensemble correspond à 150 K portes logiques, soit plus de 80 % du total.

— Un aiguilleur sert à sélectionner la source des données (hôte en 3 D et PAM en 2 D), et à connecter les deux moitiés du pipe-line, en parallèle, ou

en série. Il comprend en outre les deux additionneurs, qui assurent la conversion en format non redondant, à la fin des pipe-lines.

7.2. Performances

Tout le système opère à la fréquence, 20 MHz, des mémoires, pour calculer 128 opérations $\mathscr{H}$, à chaque pas : $\Delta r = 50$ ns.

En optimisant le code machine, pour le jeu d'instructions MIPS, nous arrivons à un total de 10 instructions par opération $\mathscr{H}$, compte tenu des conditions de Dirichlet, de la formule (12), et de la gestion des adresses. Un ordinateur séquentiel classique devra donc exécuter 25 G instructions par seconde, pour reproduire le calcul de notre matériel. Notre co-processeur PAM, double le volume de son hôte ($DecPeRLe_1$ se loge dans la même boîte que le DS5000), et multiplie ses performances par mille, pour ce problème particulier.

En ne comptant, maintenant, que l'arithmétique dans (12), on arrive à 5 G opérations effectives par seconde, addition ou décalage en virgule fixe ; avec, en prime, conservation de la quantité initiale de chaleur (7), et prise en compte des conditions de Dirichlet. Ceci dépasse les puissances de calculs mesurées par [McB 88] et [McB *et al.* 90], qui donnent la Connection Machine CM-2 à 3,8 Gflop, et le CRAY Y-MP à 0,7 Gflop.

D'après [BKH 88], la mesure en $ par Mflop, de ces super-ordinateurs est de 3 K$/Mflop pour la Connection Machine CM-2, et 10 K$/Mflop pour le CRAY Y-MP. En estimant, sur la base (discutable) des 20 prototypes construits, le prix de $PeRLe_1$ à 100 K$, nous trouvons 25 $/Mop, soit un gain de deux ordres de grandeur sur le prix de la résolution des équations de Laplace et de la chaleur.

8. CONCLUSION

Nous n'avons, bien entendu, abordé qu'un aspect partiel, et très simple, de l'analyse numérique contemporaine. Il reste à comprendre, et c'est loin d'être immédiat, comment appliquer cette technologie à des méthodes plus générales (diffusion avec second membre, milieu anisotrope, éléments finis), et autres équations de la physique (ondes, transport, Navier-Stokes, Schrödinger).

Les résultats obtenus sont cependant positifs : ils tracent un chemin direct menant à des solveurs numériques spécialisés, dont l'unité de puissance est le Top/sec, mille fois plus rapide que les super-ordinateurs actuels. C'est techniquement faisable, en quelques années, et sur un petit budget.

Les analystes numériciens sauront-ils exploiter des machines dont les possibilités algorithmiques sont aussi contraintes que celles que nous

proposons ? Notre démarche va clairement à l'encontre de la tendance actuelle qui est d'utiliser des machines massivement parallèles.

Citons, à ce propos, l'opinion du rapporteur anonyme, que nous en profitons pour remercier :

« First, the data-parallel approach taken in the CM-2 and some other SIMD computers is in essence not too different from the one that is proposed. Some people even view the CM-2 as the « the realworld » implementation of systolic arrays-although that is a little extreme. Specialized computers have basically failed commercially (SAXP, FPS's). The performance improvements of networks may make specialized computers a little more attractive because of their cost effectiveness. I believe though that is difficult to compete against machines such as the Connection Machine line because of the progress in software (more than hardware) that is currently being made. »

Nous répondrons simplement que, le jour où le domaine sera suffisamment mûr pour que les algorithmes de base en soient figés, et nous convenons bien volontiers que ce n'est généralement pas le cas aujourd'hui, le type d'approche que nous proposons a des avantages économiques et techniques auxquels il sera difficile de résister.

REMERCIEMENTS

Ce travail résulte d'une proposition originale de J. P. Quadrat, dont la première implantation a été réalisée par F. Rocheteau.

Sans l'aide des outils de CAO mis au point, à ce propos comme d'autres, par P. Bertin et F. Rocheteau, la réalisation expérimentale n'aurait pas été possible.

Merci à J. Barraquand et O. Pironneau, pour leurs contributions. Merci, enfin, à J. L. Lions et au rapporteur anonyme pour leurs critiques constructives.

REFERENCES

[BKH 88] S. BERSHADER, T. KRAAY, J. HOLLAND 1989, The Giant Fourier Transform, In *Scientific Applications of the Connection Machine, World Scientific,* 129-114.

[BRV 89] P. BERTIN, D. RONCIN et J. VUILLEMIN 1989, Introduction to Programmable Active Memories, *Systolic Array Processors,* J. McCanny, J. McWhirter, E. Swartzlander Jr., editors, pages 300-309, Prentice Hall.

[BRV 92] P. BERTIN, D. RONCIN et J. VUILLEMIN 1992, Programmable Active Memories : the Coming of Age, Submitted to *IEEE Trans. on Comp.*

[DL 88] R. DAUTRAY, J. L. LIONS 1988, *Analyse mathématique et calcul numérique, pour les sciences et les techniques,* en 9 volumes, Masson.

[FLS 63] R. P. FEYNMAN, R. B. LEIGHTON, M. SANDS 1963, *The Feynman lectures on Physics,* 3 volumes, Addison-Wesley.

[FMcB 87] P. O. FREDERICKSON and O. A. MCBRYAN 1987, *Superconvergent multigrid methods,* Cornell Theory Center Preprint.

[GK 89] J. P. GRAY, T. KEAN 1989, Configurable hardware : two case studies of micrograin computation, *Systolic Array Processors,* J. McCanny, J. McWhirter, E. Swartzlander Jr., editors, pages 310-319, Prentice Hall.

[GV 82] L. GUIBAS, J. VUILLEMIN 1982, On fast binary addition in n-MOS technologies, In *Proc. of IEEE ICCC 82 Conference,* New York, pages 147-151.

[L 87] D. P. LOPRESTI 1987, P-NAC : A systolic array for comparing nucleic acid sequences, *Computer Magazine 20* (7), 98-99.

[McB 88] O. A. MCBRYAN 1989, Connection machine application performance, In *Scientific Applications of the Connection Machine,* World Scientific, pages 94-114.

[McB *et al.* 91] O. A. MCBRYAN, P. O. FREDERICKSON, J. LINDEN, A. SCHÜLLER, K. SOLCHENBACH, K. STÜBEN, C.-A. THOLE, U. TROTTENBERG 1991, Multigrid methods on parallel computers — a survey of recent developments, *Impact of Computing in Science and Engineering, Vol. 3* (1), 1-75.

[V 91] J. E. VUILLEMIN 1991, Constant time arbitrary length synchronous binary counters, In *Proc. of 10th IEEE Symposium on Computer Arithmetic,* 301-309.

[X 87] XILINX 1987, *The Programmable Gate Array Data Book,* Product Briefs, Xilinx, Inc.