

状態方程式解析のための 並列型スーパーコンピュータ

岡村 総一郎

黒川 一夫

(東京理科大学 工学部 電気工学科)

1. まえがき

自然現象を解明したり、またそのシミュレーションを行なおうとする場合には、現象を記述する偏微分方程式を解かなければならない。これは、開発当初から電子計算機に与えられた使命の1つであったが、最近の処理能力の急速な発達と、有限要素法のような強力な解析法の出現により実現可能となってきた。しかし、それにつれてより詳しい現象の解明をどの要求も増大してきており、その場合には計算量は飛躍的に増加する。そして、この傾向は今後益々強まるものと思われる。

これに対し、現在商業的に成功を取めているスーパーコンピュータは、より高速の素子を用い、複数のパイプライン演算器を装備することによりベクトル処理能力の向上を図っている。したがって、汎用性の点では優れているが、常時最大性能を発揮することは難しく、この方式では今後期待できる回路素子の動作速度の向上（配線技術を含む）を見込んで、先の要求には応じ切れない。^{1) 2) 3)} また、データフロー計算機の研究も進められているが、演算は依然として2変数に対して行なわれるのみで、多数の変数を加算する場合にはそれだけ時間を要する。

一方、近年の半導体集積技術の進歩の結果、ある程度の機能を有する処理要素（Processing Element; PE）を1チップ上に多数搭載（VLSI化）することが可能となってきた。これにより、大幅な並列性導入への道が開け、各所で新しい形態のプロセッサ・アレイが提案されている。^{2) 4) 5)} 今後はこのように、計算機の形態は集中・大型・汎用から分散・小型・単機能へと移りゆくものと思われる。

本稿では、こうした観点に立ち、有限要素法により過渡現象を解析する場合等に現われる大次元の状態方程式を高速に処理することを目標として開発中の並列型計算機 PASE (Parallel Analyzer for State-space Equations) のアーキテクチャについて述べる。

2. 設計概念

解くべき状態方程式は次式で与えられる。

$$\dot{x} = A \cdot x + f \quad (1)$$

ここで、 x : 状態変数

A : システム・マトリクス

f : 外力項である。

アナログ・コンピュータは以下に示すような特徴を有し、こういった状態方程式の解析等には本質的に優れている。

- i) すべての変数に対して処理が並列に進行するので、演算時間は状態方程式の元数には依存せず一定となる。
- ii) 多入力の変数が一時に行なわれる。
- iii) 積分が数値積分でない。

しかし、その反面、

- iv) 電圧・時間のスケールリングが必要である。
- v) プログラムの保存性に乏しい。

といった欠点も有しており、使いづらい面がある。これに対しては、プログラムの編集・保存、スケールリング等をデジタル・コンピュータで行ない、その情報をもとにポットの設定や要素間の結線を自動的に行なうハイブリッド・システムが開発⁶⁾されており、通常の使用に対しては使い易いものとなっている。

しかし、有限要素法による解析においては、

(1)式の元数は一般に大きなものとなり、 $10^4 \sim 10^5$ にも及ぶことがある。(但しこの時、システム・マトリクス A は一般にスパースなマトリクスとなり、その1行あるいは1列あたりの非零要素の数は最大でも200~300程度である。)したがってこの場合、演算器としてアナログ素子を用いるアナログ・コンピュータやハイブリッド・コンピュータでは、ファンアウトやダイナミックレンジが不足し、更に誘導の問題等のために事実上解くことが不可能となる。これに対し、通常のいわゆるノイマン型のデジタル・コンピュータでは、そういった問題は解消されるが、各状態

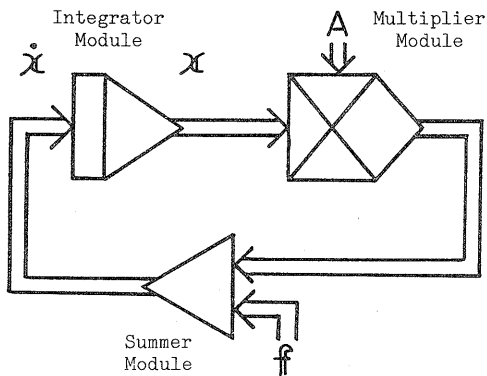


図1 PASEの基本構成

変数に関して逐次計算を行なっていくため、その処理には膨大な時間を必要とし、応答性は著しく低下する。

そこでPASEでは、まずアナログ・コンピュータのセットアップの概念を取り入れ、デジタル素子により構成されたPEを状態方程式の元数と同じ数だけ用意し、すべての状態変数に対して並列に処理を行なう。また、アナログ・コンピュータの加算の並列（同時）性にも注目し、部分的にアナログ加算を行なう多入力加算器PAMIを採用して積和の計算を一時に行なう。

以上により、1つのブランチの構造は比較的簡単となり、並列度を高くすることができる。そして、演算時間は状態方程式の元数には依存せず、1変数の場合と同様になる。そのため、1つ1つのPEはさほど高速でなくても、全体としては良好な応答性が得られる。PASEの基本的な構成を図1に示す。

3. 並列積和計算法

(1) 式を解く場合に一番時間を要するのは、 $A \cdot x$ というマトリクスとベクトルの乗算である。これは要素ごとに見れば、次式で表わされるような単なる積和の計算となる。

$$\dot{x}_i = \sum_{j=1}^n a_{ij} \cdot x_j \quad (2)$$

したがって、この計算を高速に行なうことが直接応答性の向上へとつながる。ここでは、多入力加算器PAMIを用いて(2)式の計算を一時に行なう方法について述べる。

3.1 多入力加算器PAMI⁷⁾

一般に、3つ以上の入力の和を高速に計算する加算器をデジタル素子のみで構成することは、そのキャリー処理の煩雑さから非常に困難である。これに対し多入力加算器PAMIでは、SDA (Summer & D-A converter)で入力を一度アナログ信号に変換すると同時にアナログ量としてすべて同時に加算し、それをA-D変換器により再びデジタル値に変換することにより多入力の並列加算を実現している。このため、回路構成は比較的簡単となり、その演算速度は入力数に依存せず、しかも高速である。但し、実際にはA-D変換器の分解能が有限であるため、入力数やそのbit数には制限がある。PAMIの具体的な構成例を図2に示す。これは17個の4bit 2進数を加算して8bitの結果を得るものである。このPAMIを使えば、例えば17個の24bit 2進数を加算する場合にも、図3に示したように全体を6つのブロックに分け、各ブロック毎にPAMIにより加算を行ない、得られた結果を今度は通常のデジタルの加算器により加算することで最終結果が得られる。

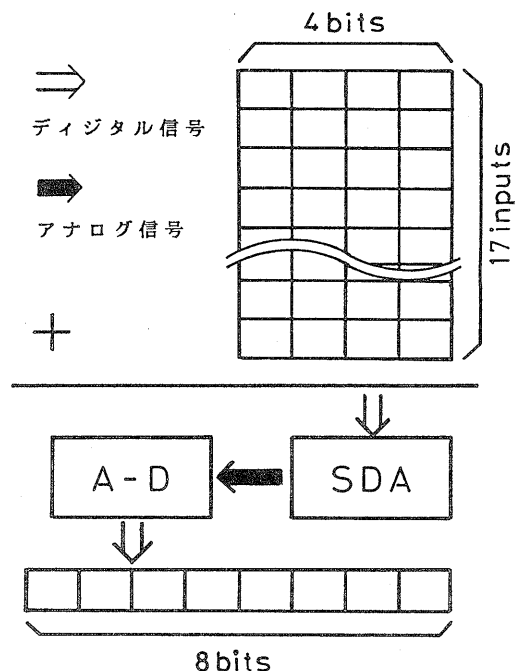


図2 多入力加算器PAMI

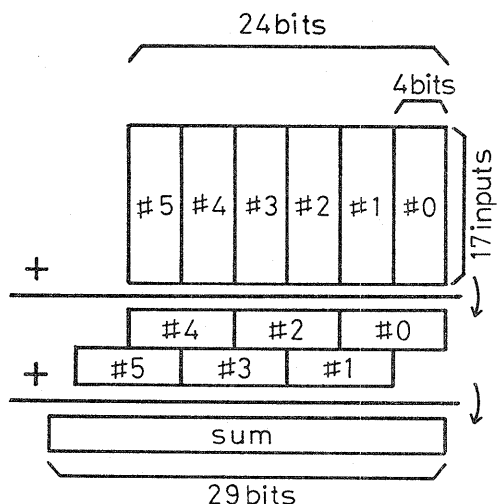


図3 多入力の並列加算

また、PAMIを用いれば積和の計算も一時に行なうことができる。すなわち、乗算は単なる部分積の加算とみなすことができるので、まずPM(Partial Multiplier)により各部分積を作りだし、それらをPAMIにより加算すればよい。この場合にも入力数やそのbit数により全体をいくつかのブロックに分割する必要がある。PAMIを用いた並列積和計算器のブロック図を図4に示す。ここでは信頼性向上のため、エラー補正回路を付加してある。これは、デジタル回路により計算結果のLSDのみを求め(これをLSD*とする)、これをA-D変換器の出力のLSDと比較し、両者が一致しなければ±1の補正を行なうものである。LSDのみの計算はキャリーの処理が必要でないので比較的簡単に行なえる。

3.2 浮動小数点数への対応

以上述べたことは、入力が整数又は固定小数点数の場合についてであり、浮動小数点数に対しては計算の前後に以下に述べるような処理を行なう必要がある。

- i) 各 a_i と x_i の指数部を加算し、それを exp_i とする。
- ii) その中から最大のものを見つけ出し、それを EXP_{max} とする。
- iii) EXP_{max} と各 exp_i の差を求め、それらをそれぞれ d_i とする。
- iv) d_i の数だけ各 x_i の仮数部を右方向にシフトする。

- v) こうして得た x_i の仮数部と a_i の仮数部との積和の計算を、3.1で述べた方法に従い一時に行なう。
- vi) 得られた値と EXP_{max} から正規化された結果を求める。

この方法をプレ・シフト法と呼ぶ。この場合、精度を最大限に保つため、 x_i をシフトする時に何bitかのガードビットを用意する必要がある。

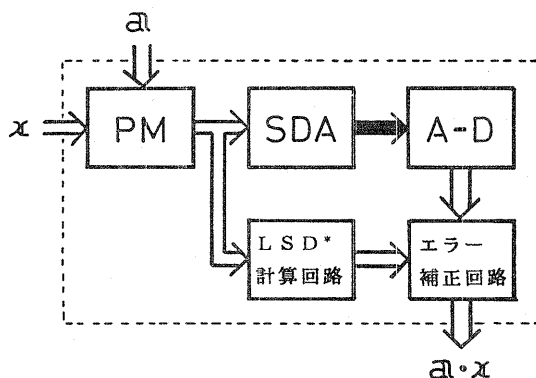


図4 固定小数点並列積和計算器

4. システム構成⁹⁾

PASEの全体のブロック図を図5に示す。その信号の流れは以下の通りである。まず、積分器Jから出力された変数 x_i はPD(Photo Distributor)に入り、ここで必要なファンアウトが稼がれる。そして、この出力はBS(Branch Selector)によりその変数を必要とする積和計算器に接続される。積和計算器では3.1で述べた方法により積和の計算を一時に行なう。すなわち全体として見れば、ここで $A \cdot x$ の計算が行なわれることになる。そしてその結果は積分器に戻され、積分器ではそれに f_i を加算したものを \dot{x}_i として数値積分を行ない、新しい x_i を求める。計算終了時刻まで以上の操作を繰り返す。

計画としては約10万個の要素を用意し、これを1ブロック255個の要素で構成される393個のブロックに分割する。そして、各ブロックにはデータの入出力を司るI/Oプロセッサを1台用意し、これをスーパーバイザのシステム・バスに接続する。その様子を図6に示す。スーパーバイザはPASEでの計算が開始され

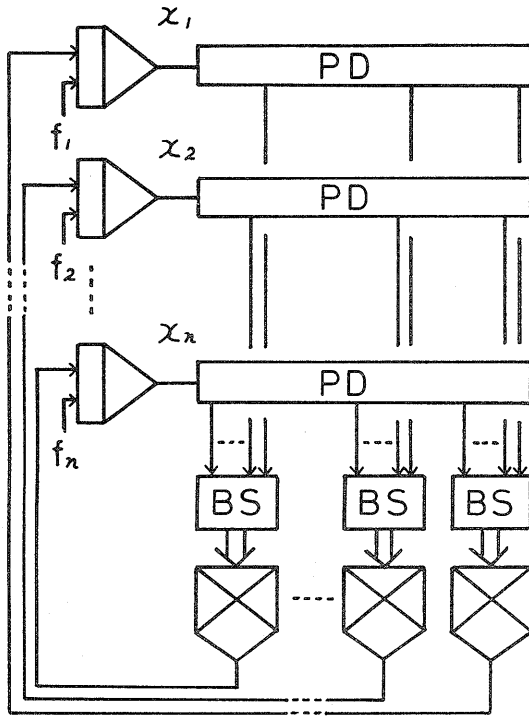


図5 PASEのブロック図

る前に、マトリクスの各係数、外力項の値及び状態変数の初期値をPASEに送り込む。そしてPASEが計算中の際は計算には直接関与せず、

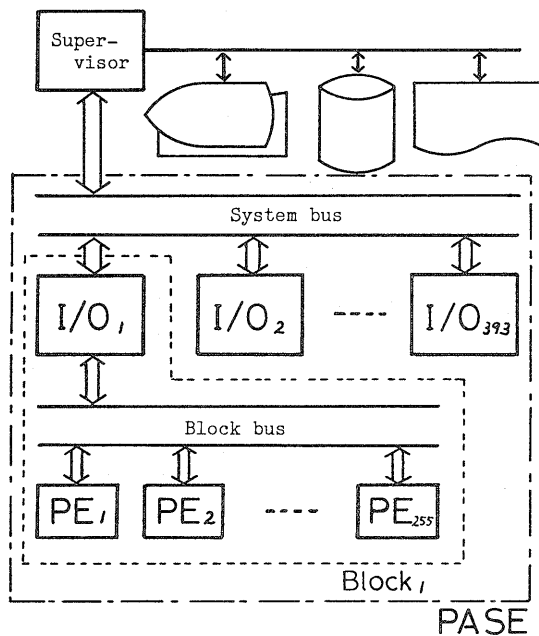


図6 PASEのシステム構成

計算終了後その結果をI/O プロセッサを介して取り出し表示する。以下に各部の詳細な構造について記す。

4.1 積分器 (J)

その構造を図7に示す。これは一種のPEであり、各々がそのメモリ中に格納された命令を順次解釈・実行していく。したがって、プログラムの与え方により各種の数値積分を行ったり、その他の操作を行ったりすることが可能である。その命令セットの主なものを表1に示す。また、これをIntelのマイクロプロセッサ8086+8087(8MHz)で構成した場合の各命令の実行時間も併せて記す。⁹⁾ 但し、データはIEEE標準規格32bit単精度フォーマットとする。これより、例えば $x_i + h \cdot \dot{x}_i$ の計算を行なうには $25 \mu s$ の時間を要すると予想される。

図7において、ローカルバスは各PEに固有なものであり、ブロックバスは同一ブロックに属するすべてのPEに共通のものである。I/O プロセッサはこのブロックバスに接続され、各PEとのデータの入出力はそのPEのメモリをブロックバス側に接続し、I/O プロセッサから直接アクセスすることにより行なわれる。

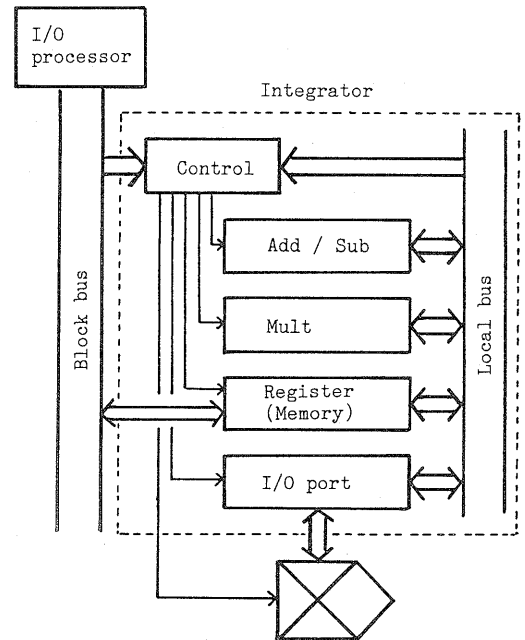


図7 積分器の構成

表1 PEの命令セット

命令	内容	実行時間(μs)
MOV	データの移動	1.5
ADD	加算	10.7
SUB	減算	10.7
MUL	乗算	12.2
CMP	比較	5.9
JMP	ジャンプ	-
STR	データの格納	11.3
IN	ポート入力	5.8
OUT	ポート出力	11.3
SYNC	同期をとる	-

4.2 フォト・ディストリビュータ (PD)

システム・マトリクスAの1行あるいは1列あたりの非零要素の数の最大値が300程度であるため、 $A \cdot x$ の計算を行なう場合には、状態変数 x の1つの要素 x_i は300箇所の積和計算器で使われる可能性がある。また、次で述べるように、BSによりその出力を適当な箇所に接続できるようにしておくため、ここでは1つの信号を約1000程度に分配する必要がある。しかし一般のTTLでは、そのファンアウトは最大でも20程度であるため、この仕様を満足しない。また、C-MOSを使ってファンアウトを稼ぐと応答性が低下する。そこで、このPDでは、1つの発光素子に対して複数の受光素子を用意し、光により信号を分配する。その構造を図8に示す。受光素子の大き

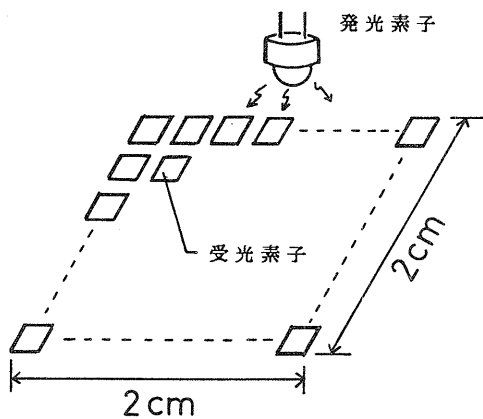


図8 フォト・ディストリビュータ

さを0.5mm角とすると、2cm四方のなかに約1600個の素子を取めることができる。これにより、応答性を低下させることなくファンアウトを稼ぐことができる。

4.3 ブランチ・セレクタ (BS)

システム・マトリクスAは多くはバンド・マトリクスとなるが、有限要素法における要素分割の仕方及び節点番号の割り付け方によっては完全なバンド・マトリクスとはならず、ある程度要素が広がる場合がある。いま並列積和計算器としては255入力のものを用意しており、固定配線ではその場合に対応できないので、ある程度の自由度を持たせる必要がある。そこで、図9に示したように中央の235入力は固定配線とし、後の20入力はその両隣りの138個の変数及び更にその外側のブロックから送られてきた3本の信号のうちの一つを選択できるようにする。この外側のブロックに関しては、その部分にまで要素が広がる確率は低いと考えられるので、1ブロックから6本の線を出し、それを交換器を介して他のブロックに接続する。そうした場合の1ブロックの構成及びそれらの相互接続を図10に示す。この時、1ブロックで使用されるスイッチの総数は約 8×10^5 個となる。

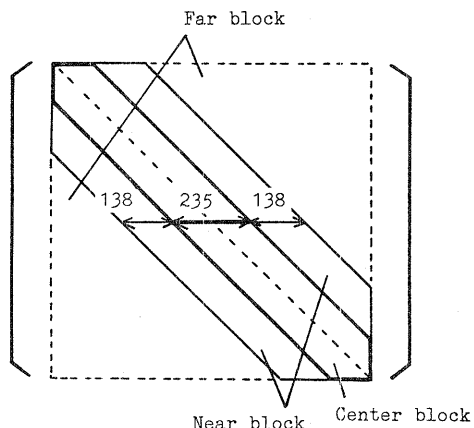


図9 システム・マトリクスのブロック分割

4.4 並列積和計算器

その構成を図11に示す。ここでは、3.2で述べた手続きに従い255組の数値の積和の計算を一時に行なう。すなわち、まずXORにより a_i と x_i の符号の乗算を行なう。それと同時

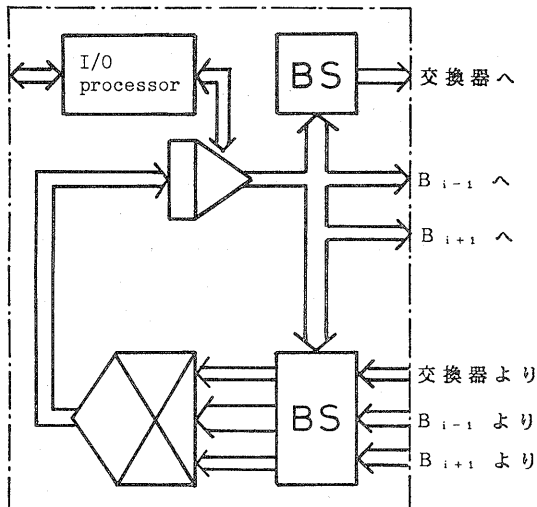


図 10 a) 1 ブロックの構成

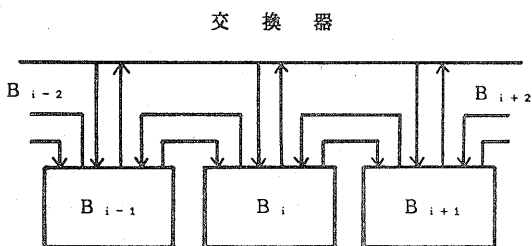


図 10 b) ブロック間の相互接続

に ADD で指数部の加算をすべての要素に関して並列に行ない、MAX ではそれらの中の最大値を選び出す。そして、SUB においてその加算値と最大値との差を求め、SHIFT ではその数だけ x_i の仮数部をそれぞれ右方向にシフトする。固定小数点積和計算器では、こうして得られた仮数と a_i の仮数との積和の計算を行なう。この時、正の成分と負の成分に分けて計算を行ない、後でそれらの差をとる。Normalize ではそうして得られた結果を正規化し、オーバーフロー・アンダーフローのチェックを行なってから出力する。

図中に示した数字はクロックの周波数を 1M Hz とした時、その処理に要するクロック数である。これにより、255 組の数値の積和の計算が約 100 μ s で終了すると予想される。

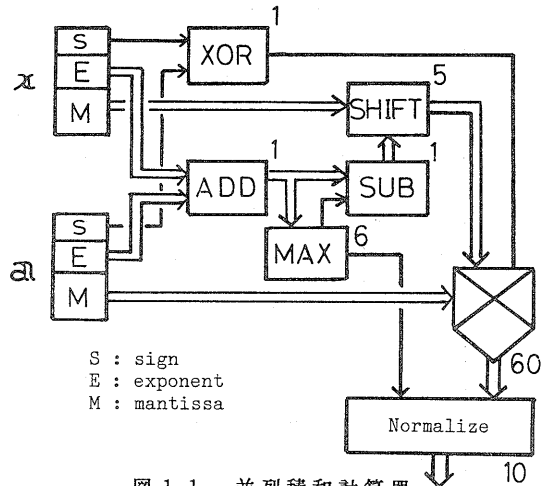


図 11 並列積和計算器

5. 試作機による検討

PASE のシステムとしての動作及び並列処理の機構について検討するため、小規模なシステム (PASE-I) を試作した。これは、2元 2 次の微分方程式に対応した次の状態方程式が解けるように構成されている。

$$\begin{bmatrix} \dot{x}_{11} \\ \dot{x}_{12} \\ \dot{x}_{21} \\ \dot{x}_{22} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k_{11} & -k_{12} & -c_{11} & -c_{12} \\ -k_{21} & -k_{22} & -c_{21} & -c_{22} \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \\ x_{21} \\ x_{22} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ f_1 \\ f_2 \end{bmatrix} \quad (3)$$

データは図 12 に示すフォーマットを有する 13 bit の浮動小数点数とした。今回は、PD については直接システムには組込まず、個別に実験を行なった。また、BS に関しては、元数が小さく有効な検討が行なえないため、試作は行なわなかった。

以下に PASE-I に関して行なった実験の結果を述べる。



指数部: 4bit エクセス表現

仮数部: 符号 bit + 8bit 符号絶対値表現

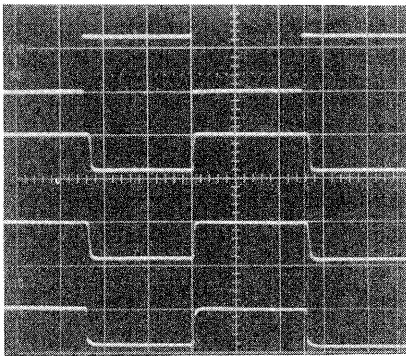
図 12 試作機の日データフォーマット

5.1 積分器 (J)

PASE-Iでは、加減算器、乗算器及び若干のレジスタを用意し、それらを外部のマイクロコンピュータから制御することにより積分器を構成した。そのため、実行には多大な時間を要したが、それは大部分マイクロコンピュータ側の処理に費やされている。PASE-I側の実際の処理時間は、加算に $2.0\mu s$ 、乗算に $1.75\mu s$ にかかるのみであった。

5.2 フォト・ディストリビュータ (PD)

発光素子に発光ダイオードTLN103を用い、受光側はフォト・トランジスタTPS603を2cm四方に5*4のマトリクス状に配置し、全体をステンレス板により覆うことによりPDを構成した。その応答波形を図13に示す。これは角に配置された2つの素子と、中央にある1つの素子の応答である。これより、2cm四方内で十分応答可能であることがわかる。また、応答速度に関しては、今回使用した素子が低速のものであったため、期待した速度は得られなかったが、TLP552(東芝)は50nsで応答するので、IC化すればこのレベルまでは応答可能だと思われる。



縦: 5 V/div
横: 0.2 ms/div

上から 入力、出力(右角、左角、中央)

図13 PDの応答波形

5.3 並列積和計算器

今回は(4)式で示される4組の数値の積和の計算と外力項の加算とを並列に行なうユニッ

トを1台製作した。

$$\text{sum} = \sum_{i=1}^4 a_i \cdot x_i + f \quad -(4)$$

そして(3)式を解く場合には \dot{x}_3 と \dot{x}_4 の計算を順次行なった。A-D変換器には8bitの逐次比較型(クロック1MHz)のものを用い、全体を6つのブロックに分割した。固定小数点積和計算器の処理時間は $10\mu s$ であり、これは大半がA-D変換器の変換時間である。そして、全体としてはその処理に $35\mu s$ を費やした。

また、PAMIのエラー補正回路に関しては、表2の実験結果がその有効性を示している。

補正	エラー/試行
なし	4/180,000
あり	0/780,000

5.4 稼動テスト

PASE-Iにより不減衰単振動の計算を行なった。すなわち、次式の状態方程式を解いた。

$$\begin{bmatrix} \dot{x}_{11} \\ \dot{x}_{12} \\ \dot{x}_{21} \\ \dot{x}_{22} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \\ x_{21} \\ x_{22} \end{bmatrix} \quad -(5)$$

初期値: $x_{11}=x_{12}=1, x_{21}=x_{22}=0$

積分法としては、Runge-Kuttaの2次公式を用いた。その計算結果を図14に示す。これは計算機上で行なったシミュレーションと完全に一致する。

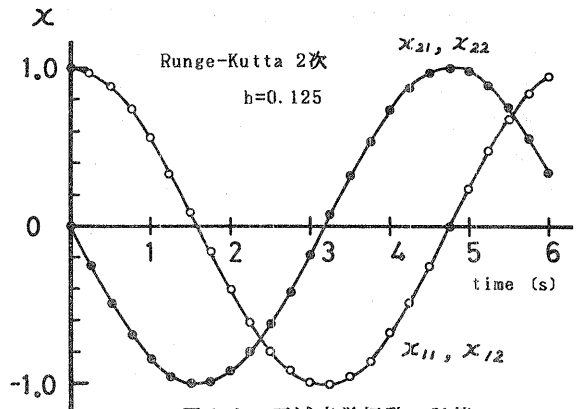


図14 不減衰単振動の計算

6. 一般化された状態方程式への対応¹⁰⁾

通常、状態方程式は (1) 式の形で与えられるが、有限要素法による解析においては、次式で示されるように、状態変数の微分項にもマトリクスが掛かる場合がある。

$$M \cdot \dot{x} = A \cdot x + f \quad (6)$$

これを一般化された状態方程式と呼ぶ。(6) 式を解く場合に、M の逆行列を求め、それを両辺に掛ければ (1) 式と同じ形となるが、そうするとマトリクスのスパース性が崩れ、大幅な演算量の増加を招く。そこで、PASE においては、ある時刻 t_n における x の値が与えられたとき、それを使って (6) 式の右辺の値を求め、それを b とおき、以下の反復式を使って \dot{x} の値を求める。

$$\dot{x}^{(k+1)} = \dot{x}^{(k)} + \alpha \cdot (b - M \cdot \dot{x}^{(k)}) \quad (7)$$

ただし、M の対角成分が 1 となるように正規化されているものとする。ここで、 α は加速係数であり、行列 M に応じて適当な値を選ぶ。また、反復の初期値としては、1 ステップ前の値を使う。以上により、比較的少ない反復回数で収束を得る。一回の反復に要する時間は約 $125 \mu s$ であり、収束に 5 回の反復を要したとしても $625 \mu s$ で収束を得る。これより、Runge-Kutta 4 次公式により 1 ステップ積分を進めるために要する時間は $2.5 ms$ 程度となると予想される。

7. むすび

大次元の状態方程式の高速処理を目標として開発中の並列型計算機 PASE のアーキテクチャについて論じ、試作機によりその並列動作機構の検討を行なった。これより、最終的には 10 万円の状態方程式において、すべての状態変数に対して 1 ステップ積分を進めるために要する時間は約 $2.5 ms$ となることが分かった。これは、現在のスーパーコンピュータに比べて約 1000 倍の処理速度である。そして、A-D 変換器に高速のものを用いれば処理速度は更に向上し、数千～1 万倍程度になると思われる。

今後は、非線形問題への対応について考察を行ない、更に積分器の部分の機能を整備し、データを 32bit に拡張した PASE-II の製作を行なう。

謝辞

白井成一氏、粕谷英雄氏（以上、現東芝勤務）、安部秀夫氏（現日本電気勤務）を始めとして本研究に携わった黒川研究室の一同に深く感謝致します。

【参考文献】

- 1) Neil R. LINCOLN: Super Computer
IEEE Trans. Computer pp345~473
Vol. C-31 No.5 MAY 1982
- 2) 村田健郎他：スーパーコンピュータ 丸善
pp35~45 (1985/3)
- 3) 田中善一郎：500MFLOPS 級のスーパーコンピュータが稼働へ、日経エレクトロニクス
pp108~126 4.11. 1983
- 4) 小原和博：階層構造の MIMD 型スーパーコンピュータ (Cedar)、情報処理 Vol.25 No.5
May 1984
- 5) R. W. Hockney: 並列計算機
共立出版 pp286~296 (1984/9)
- 6) 黒川一夫：アナログ・ハイブリッド計算機
オーム社
- 7) 岡村総一郎他：多入力加算器 PAMI とそのエラー補正、電子通信学会講演 論文集
pp7-66 PART7 (1985/3)
- 8) Kazuo KUROKAWA: PARALLEL PROCESSING
COMPUTER FOR DIFFERENTIAL EQUATION.
Vol.1 pp121~125
Proc. of '84 SCSC (1984/7)
- 9) Intel: iAPX86 ファミリー・ユーザーズ
マニュアル CQ 出版社 (1981/8)
- 10) 岡村総一郎他：並列計算機による一般化された状態方程式の解法
情報処理学会講演論文集 (III)
pp1749~1750 (1985/3)