1．計算機の歴史

1．1　バベジから第五世代へ

自動計算の試みは，イギリスの異色の数学者・発明家 C. Babbage (1792−1871) に遡る。彼は歯車やてこなどの機械仕掛けによって，10進50桁の数値を1,000個記憶することを考え，また加算における先見桁上げ方式を採用するなど，非常に先駆的なアイデアを抱いていた。詩人バイロン卿の娘でLovelace伯爵夫人Ada（エイダ）は，まだ完成していないこの計算機のためにプログラムを書いた。Adaは世界最初のプログラマであった。言うまでもなくプログラム言語adaは彼女の名に由来する。Babbageは偏執狂的な凝り性であり，ほとんど完成する間際になってから，さらに複雑な構造のものを思いつき，全部こわしてやり直そうとしたが，機械は遂に完成せず失意のうちにこの世を去った。彼は100年早く生まれ過ぎた男と言えよう。

本格的な電子計算機 ENIAC は1946年に完成しコンピュータ時代の先駆けとなったが，ほぼ同じ頃天才的数理物理学者 J.von Neumann はプログラム記憶方式のアイデアを，EDVAC 提案書の中で提示した。この方式によってコンピュータの能力は飛躍的に上昇した。

初期のコンピュータは素子として真空管を用いていた。ENIACは部屋一杯に18,000本の真空管を並べた機械だったが，その能力は今日のポケット電卓にも及ばないほどであった。真空管によるコンピュータを第一世代というが，1958年頃からトランジスタを用いたコンピュータが出現した。これを第二世代という，続いて IC の第三世代，VLSI の第四世代と発展し，今や第五世代の計算機が語られている。

1．2　第三の物理学

なぜ計算機がこのような発展を遂げたかというと，計算機へのニーズは止まるところを知らないからである。最初科学技術計算のために出発した計算機はその後事務処理の機械として発展した。世界最大のコンピュータ会社の名称に business という語が入っているのが象徴的である。事実コンピュータの最大の利用目的は事務用であり，科学技術用の計算は，その片隅を利用させてもらっているにすぎない。フォン・ノイマン達が乱流，量子力学，多体系などの科学研究のために開発したコンピュータが，事務用機器として君臨しているのは，歴史の皮肉である。その理由の一つは，科学技術計算から要求されるコンピュータの性能があまりにも大きく，これまで十分対応できなかったからであろう。

物理学においても，コンピュータは理論や実験の道具として用いられて来たが，ここ数年来コンピュータを活用した新しい研究方法が登場しつつある。すなわち，対象とする系を要素の集まりと考え，要素の運動を記述する方程式を計算機で追跡することにより，系の振舞いを理解しようとするもので，計算物理学とよばれている。一般に系が含む要素の数は大きく，その振舞いは計算してみなければ予想することができない。計算物理学は，自然そのものでなく，自然を記述すると考えられている数学モデルを対象とする点で実験物理とは異質であるが，“測定”し，帰納的な分析を行うという点で，理論物理の範疇にも入りきらない。計算物理は，理論・実験と並ぶ第三の物理と言われている。

1．3　シミュレーション

対象とする系を要素の集まりと考え，要素を支配する基礎方程式を数値的に解くことにより，系の振舞いを調べる研究方法を，一般にコンピュータ・シミュレーションとよぶ。この方法は，素粒子，原子核，天体，液体，固体，表面，プラズマなど物理の諸分野はもとより，航空，気象，原子力，石油，土木等々の応用的な分野まで広く用いられている。

シミュレーションは大きく三種に分類することができる。第一は連続体モデルと言われるもので，流体，プラズマ，熱伝導などの問題に現れる。これらの系は，複雑な境界条件を含む偏微分方程式によって記述される。連続のままでは計算機で扱えないので，差分法や有限要素法によって離散化し，時間について積分する。手法は比較的確立しているが，系の中に桁違いの空間的，時間的スケールが共存している場合には，数値計算上種々の困難が生じる。また系を特徴づける各種のパラメータ，密度，粘性係数，熱伝導率等々はあらかじめ与える必要があり，既知の物質のときはよいが，プラズマの電磁流体力学（MHD）の計算のような場合には，他の手段で求めなければならない。

これと双璧をなすのが，粒子シミュレーションといわれるもので，多数の粒子が与えられた相互作用によって運動するモデルである。ここでいう粒子は，電子のこともあれば分子のこともあり，天体物理では銀河のこともある。粒子シミュレーションの代表は分子動力学であり，多体系のニュートンの運動方程式を時間について解き，最終的には種々の物理量の

統計的な平均を求める。電磁場と粒子とが相互作用しているプラズマの粒子シミュレーションなどでは連続体モデルとの結合になる。

第三のカテゴリーに属するのが，モンテカルロ法とよばれる方法で，系の中に粒子の生成･消滅･散乱，ランダムな外力など確率的な要素を含む場合に用いられる。応用例は枚挙にいとまがないが，乱数を用いるので，統計誤差をいかに小さくするかが問題である。一般に誤差を１／C にするには，C^2倍の計算が必要なので，精度を上げるには超高速のコンピュータが必要になる。

1．4　計算機へのニーズ

以上述べたことからもわかるように，多自由度の系のシミュレーションを行うには，天文学的な計算が必要になる。たとえば流体の Navier－Stokes 方程式を100^3のメッシュで離散化して解くとする。時間の１ステップを進めるのに各点当たり10^3回の四則計算が必要とすると，10^4回の時間ステップを処理するのに10^{13}回（10 京回）の計算をしなければならない。現在の最高速の汎用計算機ではフル稼動しても10日以上かかってしまう。100^3というメッシュは，いわば100 本しか走査線のないテレビで物を見るようなもので，けっして十分ではない。少なくとも1000倍，願わくは10^6倍高速の計算機が必要なのである。

10^6倍というと夢物語のように聞こえるが，計算機の世界ではすでに経験済みの革命なのである。手まわし計算機から現在の汎用計算機まで，実に10^9倍のスピードアップを経てきたのである。

2．スーパーコンピュータ

2．1　汎用コンピュータの能力

速い計算機とは何か，計算機のスピードは何で測るか。常識的には加減乗除などの演算に何秒かかるかが計算機の速さと考えられる。しかし演算によってかかる時間が違う。もちろん整数の演算と浮動小数（Fortran の実数）の演算も速さが違う。そこで，それぞれの演算にある重みをつけて平均したものをミックス（mix）という。ギブソン･ミックスは有名である。その逆数，すなわち１秒間に平均何回の演算を実行できるかという数を，10^6を単位として MIPS（million instructions per second）という。平均１ μs に１演算が実行できれば１ MIPS の計算機である。ただし同じ整数の加算といっても，データが主記憶にあるか，キャッシュ･メモリーにあるか，レジスターにあるかで違うし，加数が０のとき，乗数が０や１のときは速いとか速くないとか，条件によって大きく変わる。当然のことながら，演算の出現率は処理の種類によって異なる。従って MIPS 値，とくにメーカーの発表する MIPS 値は一つの目安ぐらいに考えておいた方がよい。

数値計算に関する能力を表すのに，MFLOPS（million floating point operations per second）という単位も用いられる。これはその名の通り，１秒間に何回の浮動小数演算が実行できるかを示すものである。通常は繰り返し演算が多いので，DO ループの制御変数の計算も含めて，浮動小数演算の時間を示すものである。Livermore ループと称せられるテスト問題による実測によると，HITAC の M200H は4.1，M280H は6.6である。（いずれも IAP なし）。FACOM の M380は9.0。言うまでもなく，この数値も一つの目安であって，決して計算機の全能力の指標ではない。

2．2　計算機の能力の限界

それでは計算機のスピードはどこまで速くなりうるのだろうか。結局，計算機のスピードは，素子のスピードによって決まっている。リレーから LSI まで，約10^6倍のスピードアップを実現して来たが，今後どこまで進歩できるであろうか。

見通しは必ずしも明るくない。シリコンの超微細加工技術は今やサブミクロンの段階にまで到達し，そろそろ光速や発熱などの原理的限界にさしかかっている。焼き付けにも可視光では波長が長すぎて，電子線や放射光が必要だという。新しい素子としては，ガリウム･ヒ素（GaAs）やジョセフソン接合素子が研究されているが，これらが実用化されても，高々１～２桁の改善がなされるだけで，大型数値計算で必要とされるような，10^3～10^6といった進歩はとても望めそうもない。

2．3　スーパーコンピュータ

従って，今後飛躍的な能力アップをはかるには，何らかの意味での並列化が必要である。１個の素子は遅くても，何個か集まれば多くの仕事ができる。von Neumann の頃は，演算素子として真空管という高価で脆弱なものを使っていたので，いかに少ない素子で多様な演算が実行できるかが課題であった。彼の提唱したプログラム記憶方式に基づく逐次処理のアーキテクチャはまさにその目的にかなったものであった。しかしマイクロエレクトロニックスの進歩により，多数の素子を湯水のように使うことができるようになったので，並列的な処理によって高速

化する道がひらけてきた。

スーパーコンピュータとは，科学技術計算向きの超高速計算機のことである。どのくらい速ければスーパーコンピュータと言えるのかという問題は時代とともに変わっているが，現在ではピーク性能が100 MFLOPS 程度以上のものを指すようである。

スーパーコンピュータには，大きく分けて

- ベクトル計算機（パイプライン計算機）
- 並列型計算機（アレイ型計算機）

の二種に分類される。それぞれの原理については後で詳しく述べるが，堤防工事を例えとして説明しよう。今1000mの堤防を建設するとする。1人では1年に10m作れるとすると，1人だけで100年かかって完成させようというのが通常の逐次処理の計算機である。10人で作業分担をして，流れ作業で10年で完成させようというのがベクトル計算機である。1000mを10mずつ100区間に分割し，100人を配置して1年で完成してしまおうというのが並列型の計算機である。

現在，商品として売られているスーパーコンピュータは，ほとんどベクトル計算機である。しかし並列型の計算機も多くの有利な点をもっているので，今後しだいに実用化して行くであろう。

3．ベクトル計算機

3．1　原理

ベクトル計算機はパイプライン方式に基づいている。これはいわば仕事を分割してベルトコンベヤーの流れ作業にのせようとするものであり，同一演算を多数のデータに対して行うのに適している。たとえば

```
      DO  10  I＝1，N
   10 A(I)＝B(I)＋C(I)
```

という演算を実行する場合，通常の計算機では，B(1)＋C(1)の加算が終了するのを待ってA(1)へのストア命令を実行し，それから制御変数Ｉの増加と終了判定を行い，それからB(2)をロードし，C(2)との加算を行う。現在の大型汎用計算機でも，各命令の「読出し」「解読」「アドレス計算」「アドレス変換」「オペランド読み出し」などはある程度並行に処理されているが，これをもっと徹底して行おうというのが，パイプライン方式のスーパーコンピュータである。

例えば浮動小数の加算が，「指数部比較」「仮数部桁合せ」「仮数部加算」「正規化」の４つのステージに分割できるとすると，B(1)＋C(1)の指数部比較が終るやいなや，仮数部桁合せと同時にB(2)＋C(2)の指数部比較を始めてしまう。したがって４クロック後からは，１クロックごとに１個の結果が出て来ることになり，ずっと高速化される。そのように，主として１次元配列（多次元配列の断面も含む）の演算を高速化するのでベクトル計算機とよばれることが多い。またこのような流れ作業的な処理を行う命令をベクトル命令といい，コンパイラがベクトル命令のオブジェクトを出すことをベクトル化という。現在のＳ810では最内側のDOループがベクトル化される。

ベクトル計算機の特徴は，次のような点が挙げられる。

(1)　ベクトルの要素相互間の参照はできない。——I＝１の計算が終らないうちに，I＝２の計算に取りかかるから高速化されているので，前の結果が必要な計算はできない。ただしＳ810やM280Ｈでは

```
X(I)＝B(I)*X(I－1)＋C(I)
```

という型の演算だけは特別に許している。もちろん他のベクトル命令に比べてずっと遅い。それから内積の計算

```
      DO  10  I＝1，N
   10 S＝S＋A(I)*B(I)
```

も，本来直前の加算が終わらないと今回の加算が始まらないタイプの計算であるが，加算の順序を変えることによって高速化をはかっている（Ｓ810の場合)。内積は極めてよく用いられる演算だからである。

(2)　長いループほど高速になる。ベクトル命令を実行するにはある程度の準備が必要であるし，また走り出した後も，最初の何ステップ（パイプラインのステージの数）かは答が出ない。短いループではこのようなオーバーヘッドの相対的な割合が多くなり，スカラー命令（通常の命令）よりかえって遅くなることさえある。ベクトル命令とスカラー命令の処理時間が等しくなるベクトル長をクロスオーバー(交互ループ長)というが，M280Ｈ－IAPでは10前後，Ｓ810では２～３である（もちろん演算による)。

(3)　分岐（IF文）があると効率が下る。M200H－IAPではIF文は許されなかったが，M280Ｈでは単純なIF文が許されている。さらに，S810では３重入れ子のIF文までベクトル化される。しかし，IF文があると，とくにFALSEが多いと，当然のことながら効率が下る。パイプラインの流れを乱さないために，結果は不要と分かっている演算も実行するからである。S810で

は，連続８つのFALSEが続く場合は飛び越す機能があるが，実際問題ではあまり改善にはならないであろう。

(4) ループ内で許される演算に制限がつく。サブルーチンの呼び出しや，ユーザー定義関数がダメなのは当然であるが，IAPでは整数はダメとか色々な制限がある。しかし，S810ではベクトル化可能な演算をかなり拡大していて，SINなどの標準関数までサポートしている。S810で残念なのは，文関数を許していないことである。コンパイル時にインラインに展開しているのに。

3．2　加速率

ベクトル化によってどのくらい計算が高速化されるかを加速率というが，これを事前に予測することは非常に困難である。あるプログラムを逐次型の計算機で実行した場合の CPU 時間のうち，ベクトル化可能なDOループのCPU時間が α であったとする。α をベクトル化率といい，HITACではVECTIZERというソフトによって測定することができる。ベクトル演算装置の速度比を a_0 とすると，このプログラム全体での加速率aは

$$a=\frac{a_0}{(1-\alpha)a_0+\alpha} \tag{3.1}$$

となる。S810／10では，ループ長や演算にもよるが，a_0 は10〜40である。$\alpha\rightarrow 1$ の極限では $a\rightarrow a_0$ となるが，α が１から少しでも下がると，aは急激に小さくなる。 $a_0=20$ とすると，$\alpha=0.95$ でも $a=10$ となってしまう。

もう一つの数字は VPU 使用率 v，すなわち VPU 時間(ベクトル演算器の作動時間)と SPU 時間（スカラー演算器の作動時間。通常の CPU 時間と思ってよい）の比である。これは，全計算時間中VPU がどれだけ働いたかを示す数字であり，各 JOB 毎に VPU 時間が表示されているので，自分で計算することができる。ただし SPU 時間には，コンパイルやリンクの時間も入っているので，GO ステップの CPU 時間を用いるとよい。

さて，SPU は，VPU が働いている間，通常は何もせず待っているだけ（ただし時間のカウントは進む）なので，

$$v=\frac{\alpha}{(1-\alpha)a_0+\alpha} \tag{3.2}$$

の関係がある。KEK の６〜７月の統計では，Lクラスでは平均 $v=0.99$ であったので，$a_0=20$ としても，$\alpha=0.9995$　と推定される。また（3.1）と比較して

$$a=a_0\ \frac{v}{\alpha} \tag{3.3}$$

を得る。東大センターでは $v=0.4\sim0.5$ である。

3．3　世界のベクトル計算機

表３－１に，代表的なベクトル計算機のデータを示す。データは日進月歩なのですでに変っているところがあるかも知れない。1984末までに，Cray 1（1Sを含む）は63台，Cray－XMP は25台 Cyber205は27台納入されたという。1985年中に，Cray は30台（３台の Cray2を含む）を売る予定であるという。日本でも，予定を含め５台の Cray が売れている。

日本の３社のスーパーコンピュータも順調な売行きを示している。社内利用を含め HITAC S810が約７台，FCOM VP が約10台，NEC の SX－２が約２台が納入もしくは納入予定である。これ以外に VP は約３台輸出されているという。

これ以外にも，アメリカの Floating Point 社の付加プロセッサの上位機種はかなりの性能を出しているし，種々のベンチャービジネスが乗り出そうとしている。中国でも「銀河」という名の100 MFLOPS のスーパーコンピュータが作られ動いている。外観は CRAY－１にそっくりである。

3．4　ベクトル化の工夫

日本のスーパーコンピュータ（S810や VP）は，バカチョン的に使ってもかなりの高性能が出るのが特徴であるが，ちょっとした工夫でさらにスピードは倍加する。このへんが従来の汎用計算機と違っておもしろいところである。

例えば，行列の積C＝AB を考えてみると，通常図３－１のＩのようなプログラムを用いる。しかし，DO ループを入れかえてIIのように書くこともできる。実測してみると，N＝200，倍精度の場合（M 280 H－IAP で）

Ｉ　1.96秒　　II　1.27秒

となって，IIの方が断然速い。その理由は，Ｉでは内積演算であるのに対し，IIではスカラー乗算付加算（B（K，Ｉ）はループ内ではスカラー）となって，並列度が増すからであると考えられる。S810／20では，IIIのようにプログラムを多項化するとさらに速くなる（Nは偶数と仮定)。同じ例で，

Ｉ　0.41秒　　II　0.29秒　　III　0.18秒

となる。S810では右辺を４項（K〜K＋３）に増や

表3．1

機種	発表	出荷	* ピーク性能 (MFLOPS)	マシンサイクル (rs)	主記憶 最大容量 (MB)	**素子	集積度	アクセスタイム (rs)	拡張記憶 最大容量 (MB)	転送速度 (MB/sec)	備考
CRAY-1	1976	1976	80	12.5	8	B		50			
CRAY-1S	1979	7980	80	12.5	32	M		50			
CRAY-XMP	1982	1983	400	9.5	32	B		38	256	1250	2CPUの場合。
CRAY-2	1981	1985	800	4	256						
Cyber 203		1980	50	20	8	B	4K	80			
Cyber 205	1980	1983	400	20	32	B	4K	80			4パイプの場合。
Cyber 2XX	未		10000?		2000						詳細不明。
VP-100	1982	1983	280	7.5	128	M	64K	55			
VP-200	1982	1983	560	7.5	256	M	64K	55	半導体の拡張記憶はなし。		
VP- 50	1985	1985	140		128						
VP-400	1985	1986	1120		256						航技研に1号機。
S810/20	1982	1983	630	14	256	M	16K	40	1024	1000	
S810/10	1982	1985	315	14	128	M	32K	40?	1024	500	KEKに1号機。
上位機種	未	1987	1500?								分子研に1号機。
SX-1	1983		570	7	256	M	64K	40	1000	1300	
SX-2	1983	1985	1300	6	256	M	64K	40	2000	1300	

* ピーク性能は，64ビット演算の場合。会社発表による。

** B：バイポーラ，M：MOSスタティック

```
Program I

          DO 20 I=1,N
          DO 20 J=1,N
            C(J,I)=0.0
            DO 20 K=1,N
      20    C(J,I)=C(J,I) + A(J,K)*B(K,I)

Program II

          DO 20 I=1,N
            DO 10 J=1,N
      10    C(J,I)=0.0
            DO 20 K=1,N
            DO 20 J=1,N
      20    C(J,I)=C(J,I) + A(J,K)*B(K,I)

Program III

          DO 20 I=1,N
            DO 10 J=1,N
      10    C(J,I)=0.0
            DO 20 K=1,N,2
            DO 20 J=1,N
      20    C(J,I)=C(J,I) + A(J,K)*B(K,I) + A(J,K+1)*B(K+1,I)
```

図3．1

すとさらに高速化される。試みにIAPでIIIを走らせると2.06秒となって逆効果である。上の計算は，浮動小数演算が1.6×10^7回であるから，S810での演算速度は，MFLOPSを単位として，

I 39　　II 56　　III 91

である。S810の最高記録687 MFLOPSには程遠いが，100 MFLOPS前後までならば，このようなちょっとした工夫で到達できることが多い。それ以上は名人芸であって，素人には無理であるがライブラリー（S810では MATRIX／HAPなど）を使えば実現できる。だいたいの目安として，IAPで4倍速くなるプログラムは，S810にそのまま掛ければさらに4倍速くなると言われている。

ベクトル化できない本質的な理由は2つある。一つは，データの依存関係，もう一つは複雑な分岐である。データの依存関係がある場合には，上に述べたようなプログラミング上の変更に止らず，アルゴリズムそのものを大幅に変更する必要が生じること

もある。例えば，２次元領域のラプラス方程式を差分法で連立方程式に直し，それをSORで解く場合，通常は端から順に値を更新して行くが，そのためには隣の点での値が確定していなければならない。すなわちベクトルの要素相互間の参照が必要となる。そのままではベクトル化が出来ないので，一つおきに更新するとか，斜めに更新するとか種々の工夫が提案されている。これによって解への収束のスピードがどうなるかは，数値解析の研究テーマである。

複雑な分岐のある例としては，カスケードシャワーのシュミレーションがあげられる。一度生成されてしまえば，粒子同志は全く独立であるから，データの衣存関係はないが，各粒子に施すべき演算が全くマチマチなのでベクトル化は困難である。

４．並列型計算機

４．１　SIMD と MIMD

大量の演算を行う場合，同時に独立して実行できる部分に分割することができれば，複数個の演算器を用いて並列に処理することにより，高速の処理ができる。これを並列処理という。前節のパイプライン処理も一種の並列処理であるが，本節では，もっと大きな演算の単位の並列処理について考える。

並列に動作する多数の演算装置（Processing Unit, PU と略す）があったとする。各 PU ではデータが命令（instruction）に従って順次処理される。Flynn は，命令とデータの流れ方に注目して，並列計算機を次の４種と分類している。

(1) SISD（Single-Instruction stream — Single-Data stream）単一命令流・単一データ流…通常の van Neumann 型計算機の場合。

(2) SIMD（Single-Instruction stream Multi-Data stream）単一命令流・複数データ流…命令は一ヶ所で解読され，同一の命令が全 PU に放送され，同期して実行される。適当なマスクがかかってもよい。

(3) MISD（Multi-Instruction stream Single-Data stream）複数命令流・単一データ流…単一データが複数のPUで異なった一連の処理を受ける。

(4) MIMD（Multi-Instruction stream-Multi-Data stream）複数命令流・複数データ流…各 PU は独立な命令解読機能をもち，異なるデータを非同期的に処理する。

以上のうち，MISD 型の並列計算機は現実的でなく，SIMD と MIMD 型が実用化されている。前に述べた堤防工事の例えで言うならば，つるはしの動かし方からセメントのぬり方まで全く同期して行うのが SIMD であり，各人自分の好きなように作業するのが MIMD である。

MIMD の方がより融通性があり効率も高いはずであるが，PU 間の同期やデータの転送がむつかしいので，最初につくられた ILLIAC—IV は SIMD であった。後にのべる PAX は MIMD であるが，各 PU で同一のプログラムを走らせ，適当な同期点をおくことにより，この困難を回避している。

４．２　メモリーとネットーク

演算装置とメモリーとの結合方式に注目して分類することもできる。

(1) 共有メモリー型—処理すべきデータを集中して共有する。どの PU も直接データにアクセスできるので汎用性は高いが，PU の数が大きい場合には，アクセスの競合が起る。共有メモリ型は，メモリの接続方式により，バス方式，メモリスイッチ方式，マルチポートメモリ方式に分類される。

(2) ネットワーク型—相互に結合されたPUのローカルメモリにデータを分散して格納する。結合方式には大きく分けて，階層構造のものと非階層構造のものがある。階層構造には，鎖状結合，星状結合，木状結合などがある。非階層構造のネットワークには，環状結合（１次元），格子状結合（２次元円環），超立方体結合などがある。２次元格子は ILLIAC IVや PAX で採用した型で，近接相互作用の計算に適している。Caltechのグループは 2^p（p＝６）の超立方体結合で，ゲージ理論の計算を行っている。また IBM で製作中の 11 GFLOPS をめざす GF11 は，スイッチ網で PU 間の結合を行なっている。GF11 は SIMD 型であり，やはり，ゲージ理論の計算を目ざしている。

４．３　物理系のための並列計算機

物理現象の中には，近接相互作用で記述されるものが極めて多い。電磁場，熱伝導，拡散，波動などは偏微分方程式で記述され，離散化すれば，空間的に隣接する点の影響を受けながら変ってゆく。また粒子系の場合でも，クーロン力などの場合を除けば，近接性が高い例が多い。これらの特徴を生かせば，物理系に適した並列計算機をつくることができ

る。

近接性を計算機中の処理にまで生かすためには，物理空間と計算機の構成の間に何らかの相似性がなければならない。すなわち，物理空間と並列計算機の PU の空間の間に何らかの対応があり，物理空間における近接作用が，PU 間の近接通信に対応していなければならない。

2次元の物理系を2次元の PU 空間に対応させる最も簡単な方法は直接写像である。PUが32×32であるとすると，空間も32×32に分割し，各PUは一つの分割された空間の計算を担当する。また3次元以上の系でも，そのうち2次元について同様な写像を行えば，やはり近接性は保持されている。

4．4 PAX とその応用

PAX は以上の考えに基づいて星野氏らが製作した，2次元隣接結合のMIMD型並列計算機である。現在までに，PAX−9，PAX−32，PAX−128，PAX−64Jが製作された。

詳しくは別書に譲るが，640万円の材料費で作成されたPAX−128は，ポアソン方程式の解法などでは，ほぼ M200と同じ速さ（約4MFLOPS）で動いた。

主な応用は，

(1) ポアソン方程式…SOR 法
(2) ポアソン方程式…ADI 法
(3) プラズマ粒子シュミレーション（FFT を含む）
(4) アモルファス物質のシュミレーション（分子動力学）
(5) ハンゼンベククスピン系…モンテカルロ法
(6) 4次元U(1)ゲージ模型…モンテカルロ法
(7) 連立1次方程式…掃き出し法，共役傾斜法
(8) 2次元弾性問題…有限要素法
(9) 不定流計算
(10) 論理回路のシュミレーション
(11) 空気力学
(12) 沸騰水型原子炉炉心の3次元計算
(13) プラズマ中の高速イオンの損失領域計算

このうち(4)(5)は筆者が中心となって行った。

5．ベクトル計算機のアルゴリズム（省略）

6．終りに

以上述べたように，スーパーコンピュータの登場は，科学技術の諸部門に多大な影響を与えつつある。筆者は加速器については全くの素人であるが，磁場設計，空胴設計には直ちに応用可能であろう。軌道計算には多少の工夫が必要かも知れない。

今後は，汎用のスーパーコンピュータもさることながら，PAX のような目的志向型の並列計算機が活躍するようになるであろう。計算機は再び買う時代から作る時代に戻るのかも知れない。

参考文献

ベクトル計算機（とくにS810）については，
1）林田健郎，小田力，唐木幸比古著：スーパーコンピュータ；科学技術計算への適用，丸善，1985.
並列計算機（とくにPAX）については，
2）川合敏雄著；スーパーコンピュータへの挑戦，岩波書店，1985.
3）星野力編著；PAXコンピュータ　高並列処理と科学計算，オーム社，1985.