

## 解 説



## 重点領域研究：超並列原理に基づく情報処理基本体系

## 2. 応用プログラム†

小柳 義夫<sup>1)</sup> 白川 友紀<sup>2)</sup> 中嶋 正之<sup>3)</sup>  
 深澤 良彰<sup>4)</sup> 村岡 洋一<sup>4)</sup> 安浦 寛人<sup>5)</sup>  
 吉田 紀彦<sup>6)</sup>

## 1. はじめに

「超並列処理」プロジェクトにおける応用プログラム班の使命は、以下の二つである。一つには、本プロジェクトで開発する超並列計算機 JUMP-1 の上で実際に使うことのできる応用プログラムを開発することであり、また一つにはハードウェアおよびシステムプログラム設計などに際して定量的なデータを与えるとともに、新しいパラダイムなどを考えるための指向実験の対象となるべき例題を与えることである。

超並列処理の基本モデルは、データ並列処理とコントロール並列処理に大別できる。我々は両方のモデルについて、それぞれ新しい応用プログラムを試作した。

さらに、応用プログラムに留まることなく、利用者とのヒューマンインタフェース、特に計算結果の可視化技術についても研究を進めた。

以下の各章において、データ並列処理の応用、コントロール並列処理の応用、可視化およびその他の話題について、それぞれ主要な結果を報告する。

なお、誌面の都合上ここに紹介するのは、全体の研究成果のごく一部であることをお断わりしておく。

## 2. データ並列処理

## 2.1 場モデルによる超並列処理

大規模数値シミュレーションのかかなりの部分は偏微分方程式など、小要素間の相互作用が近接的局所的な場モデルのシミュレーションである。このようなモデルではモデル自体の並列性と局所性を利用して自然な並列性を持つ素朴な解法を構成することが、多くの場合可能となる。

しかし単純素朴なデータ並列的方法は一般に収束性が悪く、高い効率を得られない。速く収束し、しかも並列性も高いアルゴリズムが必要である。

本稿では楕円型偏微分方程式で最も重要な線形拡散方程式の高速な解法であるマルチグリッド前処理つき共役勾配法 (MGCG 法) を紹介する。

偏微分方程式が線形であれば、差分法で離散化してできる方程式は大規模疎行列を係数とする連立一次方程式  $Ax=b$  となる。大規模なので直接法は実用的でなく、適当な初期値から残差  $r=b-Ax$  が減少するように変数  $x$  を更新してゆく反復法が広く用いられている。

しかし、並列性の高い反復法はデータ依存が少ないので、短波長の成分の収束は速いが、長波長の成分はなかなか収束しない。この欠点を克服する解法の一つにマルチグリッド法 (MG 法) がある。MG 法では本来の細かい格子  $G_0$  より粗い格子  $G_1$  を用意する。

これらの格子はいずれももとの問題の近似であるので、適当な方法で  $G_0$  上の変数を  $G_1$  上に制限したり、 $G_1$  上の変数を  $G_0$  上に展開したりすることが可能である。そこで  $G_0$  における残差  $r$  を  $G_1$  に移して変数  $x$  の更新値を  $G_1$  で求め、これを  $G_0$  に展開することによって  $G_0$  での残差を減少させる。

† Application Programs by Yoshio OYANAGI (University of Tokyo), Tomonori SHIRAKAWA (Tsukuba University), Masayuki NAKAZIMA (Tokyo Institute of Technology), Yoshiaki FUKAZAWA, Yoichi MURAKAWA (Waseda University), Hiroto YASUURA and Norihiko YOSHIDA (Kyushu University).

<sup>1)</sup> 東京大学理学部

<sup>2)</sup> 筑波大学構造工学系

<sup>3)</sup> 東京工業大学情報理工学研究科

<sup>4)</sup> 早稲田大学理工学部

<sup>5)</sup> 九州大学大学院総合理工学研究科

<sup>6)</sup> 九州大学工学部

これを coarse grid correction という。この coarse grid correction だけでは残差は 0 にはならない。粗い格子では近似できない短波長の残差が残るためである。これを smoothing と呼ばれる処理で滑らかにする。

smoothing は coarse grid correction の前後に一度ずつ行うのが普通で、前のものを pre-smoothing, 後のものを post-smoothing と呼ぶ。

ここでは格子を 2 種類としたが、多数の格子を準備して再帰的に coarse grid correction を行うのが MG 法である。

MGCG 法はこの MG 法を共役勾配法 (CG 法) の前処理として用いたものである。注意すべきは、CG 法では前処理行列は対称でなければならないので smoothing は pre と post で同じ対称な方法を使わなければならない点である<sup>1)</sup>。

一方非対称問題を解くためには自乗共役勾配法 (CGS 法) を使うことができるが、ここでも MG 法を前処理にすることができる。この方法の場合は MG 法に特に制限はかからないが、非対称性の増大を防ぐために粗い格子では風上差分を使わなければならない<sup>2)</sup>。

これらの解法を MG 法および広く用いられてきた ICCG 法, ILUCGS 法と比較しよう。問題は正方領域上の移流拡散方程式で、格子は  $256 \times 256$  とした。

MG 法は一様な問題には有効であるが、非一様な問題では収束が遅い。これに対して MGCG 法はどの問題にも安定に収束している。また、ICCG 法に比べて反復回数が極端に少ないため高速である。同じ問題を異なる格子で解いた場合、ICCG 法は反復回数が格子の大きさにほぼ比例するのに対し、MGCG 法では反復回数はほぼ一定であることが経験的に知られている。このため MGCG 法は大規模問題ほど有利である。時間は HP 720 による (単位 sec)。

問題	MGCG	ICCG	MG
一様反復数	7	160	11
時間	16.3	107.9	31.2
非一様反復数	13	238	69
時間	29.3	157.8	189.3
非対称反復数	5	136	28
時間	45.1	301.1	100.3

MGCG 法は ICCG 法とは違い、自然な並列性を持っているので並列処理向きである。しかし格子点の数が処理装置の数より少ないと並列度が不足して無駄が発生する。一方できるだけ粗い格子を使った方が収束がよい。そのため最も粗い格子でも格子点の数は処理装置の数と同程度にするのが最適である<sup>3)</sup>。

MGCG 法を富士通の MIMD 型並列計算機 AP 1000 上に実装してその並列性を検証した<sup>4)</sup>。ここではキャッシュの影響を排除するため処理装置あたりの格子点数を  $64 \times 64$  に固定して反復回数と所要時間 (秒) を計測した。その結果、約 90% の並列化効率を得た。

処理装置数	1	4	16	64
反復回数	10	10	10	13
所要時間(sec)	2.85	3.18	3.31	4.29
MFlops	1.74	6.31	24.5	98.6
Speed-up		3.63	14.1	56.7
効率		0.91	0.88	0.89

本節では高速で高並列な線形偏微分方程式の解法である MGCG 法を紹介した。MGCG 法は移流拡散方程式に対して非常に有効で、容易に効率的な並列化が可能であることが示された。しかし MGCG 法の進む道は必ずしも平坦ではない。不規則形状の問題や異方性方程式、ヘルムホルツ方程式に適用する研究が MGCG 法の確立に必要な不可欠である。

## 2.2 並列処理シミュレータ

並列処理プログラムの開発においてデバッグ時などには、普通のワークステーション上で実行する方が効率がよい。このための方法としては、fork システムコールなどにより複数のプロセスを起動し、プロセス間通信などを用いて並列処理をシミュレーションすることが一般的である。しかし、この方法ではプロセススイッチングなどにもなうオーバーヘッドが大きく、プロセッサ数が増えると実行速度が極端に遅くなるという欠点がある。

このような欠点に対処するために、ノード記述機能 XFC と、SPMD (Single Program-Multiple Data Stream) 形式で書いたノードプログラムを、並列計算機のソースプログラム (C++) に展開するプログラム DMSTEG (Direct Mapping Scheme Translator Emulator-Generator)

とを開発した。

生成されたC++のソースでは、並列計算機の1台のプロセッサを一つのオブジェクトに対応させていて、普通のワークステーションでg++などでコンパイルして疑似的に並列処理を実行することができる。

プロセススイッチングなどを行わないので、シミュレートするプロセッサ数が多くても実行速度があまり低下しないという利点がある。また、ヘッダファイルを変えれば2次元隣接結合でない他のトポロジーの計算機用にも使用できる。現在DMSTEGがサポートしている機能は、

プロセッサアレイのノード記述プログラム  
隣接プロセッサとの共有メモリ  
隣接プロセッサとの同期  
放送  
全プロセッサの共有メモリ  
バリア同期  
ホスト計算機のプログラム

である。

DMSTEGを用いれば、サンプルプログラムのSOR法によるポアソン方程式の求解プログラムやガウス消去法による連立1次方程式の求解プログラム、あるいは2次元非圧縮流れのシミュレーションプログラム、境界要素法による2次元の弾性体のシミュレーションなどの並列処理プログラムを、普通のワークステーション上でエミュレーションできる。

電界のポテンシャル問題、非定常熱伝導問題、非圧縮性流体の圧力計算問題など、ポアソン方程式またはラプラス方程式によって記述される物理現象は大変多い。そのためポアソン方程式の求解プログラムは、連立一次方程式求解のプログラムとならんで、よく並列計算機のベンチマークプログラムに使用される。

Red-Black SOR(Successive Over Relaxation)法は、並列度が高く計算に局所性があり、超並列処理に大変適合したアルゴリズムである。しかしこの方法は、境界条件にディリクレ条件のみを持つ問題には適しているが、ノイマン条件を含む問題に適用すると、収束性が非常に悪いという問題がある。

場の問題を解く別の方法として、境界要素法がある。

この方法は境界上の点のみを考えればよいので入力が少なくて済むなど取扱いが容易であるという利点がある。しかし、境界要素法では基本解が分かっているラプラス方程式などはそのまま計算できるが、ポアソン方程式は湧き出し項を数値積分する必要がある。また内点数が非常に多い場合は内点の計算に多くの計算量を要する。

そのため、上述の二つの方法を組み合わせて、ノイマン条件を持つポアソン方程式の解を、ディリクレ条件を持つポアソン方程式の解とノイマン条件を持つラプラス方程式の解に分解し、前者をSOR法、後者の境界値を境界要素法、内点値をSOR法により解くという複合アルゴリズムが考えられている。

ところが、SOR法は計算が局所的で並列処理に向いているのに対し境界要素法は並列計算機の全プロセッサにわたっての通信ないしは放送が必要であるので、このような複合アルゴリズムを並列計算機で処理する場合、総合的に見て良い結果が得られるかどうかは一概には判断できない。

そこで、境界上でノイマン条件とディリクレ条件が混在するポアソン方程式の求解問題を、実際に4×4の16台のプロセッサを持つ並列計算機を用いて、Red-Black SOR法、境界要素法、SBS法の三つの方法により数値計算をし比較を行った。

SORのときには問題の全領域を均等に各プロセッサに割り当て、境界要素法のときは境界点を各プロセッサに均等に割り当てた。

その結果、SOR法のみではうまく収束しなかった。また、SOR法と境界要素法を組み合わせた複合解法は、境界要素法より並列計算効率の良いが処理時間は短く、総合的に見て良いことが分かった。

### 3. コントロール並列処理

#### 3.1 超並列環境における組織形成と通信方式

ここでの基本的な興味は組織のモデル化にある。個体のモデル化についてプロセスやオブジェクトといった枠組があるのと同様に、個体の集団がいかにか協調的に組織化して全体として一つの目的に向かって動作するかについても、何らかの枠組が考えられるはずである。

組織のモデル化は、一つには並列・分散システ

ムの構築基盤として意義を持つ。比喩的にいえば、たとえば百万人を動員して作業を行うという状況では、仕事割当て・指揮系統・物流形態などを何らかの形で組織化しなければならない。同様に、百万個の実行主体からなるような超並列環境においては、組織化の問題を解決することなしにはシステムの効率的な構築はほとんど不可能とも考えられる。

我々はこれまで、様々な組織がいかに形成され得るか、そして組織の構造が個体間相互作用すなわち通信の形態によっていかに特徴付けられるかについて、規則的構造や階層的構造をいかにトップダウンに与えることができるかという方向から、次のような研究を進めてきた。

●超並列プログラムやVLSIアーキテクチャの導出<sup>5)</sup>

●集団の抽象化に基づくプログラミング<sup>6)</sup>

これに対して本研究プロジェクトの一環としては、超並列環境を協調型問題解決や社会現象・生物群シミュレーションなどにも活用する道を開くべく、上とは逆に、組織がいかにボトムアップに自律形成され得るかという方向からいくつかの研究を進めているので、それらを紹介する。

#### (1) 通信戦略に基づく自律的組織形成

集団の定義は様々であるが、情報の共有が一つの重要な要因となる。動的な環境情報など事前に把握できないものは、個体間の通信によって共有を果たさなければならない。これは通信コストが0の理想的な環境では随時放送すればよいが、現実には空間的（通信の相手範囲）、時間的（通信の同期）に何らかの構造を導入する必要がある。

情報を通信で授受するよりも各個体が自ら計算してしまう方が、局所処理のみで対処するため不正確かもしれないが、効率が良い場合もあり得る。一般に、均質な集団から組織が形成されるのは、各個体が自ら何らかの評価を最適化しようとする過程においてであるから、上の通信構造化は組織形成の重要な側面を表すものと考えてよい。

この通信構造化のための戦略として、空間的には通信相手のクラスタ化を行うもの、時間的には一種のバリア同期化を行うものを設計した。各個体は、集団全体が均質という前提で過去の履歴から共有すべき情報と周囲の他者のモデルを構築

し、最適と思われる通信相手集合と同期時点を決定する。この戦略の拡張では、クラスタ統括・同期トリガのために確率的に中央管理者が現れ出ることがあり、これは階層組織の自律的形成に相当する。

我々は並列問題解決を取りあげて戦略のシミュレーションによる評価を行い、通信構造固定の場合よりも効率が改善されることを検証した<sup>7)</sup>。

#### (2) 遺伝的手法の並列処理におけるクラスタ化

ランダムな集団から何らかの秩序が創発する過程は、人工生命の分野でも活発に研究がなされている。ここでは、非平衡熱力学の成果を応用した解析的分析や、シミュレーションによる秩序創発過程の観察がなされているが、何らかの意味で有用な機能が創発された事例はまだ数少ない。この分野の周辺で工学的見地から最も成果を産んでいるのは、遺伝的アルゴリズムに代表される遺伝的手法である。

遺伝的プログラミングは、遺伝的アルゴリズムが問題をビット列に写像したものを遺伝子として淘汰させるのに対して、プログラムそのものを遺伝子として淘汰させるものであり、より高度な機能が実現できるものと期待されている。

遺伝的手法に関する組織化で最も興味深い問題の一つは、その並列処理における遺伝子集団のクラスタ分割と移住の影響である。遺伝的手法は数千から数万の遺伝子集団を扱うので、これをクラスタ分割して並列処理することで大きな高速化が図れる。ただし、そのままでは各クラスタが並列に孤立進化することになるので、適当な時間間隔で適当な周囲クラスタと遺伝子の交換移住を行う。

この移住をどのように行うのがよいかについて、まだ適当な指針が確立されていない。一般に、ある系が多くの要素から構成される場合に、要素どうしの相互作用の「程度」によって全体が一つの連続系と見なせるようになるか孤立系の単なる集まりとなるかの相転移があることが知られており、パーコレーション (Percolation) 理論として研究が進んでいる。この理論を応用してクラスタ間移住の影響を解明し、遺伝的手法の並列化におけるクラスタ化と移住の指針を確立することを目指している。

並列・分散システムの将来像としては、一言でいえば、マシンをプロセッサの離散的な集合体ではなく熱力学的な連続体として捉えるようなモデルが、一つのあるべき姿と考えている。並列・分散システムが莫大な数のプロセッサから構成されるようになると、もはやプロセッサを個々に意識してシステムを設計するのは不可能になり、全体を一つの連続体として捉えて、自律的な組織化や秩序形成のメカニズムを導入することが不可欠となるであろう。

### 3.2 実時間音楽情報処理

異質な処理を組み合わせた並列処理のもう一つの題材として、実時間音楽情報処理を選んだ。本節では、ドラムがビートを刻む音楽の音響信号から実時間でビート情報を抽出するビートトラッキングシステムについて紹介する<sup>8)</sup>。

システムは、複数のプロセス（エージェント）がそれぞれ異なった解釈に基づいて並列にビート検出を行い、それらの結果から最も確信度の高いものを選ぶ。システムは並列計算機 AP 1000 上に実装され、非常に高い検出度を達成することができた。

ビートトラッキングとは、人間が音楽に合わせて手拍子を打つように、曲のビート位置を認識することである。音響信号からビートを検出するには、以下のような課題がある。

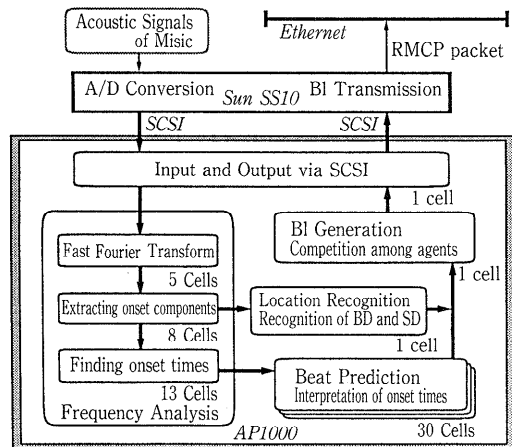
(a) 音響信号中には多数の楽器落ちが混在しており、これらから各音の発音時刻を正確に求めることは困難である。

(b) ビートは音楽中から人間が近くする概念であり、実際の音の信号中に 1 対 1 に対応するとは限らない。たとえば、音響信号のピーク検出のみではビート情報は検出できない。また、音響信号がないところにビートが存在することもある。

(c) 4 分音符の長さや、各ビートの強弱を判断することは困難である。

これらの問題を解決するに、本ビートトラッキングシステムでは、ビートの解釈の複数の可能性について、並列に調べる方法を考案した。

システムは、まず周波数解析を行い、各発音時刻を確信度付きで求める。これらの複数の発音時刻を複数のエージェントが異なった戦略で解釈して、ビート時刻の予測を行う。合わせて、主要なドラム音の発音時刻や音量、さらに様々な音楽に



BI stands for beat information.

Cell means a processing element in AP1000.

図-1 ビートトラッキングシステム

関する知識などを用いて、ビート情報の検出を行う。

本システムは、周波数解析（高速フーリエ変換、発音時刻の検出など）、複数エージェントによるビート予測、解釈の統合などをそれぞれ複数のプロセッサで実行して、これらをパイプライン処理して実時間処理を実現した（図-1）。

本システムでビート情報検出を 30 曲を対象に行ったところ、約 90% 以上の検出率を達成できた。

## 4. 可視化

### 4.1 はじめに

コンピュータグラフィックスの分野では、写真並みの高品質な画像が生成可能なレンダリングアルゴリズムであるラジオシティやレイトレーシングの高速化が重要な問題になっている。

また、科学技術計算の結果の可視化などの分野でも、高速な画像の生成が必要である。

しかし、現在のところ、高価な専用ハードウェアを利用しても一枚の画像を生成するのに莫大な時間が必要である。

そこで、超並列計算機を用いることで、これまで莫大な時間を費やしていた画像生成は現実的な許容し得る時間での処理が可能になると考える。

ここでは、レイトレーシングおよびボリュームレンダリングの並列化に関して述べる。

### 4.2 レイトレーシングの並列化<sup>9)</sup>

本研究ではレイトレーシングに内在する並列性のうち、ピクセル並列処理、物体並列処理、および二次光線(反射光, 屈折光, shadowing)の3種類の並列性を考慮した並列アルゴリズムについて検討を行った。

また、これらを組み合わせたハイブリッドな並列アルゴリズムとして、次のような3種類の処理について検討を行った。

- (a) ピクセル+二次光線の並列処理
- (b) ピクセル+物体並列処理
- (c) ピクセル+二次光線+物体並列処理

単純なピクセル並列処理だけでなく3種類のハイブリッドな並列アルゴリズムを用いていくつかの画像を生成した。

データのロード時間を含めた処理の総合時間を図-2に示す。処理時間を比較した結果、レイ数が64以下では、単純なピクセル並列処理が最も速く、次いで(a), (c), (b)の順であった。

またSUNのSS10で生成に30分程度かかる画像もピクセル並列処理の場合にはおよそ40秒で生成することができ、非常に有用であるといえる。

### 4.3 ボリュームレンダリングの並列化

前節に述べたレイトレーシングの並列化では、並列計算機AP1000のプロセッサのメモリが16Mbyteと比較的豊富であるため各プロセッサにすべてのオブジェクトデータを格納することができた。

しかし、パッチ数を非常に多くしたラジオシティ、パーティクルモデルのレンダリング、および

ボリュームデータを扱うボリュームレンダリングなどにおいては処理する物体数は非常に増える。

したがって、各プロセッサにすべてのオブジェクトデータを格納することはできなくなり、プロセッサ間の通信が多く必要になってくる。

ここでは、ボリュームレンダリングの手法の一つであるボリュームレイキャスティングの並列化について検討を行う。

データがプロセッサに分散して存在した場合でも従来より高速にボリュームレンダリングを行うために、疑似キャッシュを用いてボリュームレンダリングを効率よく行う手法を提案し、キャッシュの必要性について述べる。

また、疑似キャッシュに登録するデータを光線のコヒーレンスを用いて制限する方法について述べる。

ボリュームレイキャスティング法においては、視点から投影面の画素に投影した光線が交差するボクセルデータの透過率を足し合わせることで投影面の画像を描画する。

したがって、投影面での近傍の画素は、ボリュームデータにおける近傍のボクセルデータを使用する可能性が高くなる。

このことから、疑似キャッシュに登録するデータをアダプティブに制御し処理の高速化を行う。

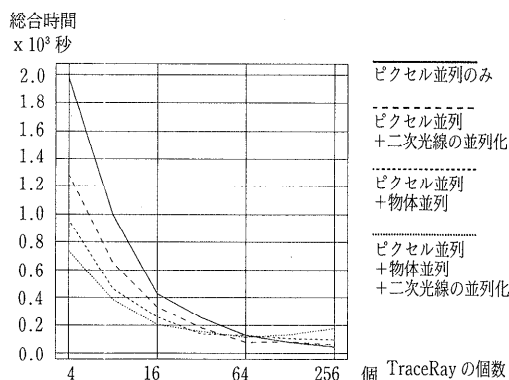


図-2 画像生成の総合時間

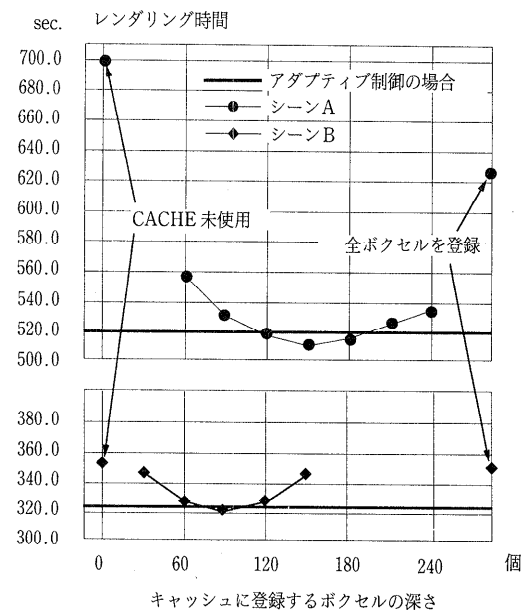


図-3 キャッシュデータのアダプティブ制御の効果

256×256×108の大きさの医療用データ(MRI)を用いて、320×320画素の画像を生成した場合の処理時間を図-3に示す。

ここで、シーンAは光線が通過するボクセルが最も多い場合で、シーンBは最も少ない場合である。

本研究では、レイトレーシングとボリュウムレンダリングの並列化において、ハイブリッド並列アルゴリズムとデータコヒーレンスを利用した方法について述べた。

実験の結果、どちらのレンダリング方法の並列化においても、良好な速度向上を得ているといえる。

しかしながら、高速化を妨げる大きな要因として、プロセッサ間の通信のボトルネックがあげられる。

今後は、さらに通信量を軽減するためのデータコヒーレンスの利用方法について検討を行う予定である。

## 5. ま と め

以上に、本プロジェクトで開発している応用プログラムの一部について紹介した。これらの他にも、論理合成、論理検証、行列計算、組合せ問題など幅広い分野への応用が期待されている論理関数の新しいデータ構造である並列2分決定グラフ(BDD: Binary Decision Diagram)の処理の並列化や、新しいネットワークポロジの提案など、離散アルゴリズムの並列化についても、研究を進めた。

さらに、逐次プログラムの自動並列化コンパイラや、有限要素法自動ソルバ(自動要素分割、結果可視化など)などの開発も進めている。

合わせて、種々の並列処理プログラムを収集したベンチマークプログラム集の作成なども行った。

## 参 考 文 献

- 1) Tatebe, O.: The Multigrid Preconditioned Conjugate Gradient Method, Proc. of 6th Copper Mountain Conference on Multigrid Methods, pp.621-634 (Apr. 1993).
- 2) 襲田 勉, 小柳義夫: マルチグリッド前処理付き自乗共役勾配法の並列化, IPSJ SIG Notes 94-HPC-52 (SWoPP '94), pp.59-64 (July 1994).
- 3) Tatebe, O. and Oyanagi, Y.: Parallelization

of the Multi-grid Preconditioned Conjugate Gradient Method, Proc. of JSPP' 93, IPSJ SIM Notes, pp.387-394 (May 1993).

- 4) Tatebe, O. and Oyanagi, Y.: Efficient Implementation of the Multigrid Preconditioned Conjugate Gradient Method on Distributed Memory Machines, Supercomputing '94 Proceedings, pp.194-203 (Nov. 1994).
- 5) 吉田紀彦: プログラム変換に基づくストリック・アレイの導出, 情報処理学会論文誌, Vol. 30, No. 12, pp.1530-1537 (Dec. 1989).
- 6) 吉田紀彦, 檜崎修二: 場と一体化したプロセスの概念に基づく並列協調処理モデル Cellula, 情報処理学会論文誌, Vol. 31, No.7, pp.1071-1079 (July 1990).
- 7) Narazaki, S., Yamamura, H. and Yoshida, N.: Strategies for Selecting Communication Structures in Cooperative Search, Proc. FGCS '94 Workshop on Heterogeneous Cooperative Knowledge-Bases, to appear (1994).
- 8) Goto, M. and Muraoka, Y.: A Beat Tracking System for Acoustic Signals of Music, Proc. of ACM Multimedia '94, pp. 365-372 (Oct. 1994).
- 9) 山田, 安浦: FMPPによるレイトレーシングの高速化手法について, 情報処理学会研究報告, ARC-97-14 (1992).

(平成6年12月13日受付)



小柳 義夫 (正会員)

1943年生。1966年東京大学理学部物理学科卒業。1971年同大学院理学系研究科物理学専門課程修了。理学博士。同年同大助手。高エネルギー物理学研究所理論部門助手、筑波大学電子情報工学系講師、助教授、教授を経て、1991年東京大学理学部情報科学科教授。並列処理、数値解析、計算物理学に関する研究に従事。特に、偏微分方程式の高速並列解法、最小二乗法の数値計算、乱数やモンテカルロ法に興味を持つ。物理学会、日本統計学会、応用統計学会、計算機統計学会、応用数学会など各会員。



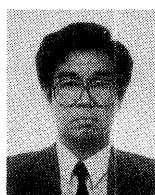
白川 友紀

1949年生。1972年大阪大学基礎工学部制御工学科卒業。1977年大阪大学大学院基礎工学研究科修了。工学博士。同年大阪大学基礎工学部助手。1978年大阪府立大学工学部助手。1981年筑波大学構造工学系講師。1987年同助教授。並列計算機とその応用について、自律分散システムについて興味を持つ。IEEE、電子情報通信学会、システム制御情報学会など各会員。



中嶋 正之 (正会員)

昭和44年東京工業大学工学部電気卒業。昭和50年同大学院博士課程修了。同年同大勤務。昭和58年同大像情報工学研究施設助教授。平成4年同大電気電子工学研究科教授。現在に至る。コンピュータグラフィックス、コンピュータアニメーション、画像処理の研究に従事。工学博士。



深澤 良彰 (正会員)

昭和51年早稲田大学理工学部電気工学科卒業。昭和58年同大学院博士課程中退。同年相模工業大学工学部情報工学科専任講師。昭和62年早稲田大学理工学部助教授。平成4年同教授。工学博士。ソフトウェア工学、コンピュータアーキテクチャなどの研究に従事。電子情報通信学会、ソフトウェア科学会、IEEE、ACM各会員。



村岡 洋一 (正会員)

昭和40年早稲田大学理工学部電気通信卒業。昭和46年イリノイ大学大学院博士課程修了。同年日本電信電話公社電気通信研究所入所。昭和60年早稲田大学理工学部教授。Ph. D. 並列処理アーキテクチャ、電子化図書、ニューロネットワークなどに興味をもつ。



安浦 寛人 (正会員)

昭和51年京都大学工学部情報工学科卒業。昭和53年京都大学工学研究科修士課程(情報工学専攻)修了。昭和55年より京都大学工学部助手。京都大学工学部電子工学科助教授を経て、平成3年より九州大学大学院総合理工学研究科情報システム学専攻教授。VLSIシステムの設計手法とCADの研究およびハードウェアアルゴリズムの研究に従事。昭和57年電子通信学会学術奨励賞、昭和63年および平成6年電子情報通信学会論文賞、平成4年情報処理学会論文賞、平成5年情報処理学会坂井記念特別賞およびBest Author賞をそれぞれ受賞。電子情報通信学会、情報処理学会、IEEE、ACM、EATCSなど各会員。



吉田 紀彦 (正会員)

1957年生。1979年東京大学工学部計数工学科卒業。1981年同大学院修士課程修了。(株)三菱総合研究所、東京大学工学部を経て、現在、九州大学工学部情報工学科助教授。工学博士。プログラミング方法論、並列・分散・協調処理、人工知能などに興味を持つ。計測自動制御学会、ソフトウェア科学会、ACM、IEEE各会員。