

多段ダイアコプティクスによる Poisson 方程式の並列計算手法†

渡辺 成良*・宮崎 卓幸*
茂木 和弘*・大出 高義*

ABSTRACT A new parallel computation method to solve Poisson equation is proposed in this paper. It is based on multi-step diakoptics that we developed from diakoptics originated by Kron. To apply the method for solving Poisson equation which is appeared in device simulation, we present the algorithm and data structure. We also discuss the computational complexity and communication complexity of one dimensional device model. Quantity of the complexities is evaluated by theoretical approach and execution on cellular array processor CAP. The efficient result was obtained from the example.

1. はじめに

近年、多数のプロセッサを結合したマルチプロセッサ型コンピュータ（高並列計算機）が実現段階に達してきており¹⁾、高速処理を必要とする特定の分野でシミュレーションエンジンが開発されるようになってきた²⁾。星野は PAX コンピュータを用いて Poisson 方程式の並列計算手法の評価を報告している³⁾。方程式の求解では線形方程式を解くことが要求され、この解法ではガウス消去法や LU 分解など解への収束性の問題がなく、しかも汎用性の高い直接法が望ましい。しかし直接法のアルゴリズムは一般に並列化が難しく^{4,5)}、星野は 3 項方程式における CR 法や反復法である SOR 法に対しては並列計算手法を導入し、それらの評価を行なった。

本文は半導体素子内部の電気的特性を数値解析により求めるいわゆるデバイスシミュレーションの並列計算手法を対象とした。計算過程で頻繁に要求される線形連立方程式は一般に非対称の帯行列か疎行列と見なせる構造になる。デバイスシミュレーションは離散化による格子点数が数万から数十万にも達するため、オ

リジナルな数値計算手法⁶⁾に工夫を加えた反復法を用いて単一計算機での高速化をはかっている⁷⁾。深尾らは Kron が提案した Diakoptics（ダイアコプティクス）を拡張して⁸⁾、並列計算手法である多段ダイアコプティクス（MSD）を開発し、これを回路シミュレーションへ適用した⁹⁾。直接法の一つである MSD は疎行列となる線形連立方程式に適した分割解法であり、網目状回路の並列計算にも適用できる。この方法は分割された部分回路ごとに計算機を割付けて部分回路問題を解き、計算機間の通信路を利用してデータ交換を繰り返し、部分回路の解を修正して正解を導出する。

そこで本文では、デバイスシミュレーションで解かれる Poisson 方程式の有限差分法近似から作られる線形連立方程式に対して MSD を適用する方法を示す。また 1 次元素子モデルについて、アルゴリズムにおける計算の複雑さや計算機間の通信の複雑さを評価する方法と、64 台のグリッド配置 MIMD 並列計算機での計算結果を報告する。更に 2 次元素子モデルへの拡張を考察し、MSD の適用性や有用性を議論する。

2. デバイスシミュレーション

Poisson 方程式の解法は Selberherr の素子モデルと数値計算⁶⁾を基本にした。すなわち

$$\lambda^2 \operatorname{div} \operatorname{grad} \Psi - (n - p - C) = 0 \quad (1)$$

を用いた。この式は電位 Ψ 、電子密度 n 、正孔密度 p を未知変数とする偏微分方程式とみなされる。他の変

An Parallel Computation Method of Poisson Equation based on Multi-Step Diakoptics. By Shigeyoshi Watanabe, Takayuki Miyazaki, Kazuhiro Motegi and Takayoshi Ohode (Dept. of Electronic Eng., Faculty of Engineering, Gumma University).

*群馬大学工学部電子工学科

†1988年10月6日受付 1989年1月17日再受付

数の定義, 値の求め方については文献⁶⁾を参照されたい.

2.1 有限差分法

有限差分法では偏微分方程式を差分近似により線形連立方程式に変換してその反復計算から解を得る. 電子と正孔の電流連続方程式を加えた3基本式は離散化され, Coupled法では離散化した各基本式を一括して, Decoupled法では単独に解かれる⁷⁾. ここでは後者を対象にした. このとき差分近似式は

$$Ax=b \tag{2}$$

なる線形連立方程式で表現される. 境界条件が与えられる境界格子点を除く格子点の数を n とすれば, 正方形行列 A とベクトル x, b の次数は n である. 図1は, x, y 軸の格子点数を共に4に離散化した2次元素子モデルの例である. 今, 境界格子点がこの外側を囲んでおり,

- a. 等格子間隔
- b. Dirichlet 境界条件

のもとで中心差分公式により離散化する. 式(1)の格子点 i における電位 Ψ_i の離散化式は

$$\frac{2\Psi_i - \Psi_r - \Psi_l}{\Delta^2} + \frac{2\Psi_i - \Psi_u - \Psi_d}{\Delta^2} = \frac{1}{\lambda^2} (C_i + p_i - n_i) \tag{3}$$

となる. ここで $\Psi_r, \Psi_l, \Psi_u, \Psi_d$ はそれぞれ格子点 i の右, 左, 上, 下にある格子点 r, l, u, d の電位を示す. これらの格子点は図1中のある格子点に隣接した枝の他方に隣接する格子点になっており, 電位が定まった境界格子点であってもよい. Δ は格子点間隔である. 従って A (次数 n は16) の要素は

$$a_{ii} = 4, \tag{4}$$

$$a_{ij} = \begin{cases} -1 : \text{格子点 } i, j \text{ が同一の隣接枝をもつ} \\ 0 : i, j \text{ が同一の隣接枝をもたない} \end{cases}$$

となるように規格化できる. 図1の格子点番号付けで

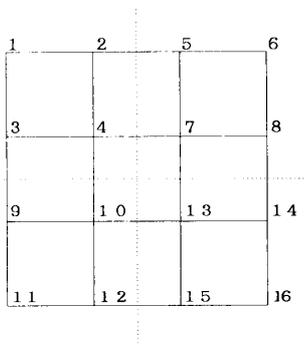


図1 有限差分法による2次元素子モデルの離散化

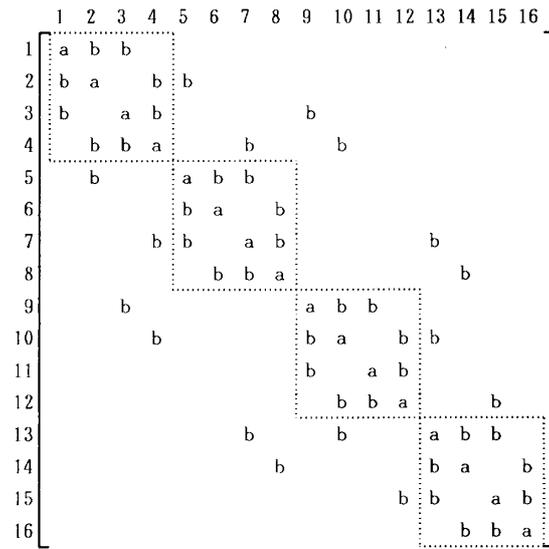


図2 図1の格子点番号付けによる行列 A の非零要素 ($a=4, b=-1$)

は A の非零要素が図2のように現われる. A はバンド幅が7の帯行列となる.

2.2 帯行列の計算量

素子モデルの格子点数が増えれば, 行列 A の次数は増加する. 表1は素子モデルと行列の次数, 行や列に含まれる非零要素数, バンド幅との関係を示す. 行列の非零要素の数は A の次数に無関係で少なく, A は疎行列となる. しかしバンド幅 w は次元で大きく変化し, Natural Ordering の番号付けでは表のようになる. このためモデルの次元と格子点数が増えるだけでバンド幅が増加し, 大規模疎行列の解法が要求される. 次数 n の正方形行列 A の LU 分解と前進・後退代入による直接法を, バンド幅 w の帯行列となる式(2)に適用したときの乗除算の計算量は (帯中は全て非零要素とみなす), LU 分解が

$$nw(w-1) - \frac{w(w-1)}{3} (2w-1) \tag{5}$$

であり, 前進・後退代入 (FB) は次のようになる.

表1 隣接格子点差分近似式による行列 A の構造

モデル	行列の次数	非零要素数	バンド幅
1次元	L_x	3	2
2次元	L_x, L_y	5	$\min_{i=x, y} L_i + 1$
3次元	L_x, L_y, L_z	7	$\min_{i,j=x, y, z} L_i, L_j + 1$

L_i : 直交座標系の i ($=x, y, z$) 軸の格子点数

$$n(2w-1) - w(w-1) \quad (6)$$

これら評価式は帯行列に対する直接法の最小乗除算回数を示している。加減算回数は乗除算のそれにほぼ比例しているため、以下では乗除算のみを考える。

さて軸格子点数がどの次元でも L に等しいときの乗除算の概数は、モデルが 1 次元で $5L$ 、2 次元では L^4 、3 次元では L^7 となり、 $L=100$ としても 3 次元では 10^{14} のオーダーの乗除算が要求される。このため直接法は実用的でない結論され、不完全 LU 分解と共役勾配法を組合せた反復法などが研究されている。しかしこの反復法の収束問題が偏微分方程式の差分近似による反復計算の収束性に加えて新たに生じた⁷⁾。そこで本文は並列計算機を用いて式(2)の解析に直接法の一つである MSD を適用し、計算の高速化をはかる。

3. 分割と結合による直接法

式(3)の対角項 a_{ii} は以下の式で作られる。

$$a_{ii} = - \sum_j a_{ij} \quad (7)$$

ここで j は格子点 i に隣接した枝の他方で隣接する格子点であり境界格子点でもよい。このとき A は対称正定値となり、境界格子点に隣接した枝と隣接する格子点（境界隣接格子点と呼ぶ：図1で5, 6, 8など）に対応する行で対角優越的となる。この A は格子点（節点）に隣接する枝が 1 オームの抵抗となる回路網において、境界隣接格子点と境界格子点（基準点）も 1 オームの抵抗を介して接続されているとして作られる節点アドミタンス行列に等しい。Kron のダイアコプティックスやそれを並列化した MSD は、回路網解析からのアナロジーとして、式(2)のような有限差分法から導かれた線形連立方程式を回路の分割と結合により効率的に解く計算手法である。ダイアコプティックスは式(2)の A を用い

$$x = A^{-1} b \quad (2)'$$

を直接解くかわりに、 A を導出したモデル（差分近似式と等価な回路方程式をもつ回路）を分割して、互いに分離した部分回路の集合とそれらを結合する素子の集合で表現する。この表現に従えば、 A は

$$A = A_0 + CZ^{-1}C' \quad (8)$$

となる。これは A を部分回路ごとに回路方程式を作成して得られる行列 A_0 と、それらを結合する素子の値をもつ行列 Z およびそれら素子と部分回路内の節点との隣接関係を示す行列 C で作られる行列 $CZ^{-1}C'$ の和で置き換える（従って A を分割して単にブロック対角行列と非ブロック対角行列の和とするのではな

い）。ここで C' は C の転置行列、 A_0 は格子点の数の次数となるブロック対角行列、 Z は結合する素子の数の次数となる対角行列である。これら行列と差分近似式との関係や、これらを素子モデルから直接導出する方法は次節以降で詳しく述べる。

式(8)を用いれば、解 x は次式から得られる。

$$x = A_0^{-1} b - A_0^{-1} C (Z + C' A_0^{-1} C)^{-1} C' A_0^{-1} b \quad (9)$$

この式は逆行列補助定理からも導出できる⁸⁾。ダイアコプティックスは、式(9)が(2)'より効率的に計算できるように A_0, Z, C を決定している。回路方程式では x と b が電圧と電流ベクトル、 A_0^{-1} と Z がインピーダンス行列となる。従って式(9)右辺は、 $A_0^{-1} b$ が分割された回路の節点電位ベクトルで、 $C' A_0^{-1} b$ と $C' A_0^{-1} C$ が結合する素子に隣接した節点の開放点電位ベクトルとそれらの節点からみた開放点インピーダンスである。このように右辺第1項は分割された回路の電圧解、第2項は結合による電圧修正を意味しており、ダイアコプティックスは分割モデルの解を、もとのモデルに復帰させる操作で修正して正解を得る方法であることが次節の説明でわかる。

3.1 モデルの分割

まず分割問題を解くために、与えられた素子モデルを幾つかの部分素子領域に分割し、それぞれの領域で方程式を作成する。分割の方法としては図1のような点線で切られる分割点の枝（結合枝）に対して、それらを取り除くか分割点を全て境界格子点とみなすかの2つが考えられる。前者はダイアコプティックスの枝分割法であるが、この分割により Dirichlet 境界条件が与えられない部分素子領域が出現すると、その領域で作成された離散化式の行列が特異行列（結果として A_0 が特異行列）となる問題が生じる。そこで自由な分割が特異行列を出現させないようにするため、ここでは後者による分割を用いる。

今、 m 本の結合枝により N 個の部分素子領域に分割されたとし、各領域ごとに格子点番号を新しく付け直す。最初に、結合枝に隣接した領域の格子点の項を除いた式(3)の離散化式を領域内の全ての格子点の周りで作成する。部分素子領域 h の作成された部分行列を A_h とおく。結合枝 p に隣接する格子点の A における（分割前の）番号を i, j とする。領域 h に格子点 i が含まれており、この領域で付け直した i の番号を f とする。このとき A_h の要素 a_{ff} を $a_{ff} - a_{ij} - a_{ji}$ に直す。ここで a_{ij} と a_{ji} は A の要素である。同様に j について直し、この修正を全結合枝に対して行なう。式(8)の行列 Z は結合枝数に等しい次数の対角行列である。

その対角要素 z_{pp} は結合枝 p に対応し, z_{pp} は $a_{ij}(=a_{ji})$ の逆数に等しい. ここで図1中の点線で素子領域を4分割する例を考えよう. 分割された部分素子領域を図3(a)のように h, i, j, k とおく. 例えば領域 h では最初

$$A_h = \begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 3 & 0 & -1 \\ -1 & 0 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix} \quad (10)$$

となる. これは図2で領域 h に対応するブロック対角項を取り出し, 残されたブロック非対角項の行要素を対応する行の対角項に加えた行列に等しい (これは結合枝を取り除く分割で作られた行列であり, ダイアコプティックスの枝分割法に対応する). 次に結合枝の1本に隣接する格子点2と3, および2本に隣接する格子点4の対角項を修正する.

$$A_h = \begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 5 & 0 & -1 \\ -1 & 0 & 5 & -1 \\ 0 & -1 & -1 & 6 \end{bmatrix} \quad (10)'$$

これは図2で領域 h に対応するブロック対角項を取り出し, 残されたブロック非対角項の行要素を対応する行の対角項から引いた行列に等しく, 式(10)に比べてより対角優越的となる. このようにして作成された行列を対角項に並べたブロック対角行列が式(8)の A_0 である. このため式(9)の $A_0^{-1}b$ は

$$A_{0h}x_h = b_h \quad h=1, \dots, N \quad (11)$$

から決定できる. しかもこれらは並列に計算できる.

各領域ごとに式(11)で示す離散化式を直接定めるには, 結合枝に対応する A のブロック非対角項の行と列要素の値をそれぞれもつ2本の枝の一方に境界隣接格子点を, 他方に2つの追加した境界格子点を隣接させ, 領域内の全格子点 (境界格子点は含まれない) の周りで離散化式を作成すればよい. 例えば格子点2では値が等しい a_{25} と a_{52} の2本の枝が追加の境界格子点に接続されるから, 式(7)の j としてはこれらの他

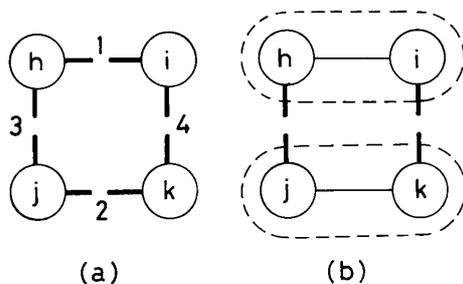


図3 図1の点線による分割と段階的結合の例

に, 格子点1と4, 更に2の上の境界格子点に加わる.

さて A_0, Z, A を式(8)に代入すれば, 行列 C は格子点と結合枝との隣接関係を表わす隣接行列となっている. その i 行 j 列要素は,

$$c_{ij} = \begin{cases} 1: \text{格子点 } i \text{ が結合枝 } j \text{ に隣接する} \\ 0: \text{格子点 } i \text{ が結合枝 } j \text{ に隣接しない} \end{cases} \quad (12)$$

で決定される.

3.2 復帰操作

2つの部分素子領域を結ぶ結合枝の集合を結合束, ある結合束の分割点をもとに戻す操作を復帰と呼ぶことにする. 今, m 本の結合枝が M 個の結合束に束ねられているとする. 分割された2つの部分素子領域はそれらに隣接した同一の結合束の復帰により, 一つの領域 (結合領域) を生成する. 復帰操作は分割された結合領域にそれぞれ含まれる部分素子領域を結合して拡大結合領域を生成したり, 同一の結合領域にある部分素子領域間を結合して更新したりする.

この章の初めに述べたように, ダイアコプティックスは結合前の節点の解と開放点インピーダンスを用い, 全結合束を一段階で復帰させて全結合を完了する. これは式(9)右辺第2項の結合における修正のための行列

$$W = Z + C^t A_0^{-1} C \quad (13)$$

の次数が結合枝の数に等しいため計算量が膨大になる. そこで MSD では復帰操作を段階的に繰り返しながら全結合を完了する. これは式(8)の右辺第2項を M 個ある結合束ごとに分解した式のと表現し,

$$CZ^{-1}C^t = \sum_{p=1}^M C_p Z_p^{-1} C_p^t \quad (14)$$

としている. 行列 C_p と Z_p は結合束 p に対応した C と Z の部分行列である. 従って式(8)の右辺は

$$A^{(p)} = A^{(p-1)} + C_p Z_p^{-1} C_p^t, \quad p=1 \dots M \quad (15)$$

なる再帰式で表現される⁹⁾ ($A^{(0)} = A_0, A^{(M)} = A$). 上つき添え字は再帰のステップ数を, 下つき添え字は結合束番号を示している. この式では結合束の番号順に段階的に復帰操作を行なうとしており, 逆行列補助定理を M 回適用して解を得ている. そこで式(9)は

$$x^{(0)} = A_0^{-1}b, \quad U_p^{(0)} = A_0^{-1}C_p, \quad p=1 \dots M$$

を結合前の解とし, ステップ p で結合束 p が結合されるとして, 以下でステップ s を1から M まで進める.

$$W_p = Z_p + C_p^t U_p^{(s-1)} \quad (16)$$

$$x^{(s)} = x^{(s-1)} - U_p^{(s-1)} W_p^{-1} C_p^t x^{(s-1)} \quad (17)$$

を求め, 更に $s < M, k = p, \dots, M$ について

$$U_k^{(s)} = U_k^{(s-1)} - U_p^{(s-1)} W_p^{-1} C_p^t U_k^{(s-1)} \quad (18)$$

を計算する。式(17), (18)はそれぞれ結合束 p の結合による解と開放点インピーダンスの修正式である。

4. 並列計算手法

4.1 MSD の並列性

MSD の並列性の 1 つに分割された部分素子領域の結合前の解の計算があった (式(11)など)。そこで各領域ごとに計算することにし、式(17), (18)を各領域ごとに分割して復帰操作の並列性を考える。図 1 は分割数 N が 4 であり、各部分素子領域はそれぞれ 2 本の結合枝からなる 2 つの結合束に隣接する。そこで図 3 (a) のような記号を用いれば、結合束 1 は部分素子領域 h と i に隣接する。このとき隣接行列 C_1 の行は各領域に対応して 2 つの零でない行列

$$C_{h,1} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, C_{i,1} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (19)$$

と零である行列 $C_{j,1}, C_{k,1}$ の 4 ブロックに分割できる。ここで各行列の第 1 列は結合枝 2-5, 第 2 列は結合枝 4-7 で、その行は各領域内の節点に対応する。従って部分素子領域 g に対して式(17)は

$$x_g^{(s)} = x_g^{(s-1)} - U_{g,p}^{(s-1)} W_p^{-1} \cdot \sum_f C_{f,p}^t x_f^{(s-1)} \quad (20)$$

となる。ここで f は結合束 p に隣接した 2 つの部分素子領域である。また結合束 r から見た領域 g における開放点インピーダンスは、式(18)から $s < M, r = p, \dots, M$ について

$$U_{g,r}^{(s)} = U_{g,r}^{(s-1)} - U_{g,p}^{(s-1)} W_p^{-1} \cdot \sum_f C_{f,p}^t U_{f,r}^{(s-1)} \quad (21)$$

となる。このため g に r が隣接していなくても、結合ステップが進み r と隣接した領域と g が同一の結合領域に含まれたならば、右辺第 2 項により左辺は零でなくなる。

今、図 3 (a) の例で結合束 1 をステップ 1 で結合するとしよう。領域 h と i の計算は式(20), (21)となるが、その計算に領域 j と k のデータは必要ない。ステップ 2 で結合束 2 を結合すると計算は逆転する。このことは結合束 1 と 2 が同一のステップで結合できることを示している。すなわち図 3 (b) の点線のように、結合束の復帰操作で作られる結合領域が互いに交わらないならば、複数の結合束を同一のステップで結合できる。これが MSD のもう 1 つの並列性である。次のステップで結合束 3 と 4 の両方を復帰するのは、3 と 4 を 1 つの結合束として計算することになり、計算量

を増大させてしまう。このため一つの結合領域に復帰させる結合束はどの段階でも高々一つとするような操作を用いる。MSD の復帰操作は多様であるが、並列性を高めるには同一段階で出来るだけ多くの結合束を復帰させるほうがよい⁹⁾。本文では議論しないが MSD には並列性の高い復帰操作の手順を決定する、すなわち結合スケジューリングの問題がある。以下で計算機台数は N であり、各計算機は部分素子領域の計算を担当するものとする。最後に記号を定義する。

S_h : h を含む結合領域を構成する部分素子領域の集合

R_h : S_h 内の部分素子領域に隣接する結合束の集合

4.2 MSD のアルゴリズム

計算機台数と部分素子領域の数が等しいとする。以下に示す部分素子領域のデータが対応する各計算機に入力される (h 部分素子領域について示す)。

- 式(11)のための A_h の導出
- 式(2)における b の部分ベクトル b_h
- 集合 S_h として h
- 集合 R_h として h に隣接する結合束の集合
- h に隣接する結合束 p の隣接行列 $C_{h,p}$ と Z_p^{-1}
- 結合束の結合スケジュール

S_h や R_h は結合によるデータ転送で変化していく。計算機 h は以下の計算を担当する。

(1) 独立解

- A_h を LU 分解 (A_h^{-1} とおく) する
- 前進後退代入により以下を求める

$$x_h := A_h^{-1} b_h, U_{h,r} := A_h^{-1} C_{h,r}, r \text{ in } R_h$$

を実行する。記号 := は右辺の計算結果を左辺に代入する操作である。ここで x_h は直接法で求めている。分割無しの場合は当然 $U_{h,r}$ や次の結合解の計算は不用となり、一台の計算機で全体を解いたことに等しくなる。

(2) 結合解

次に結合スケジュールに従い結合解を求めることになるが、そこで必要なデータの作成と送信を説明する。あるステップで結合束 p が部分素子領域 i と j に結合されるとする。計算機 i と j はこのとき

$$v_k := C_{k,p}^t x_p, V_{k,r} := C_{k,p}^t U_{p,r}, r \text{ in } R_k \quad (22)$$

を定める。 $C_{k,p}$ の要素は 1 か 0 だから行列積の計算はしない。ここで k は i または j である。計算機 i は S_j を計算機 j に送信し、計算機 j から S_i を受信する。計算機 k ($= i, j$) は式(22)の結果と Z_p^{-1}, R_k, S_k を S_i, S_j でわかる送信先に送る。図 4 はステップ 1 で結合束 1 と 2, ステップ 2 で 3, ステップ 3 で 4 の結合束を結合

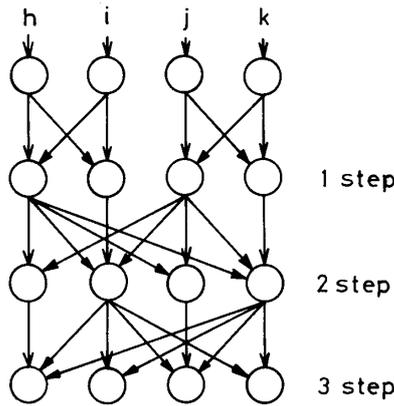


図4 並列計算機間のデータの送信

したときの各計算機のデータ送信を示す。

結合解を求めるために、ステップを1から最終ステップの L まで進めて以下の計算を繰り返す。

- ・そのステップで結合される結合束 p が h に隣接するならば、式(22)を求めてデータ送信する（この場合を以下では $h=i$ とする）。

- ・ $v := v_i + v_j$, $V_p := V_{i,p} + V_{j,p}$

$$R_h := R_h - p$$

- ・ r in R_h and r in R_j ならば

$$V_r := V_{i,r} + V_{j,r}$$

- ・ r not in R_h and r in R_j ならば $V_r := V_{j,r}$

- ・ $(Z_p + V_p)$ を LU 分解 (W_p^{-1} とおく) する

- ・ 前進後退代入により以下を求める

$$x_h := x_h - U_{h,p} W_p^{-1} v \quad (23)$$

- ・ もし最終ステップでなければ

$$R_h := R_h \cup R_j, S_h := S_h \cup S_j$$

$$U_{h,r} := U_{h,r} - U_{h,p} W_p^{-1} V_r, \quad r \text{ in } R_h$$

以上の計算により各計算機は対応する部分素子領域の要求された正解を導出している。

4.3 非対称行列への対応

格子間隔が不均一であったり、Neumann 境界条件での Newton 多項式近似の場合、Poisson 方程式の A は非対称行列となる。またデバイスシミュレーションでは、非対称行列となる放物線型の電流連続方程式も解く必要がある。有向グラフ法や2ステップ結合法⁸⁾は、非対称行列における MSD のアルゴリズムである。ここでは前者が適用できる。有向グラフ法は結合枝を外向き2本の有向枝に置き換え、部分隣接行列を外向き隣接行列 C_p^+ と内向き隣接行列 C_p^- に書き換える。図2で a_{25} と a_{52} および a_{47} と a_{74} が異なるとする。2つの結合枝に対して前者を h から i 向き、後者を i から h 向きとする。各隣接行列の第1, 第2, 第

3, 第4列をそれぞれ、2から5へ、5から2へ、4から7へ、7から4への有向枝に対応させれば、部分素子領域 h では次式となる。

$$C_{h,1}^+ = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (24)$$

$$C_{h,1}^- = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

同様な隣接行列を部分素子領域 i, j, k について作成し、行にこれらの行列を並べた結合束1の外向きと内向き部分隣接行列 C_1^+ と C_1^- を求める。このとき行列 A は

$$A = A_0' - \sum_{p=1}^M C_p^+ (Z_p')^{-1} (C_p^-)' \quad (25)$$

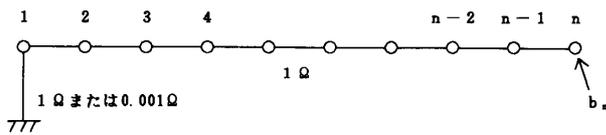
となる。ここで非対称行列 A を図2のように分割したとき、 A_0' はそのブロック対角行列そのものである。また Z_p' は例えば2から5への有向枝では a_{25} の逆数を要素 z_{11} とする対角行列となる。

5. 計算と通信の複雑さの評価

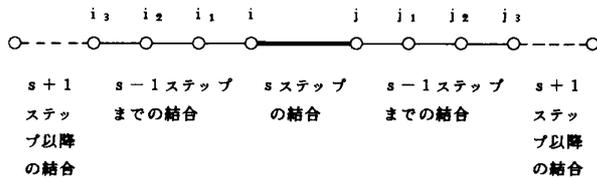
MSD のプログラムを開発し、セルラアレイプロセッサ CAP で実行した¹⁰⁾。CAP はセルと呼ばれる16ビットプロセッサ64台を2次元格子接続した MIMD 型並列計算機である。各セルは上下左右の隣接セル間に通信路を持ち、32ビットワークステーション (EWS) であるホストと共通バスで接続されている。隣接セル間以外のデータ転送には共通バスが用いられる。プログラムはC言語で書かれるが、データの送受信や同期のためのライブラリー関数が使われる。UNIX 上で CAP のプログラムを作成できしかもその実行が擬似的に模擬できるように、ソフトウェアシミュレータ CSS が用意されている。CSS はホストと同等機能の EWS で開発された並列プログラムのデバックとして利用できる。

5.1 結合スケジュール

問題として1次元素子モデルを考え、一端に境界条件を与えて差分近似式を作成すれば、問題は図5(a)の回路モデルと等価になる。格子点数や分割数は2の指数乗とした。この横並びのモデルを N 等分すれば、両端の領域を除く各部分素子領域は1本の結合枝からなる2つの隣接した結合束で左右の領域と分けられる。この問題の結合スケジュールは各領域でそれらの



(a)等価な回路モデル



(b)MSDの結合スケジュール

図5 一次元素子モデルの例

結合束（両端は1つ）をどのステップで復帰させるかの決定である。今、左端の領域から順番に1からNの番号を付ける。ステップs(≥1)において2^sで割り切れない番号の領域を選び出す。ここでs-1以前に選出された領域は除く。そして選出領域とその右領域との間の結合束を復帰させる。これは復帰操作で作られる結合領域の両端に隣接する結合束のいずれかを次のステップで復帰する方法であり、復帰の並列性が最も高い。図5(b)はs=3での復帰例(太線)であり、iは例えば4である。領域をセルに対応させれば、復帰操作は隣接セル間の通信路の設定と考えられる。

5.2 計算精度

EWSで開発したMSDのプログラムをCSSで実行し計算精度を調べた。図5(a)で式(2)のA, bは

$$a_{ii}=2, 1 \leq i < n, a_{nn}=1, \\ a_{i,i+1}=a_{i+1,i}=-1, 0 < i < n, \\ b_n=1$$

となり、他の要素は全て0である。このときAは次数nの増大に伴って悪条件に近づく。直接法(帯行列のLU分解, 前進・後退代入)とMSD(セル台数2と4)で求めたx₁の結果を表2(a)に示した。表2(b)はa₁₁=1.001の場合である。後者は前者よりも悪条件である。良条件では両者に精度の差は見られないが、悪条件ではMSDで計算機台数が増えるとやや精度の低下が見られた。

5.3 計算の複雑さ

セルでの計算量を評価するため、MSDのアルゴリズムに従って図5(a)のモデルの乗除算回数を計算した。次数nを2^u, 分割数Nを2^vとし(u>v), セルk

表2 直接法とMSDの計算精度の比較

(a) a₁₁=2.0でのx₁の計算結果

次数	直接法	MSD(2台)	MSD(4台)
16	1.0000001	0.9999998	1.0000010
32	0.9999997	0.9999994	1.0000000
64	1.0000002	0.9999994	0.9999960
128	1.0000003	0.9999961	0.9999958

(b) a₁₁=1.001でのx₁の計算結果

次数	直接法	MSD(2台)	MSD(4台)
16	999.5905	999.7778	1000.742
32	999.5452	999.3369	999.5811
64	1000.872	999.3043	998.5576
128	1000.862	999.2875	998.9216

は(k-1)・2^{u-v}+1からk・2^{u-v}までの格子点からなる領域を計算する。復帰操作はステップvで終了する。バンド幅2で次数mの行列のLU分解は式(5)より

$$t_{LU}=2m-2 \tag{26}$$

となる。1ベクトル当たりの前進・後退代入の乗除算回数は式(6)より

$$t_{FB}=3m-2 \tag{27}$$

となる。セル1台でのMSDのアルゴリズムは従来の直接法と同一になり、次数nの独立解を解くことになるから、乗除算回数は5n-4である。これをi'としてnとの関係を示せば図6の点線となる。これに対してセルN台での並列計算では、各セルが次数2^{u-v}の独立解と次数1の結合束の結合解を計算する。前者の乗除算回数はm=2^{u-v}で3ベクトルを計算するから、11m-8となる。後者は式(23)のx_hの計算をv回、U_{h,r}をv-1回行なうから、(3v-2)(m+1)となる。Nを変えてこれらの和を示せば図6の実線となる。

このモデルについてCAPによる並列計算を実行し、実行時間を実測した。図7がその結果である。セル台数を増やすと、格子点の多いモデルで並列化によるMSDの有効性が認められる。格子点が少ない場合に図6と異なる理由は、図6では通信時間を無視したためである。

5.4 通信の複雑さ

セル間のデータ転送による通信時間は、1データの通信時間がその演算処理時間の数十倍以上といわれており、セル台数が多い場合に無視できない。図7を求めたMSDのスケジュールは、ステップsで初めて

k が 2^l で割り切れないときにセル k と $k+1$ 間の結合束が復帰する。このスケジュールによれば Z_p^{-1} , R_k , S_k は各セルに前以て与えることができるために送信

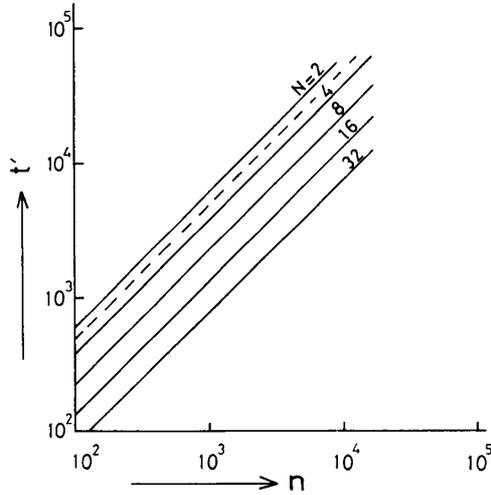


図6 行列Aの次数 n と乗除算回数 t , 計算機台数 N との関係 (図中の点線は $N=1$ を示し, 直接法の逐次処理となる)

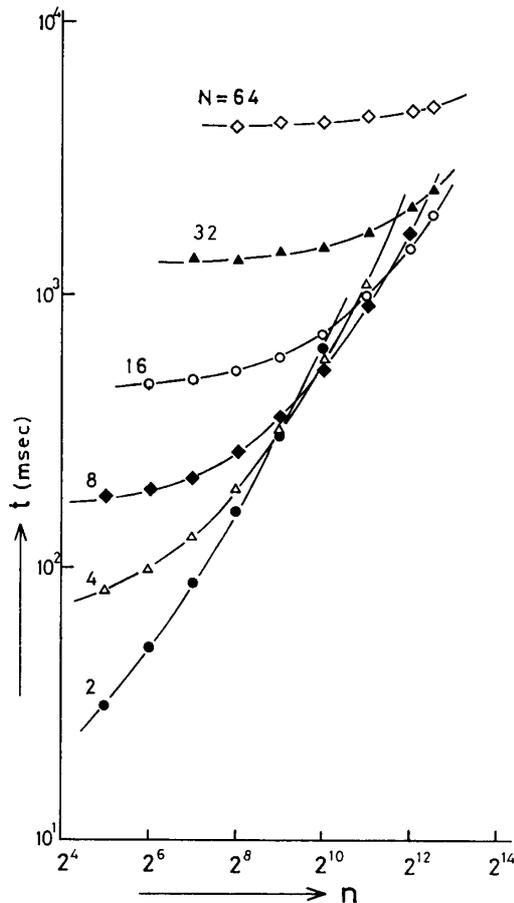


図7 並列計算機CAPによる実行時間 t の実測値

しなくてよい。セル間通信の説明のために, 図5 (b)の i と j をセルに対応させる。それらは式(22)で求めた $v_k, V_{k,r}$ (r は R_k の要素, k は i または j)をまず i から j へ, 次に j から i へ送信する。更に $v_k, V_{k,r}$ (k は i と j)を, i は i_1 へ, 並列に j は j_1 へと次々に隣接セル間の通信で送る。終端は S_k でわかる。 v_k と $V_{k,r}$ は次数1だから, 送信データ量は格子点数や分割数に無関係で6データとなる。1データの隣接セル間の通信時間を t_{send} とすれば, セル台数 N が2, 4, 8, 16と増加するにつれて, 全ステップでの1データ送信時間は t_{send} の1, 3, 7, 15倍に増加する。図8は t_{send} が乗除算1回当たりの時間の25倍として図6に加えた実行時間の推定値 t^* である。図7と同様に n のある区間で t^* を最小とする N が存在する。

5.5 2次元素子モデルへの拡張

1次元素子モデルに対するMSDの有効性がCAPの使用により示された。2次元以上の素子モデルの自由な分割に対しては, 並列性の高い結合スケジュールを求めるアルゴリズムや通信時間を短縮できるデータ転送方法の決定が実行時間の短縮のために必要である。しかし任意の分割においてこれらを決定する研究は少なく, 現在は与えられた分割に対してアドホックに対応することになる。例えば図1で x 軸と y 軸の格子点数を共に n とする。乗除算回数は式(5), (6)から, 直接法は n が大きいきほ n^4 となる。モデルを4等分した図3のスケジュールによるMSDは, 独立解に $5n^4/16$, 結合解に n^3 程度であるから, 通信時間を無視すれば4台のセルにより2倍以上高速に解け

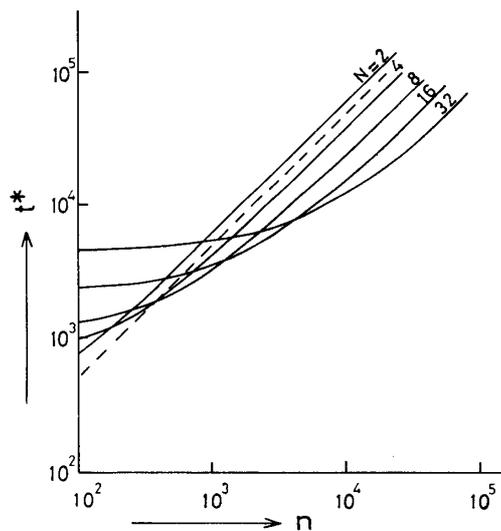


図8 行列Aの次数 n と実行時間の推定値 t^* , 計算機台数 N との関係 (図中の点線は $N=1$ を示す)

る。1次元素子モデルで用いたスケジュールや隣接セル間通信方法は、一列に並んだ形の分割に適しているため、横長の2次元素子モデルの横方向分割にも適用できる。このとき結合束の次数は1から縦の格子点数になり、1次元モデルよりも結合解の計算量とデータ転送量が増加する。

6. おわりに

多段ダイアコプティックス (MSD) のアルゴリズムを示し、Poisson 方程式への適用法を提案した。そして1次元素子モデルではあるが並列計算機 CAP による実行時間の計測と MSD の理論計算式を示し、このモデルでの MSD の有効性を示した。現在、横長や一般的な2次元素子モデルの解析に適用できるようにプログラムを開発中である。最後にこの研究への助言を下さいました東京理科大学基礎工学部教授の深尾毅先生、プログラム開発を担当した群馬大学の新堀宏和君（現在日立製作所）、鹿沼章君および金井淳一君に感謝します。また並列計算機 CAP の使用を許可されました富士通研究所部長石井光雄氏をはじめとし、CAP の利用法を御教示下さいましたシステム研究部

の方々に深く感謝致します。

参考文献

- 1) 天野英晴：マルチプロセッサ型スーパーコンピュータ，電子情報通信学会誌解説，70-12, 1263/1274 (1987)
- 2) 小池誠彦：シミュレーションエンジン，電子情報通信学会誌解説，70-7, 728/737 (1987)
- 3) 星野力：PAX コンピュータ，オーム社(1985)
- 4) D. J. Evans, : Parallel Processing Systems, Cambridge University Press (1982)
- 5) D. A. Reed, et al.: Multicomputer Network: Message-based Parallel Processing, MIT Press (1987)
- 6) S. Selberherr: Analysis and Simulation of Semiconductor Device, Springer-Verlag (1984)
- 7) 佐藤，羽根，鈴木：デバイスシミュレーション用の高収束な行列計算方法，電子通信学会論文誌 (C), 69-C-11, 1389/1396 (1986)
- 8) 深尾，寺辻：ダイアコプティックスの拡張とその回路網特異分割問題への応用，電気学会論文誌(C), 102-C-2, 29/36 (1982)
- 9) 渡辺，深尾：マルチ・ステップ・ダイアコプティックスと大規模回路網の並列計算手法，電子情報通信学会論文誌 (A), 70-A-2, 220/227 (1987)
- 10) 池坂，堀江，石井：セルラアレイプロセッサ CAP 用セル OS，電子情報通信学会技術報告，CAS86-209, 43/50 (1987)