

帯行列の線形計算におけるマルチ・ステップ・ダイアコプティックスの並列性の評価

4B-1

鹿沼 章

金井 淳一
群馬大学 工学部

渡辺 成良

1. まえがき

素子の高速化やスーパーコンピュータによる逐次処理とともに、並列処理による高速計算の研究が盛んになってきている^[1]。また、ハードの面から見ても複数の計算機を結合して利用する並列計算機が実用化されつつある^[2]。国内で開発された並列計算機システムである筑波大学のPAXでは、科学技術計算における各種の並列処理アルゴリズムの評価が行われ、本文で議論する線形計算では、3項対角行列についてADI法を用いた実測結果を得ている^[3]。しかし、この方法では、大規模回路網解析などで生じる疎行列への拡張性が乏しい。そこで、本文はこのような疎行列や帯行列の解析に適するマルチ・ステップ・ダイアコプティックス^[4]（以下、MSDと略）を、帯行列となるメッシュ構造をした線形回路網の解析に適用し、その並列化の有効性を確かめる。

2. MSDのアルゴリズム

回路網における注入電流ベクトルを i 、節点对アドミタンス行列を Y とすると、節点对電圧ベクトル e は次式で与えられる。

$$e = Y^{-1}i \quad (1)$$

ここで図1のような節点数 n の $b \times c$ ($c = n/b$) のメッシュ構造をした回路網を考えると、 Y はバンド幅 $b+1$ の $n \times n$ 次の行列となる。この回路網を図1のように縦に分割し、 m 個の部分回路網を作る。分割とは線形素子からなる、 b 本の結合枝を2分し接地することである。分割によってできた左から k 番目の部分回路網 N_k の節点数を n_k 、注入電流ベクトルを i_k 、節点对電圧ベクトルを e_k 、節点对アドミタンス行列を $Y_{0,k}$ とすると、ダイアコプティックスの解は

$$e = Y_0^{-1}i - Y_0^{-1}C(Z + C^t Y_0^{-1}C)^{-1}C^t Y_0^{-1}i \quad (2)$$

で与えられる。ここで Y_0 は $Y_{0,k}$ を k 番目の対角要素とするブロック対角行列である。 C は節点-結合枝隣接行列で、部分回路網 N_k の節点-結合枝間隣接行列 C_k の h, j 要素の値は、 N_k の節点 h に結合枝 j が隣接するときは1、節点 h に結合枝 j が隣接しないときは0で与えられる。このとき C_k は $n_k \times b$ 次の行列となる。また、 C^t は C の転置行列である。節点对アドミタンス行列 Y について逆行列補助定理を1度だけ適用すると、

$$Y = Y_0 + CZ^{-1}C^t \quad (3)$$

と書け、この式を(1)式に代入して書き換えると(2)式となる。

ダイアコプティックスでは部分回路網を1回のステップで結合し、修正を加えることによって元の回路網の(1)式をそのまま計算したのと同様な結果を得ることができる。これに対し、MSDでは全結合枝を幾つかの結合枝の集り(結合束)に分割し、これらの結合束を段階的に結合していくものであり、このことは逆行列補助定理を繰り返し適用すること

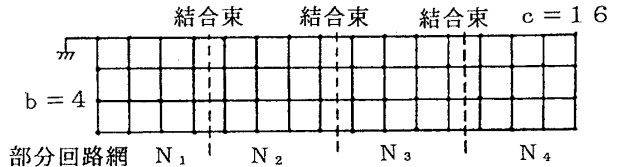


図1 メッシュ構造をした回路網 ($n = 64$)

に対応する。メッシュ構造をした回路網を図1のような縦の分割で、 $m = 2^L$ 個の部分回路網に分割した場合、ステップ s ($s \geq L$) における結合を終了すると 2^s 個の部分回路網が結合されたことになる。例えば、8個の部分回路網について1ステップが終了すると (N_1, N_2) 、 (N_3, N_4) 、 (N_5, N_6) 、 (N_7, N_8) というように2個ずつ結合されたことになる。次に2ステップ目が終了すると、 (N_1, N_2, N_3, N_4) と (N_5, N_6, N_7, N_8) の2つのグループになり、3ステップ目では全ての部分回路網が結合されて元の回路網と等しくなる。ここで1つの部分回路網の計算を並列計算機の1つの計算機に対応させる。 k 番目の計算機における計算式は次のようになる。

(イ) 部分回路網段階での計算 ($s = 0$)

$$e_k = Y_{0,k}^{-1}i_k \quad (4)$$

$$U_{k,r} = Y_{0,k}^{-1}C_{k,r} \quad (5)$$

(ロ) 結合段階での計算 (ステップ $s = 1, 2, \dots, L$)

$$p = F_k^{(s)} \quad (6)$$

$$v_i = C_{k,p}^t e_k \quad (7)$$

$$V_{i,r} = C_{k,p}^t U_{i,r} \quad (8)$$

$$v_j = v_i + v_j \quad (9)$$

$$V_p = V_{i,p} + V_{j,p} \quad (10)$$

$$Y_p^{-1} = [-Z_p + V_p]^{-1} \quad (11)$$

$$e_k = e_k - U_{k,p} Y_p^{-1} v \quad (12)$$

$$U_{k,r} = U_{k,r} - U_{k,p} Y_p^{-1} V_p \quad (s < L) \quad (12)$$

まず、部分回路網段階では(4)、(5)式によって節点对電圧ベクトル e_k 、結合束 r に関する節点-開放端間相互インピーダンス行列 $U_{k,r}$ の計算を行う。次に結合段階ではステップ s における結合スケジュール $F_k^{(s)}$ が p と一致するとき、結合束 p に隣接する部分回路網では(6)、(7)式によって b 次の開放端端子電圧ベクトル v_i 、結合束 r に関する開放端相互インピーダンス行列 $V_{i,r}$ ($b \times b$ 次) が計算される。ここまでは終了した時点でステップ s で結合される部分回路網どうしで v_i 、 $V_{i,r}$ の交換を行い、 $s-1$ ステップ以前に結合された部分回路網にもデータを送る。受け取ったデータをそれぞれ v_j 、 $V_{j,p}$ とする。(8)式で v_i と v_j の和、(9)式で $V_{i,p}$ と $V_{j,p}$ の和が計算される。(10)式では結合束 p に関するアドミタンス行列 Y_p のLU分解を行っている。(11)、(12)式では結合によって変更された e_k 、 $U_{k,r}$ を修正している。 L ステップにおいて(11)式の計算を終了すると、元の回路網と等しい節点对電圧ベクトル

ルを得ることができる。

3. 計算量と通信量の評価

節点数 n の回路網を m 個に等分割すると、結合束は $m-1$ 個となる。結合束の結合枝数を b として計算量は乗除算の回数で評価することとする。部分回路網における計算量は (4), (5), (6), (7) 式から求められるが、行列 C の要素は 0, または 1 であるから実際は乗除算を行う必要がなく、(6), (7) 式の計算は、無視することができる。 n/m 次元の行列の LU 分解と [部分回路網に隣接した 2 個の結合束の全結合枝数 + 1] 個の解を得るため前進、後退代入で必要とする乗除算は

$$MD1 = (n/m)b(b+1) + 2(n/m-b)b(b+1) \quad (13)$$

で示される。修正段階での計算量は (10), (11), (12) 式における計算量にステップ数 L を掛けた値で決まる。

$$MD2 = ((1/3)b^3 + b^2 + (n/m)b)L + (b^3 + (n/m)b^2)(L-1) \quad (14)$$

各計算機での計算量の和は (13), (14) 式の和である。

次に通信量の評価をする。通信を行うデータは b 次のベクトル v_i , $b \times b$ 次の行列 $V_{i,r}$ であるからデータの個数は $b+b^2$ である。

乗除算 1 回当たりの計算時間を t_{MD} , データ 1 個当たりの通信時間を t_{CO} , 通信回数を $CO(m)$ すると、全処理時間 T は

$$T = (MD1 + MD2)t_{MD} + CO(m)t_{CO} \quad (15)$$

となる。

なお、分割しない場合、すなわち 1 台の計算機で (1) 式を解くことは (4) 式を解くことと等しくなる。すなわち、MSD で $m=1$ とした結果は 1 台の計算機で解く従来の得られる結果と等しい。

4. 結果・考察

図 1 のような節点数 n の $b \times c$ のメッシュ構造を

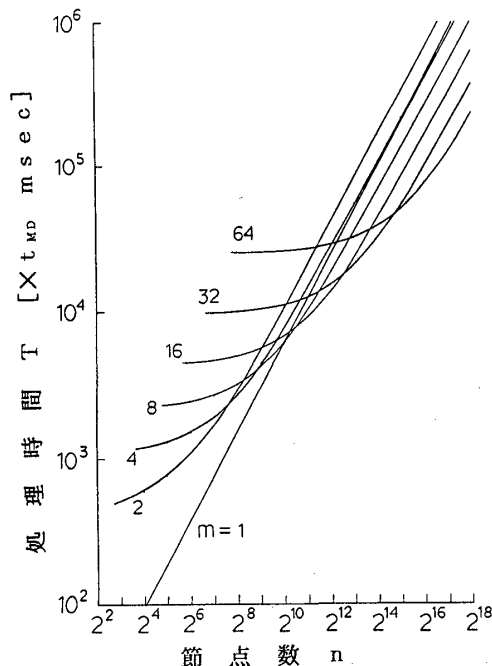


図 2 節点数 n と処理時間 T の計算機台数 m との関係 ($b=2$)

した回路網の節点番号 n の節点から 1 A の電流を流すという条件について、 m 個に等分割し各節点の電位を並列に求める処理時間を (15) 式を用いて計算した。図 2 は結合枝数 b を 2 とし、乗除算 1 回当たりの計算時間 t_{MD} とデータ 1 個当たりの通信時間 t_{CO} との比 $r_{MD/CO} (= t_{MD}/t_{CO})$ を 1.0 としたときの曲線である。 $m=1$ はバンド幅 $(b+1)$ の LU 分解法によって計算した場合の処理時間である。メッシュ構造をした回路網に関しては節点数 n が 2^{10} 以下では $m=1$ の LU 分解法による処理時間が最短となる。しかし、節点数が多くなると、回路網の分割数を多くすると処理時間の短縮がはかれる。図 3 は MSD を C 言語でプログラミングし、実際の並列計算機上で計算を行ったときに節点数 n に対し処理時間が最短となる分割数 m の処理時間を結んだ曲線である。ここでは t_{MD} は約 0.08 msec, t_{CO} は約 0.75 msec となっている。ただし、このプログラムでは実際に行わなくてよい計算 (6), (7) 式の計算を含んでいるため理論曲線との比較を行うと非常に不利な結果となる。

5. むすび

本文で MSD を用いた帯行列の並列計算の有効性を示した。しかし本手法は疎行列の線形並列計算に対して特に有効であると考えられるため、その評価を行う必要がある。最後に CAP 使用を許可された富士通研究所部長石井光雄氏をはじめとするシステム研究部の方々に感謝します。

参考文献

- [1] D.A.Reed, et.al: "Multicomputer Networks: Message-Based Parallel Processing", The MIT Press (1987).
- [2] 石井, 池坂, 石畑: "高並列計算機 CAP", 信学論(D), J71-D-8, 1375-1382 (1988).
- [3] 星野力: "PAX コンピュータ", オーム社 (1985).
- [4] 渡辺, 深尾: "マルチ・ステップ・ダイアコプティックと大規模回路網の並列計算手法", 信学論(A), J70-A-2, 220-227 (1987).

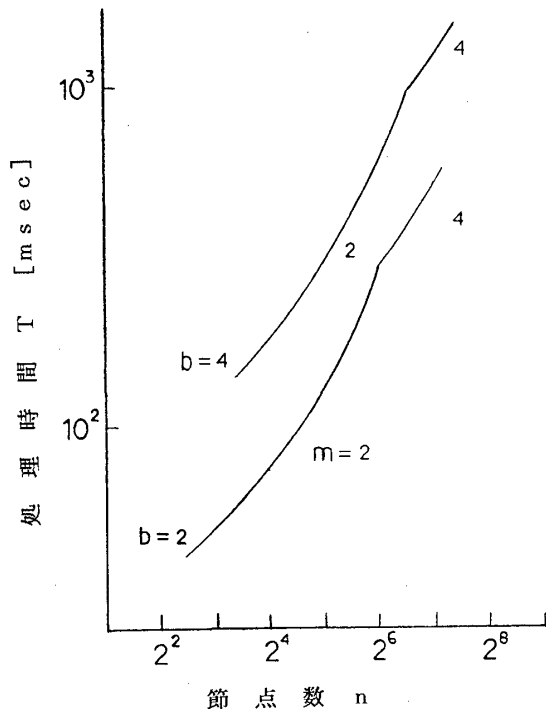


図 3 節点数 n に対する処理時間 T の実測値