

大規模回路シミュレーションに対する完全 LU 分解付 共役残差法とその適用性評価†

山本 富士男^{††} 梅谷 征雄^{††} 高橋 栄^{††}

回路の過渡応答解析においては、大規模な連立一次方程式をくり返し解く必要があるが、その係数行列は不規則非対称な疎行列であるため、従来直接解法が常用されてきた。しかし、対象回路の大規模化にともない、計算速度と必要メモリ量の両面で、これより大幅に高性能な解法が必要とされている。これに対し、本論文では、反復解法の一つである共役残差法と直接解法 (LU 分解法) を結合した方法を試みた。本方式の要点は、①行列の前処理として、従来行われていた不完全 LU 分解を用いず、過渡応答特性を利用して、過去の行列の完全 LU 分解結果を用いる、②解の修正状況パラメータの観測によって、残差が停滞してしまう状態を、従来より早い時期に検出し、無駄な反復計算を削減する、という点にある。本方式を 35~3,668 元行列を有する実回路 10 題の回路シミュレーションに適用した結果、従来直接解法の 1/100~7/100 回の LU 分解と、平均 8~40 回の共役残差法の反復で過渡応答解が得られた。この結果と、演算量の詳しい分析より、本方式は、約 6,000 元以上の回路行列から、計算速度と必要メモリ量の両面で直接解法を凌駕すると推定される。

1. はじめに

VLSI 回路設計においては、回路シミュレーションによって、回路試作前にその動作特性を精密に予測することが必要不可欠となっている。近年の高集積化にともなって回路シミュレーション、そのうちで特に過渡応答解析の対象となる回路規模は著しく増大しており、従来の計算方式では計算時間と必要メモリ量は実用的限界を越える状況になりつつある。

回路の過渡応答解析は、数学的には連立非線形常微分方程式の初期値問題である。従来の代表的な回路シミュレータ SPICE 2¹⁾ などでは、これをまず数値積分公式によって非線形代数方程式に変換し、ついでニュートン・ラプソン法によって線形化し、連立一次方程式に帰着させる。多くの回路素子は非線形特性を有するので、これを微小時間刻みごとに、収束するまでくり返し解く必要がある。このため、計算時間の大部分は、これらの連立一次方程式の求解に費やされ、その傾向は大規模回路で特に著しくなる。

これに対し、最近、回路全体の行列を係数とする連立一次方程式を直接取り扱うことを避け、常微分方程式または非線形代数方程式のレベルで、回路構成に応じて全体を分割して解く方法が研究開発されている²⁾。しかし、これらは回路の種類によって性能および解析精度にばらつきが大きく、回路の最適分割につ

いても研究課題が残されている³⁾。もう一つの別のアプローチは、一般の回路に対して信頼性の高い従来方式での回路全体行列の解法を、ベクトル計算機の利用によって加速するベクトル化方式である。これらは、2,000 トランジスタ規模の回路で、従来方式の全体計算時間を約 1 桁短縮させることが報告されている^{4), 5)}。ところが、これらは直接解法、すなわち LU 分解法にもとづいた方式であるため、その計算量は行列規模の 1.3~1.9 乗で増大し、ベクトル化方式特有のメモリ量の増加率もこれと同程度であることが今後問題となると考えられる。

本論文では、大規模回路で直接解法の性能を上回ることを目指して、回路全体行列を係数とする大規模連立一次方程式を反復系解法で解くことを試みた。回路行列は、不規則非対称性を特徴とする、非常に疎な行列であり、対角優位性や正定値性が無いため、従来、反復解法は不適とされ、ほとんど使用されていない。これに対し、ここでは、回路の過渡応答特性を利用して、反復解法の一つである共役残差法⁶⁾に直接解法の長所を結合させた方式を新たに構成した。以下に本方式の詳細と、大規模回路シミュレーションへの適用結果を示し、これをもとに、今後の超大規模化に対する本方式の有用性を直接解法との比較において明らかにする。

2. 完全 LU 分解付共役残差法

ここでは過渡解析における連立一次方程式 $Ax=b$ の解法として、共役残差法と LU 分解法を結合させた、完全 LU 分解付共役残差法、または略して

† Complete LU Decomposition Conjugate Residual Method and Its Applicability to Large-Scale Circuit Simulation by FUJIO YAMAMOTO, YUKIO UMETANI and SAKAE TAKAHASHI (8-th Department, Central Research Laboratory, Hitachi Ltd.).

†† (株)日立製作所中央研究所第 8 部

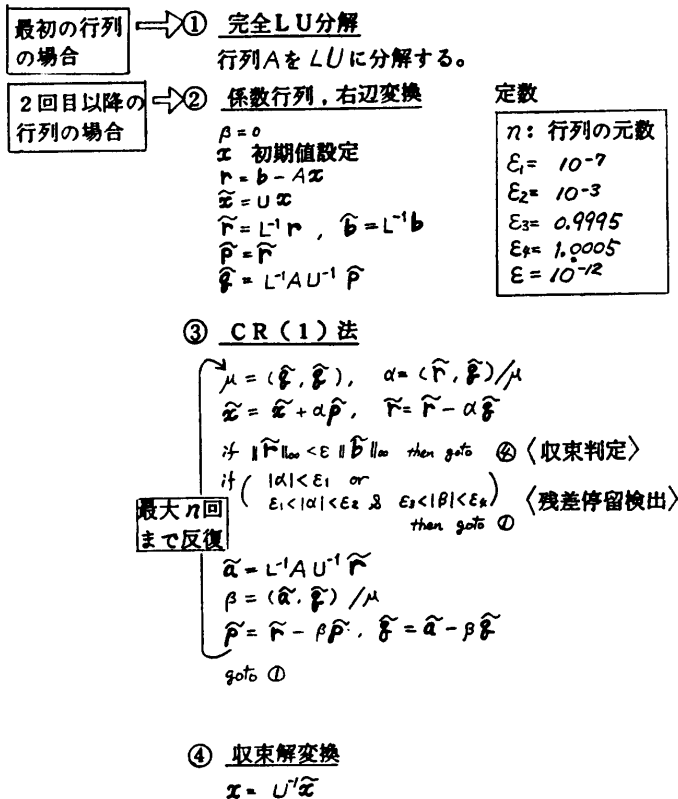


図1 完全LU分解付共役残差法 (CLUCR法) の構成
 Fig. 1 Algorithm of the Complete LU Conjugate Residual Method.

CLUCR法 (Complete LU Conjugate Residual Method) と呼ぶ方法を提案する。係数行列 A は、Modified Nodal 法¹⁾などで作られる回路方程式を離散化および線形化して得られるものであり、 b は電流および電圧を成分とするベクトルである。未知数ベクトル x の成分は電圧および電流である。

2.1 アルゴリズム

CLUCR法のアルゴリズムは図1に示すように、次の四つの部分から成る。

- ①行列 A を完全に LU 分解する。
 - ②この分解結果を使って、 A を単位行列に近づける変換を行う。
 - ③最も簡単な共役残差法である CR(1)⁷⁾ を適用する。
 - ④収束解を、もとの方程式の解に変換する。
- 過渡解析における最初の行列に対しては①から始めるが、2回目以降のすべての行列に対しては、常に②から始める。この時の下三角行列 L と上三角行列 U は、①で分解された最新の結果を指す。②における x の初期値は、最初の行列については直流解析の結果、

2回目以降の行列については最新の x の値を用いることにする。③の CR(1)法では、図中に示した定義による相対残差で収束を判定し、収束の場合は④の処理で求解が完了する。収束していない場合は、残差停留検出部にて、今後の反復の続行で残差減少の可能性があるか否かを、パラメータ α, β によって判定する。残差減少見込みが無い場合は、直ちに反復を打ち切り、①へ戻る。また、反復回数が行列の元数回に達した場合も①へ行く。①で完全LU分解された後も、相対残差を許容値以下に保つため CR(1)法反復を行う。

2.2 アルゴリズムの背景と特徴

CLUCR法を構成する前の予備的実験では、共役残差法収束の十分条件として知られている、 $(A+A^T)/2$ の正定性⁷⁾⁻⁹⁾は、回路行列では保証されないと考えられた。実際、多数の回路の過渡解析において、CR(1)法および CR(2)法は全く収束しなかった。また、フィルインを除く部分に対する不完全LU分解による行列の前処理の付加でも多くの場合収束せず、採用することはできなかった。

さらに、偏微分方程式系の離散化行列に対して有効な、Meijerink-Gustafsson 流の前処理法⁷⁾に相当するものが、回路行列ではその構造の不規則性ゆえ見つからない。

そこで CLUCR法の構成においては、以下の2点に工夫を凝らした。第1点は、行列の前処理として、その時点の行列の不完全LU分解を用いるのではなく、過去において最も新しく完全LU分解された行列の L と U を用いることである。その根拠は、過渡応答解析が通常十分小さな時間刻みで行われ、しかも、各時点において動作状態が変化しつつあるデバイスの個数の割合は非常に少ない²⁾ことに基づいている。つまり、行列の性質は比較的滑らかに変化する場合が多く、一度完全LU分解された結果はしばらくの間、後続の行列の前処理に有効に使えようと考えた。

なお、共役残差法の前処理として、間欠的に不完全LU分解を用いる方法¹⁰⁾もあるが、回路行列の場合には、その特性上、完全LU分解(不完全ではなく)の使用が必須であった。

第2点は、通常の収束判定の後に行う、 α と β による残差の停留状態の検出である。これは、反復部が収

束せず完全 LU 分解を起動する場合の、不要な反復を早期に打ち切るために行う。残差ノルム自体は、反復の途中で一時的に停留する可能性があることを観測済であったので、このような検出法をとった。図1に示した残差停留検出の判定条件の意味は以下のとおりである。

残差停留とみなす条件 (次のいずれか) :

(i) $|\alpha|$ が極めて小さい。

この場合は、経験的に解の修正活動が停止に向うと言える。

(ii) $|\beta| \approx 1$ かつ $|\alpha|$ はあまり大きくない。

この場合は、直前の解の修正量が多くなく、しかも、 $\beta = (\tilde{A}\tilde{r}, \tilde{A}\tilde{p}) / (\tilde{A}\tilde{p}, \tilde{A}\tilde{p})$ であることから、修正方向 \tilde{p} は残差 \tilde{r} と一致している可能性が高く、残差減少の見込みは無い。

なお、残差停留判定基準として用いる、各種定数を図1右上部に示したが、これらは実験にもとづいて経験的に決めたものである。

3. 残差減少状況の考察

本章では、CLUCR 法を大規模回路に適用するに先立ち、一つの小規模回路 (96 元行列を持つ) を例題に、その収束特性を示し、前章での残差停留検出法の妥当性を確認しておく。

この回路の過渡応答解析では約 200 回、CLUCR 法による行列演算を実行したが、そのうちから無作為に 12 ケースをとり出し、その収束状況を示したのが図2である。ただし、この図では実験のため、残差停留検出は行わず、常に元の個数回まで反復を許した。この場合、収束しない多くのケースでは、反復の初期においてすでに残差の減少が停止しており、無駄な反復が行われることがわかる。

次にこの回路について、パラメータ α, β の変化を、高速に収束する場合 (図3)、収束するが低速な場合 (図4)、および収束しない場合 (図5) について示した。図3では、 $|\alpha|$ は 1~5 という、かなり大き

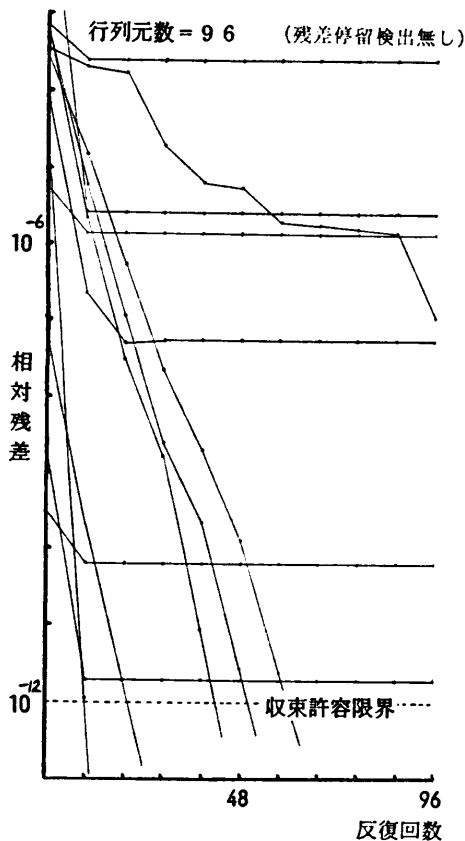


図2 CLUCR 法の残差減少パターン例
Fig. 2 Examples of the state of residual reduction for a 96 equations circuit.

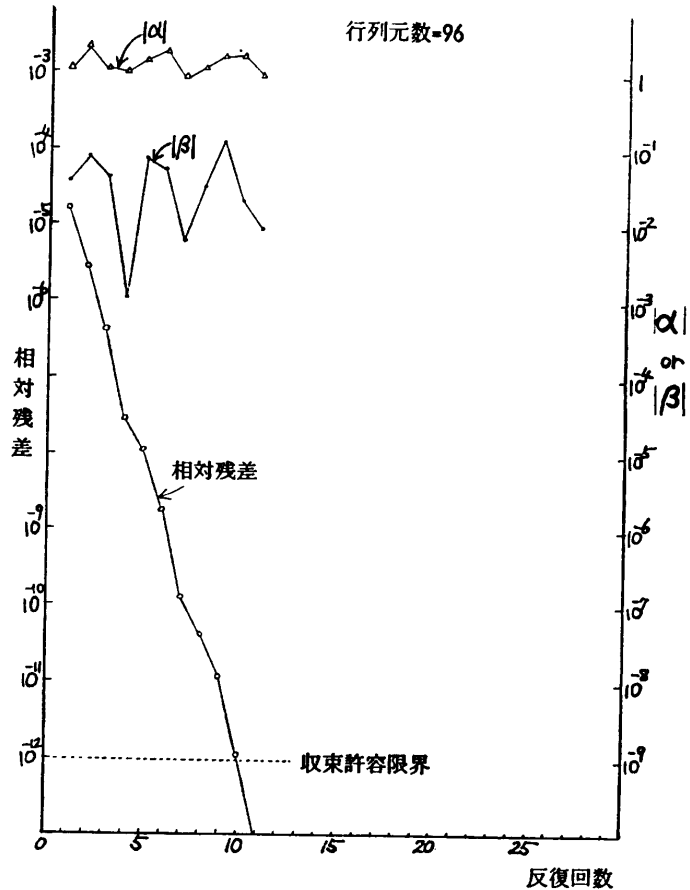


図3 CLUCR 法が高速に収束する例
Fig. 3 An example of rapid convergence in CLUCR method.

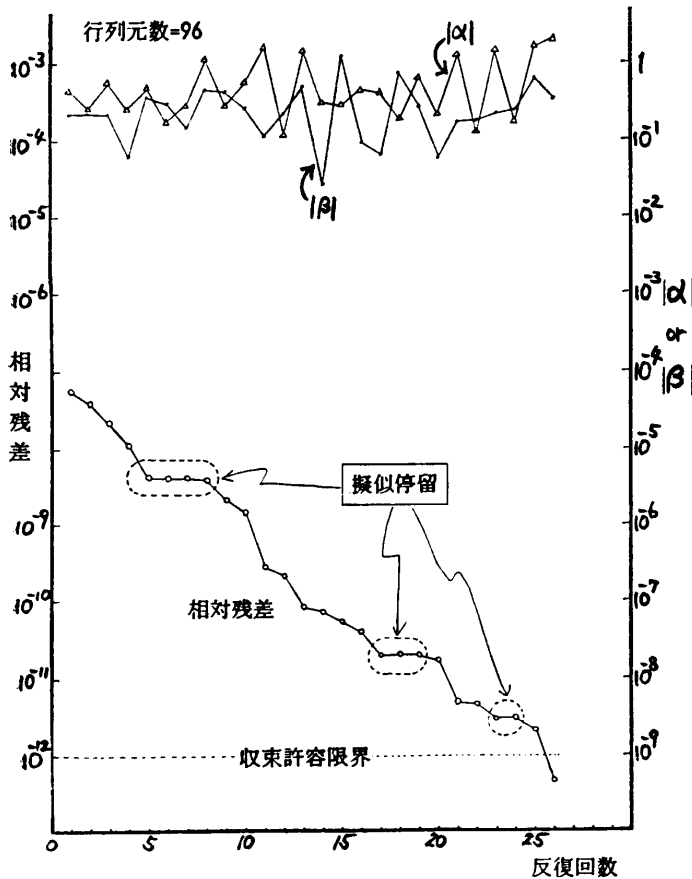


図 4 CLUCR 法の収束が低速な例
Fig. 4 An example of slow convergence in CLUCR method.

な動きを示し、 $|\beta|$ も 1 には近くない。図 4 では、 $|\alpha|$ は 1 より小さい場合が多く、 $|\beta|$ も 1 に接近しており収束速度が鈍ってきている。また、途中に残差の一時的停留（擬似停留）があるが、今回の残差停留検出法ではここをパスして反復を続行させ、最終的に収束解が得られた。これに対し、収束しない図 5 では、 $|\beta|$ が 1 に極めて接近し、かつ $|\alpha|$ が急速に小さくなる点で残差停留と判定され、反復が打ち切られて完全 LU 分解の起動が生じた。

以上により、残差停留検出法が都合良く働いていることが確認された。

4. 大規模回路への適用結果

本章では、35~3,668 元の行列を持つ現実の回路 10 題の過渡応答解析に CLUCR 法を適用した結果を検討する。

4.1 完全 LU 分解起動率と反復回数

ここでは、CLUCR 法において、完全 LU 分解が必

要な度合（完全 LU 分解起動率）と CR(1) 法の反復回数を分析する。各回路ごとにシミュレーションの時間区間、およびニュートン・ラプソン反復の総回数 N_{iter} は各々異なるので、完全 LU 分解起動率は、完全 LU 分解の実行回数を N_{iter} で割った値と定義する。

10 題のうち、入力信号の切り換えが特に激しい 1 題を除き、完全 LU 分解起動率は 0.01~0.07 (図 6)、CR(1) 法の平均反復回数は 8~40 回 (図 7) でいずれも相対残差 10^{-12} の収束解を得た。これらの図では、完全 LU 分解起動率および CR(1) 法平均反復回数とも、行列元数の増大に対して著しい増加は示しておらず、かなり高い収束性が得られていると言える。このように、少数回の完全 LU 分解による前処理が、広範囲の回路行列に対する CR(1) 法の収束加速に極めて有効であることが確認された。

第 2 章で述べた残差停留検出を採用した場合としない場合で、図 6 に示した完全 LU 分解起動率は、全回路とも同一の結果であった。仮に、収束する反復を、残差停留検出によって不当に中断する場合があるとすれば、完全 LU 分解起動率が両

者で一致する確率は非常に小さいはずである。このことから、全回路に対し、残差停留検出法は不都合な作用を与えていないと言える。CR(1) 法の平均反復回数については、この残差停留検出法は、無条件に元の個数回まで反復を許容する場合に比べ、これをほぼ半減させることが認められた (図 7)。

4.2 浮動小数点演算回数の分析

以上で、現実の回路行列に対する CLUCR 法の収束特性が明らかになったので、次にその演算量の分析を行う。図 8 はその結果である。図中、「完全 LU 分解」とは、図 1 ①部分の実行 1 回分、「前後処理」とは図 1 の②と④の部分の実行 1 回分、「CR 反復 1 回」とは図 1 ③の CR(1) 法反復 1 回分を指し、各々に要する浮動小数点演算回数を示した。回路によってばらつきはあるが、特に小規模回路では、CR(1) 法反復 1 回の演算量は、完全 LU 分解より多い傾向がある。これは、回路行列が非常に疎であり、小規模回路ではリオーダーングの効果でフィルイン発生も少ないことか

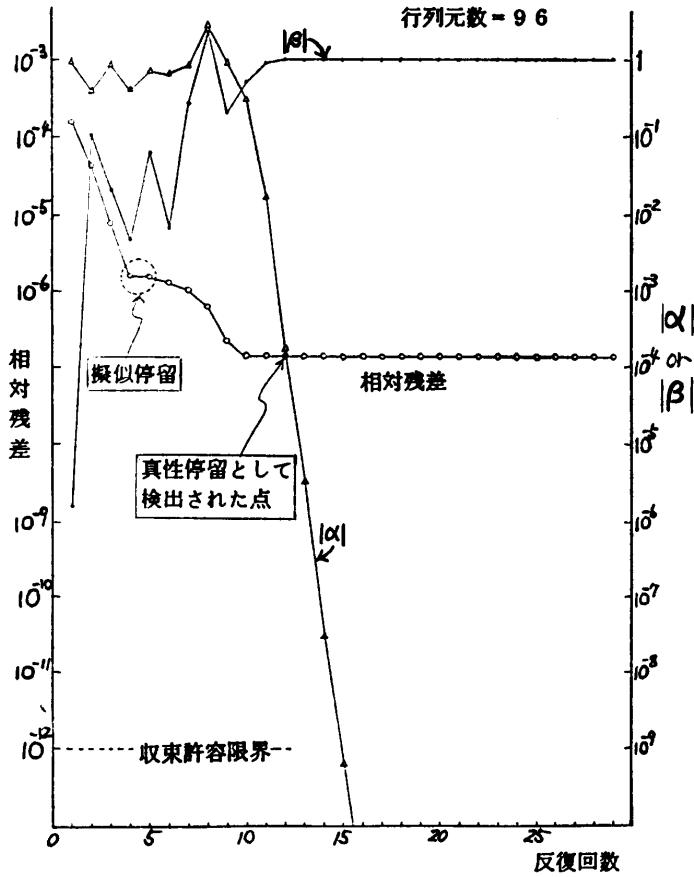


図 5 CLUCR 法が収束しない例
Fig. 5 An example of non-convergence in CLUCR method.

ら、完全 LU 分解演算量はわずかなのに対し、CR(1) 法内の行列乗算 $L^{-1}AU^{-1}$ などの演算量が相対的に多いためと考えられる。また、「前後処理」の演算量

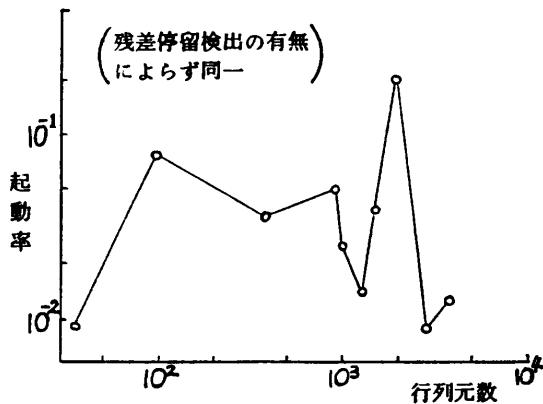


図 6 CLUCR 法の完全 LU 分解起動率
Fig. 6 Rate of complete LU decomposition activations in CLUCR method for 10 actual LSI circuits.

が多いのもこれと同様の理由による。

しかしながら、行列元数の増加に対する演算量の増加率は、「CR 反復 1 回」の方が、「完全 LU 分解」よりかなり低いことに注目すべきである。両者の差は、規模の増大で完全 LU 分解演算量が増えるとともに縮まると考えられる。また、回路の種類によっては、すでに、これら二つの演算量の多少が逆転しているものも見られる。図 8 中に示した 3 本の直線は、上記 3 種の演算量について、さらに大規模化した場合の演算量の上限を予測するために引いてある。

5. 直接解法との性能比較

5.1 演算量

CLUCR 法と直接解法 (LU 分解法) の性能を、行列演算 1 回当たりの浮動小数点演算回数で比較したのが図 9 である。この図では主として、規模の増大にともなう演算量の増加傾向を推定するのが目的なので、CLUCR 法では、第 4 章で得た各回路の収束性を正規化して予測線を引いた。すなわち、完全 LU 分解起動率と CR(1) 法平均反復回数に関して、図 6 と図 7 の実測値から、やや楽観的な Case-1 と悲観的な

Case-2 を設定し、各々について完全 LU 分解起動率は 0.03 と 0.05、CR(1) 法平均反復回数は 10 回と 30 回とした。これと、図 8 に示した演算量分析結果から、行列演算 1 回当たりの浮動小数点演算回数を各回路について計算し、図 9 にプロットした。2 組ずつの破

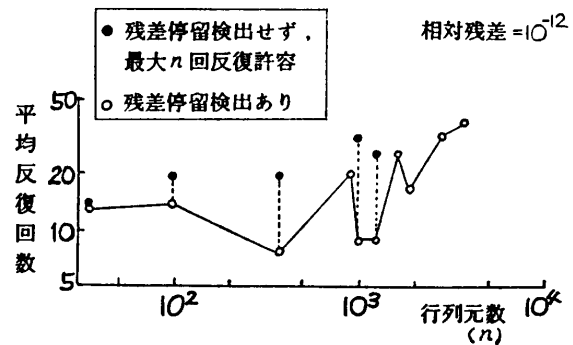


図 7 CLUCR 法の平均反復回数
Fig. 7 Average counts of iterations in CLUCR method for 10 actual LSI circuits.

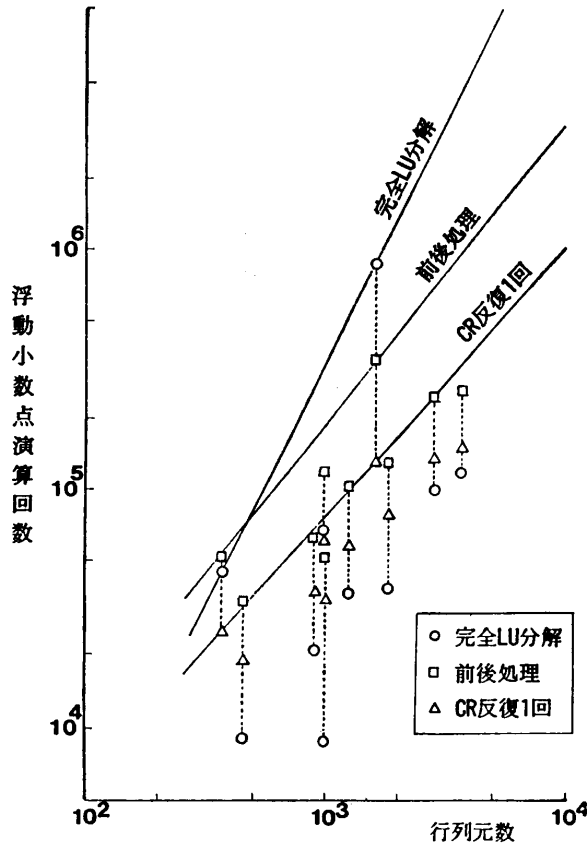


図 8 CLUCR 法の浮動小数点演算の内訳
Fig. 8 Analysis of the number of floating point operations in CLUCR method.

線は、両ケースについての、回路によるばらつきの上
限と下限の推定線である。直接解法については、実測
値をそのままプロットし、上限、下限の推定線を実線
で示した。

この図から、直接解法の演算量は、小規模では
CLUCR 法より少ないが、規模の増加とともにその傾
向は逆転することがわかる。これは、4.2 節で述べた
ように、CR(1) 法反復 1 回当りの演算量が、完全 LU
分解演算量を次第に下回るからにはかならない。上限
の予測線どうしで比較すると、CLUCR 法は、Case-1
で約 6,000 元、Case-2 で約 20,000 元以上の行列か
ら、直接解法の演算量を下回るようになると推定でき
る。

5.2 記憶容量

次に主記憶容量の比較を図 10 に示す。ここには、
行列演算に必要な主記憶容量のみを図示した。直接解
法としては、ベクトル化 LU 分解法である BVA 法⁴⁾
を用いた。CLUCR 法でも、完全 LU 分解を高速に実

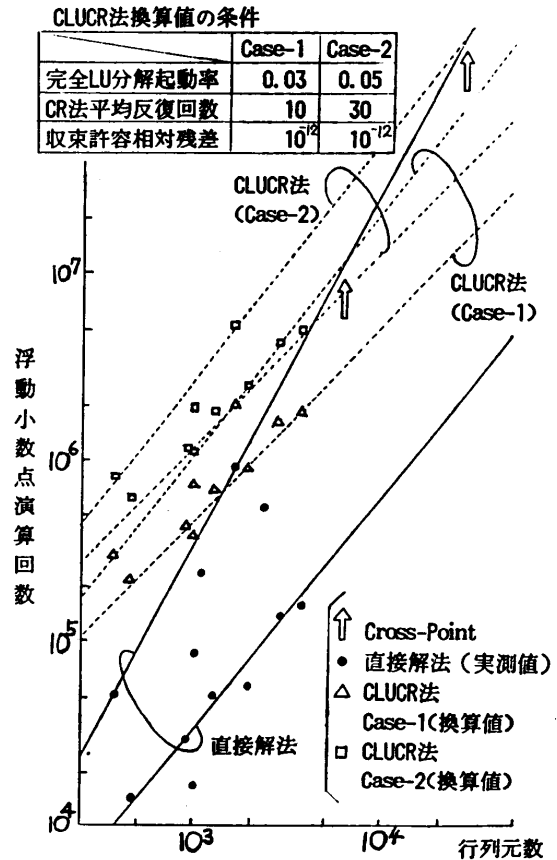


図 9 行列演算量の比較
Fig. 9 Comparison of the number of floating point operations in solving a system of linear equations.

行するには BVA 法の使用が必須であるが、これま
でに述べたようにその起動率が極めて低いことから、
BVA 法固有の大量のデータは、ベクトル計算機 S-
810 の拡張記憶装置¹¹⁾などの高速外部記憶に置くもの
とした。

以上の条件のもとでは、CLUCR 法の主記憶容量は
直接解法の場合よりかなり少なく、規模の増大に対す
る増加率も著しく少ない。たとえば、10,000 元行列に
対する、シミュレーション全体に必要な主記憶容量の
総量（行列演算用以外に、デバイスモデルパラメータ
領域などを加えた）を、図中の括弧内に示したが、直
接解法 93 MB、CLUCR 法 33 MB と推定される。こ
こで、拡張記憶を使うことによる USE 時間の増加が
問題となるが、次節で述べるようにこれは非常にわず
かであると考えられる。

5.3 1 万トランジスタ回路に対する推定性能

ここまでの検討結果の結論として、今後出現が予想

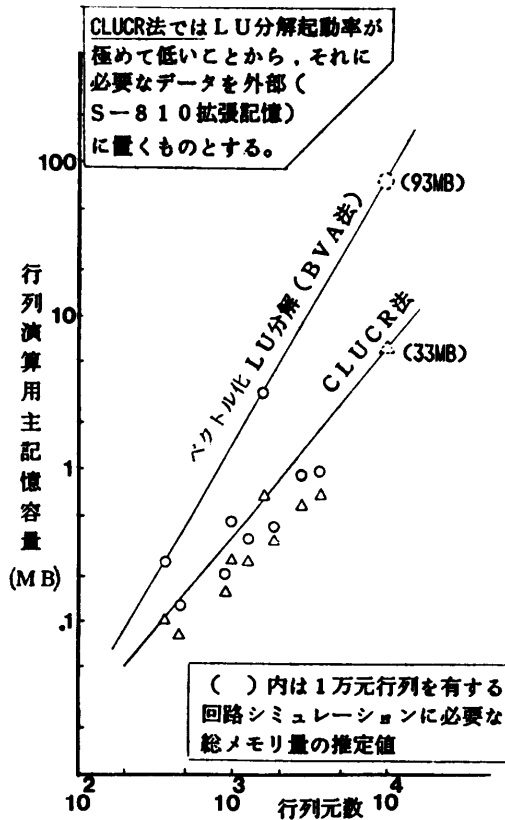


図 10 行列演算用主記憶容量の比較
Fig. 10 Comparison of main memory amount needed for matrix solution.

される 10,000 トランジスタ (MOST) 規模の回路シミュレーションに対する, CLUCR 法の性能予測を表 1 に示す。この場合, 過去のデータから, 回路行列は 10,000 元, 計算される時間点数は 800 点とした。表中の計算時間, 記憶容量はすべて, シミュレーション全体に要する推定値である。方式 A は, ベクトル化方式による直接解法であり, この性能は過去のデータより外挿した。方式 B~E はいずれも CLUCR 法だが, 完全 LU 分解の実行方法が異なる。CLUCR 法はほぼ全体をベクトル化することが可能なので, この時の CPU 時間は, 方式 A の CPU 時間と, 図 9 の演算量の比率 (直接解法対 Case-1 の) から得た。方式 D と E では, 外部記憶とのバッファを主記憶に 1 MB 用意し, ベクトル化完全 LU 分解に必要なデータ 65 MB を, 図 9 Case-1 で定めた頻度で取り込むものとした。USE 時間の増加は, Disk 使用の方式 D では実験的に, S-810 拡張記憶使用の方式 E では文献 10) のデータから推定した。

この結果, 拡張記憶使用の CLUCR 法 E は, すべて主記憶を用いる直接解法 A に対し, USE 時間を増加させることなく, 約 37% 程度の主記憶容量で実行できることがわかった。CPU 時間の減少は 18% 程度にすぎないが, これよりさらに大規模な回路ではもっと大幅なものになるであろう。

6. 他の反復解法との比較

最後に, 共役残差法以外の反復解法の, 回路行列へ

表 1 10,000 MOST* に対する性能推定値 (on S810)
Table 1 Comparison of the predicted performance for a circuit with 10,000 MOS transistors.

方式	諸元	CPU Time (HR)	USE Time (HR)	主記憶容量 (MB)	Disk 容量	ES** 容量	備考
A	ベクトル化方式 LU 分解	1.6	1.9	93	—	—	デバイスモデルはベクトル化
B	スカラー LU 分解の CLUCR	97	116	33	—	—	"
C	ベクトル化方式 LU 分解の CLUCR (すべて主記憶)	1.3	1.6	98	—	—	"
D	ベクトル化方式 LU 分解の CLUCR (with Disk)	1.3	9.8	34	65MB	—	"
E	ベクトル化方式 LU 分解の CLUCR (with ES)	1.3	1.7	34	—	65MB	"

* 行列は 1 万円とする。
** S810 拡張記憶装置

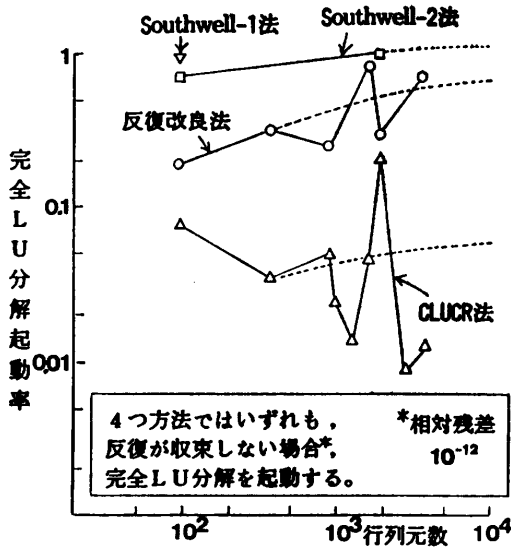


図 11 他の反復解法との比較

Fig. 11 Comparison of the rate of complete LU decomposition activations with other iterative methods.

の適用性について簡単に述べる。理論的に適用可能なものとして以下の五つを検討した。

- (a) BCG 法 (Biconjugate Gradient Method)¹²⁾
- (b) 非対称最急降下法¹³⁾
- (c) Gauss-Seidel 系解法
- (d) 反復改良法¹⁴⁾
- (e) Southwell 法¹⁵⁾

このうち、前処理付の(a)と(b)は、CLUCR 法より収束性は良いと考えられたが、アルゴリズム上、演算量が約 2 倍となるため採用しなかった。CLUCR 法の収束性は極めて高かったため、これをさらに半減させるには至らないと考えたためである。(c)については収束性が悪く、適当な加速法が無かった。残りの(d)と(e)については、CLUCR 法と同様に完全 LU 分解を結合させた方式を構成し、実験してみた。その結果の一部として、完全 LU 分解起動率を図 11 に示す。CLUCR 法の場合と異なり、これらの完全 LU 分解起動率はかなり高く、しかも、規模の増大とともにその値は 1 に極めて接近する。したがって、(d)、(e)の最大の利点である反復計算の単純さを生かすことができない。

以上のことから、回路行列に対する反復系解法としては、CLUCR 法が最も高性能である。

7. まとめと今後の課題

不規則非対称性疎行列を係数とする、回路の過渡応

答解析での連立一次方程式に対し、共役残差法と LU 分解法を結合させた解法を提案した。

- (1) 35~3,668 元行列を有する実回路 10 題にこれを適用した結果、LU 分解の必要回数は、100 回の行列演算当り 1~7 回、共役残差法の平均反復回数 8~40 回で、相対残差 10⁻¹² の収束解を得た。
- (2) この結果と演算量の分析から、約 6,000 元以上の回路行列で、本方式は直接解法の性能を上回るといふ推定を得た。
- (3) 本方式は、大規模回路において、高速な外部記憶が使えるなら、USE 時間をほとんど増加させることなく、直接解法と比べ、主記憶容量を大幅に減少させることができる。

回路の構成および過渡応答における信号伝搬にともなう行列特性の変化と、共役残差法の非収束現象との因果関係の解明は今後の課題である。

謝辞 共役残差法を含む多数の反復解法に対する前処理法およびそれらの性能特性について詳しく御教示いただいた、村田健郎図書館情報大学教授に感謝の意を表す。

参 考 文 献

- 1) Nagel, L. W.: SPICE 2—A Computer Program to Simulate Semiconductor Circuits, Memo No. ERL-M 520, Univ. of California, Berkeley, pp. 1-233 (1975).
- 2) Newton, A. R. and Sangiovanni-Vincentelli, A.: Relaxation-Based Electrical Simulation, *IEEE Trans. CAD*, Vol. CAD-3, No. 4, pp. 308-331 (1984).
- 3) White, J. and Sangiovanni-Vincentelli, A.: Partitioning Algorithms and Parallel Implementations of Wave Form Relaxation Algorithms for Circuit Simulation, *Proc. ISCAS '85*, pp. 221-224 (1985).
- 4) Yamamoto, F. and Takahashi, S.: Vectorized LU Decomposition Algorithms for Large-Scale Circuit Simulation, *IEEE Trans. CAD*, Vol. CAD-4, No. 3, pp. 232-239 (1985).
- 5) Vladimirescu, A. and Pederson, D. O.: Circuit Simulation on Vector Processors, *Proc. ICCV '82*, pp. 172-175 (1982).
- 6) 村田健郎, 小国 力, 唐木幸比古: スーパーコンピュータ—科学技術計算への適用—, pp. 136-153, 丸善, 東京 (1985).
- 7) 村田健郎: 科学技術計算と高速算法, *Computer Today*, No. 2, pp. 51-61 (1984).
- 8) Eisenstat, S. C.: A Note on the Generalized

- Conjugate Gradient Method, *SIAM J. Num. Anal.*, Vol. 20, No. 2, pp. 358-361 (1983).
- 9) Eisenstat, S. C., Elman, H. C. and Schultz, M. H.: Variational Iterative Methods for Non-symmetric Systems of Linear Equations, *SIAM J. Num. Anal.*, Vol. 20, No. 2, pp. 345-357 (1983).
- 10) 唐木幸比古: スーパーコンピュータ S-810 の応用性能, 情報処理学会数値解析研究会資料, 12-2 (1985).
- 11) 堀越 彌, 長島重夫: スーパーコンピュータ・アーキテクチャの動向, *Computer Today*, No. 2, pp. 43-50 (1984).
- 12) Saad, Y.: The Lanczos Biorthogonalization Algorithm and Other Oblique Projection Methods for Solving Large Unsymmetric Systems, *SIAM J. Num. Anal.*, Vol. 19, No. 3, pp. 485-506 (1982).
- 13) 戸川隼人: マトリックスの数値計算, p. 323, オーム社, 東京 (1971).
- 14) フォーサイス・モウラー (渋谷・田辺共訳): 線形計算の基礎, pp. 58-142, 培風館, 東京 (1969).
- 15) 赤坂 隆: 数値計算, pp. 247-265, コロナ社, 東京 (1985).
- 16) 須賀伸介, 南 知行, 森 正武: ステファン問題の有限要素解析に現われる非対称行列に対する PCR 法, 情報処理学会論文誌, Vol. 26, No. 6, pp. 1041-1048 (1985).

(昭和 61 年 3 月 14 日受付)

(昭和 61 年 5 月 15 日採録)



山本富士男 (正会員)

昭和 45 年, 北海道大学理学部数学科卒業. 同年, (株)日立製作所入社. 以来, 同社中央研究所にて, 計算機言語, VLSI 回路シミュレーション, 並列処理ソフトウェアなどの研究開発に従事し現在に至る.



梅谷 征雄 (正会員)

昭和 19 年 (1944) 生. 昭和 43 年東京大学理学部数学科卒業. 同年 (株)日立製作所に入社. 中央研究所にて, デジタル装置の故障診断, 並列処理アーキテクチャ, 並列化コンパイラ, 並列記述言語, 並列アルゴリズムの研究に従事し今日に至る.



高橋 栄 (正会員)

昭和 17 年生. 昭和 40 年東京大学工学部航空学科卒業, 45 年同大学院博士課程修了. 同年より (株)日立製作所中央研究所に勤務し, コンピュータ・グラフィックス, オペレーティング・システム, 数値シミュレーション等の研究開発に従事. 現在, 同社ソフトウェア工場主任技師. ACM, IEEE-CS 各会員.