ガウスの消去法における非零化要素数最小化手法の提案

京都大学工学部　正　員　白石成人
京都大学工学部　正　員　谷口健男
新日本製鐵（株）　正　員○酒井　誠

1)まえがき　構造解析を行うに際し，連立一次方程式の解法はその中で大半の時間を占める場合があり，また得られた解の信頼性などの諸点からそれは重要なプロセスである。そのため今日までに電算機の特性にあった多くの手法が提案されている。解法の分類に関しては直接解法，間接解法とに大別でき，一般に直接解法の方がより経済的であり，かつある大きさまでの構造物に対してはより精度の高い解が得られることがよく知られている。その直接解法の基本となるのがガウスの消去法である。消去法を実行すると係数行列は変換され，その際に最初は零であった要素が非零要素になる場合が出てくる。これをfill-inと呼び，このfill-inと最初からの非零要素のみを対象にして消去法を行おうとする手法がスパースマトリックス法である。これは数ある直接解法の中で最も演算回数の少なくてすむ手法である。本報文はこのスパースマトリックス法を有効ならしめるfill-in最小化手法の提案を行う。

2)fill-in最小化手法について　係数行列のk行目の消去過程で発生するfill-in数はグラフ的には節点(k)に隣接していた節点を完全グラフ化するために必要な辺数にあたることはよく知られている。従来よりこのfill-in最小化手法の開発がなされてきているが，主なものとしてMinimum Degree Algorithm (Min.Deg.), Minimum Deficiency Algorithm (Min.Def.)が挙げられる。前者は各消去過程で最小次数の節点から，また後者は最小Deficiencyを有する節点から消去する手法である。ここでDeficiencyとはある節点を消去することで発生するfill-in数と考えてよい。節点(i)の消去により発生するfill-in数は次式となる。

$$|\text{fill-in}(i)| \propto \{\deg.(i)\}^2 - m(i) \quad \left[\begin{array}{l}\deg(i)\text{; 節点}(i)\text{の次数} \\ m(i)\text{; 節点}(i)\text{とそれに隣接する節点のSubgraph内の辺数}\end{array}\right] \quad \text{―①}$$

Min.Deg.では右辺第1項に注目し，Min.Def.では全項に注目して消去順序を与えている。この2手法は一般にグラフの巾と呼ばれるものが小さなグラフに対しては有効であり，Min.Def.はMin.Deg.より少し巾の大きなグラフに対しても有効となる。しかし両者とも各ステップだけに注目し，それ以後のステップを考慮に入れないために巾の大きなグラフに対しては最小値を与えることができなかった。つまり従来のこの2種類の手法をベースにした手法は局所的最小値を与えるにとどまっている。

ところである節点を消去した時，その隣接していた節点は部分完全グラフになることは先に述べたが，それを今Frontal Vertex Group (FVG)と呼ぶことにする。FVGに含まれる節点の消去にあたり発生するfill-in数はその節点に隣接しているFVG以外の消去されていない節点数$|v_k|$と$|\mathrm{FVG}|$に支配され次式のようになる。

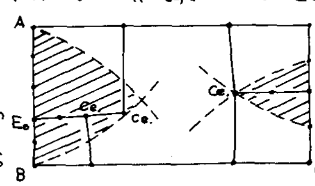
$$|\text{fill-in}(k)| \propto |\mathrm{FVG}| \times |v_k|$$

そしてv_kが新しくFVGに加えられ消去された節点が除外される。またFVG内以外の節点を消去

して異なった個所でFVGを発生させてもFUG内の節点はいづれ消去せねばならず結局|FVG|の大きさがfill-in数に影響を与えてくる。一般の構造物においては|Vk|に大差はなく、fill-in最小化手法は|FVG|に注目せねばならないことがわかる。

3) Subgraph単位によるfill-in最小化手法の提案　先に述べたとおりfill-in最小化手法は|FUG|の大きさに注目せねばならない。一般にある節点から距離1の節点群の大きさの増加が小さい場合のその出発となる節点のことをCorner Vertexと呼ぶ。2つ以上のFVGに含まれる節点を消去する場合は、その節点消去に伴い1つのFVGとなり、その大きさがそれ以前のFVGのそれぞれの大きさと大差があってはならない。何故ならその節点消去に伴い局所的にもfill-in数がそれぞれの|FVG|の乗算に準じ、他の場合に比べ大きくなるし、以後のステップでも|FVG|が大きくなるためである。本手法は2つ以上のFVGに含まれる節点で、かつその節点消去で1つになるFVGが前段階のそれぞれの|FVG|と比べ大差ないような節点グループでオリジナルグラフに境界を与え、そのSubgraph単位でfill-in最小化手法を行うものである。以下境界節点の選択手法について述べる。Corner Vertexから距離1の節点群(各節点群の消去後のFVGにあたる)を順次消去すれば|FVG|を小さく保ちうることは前述より明らかである。FVG内の節点(k)消去する時の|FVG|の増減を支配する値(α)は次式である。

$$\Delta|FVG| \propto \alpha = |Deficiency(k)| \,/\, |V_k| \qquad ――②$$

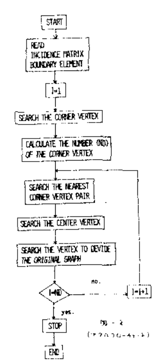


図－(1) 境界節点を求める手順を示す図

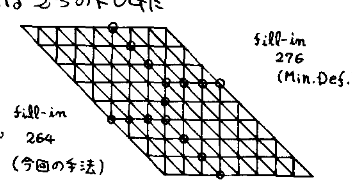
今Corner Vertexが図-1のA,B,C,D,にある場合について述べれば、図の斜線部分ではαの値は2つ与えられている。その2つのαの小さな値(例えばA方向からの値α(A)とすれば)をA方向からの一連の消去順序に組入れれば|FUG|は常に最小に保てる。境界節点はA-B斜線部分についていえば必ずA-Bの境界上の節点に存在するからまずその節点を次のように決定する。

$$\alpha_E(X) = \max_{i \to A\text{-}B} \left(\min \left(\alpha_i(A), \alpha_i(B) \right) \right) \qquad ――③$$

この場合の節点(E)が境界節点となる。節点(E)から順次距離1の節点群に対して式③を用いて境界節点を求めればよい。また求めるべき個数はA-E,B-Eの境界上の辺数に等しい。その最後の境界節点をCenter Vertex(Ce.)と呼ぶ。Ce.(A)からはA-D上の境界上の節点への最短パス、Ce.(B)からはB-C境界上の節点への最短パスを構成する節点群を境界節点とすればよい。最短パスをとる理由は境界節点内のSubgraphで|FUG|が大きくならないための配慮であり、図-(3)のような形状に対しては必要でない。この結果図-(1)では5つのSubgraphに分けられたことになる。各Corner Vertexを含む4つのSubgraph内で順次消去を行い、それぞれに終了すれば2つのFVGに含まれるE_0, E_1,を消去して1つのFVGとし、そのFVGから距離1の節点群を順次消去すればfill-in最小化が図れる。ここまでの境界節点を求めるアルゴリズムのマクロフローチャートを図-(2)に示し、その適用例を図-(3)に示す(○印の節点が境界節点にあたる)



図－2 (マクロフローチャート)



図－(3) 例題

4) あとがき　本報文ではfill-inの発生機構を考察した結果節点消去により発生するfill-in数は|FVG|という一種の節点グループの大きさに依存することがわかった。それを手段にしてオリジナルグラフを幾つかのSubgraphに分けてそれぞれについてある規則で消去を行なえば従来の手法と比べより全体的なfill-in最小化を図ることがわかり、本報文ではSubgraph単位のfill-in最小化手法の提案ができた。しかしながら一般的形状に対し|FVG|だけでfill-in最小化を進めることが必ずしも出来るとは言えず他の因子のそれへの導入が今後の課題として残される。

(参考文献) D.J.Rose; A Graph-Theoretic Study of The Numerical Solution of Sparse Positive Definite Systems of Linear Equation, Graph Theory and Computing (Ed. R.C. Read), Academic Press, 1972, 183-217