ハイパフォーマンス
コンピューティング 64−4
（1996．12．12）

LU 分解による疎行列の最小自乗最小ノルム解法について

須田 礼仁, 小柳 義夫

東京大学 理学部 情報科学科

最小自乗最小ノルム解を求める解法としては特異値分解や QR 分解がよく用いられるが、LU 分解を用いて解を得る方法も存在する。本論文では LU 分解による最小自乗最小ノルム解法を紹介し、LU 分解と QR 分解における誤差と速度について評価をおこなう。誤差は LU 分解の方が大きいと考えられるが、QR 分解も決して安定というわけではない。特に疎行列では LU 分解の方が絶対的に速く、誤差も fill-in を抑えればそれほど悪くないので LU 分解による方法が有利な場合も十分にあると考えられる。

On a Method with the LU decomposition for Pseudoinverse Solutions of Sparse Metrices

Reiji Suda and Yoshio Oyanagi

Department of Information Science, Faculty of Science, the University of Tokyo

Pseudoinverse solutions can be obtained using the LU decomposition, though the Singular Value Decomposition and the QR decomposition are widely used. This paper introduces a Pseudoinverse solver by the LU decomposition, and evaluate the error and the speed of the methods by the LU and the QR decompositions. The LU decomposition suffers a larger error, but the QR decomposition is by no means absolutely stable. The LU decomposition will be better in some cases, especially for sparse matrices, because the LU decomposition is faster than the QR method, and the error may become tolerable because of the smaller number of fill-in elements.

§1 はじめに

連立非線形方程式の解法は Newton-Raphson 法と相場が決まっている。しかし、Newton 法もそのまま単純に適用して何でも解けるという万能薬ではないことは当然である。実際に使ってみるとさまざまな困難が待ち受けている。例えば初期値が悪いと解に到達しない、解への接近が極めて遅い、不安定な挙動に陥るなどである。

本研究の出発点は、Newton 法において Jacobian のランクが落ちる、あるいは極めて条件が悪くなるような状況に遭遇したことにある。これはそもそもの連立非線形方程式がランク落ちしている場合もあるし、解に向かう途中においてランクの落ちる点を通過する場合もあるし、また解でランクが落ちる場合もある（重根の場合には解でランクが落ちるが、これは一次収束は避けられない)。また、ランクが落ちていなくても最小特異値が極めて小さい場合には条件が悪くなって Jacobian を係数とする連立一次方程式の解が不安定になる。このような場合には正確な解のノルムは大きくなりがちであるが、Newton 法は本来局所的な情報を用いて解に近づこうとする解法であるから、ノルムの大きな修正ベクトルはよくない。むしろ多少の残差を許しても、局所的な情報を活かすことのできるノルムの小さな近似解の方が Newton 法にとっては有利である。

このような状況では、Jacobian を係数とする連立一次方程式の解として最小自乗最小ノルム解（またはその近似解）を用いるのが望ましい。最小自乗最小ノルム問題の解法としては、特異値分解 (SVD) を用いるのが一般的である。これは最も安定であるが、固有値問題のように反復解法によらなければならず、計算量も一般にはかなり大きい。この計算量が高速性が要求される場合に問題となる。これに比べてやや安定性が劣るものの、直接法であって速度もそれなりに速い方法に QR 分解がある。この場合には「落ちかかった」ランクを正確に評価することは難しい（ランク顕示 QR 分解 [1: p. 108] が必要である）ものの、かなり有用な情報を得ることができる。しかし、大規模疎行列を取り扱いたい場合には fill-in の多く発生する QR 分解ですら時間がかかり過ぎる場合がある。そこで本研究では fill-in を少なく抑えることのできる LU 分解を用いた最小自乗最小ノルム問題の解法について考察する。以上のように、本論文で考察する問題は、疎な連立非線形方程式を Newton 法で解く際に現れる Jacobian を係数とする連立一次方程式の最小自乗最小ノルム問題である。ランクはほとんど full で、落ちても 2, 3 であるとする。

§2 LU 分解に基づく最小自乗最小ノルム解法

LU 分解を用いて最小自乗最小ノルム解を求めるのは一般的ではないので、ここで紹介する。Björck の最近の著書には LU 分解に基づくいくつかの方法が述べられている [1§2.5] が、問題が正方形に近くないと計算量がかかり、誤差も大きいので簡単な紹介にとどめてあるように思われる。

LU 分解の途中の状況は

$$\begin{pmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{pmatrix} = \begin{pmatrix} L_0 & 0 \\ L_1 & I \end{pmatrix} \begin{pmatrix} U_0 & U_1 \\ 0 & B \end{pmatrix} \tag{2.1}$$

のように書くことができる。ここで B が「まだ分解されていない」部分である。これが零行列となる場合には分解が続行できないが、このときランクが落ちているのである。この時方程式は

$$\begin{aligned} L_0(U_0x_0 + U_1x_1) &= b_0 \\ L_1(U_0x_0 + U_1x_1) &= b_1 \end{aligned} \tag{2.2}$$

となっている。ここで $M = L_1L_0^{-1}$ と $y = L_0(U_0x_0 + U_1x_1)$ とすると、これは

$$\begin{pmatrix} I \\ M \end{pmatrix} y = \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} \tag{2.3}$$

と書き直される。これを最小自乗問題として解くことにより、もとの問題の最小自乗解が得られる。ランク落ちが少ない場合、この問題は正規形にして CG 法を用いれば落ちたランクの数だけの反復で収束することが容易に証明でき、簡単に解けることがわかる。あるいは Sautter [2, 1: p. 74] の提案に基づいて

$$\begin{aligned} z &= (I + M^TM)^{-1}(b_1 + M^Tb_2) \\ &= b_1 + (I + M^TM)^{-1}M^T(b_2 - Mb_1) \\ &= b_1 + M^T(I + MM^T)^{-1}(b_2 - Mb_1) \end{aligned} \tag{2.4}$$

のように解くこともできる。

さて、これで $y = L_0(U_0x_0 + U_1x_1)$ が求まれば、x_1 を任意のベクトルとして

$$x_0 = U_0^{-1}(L_0^{-1}y - U_1x_1) \tag{2.5}$$

として最小自乗解が求まる。最小自乗最小ノルム解は

$$\begin{pmatrix} U_0^{-1}U_1 \\ I \end{pmatrix} x_1 \approx \begin{pmatrix} U_0^{-1}L_0^{-1}y \\ 0 \end{pmatrix} \tag{2.6}$$

という最小自乗問題を解いて x_0 を決めれば得られる。この左辺の行列は小さいので QR 分解などを用いて容易に解くことができる。

§2.1 枢軸選択について

この場合枢軸選択は、部分枢軸選択では不十分である。なぜなら要素がすべて 0 の列が一つあったからといって、残り ((2.1) の B) がすべて 0 とは限らないからである。このような場合には、零ベクトルでない列を探して列交換を行なわなければならない。のこりがすべて零ベクトルになればランク落ちとして終了すればよい。このような拡張された部分枢軸選択でも解けることは解けるが、実際には完全枢軸選択を行なう方がはるかによい。0 に近くなってしまったベクトルの要素はいずれもかなりの相対誤差を含んでいるから、他のベクトルに大きな要素が残っている時にそれを用いるのは極めて不利であるからである。

これとは別に、今回のように疎な問題を扱う Newton 法では、最初に枢軸選択を決めておいてそれを使い続けるという方法もあり得る。このような場合には、最初はランクが落ちていなかったにもかかわらず途中から落ちてしまったり、そうでなくても枢軸要素が 0 に近くなってしまったりすることがある。このような状況になっても、最初に設定した枢軸選択の情報を活かして分解を継続することができる。行列表示すると、

$$A = \begin{pmatrix} L_{00} & 0 & 0 \\ L_{10} & 1 & 0 \\ L_{20} & 0 & I \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & 0 & A_{12} \\ 0 & A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} U_{00} & U_{01} & U_{02} \\ 0 & 1 & 0 \\ 0 & 0 & I \end{pmatrix} \tag{2.7}$$

のようになっている。この時は要するに枢軸が 0 となった列行を最後に回せば良い。行列表示すると

$$A = \begin{pmatrix} L_{00} & 0 & 0 \\ L_{10} & 1 & 0 \\ L_{20} & 0 & I \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & 0 & 1 \\ 0 & I & 0 \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & A_{22} & A_{21} \\ 0 & A_{12} & 0 \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & 0 & I \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} U_{00} & U_{01} & U_{02} \\ 0 & 1 & 0 \\ 0 & 0 & I \end{pmatrix} \tag{2.8}$$

となる。A_{22} の部分は予定通りの枢軸選択で計算すれば良い。最後の列行については密行列とみなして LU 分解を実行することになる。この方法は、枢軸が零となる列行をとばして、最後にふちどり法で処理をしたものにほかならない。後回しになる列行の数が小さいうちはそれなりに有効な方法であろう。

§3 ランク判定における丸め誤差の問題

error.ps-15pt 前節で示した通り、LU 分解を用いても最小自乗最小ノルム問題を解くことが可能である。しかし QR 分解による方法と比較してみると、QR 分解の方が安定していることがある。その原因は、実際にはランクが落ちているにもかかわらず、丸め誤差のために要素が 0 にならないために 最小自乗最小ノルム解が求められていないことに起因することがわかった。そこでここでは QR 分解と LU 分解における丸め誤差を解析し、両者を比較する。ここでは QR 分解として修正 Gramm-Schmidt (MGS) 法を用いる。

LU 分解と QR 分解でのランク判定における丸め誤差の影響の違いの原因は主に二つあるものと考えられる。第一に、QR 分解では発生した誤差の一部が消滅して誤差が小さくなることがあるという点である。第二に、QR 分解と LU 分解とではそもそも発生する丸め誤差の大きさが相当に異なるということである。

§3.1 発生する丸め誤差の大きさ

計算で発生する丸め誤差の絶対的な大きさは、計算に出てくる値の絶対値で決まる。同じようにスケーリングされた行列から始めても、LU 分解では途中で要素が大きくなることがあるのに対し、QR 分解では行列要素の絶対値は直交化に伴って小さくなる一方である。この結果、発生する丸め誤差そのものの大きさが LU 分解と QR 分解とでは大きく異なる。

§3.2 誤差の伝搬

QR 分解では誤差の成分の一部が途中で消えてしまう。これは直交化をする時に、誤差もその直交成分が消去されることによる。また、直交化されていることにより、誤差を含んだベクトルの内積計算において一次の誤差が消えてしまう。このように、QR 分解では生じた誤差の一部は計算に影響を及ぼさない。

しかし QR 分解でも拡大する誤差はある。それはすでに直交化されたベクトルの成分である。これはその後直交化による消滅を受けないため、伝搬して拡大する。以下ではこの誤差の伝搬の様子を、高次の誤差を無視して調べる。

行列 A にベクトル a の成分の誤差が含まれているとする。ここでは a の成分はすでに直交化されているので、

$$a^T A = 0 \tag{3.1}$$

である。誤差をベクトル δ をもちいて

$$\tilde{A} = A + a\delta^T \tag{3.2}$$

と表現することにする。A に対する MGS 法が進んで

$$A = (A_1 A_2) = (QC)\begin{pmatrix} R_1 & R_2 \\ 0 & I \end{pmatrix} \tag{3.3}$$

となるものとする。ここで、上述のようにベクトルの内積では一次の誤差は消えてしまうので R の方には誤差が伝搬しない。そこで

$$(\tilde{A}_1 \tilde{A}_2) = (\tilde{Q}\tilde{C})\begin{pmatrix} R_1 & R_2 \\ 0 & I \end{pmatrix} \tag{3.4}$$

となる。これから $\tilde{C} = \tilde{A}_2 - \tilde{Q}R_2$ および$\tilde{A}_1 = \tilde{Q}R_1$ となるので

$$\begin{aligned} \tilde{C} &= \tilde{A}_2 - \tilde{A}_1 R^{-1} R_2 \\ &= C + a(\delta_2^T - \delta_1^T R_1^{-1} R_2) \end{aligned} \tag{3.5}$$

となることになる。よって誤差は $\|R_1^{-1}R_2\|$ を係数として拡大することがわかる。

一方 LU 分解の誤差の伝搬の様子を同様に高次の誤差を無視して解析すると以下のようになる。簡単にするため、ここでは LDU 分解を考える。誤差を含んだ行列 $\tilde{A} = A + E$ の LDU 分解の途中で

$$\tilde{A} = \begin{pmatrix} \tilde{L}_1 & 0 \\ \tilde{L}_2 & I \end{pmatrix}\begin{pmatrix} \tilde{D} & 0 \\ 0 & \tilde{B} \end{pmatrix}\begin{pmatrix} \tilde{U}_1 & \tilde{U}_2 \\ 0 & I \end{pmatrix} \tag{3.6}$$

となったとすると、$\tilde{B} = \tilde{A}_{22} - \tilde{A}_{21}\tilde{A}_{11}^{-1}\tilde{A}_{12}$ であるが、高次の誤差を無視するとこれは

$$\tilde{B} \approx B + E_{22} - E_{21}A_{11}^{-1}A_{12} - A_{21}(\tilde{A}_{11}^{-1} - A_{11}^{-1})A_{12} - A_{21}A_{11}^{-1}E_{12} \tag{3.7}$$

となる。$A_{21}A_{11}^{-1} = L_2 L_1^{-1}$ と $A_{11}^{-1}A_{12} = U_1^{-1}U_2$ に注意すると、

$$\tilde{B} - B \approx E_{22} - E_{21}U_1^{-1}U_2 + L_2 L_1^{-1} E_{11} U_1^{-1} U_2 - L_2 L_1^{-1} E_{12} \tag{3.8}$$

となるから、誤差は $\|L_2 L_1^{-1}\|\|U_1^{-1}U_2\|$ を係数として拡大することがわかる。

完全枢軸選択を用いた場合には R_1, L_1, U_1 はいずれも対角要素が 1 で非対角要素が 1 以下の三角行列となる。また、R_2, L_2, U_2 はいずれも要素の絶対値が 1 以下の行列である。そこで、$\|R_1^{-1}R_2\|, \|L_2L_1^{-1}\|,$ $\|U_1^{-1}U_2\|$ はいずれも同じような挙動を示すものと考えられる。このため LU 分解では QR 分解の場合の自乗の係数で誤差が拡大するということになる。

§3.3 ランクが落ちかかっている場合

ランクが完全に落ちているわけではないが、極めて小さい特異値が存在する場合には方程式が不安定になるので、この特異値の成分を切捨てて近似解を求めることが必要となる。この場合には、この極めて小さい特異値の存在が認識できるかどうかが問題になる。この問題についても、小さい特異値に対応する成分を「誤差」とみなせば上の議論がそのまま当てはまる。すなわち、小さい徳一の成分が QR 分解では $\|R_1^{-1}R_2\|$, LU 分解では $\|L_2L_1^{-1}\|\|U_1^{-1}U_2\|$ 倍に拡大され得る。

これらのノルムは最悪の場合 2^{n-1} 程度にまでなりうるので、QR 分解でも LU 分解でもランクの落ち具合を正確に判定することはできないことがわかる。近年これに対して最小特異値をかなり正確に評価をすることができるランク顕現分解の方法が提案されている [1: §2.7]。

§3.4 疎行列の場合

さて上記の解析はいずれも密行列の場合を想定して行なった。疎行列の場合にも上記の解析は有効であるが、QR 分解の $\|R_1^{-1}R_2\|$ と LU 分解の $\|L_2L_1^{-1}\|$, $\|U_1^{-1}U_2\|$ とでは fill-in が異なるので、同じような挙動を示すというわけにはゆかない。疎行列の性質にもよるが、fill-in によって変わる依存の深さが異なり、LU 分解の方が相当に浅い場合には LU 分解の方が誤差の伝搬が小さい可能性も十分にある。このため、疎行列の場合には QR 分解よりも LU 分解の方がわるいとは言い切れない。この点については今後更に研究して評価を進めたい。

§4 数値実験による評価

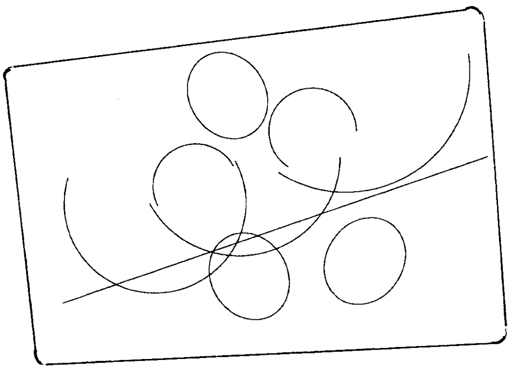

図 4.1. CFIG^{TM} による作図の簡単な例

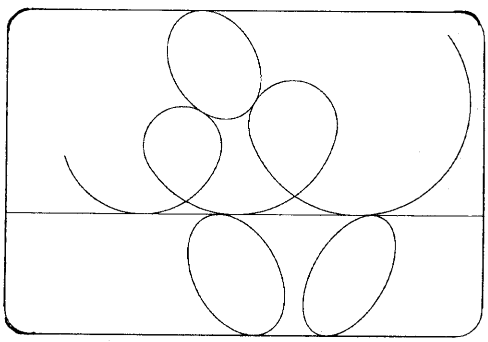

図 4.2. 制約方程式を解いた後の状態

最後に実際に連立非線形方程式を解いた場合に LU 分解による方法と QR 分解による方法とでどの程度性能が異なるかを評価する。方程式は松本尚氏*が現在開発しておられる CFIG^{TM} が生成するものを使わせて頂いた。CFIG^{TM} は制約条件を用いて上の例のような図を記述することのできる作図エディタであるが、インタラクティブなシステムであるため応答性能に直接関係する制約方程式の解法系の速度が極めて重要な意味を持つ。今回はいくつかの作図をする際に現れた制約条件方程式を出力して頂いたもののうち、多項式の等式条件のみのもので最初からまたは解の付近でランクの落ちるものを使わせて頂いた。いずれも解の存在する問題であるが、なかには初期値が解から遠く設定されており容易に収束しないものも含まれている。

次の頁の表はこれらの問題について、QR 分解と LU 分解とをもちいた非線形解法の様子を示している。ここで DQR は密行列用の QR 分解で、完全に枢軸選択をしており、比較したうちでは最も精度が高いと思われるものである。SQR は疎行列用に手を加えた QR 分解で、fill-in を少なくするために安定性を犠牲にするような枢軸選択を行なっている。PLU は部分枢軸選択つきの LU 分解で、計算量を下げること

* 東京大学 大学院・理学系研究科 情報科学専攻 助手、e-mail: tm@is.s.u-tokyo.ac.jp

表 4.1. 線形方程式による解法の挙動の相違

ランク/規模	DQR		SQR		PLU		CLU	
	time	iter	time	iter	time	iter	time	iter
11/12	0.25	12	0.17	12	0.13	12x	0.15	12
16/17	0.02	3	0.02	3	0.02	3	0.02	3
16/17	0.02	3	0.02	3x	0.03	3x	0.02	3
32/33	0.10	4	0.07	4	0.05	4	0.05	4
32/33	0.12	4	0.10	4x	0.08	4x	0.10	4
56/57	0.32	13	0.22	13x	0.27	14x	0.30	14x
56/57	0.70	17	0.42	17	0.62	31xf	1.80	57x
55/57	1.07	18	0.53	18x	0.63	20xf	0.57	18xf
117/118	5.17	9	2.10	8x	2.23	14x	1.45	7x
143/145	26.58	26f	11.73	30xf	3.25	26xf	1.15	7x
143/145	5.15	6	2.37	6x	3.22	20xf	1.40	5x
146/147	7.17	7	2.93	7x	4.62	17x	2.75	8x
147/148	7.18	7	3.13	7x	4.03	13xf	3.05	8x
147/148	6.50	7	2.37	7	2.02	9xf	1.78	8x

を第一に作ったものである。CLU は完全枢軸選択つきの LU 分解で、安定性を確保できる範囲で疎行列性を活かそうという枢軸選択になっており、さらにランクの判定の閾値に行列のサイズを掛けて緩めてある。おのおの x のついているものはランクの判定の結果が DQR とは異るもの、f のついているものは Newton 法が収束しなかったものである。

これを見てまずわかることは、ランクの判定はそれぞれの解法で異っていることである。これは誤差がランク判定の閾値 (10^{-7}) の近くまで来ているからと思われる。大体、疎行列性を活かすために安定性を犠牲にしたものは、落ちているはずのランクを落ちていないと判定しがちである。CLU が DQR に最も近いランク落ちの判定をした。このことから、安定性にかなり気を使わなければ QR でも LU でもランク落ちの判定に失敗するということがわかる。乱暴な QR よりは安全な LU の方がランク判定の精度はよい。しかし、SQR と PLU の「乱暴さ」はおなじくらいであるが QR の方がはるかに結果がよいので、今回の実験の範囲ではやはり QR の方が安定性は優れているらしい。

所要時間を比べると、CLU が最も速いことがわかる。一回あたりの計算量は PLU の方がはるかに少ないのであるが、安定性を欠くため結局損をしてしまう。CLU が DQR よりも安定に解けている場合もあるので、ここでもやはり QR 分解でも信用してはいけないということがわかる。なお、ここでは示していないが、安定にとける場合にはもちろん PLU が最も速い。

以上から知見をまとめると、(1) QR 分解でも安心はできない。問題が大きくなると誤差の拡大は避けられず、ランク落ちの判定は正確ではない。逆に LU 分解でも注意して用いれば QR 分解と同程度のランク落ちの判定ができる。LU 分解は特に疎行列の場合に非常に高速なので有益である。(2) 安定性を求めると fill-in が多くなり、計算量が増大する。しかし、これは QR 分解でも同じである。また、完全枢軸選択はかなりの計算量になる。しかしこれも QR 分解の枢軸選択でも同じである。

§5 まとめ

ランクの判定には QR 分解にせよ LU 分解にせよ、十分慎重な検討が必要である。この問題を解決すれば、LU 分解でも十分使うことができ、QR 分解に比べて高速である。最終的には、安定性と速度とが秤にかけられることになるが、これは疎行列を扱う場合には常に付きまとう問題である。

謝辞

東京大学の松本尚氏に頂いた多くの有用な議論に感謝致します。

参考文献

[1] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, 1996.

[2] W. Sautter, "Fehleranalyse für die Gauss-Elimination zur Berechnung der Lösung minimaler Länge," *Numer. Math.*, 30, pp. 165–184, 1978.