Journal of the Korean Institute of Industrial Engineers
Vol. 25, No. 3, pp. 290-297, September 1999.

내부점 방법에서 촐레스키 분해의 수치적 안정성†

설동렬[1] · 성명기[2] · 안재근[3] · 박순달[2]

[1] LG-EDS / [2] 서울대학교 산업공학과 / [3] 한경대학교 컴퓨터공학과

Numerical Stability of Cholesky Factorization in Interior Point Methods for Linear Programming

Tongryeol Seol[1] · Myeongki Seong[2] · Jaegeun Ahn[3] · Soondal Park[2]

In interior point methods for linear programming, we must solve a linear system with a symmetric positive definite matrix at every iteration, and Cholesky factorization is generally used to solve it. Therefore, if Cholesky factorization is not done successfully, many iterations are needed to find the optimal solution or we can not find it. We studied methods for improving the numerical stability of Cholesky factorization and the accuracy of the solution of the linear system.

1. 개 요

선형계획법(linear programming)의 대표적 해법으로는 단체법(simplex method)과 내부점 방법(interior point method)이 있다. 이 중 내부점 방법은 단체법에 비해서 대형문제를 효율적으로 풀 수 있기 때문에 대형문제의 해법으로 널리 사용되고 있다(박순달, 1999).

내부점 방법은 초기해에서 시작하여 매회 현재의 해를 개선할 수 있는 개선방향을 구해, 이를 따라 해를 수정하는 과정을 최적조건에 이를 때까지 반복하는 방법이다. 개선방향을 구할 때는 다음과 같은 연립선형방정식을 풀어야 한다.

$$(A\Theta A^T)\Delta y = \xi \qquad (1)$$

여기서 행렬 A는 $m\times n$ 행렬이고 Θ는 모든 대각요소의 값이 양인 $n\times n$ 대각행렬인데, 내부점 방법의 종류에 따라 다르게 구해진다. Δy와 ξ는 m차 벡터이다.

식 (1)을 풀기 위해서는 $A\Theta A^T$의 역행렬을 구해야 하는데, 이를 직접 구하는 것보다는 $A\Theta A^T$가 가지는 대칭양정치(symmetric positive definite) 특성을 활용하여 식 (2)와 같이 촐레스키 분해(Cholesky factorization)(George and Liu, 1981)를 이용하는 방법이 효율적이어서 널리 사용되고 있다(설동렬, 1999; Lustig, et al., 1994).

$$A\Theta A^T = LL^T \qquad (2)$$

여기서 L은 하삼각행렬로서 촐레스키 요소(Cholesky factor)라고 불린다.

식 (1)을 풀 때 촐레스키 분해 이외의 방법, 예를 들어 QR 분해(Golub and Van Loan, 1983), 상하분해(Duff, 1986), 켤레경사법(conjugate gradient method)(Avriel, 1976) 등을 사용할 수도 있다. 그러나 일반적인 선형계획문제에 대해서 이들 방법은 촐레스키 분해에 비해 매우 느리기 때문에 특수한 행렬을 가지는 문제 이외에는 거의 사용되지 않는다. 따라서, 이 논문에서는 범용으로 널리 사용되고 있는 촐레스키 분해를 다루었다.

대칭양정치 행렬의 촐레스키 분해는 이론적으로 수치적 안정성이 보장된다(Wilkinson, 1961). 따라서 이론적으로는 수치적 안정성에 대한 고려없이 $A\Theta A^T$의 희소성(sparsity)만을 고려하여 촐레스키 분해를 수행할 수 있다(Duff, 1986). 그러나 실제 문제를 풀 때는 $A\Theta A^T$가 항상 수치적으로 안정된 것은 아니기 때문에 촐레스키 분해가 실패하거나 개선방향을 구할 때 오차가 발생한다.

이러한 원인으로는 다음과 같다.

- A가 선형종속인 경우
- A나 Θ의 규모화(scaling)가 좋지 않은 경우
- Θ값에 의해 최적해 근처에서 $A\Theta A^T$가 불안정한 경우

이 논문에서는 위와 같은 상황에서 촐레스키 분해를 안정

† 본 연구는 한국과학재단의 특정기초연구과제 (과제번호 98-0200-07-01-2)에 의해 지원되었음.

적으로 수행하도록 하는 방법과 오차가 발생했을 때 이를 줄이는 방법에 대해 다룬다.

2. 중복행의 제거

내부점 방법의 기본적인 가정 중 하나는 행렬 A에 중복행이 없다는 것이다(박순달, 1999). 그런데 실제 문제에서는 중복행이 존재할 수 있으므로 이를 제거하지 않으면, AA^T가 대칭양정치 행렬이 아니기 때문에 촐레스키 분해가 제대로 수행될 수 없다.

따라서 해법을 수행하기 전에 중복행을 제거할 필요가 있다. 중복행은 사전처리(preprocessing)(성명기 등, 1999; Gondzio, 1997)를 통해 제거할 수 있다.

그러나 일반적인 사전처리 방법으로는 모든 중복행을 제거하지 못하므로 사전처리를 거치더라도 중복행이 남아 있을 수 있다.

Andersen(1995)은 모든 중복행을 제거할 수 있는 방법을 제안했지만, 이 방법은 많은 시간이 소요되므로 실용적인 목적의 프로그램에서는 사용되지 않는다.

비교적 적은 노력으로 중복행을 제거하는 방법으로는, AA^T에 대해 촐레스키 분해를 수행할 때 촐레스키 요소의 대각요소의 값을 이용하여 중복행을 판정하는 방법이 있다. 다음 정리를 보자.

정리[Mészáros, 1998] AA^T의 i번째 행을 $\boldsymbol{a}_i$라고 할 때, AA^T의 k번째 대각요소의 값은 $\boldsymbol{a}_1, \cdots, \boldsymbol{a}_{k-1}$이 만드는 선형부공간(linear subspace)과 $\boldsymbol{a}_k$ 사이의 거리와 같다. ■

위의 정리에 의해서 식 (1)에서 Θ를 단위행렬로 두고 촐레스키 분해를 할 때 촐레스키 요소의 대각요소의 값이 0인 행을 중복행이라고 판단할 수 있다.

이 방법을 사용하기 위해서 특별히 Θ를 단위행렬로 두고 촐레스키 분해를 할 수도 있지만, 본 논문의 내용을 구현한 LPABO(http://orlab.snu.ac.kr/software/lpabo.html)에서는 초기해를 구할 때 AA^T의 분해가 있기 때문에 부가적인 연산은 거의 없었다.

LPABO에서 초기해를 구하는 방법은 Mehrotra가 제안한 방법(Lustig, 1992)을 약간 변형한 것으로 AA^T에 대한 촐레스키 분해를 이용하고 있는데, 이와 같이 AA^T을 이용한 초기해를 구하는 방법은 성능이 좋기 때문에 CPLEX, HOPDM 등에서도 이와 비슷한 방법을 사용하고 있는 것으로 알려져 있다.

LPABO를 이용하여 Netlib 문제(ftp://orlab. snu.ac.kr/pub/lpdata/netlib)와 Kennington 문제(ftp://orlab.snu.ac.kr/pub/lpdata/kennington)를 풀었을 때, AA^T의 촐레스키 분해를 이용하여 찾은 중복행의 개수는 <표 1>과 같다. 이 때 촐레스키 요소의 대각요소의 값이 10^{-10}이하인 행에 대해 중복행이라고 판정하였다.

<표 1>에서 왼쪽 부분이 Netlib 문제이고, 오른쪽 부분이 Kennington 문제이다. Netlib 문제에서 중복행으로 판정된 개수는 사전처리 전과 후가 다르나, Kennington 문제에 대해서는 판정된 개수가 같음을 알 수 있다.

내부점 방법 프로그램에서 촐레스키 요소의 비영요소의 수는 해를 찾는 속도에 많은 영향을 준다. 따라서 문제를 풀기 전에 최소차수순서화(minimum degree ordering)(George and Liu, 1981) 등을 이용하여 비영요소의 수를 줄이는 분해순서를 찾는다. 또 효율적인 계산을 위해 촐레스키 요소의 자료구조를 미리 정해 놓게 된다.

따라서 중복행을 찾아서 이를 제거하려면 분해순서와 촐레스키 요소의 자료구조를 이에 맞게 변경해 주어야 한다. 그런데 중복행을 제거한 후 다시 순서화를 하게 되면, 순서화는 경험적(heuristic) 방법이므로 새로운 분해순서가 항상 전보다 촐레스키 요소의 비영요소 수를 더 작게 한다는 것을

표 1. 초기 촐레스키 분해에서 판정한 중복제약식의 개수

문제이름	제약식 수	사전처리 전	사전처리 후	문제이름	제약식 수	사전처리 전	사전처리 후
bore3d	233	2	0	cre-a	3516	5	5
cycle	1903	15	0	cre-b	9648	8	8
degen2	444	2	2	cre-c	3068	5	5
degen3	1503	2	2	cre-d	8926	8	8
dfl001	6071	13	13	ken-07	2426	49	49
etamacro	400	1	0	ken-11	14694	121	121
maros	846	1	0	ken-13	28632	169	169
pilotja	940	12	0	ken-18	105127	324	324
pilotnov	975	8	0	pds-02	2953	11	11
recipe	91	2	0	pds-06	9881	11	11
scorpion	388	30	0	pds-10	16558	11	11
shell	536	1	1	pds-20	33874	11	11
sierra	1227	12	10				
wood1p	244	1	1				

표 2. 개량된 촐레스키 분해의 실험 결과

문제이름	제약식 수	적용된 제약식 수
80bau3b	1992	6
cycle	1270	13
czprob	474	1
d2q06c	1972	7
degen3	1453	277
dfl001	5808	1787
greenbea	1702	813
greenbeb	1697	11
maros	579	24
pilot	1361	1
pilotja	761	8
pilotnov	808	2
ship08l	470	4
ship12l	610	2
ship12s	268	2
wood1p	169	9
woodw	709	2

$$d_{ii} = 1/\sqrt{(A\Theta A^T)_{ii}}$$

d_{ii}를 위와 같이 하면 촐레스키 분해를 수행하기 이전인, $A\Theta A^T$을 계산하는 단계에서 별도의 노력을 들이지 않고 규모화 행렬을 생성할 수 있으므로 효과적이다.

4. 개량된 촐레스키 분해

내부점 방법의 가장 중요한 어려움 중의 하나는 해가 최적해에 가까워졌을 때, 대칭양정치 행렬로 이루어진 선형시스템이 수치적으로 불안정해질 수 있다는 것이다. 퇴화문제의 경우에, 만약 활성제약식(active constraint)의 야코비안(Jacobian)이 최적해에서 선형독립이 아니게 되면 촐레스키 분해가 실패하게 된다. 이는 주로 분해과정에서 발생하는 매우 작은 선회 요소에 의해 야기된다(Mészáros, 1998).

이러한 경우를 극복하는 하나의 방법은 수치적 계산 과정에서 작은 선회요소에 해당되는 요소를 건너뛰는 방법인 개량된 촐레스키 분해(modified Cholesky factorization)를 사용하는 것이다. 이는 선회요소의 값을 매우 큰 수로 정하거나 또는 분해요소를 영으로 정하는 방법으로 구현된다(Wright, 1998).

또 다른 방법은 작은 선회요소의 값을 적당히 큰 값으로 만드는 정규화(regularization)이다. Andersen et al.(1996)에서는 2차벌금항(quadratic penalty term)을 이용하여 정규화가 어떻게 고려되는지에 대해 기술하였다. 다른 한편으로 Mészáros (1998)에서의 분석은 작은 정규화가 선형시스템의 해에서 작은 섭동(perturbation)을 야기한다는 것을 보여주었다. 그리고 이는 반복정제(iterative refinement)에 의해서 줄어들 수 있다는 것도 보였다.

이러한 접근법들의 공통점은 작은 선회요소의 값을 어떻게 정의하느냐에 매우 민감하다는 것이다. Mészáros(1998)에서는 문제행렬 A의 값에 관계없이 적용될 수 있는 방법을 제시하였다. 즉, 2절에서의 정리와 같이 촐레스키 분해 도중 대각요소의 값이 0에 가까우면 행렬 A에서 k번째와 그 후의 계산되어지는 AA^T의 촐레스키 요소는 그 전의 계산 오차에 대부분 영향을 받기 때문에 이를 이용하는 연산은 신뢰적이지 못하게 된다.

LPABO에 구현한 방법은 d_{ii}의 값이 10^{-10} 이하가 되면 L의 i번째 행과 열의 모든 요소를 0으로 두도록 하는 것이다.

<표 2>는 NETLIB 문제 중 A의 비영요소가 10,000개 이상인 문제에 대해서 촐레스키 분해를 할 때 d_{ii}의 값이 10^{-10} 이하가 되는 최대 개수를 적은 것이다.

<표 2>의 결과는 사전처리와 초기 촐레스키 분해에서 중복제약식을 제거한 다음 촐레스키 분해를 적용한 것이다. 따라서 2번째 열의 '제약식 수'는 실제 문제를 풀 때의 제약식의 개수이다. <표 2>에서 보는 바와 같이 degen3, dfl001, greenbea 등은 수치적으로 매우 불안정한 문제라는 것을 알 수 있다.

한편, 개량된 촐레스키 분해 방법을 적용한 경우와 적용하지 않은 경우에 대해서 반복수의 차이에 대해서도 실험하였는데, greenbea 한 문제에서만 반복수가 42회에서 37회로 줄어들었고 나머지는 변화가 없었다.

5. 선회순서를 바꾸는 방법

촐레스키 분해를 수행하는 도중에 0에 가까운 선회요소가 발생하는 경우에, 즉 불안정한 행이 있는 경우에, 선회순서를 바꾸어 주는 방법을 고려할 수 있다.

선회순서를 수정하는 접근 방법은 촐레스키 분해가 실패하는 원인으로 촐레스키 분해를 수행할 때에 수치적으로 불안정한 선회요소가 먼저 계산되었기 때문이라는 생각에서 출발한다. 따라서, 선회요소의 값이 0에 가까운 값($< \varepsilon$: LPABO에서는 10^{-6})을 가지게 되면 이를 선회순서의 뒤로 가도록 한다(Mészáros, 1998).

이러한 방법을 제안한 문헌들은 있으나 이를 어떻게 구현했는지에 대해서는 문헌에서 자세히 밝히고 있지 않다. 그리고 대형문제를 다루기 위해서 희소행렬 기법을 사용하는 경우에 선회순서 변경에 의한 자료구조의 변경이 간단하지 않기 때문에 이 연구에서는 두 가지 대안을 제시하고 이에 대해 LPABO에서 실험한 결과를 제시하고자 한다.

촐레스키 분해 방법 중에서 열 단위로 분해하는 방법과,

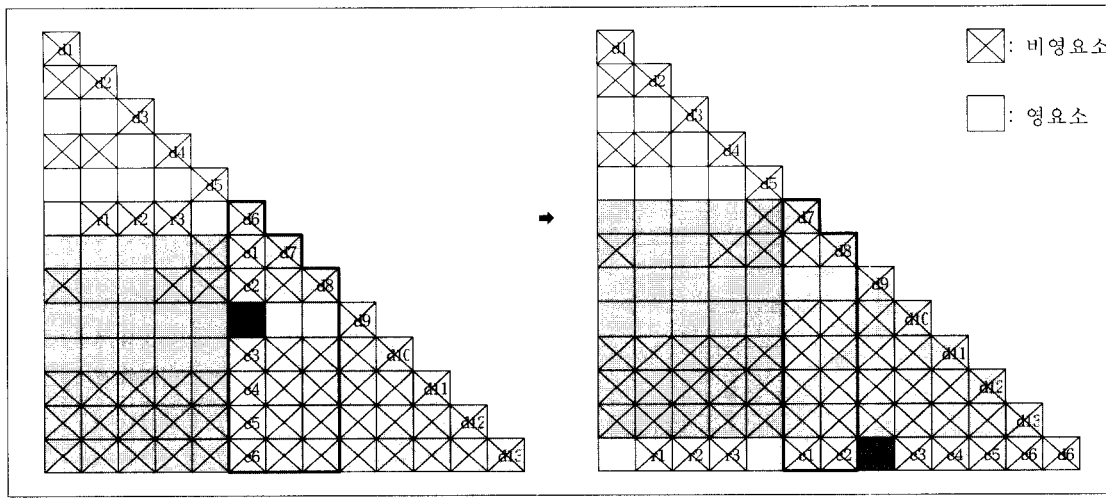

그림 1. 선회순서를 맨 뒤로 옮길 때의 비영요소 구조의 변화.

또 그에 따라 촐레스키 요소 또한 열 단위로 보관하는 방법이 많이 사용되고 있다(Lustig, et al., 1992). 따라서 본 연구에서는 촐레스키 요소가 열 단위로 보관되어 있다고 가정한다. 촐레스키 분해를 하는 행렬은 좌우대칭이므로 행과 열의 구분이 필요없다. 열촐레스키 분해이므로 편의상 열로 표현하겠다.

선회순서를 변경한다는 것은, 촐레스키 분해 중에 대각요소의 값이 ε 이하인 열이 나타나면 이 열을 분해순서의 제일 뒤쪽으로 보내는 방법이다. 그런데 이렇게 되면 촐레스키 요소의 비영요소 구조가 변경된다.

예를 들어 <그림 1>의 좌측과 같은 촐레스키 요소가 있을 때 여섯 번째 대각요소, 즉 d_6의 값이 ε 이하여서 여섯 번째 열을 제일 마지막에 선회연산하도록 했다고 하자. 그러면 이 후의 열들은 모두 순서가 하나씩 앞으로 당겨진다. <그림 1>의 우측이 그 결과인데, 그림에서 볼 수 있듯이 촐레스키 요소의 열구조가 변경되게 된다. 이때 좌측의 여섯 번째 열과 행은 우측의 촐레스키 분해요소의 마지막 행으로 옮겨지게 되는데, 그림에서처럼 여섯 번째 열에서 좌측에서는 0이었던 요소가 우측에서는 비영요소가 되는 경우(<그림 1>에서 가장 짙게 표시된 부분)도 있게 된다.

분해순서가 변경되면 이와 같이 비영요소가 추가될 뿐만 아니라 각 열의 구조도 변경되므로 고정된 촐레스키 요소의 자료구조를 사용할 수 없다. 즉, 촐레스키 요소의 자료구조가 매번 변경되기 때문에, 행 지수의 압축보관방법(compressed storage scheme)(George and Liu, 1981)이나 초마디(supernode)를 활용한 방법(Jung, et al., 1994) 등을 사용할 수 없어서 촐레스키 분해의 효율성이 떨어지게 된다.

따라서 선회연산 순서를 변경하는 방법에서는 압축보관방법과 초마디방법 등과 같이 촐레스키 분해를 빠르게 해주는 방법을 그대로 유지하면서 부가적인 연산을 최소화하도록 하는 것이 중요하다.

이 연구에서는 다음의 두 가지 대안을 고려하였다.

• 즉시처리(on-line) 방법: 촐레스키 요소의 자료구조에 미리 여유공간을 할당한 다음, 불안정한 행이 발견되면 즉시 뒤로 보내는 방법이다.

• 일괄처리(batch) 방법: 불안정한 행의 개수가 미리 정해진 수를 넘을 경우, 선회순서를 변경하고 이에 따른 촐레스키 요소의 자료구조를 새로 생성하는 방법이다.

이를 그림으로 표현하면 <그림 2>와 같다.

<그림 2>의 (가)는 즉시처리 방법을 설명한 것으로, 미리 각 열마다 여유공간을 준비한 다음 선회순서를 바꿀 때 미리 준비해 두었던 곳에 선회요소에 대응되는 값들을 옮기고, 원래의 자리에는 0으로 채우는 방법이다.

원래 사용하던 부분에 0을 넣고, 새로운 행/열로 옮겨두기 때문에 결과적으로 행렬의 크기가 그만큼 증가하게 된다. 그러나 기존의 초마디 구조와 같은 비영요소 구조의 특성은 전혀 변화가 없으므로 선회요소 변경에 따른 처리가 간단히 해결된다. 그러나 뒤로 보내는 행의 개수를 제한하지 않으면 불안정한 행의 개수가 많은 문제에서는 메모리가 많이 필요하게 되고 부가적인 연산시간도 그만큼 증가한다. 따라서 본 연구에서는 행렬 A의 행의 크기(m)에 따라 다음과 같이 뒤로 보낼 수 있는 행의 개수($m_{\max}$)를 제한하였다.

- $m < 500$ 이면, $m_{\max} = 5$
- $m < 1000$ 이면, $m_{\max} = 10$
- $m < 5000$ 이면, $m_{\max} = 15$
- $m \geq 5000$ 이면, $m_{\max} = 20$

<그림 2>의 (나)는 일괄처리 방법을 설명한 것으로, 불안정한 행의 개수가 미리 정해진 수(본 연구에서는 앞에서 설명한 $m_{\max}$)를 넘으면 이들 행을 선회순서의 맨 뒤로 돌린 다음 촐레스키 요소의 자료구조를 새로 구하는 방법이다. 이 방법은 즉시처리 방법과 달리 여유공간이 필요 없고 뒤로 보내는 행의 개수에 제한이 없다는 장점이 있지만, 촐레스키

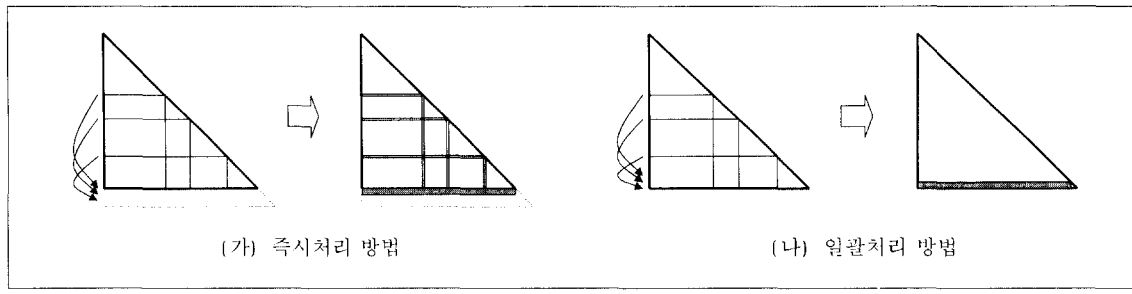

그림 2. 즉시처리 방법과 일괄처리 방법.

요소의 비영요소의 개수가 많이 달라질 수 있다는 것과 자료 구조를 새로 정해주는 데 시간이 많이 소요된다는 단점이 있다.

촐레스키 요소의 비영요소를 줄이기 위해서는 순서화를 다시 해야 할 필요가 있다. 이 연구에서는 이 순서화에 시간이 많이 소요되므로 이를 생략하고 대신 비영요소의 증가가 전보다 1.5배 이하가 되면 선회순서를 변경하지 않도록 하였다. 또 이 연구에서는 불안정한 행이 $m_{\max}$ 개 이상일 때 선회순서를 변경하도록 하였는데, 이 때 $m_{\max}=2\times m_{\max}$ 으로 수정하도록 하였다.

두 가지 방법에 대하여 어떤 것이 우수한가를 확인하기 위해서 이 연구에서는 두 가지 실험을 하였다. 하나는 수치적 안정성에 관한 실험이고 다른 하나는 수행속도에 관한 것이다.

<표 3>은 Netlib 문제와 Gondzio 문제(ftp:// orlab.snu.ac.kr /pub/lpdata/gondzio) 중에서 선회순서를 변경하지 않은 경우(A), 즉시처리 방법을 사용한 경우(B), 그리고 일괄처리 방법을 사용한 경우(C)에 대해서 불안정한 행의 개수가 $m_{\max}$ 이상이 되었을 때의 촐레스키 분해의 수치적 오차를 측정한 것이다. 표 안에서의 수치는 식 (1)의 오차, 즉 다음의 값을 의미한다.

$$\max\ |\ (\xi - A\Theta A^{T}\Delta y)_j\ |$$

<표 3>을 보면, 일괄처리 방법을 사용한 경우가 수치오차가 작게 나오는 것을 알 수 있다. 즉시처리의 경우는 뒤로 돌리는 행의 개수를 제한함으로써 원래 방법에 비해서 수치오차가 줄어들지 않는 경우도 있음을 알 수 있다.

<표 4>는 한 회당 소요되는 계산시간을 <표 3>과 같이 세 가지 경우에 대해 비교한 것이다

표 4. 선회순서 변경에 따른 한 회당 계산시간

문제이름	A	B	C
stocfor3	0.65	0.72	0.69
co9	2.24	2.43	2.34
ge	0.74	0.78	0.76
mod2	5.47	5.85	5.80
world	5.76	5.89	5.85

<표 4>에서 보는 바와 같이 선회순서를 변경하는 방법은 부가연산 때문에 계산시간이 증가되었다. 그렇지만 선회순서를 변경하는 경우는 그렇지 않은 경우에 비해서 반복횟수가 줄어드는 경우(ge)도 있고 또 풀지 못하는 문제를 풀 수 있는 경우(일괄처리 방법에서 mod2, world)도 있었기 때문에 수치적으로 불안정한 문제에 대해서는 필요한 방법이라고 할 수 있다.

6. 반복정제

앞에서 설명한 방법들이 촐레스키 분해 자체의 오류를 줄이기 위한 방법이라면, 반복정제는 주어진 촐레스키 요소를 이용하여 선형방정식의 해를 구할 때 발생하는 오차를 줄이는 방법이다.

선형시스템 $My=\xi$를 푼다고 하자. 이때 M이 가역행렬이라고 하면 $y=M^{-1}\xi$가 해가 된다. 이때 행렬 M이

표 3. 선회순서 변경에 따른 수치오차

문제이름	행의 개수	A	B	C
stocfor3	16,675	5.812097e-10	5.812097e-10	2.844319e-10
co9	10,789	1.118792e-09	1.024801e-09	6.020855e-10
ge	10,099	1.391163e-06	1.391163e-06	5.729671e-07
mod2	34,774	1.432049e-05	1.232528e-05	9.505905e-06
world	34,506	8.422197e-06	8.422197e-06	3.468013e-06

수치적으로 불안정하다면 $\xi - My$의 값이 정확하게 0이 아니다. 이를 잔여벡터(residual vector) r이라고 하자.

$$r^{(1)} = \xi - My^{(1)}$$

여기서 $y^{(1)}$은 부정확한 M^{-1}에 의해 계산된 값이고 $r^{(1)}$은 그 잔여벡터이다.

이때 다음의 관계가 성립한다.

$$\begin{aligned} M^{-1}r^{(1)} &= M^{-1}\xi - y^{(1)} \\ &= y - y^{(1)} \\ &= \Delta y^{(1)} \end{aligned}$$

그러므로 $y^{(1)}$의 수정벡터 $\Delta y^{(1)}$를 $M\Delta y^{(1)} = r^{(1)}$와 같이 구할 수 있다.

물론 실제적으로 이를 정확하게 풀 수는 없지만 새로운 근사해를 $y^{(2)} = y^{(1)} + \Delta y^{(1)}$와 같이 구할 수 있다.

일반적으로 반복적인 과정을 다음과 같이 표현한다.

각 $k(=1, 2, \cdots)$에 대하여

$$r^{(k)} = \xi - My^{(k)}$$
$$M\Delta y^{(k)} = r^{(k)}$$
$$\Delta y^{(k)} = M^{-1}r^{(k)}$$
$$y^{(k+1)} = y^{(k)} + \Delta y^{(k)}$$

LPABO에서 구현할 때는 오차를 측정한 후 일정량 이상의 오차가 발생하게 되면 적정한 횟수만큼 반복정제를 수행하게 하였다. 즉, 내부점 방법에서는 그 특성상 원제약식에서 오차가 많이 발생하기 때문에 원제약식에 해당하는 등식의 오차가 10^{-6}이상이 되면 반복정제를 실행하게 하였고, 도중에 오차가 10^{-6} 미만이 되든지 횟수가 세 번이 될 때까지 수행하였다.

행렬 A의 비영요소가 10,000개 이상인 Netlib 문제와 Gondzio 문제에 대해 실험하였을 때, 반복정제에 의해서 반복수가 변경된 문제는 greenbea, mod2인데 반복수가 각각 80과 59에서 37과 57로 줄었다.

7. 결 론

내부점 방법으로 선형계획 문제를 풀 때 가장 핵심이 되는 촐레스키 분해의 수치적 안정성을 높이는 방법들에 대해서 연구하였다.

촐레스키 분해에서 오차가 발생하는 원인을 다음과 같이 나누고 이에 대한 방법에 대해 알아보았다.

- 문제 행렬에 중복제약식이 있는 경우
- 규모화가 좋지 않은 경우
- 대칭양정치 행렬이 불안정한 경우

문제행렬에 중복제약식이 있는 경우에는 사전처리와 초기 촐레스키 분해를 통해 미리 제거할 수 있다.

내부점 방법의 해법 특성상 문제행렬의 규모화 상태가 좋더라도 대각행렬 Θ에 의해 $A\Theta A^T$의 규모화가 나빠질 수 있다. 이 때에는 $A\Theta A^T$에 대해 대각요소의 절대값이 1 이하가 되도록 대칭적으로 규모화를 적용하여 이를 해결할 수 있다.

촐레스키 분해를 수행하는 도중에 수치적으로 불안정한 선회요소가 발생한 경우에는 개량된 촐레스키 분해방법과 선회순서를 변경하는 방법을 적용할 수 있다. 이때 선회순서를 변경하는 방법은 촐레스키 요소의 비영요소 구조가 변경될 수 있기 때문에 기존의 효율화 방법을 사용하지 못할 수 있다. 이 연구에서는 두 가지 방법, 즉 즉시처리 방법과 일괄처리 방법을 제시하였다. 이 중 일괄처리 방법이 우수한 것으로 나타났다.

촐레스키 분해를 수행한 다음 구해진 촐레스키 요소를 이용하여 선형시스템의 해를 구할 때 발생하는 오차는 반복정제를 적용하여 줄일 수 있다.

이와 같은 수치적 안정성 향상 기법은 내부점 방법에서 해의 정밀도를 향상시킬 뿐 아니라 개선방향의 정확도를 높여 주므로 반복수를 줄이는 효과를 함께 얻을 수 있다.

참고문헌

박순달(1999), 선형계획법, 4판, 민영사.

서용원, 김우제, 박순달(1995), 선형계획법프로그램의 수치오차보정과 행렬희소도 유지, 한국경영과학회 '95추계학술대회논문집, 363-369.

설동렬(1999), 내부점 방법에서의 촐레스키 분해, 박사학위논문, 서울대학교.

성명기, 임성묵, 박순달(1999), 내부점 방법을 위한 사전처리의 구현, 한국경영과학회지, 24(1), 1-11.

안재근, 김우제, 박순달(1993), 단체법에서의 규모화와 허용오차, 전산활용연구, 6(1), 29-39.

Andersen, E. D. (1995), Finding all Linearly Dependent Rows in Large-Scale Linear Programming, *Optimization Methods and Software*, 6, 219-227.

Andersen, E., Gondzio, D. J. , Mészáros, C., Xu, X. (1996), Implementation of Interior Point Methods for Large Scale Linear Programming, in: Interior Point Methods in Mathematical Programming, T. Terlaky (ed.), Chapter 6, 189-252, Kluwer Academic Publisher.

Avriel, M. (1976), Nonlinear Programming: Analysis and Methods, Prentice-Hall.

Duff. I., Erisman, A. and Reid, J. (1986), Direct Methods for Sparse Matrices, Oxford University Press, New York.

George, A. and Liu, J. W. H. (1981), Computer Solution of Large Sparse Positive Definite Systems, Prentice-Hall.

Golub, G. H. and Van Loan, C. F., (1983), Matrix Computations, North Oxford Academic, Baltimore.

Gondzio, J. (1997), Presolve Analysis of Linear Programs Prior to Applying an

Interior Point Method, *INFORMS Journal on Computing*, 9(1), 73-91.

Jung, H. W., Marsten, R. E. and Salzman, M. J. (1994), Numerical Factorization Methods for Interior-Point Algorithms, *ORSA Journal on Computing*, 6(1) 94-105, .

Lustig, I. J., Marsten, R. E. and Shanno,D. F. (1992), On Implementing Mehrotra's Predictor-Corrector Interior-Point Method for Linear Programming, *SIAM Journal on Optimization*, 2(3), 435-449.

Lustig, I. J., Marsten, R. E. and Shanno, D.F. (1994), Interior Point Methods for Linear Programming: Computational State of the Art, *ORSA Journal on Computing*, 6(1), 1-15.

Mészáros, C. (1998), On a Property of the Cholesky Factorization and Its Consequences in Interior Point Methods, Technical Report, Laboratory of Operations Research and Decision Systems, Hungarian Academy of Sciences.

Wilkinson, J. H. (1961), Error Analysis of Direct Methods of Matrix Inversion, *Journal of ACM*, 8, 281-330.

Wright, S. (1998), Modified Cholesky Factorizations in Interior-Point Algorithms for Linear Programming, Technical Report, Mathematics and Computer Science Division, Argonne National Laboratory, Agonne, IL, USA.