고뇌의 고통이 없는 켤레 경사도 기법 개론 ($1\frac{1}{4}$판)

강영민[†], 박동규[‡], 유영중[††], 이도훈[‡‡] 譯

[†]동명대학교, [‡]창원대학교, [††]부산외국어대학교, [‡‡]부산대학교

요약

켤레 경사도(conjugate gradient) 기법은 희소행렬로 표현되는 선형 시스템의 해를 구하는 데에 가장 뛰어난 반복법이다. 그러나 불행하게도 많은 책들이 이 주제를 그림이나 직관적 설명도 없이 다루고 있다. 이런 책들의 희생자들은 지금도 먼지가 풀풀 나는 도서관 구석에서 의미없는 말들을 뱉어내고 있다. 이러한 이유로 이 기법에 대한 심도 있는 기하적 이해는 옛날 학자들의 우물거림을 고통스럽게 해독한 일부 명석한 엘리트들의 전유물이었다. 하지만 켤레 경사도 방법이라는 것은 간단하면서 누구나 이해할 수 있는 우아한 사고로 이루어져 있다. 여러분들처럼 똑똑한 독자들이라면 당연히 아무런 고통도 없이 배울 수 있는 것이다.

2차형식의 개념을 소개한 뒤, 이를 이용하여 최대하강(steepest descent), 켤레 방향(conjugate directions), 켤레 경사도(conjugate gradient) 기법을 유도할 것이다. 고유벡터는 야코비(Jacobi) 기법, 최대하강, 켤레 경사도 기법 등의 수렴을 설명하는데 사용될 것이다. 그리고 전처리(preconditioning)과 비선형 켤레 경사도 방법 등이 포함되어 있다. 필자는 여러분이 이 글을 쉽게 읽게 하려고 많은 노력을 하였다. 66개의 그림을 제공되며, 빽빽한 산문형태의 설명은 피했다. 개념들은 몇 가지 서로 다른 방법으로 설명되고 있으며 대부분의 방정식은 직관적인 설명과 함께 제공된다.[1]

[1] *원저자 - Jonathan Richard Shewchuk. 원문제목 - An Introduction to the Conjugate Gradient Method without the Agonizing Pain. 원 논문의 연구는 캐나다 이공 연구 협회(Natrual Science and Engineering Research Council of Canada)의 1967년 이공 장학사업과 미 과학 재단(National Science Foundation)의 지원(Grant ASC-9318163)을 받아 이루어졌다. 이 문서의 관점과 결론은 저자의 견해이며 캐나다 이공 연구 협회(NSERC), 미 과학재단(NSF), 혹은 미국 정부의 견해로 받아들여져서는 안된다.*

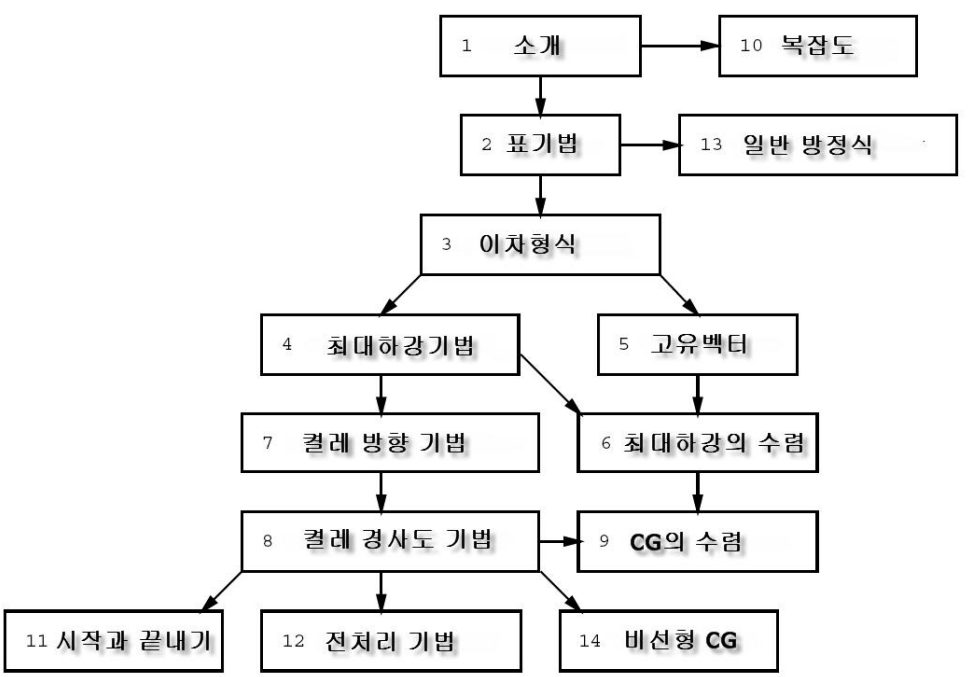
이 문서에 대해

이 문서는 켤레 경사도 기법을 가능한 쉽게 공부하고 싶은 학생들을 돕기 위하여 만들어졌다. 나에게 지적사항, 수정해야할 내용, 그리고 내가 빠뜨린 어떤 종류의 직관적 설명이라도 좋으니 알려 주기 바란다 (jrs@cs.cmu.edu); 이들 가운데 일부는 두 번째 판에 포함이 될 것이다. 나는 특히 이 문서를 수업시간에 활용한 사례에 관심이 많다.

반복 기법에 대해 더 많은 것을 공부하고자 하는 사람들에게 나는 윌리엄 브릭스(William L. Briggs)의 "A multigrid Tutorial" [2]을 추천한다. 이 책은 지금까지 내가 읽은 수학 서적 가운데 가장 잘 쓰여진 책이다.

수치 기법에 관해 내가 알고 있는 것의 상당부분을 가르쳐주고 이 문서의 초고에 깊이 있는 비평을 제공해 주었던 오마르 가타스(Omar Ghattas)에게 특별한 감사를 드린다. 비평을 해 주었던 제임스 에퍼슨(James Epperson), 데이비드 오핼러런(David O'Hallaron), 제임스 스티치노스(James Stichnoth), 닉 트레페텐(Nick Trefethen), 그리고 다니엘 툰켈랑(Daniel Tunkelang)에게도 감사한다.

필요한 부분만을 읽을 수 있도록 각 절의 상호의존 그래프를 보인다:



1 들어가기

필자가 켤레 경사도(conjugate gradient method)을 배우려고 했을 때, 4편의 서로 다른 문헌을 읽었는데, 이 책들이 어떤 책들인지는 예의상 밝히지 않겠다. 필자는 이 책들을 하나도 이해하지 못했다. 이 책들은 대부분 간단하게 켤레 경사도를 기술했으며, 어떠한 직관적 설명도 없었고 어떻게 켤레 경사도 기법이 처음에 고안되었는지에 대한 힌트(hint)도 없이 그 기법의 특성을 설명하고 있었다. 이 글은 이런 필자의 좌절 덕분에 탄생하게 되었으며, 장차 켤레 경사도 기법을 배우려는 학생들이 혼란스러운 방정식 덩어리가 아니라 풍부하고 우아한 알고리즘을 배우기를 바라면서 쓰여졌다.

켤레 경사도 기법은 선형방정식의 거대한 시스템을 풀기 위해 가장 많이 사용되는 반복법이다. 켤레 경사도 기법은 다음 형태의 시스템에 아주 효과적이다.

$$Ax = b \tag{1}$$

이때 x는 모르는 벡터이고, b는 알고 있는 벡터이며 A는 알고 있는 정방(square), 대칭(symmetric), 양의 정부호(positive-define) 행렬이다 - 혹은 양의 부정부호(positive-indefinite) 행렬일 수도 있다. "양의 정부호"가 무엇을 의미하는지 잊어버렸다고 할지라도 나중에 다시 살펴볼 것이니 걱정할 필요는 없다. 이 시스템은 편미분 방정식, 구조 분석, 회로 분석, 수학 숙제 등을 해결할 때 종종 나타나는 유한차분법(finite difference)과 유한 요소법(finite element method) 등과 같은 중요한 풀이법에서 자주 나타난다.

켤레 경사도 기법과 같은 반복 방법은 희소 행렬을 사용할 때에 적합하다. A가 조밀한 경우에 가장 좋은 방법은 A를 분해하고 역치환(backsubstitution)에 의해 방정식을 풀는 것이다. 조밀한 A을 분해하는데 걸리는 시간은 대충 시스템을 반복해서 해결하는데 걸리는 시간과 비슷하다. 그리고 한번 A가 분해되면 시스템은 b의 여러 값들을 위해서 빠르게 꺼꾸로 풀 수 있다. 이 조밀한 행렬을 같은 메모리 크기를 가지는 더 큰 크기의 희소 행렬과 비교해보자. 희소행렬 A의 삼각요소(triangular factors)에는 0이 아닌 원소가 일반적으로 원래의 A보다 훨씬 더 많다. 분해 자체가 메모리의 한계 때문에 불가능할 수도 있으며 시간이 많이 걸릴 수도 있다. 심지어 역치환을 과정이 반복기법보다 더 느릴 수도 있다. 반면에 대부분의 반복적 방법은 메모리 효율이 높고 희소행렬에 대해 빠르게 동작한다.

필자는 여러분이 선형대수학을 이수했다고 가정하며, 고유벡터니 고유값과 같은 것에 대해서는 잘 모른다고 할지라도 행렬 곱셈과 선형 독립 등에 대해서는 잘 알고 있다고 가정한다. 이런 기본적인 가정을 바탕으로 필자는 켤레 경사도 기법의 체계를 가능한한 명쾌하게 구축해 나갈 것이다.

2 표기법

몇가지 정의와 표기에 대해 살펴보면서 시작하자.

행렬을 표시하기 위해 대문자를 사용할 것이며, 소문자는 벡터를 나타내는데 사용된다. 스칼라는 그리스 문자로 표시한다. A는 $n \times n$ 행렬이고 x와 b는 벡터, 즉, $n \times 1$ 행렬이다. 수식 1은 다음과 같이 표현한다.

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & & A_{2n} \\ \vdots & & \ddots & \vdots \\ A_{n1} & A_{n1} & \cdots & A_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

두 벡터의 *내적(inner product)*은 $x^T y$이고 스칼라 합 $\sum_{i=1}^{n} x_i y_i$을 나타낸다. $x^T y = y^T x$이다. x와 y가 직교하면 $x^T y = 0$이다. 일반적으로 $x^T y$와 $y^T Ax$와 같이 1×1 행렬이 되는 표현식은 스칼라 값으로 취급된다.

만약 모든 0이 아닌 벡터 x에 대해 다음 부등식을 만족할 때, 행렬 A가 *양의 정부호(positive-definite)*이다.

$$x^T Ax > 0 \tag{2}$$

여기서 아무런 의미를 느끼지 못한다고 실망하지 말라. 이 개념은 그다지 직관적인 개념이 아니어서 양의 정부호인 행렬이 그렇지 않은 것과 어떻게 다르게 보이는지 이해하는 것이 쉬운 일이 아닌다. 이를 이해하기 위해서는 양의 정부호(positive-definite)라는 것이 2차 형식에 어떻게 영향을 미치는지 살펴볼 필요가 있다.

마지막으로 $(AB)^T = B^T A^T$과 $(AB)-1 = B^{-1}A^{-1}$라는 중요한 기본 항등식을 잊지말고 기억하라.

3 2차형식

2차 형식(quadratic form)은 간단히 말해 다음과 같은 형태의 스칼라 값을 갖는 2차 함수이다.

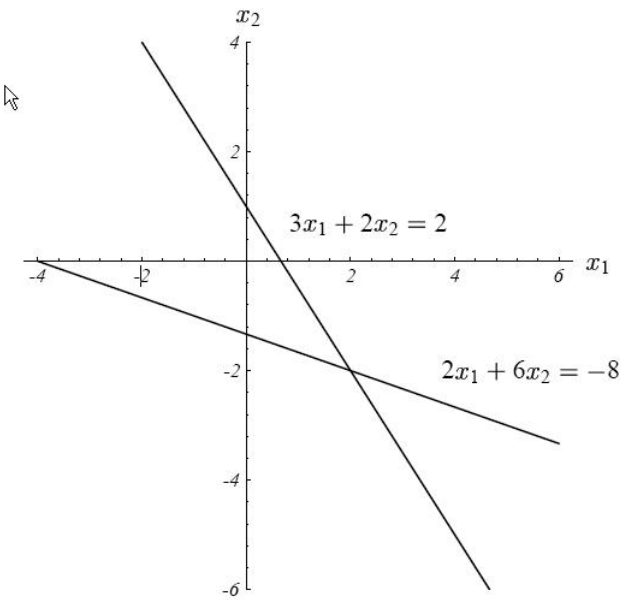
$$f(x) = \frac{1}{2} x^T Ax - b^T x + c \tag{3}$$

A는 행렬이고 x와 b는 벡터이다. 그리고 c는 스칼라 상수이다. 필자는 여기서 A가 대칭적이고(symmetric) 양의 정부호인 경우, $Ax = b$의 해를 이용하여 $f(x)$를 최소화할 수 있음을 간단히 보이려 한다.

이 글의 전체에서 필자는 다음과 같이 간단한 실제 예제 문제를 가지고 다양한 개념을 설명할 것이다.

$$A = \begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ -8 \end{bmatrix}, \quad c = 0. \tag{4}$$

시스템 $Ax = b$가 그림 1에 나타나 있다. 일반적으로 해 x는 n 개의 초평면(超平面)의 교차점이다 (이때 각각의 초평면은 각각 $n-1$ 차원을 가진다). 이 문제의 해는 $x = [2, -2]^T$이다. 대응되는 이차식 $f(x)$는 그림 2에 나타난다. $f(x)$의 등고선의 모양은 그림 3과 같다. A가 양의 정부호이기 때문에 $f(x)$에 의해 정의되는 표면은 포물선형태의 사발과 같은 모양이다. (이에 대해 잠시 뒤에 더 자세히 설명하겠다)



[그림 1] 2차원 선형시스템 예. 해는 선분들의 교차점이다

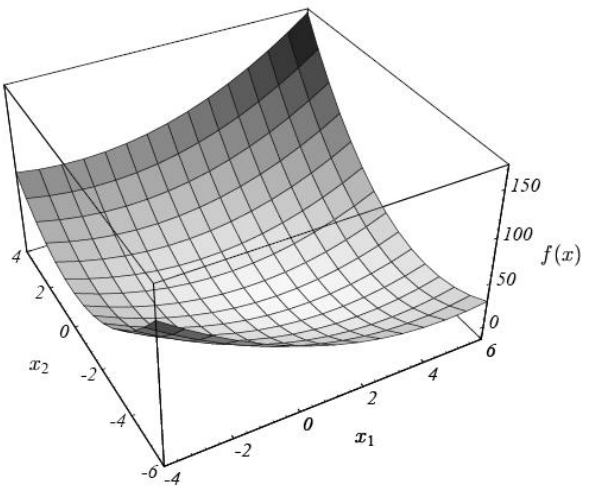
2차식의 *기울기(gradient)*는 다음과 같이 정의된다.

$$f'(x) = \begin{bmatrix} \frac{\partial}{\partial x_1} f(x) \\ \frac{\partial}{\partial x_2} f(x) \\ \vdots \\ \frac{\partial}{\partial x_n} f(x) \end{bmatrix}. \tag{5}$$

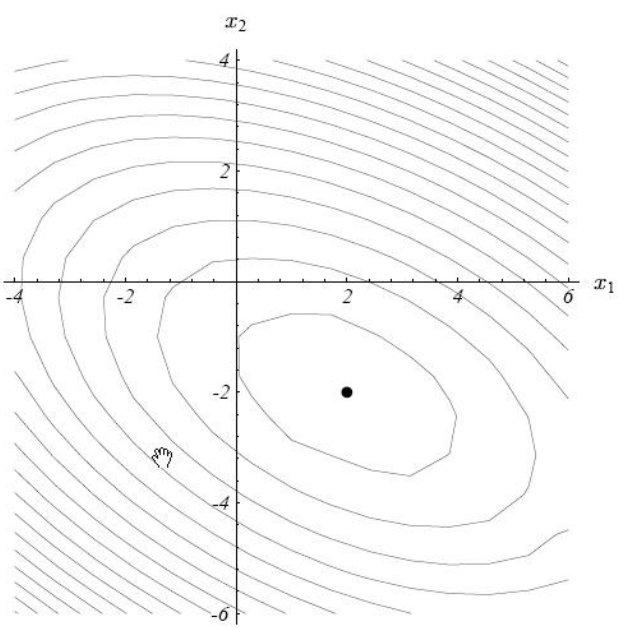
기울기는 주어진 점 x에 대해, $f(x)$의 가장 큰 증가방향을 가르키는 벡터필드이다. 그림 4은 수식 4 과 같은 상수를 수식 3에 대입하여 얻어진 기울기 벡터를 설명하고 있다. 사발모양의 포물면 바닥에서는 기울기가 0이다. $f'(x)$에 0이 되도록 함으로써 $f(x)$은 최소화할 수 있다.

다소 지루한 작업을 통해 수식 5를 수식 3에 대입하여 다음을 유도할 수 있다.

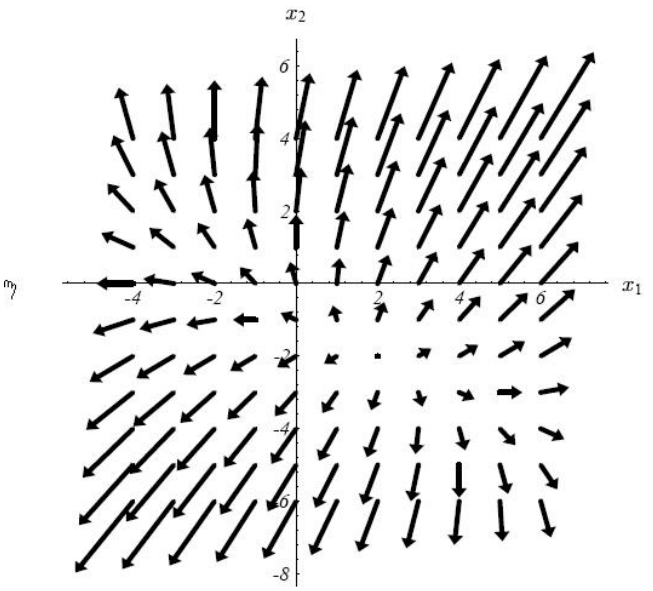
$$f'(x) = \frac{1}{2}A^T x + \frac{1}{2}Ax - b. \tag{6}$$



[그림 2] 2차형식 $f(x)$의 그래프. 이 표면의 최저점은 $Ax = b$의 해이다.



[그림 3] 2차형식의 등고선. 각 타원 곡선은 동일한 $f(x)$ 값을 갖는다.



[그림 4] 2차형식의 기울기 $f'(x)$. 모든 x에 대하여, 기울기 벡터는 $f(x)$가 가장 크게 상승하는 방향을 가리키며 등고선에 직교한다.

A가 대칭이면 이 방정식은 다음과 같이 간략화된다.

$$f'(x) = Ax - b. \tag{7}$$

기울기를 0으로 하면 우리가 얻고자 하는 선형시스템 수식 1을 얻는다. 그러므로 $Ax = b$의 해는 $f(x)$의 임계점(critical point)이다. A가 대칭일 뿐만 아니라 양의 정부호(positive-definite)라면 해는 $f(x)$의 최소값이다. 따라서 $Ax = b$의 해는 $f(x)$을 최소화하는 x을 찾음으로써 구할 수 있다. (A가 대칭적이지 않으면 수식 6는 켤레 경사도 기법이 시스템 $\frac{1}{2}(A^T + A)x = b$의 해를 찾게된다는 것을 보여주고 있다. 이때 $\frac{1}{2}(A^T + A)$는 대칭 행렬이다.)

대칭인 양의 정부호 행렬은 왜 이렇게 좋은 성질을 갖고 있을까? 어떤 임의의 점 p에서의 f와 선형시스템의 해 $x = A^{-1}b$ 사이의 관계를 살펴보자. 수식 3을 바탕으로 A가 대칭인 경우 (이 행렬이 양의 정부호(positivie-definite)인지에 관계 없이) 다음이 성립함을 보일 수 있다. (부록 C1).

$$f(p) = f(x) + \frac{1}{2}(p - x)^T A(p - x). \tag{8}$$

A가 양의 정부호(positive-definite)라고 하면, 부등식 2에 의해 후자의 항이 모든 $p \neq x$에 대해 양수이다. 이는 x가 f의 전역 최소값임을 뜻한다.

양의 정부호(positive-definite) 행렬에 관한 가장 직관적인 이해는 이 행렬의 2차 형식 $f(x)$가 포물면이된다는 것이다. A가 만약 양의 정부호가 아니라면 다른 가능성이 있다. A는 음의 정부호가 될 수 있는데 이는 양의 정부호 행렬에 대해 부호를 바꾼 결과, 즉 그림 2의 아래위를 뒤집어 놓은 모양이 된다. A가 특이행렬인 경우에는 유일한 해가 없는 경우이다; 해의 집합은 하나의 직선, 혹은 일정한 f 값을 가지는 초평면(hyperplane)이 된다. A가 위에서 설명한 어떤 경우도 아니라면, x는 안장점(saddle point)이 되며 최대하강이나 켤레 경사도 기법 등이 실패할 것이다. 그림 5는 이러한 가능성들을 보여주고 있다. b와 c의 값은 포물형의 최소점이 어디인지를 결정한다. 그러나 포물형 모양에는 영향을 미치지 않는다.

선형시스템을 왜 더 어려워 보이는 문제로 변형할까? 최대하강과 켤레 경사도 기법 등의 방법들은 그림 1와 같이 초평면의 교차를 통해서가 아니라 그림 2와 같이 함수의 최소화를 통해 직관적인 이해가 가능하기 때문이다.

4 최대하강 방법

최대하강 방법에서는, 어떤 임의 점 $x_{(0)}$에서 시작하여 포물선의 밑바닥으로 미끄러져 내려간다. 해 x에 충분히 가까게 되는 조건을 만족할 때까지 각 단계 $x_{(0)}, x_{(1)} \cdots$을 취한다.

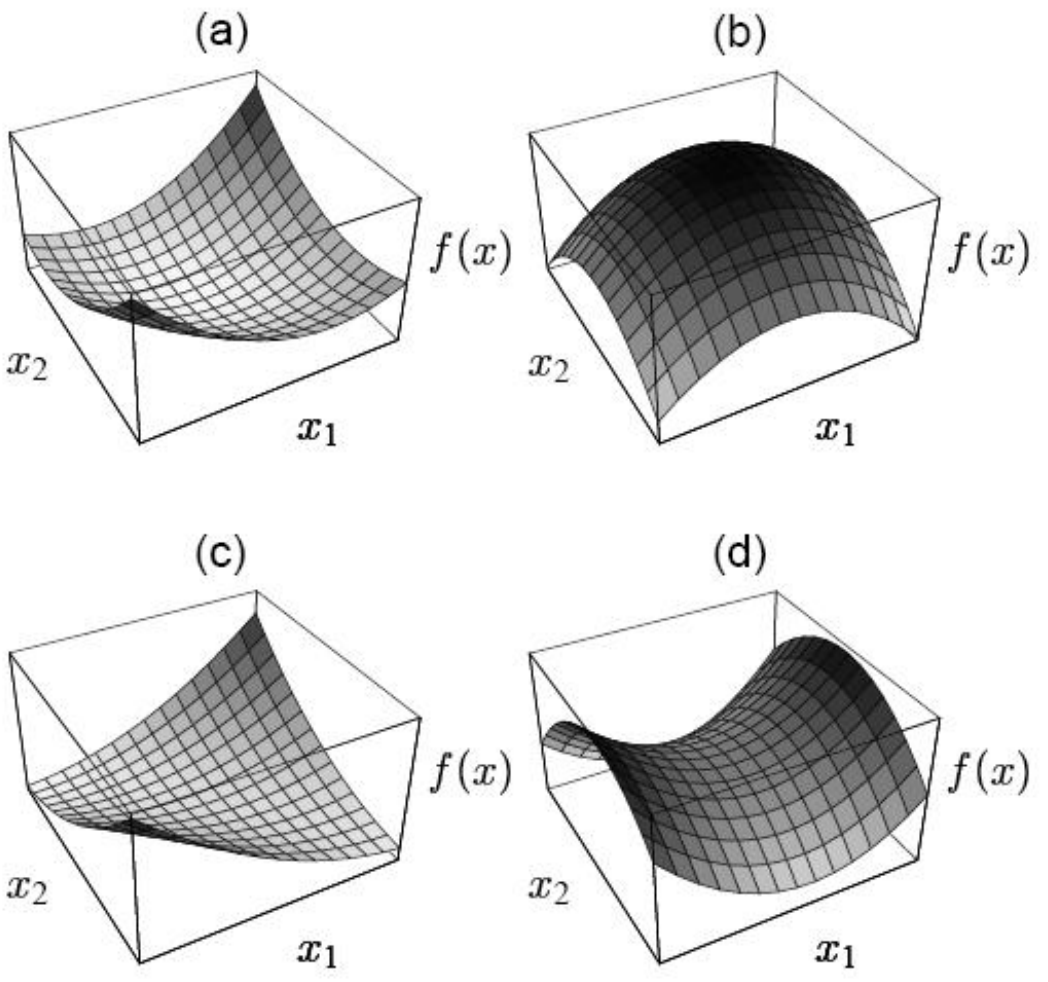
하나의 단계에서 f가 가장 빠르게 감소하는 방향을 선택한다. 이 방향은 $f'(x_{(i)})$의 반대방향이다. 수식 7에 따르면, 이 방향은 $-f'(x_{(i)}) = b - Ax_{(i)}$이다.

여기서 독자들이 기억해야 할 몇가지 정의를 소개한다. *오차(error)* $e_{(i)} = x_{(i)} - x$는 해로부터 얼마나 멀리 떨어져 있나를 나타내는 벡터이다. 나머지(*residual*) $r_{(i)} = b - Ax_{(i)}$은 정확한 b 값으로 부터 얼마나 차이가 있나를 나타낸다. $r_{(i)} = -Ae_{(i)}$임은 쉽게 알 수 있고, 이 나머지는 결국 오차를 행렬 A를 이용하여 b와 같은 공간으로 변환한 것으로 볼 수 있다. 더 중요한 것은 $r_{(i)} = -f'(x_{(i)})$이라는 것이며, 나머지(residula)를 최대하강 기법의 진행방향과 같은 것으로 생각하여야 한다. 비선형문제에서는 단지 후자의 정의을 적용하여 14절에서 논의한다. 그러므로 "나머지(residual)"이라는 용어를 읽게 되면 "최대 하강 방향"이라는 의미를 떠올리도록 기억해 두라.

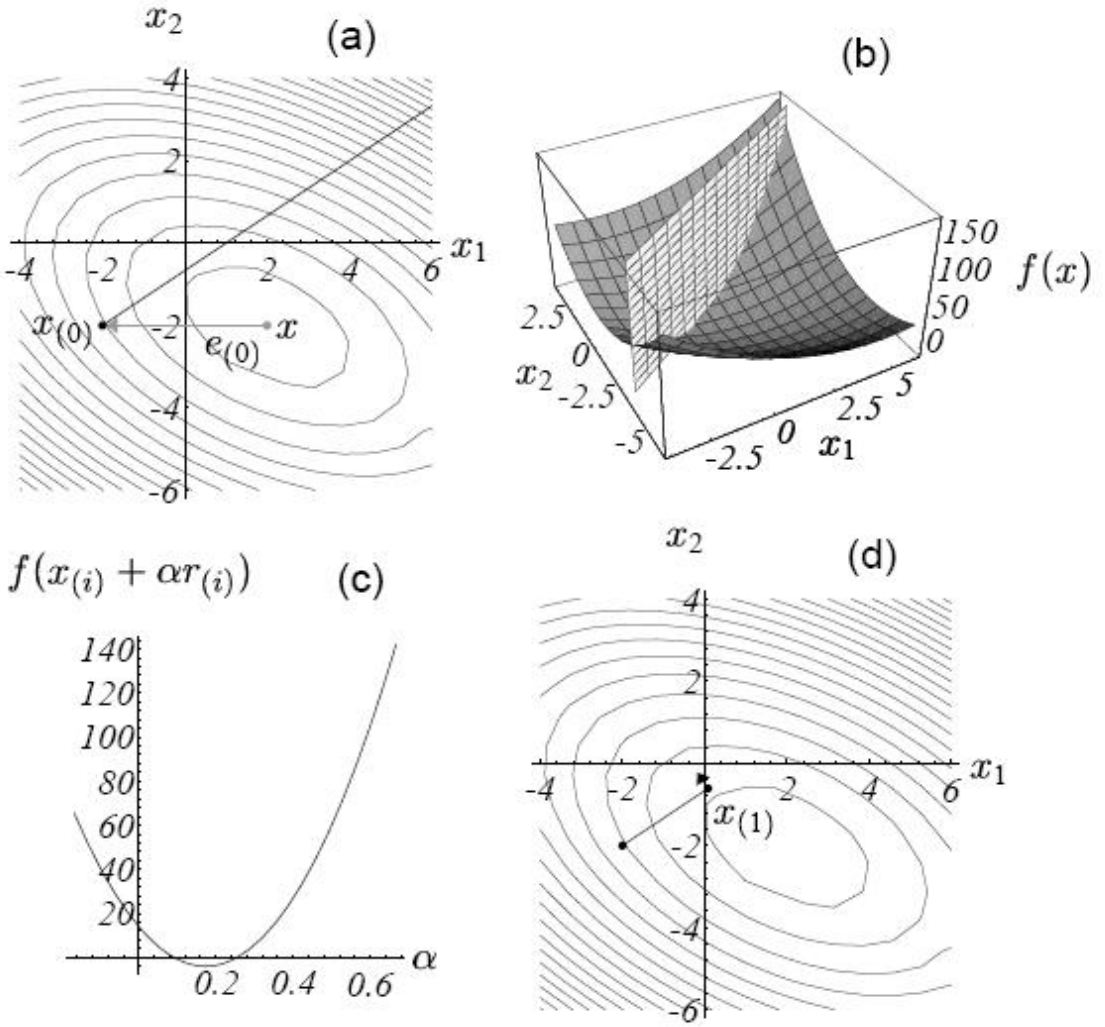
$x_{(0)} = [-2, -2]^T$에서 출발한다고 가정하자. 처음 수행하는 작업은 최대하강(steepest descent) 방향을 따라서 그림 6 (a)에 있는 실선상의 어디에 떨어질 것이다. 즉, 다음과 같이 새로운 어떤 점을 취할 것이다.

$$x_{(1)} = x_{(0)} + \alpha r_{(0)}. \tag{9}$$

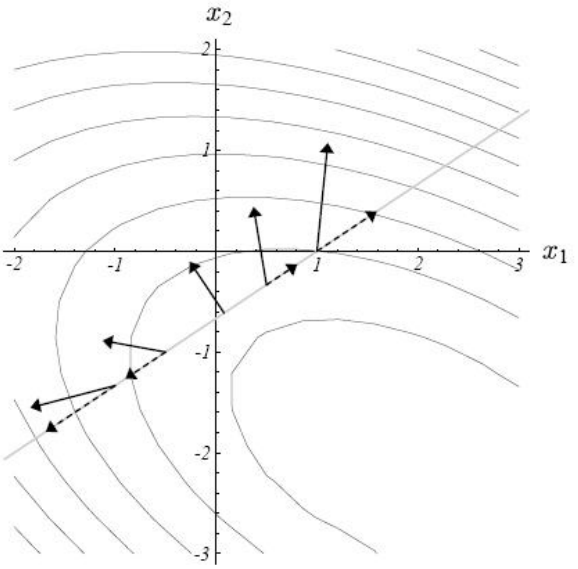
문제는 이 방향을 따라 얼마나 진행한 곳을 선택할 것인가이다. (즉, 어떤 α를 선택할 것인가를 결정하는 것이다.)



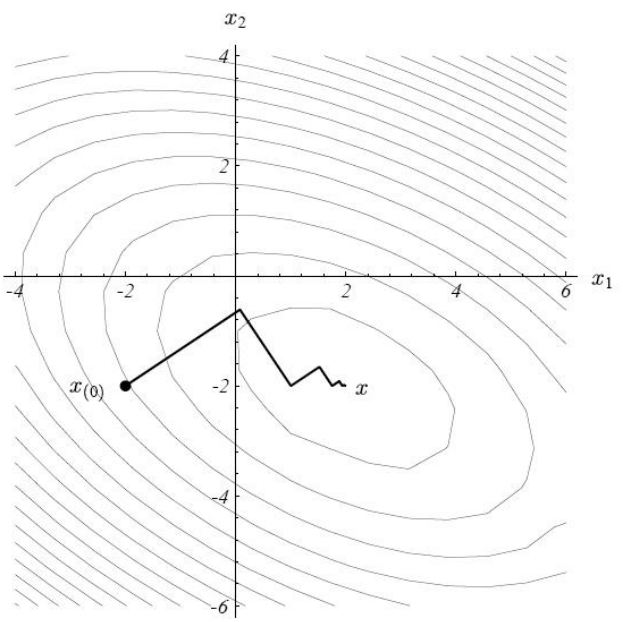
[그림 5] (a) 양의 정부호 행렬을 위한 2차식. (b) 음의 정부호 행렬을 위한 2차식. (c) 특이행렬(그리고 양의 부정부호). 계곡의 바닥을 지나는 선분이 해의 집합이다. (d) 부정부호 행렬일 때. 해가 안장 점이기 때문에 최대하강과 켤레 경사도 기법이 적용되지 않는다. 3차원이나 그 이상에서는 특이행렬 역시 안장모양을 가질 수 있다.



[그림 6] 최대하강 방법. (a)$x_{(0)} = [-2, -2]^T$에서 출발해서 f의 최대하강의 방향으로 한 단계를 취한다. (b) 두 면의 교차선상에 있는 점들 중에서 f를 최소화하는 점을 찾는다. (c) 이 포물선은 두 면(2차형식 포물면과 탐색선 방향의 수직 평면)이 교차하는 선이다. (d) 최저점에서의 기울기는 이전 단계의 기울기와 직교한다.



[그림 7] 탐색 직선(실선 화살표)을 따라 여러곳에서 기울기 f'를 보여주고 있다. 선분위로 각 기울기의 투영 역시 보이고 있다(점선 화살표). 기울기 벡터는 f의 기울기의 증가 방향을 나타내고 투영은 탐색선을 따라 갈 때의 증가율을 나타낸다. 탐색선 상에서 기울기가 탐색선과 직교하는 곳에서 f가 최소값을 갖는다.



[그림 8] 여기, 최대하강의 방법은 $[-2,-2]^T$ 에서 출발하고 $[2,-2]^T$에서 수렴한다.

*직선탐색(line search)*은 직선을 따라 f을 최소화하는 α를 선택하는 과정이다. 그림 6 (b)는 이 과정을 설명한다. 수직평면과 포물면이 만나는 교차선 위의 한 점을 선택하도록 제약한다. 그림 6 (c)는 이 두 평면의 교차에 의해 정의된 포물선이다. 포물선의 최저점에서 α값은 얼마인가?

기본적인 미적분을 통해, α는 *방향 도함수(directional derivative)* $\frac{d}{d\alpha}f(x_{(1)})$가 0일때 f를 최소화한다. 연쇄율(chain rule)에 의해, $\frac{d}{d\alpha}f(x_{(1)}) = f'(x_{(1)})^T\frac{d}{d\alpha}x_{(1)} = f'(x_{(1)})^T r_{(0)}$ 이다. 이 수식을 0으로 하고, $r_{(0)}$와 $f'(x_{(1)})$들이 직교되게 하는 α가 선택되어야 한다(그림 6 (d)를 보라).

이 두 벡터가 최소점에서 서로 직교가 되어야 하는 직관적인 이유가 있다. 그림 7 는 탐색선을 따라서 다양한 점들에서 기울기 벡터를 보여주고 있다. 포물선(그림 6 (c)) 위의 임의의 점에서 가지는 포물선 기울기 값(slope)는 그 점에서의 2차 형식이 가지는 기울기(gradient)를 탐색 직선 위에 투영했을 때 (그림 7, 점선 화살표) 생기는 벡터의 길이와 같다. 투영된 이 벡터들은 탐색선을 따라갈 때 f의 증가률을 표현한다. f는 투영된 기울기 벡터의 길이가 0인 곳-기울기가 탐색선에 직교하는 곳-에서 최소값을 갖는다.

α을 결정하기 위해, $f'(x_{(1)}) = -r_{(1)}$임에 유의하면 다음을 얻을 수 있다.

$$\begin{aligned}
r_{(1)}^T r_{(0)} &= 0 \\
(b - Ax_{(1)})^T r_{(0)} &= 0 \\
(b - A(x_{(0)} + \alpha r_{(0)}))^T r_{(0)} &= 0 \\
(b - A(x_{(0)})^T r_{(0)} - \alpha(Ar_{(0)})^T r_{(0)} &= 0 \\
(b - Ax_{(0)})^T r_{(0)} &= \alpha(Ar_{(0)})^T r_{(0)} \\
r_{(0)}^T r_{(0)} &= \alpha r_{(0)}^T (Ar_{(0)}) \\
\alpha &= \frac{r_{(0)}^T r_{(0)}}{r_{(0)}^T Ar_{(0)}}
\end{aligned}$$

이를 모두 대입해 보면, 최대 하강(Steepest Descent) 기법은 다음과 같다.

$$r_{(i)} = b - Ax_{(i)}, \tag{10}$$

$$\alpha_{(i)} = \frac{r_{(i)}^T r_{(i)}}{r_{(i)}^T Ar_{(i)}}, \tag{11}$$

$$x_{(i+1)} = x_{(i)} + \alpha_{(i)} r_{(i)}. \tag{12}$$

이 예제는 그림 8에서와 같이 수렴할 때 까지 수행된다. 지그재그 경로임에 유의하자. 이는 각각의 기울기가 이전의 기울기에 대해 직교하기 때문에 나타나는 현상이다.

위에 언급한 알고리즘은 한 번의 반복 때마다 두개의 행렬-벡터 곱셈을 요구한다. 최대 하강(Steepest Descent) 기법의 계산 비용은 행렬-벡터 곱에 의해 좌우되는데, 다행히도 이 가운데 하나는 없앨 수가 있다.

수식 12의 양변의 앞에 $-A$을 곱하고 b을 더하면 다음을 얻는다.

$$r_{(i+1)} = r_{(i)} - \alpha A r_{(i)}. \tag{13}$$

$r_{(0)}$을 계산하기 위해 수식 10을 한 번 계산하는 것은 반드시 필요하지만, 그 이후의 반복에서는 수식 13를 계속해서 사용할 수 있다. 수식 11과 수식 13 모두에서 나타나는 행렬-벡터 곱 Ar은 한 번만 계산하면 된다. 이 식만을 이용하여 반복을 할 때 생기는 문제점은 수식 13을 통해 정의되는 수열이 $x_{(i)}$ 값으로부터 어떤 되먹임(feedback)도 없이 생성된다는 점이다. 따라서 부동소수점 반올림 오류의 누적으로 인한 정확한 x가 아니라 그 근처의 어떤 점에 수렴할 수 있다는 것이다. 이러한 단점은 수식 10을 이용하여 주기적으로 정확한 나머지(residual)를 다시 계산함으로써 해결할 수 있다.

최대하강(Steepest Descent) 기법의 수렴에 대한 분석 이전에, 여러분들이 고유벡터에 대한 확실한 이해를 할 수 있도록 잠깐 옆길로 빠지려고 한다.

5 고유벡터(eigenvector)와 고유값(eigenvalue)에 대한 고찰

필자가 선형대수학 한 과목을 수강한 후에, 고유벡터(eigenvector)와 고유값(eigenvalue)이 무엇인지 도무지 알 수가 없었다. 여러분들을 가르친 강사가 필자를 가르친 강사와 비슷하다면, 고유값인지 뭔지하는 것을 가지고 문제를 푸는 방법은 떠오르겠지만 결코 그것들이 도대체 무엇인지 "진정으로" 이해하고 있진 않을 것이다. 불행하게도 이 고유벡터와 고유값에 대한 직관적인 이해가 없다면, 켤레 경사도 기법 역시 이해하지 못할 것이다. 여러분이 만약 고유벡터니 고유값이니 하는 것들에 대해 타고난 재능을 가지고 있다면 이 절을 건너 뛰어도 된다.

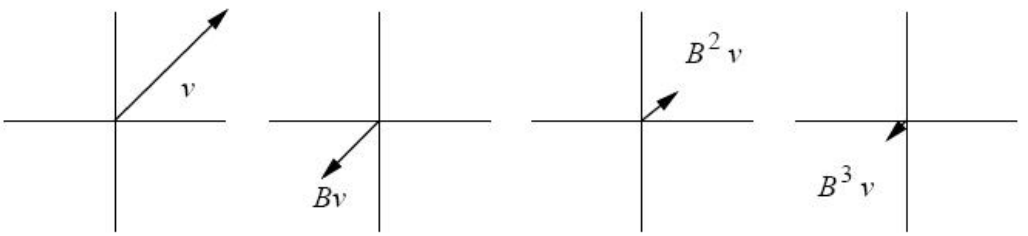
고유벡터들은 기본적으로 분석 도구로서 사용된다; 최대 하강(Steepest Descent) 기법이나 켤레 경사도 기법은 알고리즘을 수행하기 위해 고유벡터를 계산할 필요는 없다.

5.1 고유값이 뭔고유?

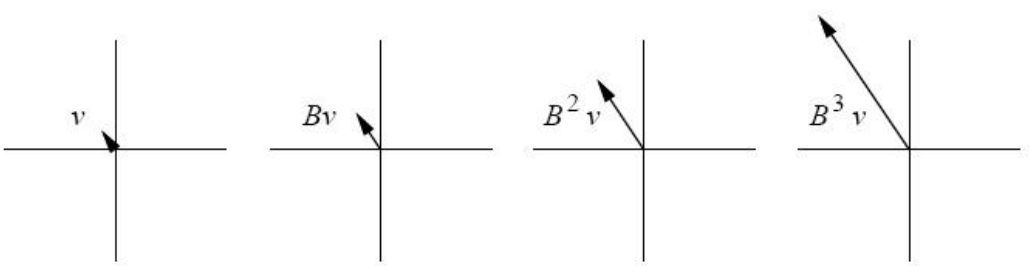
행렬 B의 *고유벡터* v는 B를 적용하여 변환을 하여도 방향의 변화가 일어나지 않는 0이 아닌 벡터이다(정확하게 반대 방향으로 변환되는 것은 제외). v는 길이가 변하거나 그 방향이 반대가 될 수는 있지만, 옆으로 돌지

는 않을 것이다. 다시 말해, $Bv = \lambda v$인 어떤 스칼라 상수 λ가 존재한다. 이때 λ 값이 B의 고유값이다. 임의의 상수 α에 대해, 벡터 αv도 역시 동일한 고유값 λ를 갖는 고유벡터이다. 이는 $B(\alpha v) = \alpha Bv = \lambda \alpha v$이기 때문이다. 다시 말해, 고유벡터를 크기를 늘이거나 줄여도 여전히 고유벡터이다.

이걸 왜 신경써야 할까? 반복법은 종종 B를 어떤 벡터에 계속해서 반복 적용하여 해를 구한다. B가 하나의 고유벡터에 반복적으로 적용될 때, 다음 두 가지 가운데 하나의 상황이 발생할 수 있다. $|\lambda| < 1$인 경우라면, i가 무한대로 갈 때 $B^i v = \lambda^i v$는 0 벡터로 수렴할 것이다(그림 9). $|\lambda| > 1$ 이면, $B^i v$는 무한대로 커질 것이다(10). 매 번 B가 적용될 때마다, 벡터는 $|\lambda|$의 값에 따라 커지거나 작아지게 된다.



[그림 9] v는 -0.5의 고유값에 대응되는 B의 고유벡터. i가 증가하면, $B^i v$는 0에 수렴한다.



[그림 10] 이 예에서 v는 고유값 2를 가지는 고유벡터이다. i가 증가하면, $B^i v$는 무한대로 발산한다.

B가 대칭행렬인 경우(종종 그렇지 않은 경우에도 가능하다), B는 선형적 독립인 n 개의 고유벡터를 가진다. 이를 $v_1, v_2, \cdots, v_n$이라 표기하자. 이 집합은 유일하지 않다. 이는 각 고유벡터가 0이 아닌 적당한 상수만큼 크기가 변경될 수 있기 때문이다. 각각의 고유벡터는 자신에게 대응되는 고유값을 가지는데, 이를 $\lambda_1, \lambda_2, \cdots, \lambda_n$로 표시하자. 이 고유값들은 주어진 행렬에 대해 유일하게 정의된다. 고유값은 서로 같은 값을 가질 수도 있고 서로 다를 수도 있다. 예를 들어, 항등행렬 I의 고유값은 모두 1이고, 0 벡터가 아닌 임의의 벡터가 I의 고유벡터이다.

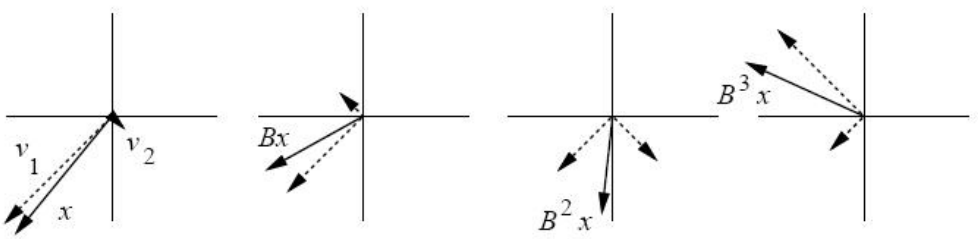
B가 고유벡터가 아닌 벡터에 적용되면 어떤 일이 일어날까? 선형대수을 이해함에 있어 매우 중요한 기술-이 절에서 가르치고자 하는 그 기술-은 하나의 벡터를 행동이 알려진 다른 벡터의 합으로 간주하는 것이다. 고유벡터들의 집합 $\{v_i\}$가 $\mathbb{R}^n$의 기저를 이루는 상황을 고려해 보자(대칭행렬 B는 선형 독립인 n개의 고유벡

터들을 가진다). 임의의 n-차원 벡터는 이들 고유벡터들의 선형 조합으로 표현될 수 있다. 또한 행렬과 벡터의 곱은 분배법칙이 적용되므로 각각의 고유벡터에 곱해지는 B의 효과는 독립적으로 조사할 수 있다.

그림 11에서 벡터 x는 두 고유벡터 v_1과 v_2의 합으로 나타나 있다. B를 x에 곱하는 것은 B를 고유벡터들에 각각 곱하여 합을 구하는 것과 같다. B를 반복해서 곱해 나가면, $B^i x = B^i v_1 + B^i v_2 = \lambda_2^i v_2$가 된다. 모든 고유값의 크기가 1보다 작으면, $B^i x$는 0에 수렴을 할 것이다. (이는 x을 구성하고 있는 고유벡터들 모두가 B를 반복적으로 곱할 경우 0에 수렴하기 때문이다). 고유값들 중 하나라도 1보다 큰 값을 갖는다면, x는 무한대에 발산할 것이다. 이것이 바로 수치해석 학자들이 행렬의 스펙트럼 반경(spectral radius)에 중요성을 부여하는 이유이다. 행렬의 스펙트럼 반경은 다음과 같다.

$$\rho(B) = \max|\lambda_i|, \lambda_i \text{ 은 } B\text{의 고유벡터}$$

x가 0에 빠르게 수렴하기를 원하면 $\rho(B)$은 1보다 작아야 하고, 가능하면 작은 것이 좋다.



[그림 11] 벡터 v(실선 화살표)는 고유벡터들(점선 화살표)의 선형 조합으로 표현될 수 있다. 이와 연관된 고유값이 $\lambda_1 = 0.7$과 $\lambda_2 = -2$ 이다. B가 반복적으로 적용되면 한 고유벡터는 0에 수렴하고 다른 하나는 발산한다. 따라서 $B^i x$ 역시 발산한다.

선형 독립인 n 개의 고유벡터를 다 가지지 못하는 비대칭 행렬군(群)이 존재한다는 사실을 언급할 필요가 있다. 이같은 행렬들을 *부족 행렬(defective matrix)*라고 한다. 이 이름은 좌절한 선형 대수학자들이 이 행렬에 대해 보이는 당연한 적개심을 잘 나타내고 있다. 상세한 설명은 이 문서에서 다루기에 너무 복잡하지만, 부족 행렬(defective matrix)의 행동 특성은 일반화된 고유벡터(generalized eigenvectors)와 일반화된 고유값(generalized eigenvalue)를 이용하여 분석할 수 있다. $B^i x$는 0에 수렴하기 위한 필요충분조건으로 모든 일반화된 고유값(generalized eigenvalues)들이 1보다 작은 값을 가져야 한다는 법칙은 여전히 유효하지만, 이를 증명하는 것은 더욱 어려워진다.

여기서 유용한 사실은 다음과 같다: 양의 정부호인 행렬의 고유값은 모두 양이다. 이 사실은 고유값의 정의를 이용하여 다음과 같이 증명할 수 있다.

$$\begin{aligned} Bv &= \lambda v \\ v^T Bv &= \lambda v^T v. \end{aligned}$$

양의 정부호(positive-definite) 정의에 의하여, $v^T Bv$는 양이다 (영이 아닌 v에 대하여). 따라서 λ 역시 양일 수 밖에 없다.

5.2 야코비(Jacobi) 반복법

0에 늘상 수렴하는 과정을 아는 것이 친구들 사이에서 인기를 끄는 데에 도움이 되는 것은 물론 아닌다. 좀 더 유용한 과정을 살펴보자: $Ax = b$을 풀기 위한 야코비(Jacobi) 방법이다. 행렬 A는 두 부분으로 *분리(split)*된다: A의 대각 성분을 그대로 갖고 다른 성분들은 모두 0인 D와 대각선 성분은 모두 0이고 다른 성분들은 A와 같은 E. 따라서 $A = D + E$이다. 자코비 방법은 다음과 같다:

$$\begin{aligned} Ax &= b \\ Dx &= -Ex + b \\ x &= -D^{-1}Ex + D^{-1}b \\ x &= Bx + x, \quad \text{where } B = -D^{-1}E,\ z = D^{-1}b. \end{aligned} \tag{14}$$

D가 대각행렬이므로 쉽게 역행렬을 구할 수 있다. 이 항등식은 다음과 같은 점화식을 통해 쉽게 반복 기법으로 변환할 수 있다.

$$x_{(i+1)} = Bx_{(i)} + z. \tag{15}$$

주어진 시작 벡터 x_0에 대해, 이 수식은 일련의 벡터들을 생성한다. 우리는 이 반복기법을 통해 연속적으로 나타나는 벡터들 하나 하나가 직전의 값과 비교했을 때 해 x에 더 가까워지기를 바란다. x는 수식 15의 고정점이라고 부르는데, 이는 $x_{(i)} = x$일 때 $x_{(i+1)}$도 역시 x와 같은 값이 되기 때문이다.

여기서 뒤의 수식 유도과정이 매우 자의적으로 보일 것이다. 사실 이 유도는 자의적인 것이다. 수식 14 대신에 x에 대한 항등식을 얼마든지 만들어낼 수 있다. 사실 A를 여러 가지로 분리하기만 하면 - 즉, 위의 예와 다른 D와 E을 선택하기만 하면 - 가우스-자이델(Gauss-Seidel) 방법, 축차가속완화법(Successive Over-Relaxation, SOR) 등을 유도할 수 있다. 이때 바라는 바는 우리가 선택한 분리 방법에 해당하는 B가 작은 값의 스펙트럼 반경(spectral radius)를 갖는 것이다. 여기서 필자는 편의상 야코피 분리 방법을 선택하였다.

임의의 어떤 벡터 $x_{(0)}$에서 시작한다고 하자. 각 반복에 대해, B을 이 벡터에 적용한 뒤에 z을 그 결과에 더한다. 각 반복 단계에서 어떤 일이 벌어지는가?

하나의 벡터를 우리가 이미 잘 이해하고 있는 다른 벡터들의 합으로 여기는 원칙을 다시 한 번 적용해 보자. 반복을 통해 나타나는 $x_{(i)}$을 정확한 해 x와 오차항 $e_{(i)}$의 합으로 표현해 보자. 그러면 수식 15는 다음과 같이 된다.

$$\begin{aligned} x_{(i+1)} &= Bx_{(i)} + z \\ &= B(x + e_{(i)}) + z \\ &= Bx + z + Be_{(i)} \\ &= x + Be_{(i)} \qquad \text{수식 14에 의해,} \\ \therefore e_{(i+1)} &= Be_{(i)}. \end{aligned} \tag{16}$$

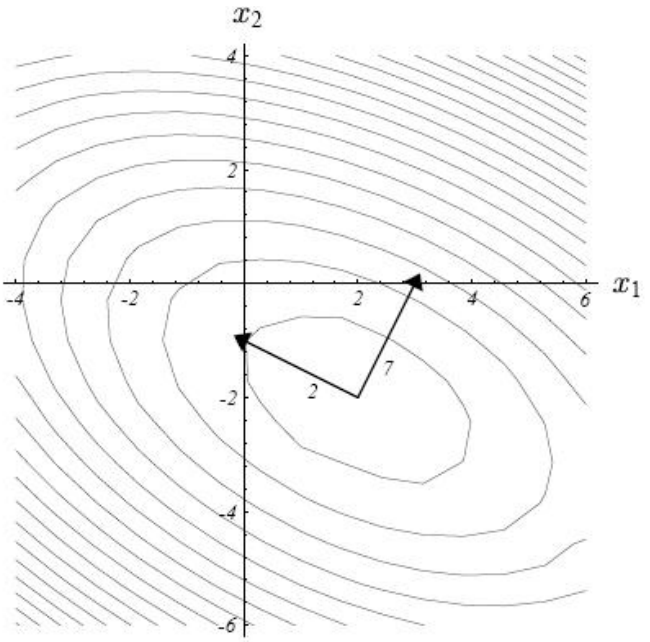
각 반복은 $x_{(i)}$의 "정확한 부분"에 영향을 주지 않지만(x가 고정점이기 때문이다); 매번 반복 때마다 오류항은 영향을 받는다. 수식 16를 통해 명백히 알 수 있는 점은 $\rho(B) < 1$인 경우에 i가 무한대로 접근하면 오차항 $e_{(i)}$가 0에 수렴한다는 것이다. 따라서 초기벡터 $x_{(0)}$를 어떻게 선택하느냐가 최종적인 해에 영향을 미치지 못한다.

물론 $x_{(0)}$를 어떻게 선택하느냐에 따라 주어진 허용오차 내에서 x에 수렴하는 데에 필요한 반복횟수에는 영향을 미치게 된다. 하지만 그 영향은 스펙트럼 반경 $\rho(B)$의 영향에 비해 훨씬 덜 중요하며, 스펙트럼 반경은 수렴의 속도를 결정한다. v_j가 B의 고유벡터들 가운데 가장 큰 고유값을 가지는 (즉, $\rho(B) = \lambda_j$) 고유벡터라고 하자. 고유벡터들의 선형 조합으로 표현된 초기 오차 $e_{(0)}$가 v_j의 방향으로 성분을 가진다면, 이 성분이 가장 느리게 수렴할 것이다.

B는 일반적으로 대칭도 아니고(비록 A가 대칭일지라도) 불완전할 지도 모른다. 하지만 야코비(Jacobi) 방법의 수렴속도는 $\rho(B)$에 크게 좌우되며, 이 값은 또한 또한 A에 달려있다. 불행하게도 야코비 기법은 모든 A에 대해 수렴하는 것은 아니며, 심지어 양의 정부호(positivie-definite)인 A에 대해서도 수렴하지 않을 수 있다.

5.3 구체적 예

이런 개념들을 구체적으로 보이기 위해, 필자는 수식 4에 나타난 예를 풀어 보려고 한다. 먼저, 고유값과 고유벡터를 찾는 방법이 필요하다. 정의에 의해 고유값 λ를 가지는 임의의 고유벡터 v에 대해 다음이 성립한다.



[그림 12] A의 고유벡터는 2차형식 $f(x)$에 의해 정의된 포물면의 축 방향과 일치한다. 각각의 고유벡터는 연관된 고유값이 같이 적혀 있다. 각각의 고유값은 해당되는 경사의 가파른 정도에 비례한다.

$$Av = \lambda v = \lambda I v$$
$$(\lambda I - A)v = 0.$$

고유벡터는 0 벡터가 아니다. 따라서 $\lambda I - A$는 특이행렬이어야 한다. 따라서 다음이 성립한다.

$$\det(\lambda I - A) = 0.$$

$\lambda I - A$의 행렬식을 *특성 다항식(characteristic polynomial)*이라 부른다. 이것은 λ에 대한 n차 다항식으로, 이 다항식의 근(根)이 행렬 A의 고유값이 된다. 수식 4에 있는 A의 특성 다항식은 다음과 같다.

$$\det \begin{bmatrix} \lambda - 3 & -2 \\ -2 & \lambda - 6 \end{bmatrix} = \lambda^2 - 9\lambda + 14 = (\lambda - 7)(\lambda - 2),$$

다항식의 근이 고유값이므로 행렬의 고유값은 7과 2이다. $\lambda = 7$일 때 고유벡터는 다음과 같이 구할 수 있다.

$$(\lambda I - A)v = \begin{bmatrix} 4 & -2 \\ -2 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0$$
$$\therefore 4v_1 - 2v_2 = 0.$$

이 방정식을 만족하는 임의의 해가 바로 고유벡터이다. 예를 들어, $v = [1, 2]^T$가 고유벡터이다. 같은 방법으로 고유값 2에 해당하는 고유벡터로 $[-2, 1]^T$를 구할 수 있다. 그림 12에서 볼 수 있는 바와 같이, 이 고유벡터들이 익숙한 타원의 축과 일치한다는 것을 볼 수 있다. 또한 고유값이 크면 더 급한 경사를 가진다는 것도 확인할 수 있다. (음수 고유값은 그림 5(b)와 5(d)에 보이는 것과 같이 축을 따라 이동할 때 f의 값이 감소함을 의미한다.)

이제 야코비(Jacobi) 방법이 작동하는 모습을 보자. 수식 4에 있는 상수들을 이용하여 다음과 같은 반복식을 얻는다.

$$x_{(i+1)} = -\begin{bmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{6} \end{bmatrix} \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix} x_{(i)} + \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{6} \end{bmatrix} \begin{bmatrix} 2 \\ -8 \end{bmatrix}$$
$$= \begin{bmatrix} 0 & -\frac{2}{3} \\ -\frac{1}{3} & 0 \end{bmatrix} x_{(i)} + \begin{bmatrix} \frac{2}{3} \\ -\frac{4}{3} \end{bmatrix}$$

B의 고유벡터를 구하면 고유값이 $-\sqrt{2}/3$일 때 $[\sqrt{2}, 1]^T$이고, 고유값이 $\sqrt{2}/3$일 때 $[-\sqrt{2}, 1]^T$이다. 이들은 그림 13(a)에서 그림으로 나타나 있다. 이들은 A의 고유벡터와 일치하지 않으며, 포물면의 축과도 상관이 없다.

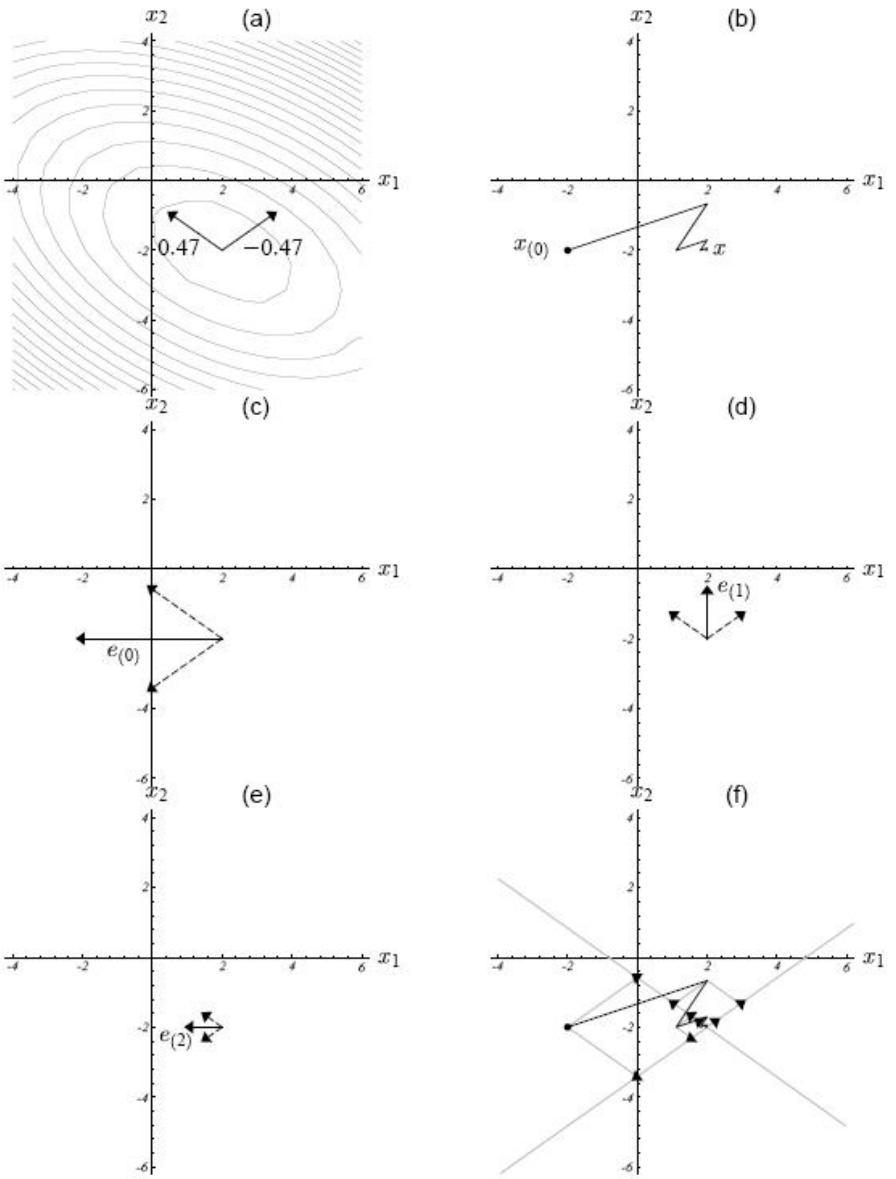
그림 13(b)는 야코비(Jacobi) 방법의 수렴성을 보여주고 있다. 이 알고리즘이 수행되면서 따라가는 신기한 경로는 반복 기법을 통해 연속적으로 나타나는 오차항 각각이 가지는 고유벡터 성분들이 어떻게 변화하는지 살펴봄으로써 이해할 수 있다.(그림 13(c), (d), (e)). 그림 13(f)는 화살촉 표시를 통해 고유벡터 성분들을 그리고 있다. 이들 성분은 그림 11에서 보는 바와 같이 각각의 고유값에 의해 결정된 속도로 수렴한다.

필자는 이절을 통해 여러분들이 "고유벡터"라는 것이 당신들의 고통을 보며 즐기기 위해 교수들이 만들어낸 괴상한 고문도구가 아니라, 매우 유용한 도구라는 사실을 확실히 알았으면 좋겠다.

6 최대 하강(Steepest Descent) 기법의 수렴 분석

6.1 한 번만에 해 찾기

최대 하강 기법의 수렴을 이해하기 위해 우선 $e_{(i)}$가 고유값 λ_e를 가지는 고유벡터라고 하자. 그러면 나머

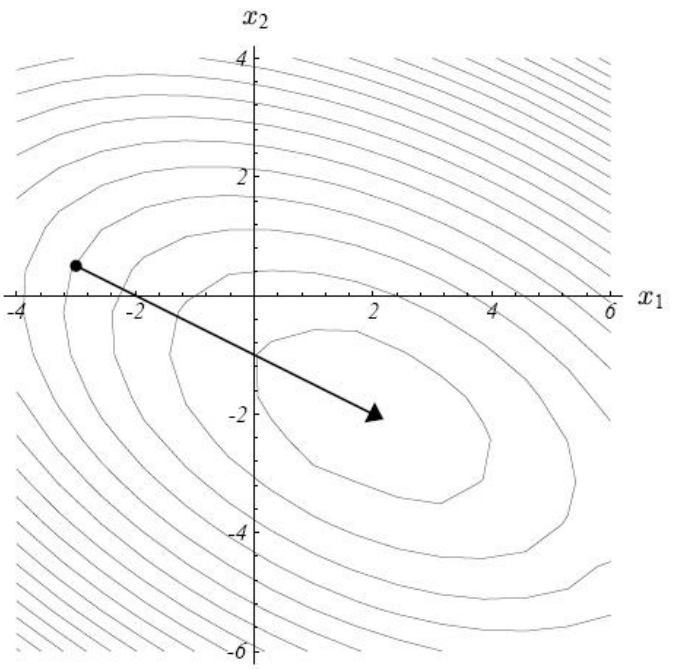


[그림 13] 야코비 방법의 수렴성. (a) B의 고유벡터들이 각각의 고유값과 함께 나타나있다. A의 고유벡터와 달리 이들 고유벡터는 보물면의 축이 아니다. (b) 야코비 기법이 $[-2,-2]^T$에서 시작하여 $[2,-2]^T$에서 수렴한다. (c,d,e) 오차 벡터 $e_{(0)}$, $e_{(1)}$, $e_{(2)}$(실선 화살표)와 이들의 고유벡터 성분(점선 화살표). (f) 화살촉은 처음 4 개 오차 벡터의 고유벡터 성분을 표현한다. 오차를 표현하는 각각의 고유벡터 성분은 이들의 고유값에 따라 예상할 수 있는 속도로 0에 수렴한다.

지(residual) $r_{(i)} = -Ae_{(i)} = -\lambda_e e_{(i)}$도 역시 고유벡터이다. 식 12를 이용하여 다음을 얻을 수 있다.

$$\begin{aligned} e_{(i+1)} &= e_{(i)} + \frac{r_{(i)}^T r_{(i)}}{r_{(i)}^T A r_{(i)}} r_{(i)} \\ &= e_{(i)} + \frac{r_{(i)}^T r_{(i)}}{\lambda_e r_{(i)}^T A r_{(i)}} (-\lambda_e e_{(i)}) \\ &= 0. \end{aligned}$$

그림 14는 이 방법이 왜 한 번만에 해에 도달하는지를 보여준다. 점 $x_{(i)}$는 타원의 축 가운데 하나에 놓여 있다. 따라서 나머지는 타원의 중심을 바로 가리키게 된다. $\alpha_{(i)} = \lambda_e^{-1}$를 선택함으로써 즉각적인 수렴을 얻을 수 있다.



[그림 14] 최대 하강(Steepest Descent) 기법은 오차항이 하나의 고유벡터로 표현될 경우 한번의 반복만으로 정확한 해에 수렴한다.

더욱 일반적인 분석을 위해, $e_{(i)}$를 고유벡터의 선형 조합으로 표현해야 하며, 또한 이 고유벡터들이 정규직교(orthonormal)이어야 한다. 부록 C2에서 행렬 A가 대칭일 경우 n 개의 직교 고유벡터가 존재함을 증명한다. 고유벡터는 임의의 크기로 변경할 수 있기 때문에 각각의 고유벡터를 정규화할 수 있다. 이렇게 고유벡터를 선택하면 다음과 같은 유용한 특성을 갖는다.

$$v_j^T v_k = \begin{cases} 1, & j = k, \\ 0, & j \neq k. \end{cases} \tag{17}$$

오차항을 고유벡터의 선형 조합으로 표현하자.

$$e_{(i)} = \sum_{j=1}^{n} \xi_j v_j, \tag{18}$$

이때 ξ_j는 $e_{(i)}$의 성분별 길이이다. 식 17과 식 18로부터 다음의 항등식을 얻는다:

$$r_{(i)} = -Ae_{(i)} = -\sum_j \xi_j v_j, \tag{19}$$

$$||e_{(i)}||^2 = e_{(i)}^T e_{(i)} = \sum_j \xi_j^2, \tag{20}$$

$$\begin{aligned} e_{(i)}^T A e_{(i)} &= (\sum_j \xi_j v_j^T)(\sum_j \xi_j \lambda_j v_j) \\ &= \sum_j \xi_j^2 \lambda_j, \end{aligned} \tag{21}$$

$$||r_{(i)}||^2 = r_{(i)}^T r_{(i)} = \sum_j \xi_j^2 \lambda_j^2, \tag{22}$$

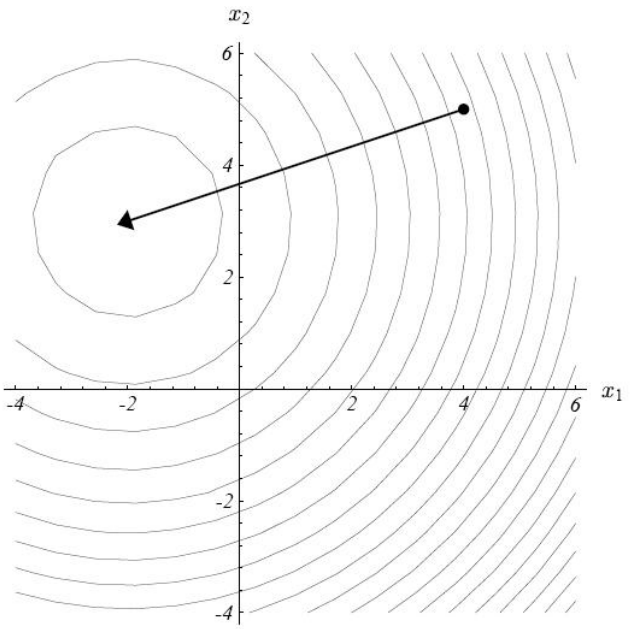
$$r_{(i)}^T A r_{(i)} = \sum_j \xi_j^2 \lambda_j^3, \tag{23}$$

식 19는 $r_{(i)}$ 역시 고유벡터 성분의 합으로 표현될 수 있음을 보여주며, 이들 성분의 길이는 $-\xi_j \lambda_j$이다. 식 20과 22는 바로 피타고라스(Pythagoras)의 법칙이다.

이제 분석을 해보자. 식 12에서 다음을 얻을 수 있다.

$$\begin{aligned} e_{(i+1)} &= e_{(i)} + \frac{r_{(i)}^T r_{(i)}}{r_{(i)}^T A r_{(i)}} r_{(i)} \\ &= e_{(i)} + \frac{\sum_j \xi_j^2 \lambda_j^2}{\sum_j \xi_j^2 \lambda_j^3} r_{(i)} \end{aligned} \tag{24}$$

마지막 예에서 우리는 $e_{(i)}$가 하나의 고유벡터 성분만을 가질 경우 $\alpha_{(i)} = \lambda_e^{-1}$를 선택함으로써 한 번만에 해에 수렴함을 보았다. 이제 $e_{(i)}$가 임의의 값이지만 모든 고유벡터가 동일한 고유치 λ를 가지는 경우를 살펴보자. 식 24를 이용하여 다음을 얻는다.

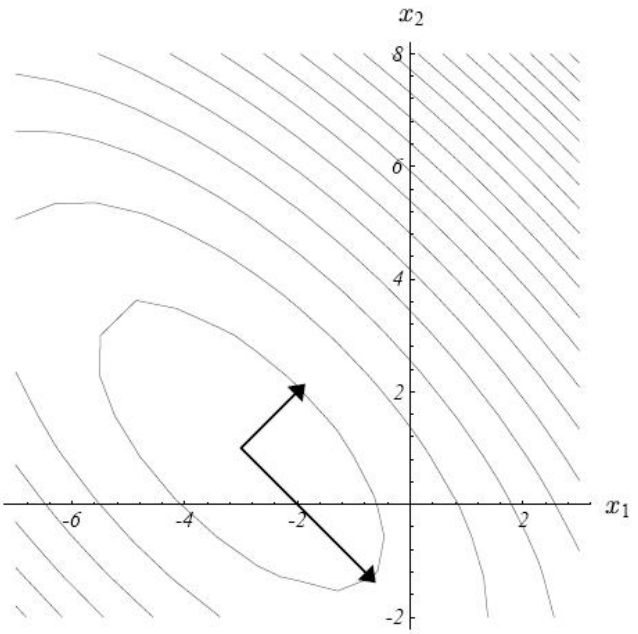


[그림 15] 최대하강은 고유값이 모두 같다면 첫번째 반복에서 정확한 해에 수렴한다.

$$\begin{aligned} e_{(i+1)} &= e_{(i)} + \frac{\lambda^2 \sum_j \xi_j^2}{\lambda^3 \sum_j \xi_j^2}(-\lambda e_{(i)}) \\ &= 0 \end{aligned}$$

그림 15는 이 방법 역시 즉각적으로 해에 수렴함을 보이고 있다. 모든 고유값들이 동일하기 때문에 타원은 구형(spherical)이 되어 어떤 위치에서 시작하든 관계없이 나머지(residual)가 구의 중심을 가리키게 된다. α를 선택하는 것은 앞에서와 마찬가지로 $\alpha_{(i)} = \lambda^{-1}$로 선택한다.

그러나, 몇 개의 서로 다른 고유값이 존재할 경우 어떤 식으로 $\alpha_{(i)}$를 선택하더라도 고유벡터 성분들을 모두 제거할 수는 없다. 따라서 이때 선택할 수 있는 방법은 일종의 타협이 된다. 사실, 식 24에 나타나 있는 분수는 λ_j^{-1} 값에 대해 가중치 평균으로 볼 수 있다. 가중치 ξ_j^2는 $e_{(i)}$의 성분들 가운데 큰 성분에 우선순위를 두게 한다. 결과적으로 매 반복 때마다 $e_{(i)}$의 짧은 성분들 가운데 일부는 실제로 길이가 늘어나기도 한다 (비록 영원히는 아니지만). 이런 이유로 최대 하강 기법과 켤레 경사도 기법은 러퍼(rougher)라고 부른다. 반대로 자코비 기법은 스무더(smoother)인데, 그 이유는 모든 고유벡터 성분들이 매 반복 때마다 줄어들기 때문이다. 최대 하강과 켤레 경사도는 수학 문헌의 일부에서 종종 스무터로 잘못 인식되고 있지만 실제로는 스무더가 아니다.



[그림 16] 그림의 두 벡터는 동일한 에너지 놈(norm)을 갖는다.

6.2 일반 수렴성

일반적인 경우에 대해 최대 하강 기법의 수렴성을 이야기하려면 *에너지 놈(energy norm)* $||e||_A = (e^T Ae)^{1/2}$을 정의해야 한다 (그림 16을 보라). 이 놈(norm)은 유클리드 놈(Euclidean norm)보다 다루기가 쉬우며 어떤 면에서는 더욱 자연스러운 놈(norm)이다; 식 8을 살펴보면 $||e||_A$를 최소화하는 것이 결국 $f(x_{(i)})$를 최소하는 것임을 알 수 있다. 이 놈(norm)을 이용하여 다음을 얻을 수 있다.

$$
\begin{aligned}
||e_{(i+1)}||_A^2 &= e_{(i+1)}^T A e_{i+1} \\
&= (e_{(i)}^T + \alpha_{(i)} r_{(i)}^T) A (e_{(i)} + \alpha_{(i)} r_{(i)}) \quad \text{(수식 12에 의해)} \\
&= e_{(i)}^T A e_{(i)} + 2\alpha_{(i)} r_{(i)}^T A e_{(i)} + \alpha_{(i)}^2 r_{(i)}^T A r_{(i)} \quad \text{(수식 14에 의해)} \\
&= ||e_{(i)}||_A^2 + 2\frac{r_{(i)}^T r_{(i)}}{r_{(i)}^T A r_{(i)}} \left(-r_{(i)}^T r_{(i)}\right) + \left(\frac{r_{(i)}^T r_{(i)}}{r_{(i)}^T A r_{(i)}}\right)^2 r_{(i)}^T A r_{(i)} \\
&= ||e_{(i)}||_A^2 - \frac{(r_{(i)}^T r_{(i)})^2}{r_{(i)}^T A r_{(i)}} \\
&= ||e_{(i)}||_A^2 \left(1 - \frac{(r_{(i)}^T r_{(i)})^2}{(r_{(i)}^T A r_{(i)})(e_{(i)}^T A e_{(i)})}\right) \\
&= ||e_{(i)}||_A^2 \left(1 - \frac{(\sum_j \xi_j^2 \lambda_j^2)^2}{(\sum_j \xi_j^2 \lambda_j^3)(\sum_j \xi_j^2 \lambda_j)}\right) \quad \text{(항등식 21, 22, 23에 의해)} \\
&= ||e_{(i)}||_A^2 \omega^2, \quad \omega^2 = 1 - \frac{(\sum_j \xi_j^2 \lambda_j^2)^2}{(\sum_j \xi_j^2 \lambda_j^3)(\sum_j \xi_j^2 \lambda_j)}
\end{aligned}
\tag{25}
$$

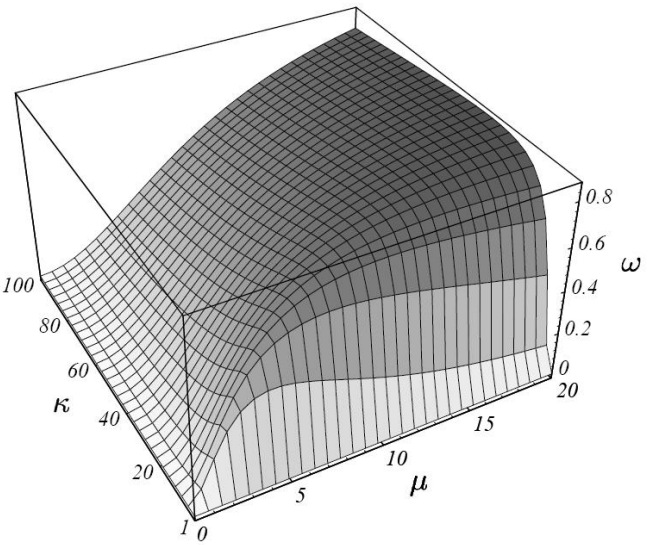
분석은 ω의 상한을 찾는 것에 달려있다. 가중치와 고유값이 어떻게 수렴에 영향을 미치는지를 보이기 위해 $n = 2$인 경우에 대해 결과를 유도할 것이다. $\lambda_1 \geq \lambda_2$라고 하자. 행렬 A의 스펙트럼 조건 수(spectral condition number)는 $\kappa = \lambda_1/\lambda_2 \geq 1$로 정의된다. $e_{(i)}$의 기울기 (고유벡터에 의해 정의되는 좌표계에 대해)는 시작점에 의존적이며 $\mu = \xi_2/\xi_1$이 된다. 이제 다음 식을 얻는다.

$$
\begin{aligned}
\omega^2 &= 1 - \frac{(\xi_1^2 \lambda_1^2 + \xi_2^2 \lambda_2^2)^2}{(\xi_1^2 \lambda_1 + \xi_2^2 \lambda_2)(\xi_1^2 \lambda_1^3 + \xi_2^2 \lambda_2^3)} \\
&= 1 - \frac{(\kappa^2 + \mu^2)^2}{(\kappa + \mu^2)(\kappa^3 + \mu^2)}
\end{aligned}
\tag{26}
$$

ω의 값은 최대 하강 기법의 수렴 속도를 결정하며 μ와 κ에 대한 함수로 그림 17과 같이 그려진다. 이 그래프로 앞서 보인 두 가지 예를 확인할 수 있다. 만약 $e_{(0)}$가 고유벡터라면 기울기 μ가 0이 되며 (혹은 무한), 이때 그래프에서 ω가 0이 됨을 볼 수 있다. 따라서 수렴은 즉각적으로 이루어진다. 또한 고유값이 모두 동일하다면 조건 수 κ가 1이되며, 이때 역시 ω가 0이된다.

그림 18은 그림 17의 네 귀퉁이 각각의 근처에 해당하는 예를 보이고 있다. 이들 이차 형식들은 고유벡터에 의해 정의되는 좌표계에서 그려졌다. 그림 18(a)와 18(b)는 큰 조건 수를 가진 예이다. 최대하강은 시작점이 운이 좋게 선택되었다면 빠르게 수렴한다 (그림 18(a)). 그러나 일반적으로 κ가 큰 값을 가지면 매우 나쁜 성능을 보인다 (그림 18(b)). 두 번째 그림은 큰 조건 수가 왜 나쁜지를 가장 직관적으로 보이고 있다: $f(x)$는 구유형태를 만들며 최대 하강 기법은 이 구유의 양쪽을 왔다 갔다하며 거의 진행하지 못하는 상태가 된다. 그림

18(c)와 18(d)에서 조건 수는 작은 값이므로 이차 형식은 거의 구형이 된다. 이때 수렴은 시작점에 관계 없이 빠르게 이루어진다.

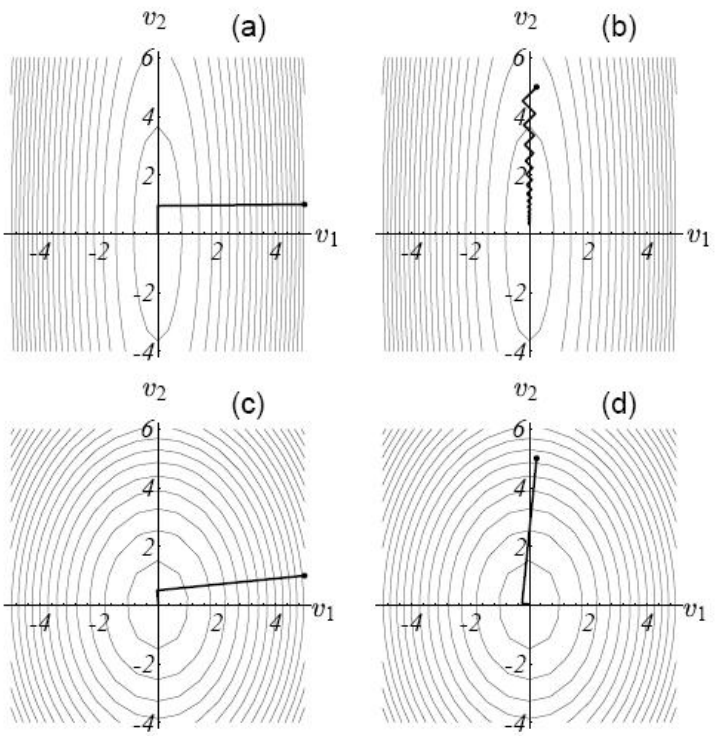


[그림 17] 최대 하강(Steepest Descent) 기법의 수렴도 ω. 수렴도 ω는 μ (오차항 $e_{(i)}$의 기울기값)과 κ (A의 조건 수)의 함수이다. 수렴은 μ와 κ가 작은 값일 때 빠르다. 고정된 행렬에서 $\mu = \pm\kappa$일 때 최악의 수렴을 보인다.

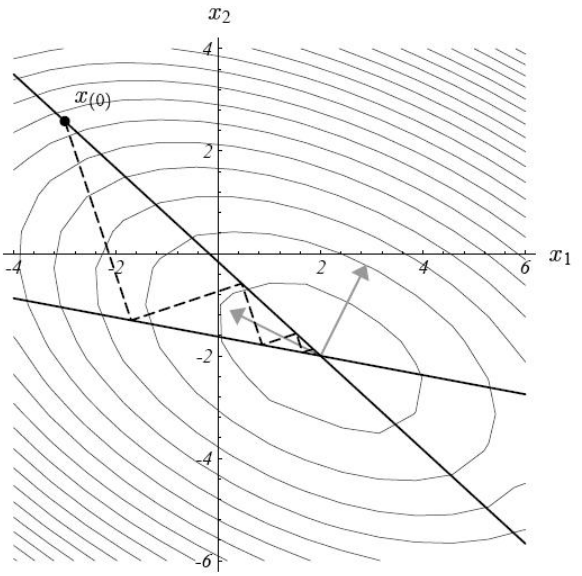
κ를 상수로 할 때(A가 고정되어 있으므로), 간단한 미적분을 통해 식 26이 $\mu = \pm\kappa$일 때 최대가 됨을 알 수 있다. 그림 17에서 이 선에 의해 만들어지는 희미한 융기선을 볼 수 있다. 그림 19는 우리가 사용해 왔던 행렬 A에 대해 최악의 시작점을 그리고 있다. 이 시작점들은 $\xi_2/\xi_1 = \pm\kappa$로 정의되는 선위에 놓인다. ω의 상한(최악의 시작점에 해당)은 $\mu^2 = \kappa^2$로 설정함으로써 찾을 수 있다.

$$\begin{aligned}
\omega^2 &\leq 1 - \frac{4\kappa^4}{\kappa^5 + 2\kappa^4 + \kappa^3} \\
&= \frac{\kappa^5 - 2\kappa^4 + \kappa^3}{\kappa^5 + 2\kappa^4 + \kappa^3} \\
&= \frac{(\kappa - 1)^2}{(\kappa + 1)^2} \\
\omega &\leq \frac{\kappa - 1}{\kappa + 1}.
\end{aligned} \tag{27}$$

식 27의 부등식 이 그림 20에 그려져 있다. 행렬이 나쁜 상태에 있을수록(즉, 조건 수 κ가 클 수록), 최대 하강의 수렴 속도는 더욱 느려진다. 절 9.2에서 대칭이며 양의 정부호인 행렬이 다음과 같은 조건 수를 가진다면 식 27이 $n > 2$인 경우에도 여전히 성립한다는 것을 증명한다.



[그림 18] 이 네 예제는 그림 17의 네 개 귀퉁이에 해당하는 지점 근처를 표현한다. (a)는 큰 κ, 작은 μ. (b) 나쁜 수렴의 예. κ와 μ 모두 크다. (c) 작은 κ와 μ. (d) 작은 κ와 큰 μ.



[그림 19] 실선들은 최대 하강 기법에서 최악의 수렴을 보이는 시작점을 표현하고 있다. 점선은 수렴 과정에서 거치는 단계들을 보이고 있다. 첫 시작이 최악의 지점에서 이루어질 경우, 그 다음 단계 역시 최악의 지엄에서 이루어 진다. 각각의 단계에서 포물면 축(회색 화살표)를 정확히 45도로 교차하게 된다. 여기서 κ는 3.5이다.

$$\kappa = \lambda_{max}/\lambda_{min},$$

이는 곧 최대 고유값과 최소고유값의 비(ratio)이다. 최대 하강 기법의 수렴 결과는 다음과 같다.

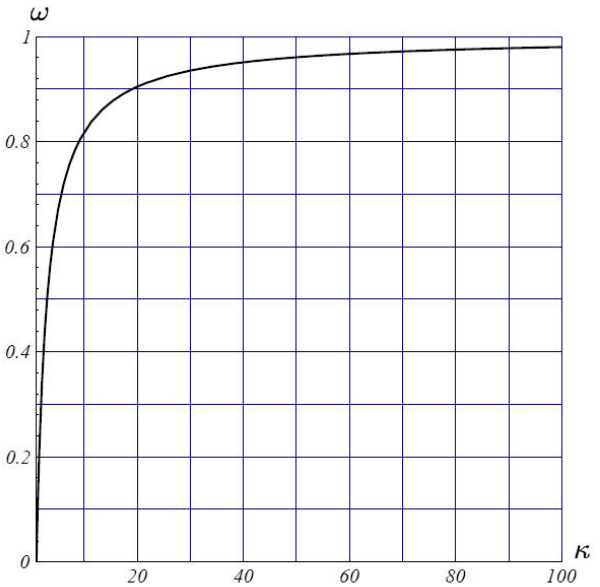
$$||e_{(i)}||_2 \leq \left(\frac{\kappa-1}{\kappa+1}\right)^i ||e_{(0)}||_A, \text{ 그리고} \tag{28}$$

$$\begin{aligned}\frac{f(x_{(i)})-f(x)}{f(x_{(0)})-f(x)} &= \frac{\frac{1}{2}e_{(i)}^T A e_{(i)}}{\frac{1}{2}e_{(0)}^T A e_{(0)}} \quad \text{(수식 8에 의해)} \\ &= \left(\frac{\kappa-1}{\kappa+1}\right)^{2i}.\end{aligned}$$

7 켤레 방향 기법

7.1 켤레성

최대 하강(Steepest Descent) 기법은 종종 그림 8과 같이 이전에 이미 탐색했던 방향을 다시 탐색한다. 그런



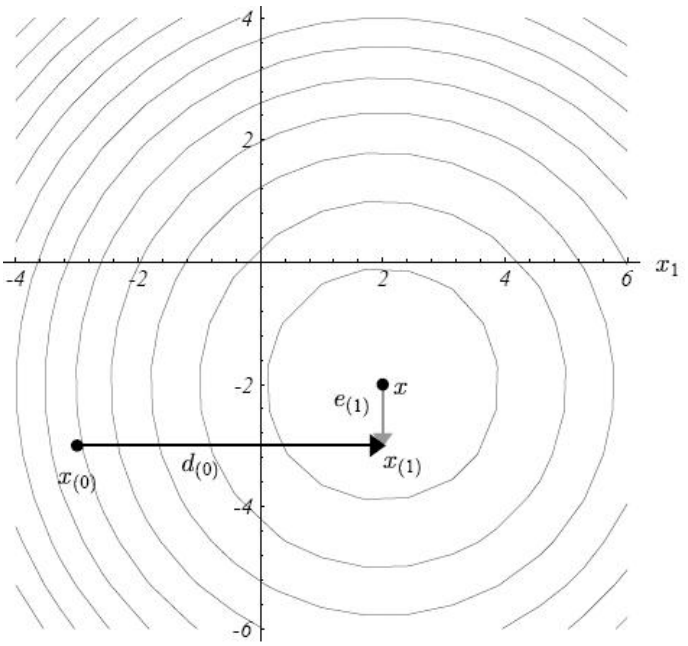
[그림 20] 최대하강의 수렴성은 행렬의 조건 수(condition number)가 증가함에 따라 악화된다.

데 이렇게 같은 방향의 탐색을 나누어서 여러 번 수행할 것이 아니라 하나의 탐색 방향에 대해 가야할 거리를 한 번에 바로 갈 수 있다면 더 낫지 않을까? 아이디어는 다음과 같다: 우선 서로 직교하는 *탐색 방향* $d_{(0)}, d_{(1)}, \cdots, d_{(n-1)}$의 집합을 골라보자. 각 탐색 방향에 대해서 우리는 한 번의 단계로 곧장 x와 가장 가까운 단계까지 가도록 하는 것이다. 이때 필요한 진행 거리를 알 수 있다면, 모든 탐색 방향에 대해 1 번의 탐색만 수행하면 되고, 결국 n 단계 후에 해에 도달할 것이다.

그림 21은 좌표축을 탐색 방향으로 사용하여 설명하고 있다. 첫 (수평) 단계에서는 정확한 x_1-좌표와 일치할 때까지 탐색을 진행하며, 두 번째 (수직) 단계에서는 원하는 해을 얻게 된다. $e_{(1)}$이 $d_{(0)}$와 수직임을 주목해 보라. 이를 일반화 하면, 매 단계에서 다음과 같이 위치를 선택하는 것이다.

$$x_{(i+1)} = x_{(i)} + \alpha_{(i)} d_{(i)} \tag{29}$$

한 번 탐색한 $d_{(i)}$ 방향으로는 앞으로 더이상 탐색이 필요 없도록 하기 위해, $e_{(e+1)}$가 $d_{(i)}$가 서로 직교해야 한다는 사실을 이용하여 $\alpha_{(i)}$ 값을 구한다. 이런 조건을 이용하여 다음 식을 얻을 수 있다.



[그림 21] 직교 방향 기법. 아쉽게도 이 방법은 우리가 이미 답을 알고 있을 경우에만 이용 할 수 있다

$$\begin{aligned} d_{(i)}^T e_{(i+1)} &= 0 \\ d_{(i)}^T (e_{(i)} + \alpha_{(i)} d_{(i)}) &= 0 \quad \text{(수식 29에 의하여)} \\ \alpha_{(i)} &= -\frac{d_{(i)}^T e_{(i)}}{d_{(i)}^T d_{(i)}} \end{aligned} \tag{30}$$

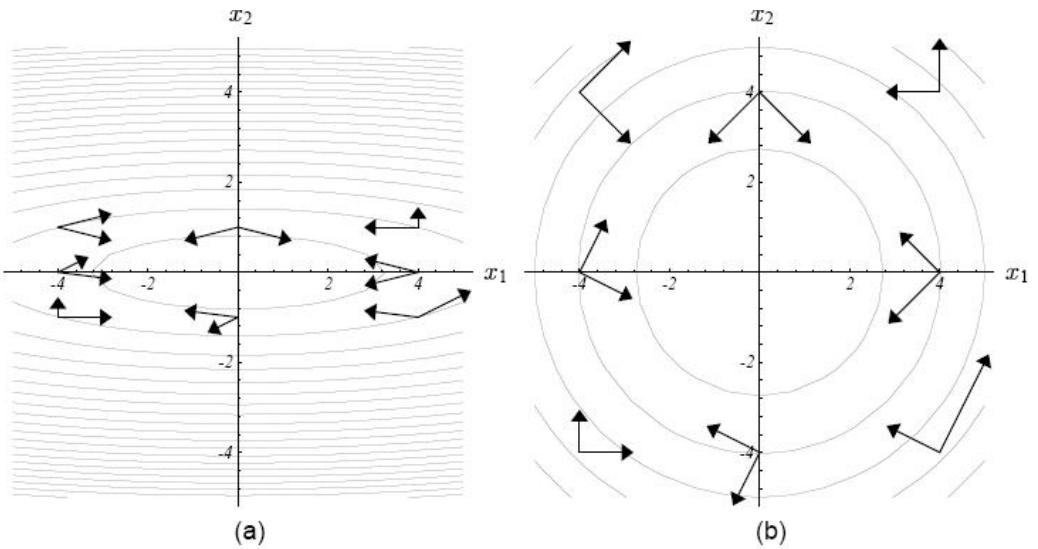
불행하게도 이 방법은 사실 쓸모 없는 방법이다. 이 방법을 사용하여 $\alpha_{(i)}$를 계산 하려면 $e_{(i)}$를 알아야만하는데, $e_{(i)}$를 안다는 것은 이미 해를 알고 있다는 것이므로 문제 자체가 이미 풀려 있는 상태인 것이다.

이런 문제점을 해결하려면, 직교하는 탐색 방향이 아니라 A-직교인 탐색 방향을 사용하여야 한다. 두 벡터 $d_{(i)}$ 와 $d_{(j)}$가 서로 A-직교 또는 켤레(conjugate)가 되려면 다음 조건을 만족해야 한다.

$$d_{(i)}^T A d_{(i)} = 0$$

그림 22 (a)는 A-직교 벡터가 어떻게 보이는가를 보여주고 있다. 이 글이 풍선 껌위에 출력되어 있다고 상상해보자. 만일 여러분이 그림 22 (a)의 양끝을 붙잡아서 타원이 원처럼 보일때까지 잡아당겼다고 생각해 보자. 그러면 벡터들이 그림 22 (b)처럼 직교하는 것처럼 보일 것이다

새로운 요구사항은 $e_{(i+1)}$와 $d_{(i)}$가 서로 A-직교(그림 23 (b))를 이루어야 한다는 것이다. 이 직교 조건은 최



[그림 22] 그림 (b) 벡터의 쌍들이 직교하므로 그림 (a) 벡터의 쌍은 A-직교이다.

대 하강 기법과 마찬가지로 검색 방향 $d_{(i)}$을 따라서 최소점을 찾는 것과 동일하다. 이를 확인하가 위해, 방향 도함수를 0으로 두고 아래와 같이 풀어보자.

$$\begin{aligned}
\frac{d}{d\alpha} f(x_{(i+1)}) &= 0 \\
f'(x_{(i+1)})^T \frac{d}{d\alpha} x_{(i+1)} &= 0 \\
-r_{(i+1)}^T d_{(i)} &= 0 \\
d_{(i)}^T A e_{(i)} &= 0
\end{aligned}$$

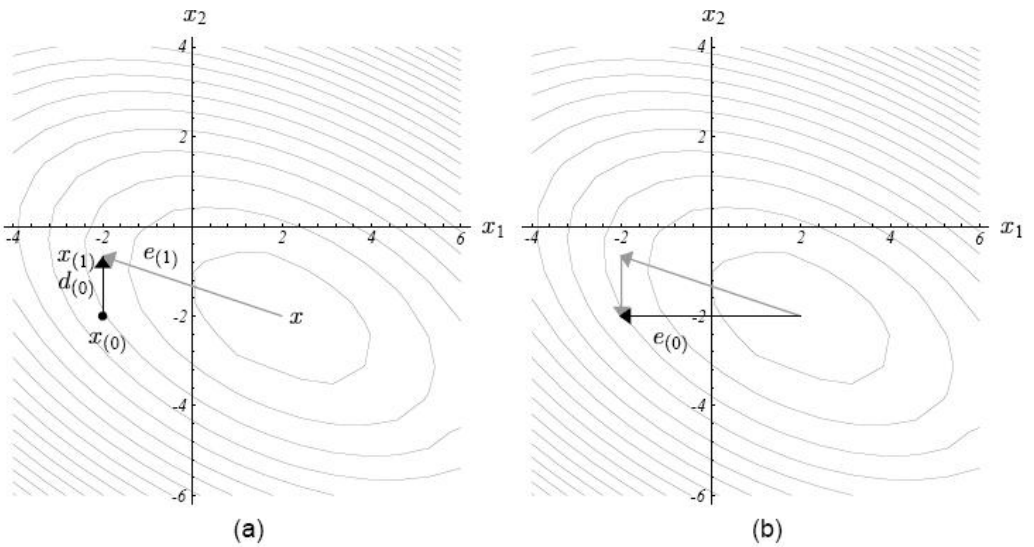
식 30의 유도 방법을 이용하여, 검색 방향들이 서로 A-직교가 되도록 하는 $\alpha_{(i)}$는 다음과 같이 표현된다.

$$\alpha_{(i)} = -\frac{d_{(i)}^T A e_{(i)}}{d_{(i)}^T A d_{(i)}} \tag{31}$$

$$= \frac{d_{(i)}^T r_{(i)}}{d_{(i)}^T A d_{(i)}} \tag{32}$$

수식 30과는 달리 이 식은 계산이 가능하다. 이 식에서 검색 벡터가 나머지(residual) 벡터와 동일하다면, 이 식은 결국 최대 하강 기법의 식과 같은 것이 된다는 점에 주목하라(수식 11을 참고하시오).

이 방법이 n 개의 단계 내에 x를 계산할 수 있다는 것을 증명하기 위해, 오차항(error term)을 검색 방향들의 선형 조합으로 다음과 같이 표현해 보자.



[그림 23] 켤레 방향 기법은 n 개의 반복 내에 수렴한다. (a) 첫 번째 단계에서는 어떤 방향 $d_{(0)}$를 따라 수행 된다. $e_{(1)}$이 반드시 $d_{(0)}$에 A-직교해야 한다는 제한조건을 이용하여 최소점 $x_{(i)}$이 선택된다. (b) 초기 오차 $e_{(0)}$는 회색 화살표로 표시된 A-직교 성분의 합으로 표현할 수 있다. 켤레 방향 기법의 매 반복 단계가 한 번 수행될 때마다 이들 성분 중에서 하나의 성분이 차례로 제거된다.

$$e_{(0)} = \sum_{j=0}^{n-1} \delta_j d_{(j)} \tag{33}$$

이 때 δ_j 값은 간단한 수학적 기교를 통해 얻을 수 있다. 검색 방향들은 상호 A-직교이므로, $d_{(k)}^T A$를 수식 33의 양변에 각각의 앞에 곱하면 하나의 탐색 방향을 제외한 다른 모든 탐색 방향에 대응하는 δ_j 값들을 제거할 수 있다:

$$\begin{aligned}
d_{(k)}^T A e_{(0)} &= \sum_j \delta_j d_{(k)}^T A d_{(j)} \\
d_{(k)}^T A e_{(0)} &= \delta_{(k)} d_{(k)}^T A d_{(k)} \quad \text{(d 벡터의 A-직교성에 의하여 얻어짐)} \\
\delta_{(k)} &= \frac{d_{(k)}^T A e_{(0)}}{d_{(k)}^T A d_{(k)}} \\
&= \frac{d_{(k)}^T A (e_{(0)} + \sum_{i=0}^{k-1} \alpha_{(i)} d_{(i)})}{d_{(k)}^T A d_{(k)}} \quad \text{(d 벡터의 A 직교성에 의하여 얻어짐)} \\
&= \frac{d_{(k)}^T A e_{(k)}}{d_{(k)}^T A d_{(k)}} \quad \text{(수식 29에 의하여 유도)}
\end{aligned} \tag{34}$$

수식 31과 34에 의하여, 우리는 $\alpha_{(i)} = -\delta_{(i)}$임을 알 수 있다. 이 사실은 오차항에 대한 새로운 관점을 제공한다. 다음 수식에서 보는 바와 같이, x를 구성하는 성분 하나 하나를 찾아나가는 과정은 결국 오차항을 성분별로 하나씩 제거해 나가는 것으로 볼 수 있다.(그림 23(b)를 보라)

$$\begin{aligned} e_{(i)} &= e_{(0)} + \sum_{j=0}^{i-1} \alpha_{(j)} d_{(j)} \\ &= \sum_{j=0}^{n-1} \delta_{(j)} d_{(j)} - \sum_{j=0}^{i-1} \delta_{(j)} d_{(j)} \\ &= \sum_{j=i}^{n-1} \delta_{(j)} d_{(j)} \end{aligned} \tag{35}$$

n 번의 반복이 끝나면, 모든 성분이 제거되어 $e_{(n)} = 0$이 된다; 증명 끝.

7.2 그람-슈미트(Gram-Schmidt) 켤레성

이제 A-직교 검색 방향으로 이루어진 집합 $\{d_{(j)}\}$을 구하기만 하면 된다. 다행스럽게도 이를 간단하게 생성하는 방법이 있는데 이 방법이 바로 *켤레 그람-슈미트 과정(conjugate Gram-Schmidt process)*이다.

n개의 선형 독립 벡터 $u_0, u_1, \cdots, u_{n-1}$를 가지고 있다고 하자. 비록 보다 지능적인 선택이 가능하기는 하지만 좌표 축을 이 n 개의 선형 독립 벡터로 쓸 수 있을 것이다. 탐색 방향 $d_{(i)}$를 만들어 내기 위해 우선 u_i를 가지고 온 뒤, 여기서 이전의 모든 탐색 방향 벡터(그림 24를 볼 것)와 A-직교하지 않는 모든 성분을 뺀다. 다 빼고 나서 남아 있는 벡터가 바로 이전의 모든 탐색 방향과 A-직교인 탐색 방향 $d_{(i)}$가 된다. 다른 말로 하자면 $d_{(0)} = u_0$로 두고, $i > 0$인 모든 경우에 대해 다음과 같이 탐색 방향 벡터를 구할 수 있다.

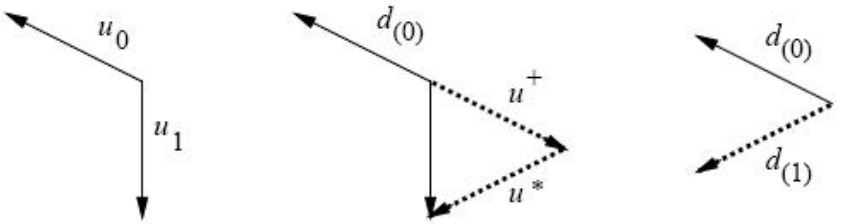
$$d_{(i)} = u_i + \sum_{k=0}^{i-1} \beta_{ik} d_{(k)} \tag{36}$$

이 때 β_{ik}는 $i > k$인 경우에 대하여 정의된다. 이 값들을 구하기 위하여, δ_j를 구할 때 사용한 것과 동일한 방법을 사용할 수 있다.

$$\begin{aligned} d_{(i)}^T A d_{(j)} &= u_i^T A d_{(j)} + \sum_{k=0}^{i-1} \beta_{ik} d_{(k)}^T A d_{(j)} \\ 0 &= u_i^T A d_{(j)} + \beta_{ij} d_{(j)}^T A d_{(j)}, \quad i > j \quad (d \text{ 벡터들 사이의 } A\text{-직교성에 의하여}) \\ \beta_{ij} &= -\frac{u_i^T A d_{(j)}}{d_{(j)}^T A d_{(j)}} \end{aligned} \tag{37}$$

켤레 방향 (conjugate direction) 기법에 그람-슈미트 켤레화 (Gram-Schmidt conjugation)을 사용할 때의 문제점은 새로운 탐색 방향을 결정하기 위해 이전의 모든 탐색 방향 벡터들을 메모리에 가지고 있어야 한다는 것이며, 더구나 전체 탐색 방향을 결정하기 위해서는 모두 $O(n^3)$의 연산이 필요하다는 것이다. 사실 단위 길이의 축 벡터를 바탕으로 켤레화(conjugation)를 수행하여 탐색 방향들을 결정하였다면, 켤레 방향 기법은 결국 가우스 소거법(그림 25)과 동일한 것이 된다. 결과적으로 켤레 방향 기법은 켤레 경사도 기법이 -켤레 경사도 기법 역시 켤레 방향 기법의 한 종류이다- 나타나 이런 이런 문제를 해결할 때까지 거의 사용되지 않았다.

켤레 방향 기법을 이해하는 데에 핵심적인 요소는 (켤레 경사도 기법도 마찬가지이다) 그림 25가 그림 21를 늘인 것에 불과하다는 것을 알아차리는 것이다. 켤레 방향 기법을 수행할 때(켤레 경사도 기법을 포함하여), 그 방법이 사실 늘어난 (혹은 크기가 변형된) 공간에서 볼 때 직교 방향 (orthogonal direction) 기법으로 문제를 풀고 있는 것과 같다는 것을 기억하라.

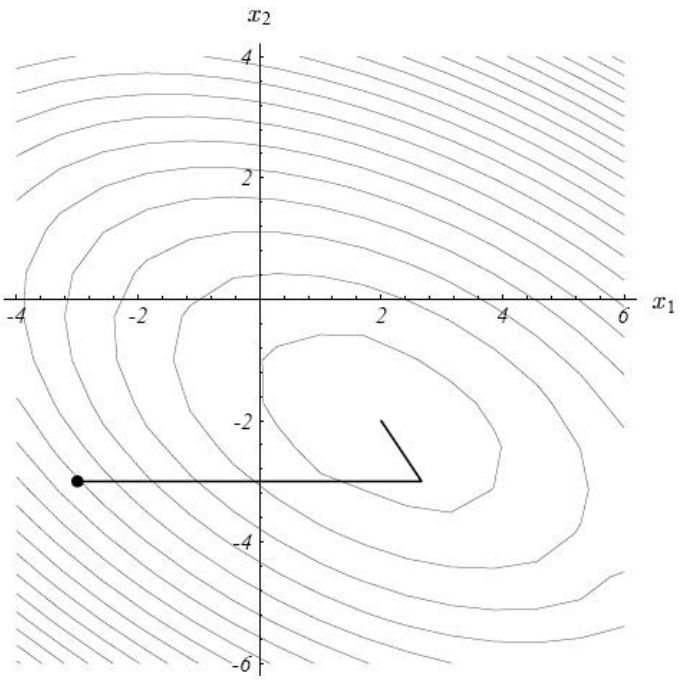


[그림 24] 두 벡터의 그람-슈미트 켤레화(Gram-Schmidt conjugate). 두 개의 선형 독립 벡터 u_0와 u_1 벡터에서 시작한다. $d_{(0)} = u_{(0)}$로 설정한다. 벡터 u_1은 d_0에 A-직교(또는 conjugate)인 u^*와 d_0에 평행인 u^+두 성분으로 분해될 수 있다. 켤레화 이후 A-직교 부분만이 남게 되어 $d_1 = u^*$가 된다

7.3 오차항의 최적성

켤레 방향 기법은 다음과 같은 재미있는 특징이 있다: 이 방법은 매 번의 단계에서 탐색이 하용된 범위 내에서 최선의 해를 구한다. 탐색이 허용된 영역이 어디일까? span$\{d_{(0)}, d_{(1)}, \cdots, d_{(i-1)}\}$는 탐색 벡터 $d_{(0)}, d_{(1)}, \cdots, d_{(i-1)}$에 의해 생성되는 i-차원 부분공간이며 이를 $\mathcal{D}_i$라고 표현하자. $e_{(i)}$ 값은 $e_{(0)} + \mathcal{D}_i$로 부터 선택될 수 있다. 여기에서 말하는 "최선의 해"라는 게 무슨 뜻일까? 켤레 방향 기법이 최선의 해를 찾는다는 것은 이 기법이 $e_{(0)} + \mathcal{D}_i$의 부분 공간 내에서 값을 선택할 때 $||e_{(i)}||_A$값이 최소가 되는 값을 선택한다는 것이다(그림 26을 보라). 사실 어떤 학자들은 부분 공간 $e_{(0)} + \mathcal{D}_i$ 내에서 $||e_{(i)}||_A$ 값을 최소화함으로써 켤레 경사도 (conjugate gradient) 유도하기도 한다.

동일한 방법으로 오차항 역시 검색 방향의 선형 조합으로 표현될 수 있으며(수식 35), 오차항의 에너지 놈(norm)은 다음과 같은 합산 표현으로 나타낼 수 있다.



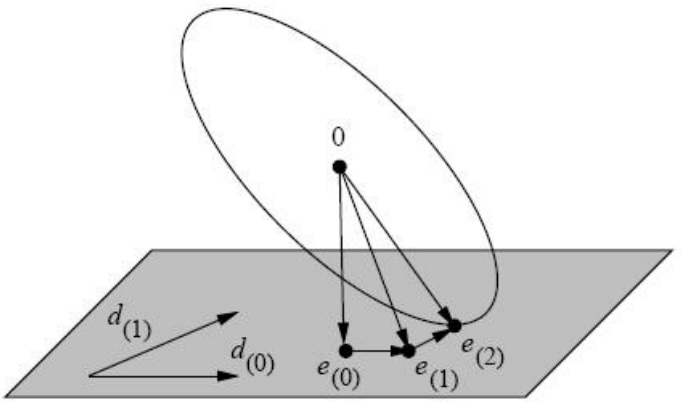
[그림 25] 축방향 단위벡터를 사용한 켤레 방향 기법은 가우스 소거법(Gauss Elimination)으로도 알려져 있다.

$$
\begin{aligned}
||e_{(i)}||_A &= \sum_{j=i}^{n-1}\sum_{k=i}^{n-1}\delta_{(j)}\delta_{(k)}d_{(j)}^T A d_{(k)} \quad \text{(수식 35에 의해 유도)} \\
&= \sum_{j=i}^{n-1}\delta_{(j)}^2 d_{(j)}^T A d_{(j)} \quad \text{(d 벡터의 A-직교성에 의하여 유도)}
\end{aligned}
$$

이 합산(summation) 표현식에 있는 모든 항은 아직 사용되지 않은 탐색 방향에만 연관되어 있다. $e_{(0)} + \mathcal{D}_i$ 부분공간에서 선택된 임의의 다른 벡터 e는 이 합산 표현식을 확장했을 때 나타나는 것과 동일한 항들을 가져야 하며, 이는 결국 $e_{(i)}$가 반드시 최소 에너지 놈(norm)을 가진다는 것을 증명한다.

수식을 통해 최적성을 증명하였으니, 이젠 직관적으로 이를 이해할 수 있도록 해보자. 켤레 방향 기법의 작업 과정을 시각적으로 표현할 수 있는 최선의 방법은 그림 22와 같이 우리가 작업하는 공간과 "늘어난(stretched)" 공간을 서로 비교하는 것이다. 그림 27 (a)와 27 (c)는 켤레 방향 기법이 $\mathbb{R}^2$ 공간과 $\mathbb{R}^3$공간에서 어떻게 동작하는지를 보여주고 있다; 이 그림에서 서로 수직으로 보이는 선들은 서로 직교(orthogonal)한다. 반면에, 그림 27(b)와 27(d)에서는 동일한 그림을 고유벡터 축을 따라 늘려 타원형 등고선이 구형(spherical)이 되도록 한 것이다. 이 그림에서 수직으로 보이는 선들은 서로 A-직교이다.

그림 27(a)을 보면 켤레 방향 기법이 초기 추정치 $x_{(0)}$에서 시작하며, 탐색 방향 $d_{(0)}$를 이용하여 한 단계 이동하여 다음 추정치인 $x_{(1)}$ 점에서 멈추고 있다. 이때 새로운 오차 벡터 $e_{(1)}$는 탐색 방향 $d_{(0)}$과 A-직교이다. 이



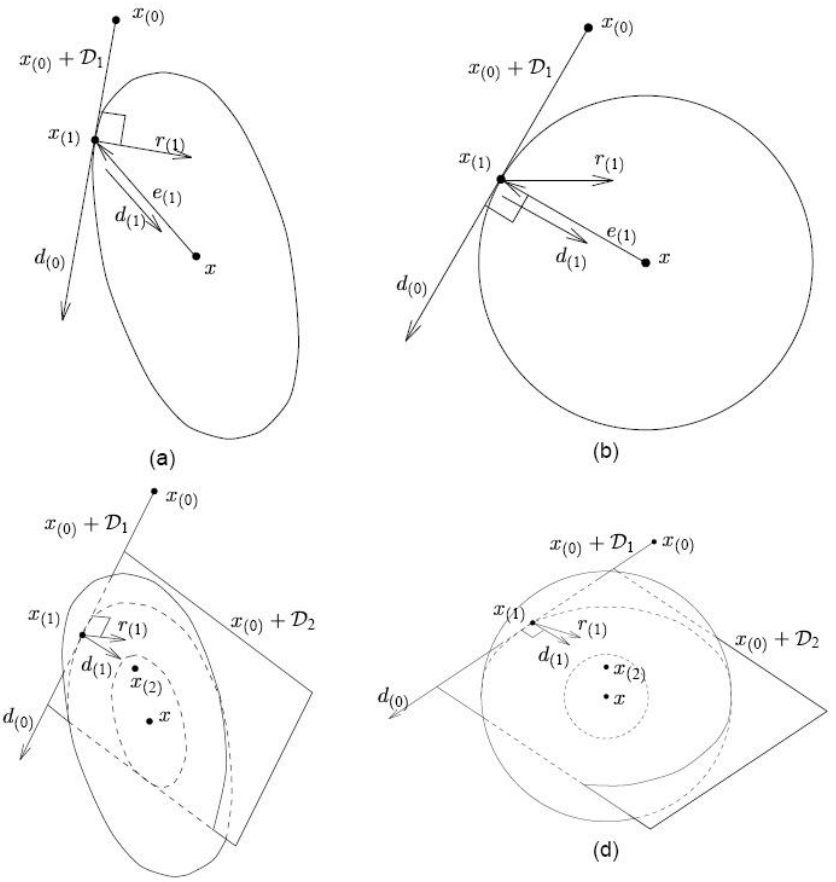
[그림 26] 이 그림에서 어둡게 칠해진 영역이 $e_{(0)} + \mathcal{D}_2 = e_{(0)} + span\{d_{(0)}, d_{(1)}\}$이다. 타원은 에너지 놈(norm)이 일정한 값을 갖는 등고선이다. 두 단계가 수행된 뒤, 켤레 방향 기법은 $e_{(0)} + \mathcal{D}_2$ 상에서 $||e_{(i)}||_A$ 값이 최소가 되는 점 $e(2)$를 찾게 된다.

렇게 찾은 $x_{(1)}$이 어떻게 부분공간 $x_{(0)} + \mathcal{D}_1$에서 최소점이라는 것을 보장할 수 있을까? 답은 그림 27(b)에 나타나 있다: $e_{(1)}$와 $d_{(0)}$가 서로 A-직교이기 때문에 여기 보이는 늘어난 공간(stretched space)에서 두 벡터가 수직을 이루고 있다. 에러 벡터 $e_{(1)}$은 동일한 에너지 놈(norm)의 오차를 갖는 (즉, 동일한 $||e||_A$를 가지는) 등고선들을 표현하는 동심원들의 반지름이다. 따라서 부분 공간 $x_{(0)} + \mathcal{D}_1$이 바로 x_1이 놓여있는 동심원-즉, $x_{(1)}$과 동일한 에너지 놈(norm)의 오차를 갖는 등고선-에 접하는 접평면이 되며 이때 이 접평면과 등고선이 접하는 지점이 바로 $x_{(1)}$이다.

이것은 사실 그리 놀랄만한 일은 아니다; 이미 7.1 절에서 살펴본 바와 같이, 탐색 방향과 오차항이 서로 A-켤레라는 사실이 결국 탐색 방향을 따라 f를 최소화 시키는 것(따라서 $||e||_A$를 최소화 시키는 것)과 동일하다. 그러나 켤레 방향 기법이 두 번 째 단계를 수행하여 두 번째 탐색 방향 $d_{(1)}$ 방향으로 $||e||_A$을 수행한 뒤에도 여전히 $||e||_A$가 이전에 사용했던 탐색 방향 $d_{(0)}$ 방향에서 최소임이 유지된다는 것을 어떻게 보장할 수 있을까? i 번의 반복을 수행하고 나면, 어떻게 $f(x_{(i)})$가 부분공간 $x_{(0)} + \mathcal{D}_i$ 전체에서 최소화될 수 있는 것일까?

$d_{(0)}$와 $d_{(1)}$는 서로 A-직교이기 때문에 그림 27(b)에서 수직으로 만난다. 여기서 탐색 방향 $d_{(1)}$이 해 x를 향한다 것은 명백한데, 그 이유는 $d_{(0)}$ 벡터가 x를 중심으로 하고 $x_{(1)}$를 지나는 원에 대해 $x_{(1)}$에서 접하기 때문이다. 삼차원 예제는 이를 더욱 잘 보여준다. 그림 27(c)와 27(d) 각각에는 중심이 같은 두 개의 타원체가 그려져 있다. $x_{(1)}$는 외부 타원체 위에 놓여 있고, $x_{2)}$는 내부 타원체에 놓여 있다. 이들 그림을 세심히 살펴 보라: 평면 $x_{(0)} + \mathcal{D}_2$는 큰 타원을 통과하며 자르고 있으며, 이 평면은 내부 타원과 $x_{(2)}$에서 접하는 평면이 된다. 해 x는 이 평면 아래의 타원체의 중심에 있다.

그림 27(c)를 보면 앞서의 질문을 다시 고칠 수 있다. 만일 여러분과 내가 $x_{(1)}$ 위치에 서 있는데 부분 공간



[그림 27] 켤레 방향 기법의 최적성. (a) 이차원 문제. 수직으로 만나는 선들은 서로 직교(orthogonal)한다. (b) "늘어난(stretched)" 공간에서 표현된 동일한 문제. 여기서 수직으로 만나는 선들은 서로 A-직교이다. (c) 삼차원 문제, x를 중심으로 하는 두 개의 동심 타원체가 보이고 있다. 직선 $x_{(0)} + \mathcal{D}_1$은 외부 타원체에 대해 $x_{(1)}$점에서 접한다. 평면 $x_{(0)} + \mathcal{D}_2$는 내부 타원에 대해 $x_{(2)}$점에서 접한다. (d) 삼차원 문제를 늘어난 공간(stretched space)으로 옮겼을 때 모습.

$x_{(0)} + \mathcal{D}_2$ 상을 걸어 다니면서 오차의 놈(norm) $||e||$를 최소로 할 수 있는 곳으로 이동하려고 한다고 가정하자; 그런데 우리가 다닐 수 있는 곳은 탐색 바향 $d_{(1)}$에서 벗어날 수가 없다. 만약 이 탐색 방향 $d_{(1)}$이 최소점을 바로 가리키고 있다면 우리의 목표를 달성할 수 있다. 이 탐색 벡터 $d_{(1)}$이 최소점을 곧장 가리키고 있다고 보장할 수 있는 이유가 있는가?

그림 27(d)가 그 답을 보여준다. $d_{(1)}$이 $d_{(0)}$에 대하여 A-직교이므로, 이 그림에서 서로 수직으로 만나고 있다. 이제 여러분이 평면 $x_{(0)} + \mathcal{D}_2$에 이 평면이 한 장의 종이인 것처럼 생각하고 아래를 쳐다 보자; 이때 보게 되는 장면은 그림 27(b)와 동일할 것이다. 점 $x_{(2)}$은 종이의 중심이 될 것이고, 점 x는 종이의 아래 쪽에 있게 되는데, 그 위치는 $x_{(2)}$에서 수직으로 곧장 아래로 내려간 위치가 될 것이다. $d_{(1)}$과 $d_{(0)}$이 수직으로 만나고 있으므로, 탐색 방향 $d_{(1)}$는 정확히 $x_{(2)}$를 향하게 되며, 이 $x_{(2)}$는 평면 $x_{(0)} + \mathcal{D}_2$ 상에 있는 점들 가운데 해 x에 가장 가까운 점이 될 것이다. 평면 $x_{(0)} + \mathcal{D}_2$는 $x_{(2)}$가 놓여있는 구에 접하는 평면이다. 여기서 세 번째 단계를 수행하게 되면, $\mathcal{D}_2$와 A-직교하는 방향으로 움직여 $x_{(2)}$에서 해 x로 곧장 내려가게 될 것이다.

그림 27(d)에서 무슨 일이 벌어지는지를 이해하는 또 다른 방법은 여러분 자신이 해 x에 서서 부분 공간 $x_{(0)} + \mathcal{D}_i$에서만 움직이도록 제한되어 있는 구슬을 끈으로 잡아 당기고 있다고 상상하는 것이다. 매 단계마다 *확장하는 부분공간(expanding subspace)* $\mathcal{D}$의 차원이 하나씩 늘어나고, 이 때마다 구슬은 조금씩 더 여러분 가까이 다가올 수 있도록 자유로워질 것이다. 이제 공간을 찌그러뜨려 그림 27(b)처럼 보이게 만들면 그것이 바로 켤레 방향 기법이 된다.

켤레 방향 기법의 또다른 중요한 특성이 이들 그림에 나타나 있다. 우리는 매 번의 수행 단계에서 초평면(hyperplane) $x_{(0)} + \mathcal{D}_i$가 $x_{(i)}$가 놓여있는 타원에 접하게 된다는 것을 살펴 보았다. 4 절에 설명한 바와 같이 임의의 점에서 가지는 나머지(residual)는 이 지점을 지나는 타원체 표면과 직교한다는 사실을 기억하라. 이것은 결국 $r_{(i)}$가 $\mathcal{D}_i$와 직교한다는 것을 의미한다. 이를 수학적으로 보이기 위해 식 35에 $-d_{(i)}^T A$를 각각의 양변 앞에 곱해 보자.

$$-d_{(i)}^T A e_{(i)} = -\sum_{j=i}^{n-1} \delta_{(j)} d_{(i)}^T A d_{(j)} \tag{38}$$

$$d_{(i)}^T r_{(j)} = 0, \quad i < j \quad \text{(d-벡터의 A-직교성에 의하여.)} \tag{39}$$

이 항등식은 다른 방법으로 유도할 수도 있다. 탐색 방향에 대해 한 번의 단계를 수행하고 나면 앞으로 더이상은 그 방향으로 탐색할 필요가 없다는 사실을 상기하자; 오차항은 이전의 모든 탐색 방향에 대해 항상 A-직교이며, $r_{(i)} = -Ae_{(i)}$이므로 나머지(residual)는 이전의 탐색 방향과 언제나 직교가 된다.

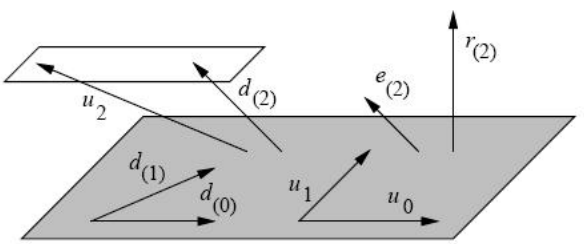
탐색 방향이 u 벡터들을 이용하여 만들어졌으므로, $u_0, \cdots, u_{i-1}$가 생성하는 부분공간이 바로 $\mathcal{D}_i$이며, 나머지(residual) $r_{(i)}$는 이러한 이전 u벡터와 직교한다 (그림 28을 보라). 이것은 수식 36과 $r_{(j)}$의 내적을 취하여 증명할 수 있다.

$$d_{(i)}^T r_{(j)} = u_i^T r_{(j)} + \sum_{k=0}^{i-1} \beta_{ij} d_{(k)}^T r_{(j)} \quad (40)$$

$$0 = u_i^T r_{(j)}, \quad i < j \quad \text{(수식 39에 의하여)} \quad (41)$$

앞으로 사용될 항등식이 하나 더 있는데. 이 항등식은 수식 40(그리고 그림 28)으로 부터 얻을 수 있는 다음과 같은 식이다.

$$d_{(i)}^T r_{(i)} = u_i^T r_{(i)}. \quad (42)$$



[그림 28] 탐색 방향 $d_{(0)}$와 $d_{(1)}$이 벡터 u_0, u_1를 이용하여 만들어졌으므로, 이들은 동일한 부분공간 $\mathcal{D}_2$(회색 영역)를 생성한다. 오차항 $e_{(2)}$는 $\mathcal{D}_2$에 A-직교이며, 나머지(residual) $r_{(2)}$는 $\mathcal{D}_2$와 직교이다. 다음 반복을 위한 새로운 탐색 방향 $d_{(2)}$은 지금까지의 탐색 방향 벡터들로 생성되는 부분 공간 $\mathcal{D}_2$에 A-직교하도록 (u_2를 이용하여) 만들어 진다. $d_{(2)}$를 생성할 때 u_2를 가지고 그람-슈미트 켤레화를 적용하여 생성하기 때문에, $d_{(2)}$와 u_2의 끝점들은 $\mathcal{D}_2$에 평행한 평면에 놓여있다.

이 절을 끝맺으면서 한 가지 더 언급할 것은 최대 하강 기법을 사용할 때와 마찬가지로 켤레 방향 기법 역시 행렬-벡터 곱하기 연산을 매 반복 시기마다 한 번만 하도록 할 수 있다는 점이다. 이를 위해서는 나머지(residual)을 계산할 때 다음과 같은 점화식을 사용하여야 한다.

$$\begin{aligned} r_{(i+1)} &= -Ae_{(i+1)} \\ &= -A(e_{(i)} + \alpha_{(i)} d_{(i)} \\ &= r_{(i)} - \alpha_{(i)} A d_{(i)} \end{aligned} \quad (43)$$

8 켤레 경사도(conjugate gradient) 기법

켤레 경사도 기법을 다루는 글에서 켤레 경사도 기법에 대한 내용이 지금까지도 언급되지 않는 것이 이상할 것이지만, 필요한 모든 장치가 이제 갖춰졌다. 사실 켤레 경사도 기법이라는 것은 켤레 방향 기법과 동일한데, 단지 탐색 방향을 만들때 나머지(residual)를 켤레화하여 만든다는 것이다 (즉, $u_i = r_{(i)}$로 설정한다).

이러한 선택 방법은 여려 가지 이유로 합리적이다. 우선 최대 하강 기법에서 나머지(residual)을 사용하는 것이 잘 동작했으니 켤레 경사도 기법에서는 사용하지 못할 이유가 없다. 둘째, 나머지(residual)은 이전의 탐색 벡터와 직교한다는 좋은 특성을 가지고 있다(수식 39). 따라서 나머지(residual) 벡터가 0벡터가 아니라면 항상 이전의 탐색 방향에 선형 독립인 새 탐색 방향을 생성할 수가 있다. 나머지 벡터가 0 벡터인 경우는 그 자체로 해가 얻어진 경우이므로 문제가 되지 않는다. 나중에 보게 되겠지만, 나머지(residual)을 사용하는 데에는 더욱 합리적인 이유가 존재한다.

나머지 벡터를 사용하는 것이 어떤 의미를 가지는지 생각해 보자. 탐색 벡터들이 나머지(residual)을 이용하여 만들어지기 때문에, 나머지 벡터들이 생성하는 부분 공간, 즉 span$\{r_{(0)}, r_{(1)}, \cdots, r_{(i-1)}\}$은 $\mathcal{D}_i$와 동일하다. 또한 각각의 나머지(residual)는 이전 검색 방향과 직교이므로 이전의 나머지 벡터 모두와도 역시 직교한다((그림 29 참조). 따라서 수식 41은 다음과 같이 다시 표현된다.

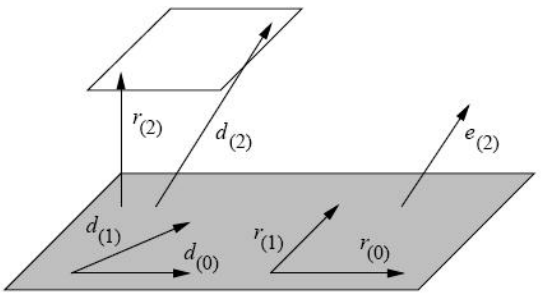
$$r_{(i)}^T r_{(j)} = 0, \quad i \neq j \tag{44}$$

43에서 흥미로운 사실은 각각의 새 나머지(residual) $r_{(i)}$가 이전 나머지(residual)와 $Ad_{(i-1)}$의 선형조합이라는 것이다. $d_{(i-1)} \in \mathcal{D}_i$임을 상기하면, 이 사실은 결국 각각의 새로운 부분공간 $\mathcal{D}_{i+1}$이 이전 부분공간 $\mathcal{D}_i$와 부분공간 $A\mathcal{D}_i$의 합으로부터 형성된다는 것이다.

$$\begin{aligned} \mathcal{D}_{(i)} &= \mathrm{span}\{\mathrm{d}_{(0)}, \mathrm{Ad}_{(0)}, \mathrm{A}^2\mathrm{d}_{(0)}, \cdots, \mathrm{A}^{\mathrm{i}-1}\mathrm{d}_{(0)}\} \\ &= \mathrm{span}\{\mathrm{r}_{(0)}, \mathrm{Ar}_{(0)}, \mathrm{A}^2\mathrm{r}_{(0)}, \cdots, \mathrm{A}^{\mathrm{i}-1}\mathrm{r}_{(0)}\}. \end{aligned}$$

이 부분공간을 *크릴로프 부분공간(Krylov subspace)*이라고 하는데, 이것은 하나의 벡터에 어떤 행렬을 반복적으로 적용하여 생성할 수 있다. 이 공간은 매우 좋은 특성을 가지고 있다: $A\mathcal{D}i$가 $\mathcal{D}_{i+1}$에 포함되어 있고, 다음 나머지 r_{i+1}과 $\mathcal{D}_{i+1}$과 직교하므로(수식 39), r_{i+1}이 $\mathcal{D}_i$과 A-직교를 이룬다는 사실을 알 수 있다. r_{i+1}이 $d_{(i)}$를 제외한 이전의 모든 탐색 방향과 이미 A-직교를 이루고 있기 때문에 그람-슈미트 결레화 작업이 간단해지는 것이다!

수식 37에 나타나는 그람-슈미트 상수가 $\beta_{ij} = -u_i^T Ad_{(j)} / d_{(j)}^T Ad_{(j)}$임을 상기하자; 이 식을 간략하게 만들어 보자. 우선 수식 43과 r_i을 내적하면 다음과 같은 결과를 얻는다.



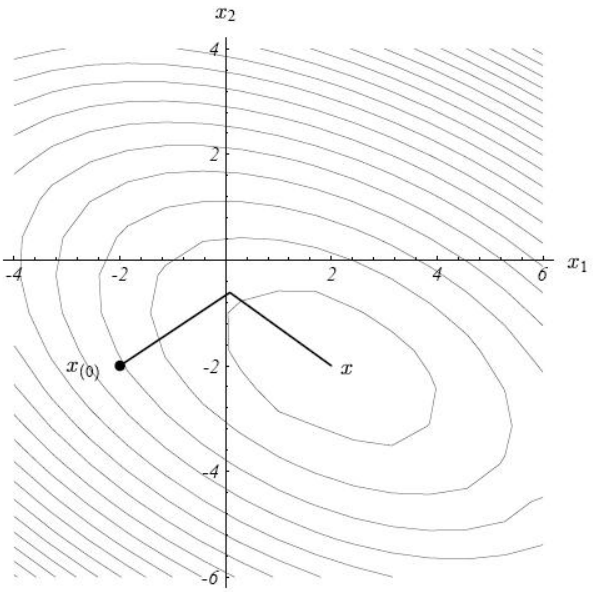
[그림 29] 켤레 경사도 기법에서, 매 단계에서 새로 구해지는 나머지(residual)는 이전의 단계에서 나타났던 나머지(residual) 무두에 대하여 직교이며, 새롭게 선택되는 탐색 방향은 이전의 모든 나머지와 탐색 방향에 대해 A-직교가 되도록 만들어진다. (탐색 방향은 나머지 벡터를 이용하여 만든다). $r_{(2)}$와 $d_{(2)}$의 끝점은 $\mathcal{D}_{(2)}$(어둡게 그려진 부분공간)와 평행한 평면에 놓여 있다. 켤레 경사도 기법에서 $d_{(2)}$는 $r_{(2)}$와 $d_{(1)}$의 선형 조합이다.

$$
\begin{aligned}
r_{(i)}^T r_{(j+1)} &= r_{(i)}^T r_{(j)} - \alpha_{(j)} r_{(i)}^T A d_{(j)} \\
\alpha_{(j)} r_{(i)}^T A d_{(j)} &= r_{(i)}^T r_{(j)} - r_{(i)}^T r_{(j+1)} \\
r_{(i)}^T A d_{(j)} &= \begin{cases} \frac{1}{\alpha_{(i)}} r_{(i)}^T r_{(i)}, & i = j, \\ \frac{-1}{\alpha_{(i-1)}} r_{(i)}^T r_{(i)}, & i = j+1, \qquad \text{(식 44에 의하여)} \\ 0, & \text{그외의 경우.} \end{cases} \\
\therefore \beta_{ij} &= \begin{cases} \frac{1}{\alpha_{(i-1)}} \frac{r_{(i)}^T r_{(i)}}{d_{(i-1)}^T A d_{(i-1)}}, & i = j+1 \\ 0, & i > j+1, \ \text{(식 37에 의하여.)} \end{cases}
\end{aligned}
$$

마술처럼 대부분의 β_{ij} 항은 사라져 버렸다. 이제 새로운 벡터가 A-직교가 되도록 하기 위해 더이상 이전에 사용했던 탐색 벡터을 모두를 저장할 필요가 없다. 이러한 중요한 장점이 켤레 경사도 기법의 중요한 점이며, 반복시 공간 복잡도와 시간 복잡도 면에서 $O(n^2)$에서 $O(m)$로 감소된다. 이때 m은 A의 0이 아닌 성분의 수이다. 이제부터 나는 $\beta_{(i)} = \beta_{i,i-1}$의 축약형으로 표기할 것이다. 좀더 간략히 만들면 다음과 같다.

$$
\begin{aligned}
\beta_{(i)} &= \frac{r_{(i)}^T r_{(i)}}{d_{(i-1)}^T r_{(i-1)}} \quad \text{(수식 32에 의하여)} \\
&= \frac{r_{(i)}^T r_{(i)}}{r_{(i-1)}^T r_{(i-1)}} \quad \text{(수식 42에 의하여)}
\end{aligned}
$$

이제 지금까지 설명한 내용 모두 모아 하나로 정리해 보자. 켤레 경사도 기법이란 다음과 같다.



[그림 30] 켤레 경사도 기법

$$d_{(0)} = r_{(0)} = b - Ax_{(0)}, \tag{45}$$

$$\alpha_{(i)} = \frac{r_{(i)}^T r_{(i)}}{d_{(i)}^T A d_{(i)}} \text{ (수식 32와 42에 의하여)} \tag{46}$$

$$x_{(i+1)} = x_{(i)} + \alpha_{(i)} d_{(i)}, \tag{47}$$

$$r_{(i+1)} = r_{(i)} - \alpha_{(i)} d_{(i)},$$

$$\beta_{(i+1)} = \frac{r_{(i+1)}^T r_{(i+1)}}{r_{(i)}^T r_{(i)}} \tag{48}$$

$$d_{(i+1)} = r_{(i+1)} + \beta_{(i+1)} d_{(i)} \tag{49}$$

그림 30은 켤레 경사도 기법을 우리가 사용하고 있는 예제에 적용했을 때의 얻게 되는 성능을 보여주고 있다. 사실 이 기법에서 나타나는 기울기(gradient)는 서로 켤레가 아니기 때문에 “켤레 경사도(conjugate gradient)”라는 용어는 다소 잘못된 것이라고 할 수 있다. 이 기법은 기울기를 켤레화하여 탐색 방향을 얻기 때문에 “켤레화된 기울기(conjugated gradient)” 기법이라고 하는 것이 더 정확한 표현이다.

9 켤레 경사도의 수렴 분석

켤레 경사도는 n 번의 반복 후에 완료된다. 그렇다면 켤레 경사도 기법의 수렴 분석에 왜 관심을 두어야 할

까? 실제로 이 기법을 적용하면 부동소수점 반올림 오차 때문에 나머지(residual)이 점점 부정확해지며, 버림 오차(cancellation error) 때문에 탐색 벡터들 사이의 A-직교성이 점점 상실된다. 반올림 오차에 의한 문제는 최대 하강 기법에서 해결한 바와 같이 다룰 수 있지만, 버림 오차에 의한 문제는 쉽게 해결할 수 있는 문제가 아니다. 이처럼 오차에 의해 탐색 벡터들 사이의 켤레성이 깨어지는 문제 때문에 1960년대가 끝날 때까지 수학계는 이 켤레 경사도 기법에 관심을 두지 않았으며, 1970년대에 들어서 반복을 통해 해를 구하는 방법으로써 켤레 경사도 기법이 효과적이라는 결과가 발표된 이후에야 비로서 다시 관심을 받게 된다.

시간이 흘러 켤레 경사도에 대한 견해도 변했다. 오늘날 켤레 경사도가 실제로 적용되는 문제는 그 크기가 너무나 커서 n 번의 반복 수행이 불가능한 경우가 대부분이기 때문에, 수렴 분석이 중요한 의미를 가진다. 수렴 분석은 부동소수점 오차를 막기 위한 것이 아니라, 정확한 알고리즘으로 해결할 수 없는 문제들에 대해 켤레 경사도 기법이 유용하다는 점을 증명하기 위한 것이다.

켤레 경사도 기법의 첫 번 째 반복 단계는 최대 하강 기법의 첫 번 째 반복 단계와 동일하다. 따라서 6.1 절의 내용을 그대로 이용하여 켤레 경사도 기법이 단 한 번의 반복만에 수렴하는 조건을 설명할 수 있다.

9.1 완벽한 다항식 고르기

켤레 경사도 기법의 각 단계에서, 값 $e_{(i)}$는 $e_{(0)} + \mathcal{D}_i$로부터 선택되고, 이때 $\mathcal{D}_i$는 다음과 같은 부분공간임을 이미 살펴 보았다.

$$\begin{aligned}\mathcal{D}_i &= span\{r_{(0)}, Ar_{(0)}, A^2r_{(0)}, \cdots, A^{i-1}r_{(0)}\} \\ &= span\{Ae_{(0)}, A^2e_{(0)}, A^3e_{(0)}, \cdots, A^ie_{(0)}\}.\end{aligned}$$

이와 같은 크릴로프(Krylov) 부분공간은 또 한 가지 기분좋은 특성을 가진다. 고정된 i에 대해, 오차항은 다음과 같이 표현된다.

$$e_{(0)} = \left(I + \sum_{j=1}^{i} \psi_j A^j\right) e_{(0)}$$

계수 ψ_j는 값 $\alpha_{(i)}, \beta_{(i)}$와 관계되며, 정확한 관계성은 여기서 중요하지 않다. 중요한 것은 7.3에 나와 있는 증명, 즉, 켤레 경사도 기법이 $||e_{(i)}||_A$를 최소화하는 계수 ψ_j를 선택한다는 사실이다.

위 식에서 괄호안의 표현은 다항식으로 표현될 수 있다. $P_i(\lambda)$를 차수 i의 다항식이라 하자. P_i은 인자로 값이나 행렬이 올 수 있으며, 이들은 동일하게 계산된다. 예를 들어, 만일 $P_2(\lambda) = 2\lambda^2 + 1$이라고 하면, $P_2(A) = 2A^2 + I$가 된다. 이러한 융통성 있는 표기법은 $P_i(A)v = P_i(\lambda)v (Av = \lambda v, A^2v = \lambda^2 v, \cdots .)$와 같이 고유벡터에서 유용하다.

만일 $P_i(0) = 1$이 되도록 하려고 하면, 오차항은 다음과 같이 표현될 수 있다.

$$e_{(i)} = P_i(A)e_{(0)},$$

켤레 경사도 기법은 ψ_j 계수를 선택할 때, 이 다항식을 선택한다. 이 다항식을 $e_{(0)}$에 적용할 때의 효과에 대해 조사하자. 최대 하강 기법의 분석에서와 마찬가지로, $e_{(0)}$를 직교단위고유벡터들의 선형 조합으로 표현한다.

$$e_{(0)} = \sum_{j=1}^{n} \xi_j v_j,$$

그리고 다음의 식들을 이끌어낼 수 있다.

$$e_{(i)} = \sum_j \xi_j P_i(\lambda_j) v_j$$

$$Ae_{(i)} = \sum_j \xi_j P_i(\lambda_j) \lambda_j v_j$$

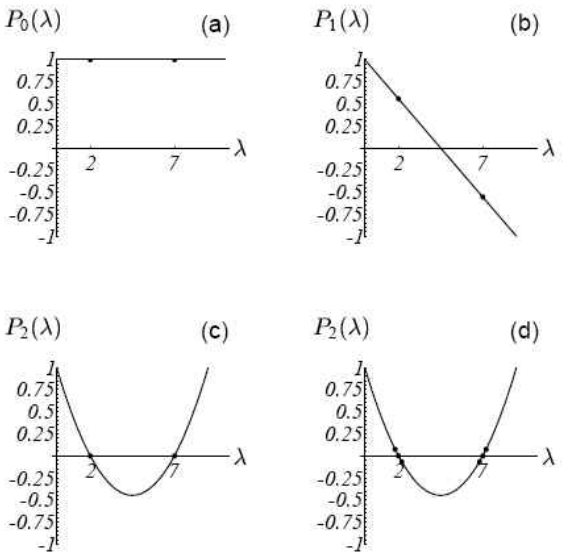
$$\|e_{(i)}\|^2 A = \sum_j \xi_j^2 [P_i(\lambda_j)]^2 \lambda_j.$$

켤레 경사도 기법은 이 표현식을 최소화하는 다항식을 찾으며, 이 방법의 수렴 속도는 아무리 좋아도 최악의 고유벡터가 수렴하는 정도보다 좋을 수가 없다. $\Lambda(A)$가 A의 고유값들의 집합이라 하면 오차의 에너지 놈(norm)은 다음과 같다.

$$\begin{aligned} \|e_{(i)}\|_A^2 &\leq \min_{P_i} \max_{\lambda \in \Lambda(A)} [P_i(\lambda)]^2 \sum_j \xi_j^2 \lambda_j \\ &= \min_{P_i} \max_{\lambda \in \Lambda(A)} [P_i(\lambda)]^2 \|e_{(0)}\|_A^2 \end{aligned} \tag{50}$$

그림31은 고유값 2와 7을 가지는 예제에 대해 위의 표현식을 최소화하는 다항식을 몇 가지 종류의 차수에 대해 보여주고 있다. $P_0(0) = 1$을 만족하는 차수 0의 다항식은 하나 밖에 없으며, 이는 31(a)에서 보는 바와 같이 $P_0(\lambda) = 1$이다. 차수 1의 최적 다항식은 $P_1(\lambda) = 1 - 2x/9$이고, 그림31(b)에서 볼 수 있다. $P_1(2) = 5/9$, $P_1(7) = -5/9$이고, 켤레 경사도의 첫번째 반복 후 오차항의 에너지놈이 초기치인 $5/9$보다 크지 않음을 알 수 있다. 그림31(c)는 두번의 반복 후 식50의 결과가 0이 됨을 보여준다. 이것은 2차 다항식이 세 점($P_2(0) = 1, P_2(2) = 0, P_2(7) = 0$)을 지나도록 할 수 있기 때문이다. 일반적으로 차수 n의 다항식은 $n+1$의 점들을 지나로록 만들 수 있으므로 n개의 서로 다른 고유값들을 수용할 수 있다.

지금까지의 논의를 통해 켤레경사도기법이 n번의 반복후에 정확한 결과를 얻는다는 것을 더 잘 알 수 있다: 더우기 같은 고유값들이 존재한다면 켤레경사도기법이 더 빠르게 수렴한다는 것에 대한 증명이다. 무한 수준의 부동소수점 정밀도를 가진다고 할 때, 정확한 해를 계산하기 위해 요구되는 반복횟수는 기껏해야 서로 다른 고유값들의 수가 된다. (이른 종료에 대한 한가지 다른 가능성이 있다: $x_(0)$가 A의 일부 고유벡터들과 이미 $A-$직교를 이루고 있는 경우이다. 만약 $x_(0)$를 전개할 때 나타나지 않는 고유벡터들은 그에 해당하는 고유값들 역시 식 50에 나타나지 않을 것이다. 그러나 앞서 경고한 바와 같이, 이러한 고유 벡터들 역시 부동 소수점 반올림 오차에 의해 다시 나타나게 될 것이다.)



[그림 31] i번의 반복을 수행한 이후 켤레 경사도 기법이 얼마나 수렴했는지는 $P_i(0) = 1$라는 주어진 제한 조건에서 i 차 다항식 P_i가 각각 고유값에서 얼마나 0에 가까운지에 달려있다.

고유값들이 λ_{min}과 λ_{max} 사이에 불규칙하게 분포되어 있는 것보다, 그림31(d)와 같이 근처에 모여 있는 경우 켤레경사도기법이 더욱 빨리 수렵할 수 있음을 알 수 있다. 이것은 켤레경사도기법에서 식50의 값을 작게 만드는 다항식을 선택하는 것이 더욱 쉬어지기 때문이다.

만일 행렬A의 고유값들의 특성을 알고 있다면 빠른 수렴을 이끌어 내는 다항식을 제안할 수도 있지만, 여기서는 가장 일반적인 경우, 즉 고유값들이 λ_{min}과 λ_{max} 사이에 고르게 분포하고, 중복되는 고유값의 수가 적어 많은 수의 서로 다른 고유값이 나타나며, 부동 소수점 반올림 오차가 발생하는 경우를 가정한다.

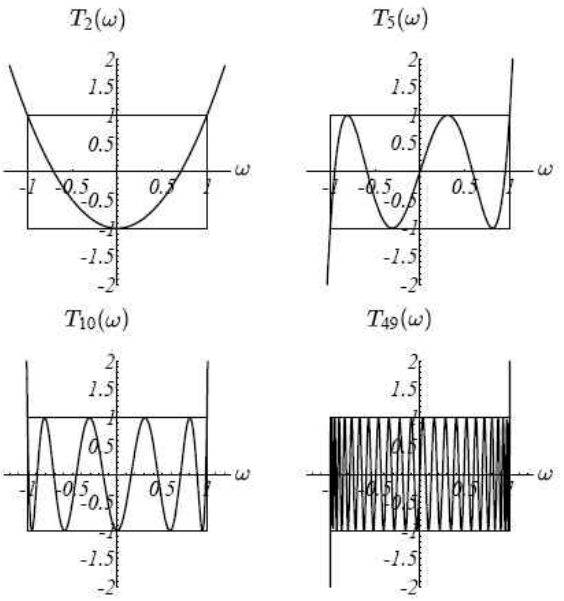
9.2 체비쉐프(Chebyshev) 다항식

한 가지 유용한 접근방법은 유한 개의 점에 대해서가 아니라 범위$[\lambda_{min}, \lambda_{max}]$에서 식50을 최소화하는 것이다. 이를 수행할 수 있는 다항식들은 체비쉐프(Chebyshev) 다항식을 기반으로 한다.

차수 i의 체비쉐프(Chebyshev) 다항식은 다음과 같다.

$$T_i(\omega) = \frac{1}{2}[(\omega + \sqrt{\omega^2 - 1})^i + (\omega - \sqrt{\omega^2 - 1})^i].$$

(이 식이 다항식처럼 보이지 않는다면, i가 1이나 2일 경우에 대해 풀어보라.) 몇가지 체비쉐프(Chebyshev) 다항식들이 그림32에 그려져 있다. 체비쉐프(Chebyshev) 다항식들은 정의역 $\omega \in [-1, 1]$에서 $|T_i(\omega) \leq 1|$인 특성을 가지며 (사실 -1과 1의 사이에서 진동한다), 모든 다항식에 있어서 $|T_i(\omega)|$의 값은 $\omega \notin [-1, 1]$인 정의역에



[그림 32] 차수 2, 5, 10, 49의 체비쉐프(Chebyshev) 다항식들

서 최대치를 갖게 된다. 엄밀하지 않은 표현으로 이야기 하면, $|T_i(\omega)|$의 값이 그림에 나타나 있는 상자의 밖에서 가능한한 가장 빠르게 증가한다.

수식 50이 다음과 같은 $P_i(\lambda)$를 선택함에 의해 최소화 되는 것은 부록 $C3$에서 설명된다.

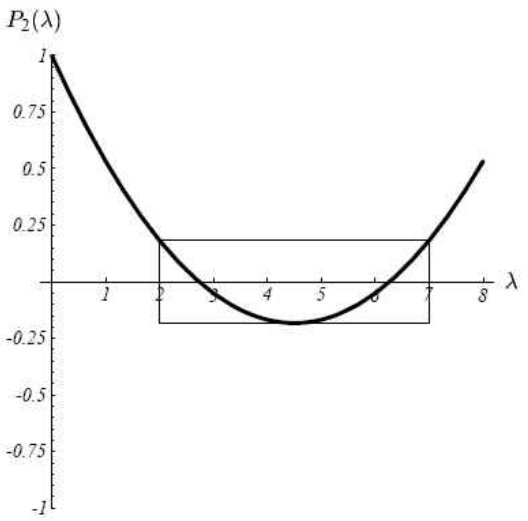
$$P_i(\lambda) = \frac{T_i(\frac{\lambda_{max}+\lambda_{min}-2\lambda}{\lambda_{max}-\lambda_{min}})}{T_i(\frac{\lambda_max+\lambda_min}{\lambda_max-\lambda_min})}$$

이 다항식은 영역 $\lambda_{min} \leq \lambda \leq \lambda_{max}$에서 체비쉐프(Chebyshev) 다항식의 진동 특성을 가진다(그림33 참조). 분모는 $P_i(0) = 1$이라는 요구사항을 충족하게 하며, 분자는 λ_{min}과 λ_{max} 사이의 구간에서 최대값 1을 가진다. 따라서 식50으로부터 다음의 식이 성립한다.

$$\begin{aligned}\|e_{(i)}\|_A &\leq T_i\left(\frac{\lambda_{max}+\lambda_{min}}{\lambda_{max}-\lambda_{min}}\right)^{-1}\|e_{(0)}\|_A \\ &= T_i\left(\frac{\kappa+1}{\kappa-1}\right)^{-1}\|e_{(0)}\|_A \\ &= 2\left[\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^i + \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i\right]^{-1}\|e_{(0)}\|_A. \end{aligned} \quad (51)$$

사각형 괄호안의 두번째 가수(加數, addend)는 i가 증가함에 따라 0으로 수렴한다. 따라서 켤레경사도의 수렴은 좀 더 약한 부등식으로 다음과 같이 표현하는 것이 일반적이다.

$$\|e_{(i)}\|_A \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i\|e_{(0)}\|_A \quad (52)$$



[그림 33] 일반적인 경우에서 영역 $\lambda_{min} = 2$와 $\lambda_{max} = 7$에서 식 50을 최소화하는 다항식 $P_2(\lambda)$. 이 곡선은 차수 2의 Chebyshev 다항식의 크기를 변형한 것이다. 두 번의 반복 이후의 오차항의 에너지 놈(norm)은 초기값의 0.183배를 넘지 않는다. 2개의 고유값만이 존재하는 것으로 이미 알려져 있던 그림 31(c)와 비교하라.

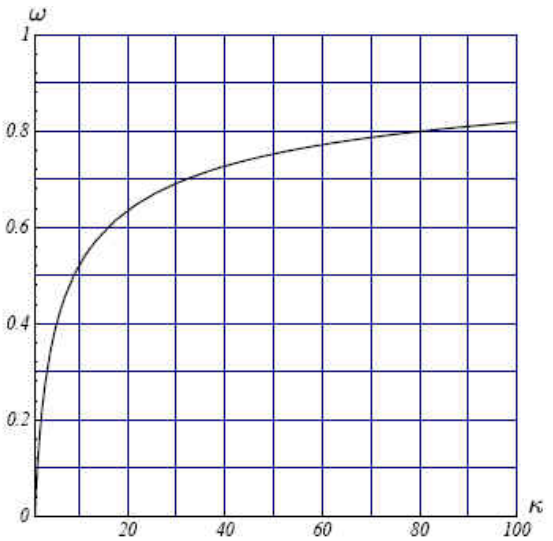
켤레경사도의 첫번째 단계는 최대하강의 첫번째 단계와 동일하다. 식51에서 $i = 1$로 두면, 최대하강에 관한 수렴 결과인 식28을 얻는다. 이것은 그림 31(b)에 그려진 선형 다항식의 경우다.

그림 34는 켤레경사도의 반복마다의 수렴에 대한 도표이다. 실제로 적용했을 때, 켤레경사도는 좋은 고유값 분포나 좋은 시작 위치 등의 이유 때문에 대개의 경우 식 52가 시사하는 수렴 속도보다 더 빠르게 수렴한다. 식 52와 식 28을 비교하면, 켤레 경사도(conjugate gradient) 기법이 최대 하강 (steepest descent) 기법보다 더 빠르게 수렴하는다는 것이 명확하다 (그림 35 참조). 그러나 켤레경사도의 모든 반복 단계가 최대하강 기법보다 더 빠르게 수렴하는 것은 아니다; 예를 들면, 켤레 경사도 기법의 첫번째 단계는 최대 하강 기법의 첫번째 단계와 동일하다. 식52에서 곱해져 있는 수 2에 의해서 켤레 경사도 기법이 일부 반복 단계에서 약간의 속도 저하를 보일 수 있게 된다.

10 복잡도

최대하강이나 켤레경사도 기법의 각 단계에서 가장 많은 계산시간을 필요로 하는 연산은 행렬-벡터 곱셈이다. 일반적으로, 행렬-벡터 곱셈은 행렬의 0이 아닌 원소의 수를 m이라 할 경우 $\mathcal{O}(m)$의 계산 시간을 필요로한다. 이 문서의 1 절에서 나열된 다양한 문제들에서 보통 A는 희소행렬이고 $m \in \mathcal{O}(n)$이다.

ε의 배수로 오차의 크기를 줄이기 위해 충분한 반복을 수행하기를 원한다고 가정하자: 즉, $\|e_{(i)}\| \leq \varepsilon \|e_{(0)}\|$.



[그림 34] 상태 수의 함수로서의 켤레경사도(반복마다)의 수렴. 그림 20과 비교하라.

식 28은 최대 하강 기법을 사용하여 이 한계를 만족하기 위해 요구되는 최대의 반복 횟수가 다음과 같음을 보여준다.

$$i \leq \left\lceil \frac{1}{2}\kappa \ln\left(\frac{1}{\epsilon}\right)\right\rceil,$$

한 편, 식 52에 의하면 켤레 경사도 기법이 요구하는 최대의 반복횟수가 다음과 같다.

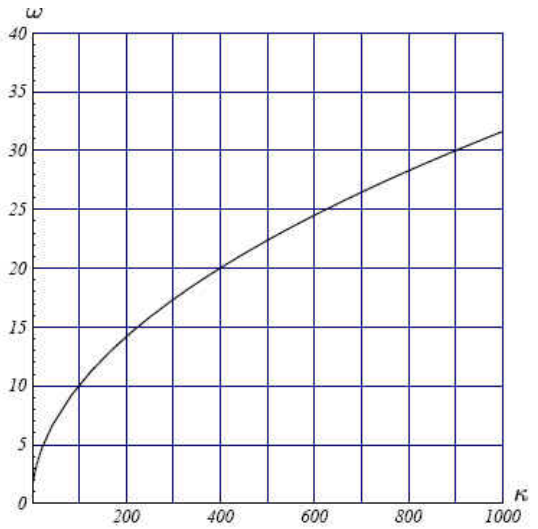
$$i \leq \left\lceil \frac{1}{2}\sqrt{\kappa} \ln\left(\frac{2}{\epsilon}\right)\right\rceil.$$

최대하강은 $\mathcal{O}(m\kappa)$의 시간복잡도를 가지며, 켤레경사도는 $\mathcal{O}(m\sqrt{\kappa})$의 시간복잡도를 가진다는 결론이다. 두 알고리즘 모두 공간 복잡도는 $\mathcal{O}(m)$이다.

$d-$차원 정의역에서 이차 타원 경계값 문제들을 유한차분과 유한요소법으로 근사할 때의 복잡도는 종종 $\kappa \in \mathcal{O}(n^{2/d})$이 된다. 따라서, 최대 하강 기법은 이차원 문제들에 대해 $\mathcal{O}(n^2)$의 시간복잡도를 가지지만 켤레경사도는 $\mathcal{O}(n^{3/2})$이다. 또한 삼차원 문제들에 대해 최대하강은 $\mathcal{O}(n^{5/3})$의 복잡도지만 켤레 경사도 기법은 $\mathcal{O}(n^{4/3})$의 시간복잡도를 가진다.

11 시작과 종료

앞서 설명한 최대하강과 켤레경사도 알고리즘에서, 몇가지 세부적인 것들이 생략되었다; 특히, 시작점을 어떻게 선택하고 언제 종료하느냐의 문제이다.



[그림 35] 켤레경사도의 한번의 반복에 일치하기 위해 요구되는 최대하강의 반복횟수.

11.1 시작

시작에 관하여 설명할 것은 많지 않다. 만일 x의 값에 대해 대략적인 추정치가 있다면, 그것을 시작 값 $x_{(0)}$로 사용하라. 그렇지않다면, $x_{(0)} = 0$으로 두어라; 선형시스템을 푸는 경우 최대 하강이나 켤레 경사도 기법은 결국 수렴하게 되어 있다. 비선형 최소화(14 절에서 설명)는 여러개의 지역최소가 존재할 수 있고 시작 위치의 선택에 따라 어느 최소점에 수렴할지가 결정되며, 어떤 경우에는 수렴이 이루어질지 그렇지 않을지를 결정하기 때문에 까다로운 문제가 된다.

11.2 종료

최대하강이나 켤레경사도가 최소점에 도달할 때 나머지(residual)는 0이 되고, 만일 식11이나 48이 이후의 한 번의 반복을 더하게 한다면, 분모가 0이 된다. 따라서 나머지(residual)이 0이 되면 즉시 계산이 종료되어야 한다. 나머지(residual)의 점화식 형태(47) 계산에서 반올림 오차가 누적되는 경우에는 나머지가 0이 아닌데도 0으로 계산될 수가 있다; 이 문제는 식45를 이용하여 계산을 새로 시작함으로서 해결될 수 있다.

보통의 경우 수렴이 완전히 이루어지기 전에 알고리즘이 종료되기를 원한다. 오차항을 이용할 수 없기 때문에 나머지(residual)의 크기가 사전에 명시된 값 이하로 내려가면 종료하는 것이 보통이다; 종종 이 값으로 초기 나머지(residual) 값에 0보다 작은 값 ε을 곱해 얻은 작은 값을 사용한다. ($\|r_{(i)}\| < \varepsilon\|r_{(0)}\|$). 예제 코드는 부록 B에서 볼 수 있다.

12 전처리

전처리는 행렬의 조건 수(condition number)를 개선하기 위한 기법이다. 행렬 M이 행렬 A를 근사하는 대칭이면서 양의 정부호인 행렬이고, 역행렬을 쉽게 계산할 수 있다고 하자. 그렇다면 다음 식을 풀어서 간접적으로 $Ax = b$를 풀 수 있다.

$$M^{-1}Ax = M^{-1}b. \tag{53}$$

만일 $\kappa(M^{-1}A) \ll \kappa(A)$이거나 $M^{-1}A$의 고유값들이 A의 고유값들보다 더 잘 모여 있다면, 원래의 문제보다 식 53의 문제가 반복법을 통해 더 잘 수렴한다. 문제는 행렬 M이나 A이가 대칭이고 양의 정부호 행렬이라 하더라도 행렬 $M^{-1}A$은 일반적으로 그렇지 않다는 것이다.

대칭이며 양의 정부호인 모든 행렬 M에 대해 $EE^T = M$인 특성을 가지는 행렬 E가 (유일한 것은 아니지만) 존재하기 때문에, 이 어려움은 피할 수 있다. (그런 행렬 E는 촐레스키(Cholesky) 분해와 같은 방법으로 구할 수 있다.) 행렬 $M^{-1}A$와 $E^{-1}AE^{-T}$는 동일한 고유값들을 가진다. 이것이 성립하는 이유는 v가 고유값 λ를 가지는 행렬 $M^{-1}A$의 고유벡터일 때, E^Tv는 다음과 같이 고유값 λ를 가지는 행렬 $E^{-1}AE^{-T}$의 고유벡터이기 때문이다.

$$(E^{-1}AE^{-T})(E^Tv) = (E^TE^{-T})E^{-1}Av = E^TM^{-1}Av = \lambda E^Tv.$$

선형 시스템 $Ax = b$는 다음의 문제로 변형될 수 있다.

$$E^{-1}AE^{-T}\widehat{x} = E^{-1}b, \qquad \widehat{x} = E^Tx,$$

이 문제에서 우리는 우선 $\widehat{x}$에 대해 풀고 나서 x에 대해 푼다. 행렬 $E^{-1}AE^{-T}$가 대칭이면서 양의 정부호이기 때문에, $\widehat{x}$는 최대하강이나 켤레경사도에 의해 구할 수 있다. 켤레 경사도 기법을 사용하여 이 문제를 해결하는 과정을 "변환된 전처리 켤레 경사도 기법(Transformed preconditioned conjugate gradient method)"이라 부른다.

$$\widehat{d}_{(0)} = \widehat{r}_{(0)} = E^{-1}b - E^{-1}AE^T\widehat{x}_{(0)},$$

$$\alpha_{(i)} = \frac{\widehat{r}_{(i)}^T\widehat{r}_{(i)}}{\widehat{d}_{(i)}^TE^{-1}AE^{-T}\widehat{d}(i)},$$

$$\widehat{x}_{(i+1)} = \widehat{x}_{(i)} + \alpha_{(i)}\widehat{d}_{(i)},$$

$$\widehat{r}_{(i+1)} = \widehat{r}_{(i)} - \alpha_{(i)}E^{-1}AE^{-T}\widehat{d}_{(i)},$$

$$\beta_{(i+1)} = \frac{\widehat{r}_{(i+1)}^T\widehat{r}_{(i+1)}}{\widehat{r}_{(i)}^T\widehat{r}_{(i)}},$$

$$\widehat{d}_{(i+1)} = \widehat{r}_{(i+1)} + \beta_{(i+1)}\widehat{d}_{(i)}.$$

이 방법은 행렬 E를 계산해야 한다는 문제가 있다. 그러나 약간만 주의해서 변수를 치환하면 행렬 E를 제거할 수 있다. $\widehat{r}_{(i)} = E^{-1}r_{(i)}$와 $\widehat{d}_{(i)} = E^T d_{(i)}$로 두고 $\widehat{x}_{(i)} = E^T x_{(i)}$와 $E^{-T}E^{-1} = M^{-1}$라는 항등식을 적용하면, 다음과 같이 "변환되지 않은 전처리 켤레 경사도 기법(Untransformed preconditioned conjugate gradient method)"를 얻을 수 있다.

$$r_{(0)} = b - Ax_{(0)},$$

$$d_{(0)} = M^{-1}r_{(0)},$$

$$\alpha_{(i)} = \frac{r_{(i)}^T M^{-1} r_{(i)}}{d_{(i)}^T A d_{(i)}},$$

$$x_{(i+1)} = x_{(i)} + \alpha_{(i)} d_{(i)},$$

$$r_{(i+1)} = r_{(i)} - \alpha_{(i)} A d_{(i)},$$

$$\beta_{(i+1)} = \frac{r_{(i+1)}^T M^{-1} r_{(i+1)}}{r_{(i)}^T M^{-1} r_{(i)}},$$

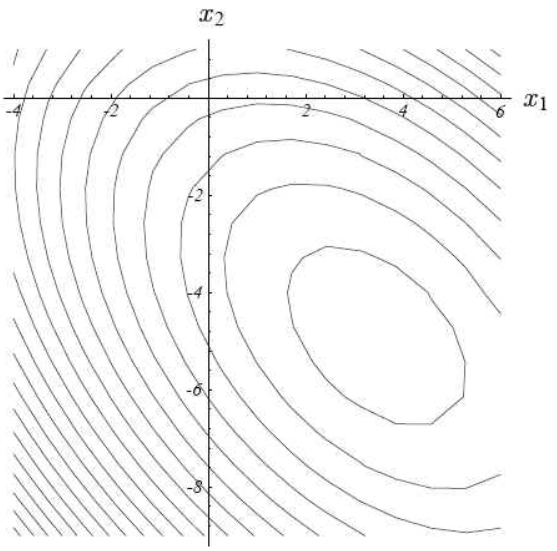
$$d_{(i+1)} = M^{-1}r_{(i+1)} + \beta_{(i+1)} d_{(i)}.$$

행렬 E는 이 방정식에서는 나타나지 않는다; 여기서는 행렬 M^{-1}만 필요하다. 같은 방법에 의해, 행렬 E를 사용하지 않는 전처리 최대 하강 기법을 이끌어내는 것도 가능하다.

전처리자 M의 유효성은 행렬 $M^{-1}A$의 조건 수에 의해 결정되고, 경우에 따라 고유값들의 클러스트링 정도에 의해 결정된다. 문제는 매 번 반복이 일어날 때마다 한 번씩 $M^{-1}r_{(i)}$ 곱셈을 수행해야 하기 때문에 발생하는 비용을 상쇄할 수 있을 정도로 A를 잘 근사하여 수렴 속도를 높일 수 있는 전제조건자를 찾는것이다. (명시적으로 M 또는 M^{-1}을 계산할 필요는 없다; 필요한 것은 단지 행렬 M^{-1}을 벡터에 적용했을 때 발생하는 효과를 계산하는 것이다.) 이러한 제한조건 속에서, 놀랍게도 풍부한 가능성들이 있는데 여기서는 수박 겉핥기와 같이 일부만 다룰 것이다.

직관적으로, 전처리는 이차 형식을 더욱 구형에 가깝게 늘이려는 시도이다. 따라서 고유값들은 서로에게 근접한다. 완전 전처리자(perfect preconditioner)의 한 예는 $M = A$이다; 이 전처리자에 대해, $M^{-1}A$의 조건수는 1이 되며, 이차 형식은 완벽한 구형이 되어 단 한 번의 반복만으로도 해를 구할 수 있다. 그러나, 불행하게도, 전처리 단계 자체가 $Mx = b$를 푸는 것이어서 이 전처리자는 결코 유용한 전처리자가 아니다.

가장 간단한 전처리자는 대각 성분 값들이 행렬 A과 동일한 대각 행렬이다. 이 전처리자를 적용하는 과정은 대각 전처리 또는 야코비 전처리라는 이름으로 알려져 있으며, 이 과정은 좌표축 방향으로 이차 형식의 크기를 변경하는 것과 같다. (비교를 하면, 완전 전처리자 $M = A$는 고유벡터축을 따라 이차형식의 크기를 조절한다.) 대각 행렬의 역행렬을 구하는 것은 매우 간단하지만, 이 행렬은 보통 평범한 수준의 전처리자일 뿐이다. 대각 전처리 이후에 예제 문제의 이차 형식이 가지는 등고선들이 그림36에 보여진다. 그림3과 비교하면, 일정



[그림 36] 예제 문제를 대각 전처리한 이후 얻는 이차 형식의 등고선들

정도 개선되었다는 것이 명백하다. 3.5였던 조건수는 대략 2.8까지 향상된다. 물론, 이러한 개선은 $n \gg 2$인 선형 시스템에서 더 유용하다.

더 정교한 전처리자는 불완전 촐레스키 전처리(incomplete Cholesky preconditioning)이다. 촐레스키 분해는 행렬 A를 LL^T의 형태로 분해하는 기술이다. 여기서 L은 아래삼각행렬(lower triangular matrix)이다. 불완전한 촐래스키 분해는 채움이 허용되지 않는 변형이다; A는 $\widehat{L}\widehat{L}^T$와 같은 행렬곱 형태로 근사되는데, 이때 $\widehat{L}$는 A와 동일한 형태로 0이 아닌 원소가 나타나야 한다; L의 다른 원소들은 모두 버려진다. $\widehat{L}\widehat{L}^T$를 전처리자로 사용하기 위해, $\widehat{L}\widehat{L}^T w = z$의 해가 역치환 방법으로 계산된다. ($\widehat{L}\widehat{L}^T$의 역행렬은 결코 명시적으로 계산할 필요가 없다). 불행하게도 불완전한 촐래스키 전처리가 항상 안정적이지는 않다.

많은 전처리자들이 (그 중 일부는 매우 복잡하다) 개발되었다. 전처리자를 어떤 것을 선택하든지는 간에, 대규모 문제에서 켤레 경사도를 사용할 때에는 거의 언제나 전처리자를 이용하여 문제를 풀어야 한다는 사실이 일반적으로 받아들여지고 있다.

13 켤레 경사도

켤레 경사도는 대칭이며서 양의 정부호가 아닌, 그리고 심지어 정방행렬이 아닌 경우에도 사용될 수 있다. 다음과 같은 최소 자승 문제를 보자.

$$\min_x \|Ax - b\|^2 \tag{54}$$

이 문제의 해는 식54의 미분값을 0으로 둠으로서 구할 수 있다.

$$A^T Ax = A^T b. \tag{55}$$

만일 A가 정방형의 비특이(non-singular) 행렬이라 하면, 식 55의 해는 $Ax = b$의 해와 동일하다. 만일 A가 정방 행렬이 아니고 $Ax = b$가 변수보다 더많은 선형 독립 방정식을 가지는 *과대 제약(overconstrained)*인 경우, $Ax = b$의 해가 존재할 수도 있고 존재하지 않을 수도 있다. 하지만 각 선형방정식의 오차제곱을 합한 수식54을 최소화하는 x 값을 찾는 것은 항상 가능하다.

$A^T A$는 대칭이고 양(임의의 x에 대해, $x^T A^T Ax = \|Ax\|^2 \geq 0$)이다. 만일 $Ax = b$가 과소제약(underconstrained)이 아니면, $A^T A$는 비특이 행렬이고, 최소하강이나 켤레 경사도 기법과 같은 방법이 식55를 풀기 위해 사용될 수 있다. 단지 한가지 문제는 $A^T A$의 조건수가 A가 가지는 조건수의 제곱이고, 따라서 수렴이 느리다는 것이다.

여기서 기술적으로 중요한 사항은 행렬 $A^T A$이 A보다 덜 희소 행렬이기 때문에 명시적으로 형성되지 않는다는 것이다. 대신, Ad를 구한 뒤에 $A^T Ad$를 구함으로써, $A^T A$에는 d가 곱한 상태가 된다. 또한 Ad를 자기 자신과 내적하여 $d^T A^T Ad$ (식46)를 계산하면 수치적인 안정성이 향상된다.

14 비선형 켤레 경사도 기법

켤레 경사도 기법은 이차 형식(quadratic form)의 최소점을 찾는 데에 사용될 뿐만 아니라 기울기 f'를 계산할 수 있는 어떤 연속함수 $f(x)$에 대해서도 최소화를 위해 사용될 수 있다. 이는 공학 설계, 신경망 학습, 그리고 비선형 회귀(regression) 등과 같은 다양한 종류의 최적화 문제에 응용될 수 있다.

14.1 비선형 켤레 경사도 기법의 개요

비선형 켤레 경사도 기법을 유도할 때 선형 알고리즘과 비교하여 세 가지 변화가 있다: 나머지(residual)에 대한 재귀적 수식이 사용될 수 없으며, 간격 α를 계산하는 것이 더욱 복잡하며, β 값에 대해 몇 가지 다른 선택 방법이 있다는 것이다.

비선형 켤레 경사도 기법에서, 나머지는 항상 기울기의 부호를 반대로 한 것으로 설정된다; $r_{(i)} = -f'(x_{(i)})$. 탐색 방향은 선형 켤레 경사도 기법처럼 나머지에 대한 그램-슈미트(Gram-Schmidt) 켤레화를 이용하여 계산된다. 이 탐색 방향을 따라 직선 탐색을 하는 것은 선형인 경우에 비해 더욱 어려우며, 다양한 절차가 사용될 수 있다. 선형 기법과 같이, $f(x_{(i)} + \alpha_{(i)} d_{(i)})$를 최소화하는 $\alpha_{(i)}$의 값은 기울기가 탐색 방향에 직교하도록 함으로써 찾을 수 있다. 이를 위해 $[f(x_{(i)} + \alpha_{(i)} d_{(i)})]^T d_{(i)}$이 0이 되는 값을 찾는 어떤 종류의 알고리즘이라도 사용할 수 있다.

선형 켤레 경사도 기법에서는 β의 값에 대한 몇 개의 동치(同値) 표현들이 있다. 비선형 켤레 경사도 기법에서는 서로 다른 이 표현들이 더 이상 동치가 아니다; 연구자들은 최선의 선택이 무엇인지 아직도 조사하

고 있다. 두 가지 선택 가운데 하나는 계산의 용이성 때문에 선형 기법에 사용되었던 플레쳐-리브스(Fletcher-Reeves) 식이며, 다른 하나는 폴락-리비에르(Polak-Ribière) 식이다:

$$\beta^{FR}_{(i+1)} = \frac{r^T_{(i+1)} r_{(i+1)}}{r^T_{(i)} r_{(i)}}, \qquad \beta^{PR}_{(i+1)} = \frac{r^T_{(i+1)} (r_{(i+1)} - r_{(i)}}{r^T_{(i)} r_{(i)}}.$$

플레쳐-리브스 기법은 시작점이 원하는 최소점과 충분히 가까울 때에 수렴하며, 폴락-리비에르 기법은 드물긴 하지만 영원히 수렴하지 않고 반복할 수 있다. 그러나 폴락-리비에르 기법이 종종 더 빠르게 수렴한다.

다행히 폴락-리비에르 기법의 수렴은 $\beta = \max\{\beta^{PR}, 0\}$으로 선택함으로써 보장될 수 있다. 이 값을 사용하는 것은 $\beta^{PR} < 0$인 경우에 켤레 경사도 기법을 새로 시작하는 것과 같다. 켤레 경사도를 새로 시작하는 것은 이전의 탐색 방향들을 모두 잊고 가장 급하게 하강하는 경사 방향을 따라 새롭게 켤레 경사도 기법을 수행하는 것이다.

비선형 켤레 경사 기법의 개요는 다음과 같다:

$$d_{(0)} = r_{(0)} = -f(x_{(0)}),$$

$$\text{Find } \alpha_{(i)} \text{ that minimizes } f(x_{(i)} + \alpha_{(i)} d_{(i)}),$$

$$x_{(i+1)} = x_{(i)} + \alpha_{(i)} d_{(i)},$$

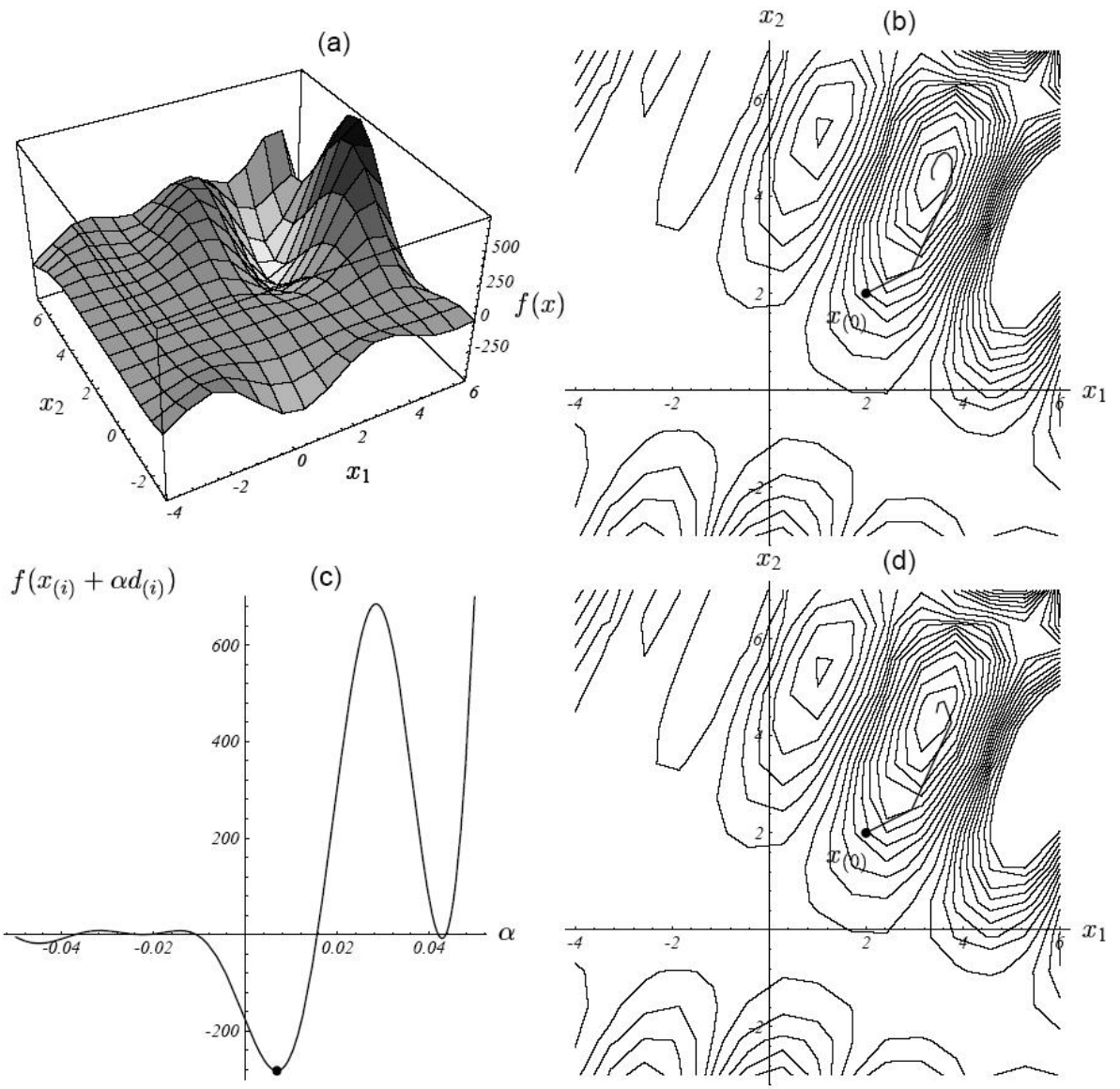
$$r_{(i+1)} = -f'(x_{(i+1)}),$$

$$\beta_{(i+1)} = \frac{r^T_{(i+1)} r_{(i+1)}}{r^T_{(i)} r_{(i)}} \quad or \quad \beta_{(i+1)} = \max\left\{ \frac{r^T_{(i+1)} (r_{(i+1)} - r_{(i)})}{r^T_{(i)} r_{(i)}}, 0 \right\},$$

$$d_{(i+1)} = r_{(i+1)} + \beta_{(i+1)} d_{(i)}.$$

비선형 켤레 경사도 기법은 선형 켤레 경사도 기법의 수렴 보장성들이 거의 적용되지 않는다. 함수 f가 이차 함수와 유사하지 않으면 않을수록, 탐색 방향들은 더 빨리 켤레성(conjugacy)을 잃는다. (비선형 켤레 경사도 기법에서도 "켤레성"이라는 용어가 여전히 의미를 가진다는 것이 곧 명백해 질 것이다.) 또 다른 문제는 일반적인 함수 f는 많은 지역 최소점을 가질 수 있다는 것이다. 켤레 경사도 기법은 전역 최소점에 수렴할 것이라 보장할 수 없으며, 심지어 f가 하한(lower bound)가 없는 경우 지역 최소조차 찾지 못할 수도 있다.

그림 37은 비선형 켤레 경사도 기법을 그림으로 보이고 있다. 그림 37(a)는 다수의 지역 최소를 가지고 있는 함수이다. 그림 37(b)는 플레쳐-리브스(Fletcher-Reeves) 식을 사용한 비선형 켤레 경사도 기법의 수렴을 보이고 있다. 이 예에서 선형의 경우처럼 효과적이지 않다; 이 함수는 최소화하기가 매우 어렵다. 그림 37(c)는 그림 37(b)의 최초 직선 탐색 방향을 따라 펴면을 절단한 면을 보이고 있다. 이 절단면에 다수의 최소점이 존재함에 유의하라; 직선 탐색은 가까운 최소점에 해당하는 α 값을 찾는다. 그림 37(d)는 폴락-리비에르(Polak-Ribière) 기법의 더 나은 수렴성을 보인다.



[그림 37] 비선형 켤레 경사도 기법의 수렴. (a) 다수의 지역 최소와 최대를 가진 복잡한 함수. (b) 플레쳐-리브스(Fletcher-Reeves) 기법의 수렴 경로. 선형 기법과 달리 두 단계만에 수렴하지 않는다. (c) 최초의 직선 탐색을 따른 표면 절단 결과. (d) 폴락-리비에르(Polak-Ribière) 기법의 수렴 경로.

14.2 일반적 직선 탐색

f'의 값에 따라, ${f'}^T d$가 0이 되는 해를 찾는 빠른 알고리즘을 사용하는 것이 가능하다. 예를 들어, f'가 α의 다항식일 때, 다항식의 근을 구하는 효율적인 알고리즘을 사용할 수 있다. 그러나 우리는 범용 알고리즘에 대해서만 고려할 것이다.

두 종류의 반복 기법으로 뉴턴-랩슨(Newton-Raphson) 기법과 할선법(割線法 secant method)이 있다. 두 기법은 모두 f가 이차 미분이 가능하여야 한다. 뉴턴-랩슨 기법은 $f(x+\alpha d)$가 α에 대하여 2차 미분을 계산할 수 있어야 한다.

뉴턴-랩슨 기법은 테일러(Taylor) 급수 근사를 이용한다.

$$\begin{aligned} f(x+\alpha d) &\approx f(x) + \alpha \left[\frac{d}{d\alpha} f(x+\alpha d)\right]_{\alpha=0} + \frac{\alpha^2}{2} && (56) \\ &= f(x) + \alpha [f'(x)]^T d + \frac{\alpha^2}{2} d^T f''(x) d \\ \frac{d}{d\alpha} f(x+\alpha d) &\approx [f'(x)]^T d + \alpha d^T f''(x) d && (57) \end{aligned}$$

이때 $f''(x)$는 다음과 같은 헤시안(Hessian) 행렬이다.

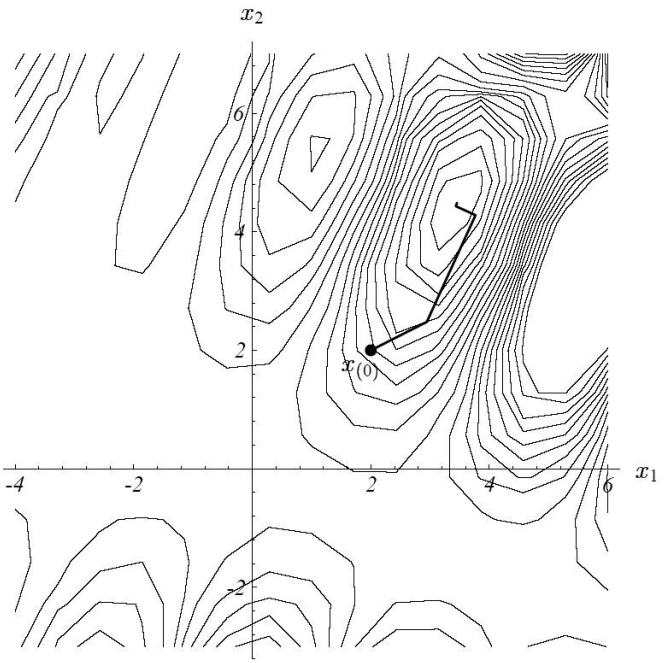
$$f''(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}.$$

함수 $f(x+\alpha d)$는 식 57가 0이 되도록 함으로써 근사적으로 최소화할 수 있으며 이를 통해 다음을 얻는다.

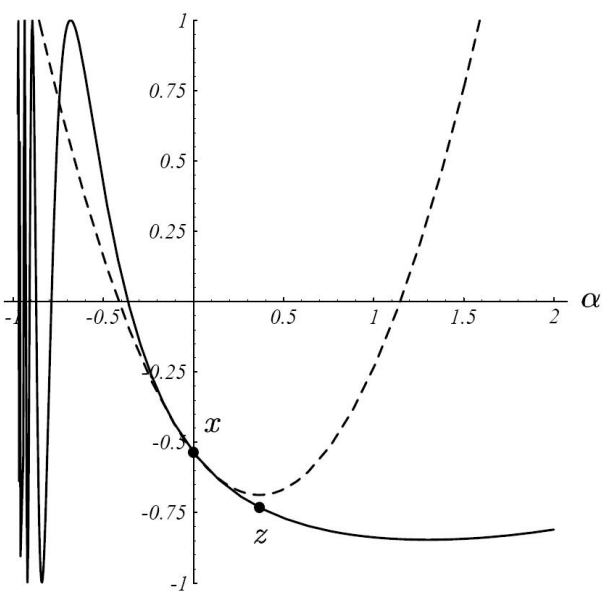
$$\alpha = -\frac{{f'}^T d}{d^T f'' d}.$$

잘려진 테일러(Tayler) 급수는 $f(x+\alpha d)$를 포물선으로 근사한다; 포물선의 최소점까지 내려간다 (그림 39을 보라). 사실 f가 이차 형식이라면, 이 포물선 근사는 정확한 것이 된다. 그 이유는 f''가 바로 친근한 행렬 A이기 때문이다. 일반적으로, 탐색 방향은 f''-직교 (f''-orthogonal)인 경우에 켤레성을 갖는다고 한다. 그러나 "켤레"의 의미는 계속해서 바뀌는데, 이는 f''가 x에 따라 변하기 때문이다. f''가 x에 따라 빠르게 변하면 변할수록 탐색 방향들 사이의 켤레성이 사라진다. 반면 $x_{(i)}$가 해에 가까우면 가까울수록, f''가 반복마다 변하는 정도가 줄어든다. 시작점이 해에 가까울수록 비선형 켤레 경사도 기법의 수렴성이 선형 켤레 경사도 기법과 비슷해진다.

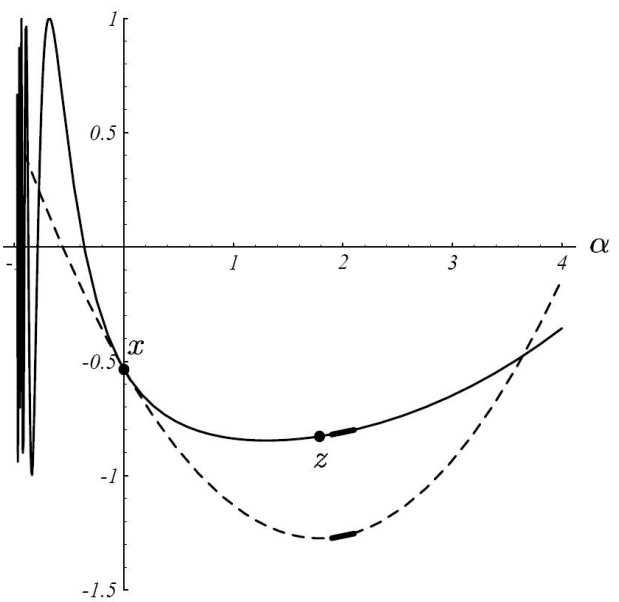
이차가 아닌 함수에 정확한 직선 탐색을 수행하기 위해서는 ${f'}^T$가 0이 될 때까지 직선을 따라 탐색을 반복해야만 한다; 따라서 하나의 켤레 경사도 기법 반복 내에 다수의 뉴턴-랩슨 반복이 포함되어 있을 수 있다.



[그림 38] 비선형 켤레 경사도 기법은 주기적 재시작을 사용할 때 더 효과적이다.



[그림 39] 일차원 함수(실선)를 최소화하는 뉴턴-랩슨 기법. 점 x에서 시작하여 1차, 2차 도함수를 계산하고, 이를 이용하여 함수에 대한 2차 근사를 한다(점선). 포물선의 최소점에서 새로운 점 z를 선택한다. 수렴할 때까지 이러한 단계를 반복한다.



[그림 40] 1차원 함수(실선)을 최소화하는 할선법. 점 x에서 시작하여 서로 다른 두 지점에서 1차 도함수를 계산하며 (여기서는 $\alpha = 0$와 $\alpha = 2$), 이를 이용하여 함수에 대한 2차 근사(점선)을 수행한다. $\alpha = 0$인 위치와 $\alpha = 2$인 위치에서 두 곡선이 모두 동일한 기울기를 갖는다는 것에 유의하라. 뉴턴-랩슨 기법과 같이, 새로운 점 z는 포물선의 최소점에서 선택되며, 수렴될 때까지 반복한다.

$f'^T d$와 $d^T f'' d$의 값들은 각 단계마다 계산되어야 한다. 이러한 계산은 $d^T f'' d$의 값이 해석적으로 간략화될 수 있다면 비용이 크지 않지만, f'' 행렬 전체가 반복적으로 계산되어야만 하므로 알고리즘이 사용할 수 없을 정도로 느리다. 일부 응용에서, f''의 대각 성분만을 사용하여 근사적인 직선 탐색을 수행하여 이러한 문제를 회피할 수 있다. 물론 f''를 전혀 계산할 수 없는 함수들도 있다.

f''를 계산하지 않고 정확한 직선 탐색을 수행하기 위해 할선법(secant method)는 $\alpha = 0$와 $\alpha = \sigma$인 두 개의 서로 다른 위치에서 $f(x + \alpha d)$의 1차 도함수를 계산하고 이를 이용하여 2차 도함수를 근사한다. 이때 σ는 0이 아닌 임의의 작은 수이다:

$$\begin{aligned} \frac{d^2}{d\alpha^2} f(x + \alpha d) &\approx \frac{[\frac{d}{d\alpha} f(x + \alpha d)]_{\alpha=\sigma} - [\frac{d}{d\alpha} f(x + \alpha d)]_{\alpha=0}}{\sigma} \quad \sigma \neq 0 \\ &= \frac{[f'(x + \sigma d)]^T - [f'(x)]^T d}{\sigma}, \end{aligned} \tag{58}$$

이 수는 α와 σ가 0에 가까울수록 더 좋은 근사가 된다. 식 58을 테일러 급수(식 **??**)의 3 번째 항으로 바꾸면 다음을 얻는다:

$$\frac{d}{d\alpha} f(x + \alpha d) \approx [f'(x)]^T d + \frac{\alpha}{\sigma} \left\{ [f'(x + \sigma d)]^T d - [f'(x)]^T d \right\}.$$

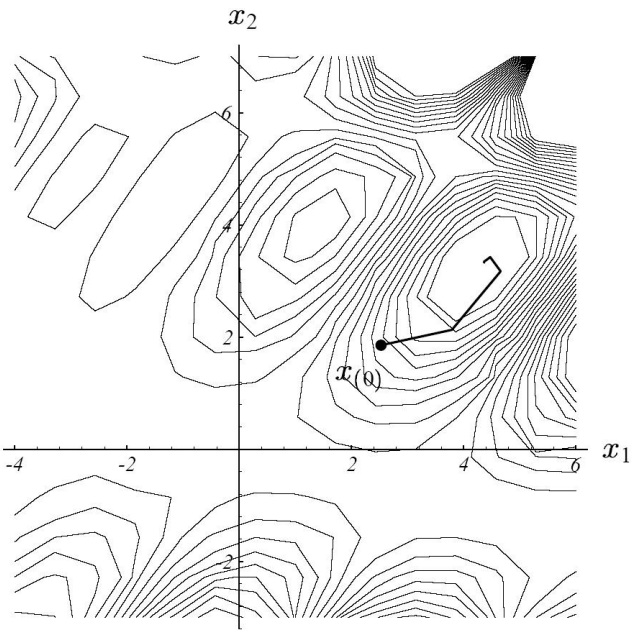
도함수를 0으로 만들어 $f(x + \alpha d)$를 최소화하는 α를 다음과 같이 구할 수 있다:

$$\alpha = -\sigma \frac{[f'(x)]^T d}{[f'(x + \sigma d)]^T d - [f'(x)]^T d} \tag{59}$$

뉴턴-랩슨 기법과 같이, 할선법(secant method) 역시 $f(x + \alpha d)$를 포물선 형태로 근사하지만, 한 점에서의 1차, 2차 도함수를 계산하는 것이 아니라 서로 다른 두 점에서 1차 도함수를 계산하여 (그림 40) 포물선을 선택한다. 할선법의 반복 작업의 첫 번 째 단계에서는 임의의 σ를 선택한다; 이어지는 반복 단계에서는 $x + \sigma d$가 이전 단계의 x가 되도록 선택한다. 다시 말해, $\alpha_{[i]}$가 할선번 i 번째 반복 단계에서 계산된 α 값이라면 $\sigma_{[i+1]} = -\alpha_{[i]}$이다.

뉴턴-랩슨과 할선법 모두 x가 해에 충분히 가까우면 종료되어야 한다. 지나치게 낮은 정밀도를 요구하면 수렴에 실패할 수도 있지만, 지나치게 높은 정밀도를 요구하면 불필요하게 계산이 느려지고 특별한 이익을 얻지 못한다. 이는 $f''(x)$가 x에 대해 많이 변할 경우 켤레성이 빠르게 훼손되기 때문이다. 따라서, 빠르지만 부정확한 직선 탐색이 종종 더 나은 정책이다 (예를 들어, 뉴턴-랩슨이나 할선법에서 제한된 수의 반복만을 수행하는 것이다). 불행하게도 부정확한 직선 탐색은 하강하는 방향이 아닌 탐색 방향을 생성하게 될 수도 있다. 일반적인 해법은 이러한 사태를 미리 검사하는 것이며 ($r^T d$가 양수가 아닌가?), 필요한 경우 $d = r$로 설정하여 켤레 경사도 기법을 새로 시작하는 것이다.

두 기법의 더 큰 문제는 이들 기법이 최소점과 최대점을 구분하지 못한다는 것이다. 비선형 켤레 경사도 기법의 결과는 시작점에 강하게 의존하며, 뉴턴-랩슨 혹은 할선법을 사용한 켤레 경사도 기법이 지역 최소점 근처에서 시작되었다면 해당 위치로 수렴할 가능성이 크다.



[그림 41] 폴락-리비에르(Polak-Ribière) 식을 이용한 비선형 켤레 경사도 기법을 대각 전처리 행렬로 전처리한 결과. 공간이 "펼쳐져" 최소점 주변 등고선이 더 원형(圓形)에 가까워졌음을 보이고 있다.

각각의 방법은 나름의 장점을 가진다. 뉴턴-랩슨 기법은 더 나은 수렴 속도를 가지며, $d^T f'' d$가 빠르게 (즉, $O(n)$ 시간 안에) 계산될 수 있는 경우에 (혹은 근사될 수 있는 경우에) 선호된다. 할선법은 f의 1 차 미분만을 요구하지만, 이 방법의 성공은 파라미터 σ를 얼마나 잘 선택하느냐에 좌우된다. 이들 방법 외에 다른 다양한 기법도 쉽게 유도할 수 있다. 예를 들어 f를 세 개의 서로 다른 지점에서 샘플링(sampling)함으로써 f의 1차 도함수조차 계산할 필요 없이 $f(x + \alpha d)$를 근사하는 포물선을 생성하는 것도 가능하다.

14.3 전처리(Preconditioning)

비선형 켤레 경사도 기법은 f''를 근사하면서 $M^{-1}r$이 쉽게 계산될 수 있는 전처리 행렬(preconditioner matrix) M을 선택하여 전처리를 할 수 있다. 선형 켤레 경사도 기법에서 전처리 행렬은 이차 형식을 변환하여 구와 유사하게 만드려고 한다; 비선형 켤레 경사도 기법의 전처리 행렬은 이러한 변환을 $x_{(i)}$ 근방의 영역에 대해 수행한다.

전체 헤시안 행렬 f''를 계산하는 것이 지나치게 높은 비용을 들때에도 이 행렬의 대각 성분을 계산하여 이를 전처리 행렬로 이용하는 것이 종종 유용하다. 그러나 x가 지역 최소점에서 충분히 떨어져 있을 때에 헤시안 행렬의 대각 성분 중에 양(陽)이 아닌 것이 있을 수도 있다는 것에 주의하라. 전처리 행렬은 양의 정부호(positive-definite)이어야 하므로 양이 아닌 대각 성분은 허용될 수가 없다. 보수적인 해결책은 헤시안이 양

의 정부호임을 보장할 수 없을 때에는 전처리를 하지 않는 것이다 ($M = I$로 설정). 그림 41은 대각성분에 대해 전처리를 수행한 폴락-리비에르(Polak-Ribière) 비선형 켤레 경사도 기법의 수렴을 보이는 것으로 그림 37과 동일한 함수이다. 여기서 매 단계의 전처리를 위해 해 x에서 f''가 가지는 대각성분을 사용하는 수법을 사용하였다[2].

부록

A. 관련 연구

켤레 방향 기법은 1908년 슈미트(Schmidt)에 의해[14] 최초로 발표되었을 것으로 보이며, 1948년에 폭스(Fox), 허스키(Huskey), 그리고 윌킨슨(Wilkinson)에 의해[7] 독립적으로 발견되었다. 50년대 초기에 켤레 경사도 기법이 헤스틴스(Hestenes)와[10] 스티펠(Stiefel)에[15] 의해 독립적으로 발견되었다; 그리고 곧이어, 두 사람은 공동으로 켤레 경사도 기법의 핵심적인 참고문헌으로 여겨지는 내용을 출판하였다[11]. 쳬비쳬프(Chebyshev) 다항식으로의 켤레 경사도 수렴 한계는 카니엘(Kaniel)에 의해 개발되었다[12]. 켤레 경사도 기법은 1971년 라이드(Reid)에 의해 대규모 희소 행렬에 적용되는 반복적 기법으로 대중화되었다[13].

켤레 경사도 기법은 데이비든(Davidon)의 연구[4], 그리고 플레쳐(Fletcher)와 파월(Powell)의 연구[5]를 기반으로 1964년 플레쳐(Fletcher)와 리브스(Reeves)에 의해 비선형 문제로 일반화되었다[6]. 부정확한 직선 탐색을 사용한 비선형 켤레 경사도 기법의 수렴은 다니엘(Daniel)에 의해 분석되었다[3]. 비선형 켤레 경사도 기법에서 β를 선택하는 문제는 길버트(Gilbert)와 노시덜(Nocedal)에 의해 논의되었다[8].

골럽(Golub)과 올리어리(O'Leary)는 70년대 중기의 켤레 경사도 기법과 관련된 역사와 방대한 문헌에 대한 주석을 제공하였다. 이후 대부분의 연구는 비대칭 시스템에 초점을 맞춘다. 선형 시스템의 풀이를 위한 반복적 기법에 대한 조사연구는 바레트(Barrett) 등에 의해 이루어졌다[1].

B. 알고리듬 통조림

이 절에 있는 코드(code)는 이 논고에서 논의된 여러 알고리듬의 효율적인 구현을 보이고 있다.

B1. 최대하강(Steepest Descent)

주어진 입력 A, b에 대해, 시작점은 x이며, 최대 반복 회수를 i_{max}, 오차 허용 범위를 $\varepsilon < 1$이다.

[2] 역자註: 목표가 해 x를 찾는 것이므로, 이는 해를 찾기 위해 해를 이용한 부정행위에 해당한다. 여기서는 전처리를 통한 수렴성을 보이기 위해 이미 알고 있는 해를 이용하여 좋은 전처리 행렬을 구한 것이다.

$i \Longleftarrow 0$
$r \Longleftarrow b - Ax$
$\delta \Longleftarrow r^T r$
$\delta_0 \Longleftarrow \delta$
While $i < i_{max}$ and $\delta > \varepsilon^2 \delta_0$ do
    $q \Longleftarrow Ar$
    $\alpha \Longleftarrow \frac{\delta}{r^T q}$
    $x \Longleftarrow x + \alpha r$
    If i is divisible by 50
        $r \Longleftarrow b - Ax$
    else
        $r \Longleftarrow r - \alpha q$
    $\delta \Longleftarrow r^T r$
    $i \Longleftarrow i + 1$

이 알고리듬은 반복이 i_{max} 회에 도달하거나 $||r_{(i)}|| \leq \varepsilon ||r_{(0)}||$가 되면 중단한다.

나머지는 빠른 점화식으로 계산되지만 50번 반복될 때마다 한 번씩은 정확한 값으로 재계산되어 누적된 부동소수점 오류를 제거한다. 물론 여기 사용된 50이라는 수는 임의의 수이다; 큰 n에 대해 $\sqrt{n}$이 적당하다. 오차 허용치가 크다면, 나머지(residual)를 수정할 필요가 전혀 없다 (실제로 수정은 거의 사용되지 않는다). 오차 허용치가 기계의 부동 소수점 정밀도에 한계에 근접할 경우, δ가 계산된 뒤에 $\delta < \varepsilon^2 \delta_0$인지 확인하는 검사가 추가되어야 하며, 이 검사 결과가 참이라면 정확한 나머지(residual)가 다시 계산되어야 하며 δ도 다시 평가되어야 한다. 이것은 부동 소수점 반올림 오차에 의해 알고리듬이 일찍 종료되는 것을 막는다.

B2. 켤레 경사도(Conjugate Gradient)

주어진 입력 A, b에 대해, 시작점은 x이며, 최대 반복 회수를 i_{max}, 오차 허용 범위를 $\varepsilon < 1$이다.

$i \Longleftarrow 0$
$r \Longleftarrow b - Ax$
$d \Longleftarrow r$
$\delta_{new} \Longleftarrow r^T r$
$\delta_0 \Longleftarrow \delta_{new}$
While $i < i_{max}$ and $\delta_{new} > \varepsilon^2 \delta_0$ do
    $q \Longleftarrow Ad$
    $\alpha \Longleftarrow \frac{\delta_{new}}{d^T q}$
    $x \Longleftarrow x + \alpha d$
    If i is divisible by 50
        $r \Longleftarrow b - Ax$
    else
        $r \Longleftarrow r - \alpha q$
    $\delta_{old} \Longleftarrow \delta_{new}$
    $\beta \Longleftarrow \frac{\delta_{new}}{\delta_{old}}$
    $d \Longleftarrow r + \beta d$
    $i \Longleftarrow i + 1$

이전 절 B1의 마지막에 있는 내용을 참조하라.

B3. 전처리된 켤레 경사도(Preconditioned Conjugate Gradients)

주어진 입력 A, b에 대해, 시작점은 x이며, 전처리 행렬은 M (암시적으로 정의될 수도 있음), 최대 반복 회수를 i_{max}, 오차 허용 범위를 $\varepsilon < 1$이다.

$i \Longleftarrow 0$
$r \Longleftarrow b - Ax$
$d \Longleftarrow M^{-1}r$
$\delta_{new} \Longleftarrow r^T d$
$\delta_0 \Longleftarrow \delta_{new}$
While $i < i_{max}$ and $\delta_{new} > \varepsilon^2 \delta_0$ do
    $q \Longleftarrow Ad$
    $\alpha \Longleftarrow \frac{\delta_{new}}{d^T q}$
    $x \Longleftarrow x + \alpha d$
    If i is divisible by 50
        $r \Longleftarrow b - Ax$
    else
        $r \Longleftarrow r - \alpha q$
    $s \Longleftarrow M^{-1}r$
    $\delta_{old} \Longleftarrow \delta_{new}$
    $\delta_{new} \Longleftarrow r^T s$
    $\beta \Longleftarrow \frac{\delta_{new}}{\delta_{old}}$
    $d \Longleftarrow s + \beta d$
    $i \Longleftarrow i + 1$

"$s \Longleftarrow M^{-1}r$"은 전처리자를 적용해야 함을 암시하고 있으며, 이때 전처리자는 행렬의 형태를 가지지 않을 수도 있다.

역시 이전 절 B1의 마지막에 있는 내용을 참조하라.

B4. 뉴턴-랩슨(Newton-Raphson) 기법을 이용한 전처리 플레쳐-리브스(Fletcher-Reeves) 비선형 켤레 경사도

주어진 함수 f에 대해, 시작점이 x이며, 켤레 경사도 기법의 최대 반복이 i_{max}, 켤레 경사도 기법의 오차 허용치가 $\varepsilon < 1$이고, 뉴턴-랩슨 기법의 최대 반복 회수가 j_{max}, 뉴턴-랩슨법 오차 허용치가 $\epsilon < 1$이다.

$i \Longleftarrow 0$
$k \Longleftarrow 0$
$r \Longleftarrow -f'(x)$
$d \Longleftarrow r$
$\delta_{new} \Longleftarrow r^T r$
$\delta_0 \Longleftarrow \delta_{new}$
While $i < i_{max}$ and $\delta_{new} > \varepsilon^2 \delta_0$ do
    $j \Longleftarrow 0$
    $\delta_d \Longleftarrow d^T d$
    Do
        $\alpha \Longleftarrow -\frac{[f'(x)]^T d}{d^T f''(x) d}$
        $x \Longleftarrow x + \alpha d$
        $j \Longleftarrow j + 1$
    While $j < j_{max}$ and $\alpha^2 \delta_d > \epsilon^2$
    $r \Longleftarrow -f'(x)$
    $\delta_{old} \Longleftarrow \delta_{new}$
    $\delta_{new} \Longleftarrow r^T r$
    $\beta \Longleftarrow \frac{\delta_{new}}{\delta_{old}}$
    $d \Longleftarrow r + \beta d$
    $k \Longleftarrow k + 1$
    If $k = n$ or $r^T d \leq 0$
        $d \Longleftarrow r$
        $k \Longleftarrow 0$
    $i \Longleftarrow i + 1$

이 알고리듬은 최대 반복 회수 i_{max}를 초과하거나, $||r_{(i)}|| \leq \varepsilon ||r_{(0)}||$인 경우에 종료한다.

각각의 뉴턴-랩슨 반복마다 αd를 x에 더한다; 이 반복은 αd가 주어진 허용치 이하로 떨어질 때 ($||\alpha d|| \leq \epsilon$) 혹은 반복 회수가 j_{max}를 초과할 때 종료한다. 빠르지만 부정확한 직선 탐색은 j_{max}를 작은 값으로 사용하거나 헤시안(Hessian) 행렬 $f''(x)$를 대각성분만으로 근사하여 할 수 있다.

비선형 켤레 경사도 기법은 탐색 방향이 하강하는 방향이 아닌 것으로 계산될 때마다 새로 시작된다 ($d \Longleftarrow r$로 설정). 작은 수 n에 대해 수렴 속도를 높이기 위해, 매 n 번의 반복마다 역시 한 번씩 재시작된다.

α 값의 계산은 '0으로 나누기(divide-by-zero)' 오류를 일으킬 수 있다. 이것은 시작점 $x_{(0)}$이 원하는 최소값에 충분히 가깝지 않거나 f가 2차 미분이 가능하지 않을 경우에 발생할 수 있다. 전자의 경우 더 나은 시작점을 선택하거나 더 복잡한 직선 탐색을 사용하여 해결할 수 있다. 후자의 경우에는 켤레 경사도 기법이 최소화에 적합한 알고리듬이 아닐 수 있다.

역시 이전 절 B1의 마지막에 있는 내용을 참조하라.

B5. 할선법을 이용한 전처리 폴락-리비에르(Polak-Ribière) 비선형 켤레 경사도

주어진 함수 f에 대해, 시작점이 x이며, 켤레 경사도 기법의 최대 반복이 i_{max}, 켤레 경사도 기법의 오차 허용치가 $\varepsilon < 1$이고, 할선법 간격 파라미터가 σ_0, 할선법의 최대 반복 회수가 j_{max}, 할선법 오차 허용치가 $\epsilon < 1$이다.

$i \Longleftarrow 0$
$k \Longleftarrow 0$
$r \Longleftarrow -f'(x)$
Calculate a preconditioner $M \approx f''(x)$
$s \Longleftarrow M^{-1}r$
$d \Longleftarrow s$
$\delta_{new} \Longleftarrow r^T d$
$\delta_0 \Longleftarrow \delta_{new}$
While $i < i_{max}$ and $\delta_{new} > \varepsilon^2 \delta_0$ do
    $j \Longleftarrow 0$
    $\delta_d \Longleftarrow d^T d$
    $\alpha \Longleftarrow -\sigma_0$
    $\nu_{prev} \Longleftarrow [f'(x + \sigma_0 d)]^T d$
    Do
        $\nu \Longleftarrow [f'(x)]^T d$
        $\alpha \Longleftarrow \alpha \frac{\nu}{\nu_{prev} - \nu}$
        $x \Longleftarrow x + \alpha d$
        $\nu_{prev} \Longleftarrow \nu$
        $j \Longleftarrow j + 1$
    While $j < j_{max}$ and $\alpha^2 \delta_d > \epsilon^2$
    $r \Longleftarrow -f'(x)$
    $\delta_{old} \Longleftarrow \delta_{new}$
    $\delta_{mid} \Longleftarrow r^T s$
    Calculate a preconditioner $M \approx f''(x)$
    $s \Longleftarrow M^{-1}r$
    $\delta_{new} \Longleftarrow r^T s$
    $\beta \Longleftarrow \frac{\delta_{new} - \delta_{mid}}{\delta_{old}}$
    $k \Longleftarrow k + 1$
    If $k = n$ or $\beta \leq 0$
        $d \Longleftarrow s$
        $k \Longleftarrow 0$
    else
        $d \Longleftarrow s + \beta d$
    $i \Longleftarrow i + 1$

이 알고리듬은 최대 반복 회수 i_{max}를 초과하거나, $||r_{(i)}|| \leq \varepsilon ||r_{(0)}||$인 경우에 종료한다.

각각의 할선법 반복마다 αd를 x에 더한다; 이 반복은 αd가 주어진 허용치 이하로 떨어질 때 ($||\alpha d|| \leq \epsilon$) 혹은 반복 회수가 j_{max}를 초과할 때 종료한다. 빠르지만 부정확한 직선 탐색은 j_{max}를 작은 값으로 사용하여 수행할 수 있다. 파라미터 σ_0은 식 59의 σ 값을 결정하며 할선법을 이용한 최소화 각각의 첫 단계에 사용된다. 불행하게도, 이 파라미터는 수렴을 위해 조정되어야 할 수도 있다.

폴락-리비에르(Polak-Ribière) β 파라미터는 $\frac{\delta_{new}-\delta_{mid}}{\delta_{old}}$로 이는 $\frac{r_{(i+1)}^T s_{(i+1)} - r_{(i+1)}^T s_{(i)}}{r_{(i)}^T s_{(i)}}$와 같고, 이 값은 또한 $\frac{r_{(i+1)}^T M^{-1}(r_{(i_1)} - r_{(i)})}{r_{(i)}^T M^{-1} r_{(i)}}$이다. 전처리 행렬 M이 항상 양의 정부호(positive-definite)이도록 주의를 기울여야 한다. 전처리자는 꼭 행렬의 형태를 가지는 것은 아니다.

이 비선형 켤레 경사도 기법은 폴락-리비에르(Polak-Ribière) β 파라미터가 음수일 때마다 새로 시작된다 ($d \Longleftarrow r$로 설정). 작은 수 n에 대해 수렴 속도를 높이기 위해, 매 n 번의 반복마다 역시 한 번씩 재시작된다.

비선형 켤레 경사도 기법은 몇 가지 선택을 제시한다: 전처리를 할 것이지와 그렇지 않은지, 뉴턴-랩슨(Newton-Raphson) 기법, 할선법(Secant method) 혹은 기타의 방법, 플레쳐-리브스(Fletcher-Reeves) 혹은 폴락-리비에르(Polak-Ribière) 등의 선택이 필요. 위에서 제시된 선택을 통해 다양한 변형이 가능하다. (폴락-리비에르(Polak-Ribière)를 선택하는 것은 언제나 바람직하다.)

C. 추한 증명

C1. $Ax = b$의 해가 이차 형식을 최소화시킨다

A가 대칭이라고 가정하자. x는 $Ax = b$를 만족하는 점이며, 이차 형식 (식 3)를 최소화하는 점이라고 하고, e는 오류항이라고 하자. 그러면 다음이 만족한다.

$$\begin{aligned} f(x+e) &= \frac{1}{2}(x+e)^T A(x+e) - b^T(x+e) + c \quad \text{(by Equation 3)} \\ &= \frac{1}{2}x^T Ax + e^T Ax + \frac{1}{2}e^T Ae - b^T x - b^T e + c \quad \text{(by symmetry of A)} \\ &= \frac{1}{2}x^T Ax - b^T x + c + e^T b + \frac{1}{2}e^T Ae - b^T e \\ &= f(x) + \frac{1}{2}e^T Ae. \end{aligned}$$

A가 양의 정부호(positive-definite)라고 하면, 마지막 항은 0이 아닌 모든 e에 대하여 항상 양(positive)이다; 따라서 x는 f를 최소화한다.

C2. 대칭 행렬은 n 개의 직교 고유벡터를 가진다

어떤 행렬이든 최소한 하나의 고유벡터(eigenvector)를 가진다. 이를 확인하기 위해 $det(A - \lambda I)$가 λ의 다항식이라는 것에 주목하면, 최소한 하나(중근) 이상의 해를 가진다; 이를 λ_A라 하자. 행렬 $A - \lambda_A I$는 행렬식이 0이며, 따라서 특이행렬(singular)이므로 $A - \lambda_A I)v = 0$인 0이 아닌 벡터 v가 반드시 존재해야 한다. $Av = \lambda_A v$이므로, 이 벡터 v가 바로 고유벡터이다.

모든 대칭행렬은 n 개의 직교 고유벡터를 가진다. 이를 증명하기 위해, 4×4 행렬의 경우에 대해 이것이 성립함을 보일 것이며, 임의의 크기를 가진 행렬로의 명백한 일반화는 독자들에게 맡길 것이다. 행렬 A가 최소한 하나의 고유값 λ_A와 고유벡터 v를 가진다는 것은 이미 증명되었다. $x_1 = v/||v||$라고 하면 이는 단위 길이를 가진다. 서로 직교하며 단위 길이를 가진 임의의 세 벡터 x_2, x_3, x_4를 선택하자 (이러한 벡터들은 그람-슈미트 직교화를 통해 찾을 수 있다). $X = [x_1, x_2, x_3, x_4]$라고 하자. x_i가 정규직교(orthonormal)이므로 $X^T X = I$이며, $X^T = X^{-1}$이다. 또한 $i \neq 1$에 대해, $x_i^T A x_i = x_i^T A x_1 = x_i^T \lambda_A x_1 = 0$이므로 다음과 같다.

$$\begin{aligned} X^T AX &= \begin{bmatrix} x_1^T \\ x_2^T \\ x_3^T \\ x_4^T \end{bmatrix} A[x_1 x_2 x_3 x_4] = \begin{matrix} x_1^T \\ x_2^T \\ x_3^T \\ x_4^T \end{matrix} [\lambda_A x_1 A x_2 A x_3 A x_4] \\ &= \begin{bmatrix} \lambda_A & 0 & 0 & 0 \\ 0 & & & \\ 0 & & B & \\ 0 & & & \end{bmatrix} \end{aligned}$$

이때, B는 3×3 대칭 행렬이다. B는 공값 λ_B를 갖는 고유벡터 w를 가져야만 한다. 이때 $\hat{w}$를 첫 번째 원소가 0이고 나머지 원소는 w와 동일한 4 개 원소 벡터라고 하자. 그러면 다음이 명백하다.

$$X^{-1}AX\hat{w} = X^TAX\hat{w} = \begin{bmatrix} \lambda_A & 0 & 0 & 0 \\ 0 & & & \\ 0 & & B & \\ 0 & & & \end{bmatrix} \hat{w} = \lambda_B \hat{w}$$

다시 말해, $AX\hat{w} = \lambda_B X\hat{w}$이며, 따라서 $X\hat{w}$가 A의 고유벡터가 된다. 또한, $x_1^T X\hat{w} = [1000]\hat{w} = 0$이므로, x_1과 $X\hat{w}$가 직교이다. 따라서 최소한 두 개의 직교 고유벡터가 존재한다!

대칭 행렬이 n 개의 직교 고유벡터를 가진다는 내용의 더욱 일반화된 진술은 귀납법으로 증명된다. 위의 예에서, 임의의 3×3 행렬 (B와 같은)이 3 개의 직교 고유벡터를 가진다고 가정하자; 이들을 각각 $\hat{w}_1$, $\hat{w}_2$, $\hat{w}_3$이라고 하자. 그러면 $X\hat{w}_1$, $X\hat{w}_2$, $X\hat{w}_3$는 A의 고유벡터들이며, X의 열들이 정규직교이므로 직교 벡터들을 직교 벡터로 매핑(mapping)하므로 세 고유벡터들 역시 직교이다. 따라서 A는 4 개의 직교 고유벡터를 가진다.

C3. 체비쉐프(Chebyshev) 다항식의 최적성

체비쉐프(Chebyshev) 다항식은 식50과 같은 표현의 최소화에 최적이다. 이는 이 다항식이 범위 $[-1, 1]$의 내부에서 크기가 1을 넘지 않도록 제한된 다항식들 중에서 범위 $[-1, 1]$ 밖에서 가장 빠르게 크기가 증가하는 다항식이기 때문이다.

차수 i의 체비쉐프 다항식은 다음과 같고,

$$T_i(\omega) = \frac{1}{2}[(\omega + \sqrt{\omega^2 + 1})^i + (\omega - \sqrt{\omega^2 - 1})^i],$$

이는 범위 $[-1, 1]$에 대해 다음과 같이 표현될 수 있다.

$$T_i(\omega) = \cos(i \cos^{-1} \omega), \qquad -1 \le \omega \le 1.$$

이 표현에서 (그리고 그림 32에서), 체비쉐프 다항식이 다음과 같은 특성을 갖는다는 것을 알 수 있다.

$$|T_i(\omega) \le 1, \qquad -1 \le \omega \le 1$$

그리고 -1과 1 사이에서 빠르게 진동한다:

$$T_i\left(\cos\left(\frac{\kappa\pi}{i}\right)\right) = (-1)^k, \qquad k = 0, 1, \cdots, i.$$

T_i의 i 개의 해는 범위 $[-1, 1]$ 내에 있는 T_i의 $i+1$ 개의 극값(extrema) 사이에 놓인다. 예를 들어 $T_5(\omega)$의 5 개 해는 그림 32와 같다.

유사한 경우로 다음 함수를 보자.

$$P_i(\lambda) = \frac{T_i(\frac{\lambda_{max}+\lambda_{min}-2\lambda}{\lambda_{max}-\lambda_{min}})}{T_i(\frac{\lambda_{max}+\lambda_{min}}{\lambda_{max}-\lambda_{min}})}$$

위의 함수는 정의역 $[\lambda_{min}, \lambda_{max}]$에 대해 $\pm T_i(\frac{\lambda_{max}+\lambda_{min}}{\lambda_{max}-\lambda_{min}})^{-1}$의 범위 내에서 진동한다. $P_i(\lambda)$는 $P_i(0) = 1$이라는 요구조건을 만족한다.

증명은 간단한 기교를 사용한다. 차수 i인 다항식으로서 $Q_i(0) = 1$이면서 범위 $[\lambda_{min}, \lambda_{max}]$에 대해 P_i보다 나은 다항식 $Q_i(\lambda)$는 존재하지 않는다. 이를 증명하기 위해, 이러한 다항식이 존재한다고 가정하자; 그러면 qjadnl $[\lambda_{min}, \lambda_{max}]$ 내에서 $Q_i(\lambda) < T_i(\frac{\lambda_{max}+lambda_{min}}{\lambda_{max}-\lambda_{min}})^{-1}$이다. 이것은 곧 $P_i - Q_i$가 $\lambda = 0$에서 해를 갖는다는 것이다. 따라서 $P_i - Q_i$는 i 차 다항식이면서 최소한 $i+1$ 개의 해를 가지게 되는데, 이는 불가능한 일이다. 이러한 모순에 의해 i차 체비쉐프 다항식이 식 50을 최적화함을 증명할 수 있다.

D. 연습 과제

다음 질문에 대해, (다른 말이 없는 한) 여러분이 정확한 산술을 사용하여 부동소수점 반올림 오차가 없다고 가정한다.

1. (초급) 전처리된 켤레 경사도 기법은 전처리 행렬에 스칼라 값을 곱해도 영향을 받지 않음을 증명하라. 다시 말해, 임의은 0이 아닌 상수 γ에 대해, 전처리 행렬 γM을 사용하여 얻은 $x_{(0)}, x_{(1)}, \cdots$가 전처리 행렬 M을 사용하여 얻은 단계들과 동일함을 보이라.

2. (고급, 그러나 흥미로운 문제) 대칭이며 양의 정부호인 $n \times n$ 행렬 A가 포함된 $Ax = b$의 해를 구하려고 한다고 가정하다. 불행하게도, 선형 대수 수업의 정신적 충격으로 켤레 경사도 기법에 대한 기억이 나지 않는다. 이때 당신의 고민을 본 착한 고유벡터 요정이 나타나서는 당신에게 행렬 A가 가지는 d 개의 서로 다른 고유값(eigenvalue)을 알려 주었다 (고유벡터는 알려주지 않는다). 그러나 각각의 고유값이 얼마나 중복되어 나타나는지는 알지 못한다.

   당신은 똑똑한 사람이라서, 오늘 아침 잠 속에서 다음 알고리즘을 중얼거리게 되었다.

   Choose an arbitrary starting point $x_{(0)}$
   For $i \Longleftarrow 0$ to $d-1$
   $r_{(i)} \Longleftarrow b - Ax_{(i)}$
   Remove an arbitrary eigenvalue from the list and call it λ_i
   $x_{(i+1)} \Longleftarrow x_{(i)} + \lambda_i^{-1} r_{(i)}$

   고유값이 두번 사용되는 경우는 없으며; 종료시에 고유값 목록(list)은 비게 된다.

   (a) 이 알고리즘이 종료할 때, $x_{(d)}$가 $Ax = b$의 해가 됨을 보이라. ¡힌트¿ **??** 절을 주의 깊게 읽어보라. 2차원 예의 수렴을 그려보라; 고유벡터 축을 그린 뒤에 각각의 고유값을 먼저 선택하려고 해보라.

(알고리즘이 무엇을 하는지를 직관적으로 설명하는 것이 가장 중요하다; 그게 가능하면 그 뒤에 증명에 필요한 수식을 제시하라.)

(b) 이 알고리즘이 d 회의 반복을 통해 정확한 해를 찾기는 하지만, 여러분은 매 번 얻게 되는 $x_{(i)}$가 가능한한 해에 가까운 값을 갖기를 원할 것이다. 매 반복 때마다 고유값 목록에서 어떤 고유값을 선택하는 것이 좋은지를 결정하는 대략의 법칙을 제시해 보라. (다시말해, 어떤 순수로 고유값을 사용해야 하는가?)

(c) 부동소수점 반올림 오류가 일어날 경우 이 알고리즘은 어떻게 엉망이 될 수 있는가?

(d) 부동소수점 반올림 오류가 확대되는 것을 막기 위해 매 반복 시기마다 목록에서 어떤 고유값을 선택해야 하는지 대략의 방법을 제시하라. ¡힌트¿ 이 문제의 답은 문제 (b)의 답과 동일하다.

참고문헌

[1] Richard Barrett, Michael Berry, Tony Chan, James Demmel, June Donato, Jack Dongarra, Victor Eijkout, Roldan Pozo, Charles Romine, and Henk van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, Philadelphia, Pennsylvania, 1993.

[2] William L. Briggs. *A Multigrid Tutorial*. SIAM, Philadelphia, Pennsylvania, 1987.

[3] James W. Daniel. Convergence of the conjugate gradient method with computationally convenient modifications. *Numerische Mathematik*, 10:125–131, 1967.

[4] W. C. Davidon. Variable metric method for minimization. *Technical Report ANL-5990 (Argonne National Laboratory, Argonne, Illinois)*, 1959.

[5] R. Fletcher and M. J. D. Powell. A rapidly convergent descent method for minimization. *Computer Journal*, 6:163–168, 1963.

[6] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *Computer Journal*, 7:149–154, 1964.

[7] L. Fox, H. D. Huskey, and J. H. Wilkinson. Notes on the solution of algebraic inear simultaneous equations. *Quaterly Journal of Mechanics and Applied Mathematics*, 1:149–173, 1948.

[8] Jean Charles Gilbert and Jorge Nocedal. Global convergence properties of conjugate gradient methods for optimization. *SIAM Journal on Optimization*, 2(1):21–42, 1992.

[9] Gene H. Golub and Dianne P. O'Leary. Some history of the conjugate gradient and lanczos algorithms: 1948-1976. *SIAM Review*, 31(1):50–102, 1989.

[10] Magnus R. Hestenes. Iterative methods for solving linear equations. *Journal of Optimization Theory and Applications*, 11(4):323–334, Originally published in 1951 as NAML Report No. 52–9, 1973.

[11] Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, 1952.

[12] Shmuel Kaniel. Estimates for some computational techniques in linear algebra. *Mathematics of Computation*, 20:369–378, 1966.

[13] John K. Reid. On the methods of conjugate gradients for the solution of large sparse systems of linear equations. *Large Sparse Sets of Linear Equations*, pages 231–254, (John K. Reid, ed.), Academic Press, London and New York, 1971.

[14] E. Schmidt. Title unknown. *Rendiconti del Circolo Matematico di Palermo*, 25:53–77, 1908.

[15] Eduard Stiefel. Über einige methoden der relaxationsrechnung. *Zeitschrift für Angewandte Mathematik und Physik*, 3(1):1–33, 1952.

[16] A. van der Sluis and H. A. van der Vorst. The rate of convergence of conjugate gradients. *Numerische Mathematik*, 48(5):543–560, 1986.