

## 감쇠 시스템의 고유진동수와 모드의 민감도를 계산하기 위한 대수적 방법의 개선

### Improved Algebraic Method for Computing Eigenpair Sensitivities of Damped Systems

조홍기\*                      박선규\*\*                      이인원\*\*\*  
Jo. Hong-Ki                  Park. Sun-Kyu                  Lee. In-Won

#### ABSTRACT

This paper presents a very simple procedure for determining the sensitivities of the eigenpairs of damped vibratory system with distinct eigenvalues. The eigenpairs derivatives can be obtained by solving algebraic equation with a symmetric coefficient matrix whose order is  $(n+1) \times (n+1)$ , where  $n$  is the number of degree of freedom. the method is an improvement of recent work by I. W. Lee, D. O. Kim and G. H. Jung; the key idea is that the eigenvalue derivatives and the eigenvector derivatives are obtained at once via only one algebraic equation, instead of using two equations separately as like in Lee and Jung's method. Of course, the method preserves the advantages of Lee and Jung's method.

#### 1. 서론

시스템의 거동은 고유치 문제의 해, 즉 고유진동수와 모드를 구함으로서 완전히 결정된다. 이러한 고유치 문제의 해는 시스템의 설계변수가 변함에 따라 변하게 되고, 따라서 시스템의 동응답 특성도 변하게 된다. 그래서 고유쌍의 미분값들은 동응답의 민감도를 결정할때 유용할 뿐만 아니라, 최적화 설계경향 연구나 실제 시스템 거동에 대한 직관을 얻는데 있어서도 대단히 유용하다.

1975년에 Rudisill과 Chu<sup>(1),(2)</sup>가 고유치와 모드의 미분을 쉽게 구할 수 있는 대수적인 방법을 제안했다. 하지만 이 방법은 계수행렬이 비대칭이어서 계산 시간이 많이 걸릴 뿐 아니라 서로 다른 고유치의 경우에 국한되어 있는 방법이다. 1976년에 Nelson<sup>(3)</sup>은 특이해와 비특이해의 합으로 모드의 미분을 구할 수 있는 알고리즘을 제안했다. 이 방법은 매우 효율적이긴 하지만 알고리즘이 복잡하고 역시 중복근을 갖는 시스템의 경우 적용이 어렵다. Nelson의 방법을 중복근의 경우까지 확장한 Ojalvo<sup>(4),(5)</sup>방법이 있고, 이를 개선한 Mills-Curren<sup>(6)</sup>방법과 Dailey<sup>(7)</sup>방법이 있는데, 이들은 Nelson방법에 기초한 것이어서 알고리즘이 역시 복잡하다. 1997년에 Lee와 Jung<sup>(8),(9)</sup>은 획기적인 알고리즘을 제안했다. 부가조건을 이용해 고유치와 고유모드의 미분을 구하는 방법으로 대단히 간단한 알고리즘을 가지며, 계수행렬이 대칭을 띠고 있어 계산시간 및 연산회수에 있어 매우 효율적이다. 중복근을 가지는 경우에도 쉽게 적용이 되며 수치적인 안정성까지 증명된 방법이다. 이를 1999년에 Lee와 Kim<sup>(10),(11)</sup>이 감쇠 시스템까지 확장시켰다.

\* 한국과학기술원 토목공학과 석사과정

\*\* 성균관대학교 토목공학과 교수

\*\*\* 한국과학기술원 토목공학과 교수

제안방법은 감쇠 시스템이 서로 다른 고유치를 가지는 경우 있어 Lee & Jung's method<sup>(10),(11)</sup>의 개선된 방법이다. 감쇠 시스템의 경우 Lee & Jung's method<sup>(10),(11)</sup>는 고유치의 미분과 고유모드의 미분을 두개의 식을 이용해 따로 구했으나, 본 방법에서는 이를 동시에 구할 수 있는 알고리즘을 이용함으로써 계산 시간 및 연산 회수를 줄일 수 있다. 물론 Lee & Jung's method가 가지는 장점은 유지한다.

2장에서는 기존 방법들의 고유치 문제에 대한 민감도 해석을 간략히 살펴보고, 3장에서는 제안방법 및 제안방법의 수치적 안정성을 보인다. 4장에서는 예제를 통해 제안방법의 효율성을 보인다.

## 2. 기존의 민감도 해석기법

### 2.1 이론적 배경

감쇠가 없는 구조물의 동적해석에서 일반화된 고유문제는 다음과 같다.

$$K\phi_j = \lambda_j M\phi_j \quad (1)$$

여기서  $K$ 와  $M$ 은 각각 시스템의 강성행렬과 질량행렬로서  $M$ 은 양의 한정(positive definite)  $K$ 는 양의 한정 또는 반양의 한정(semi-positive definite)행렬이다.  $K$ 와  $M$ 은  $n$ 차의 대칭행렬이고,  $\lambda_j$ 는 시스템의  $j$ 번째 고유치이고,  $\phi_j$ 는 시스템의  $j$ 번째 고유모드이다. 모드는 일반적으로 질량에 대해 다음과 같이 정규화한다.

$$\phi_j^T M \phi_j = 1 \quad (2)$$

우선 고유치의 미분을 구하기 위해 식(1)을 미분하면,

$$(K - \lambda_j M) \phi_j' = - (K' - \lambda_j M') \phi_j + \lambda_j' M \phi_j \quad (3)$$

식(3)의 양변에  $\phi_j^T$ 를 곱하고 정리하면 고유치의 미분을 다음과 같이 얻을 수 있다.

$$\lambda_j' = \phi_j^T (K' - \lambda_j M') \phi_j \quad (4)$$

여기서,  $\lambda_j' = \partial \lambda_j / \partial p$ ,  $K' = \partial K / \partial p$  그리고  $M' = \partial M / \partial p$ 이고,  $p$ 는 설계 변수이다.  $\lambda_j'$ 를 식(4)와 같이 구했으므로 식(3)의 우변항은 아는 값이다. 그러나 행렬  $(K - \lambda_j M)$ 가 정칙행렬이 아니기 때문에(singular matrix) 식(3)에서 직접적으로  $\phi_j'$ 를 구할 수 없다. 이 문제를 해결하기 위해 많은 학자들이 연구해 왔다.

Nelson<sup>(3)</sup>은 다음과 같이 특이해(particular solution)과 비특이해(homeogeneous solution)의 합으로  $\phi_j'$ 를 구하는 알고리즘을 제시하였다.

$$\phi_j' = v + c \phi_j \quad (5)$$

여기서  $c$ 는 계수이다.  $j$ 번째 모드의 요소중에서 가장 큰 값에 해당되는 위치를  $k$ 라고 하면,  $\phi_j$ 의 요소중에서  $k$ 번째 값을 0으로 함으로서 특이해  $v$ 를 계산할 수 있다. 계수  $c$ 는 식(2)를 미분한 식에서 식(5)를 대입하면 다음과 같이 구할 수 있다.

$$c = -v M \phi_j - 0.5 \phi_j^T M' \phi_j \quad (6)$$

이 방법은 가장 효율적인 방법중에 하나이지만 알고리즘이 다소 복잡할 뿐만 아니라 중복근을 갖는 경우에  $\phi_j'$ 를 구할 수 없다. Ojalvo, Mills-Curren 그리고 Dailey등이 이 방법을 확장하여 중복근을 갖는 경우의 해법을 개발하였지만, 알고리즘이 대단히 복잡하고 계산시간이 오래 걸리는 단점이 있다.

Rudisill과 Chu는 부가조건을 첨가해서 비대칭 정칙행렬을 구성하여  $\phi_j$ 를 계산했다. 고유치문제 식(1)을 미분한 식(3)과 길이를 정규화한 식( $\phi_j^T \phi_j = 1$ )을 미분해( $\phi_j^T \dot{\phi}_j = 1$ ) 부가조건으로 사용함으로써  $\dot{\phi}_j$ 를 다음과 같이 구했다.

$$\begin{bmatrix} \phi_j^T & 0 \\ K - \lambda_j M & -M\phi_j \end{bmatrix} \begin{Bmatrix} \dot{\phi}_j \\ \lambda_j \end{Bmatrix} = \begin{Bmatrix} 0 \\ -(K' - \lambda_j M')\phi_j \end{Bmatrix} \quad (7)$$

이 방법은 알고리즘이 대단히 간단하고 정확한 해를 구할수 있는 방법이지만 계수행렬이 비대칭이라는 큰 단점을 지니고 있어 계산 시간이 많이 소요된다. 그리고 수치적 안정성에 대한 증명이 없고, 중복근의 경우에는 사용할 수 없는 단점이 있다.

최근들어 Lee와 Jung<sup>(8),(9)</sup>은 계수행렬이 대칭이 되도록 부가조건을 첨가한 획기적인 알고리즘을 이용해  $\dot{\phi}_j$ 를 구하였고, 이는 Lee와 Kim<sup>(10),(11)</sup>에 의해 감쇠 시스템으로 확장 적용되었다. 이는 다음에서 살펴본다.

이외에도 모드의 미분을 모드들의 조합으로 표시하여 계산하는 modal method<sup>(12)~(14)</sup>와 이를 수정·보완한 많은 방법<sup>(15),(16)</sup>들이 있지만, 한개의 모드의 미분을 구하기 위하여 전부, 또는 여러개의 모드들이 필요하기 때문에 Nelson 방법에 비해서 계산량이 많은 단점이 있다. 또한 반복과정을 통한 반복적 방법<sup>(17),(18)</sup>들이 있지만 근사해를 구하기 때문에 고려대상에서 제외한다.

## 2.2 Lee & Jung's method<sup>(8),(9)</sup>

이는 부가조건으로서 식(2)를 미분한 것을 사용한다. 이 제약조건을 이용하여 계수행렬을 대칭으로 만들고, 수치적으로 안정한, 즉 역행렬을 구할 수 있는(nonsingular) 선형 대수방정식을 풀므로써 모드의 미분을 구하는 방법이다. 식(2)를 미분하여 정리하면

$$\phi_j^T M \dot{\phi}_j + 0.5 \phi_j^T M' \phi_j = 0 \quad (8)$$

식(3)을 다시 쓰면 다음과 같다.

$$(K - \lambda_j M) \dot{\phi}_j - \lambda_j' M \phi_j = -(K' - \lambda_j M') \phi_j \quad (9)$$

식(8)과 식(9)를 합해서 선형대수방정식 형태로 정리하면 다음과 같다.

$$\begin{bmatrix} K - \lambda_j M & -M\phi_j \\ -\phi_j^T M & 0 \end{bmatrix} \begin{Bmatrix} \dot{\phi}_j \\ \lambda_j \end{Bmatrix} = \begin{Bmatrix} -(K' - \lambda_j M') \phi_j \\ 0.5 \phi_j^T M \phi_j \end{Bmatrix} \quad (10)$$

여기서 계수행렬의 차수는  $(n+1) \times (n+1)$ 이다. 이 식은 알고리즘이 대단히 간단하며, 계수행렬이 대칭인 특징을 가지고 있다. 계수행렬을 band형태나 sky-line형태로 저장할 수 있기 때문에 컴퓨터 메모리나 계산시간이 절약된다. 그리고 계수행렬의 nonsingularity가 증명되었고, 중복근의 경우에도 쉽게 적용될 수 있다. 중복근의 경우 식(10)은 다음과 같이 변형된다.

$$\begin{bmatrix} K - \lambda_j M & -MZ_j \\ -Z_j^T M & 0 \end{bmatrix} \begin{Bmatrix} Z_j \\ \Lambda_j \end{Bmatrix} = \begin{Bmatrix} -(K' - \lambda_j M') Z_j \\ 0.5 Z_j^T M Z_j \end{Bmatrix} \quad (11)$$

여기서  $Z$ 는 시스템이 중복근을 가지는 경우, 모드의 미분이 존재하도록 가정하는 근접한 모드이다. 고유치가  $m$ 번 중복되었다고 하면 계수행렬은  $(n+m) \times (n+m)$  차수를 가진다.

## 2.3 감쇠를 고려한 Lee & Jung's method<sup>(10),(11)</sup> (By Lee & Kim)

이는 상태공간방정식(state-space-equation)을 이용해 감쇠를 처리하여 민감도를 구하는 방법이다. 감쇠를

고려한 경우 고유문제는 다음과 같다.

$$(\lambda_j^2 \mathbf{M} + \lambda_j \mathbf{C} + \mathbf{K}) \phi_j = 0 \quad (12)$$

감쇠 시스템의 고유치와 고유모드를 구하기 위해선 식(12)를 이용해 다음의 상태공간방정식을 구성한다.

$$\begin{bmatrix} -\mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} \begin{Bmatrix} \phi_j \\ \lambda_j \phi_j \end{Bmatrix} = \lambda_j \begin{bmatrix} \mathbf{C} & \mathbf{M} \\ \mathbf{M} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \phi_j \\ \lambda_j \phi_j \end{Bmatrix} \quad (13)$$

이 경우의 고유모드는 다음과 같이 정규화한다.

$$\begin{Bmatrix} \phi_j \\ \lambda_j \phi_j \end{Bmatrix}^T \begin{bmatrix} \mathbf{C} & \mathbf{M} \\ \mathbf{M} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \phi_j \\ \lambda_j \phi_j \end{Bmatrix} = \phi_j^T (2\lambda_j \mathbf{M} + \mathbf{C}) \phi_j = 1 \quad (14)$$

우선 고유치의 미분을 구하기 위해 식(12)를 미분하면,

$$(\lambda_j^2 \mathbf{M} + \lambda_j \mathbf{C} + \mathbf{K}) \frac{\partial \phi_j}{\partial p} = -(2\lambda_j \mathbf{M} + \mathbf{C}) \phi_j \frac{\partial \lambda_j}{\partial p} - \left( \lambda_j^2 \frac{\partial \mathbf{M}}{\partial p} + \lambda_j \frac{\partial \mathbf{C}}{\partial p} + \frac{\partial \mathbf{K}}{\partial p} \right) \phi_j \quad (15)$$

양변에  $\phi_j^T$ 를 곱해 정리하면 다음과 같이 고유치의 미분이 구해진다.

$$\frac{\partial \lambda_j}{\partial p} = -\phi_j^T \left( \lambda_j^2 \frac{\partial \mathbf{M}}{\partial p} + \lambda_j \frac{\partial \mathbf{C}}{\partial p} + \frac{\partial \mathbf{K}}{\partial p} \right) \phi_j \quad (16)$$

고유 모드의 미분은 정규화 조건의 미분을 이용해 구한다. 식(14)을 미분하면 다음과 같다.

$$\phi_j^T (2\lambda_j \mathbf{M} + \mathbf{C}) \frac{\partial \phi_j}{\partial p} + \frac{1}{2} \phi_j^T \left[ 2 \left( \frac{\partial \lambda_j}{\partial p} \mathbf{M} + \lambda_j \frac{\partial \mathbf{M}}{\partial p} \right) + \frac{\partial \mathbf{C}}{\partial p} \right] \phi_j = 0 \quad (17)$$

식(15)와 식(17)을 합해서 선형대수방정식 형태로 정리하면 다음과 같다.

$$\begin{bmatrix} \lambda_j^2 \mathbf{M} + \lambda_j \mathbf{C} + \mathbf{K} & (2\lambda_j \mathbf{M} + \mathbf{C}) \phi_j \\ \phi_j^T (2\lambda_j \mathbf{M} + \mathbf{C}) & 0 \end{bmatrix} \begin{Bmatrix} \frac{\partial \phi_j}{\partial p} \\ 0 \end{Bmatrix} \quad (18)$$

$$= \begin{pmatrix} -(2\lambda_j \mathbf{M} + \mathbf{C}) \phi_j \frac{\partial \lambda_j}{\partial p} - \left( \lambda_j^2 \frac{\partial \mathbf{M}}{\partial p} + \lambda_j \frac{\partial \mathbf{C}}{\partial p} + \frac{\partial \mathbf{K}}{\partial p} \right) \phi_j \\ -\frac{1}{2} \phi_j^T \left[ 2 \left( \frac{\partial \lambda_j}{\partial p} \mathbf{M} + \lambda_j \frac{\partial \mathbf{M}}{\partial p} \right) + \frac{\partial \mathbf{C}}{\partial p} \right] \phi_j \end{pmatrix}$$

이렇게 식(16)과 식(18)을 통해 고유치와 고유모드의 미분을 구한다.

이 방법은 Lee & Jung's method<sup>(8),(9)</sup>의 장점은 모두 유지 하면서 감쇠 시스템으로 확장한 경우이다. 물론 중복근의 경우에도 Lee & Jung's method<sup>(8),(9)</sup>와 같이 근접 모드를 이용해 적용할 수 있다.

### 3. 제안방법

#### 3.1 제안방법

제안방법은 시스템이 서로 다른 고유치를 가지는 경우, 감쇠를 고려한 Lee & Jung's method<sup>(10),(11)</sup>의 개선된 방법이다. 전 방법에서는 고유치의 미분을 고유모드의 미분과 분리해서 따로 계산하였으나 제안방법은 이를 동시에 계산함으로써 연산횟수 및 계산 시간을 줄일 수 있는 방법이다. 물론 Lee & Jung's method<sup>(10),(11)</sup>가 가지는 장점은 그대로 유지한다. 감쇠 시스템의 고유문제 식(12)과 정규화 조건 식(14)를 다시 써보면 다음과 같다.

$$(\lambda_j^2 \mathbf{M} + \lambda_j \mathbf{C} + \mathbf{K}) \phi_j = 0 \quad (19)$$

$$\phi_j^T(2\lambda_j \mathbf{M} + \mathbf{C}) \phi_j = 1 \quad (20)$$

고유문제 식(19)를 설계변수에 대해 미분한 후 재정리하면,

$$(\lambda_j^2 \mathbf{M} + \lambda_j \mathbf{C} + \mathbf{K}) \frac{\partial \phi_j}{\partial p} + (2\lambda_j \mathbf{M} + \mathbf{C}) \phi_j \frac{\partial \lambda_j}{\partial p} = - \left( \lambda_j^2 \frac{\partial \mathbf{M}}{\partial p} + \lambda_j \frac{\partial \mathbf{C}}{\partial p} + \frac{\partial \mathbf{K}}{\partial p} \right) \phi_j \quad (21)$$

정규화 조건 식(20)을 설계변수에 대해 미분하면 다음과 같다.

$$\phi_j^T(2\lambda_j \mathbf{M} + \mathbf{C}) \frac{\partial \phi_j}{\partial p} + \phi_j^T \mathbf{M} \phi_j \frac{\partial \lambda_j}{\partial p} = - \frac{1}{2} \phi_j^T \left( 2\lambda_j \frac{\partial \mathbf{M}}{\partial p} + \frac{\partial \mathbf{C}}{\partial p} \right) \phi_j \quad (22)$$

이 두식을 합해서 선형대수방정식으로 정리하면 다음과 같다.

$$\begin{bmatrix} \lambda_j^2 \mathbf{M} + \lambda_j \mathbf{C} + \mathbf{K} & (2\lambda_j \mathbf{M} + \mathbf{C}) \phi_j \\ \phi_j^T(2\lambda_j \mathbf{M} + \mathbf{C}) & \phi_j^T \mathbf{M} \phi_j \end{bmatrix} \begin{bmatrix} \frac{\partial \phi_j}{\partial p} \\ \frac{\partial \lambda_j}{\partial p} \end{bmatrix} \quad (23)$$

$$= \begin{bmatrix} - \left( \lambda_j^2 \frac{\partial \mathbf{M}}{\partial p} + \lambda_j \frac{\partial \mathbf{C}}{\partial p} + \frac{\partial \mathbf{K}}{\partial p} \right) \phi_j \\ - \frac{1}{2} \phi_j^T \left( 2\lambda_j \frac{\partial \mathbf{M}}{\partial p} + \frac{\partial \mathbf{C}}{\partial p} \right) \phi_j \end{bmatrix}$$

이 식이 제안방법의 핵심이다. 전 방법<sup>(10),(11)</sup>에서는 고유문제를 미분한 식(15)의 양변에  $\phi_j^T$ 를 곱함으로써  $\frac{\partial \lambda_j}{\partial p}$ 를 따로 구했으나 본 방법에서는 식(15)을 다시 정리해서 그대로 사용해,  $\frac{\partial \lambda_j}{\partial p}$ 와  $\frac{\partial \phi_j}{\partial p}$ 를 동시에 구한다.

### 3.2 제안방법의 수치적 안정성

식(23)의 계수행렬이 정칙행렬(nonsingular matrix)임을 증명하기 위해 다음의 성질을 도입한다.

$$\det(\mathbf{Y}^T \mathbf{A}^* \mathbf{Y}) = \det(\mathbf{Y}^T) \det(\mathbf{A}^*) \det(\mathbf{Y}) \quad (24)$$

즉  $\det(\mathbf{Y}) \neq 0$ 인 행렬  $\mathbf{Y}$ 에 대해  $\det(\mathbf{Y}^T \mathbf{A}^* \mathbf{Y}) \neq 0$ 이라면,  $\det(\mathbf{A}^*) \neq 0$  즉 행렬  $\mathbf{A}^*$ 는 정칙행렬이다. 여기서  $\mathbf{A}^*$ 는 식(23)의 계수행렬이다.

$\mathbf{Y}$ 를 다음과 같이 가정해 정칙행렬로 만들자.

$$\mathbf{Y} = \begin{bmatrix} \boldsymbol{\Psi} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \quad (25)$$

여기서  $\boldsymbol{\Psi} = [\phi_1 \ \phi_2 \ \dots \ \phi_{n-1} \ \phi_n]$ ,  $\phi_j$ 는  $j$ 번째 고유모드이고,  $\phi_j$ 's는  $\phi_j$ 와 서로 독립인 임의의 벡터이다.

$\boldsymbol{\Psi}$ 는  $n \times n$ 행렬이고,  $\mathbf{Y}$ 는  $(n+1) \times (n+1)$ 행렬이다. 식(23)의 계수행렬에  $\mathbf{Y}^T$ 와  $\mathbf{Y}$ 를 곱하면 다음과 같다.

$$\begin{aligned} \mathbf{Y}^T \mathbf{A}^* \mathbf{Y} &= \begin{bmatrix} \boldsymbol{\Psi} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}^T \begin{bmatrix} \lambda_j^2 \mathbf{M} + \lambda_j \mathbf{C} + \mathbf{K} & (2\lambda_j \mathbf{M} + \mathbf{C}) \phi_j \\ \phi_j^T(2\lambda_j \mathbf{M} + \mathbf{C}) & \phi_j^T \mathbf{M} \phi_j \end{bmatrix} \begin{bmatrix} \boldsymbol{\Psi} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{\Psi}^T (\lambda_j^2 \mathbf{M} + \lambda_j \mathbf{C} + \mathbf{K}) \boldsymbol{\Psi} & \boldsymbol{\Psi}^T (2\lambda_j \mathbf{M} + \mathbf{C}) \phi_j \\ \phi_j^T (2\lambda_j \mathbf{M} + \mathbf{C}) \boldsymbol{\Psi} & \phi_j^T \mathbf{M} \phi_j \end{bmatrix} \end{aligned} \quad (26)$$

여기서 행렬  $\Psi^T(\lambda_j^2 \mathbf{M} + \lambda_j \mathbf{C} + \mathbf{K}) \Psi$ 의 마지막 행과 열은 0이다. 왜냐하면  $\Psi$ 의 마지막 열이  $\phi_j$ 이기 때문이다. 따라서 행렬  $\Psi^T(\lambda_j^2 \mathbf{M} + \lambda_j \mathbf{C} + \mathbf{K}) \Psi$ 는 다음과 같이 나타낼 수 있다.

$$\Psi^T(\lambda_j^2 \mathbf{M} + \lambda_j \mathbf{C} + \mathbf{K}) \Psi = \begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} \quad (27)$$

여기서  $\mathbf{B}$ 의 차수는  $(n-1) \times (n-1)$ 이고 랭크(rank)도  $(n-1)$ 이다. 그리고 식(26)의 행벡터와 열벡터 마지막 항은 정규화 조건에 의해 다음과 같이 된다.

$$\Psi^T(2\lambda_j \mathbf{M} + \mathbf{C}) \phi_j = \begin{Bmatrix} \mathbf{b} \\ 1 \end{Bmatrix}, \quad \phi_j^T(2\lambda_j \mathbf{M} + \mathbf{C}) \Psi = \begin{Bmatrix} \mathbf{b} \\ 1 \end{Bmatrix}^T \quad (28)$$

따라서 식(26)은 다음과 같이 정리될 수 있고,

$$\mathbf{Y}^T \mathbf{A}^* \mathbf{Y} = \begin{bmatrix} \mathbf{B} & \mathbf{0} & \mathbf{b} \\ \mathbf{0} & 0 & 1 \\ \mathbf{b}^T & 1 & \phi_j^T \mathbf{M} \phi_j \end{bmatrix} \quad (29)$$

determinant의 성질을 적용하면 식(29)의 determinant는 다음과 같이 구할 수 있다.

$$\det(\mathbf{Y}^T \mathbf{A}^* \mathbf{Y}) = \det \begin{bmatrix} 0 & 1 \\ 1 & \phi_j^T \mathbf{M} \phi_j \end{bmatrix} \det \left( \mathbf{B} - \begin{bmatrix} 0 & \mathbf{b} \\ 1 & \phi_j^T \mathbf{M} \phi_j \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{b}^T \end{bmatrix} \right) \quad (30)$$

여기서  $\begin{bmatrix} 0 & \mathbf{b} \\ 1 & \phi_j^T \mathbf{M} \phi_j \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{b}^T \end{bmatrix} = 0$  이고  $\det \begin{bmatrix} 0 & 1 \\ 1 & \phi_j^T \mathbf{M} \phi_j \end{bmatrix} = -1$  이다.

따라서  $\det(\mathbf{Y}^T \mathbf{A}^* \mathbf{Y}) = -\det(\mathbf{B}) \neq 0$ 이 되고,  $\det(\mathbf{A}^*) \neq 0$ , 즉 행렬  $\mathbf{A}^*$ 는 정칙행렬임이 증명된다.

#### 4. 수치예제

제안방법의 효율성을 증명하기 위하여 서로다른 고유치의 경우의 시스템 예로 36개의 삼각형 요소를 가지고 자유도 72인 평판예제를 고려하자. 우리는 Lee & Jung's method<sup>(8)-(11)</sup>가 다른 방법에 비해 계산시간 및 수치적 안정성, 정확성 등에서 다른 방법에 비해 효율적임을 참고문헌[8~11]에서 보았다. 이에 본 예제를 통해 제안방법을 감쇠시스템의 경우 기존 Lee & Jung's method<sup>(10),(11)</sup>와 비교하고자 한다. 해석에 사용한 컴퓨터는 RAM 40Mega, CPU capacity 120Hz를 가지는 펜티엄 120이다.

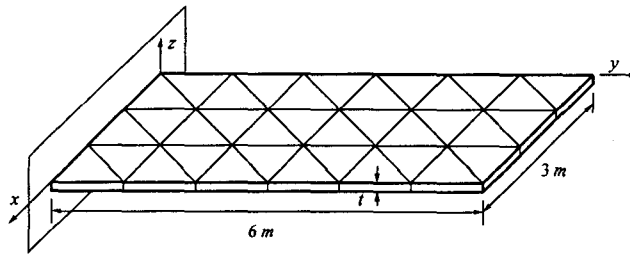


Fig. 1. Cantilever plate with the thickness  $t$  design parameter

평판은 28개의 절점을 가지고 절점당 3자유도( $y$ -변위,  $z$ -변위,  $x$ -각변위)를 가진다. 탄성계수는  $10.5 \times 10^5 \text{ N/m}^2$ 이고, 밀도는  $5.88 \times 10^{-3} \text{ kg/m}^3$ , Poisson비는 0.3이다. 설계변수는 평판의 두께  $t$ 이다.

감쇠행렬은 다음과 같이 Rayleigh damping으로 가정한다.

$$C = \alpha K + \beta M \quad (31)$$

여기서  $\alpha = \beta = 0.01$ 로 가정하겠다. 평판의 강성행렬과 질량행렬은 각각  $t^3$ 과  $t$ 에 비례하므로  $t$ 에 대해 미분함으로써  $\frac{\partial M}{\partial t}$ 와  $\frac{\partial K}{\partial t}$ 를 구할 수 있다.

제안방법과 Lee & Jung's method<sup>(10),(11)</sup>에 의한 결과 일부를 표1과 표2에서 나타내었고, 표3에서 계산시간을 비교하였다.

결과를 보면 일단 둘다 정확한 같은 값을 가진다. 계산 시간을 보면, 20개의 고유쌍 미분을 구하는데 걸린 시간은 Lee & Jung's method의 경우 6.26초 제안방법은 5.32초, 약 15%의 계산시간 절감 효과를 가져온다. 이는 표3에서 보드시피 고유치의 미분을 계산하는데 걸린시간이 절약된 결과이다.

Table 1. The lowest twenty eigenvalue and their derivatives

| Mode number | Eigenvalue              | Eigenvalue derivative (Lee&Jung's method) | Eigenvalue derivative (Proposed method) |
|-------------|-------------------------|---|---|
| 1,2         | -1.537E-02j ± 5.453E+00 | -2.973E+00j ± 5.453E+02                   | -2.973E+00j ± 5.453E+02                 |
| 3,4         | -2.367E-01j ± 2.173E+01 | -4.724E+01j ± 2.173E+03                   | -4.724E+01j ± 2.173E+03                 |
| 5,6         | -5.480E-01j ± 3.309E+01 | -1.095E+02j ± 3.308E+03                   | -1.095E+02j ± 3.308E+03                 |
| 7,8         | -2.345E+00j ± 6.844E+01 | -4.689E+02j ± 6.836E+03                   | -4.689E+02j ± 6.836E+03                 |
| 9,10        | -4.268E+00j ± 9.229E+01 | -8.535E+02j ± 9.209E+03                   | -8.535E+02j ± 9.209E+03                 |
| 11,12       | -7.023E+00j ± 1.183E+02 | -1.405E+03j ± 1.179E+04                   | -1.405E+03j ± 1.179E+04                 |
| 13,14       | -8.693E+00j ± 1.316E+02 | -1.738E+03j ± 1.310E+04                   | -1.738E+03j ± 1.310E+04                 |
| 15,16       | -1.187E+01j ± 1.536E+02 | -2.373E+03j ± 1.527E+04                   | -2.373E+03j ± 1.527E+04                 |
| 17,18       | -1.569E+01j ± 1.764E+02 | -3.138E+03j ± 1.750E+04                   | -3.138E+03j ± 1.750E+04                 |
| 19,20       | -2.196E+01j ± 2.084E+02 | -4.392E+03j ± 2.061E+04                   | -4.392E+03j ± 2.061E+04                 |

Table 2. Some components of the first eigenvector and its derivatives

| Eqn. number | Eigenvector           | Eigenvector derivative (Lee&Jung's method) | Eigenvector derivative (Proposed method) |
|-------------|-----------------------|--|--|
| 1           | -7.380E-01-j7.380E-01 | 7.380E+01+j7.380E+01                       | 7.380E+01+j7.380E+01                     |
| 2           | -1.498E+00-j1.498E+00 | 1.498E+02+j1.498E+02                       | 1.498E+02+j1.498E+02                     |
| 3           | 2.736E-01+j2.736E-01  | -2.736E+01-j2.736E+01                      | -2.736E+01-j2.736E+01                    |
| 4           | -8.554E-01-j8.554E-01 | 8.554E+01+j8.554E+01                       | 8.554E+01+j8.554E+01                     |
| 5           | -1.581E+00-j1.581E+00 | 1.581E+02+j1.581E+02                       | 1.581E+02+j1.581E+02                     |
| :           |                       |  |  |
| 69          | -7.399E-02-j7.399E-02 | 7.399E+00+j7.399E+00                       | 7.399E+00+j7.399E+00                     |
| 70          | -1.867E+01-j1.867E+01 | 1.867E+03+j1.867E+03                       | 1.867E+03+j1.867E+03                     |
| 71          | -4.472E+00-j4.472E+00 | 4.472E+02+j4.472E+02                       | 4.472E+02+j4.472E+02                     |
| 72          | -1.223E-01-j1.223E-01 | 1.223E+01+j1.223E+01                       | 1.223E+01+j1.223E+01                     |

## 5. 결론

본 논문에서는 부가조건을 이용해 고유치와 고유모드의 미분을 대수적으로 구하는 Lee & Jung's method를 개선한 식을 제안하였다. 제안방법에서는 Lee & Jung's method와는 달리 고유치와 고유모드의 미분을 동시에 계산함으로써 계산시간을 현저하게 줄일 수 있다. 물론 전 방법이 가지는 정확해, 수치적 안정성, 계수행렬의 대칭성등의 장점은 모두 유지한다.

Table 3. CPU time spent on the calculation of the first twenty eigenpair derivatives

| Method              | Operations   | CPUtime (sec) |
|---------------------|--|---------------|
| Lee & Jung's method | $\frac{\partial \lambda_j}{\partial p} = -\phi_j^T \left( \lambda_j^2 \frac{\partial \mathbf{M}}{\partial p} + \lambda_j \frac{\partial \mathbf{C}}{\partial p} + \frac{\partial \mathbf{K}}{\partial p} \right) \phi_j$   | 0.95          |
|                     | $\mathbf{A} = \begin{bmatrix} \lambda_j^2 \mathbf{M} + \lambda_j \mathbf{C} + \mathbf{K} & (2\lambda_j \mathbf{M} + \mathbf{C}) \phi_j \\ \phi_j^T (2\lambda_j \mathbf{M} + \mathbf{C}) & 0 \end{bmatrix}$   | 1.71          |
|                     | $f_j = \begin{cases} -(2\lambda_j \mathbf{M} + \mathbf{C}) \phi_j \frac{\partial \lambda_j}{\partial p} - \left( \lambda_j^2 \frac{\partial \mathbf{M}}{\partial p} + \lambda_j \frac{\partial \mathbf{C}}{\partial p} + \frac{\partial \mathbf{K}}{\partial p} \right) \phi_j \\ -\frac{1}{2} \phi_j^T \left[ 2 \left( \frac{\partial \lambda_j}{\partial p} \mathbf{M} + \lambda_j \frac{\partial \mathbf{M}}{\partial p} \right) + \frac{\partial \mathbf{C}}{\partial p} \right] \phi_j \end{cases}$ | 1.32          |
|                     | $\begin{cases} \frac{\partial \phi_j}{\partial p} \\ 0 \end{cases} = [\mathbf{A}]^{-1} f_j$  | 2.28          |
|                     | Total  | <b>6.26</b>   |
| Proposed method     | $\mathbf{A}^* = \begin{bmatrix} \lambda_j^2 \mathbf{M} + \lambda_j \mathbf{C} + \mathbf{K} & (2\lambda_j \mathbf{M} + \mathbf{C}) \phi_j \\ \phi_j^T (2\lambda_j \mathbf{M} + \mathbf{C}) & \phi_j^T \mathbf{M} \phi_j \end{bmatrix}$  | 1.73          |
|                     | $f_j^* = \begin{cases} -\left( \lambda_j^2 \frac{\partial \mathbf{M}}{\partial p} + \lambda_j \frac{\partial \mathbf{C}}{\partial p} + \frac{\partial \mathbf{K}}{\partial p} \right) \phi_j \\ -\frac{1}{2} \phi_j^T \left( 2\lambda_j \frac{\partial \mathbf{M}}{\partial p} + \frac{\partial \mathbf{C}}{\partial p} \right) \phi_j \end{cases}$  | 1.31          |
|                     | $\begin{cases} \frac{\partial \phi_j}{\partial p} \\ \frac{\partial \lambda_j}{\partial p} \end{cases} = [\mathbf{A}^*]^{-1} f_j^*$  | 2.28          |
|                     | Total  | <b>5.32</b>   |

## 참고문헌

1. C. S. Rudisill 1974 *AIAA Journal* 12, 721-722. Derivatives of Eigenvalues and Eigenvectors for a General Matrix.
2. C. S. Rudisill and Y. Chu 1975 *AIAA Journal* 13, 834-837. Numerical Methods for Evaluating the Derivatives of Eigenvalues and Eigenvectors.
3. R. B. Nelson 1976 *AIAA Journal* 14, 1201-1205. Simplified Calculations of Eigenvector Derivatives.



4. I. U. Ojalvo 1987 *AIAA Journal* **25** 1386-1390. Efficient Computation of Mode-Shape Derivatives for Large Dynamic Systems.
5. I. U. Ojalvo 1988 *AIAA Journal* **26** 361-366. Efficient Computation of Modal Sensitivities for Systems with Repeated Frequencies.
6. W. C. Mills-Curran 1988 *AIAA Journal* **26** 867-881. Calculation of Derivatives for Structures with Repeated Eigenvalues.
7. R. L. Dailey 1989 *AIAA Journal* **27**, 486-491. Eigenvector Derivatives with Repeated Eigenvalues.
8. I. W. Lee and G. H. Jung 1997 *Computers and Structures* **62**(3), 429-435. An Efficient Algebraic Method for Computation of Natural Frequency and Mode Shape Sensitivities: Part I, Distinct Natural Frequencies.
9. I. W. Lee and G. H. Jung 1997 *Computers and Structures* **62**(3), 437-443. An Efficient Algebraic Method for Computation of Natural Frequency and Mode Shape Sensitivities: Part II, Multiple Natural Frequencies.
10. I. W. Lee and D. O. Kim 1999 *Journal of Sound and Vibration* **223**(3), 399-412. Natural Frequency and Mode Shape Sensitivities of Damped System : Part I, Distinct Natural Frequencies.
11. I. W. Lee and D. O. Kim 1999 *Journal of Sound and Vibration* **223**(3), 413-424. Natural Frequency and Mode Shape Sensitivities of Damped System : Part II, Multiple Natural Frequencies.
12. J. N. Juang, P. Ghaemmaghami and K. B. Lim 1989 *Journal of Guidance, Control Dynamics* **12** 480-486. Eigenvalue and Eigenvector Derivatives of a Nondefective Matrix.
13. D. V. Murthy and R. T. Haftka 1988 *International Journal for Numerical Methods in Engineering* **26**, 293-311. Derivatives of Eigenvalues and Eigenvectors for a General Complex Matrix.
14. K. B. Lim and J. L. Junkins 1987 *Journal of Guidance* **10**, 581-587. Re-examination of Eigenvector Derivatives.
15. B. P. Wang 1985 *AIAA Journal* **29** 1018-1020. Improved Approximate Method for Computing Eigenvector Derivatives in Structural Dynamics.
16. Z. S. Liu, S. H. Chen and Y. Q. Zhao 1994 *Computers and Structures* **52**, 1135-1143. An Accurate Method for Computing Eigenvector Derivatives for Free-Free Structures.
17. A. L. Andrew 1978 *Journal of Comput. Phys.* **26**, 107-112. Convergence of an Iterative Method for Derivatives of Eigensystems.
18. R. C. E. Tan 1986 *Journal of Comput. Phys.* **67**, 230-235. Accelerating the Convergence of an Iterative Method for Derivatives of Eigensystems.