Latin American Journal of
Solids and Structures
www.lajss.org

Adaptation of the Lanczos Algorithm for the Solution of Buckling Eigenvalue Problems

Abstract

An adaptation of the conventional Lanczos algorithm is proposed to solve the general symmetric eigenvalue problem $\mathbf{K}\boldsymbol{\phi} = \lambda\mathbf{K}_G\boldsymbol{\phi}$ in the case when the geometric stiffness matrix $\mathbf{K}_G$ is not necessarily positive-definite. The only requirement for the new algorithm to work is that matrix $\mathbf{K}$ must be positive-definite. Firstly, the algorithm is presented for the standard situation where no shifting is assumed. Secondly, the algorithm is extended to include shifting since this procedure may be important for enhanced precision or acceleration of convergence rates. Neither version of the algorithm requires matrix inversion, but more resources in terms of memory allocation are needed by the version with shifting.

Keywords

Lanczos algorithm, general eigenproblem, shear buckling.

Alfredo R. de Faria [a]

[a] Instituto Tecnológico de Aeronáutica,
Department of Mechanical Engineering,
São José dos Campos, SP, Brazil
arfaria@ita.br

http://dx.doi.org/10.1590/1679-78253932

Received 14.04.2017
In revised form 30.05.2017
Accepted 22.09.2017
Available online 30.09.2017

1 INTRODUCTION

One of the most efficient methods for extracting eigenpairs (eigenvalues and eigenvectors) was proposed by Lanczos (Lanczos, 1950). Lanczos algorithm is considered today to be the most efficient (if not the most efficient) numerical method for computing extreme eigenvalues and associated eigenvectors (Paige, 1971; Paige, 1972) for large problems. The finite arithmetic of computers introduce some difficulties in the procedure since it may lead to loss of orthogonality of the eigenvectors being computed. However, the loss of orthogonality can be treated through corrective schemes (Parlett and Scott, 1979) already available.

In structural modal analysis the eigenvalue problem to be solved is usually not in the standard form $\mathbf{A}\boldsymbol{\phi} = \lambda\boldsymbol{\phi}$, where $\mathbf{A}$ is a symmetric matrix. Because of inertial effects, in addition to the stiffness matrix $\mathbf{K}$, there is always a mass matrix $\mathbf{M}$ involved, such that the eigenvalue problem is stated as $\mathbf{K}\boldsymbol{\phi} = \lambda\mathbf{M}\boldsymbol{\phi}$, known as the generalized form. In modal analyses the stiffness matrix is guaranteed to be positive semi-definite, whereas the mass matrix is positive-definite (Ericsson and Ruhe, 1980; Ramaswamy, 1980; Bathe 1996). Nevertheless, the traditional linearized buckling analysis requires computation of a geometric stiffness matrix $\mathbf{K}_G$, which is dependent on the prebuckling stress distribution (Zienkiewicz, 1991). In the linearized buckling analysis matrix $\mathbf{K}_G$ replaces matrix $\mathbf{M}$. However, the positive-definiteness of $\mathbf{K}_G$ is not guaranteed what brings apparently unsurmountable an obstacle to the conventional Lanczos algorithm applied to the generalized eigenproblem. Jones and Patrick (Jones and Patrick, 1989) propose the use of a spectral transformation to solve buckling eigenproblems, but it relies on a shifted operator and does not address the basic difficulty of taking the square root of a possibly negative number, except by taking preventive action of selecting appropriate initial guess vectors that must be preconditioned. However, if $\mathbf{K}_G$ is indefinite, their method crashes.

Given the incapacity of the conventional Lanczos algorithm to handle indefinite matrices $\mathbf{K}_G$ this work proposes an adaptation of the algorithm to compute eigenvalues (critical loads) in linearized buckling problems. The strategy consists basically in relying on the positive-definiteness of $\mathbf{K}$, despite the fact that $\mathbf{K}_G$ is indefinite. Initially, a version of the adapted algorithm is proposed without shifting that is sufficient to obtain the lowest positive eigenvalues usually sought. Subsequently, a version of the algorithm with shifting is proposed in order to either enhance precision or accelerate rates of convergence. It is shown that the version with shifting requires more computer resources in terms of memory allocation.

2 THE ADAPTED LANCZOS ALGORITHM

The generalized linear eigenvalue buckling problem is

$$\mathbf{K}\boldsymbol{\phi} = \lambda\mathbf{K_G}\boldsymbol{\phi} \tag{1}$$

where K is the stiffness matrix, ϕ is the eigenvector, λ is the eigenvalue and K_G is the geometric stiffness matrix. K is symmetric and positive-definite. K_G is symmetric but may not be positive-definite. In classical linearized buckling analysis when there is only compressive prebuckling loads matrix K_G can be assumed to be positive-definite. However, this is not always the case. Structural components subject to shear loads invariably lead to indefinite K_G matrices. Two simple examples of these situation are panels subject to shear and closed box beams subject to torsion, both of which consist of practical cases encountered in aircraft wing design. Nonetheless, even compressive loads may result in indefinite K_G matrices. Almeida and Hansen (2000) investigated rectangular composite plates with circular cutouts and observed local buckling modes related to negative eigenvalues. The absolute values of those negative eigenvalues were high in comparison to the positive eigenvalues. Even though these negative eigenvalues are not of practical relevance, they cripple the traditional Lanczos algorithm that will eventually be required to compute the square root of a negative number as will be shown. Figure 1 presents the traditional Lanczos algorithm proposed by Bathe (Bathe, 1996) to solve the generalized eigenvalue posed in Eq. (1).

```
1. Arbitrarily choose starting vector $\boldsymbol{\phi}$ and normalize it with respect to matrix $\mathbf{K}_G$, i.e., compute $\gamma = (\boldsymbol{\phi}^T\mathbf{K}_G\boldsymbol{\phi})^{1/2}$ and subsequently $\mathbf{x}_1 = \boldsymbol{\phi}/\gamma$.

2. Define $\beta_1 = 0$ and $\boldsymbol{\phi}_0 = 0$.

3. For $i = 1, 2,..., q - 1$ do
        $\mathbf{K}\bar{\boldsymbol{\phi}}_{i+1} = \mathbf{K}_G\boldsymbol{\phi}_i$
        $\alpha_i = \bar{\boldsymbol{\phi}}_{i+1}^T\mathbf{K_G}\boldsymbol{\phi}_i$
        $\tilde{\boldsymbol{\phi}}_{i+1} = \bar{\boldsymbol{\phi}}_{i+1} - \alpha_i\boldsymbol{\phi}_i - \beta_i\boldsymbol{\phi}_{i-1}$
        $\beta_{i+1}=(\tilde{\boldsymbol{\phi}}_{i+1}^T\mathbf{K_G}\tilde{\boldsymbol{\phi}}_{i+1})^{1/2}$
        $\boldsymbol{\phi}_{i+1} = \tilde{\boldsymbol{\phi}}_{i+1}/\beta_{i+1}$
```

Figure 1: Algorithm 1.

The algorithm in Fig. 1 generates a sequence of vectors ϕ_i, at least theoretically, orthonormal to K_G. As q in algorithm 1 increases, the symmetric tridiagonal matrix

$$\mathbf{T} = \begin{bmatrix} \alpha_1 & \beta_2 & & & & \\ \beta_2 & \alpha_2 & \beta_3 & & & \\ & & & \ddots & & \\ & & & \beta_{q-1} & \alpha_{q-1} & \beta_q \\ & & & & \beta_q & \alpha_q \end{bmatrix} \tag{2}$$

will possess eigenvalues that converge to the inverse of the eigenvalues of the original problem. In theory, it is shown that, when q is exactly the dimension of the eigenvalue problem, matrix T given in Eq. (2) will have eigenvalues that are exactly the inverse of those of the original eigenproblem.

Observe that, in step 3 of algorithm 1, matrix products like $\phi^T K_G \phi$ must be computed and, subsequently, their square roots must be taken. In the case of buckling problems K_G is indefinite, therefore, $\phi^T K_G \phi$ may be negative, fatally crippling the algorithm. If one insists in using the conventional algorithm and takes, instead, the square root of the absolute value of $\phi^T K_G \phi$, the algorithm flies off and converges to absolutely meaningless results. If the inverted eigenvalue problem $K_G\phi = \upsilon K\phi$ is solved instead, where $\upsilon = 1/\lambda$, matrix products of the form $\phi^T K\phi$ would be computed and, because K is positive-definite, it would lead necessarily to a positive number. Since the conventional Lanczos algorithm computes approximated values to the lowest eigenvalues, the lowest υ's are obtained and they correspond to the highest λ's (recall that $\upsilon = 1/\lambda$). Hence, solving the inverted eigenproblem is not an option if the lowest λ's are to be obtained.

The fundamental aspect to be noted in proposing a new variant of the Lanczos algorithm is to realize that, since K is positive-definite, its inverse, K^{-1}, is also positive-definite. One can therefore define the transformation

$$\boldsymbol{\phi} = \mathbf{K}^{-1}\mathbf{x} \tag{3}$$

that is always unique because K is positive-definite, therefore invertible. Substitution of Eq. (3) into Eq. (1) and pre-multiplication by (K$_G$)$^{-1}$ yields

$$\mathbf{K}_G^{-1}\mathbf{x} = \lambda\mathbf{K}^{-1}\mathbf{x} \tag{4}$$

There is apparently one inconsistency in Eq. (2): K$_G$ is indefinite and may be singular, such that (K$_G$)$^{-1}$ may not even exist. Let us forget this inconsistency for now and employ the conventional Lanczos algorithm to the eigenproblem posed in Eq. (4).

```
1. Arbitrarily choose starting vector $\mathbf{x}$ and normalize it with respect to matrix $\mathbf{K}^{-1}$, i.e., compute
   $\gamma = (\mathbf{x}^T \mathbf{K}^{-1}\mathbf{x})^{1/2}$ and subsequently $\mathbf{x}_1 = \mathbf{x}/\gamma$.

2. Define $\beta_1 = 0$ and $\mathbf{x}_0 = 0$.

3. For $i = 1, 2,\ldots, q - 1$ do
       $\mathbf{K}_G^{-1}\bar{\mathbf{x}}_{i+1} = \mathbf{K}^{-1}\mathbf{x}_i$
       $\alpha_i = \bar{\mathbf{x}}_{i+1}^T\mathbf{K}^{-1}\mathbf{x}_i$
       $\tilde{\mathbf{x}}_{i+1} = \bar{\mathbf{x}}_{i+1} - \alpha_i\mathbf{x}_i - \beta_i\mathbf{x}_{i-1}$
       $\beta_{i+1} = (\tilde{\mathbf{x}}_{i+1}^T\mathbf{K}^{-1}\tilde{\mathbf{x}}_{i+1})^{1/2}$
       $\mathbf{x}_{i+1} = \tilde{\mathbf{x}}_{i+1}/\beta_{i+1}$
```

Figure 2: Algorithm 2

The difficulty of having to take the square root of a negative number is now resolved since the product $\mathbf{x}^T$K$^{-1}\mathbf{x}$ results in a positive number. However, the algorithm presented in Fig. 2 is apparently highly awkward because both K^{-1} and (K$_G$) $^{-1}$ are required. If one carefully examines the algorithm it will be clear that inversions are unnecessary. The inverse of K$_G$ is required only in the evaluation of $\mathbf{K}_G^{-1}\bar{\mathbf{x}}_{i+1} = \mathbf{K}^{-1}\mathbf{x}_i$ that can also be recast as $\bar{\mathbf{x}}_{i+1} = \mathbf{K}_G\mathbf{K}^{-1}\mathbf{x}_i$. Therefore, (K$_G$) $^{-1}$ is not actually needed. As for K^{-1}, notice that only products of the form K$^{-1}\mathbf{x}$ are required. These can be more effectively evaluated if recast as K$\boldsymbol{\phi}$ = $\mathbf{x}$. Numerical solution of this system is achieved through traditional **LU** decomposition of K (Crout, Cholesky, etc.). Before beginning algorithm 2, matrix K can be **LU** decomposed and, since it is symmetric, only **L** must be stored. Refinement of algorithm 2 results in algorithm 3 presented in Fig. 3.

Vector **v** is only auxiliary. Computation of **v** in step 5 ($\mathbf{v} = \mathbf{K}^{-1}\tilde{\mathbf{x}}_{i+1}$), is obtained through solution of the system $\mathbf{Kv} = \tilde{\mathbf{x}}_{i+1}$, which is not troublesome since matrix K has already been **LU** decomposed. A particularly effective way of selecting the starting vector $\boldsymbol{\phi}_1$ is to make it equal to the main diagonal of matrix **K**. Since the transformation proposed in Eq. (3) was adopted, the eigenvectors of the original and transformed problems are related by $\boldsymbol{\phi}_i$ = K$^{-1}\mathbf{x}_i$.

The eigenproblem with shifting is posed in Eq. (5)

$$(\mathbf{K} - \sigma\mathbf{K}_G)\boldsymbol{\phi} = \mu\mathbf{K}_G\boldsymbol{\phi} \tag{5}$$

where σ is the shift and the eigenvalues μ to be determined satisfy $\mu = \lambda - \sigma$. If the transformation proposed in Eq. (3) is used in Eq. (5), pre-multiplication by (K$_G$)$^{-1}$ leads to

$$\mathbf{K}_G^{-1}(\mathbf{K} - \sigma\mathbf{K}_G)\mathbf{K}^{-1}\mathbf{x} = \mu\mathbf{K}^{-1}\mathbf{x} \tag{6}$$

```
1. Choose $\boldsymbol{\phi}_1$ and compute $\mathbf{x}_1 = \mathbf{K}\boldsymbol{\phi}_1$.

2. Decompose $\mathbf{K} = \mathbf{LU}$

3. Compute $\gamma = (\mathbf{x}_1^T\mathbf{K}^{-1}\mathbf{x}_1)^{1/2} = (\mathbf{x}_1^T\boldsymbol{\phi}_1)^{1/2}$

4. $\mathbf{x}_1 = \mathbf{x}_1/\gamma$, $\mathbf{v} = \boldsymbol{\phi}_1/\gamma$, $\beta_1 = 0$, $\mathbf{x}_0 = 0$.

5. For $i = 1, 2,..., q - 1$ do
       $\bar{\mathbf{x}}_{i+1} = \mathbf{K}_G\mathbf{v}$
       $\alpha_i = \bar{\mathbf{x}}_{i+1}^T\mathbf{v}$
       $\tilde{\mathbf{x}}_{i+1} = \bar{\mathbf{x}}_{i+1} - \alpha_i\mathbf{x}_i - \beta_i\mathbf{x}_{i-1}$
       $\mathbf{v} = \mathbf{K}^{-1}\tilde{\mathbf{x}}_{i+1}$
       $\beta_{i+1=}(\tilde{\mathbf{x}}_{i+1}^T\mathbf{v})^{1/2}$
       $\mathbf{x}_{i+1} = \tilde{\mathbf{x}}_{i+1}/\beta_{i+1}$
       $\mathbf{v} = \mathbf{v}/\beta_{i+1}$
```

Figure 3: Algorithm 3

The particularly complicated matrix product $\mathbf{K}_G^{-1}(\mathbf{K}-\sigma\mathbf{K}_G)\mathbf{K}^{-1}$ is purposely not computed. Both K^{-1} and $(\mathrm{K}_G)^{-1}$ in this product are not actually evaluated in the algorithm proposed in Fig. 4. However, matrix, $(\mathrm{K} - \sigma\mathrm{K}_G)$, is now present and it must be **LU** decomposed. The adapted Lanczos algorithm with shifting is:

```
1. Choose $\boldsymbol{\phi}_1$ and compute $\mathbf{x}_1 = \mathbf{K}\boldsymbol{\phi}_1$.

2. Decompose $(\mathbf{K} - \sigma\mathbf{K}_G) = \mathbf{LU}$

3. Compute $\gamma = (\mathbf{x}_1^T\boldsymbol{\phi}_1)^{1/2}$

4. $\mathbf{x}_1 = \mathbf{x}_1/\gamma$, $\boldsymbol{\phi}_1 = \boldsymbol{\phi}_1/\gamma$, $\beta_1 = 0$, $\mathbf{x}_0 = \boldsymbol{\phi}_0 = 0$.

5. For $i = 1, 2,..., q - 1$ do
       $(\mathbf{K} - \sigma\mathbf{K}_G)\bar{\boldsymbol{\phi}}_{i+1} = \mathbf{K}_G\boldsymbol{\phi}_i$
       $\bar{\mathbf{x}}_{i+1} = \mathbf{K}\bar{\boldsymbol{\phi}}_{i+1}$
       $\alpha_i = \bar{\mathbf{x}}_{i+1}^T\boldsymbol{\phi}_i$
       $\tilde{\boldsymbol{\phi}}_{i+1} = \bar{\boldsymbol{\phi}}_{i+1} - \alpha_i\boldsymbol{\phi}_i - \beta_i\boldsymbol{\phi}_{i-1}$
       $\tilde{\mathbf{x}}_{i+1} = \bar{\mathbf{x}}_{i+1} - \alpha_i\mathbf{x}_i - \beta_i\mathbf{x}_{i-1}$
       $\beta_{i+1=}(\tilde{\mathbf{x}}_{i+1}^T\tilde{\boldsymbol{\phi}}_{i+1})^{1/2}$
       $\boldsymbol{\phi}_{i+1} = \tilde{\boldsymbol{\phi}}_{i+1}/\beta_{i+1}$
       $\mathbf{x}_{i+1} = \tilde{\mathbf{x}}_{i+1}/\beta_{i+1}$
```

Figure 4: Algorithm 4

In algorithm 4 it is suggested that the two sequences of eigenvectors $\mathbf{x}_i$ and $\boldsymbol{\phi}_i$ are stored. This procedure precludes the necessity to solve systems of the form $\mathrm{K}\boldsymbol{\phi} = \mathbf{x}$. However, if computer memory is scarce, then either $\mathbf{x}$ or $\boldsymbol{\phi}$ only may be stored. Storing the sequence of $\boldsymbol{\phi}_i$ makes more sense since these are the eigenvectors of the original eigenvalue problem. Computer memory may also become an issue because, in addition to K and K_G, matrix $\mathrm{K} - \sigma\mathrm{K}_G$ decomposed must also be stored. Another source of hardship may be the decomposition of $\mathrm{K} - \sigma\mathrm{K}_G$. If the shift σ is near one of the eigenvalues of the problem stated in Eq. (1), $\mathrm{K} - \sigma\mathrm{K}_G$ may become singular, therefore not invertible.

3 NUMERICAL EXAMPLES

A simple eigenvalue problem was proposed by Ramaswamy (1980): find all the eigenvalues of Eq. (1) for matrices K and K_G given by

$$\mathbf{K}=\begin{bmatrix}1&0&0&0&0\\0&3&0&0&0\\0&0&5&0&0\\0&0&0&4&0\\0&0&0&0&2\end{bmatrix}\quad\text{and}\quad\mathbf{K}_G=\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&-1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{bmatrix}\tag{7}$$

Since $\mathbf{K}_G$ is indefinite, this simple problem helps illustrating the obstacles encountered by the conventional Lanczos algorithm presented in Fig. 1. The starting vector is chosen as

$$\boldsymbol{\phi}=\{1/\sqrt{5}\quad 1/\sqrt{5}\quad 1/\sqrt{5}\quad 1/\sqrt{5}\quad 1/\sqrt{5}\}^T\tag{8}$$

Blunt application of algorithm 1, requires, in the first iteration, evaluation of $\beta_2{}^2=-0.1181$, halting the algorithm. Stubbornly taking the square root of the absolute value of $(\widetilde{\boldsymbol{\phi}}_{i+1}^T\mathbf{K_G}\widetilde{\boldsymbol{\phi}}_{i+1})^{1/2}$ to compute β_{i+1} does not help; it does not fix the algorithm. In this simple example this procedure would lead to the wrong set of eigenvalues 1.3409, 0.3819, –0.8515, 0.0392 and –0.0809. On the other hand, algorithm 3 produces the correct eigenvalues and eigenvectors placed columnwise:

$$\begin{Bmatrix}1.0000\\2.0000\\-5.0000\\4.0000\\3.0000\end{Bmatrix}\quad\text{and}\quad\begin{bmatrix}1.0000&0.0000&0.0003&0.0006&0.0005\\-0.0003&0.0000&0.0000&0.0000&0.5774\\-0.0002&0.0002&0.4472&0.0000&0.0000\\-0.0003&0.0006&0.0000&0.5000&0.0000\\0.0000&-0.7071&0.0003&0.0008&-0.0001\end{bmatrix}\tag{9}$$

If one makes the geometric stiffness matrix $\mathbf{K}_G$ indefinite by, for instance assuming that

$$\mathbf{K}_G=\begin{bmatrix}1&0&0&0&0\\0&0&0&0&0\\0&0&-1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{bmatrix}\tag{10}$$

then the new variant of the Lanczos algorithm delivers the solution

$$\begin{Bmatrix}1.0000\\-5.0000\\2.0000\\4.7661\times10^{16}\\4.0000\end{Bmatrix}\quad\text{and}\quad\begin{bmatrix}1.0000&0.0000&-0.0023&0.0000&-0.0007\\0.0000&0.0000&0.0000&0.5774&0.0000\\0.0000&-0.4472&0.0000&0.0000&-0.0005\\0.0004&-0.0005&0.0000&0.0000&0.5000\\-0.0016&0.0000&-0.7071&0.0000&0.0000\end{bmatrix}\tag{11}$$

Notice the "infinite" eigenvalue obtained numerically (4.7661×10^{16}) and the correct eigenvector associated to it (column 4).

A more practical example consists of a 40 cm × 40 cm square composite plate whose laminate is just a single layer of T300-5208 graphite/epoxy (properties in Tab. 1) of 0.15 mm thickness. The single layer is oriented according to angle θ as shown in Fig. 5. Simply supported boundary conditions are imposed along the four edges and pure shear loading is applied.

Property	Value
Longitudinal modulus of elasticity E_1	154.0 GPa
Transverse modulus of elasticity E_2	11.13 GPa
In-plane Poisson ratio ν_{12}	0.304
In-plane shear modulus G_{12}	6.98 GPa
Transverse shear modulus $G_{13}=G_{12}$	6.98 GPa
Transverse shear modulus G_{23}	3.36 GPa

Table 1: Mechanical properties of the T300-5208 graphite/epoxy.

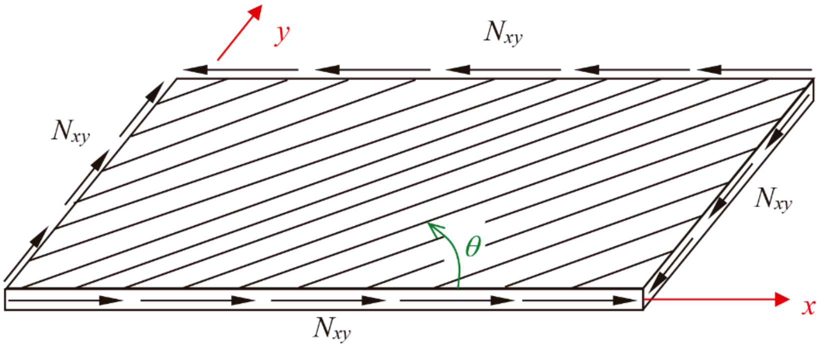

Figure 5: Single ply composite square plate.

The composite plate is modeled with a 4×4 mesh of bicubic elements (Faria, 2000), and the stiffness and geometric stiffness matrices are computed. The Lanczos algorithm 3 for $\theta = 45^o$ leads to the critical loads 20.91 N/m (lowest positive eigenvalue) and –3.17 N/m (highest negative eigenvalue). Along the plate diagonal, where $x = y$, the pure shear load induces compressive normal stresses. Therefore, a high critical buckling load is expected (20.91 N/m). When the layer angle is $\theta = -45^o$ the critical loads become 3.17 N/m (lowest positive eigenvalue) and –20.91 N/m (highest negative eigenvalue).

Figure 6 shows how the lowest positive eigenvalue (λ_{pos}) and highest negative eigenvalue (λ_{neg}) vary with the layer angle θ. λ_{pos} and λ_{neg} have the same absolute values for $\theta = 0^o$ and $\theta = \pm 90^o$. The maximum $|\lambda|$ is obtained for $\theta = \pm 45^o$. If the plate is required to equally support either positive or negative shear loads then the optimal strategy would be to choose $\theta = 0^o$ or $\theta = 90^o$.

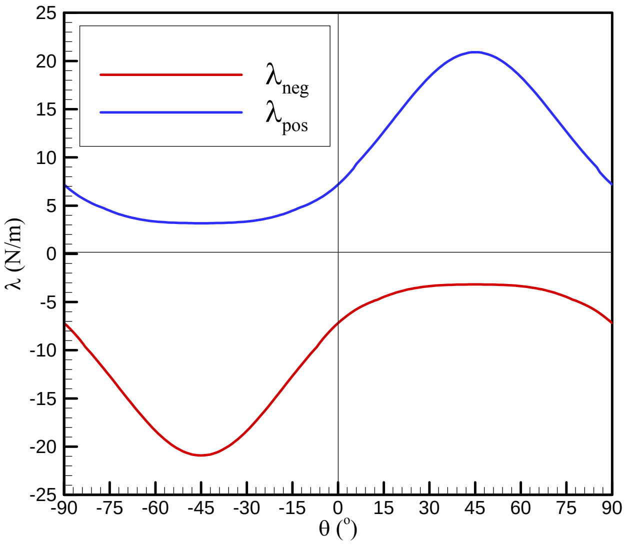

Figure 6: Critical load variation.

4 CONCLUSIONS

A deeper investigation of the Lanczos algorithm applied to the solution of the eigenvalue problem stated in Eq. (3) reveals that the principal contribution to vector $\mathbf{x}_{i+1}$ is from ϕ_i, where ϕ_i is the eigenvector related to the ith least eigenvalue in absolute value. Hence, vector $\mathbf{x}_2$ consists mostly of ϕ_1; vector $\mathbf{x}_3$ consists mostly of ϕ_2; so on so forth, where the eigenvalues are ordered such that $|\lambda_1| < |\lambda_2| < \ldots < |\lambda_n|$. Thus, one expects that the procedure converges quickly to the least eigenvalues in absolute value and related eigenvectors. If a structure is subject to loadings in either direction (positive or negative), the relevant critical buckling load is the one associated to the least eigenvalue in absolute value. Therefore, it is clear that eigensolvers of practical relevance must be able to compute eigenvalues of either sign.

References

Almeida, S.F.M., Hansen, J.S., (2000). Natural frequencies of composite plates with thermal residual stresses. International Journal of Solids and Structures 36: 3517-3539.

Bathe, K-J., (1996). Finite Element Procedures, Prentice Hall (New Jersey).

Ericsson, T., Ruhe, A., (1980). The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric eigenvalue problems. Mathematics of Computation 35: 1251-1268.

Faria, A.R., (2000). Buckling optimization of composite plates and cylindrical shells: uncertain loading combinations, Ph.D. Thesis, University of Toronto.

Jones, M.T., Patrick, M.L., (1989) The use of Lanczo's method to solve the large generalized symmetric definite N eigenvalue problem. NASA Contractor Report 181914, ICASE Report No. 89-69.

Lanczos, C., (1950). An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. Journal of Research of the Natural Bureau of Standards 45: 255-282.

Paige, C.C., (1971). The computation of eigenvalues and eigenvectors of very large sparse matrices, Ph.D. Thesis, University of London.

Paige, C.C., (1972). Computational variants of the Lanczos method for the eigenproblem. Journal of the Institute of Mathematics and its Applications 10: 373-381.

Parlett, B.N., Scott, D.S., (1979). The Lanczos algorithm with selective orthogonalization. Mathematics of Computation 33: 217-238.

Ramaswamy, S., (1980). On the effectiveness of the Lanczos method for the solution of large eigenvalue problems. Journal of Sound and Vibration 73:405-418.

Zienkiewicz, O.C., (1991). The Finite Element Method, Vol. 2: Solid and Fluid Mechanics, Dynamics and Non-linearity. 4th ed., McGraw-Hill: London