SOLUTION OF BUCKLING EIGENPROBLEMS THROUGH A NEW VARIANT OF THE LANCZOS ALGORITHM

Alfredo R. de Faria
Instituto Tecnológico de Aeronáutica, CTA - ITA - IEM, São José dos Campos, SP 12228-900, Brazil
arfaria@ita.br

Abstract. *The Lanczos algorithm applied to the general symmetric eigenvalue problem $\mathbf{K}\phi = \lambda\mathbf{K}_G\phi$ is adapted to handle situations where the geometric stiffness matrix $\mathbf{K}_G$ fails to be positive-definite. It is shown that, provided $\mathbf{K}$ is positive-definite, the Lanczos algorithm with minor modifications can be used. Two versions of the algorithm are presented: with and without shifting. Neither version involves direct matrix inversion although the algorithm with shifting requires more allocation memory.*

Keywords: *Lanczos algorithm, general eigenproblem, shear buckling*

1. Introduction

The Lanczos algorithm (Lanczos, 1950) for obtaining extreme eigenvalues and associated eigenvectors is now recognized to be one of the fastest (if not the fastest) methods available (Paige, 1971; Paige, 1972) for large problems. Although numerical difficulties exist when the finite arithmetic of computers is considered corrective schemes have been proposed to remedy the loss of orthogonality of the generated eigenvectors observed in some situations (Parlett and Scott, 1979).

The eigenproblem encountered in structural modal analyses are not in the standard form ($\mathbf{A}\phi = \lambda\phi$) involving only one symmetric matrix $\mathbf{A}$. It involves, instead, two symmetric matrices: $\mathbf{K}$ and $\mathbf{M}$, where $\mathbf{K}$ is the stiffness matrix and $\mathbf{M}$ is the mass matrix in the form $\mathbf{K}\phi = \lambda\mathbf{M}\phi$. Both the stiffness and the mass matrices are assumed positive-definite (Ericsson and Ruhe, 1980; Ramaswamy, 1980; Bathe 1996). However, in classical buckling analysis, the mass matrix is replaced by the geometric stiffness matrix $\mathbf{K}_G$ that depends on the stress distribution in the prebuckling state (Zienkiewicz, 1991). Unlike $\mathbf{M}$, the positive-definiteness of $\mathbf{K}_G$ cannot be assumed and the traditional Lanczos algorithm applied to the general eigenproblem breaks down.

This work is concerned with numerical evaluation of critical loads (eigenvalues) in buckling problems using an adapted version of the Lanczos algorithm. Although $\mathbf{K}_G$ is indefinite the positive-definiteness of $\mathbf{K}$ is used to advantage. Unlike modal analysis, in the buckling analysis only the lowest positive eigenvalue is usually sought. Nevertheless, a version of the algorithm with shifting is provided for the purpose of obtaining a particular buckling mode or for convergence acceleration.

2. A new variant of the the Lanczos algorithm

Consider the general eigenvalue buckling problem stated as

$$\mathbf{K}\phi = \lambda\mathbf{K}_G\phi, \tag{1}$$

where $\mathbf{K}$ is the stiffness matrix, $\mathbf{K}_G$ is the geometric stiffness matrix, λ is the eigenvalue and ϕ is the eigenvector. $\mathbf{K}$ and $\mathbf{K}_G$ are symmetric matrices and it is assumed that $\mathbf{K}$ is positive-definite. $\mathbf{K}_G$, however, is indefinite. A simple buckling problem that possesses such an indefinite geometric stiffness matrix is, for example, the simply supported rectangular plate under pure shear loading. Another example is the circular cylindrical shell in torsion. Generally, whenever there are shear loadings, indefinite $\mathbf{K}_G$ matrices arise. However, even under normal loadings only, indefinite $\mathbf{K}_G$ matrices can be encountered for relatively simple geometries. For instance, the rectangular plate with a circular cutout has local buckling modes associated with negative eigenvalues (de Almeida and Hansenm, 2000). In this particular example the absolute value of the highest negative eigenvalue is usually much greater than the absolute value of the lowest positive eigenvalue. This, however, does not prevent the traditional Lanczos algorithm from breaking down whenever it needs to find the square root of a negative number as will be shown. The Lanczos algorithm proposed by Bathe (Bathe, 1996) to solve generalized eigenproblems as posed in Eq. (1) is

Algorithm 1

---

1. Select arbitrary starting vector ϕ and normalize it with respect to matrix $\mathbf{K}_G$, i.e., find $\gamma = (\phi^T\mathbf{K}_G\phi)^{1/2}$ and make $\mathbf{x}_1 = \phi/\gamma$.
2. Define $\beta_1 = 0$ and $\phi_0 = \mathbf{0}$.

3. For $i = 1, 2, ..., q-1$ compute

$$\mathbf{K}\overline{\boldsymbol{\phi}}_{i+1} = \mathbf{K}_G\boldsymbol{\phi}_i$$

$$\alpha_i = \overline{\boldsymbol{\phi}}_{i+1}^T\mathbf{K}_G\boldsymbol{\phi}_i$$

$$\tilde{\boldsymbol{\phi}}_{i+1} = \overline{\boldsymbol{\phi}}_{i+1} - \alpha_i\boldsymbol{\phi}_i - \beta_i\boldsymbol{\phi}_{i-1}$$

$$\beta_{i+1} = (\tilde{\boldsymbol{\phi}}_{i+1}^T\mathbf{K}_G\tilde{\boldsymbol{\phi}}_{i+1})^{1/2}$$

$$\boldsymbol{\phi}_{i+1} = \tilde{\boldsymbol{\phi}}_{i+1}/\beta_{i+1}$$

The sequence of vectors $\boldsymbol{\phi}_i$ thereby generated is, theoretically, orthonormal to $\mathbf{K}_G$. The eigenvalues of the symmetric tridiagonal matrix

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

converge to the reciprocals of the eigenvalues of the original problem as q increases. When q is exactly the dimension of the eigenproblem, the eigenvalues of $\mathbf{T}$ match exactly those of the original problem.

Notice that the algorithm above requires computation of matrix products of the form $\boldsymbol{\phi}^T\mathbf{K}_G\boldsymbol{\phi}$ and subsequent evaluation of its square root. However, since $\mathbf{K}_G$ is indefinite, this matrix product may yield a negative number and the algorithm crashes. Blunt evaluation of the square root of the absolute value of $\boldsymbol{\phi}^T\mathbf{K}_G\boldsymbol{\phi}$ does not solve the difficulty because it would ruin the convergence properties of the algorithm. Another option would be solving the modified eigenproblem $\mathbf{K}_G\boldsymbol{\phi} = \upsilon\mathbf{K}\boldsymbol{\phi}$ instead, where $\upsilon = 1/\lambda$, since $\mathbf{K}$ is assumed positive-definite. Because the algorithm provides an approximation to the lowest eigenvalues one would obtain the lowest υ's which correspond to the highest λ's. Thus, this particular transformation of the eigenproblem cannot be used if the lowest λ's are sought.

The key to proposing a new algorithm that can handle the indefiniteness of $\mathbf{K}_G$ consists in noting that, because $\mathbf{K}$ is positive-definite, so is the inverse of $\mathbf{K}$. Hence, let us define the transformation

$$\boldsymbol{\phi} = \mathbf{K}^{-1}\mathbf{x} \tag{2}$$

that is always resolvable since $\mathbf{K}$ is not singular. Substituting Eq. (2) into Eq. (1) and pre-multiplying by $\mathbf{K}_G^{-1}$ leads to

$$\mathbf{K}_G^{-1}\mathbf{x} = \lambda\mathbf{K}^{-1}\mathbf{x}. \tag{3}$$

Notice that $\mathbf{K}_G$ is assumed indefinite and it may be singular, meaning that $\mathbf{K}_G^{-1}$ may not be well-defined. However, for the moment, let us apply the Lanczos algorithm to the eigenproblem in Eq. (3).

Algorithm 2

---

1. Select arbitrary starting vector $\mathbf{x}$ and normalize it with respect to matrix $\mathbf{K}^{-1}$, i.e., find $\gamma = (\mathbf{x}^T\mathbf{K}^{-1}\mathbf{x})^{1/2}$ and make $\mathbf{x}_1 = \mathbf{x}/\gamma$.
2. Define $\beta_1 = 0$ and $\mathbf{x}_0 = \mathbf{0}$.
3. For $i = 1, 2, ..., q-1$ compute

$$\mathbf{K}_G^{-1}\overline{\mathbf{x}}_{i+1} = \mathbf{K}^{-1}\mathbf{x}_i$$

$$\alpha_i = \overline{\mathbf{x}}_{i+1}^T\mathbf{K}^{-1}\mathbf{x}_i$$

$$\tilde{\mathbf{x}}_{i+1} = \overline{\mathbf{x}}_{i+1} - \alpha_i\mathbf{x}_i - \beta_i\mathbf{x}_{i-1}$$

$$\beta_{i+1} = (\tilde{\mathbf{x}}_{i+1}^T\mathbf{K}^{-1}\tilde{\mathbf{x}}_{i+1})^{1/2}$$

$$\mathbf{x}_{i+1} = \tilde{\mathbf{x}}_{i+1}/\beta_{i+1}$$

The matrix product $\mathbf{x}^T\mathbf{K}^{-1}\mathbf{x}$ is guaranteed to be positive and, therefore, its square root can always be evaluated. Apparently matrices $\mathbf{K}$ and $\mathbf{K}_G$ must be inverted in the above algorithm. However, closer inspection reveals that the inversions are unnecessary. Firstly, the only computation requiring $\mathbf{K}_G^{-1}$ is the first equation in step 3 what can also be written as $\overline{\mathbf{x}}_{i+1} = \mathbf{K}_G\mathbf{K}^{-1}\mathbf{x}_i$. Hence, $\mathbf{K}_G$ must not be inverted and the difficulty associated with its possible singularity is no longer a concern. Secondly, computation of the products of the form $\mathbf{K}^{-1}\mathbf{x}$ is more easily accomplished through consideration of Eq. (2) recast as $\mathbf{K}\boldsymbol{\phi} = \mathbf{x}$. This system is solved by decomposing $\mathbf{K}$ using traditional schemes (Crout, Cholesky, etc.). Notice that $\mathbf{K}$ must be decomposed only once, perhaps in between steps 2 and 3. A more efficient algorithm is:

Algorithm 3

1. Select $\boldsymbol{\phi}_1$ and compute $\mathbf{x}_1 = \mathbf{K}\boldsymbol{\phi}_1$.
2. Decompose $\mathbf{K}$
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. $\mathbf{x}_1 = \mathbf{x}_1/\gamma$, $\mathbf{v} = \boldsymbol{\phi}_1/\gamma$, $\beta_1 = 0$, $\mathbf{x}_0 = \mathbf{0}$
5. For $i = 1, 2, ..., q-1$ compute

   $\overline{\mathbf{x}}_{i+1} = \mathbf{K}_G\mathbf{v}$

   $\alpha_i = \overline{\mathbf{x}}_{i+1}^T\mathbf{v}$

   $\tilde{\mathbf{x}}_{i+1} = \overline{\mathbf{x}}_{i+1} - \alpha_i\mathbf{x}_i - \beta_i\mathbf{x}_{i-1}$

   $\mathbf{v} = \mathbf{K}^{-1}\tilde{\mathbf{x}}_{i+1}$

   $\beta_{i+1} = (\tilde{\mathbf{x}}_{i+1}^T\mathbf{v})^{1/2}$

   $\mathbf{x}_{i+1} = \tilde{\mathbf{x}}_{i+1}/\beta_{i+1}$

   $\mathbf{v} = \mathbf{v}/\beta_{i+1}$

Vector $\mathbf{v}$ is only an auxiliary vector and can be used to store $\boldsymbol{\phi}_1$. Since matrix $\mathbf{K}$ is decomposed the computation of $\mathbf{v}$ in step 5 corresponds, actually, to the solution of the system of linear equations $\mathbf{K}\mathbf{v} = \tilde{\mathbf{x}}_{i+1}$. The starting vector $\boldsymbol{\phi}_1$ can be selected in several ways, one of the most common is to make it coincident with the main diagonal of $\mathbf{K}$. In view of the transformation defined in Eq. (2) the eigenvectors of the original problem Eq. (1) are related to the eigenvectors of the transformed problem Eq. (3) by $\boldsymbol{\phi}_i = \mathbf{K}^{-1}\mathbf{x}_i$.

Consider now the eigenvalue problem with shifting as given in Eq. (4)

$$(\mathbf{K} - \sigma\mathbf{K}_G)\boldsymbol{\phi} = \mu\mathbf{K}_G\boldsymbol{\phi} \tag{4}$$

where σ is the shift and $\mu = \lambda - \sigma$. Employing again the transformation in Eq. (2) and pre-multiplication by $\mathbf{K}_G^{-1}$ yields

$$\mathbf{K}_G^{-1}(\mathbf{K} - \sigma\mathbf{K}_G)\mathbf{K}^{-1}\mathbf{x} = \mu\mathbf{K}^{-1}\mathbf{x} \tag{5}$$

The matrix product $\mathbf{K}_G^{-1}(\mathbf{K} - \sigma\mathbf{K}_G)\mathbf{K}^{-1}$ is intentionally not evaluated. Notice that the inverse of both $\mathbf{K}$ and $\mathbf{K}_G$ appear but they are not explicitly computed in the algorithm proposed below. However, another matrix, $(\mathbf{K} - \sigma\mathbf{K}_G)$, comes into play that must be decomposed. The algorithm with shifting is:

Algorithm 4

1. Select $\boldsymbol{\phi}_1$ and compute $\mathbf{x}_1 = \mathbf{K}\boldsymbol{\phi}_1$.
2. Decompose $(\mathbf{K} - \sigma\mathbf{K}_G)$

3. Compute $\gamma = (\mathbf{x}_1^T \boldsymbol{\phi}_1)^{1/2}$
4. $\mathbf{x}_1 = \mathbf{x}_1/\gamma$, $\boldsymbol{\phi}_1 = \boldsymbol{\phi}_1/\gamma$, $\beta_1 = 0$, $\mathbf{x}_0 = \boldsymbol{\phi}_0 = \mathbf{0}$
5. For $i = 1, 2, ..., q-1$ compute

$$(\mathbf{K} - \sigma\mathbf{K}_G)\overline{\boldsymbol{\phi}}_{i+1} = \mathbf{K}_G\boldsymbol{\phi}_i$$

$$\overline{\mathbf{x}}_{i+1} = \mathbf{K}\overline{\boldsymbol{\phi}}_{i+1}$$

$$\alpha_i = \overline{\mathbf{x}}_{i+1}^T \boldsymbol{\phi}_i$$

$$\tilde{\boldsymbol{\phi}}_{i+1} = \overline{\boldsymbol{\phi}}_{i+1} - \alpha_i\boldsymbol{\phi}_i - \beta_i\boldsymbol{\phi}_{i-1}$$

$$\tilde{\mathbf{x}}_{i+1} = \overline{\mathbf{x}}_{i+1} - \alpha_i\mathbf{x}_i - \beta_i\mathbf{x}_{i-1}$$

$$\beta_{i+1} = (\tilde{\mathbf{x}}_{i+1}^T \tilde{\boldsymbol{\phi}}_{i+1})^{1/2}$$

$$\boldsymbol{\phi}_{i+1} = \tilde{\boldsymbol{\phi}}_{i+1}/\beta_{i+1}$$

$$\mathbf{x}_{i+1} = \tilde{\mathbf{x}}_{i+1}/\beta_{i+1}$$

Because of the form of the eigenproblem shown in Eq. (5) it is more advantageous to store both sequences of eigenvectors $\mathbf{x}_i$ and $\boldsymbol{\phi}_i$ in order to avoid solution of systems of equations of the form $\mathbf{K}\boldsymbol{\phi} = \mathbf{x}$. However, if memory is a critical issue, either $\mathbf{x}$ or $\boldsymbol{\phi}$ only can be stored. Keeping $\boldsymbol{\phi}_i$ may be more convenient because these correspond to the eigenvectors of the original problem. Memory requirements increase considerably not only because vectors $\mathbf{x}$ and $\boldsymbol{\phi}$ must be stored but specially because three matrices must be available: $\mathbf{K}$, $\mathbf{K}_G$ and $(\mathbf{K} - \sigma\mathbf{K}_G)$ decomposed. Since $(\mathbf{K} - \sigma\mathbf{K}_G)$ must be decomposed, when σ is close to one eigenvalue of the eigenproblem stated in Eq. (1), numerical ill-conditioning may happen.

3. Numerical Examples

Consider a problem, similar to the one proposed by Ramaswamy (Ramaswamy, 1980), of finding all the eigenvalues of Eq. (1) when

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

This problem is very simple but it serves to illustrate the difficulties encountered by algorithm 1 when $\mathbf{K}_G$ is indefinite. Selecting the starting vector

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

and blindly applying algorithm 1, one finds that, in the first iteration already, $\beta_2^2 = -0.1181$ and the procedure comes to a halt. If, instead of taking the square root of $\tilde{\boldsymbol{\phi}}_{i+1}^T \mathbf{K}_G \tilde{\boldsymbol{\phi}}_{i+1}$ to compute β_{i+1}, the square root of its absolute value is taken, the algorithm delivers wrong results. In this particular example it would result in the eigenvalues 1.3409, 0.3819, -0.8515, 0.0392 and -0.0809 what is absurd. Algorithm 3, on the other hand, delivers the following eigenvalues and associated eigenvectors arranged columnwise:

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

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

Property	Value
Principal modulus of elasticity E_{11}	154.0 GPa
Principal modulus of elasticity E_{22}	11.13 GPa
In-plane Poisson's ratio ν_{12}	0.304
In-plane shear modulus G_{12}	6.98 GPa
Transverse shear modulus G_{13}	6.98 GPa
Transverse shear modulus G_{23}	3.36 GPa

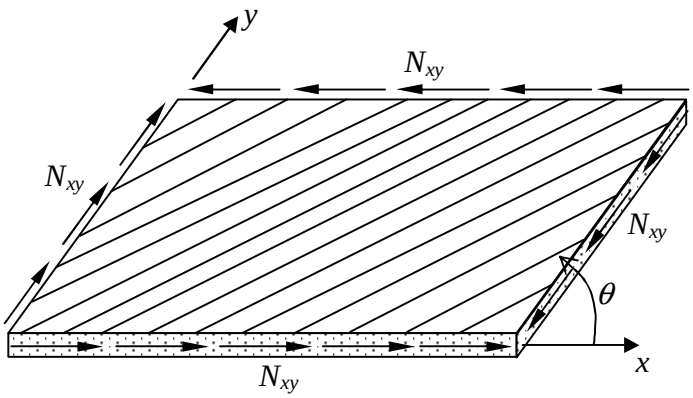

Figure 1. Single ply composite square plate

Consider now a more practical example of a 40 cm $\times$ 40 cm composite square plate made up of a single T300-5208 graphite/epoxy (Table 1) ply of 0.15 mm thickness oriented at an angle θ as illustrated in Fig. 1. This plate is simply supported along the four edges and is subjected to a pure shear loading.

The finite element method is used to generate the stiffness matrix and the geometric stiffness matrix where a 4 $\times$ 4 mesh of bicubic elements has been employed (de Almeida and Hansen, 2000). Running the Lanczos algorithm 3 proposed for $\theta = 45^{\circ}$ yields the following values for the lowest positive and highest negative critical loads (eigenvalues): 20.91 N/m and -3.17 N/m. Physically this result is expected because the compressive normal stresses are acting along the plate diagonal where $x = y$ and the fibers are oriented parallel to that diagonal. On the other hand, when the ply angle is $\theta = -45^{\circ}$, the lowest positive and highest negative critical loads become 3.17 N/m and -20.91 N/m.

A plotting of the lowest positive ($\lambda_{\rm pos}$) and highest negative ($\lambda_{\rm neg}$) eigenvalues versus the fiber orientation angle is displayed in Fig. 2. Notice that $\lambda_{\rm pos}$ and $\lambda_{\rm neg}$ possess equal absolute values for $\theta = 0^{\circ}$ and $\pm 90^{\circ}$. As θ varies the maximum $|\lambda|$ is reached for $\theta = \pm 45^{\circ}$. Should the plate resist against shear loadings of either sign the best strategy is to make $\theta = 0^{\circ}$ or $\theta = 90^{\circ}$.

4. Comments and Conclusions

Other methods for extraction of eigenvalues can be used to obtain the p least buckling loads in problems where the geometric stiffness matrix is indefinite such as the subspace iteration method described below (Bathe, 1996):

1. Solve for $\mathbf{Y}_{k+1}$ in $\mathbf{K}\mathbf{Y}_{k+1} = \mathbf{K}_G\mathbf{X}_k$;
2. Find the projections of $\mathbf{K}$ and $\mathbf{K}_G$;

$$\mathbf{K}_{k+1} = \mathbf{Y}_{k+1}^T\mathbf{K}\mathbf{Y}_{k+1} \qquad \mathbf{K}_{G_{k+1}} = \mathbf{Y}_{k+1}^T\mathbf{K}_G\mathbf{Y}_{k+1}$$

3. Solve projected eigenproblem

$$\mathbf{K}_{k+1}\mathbf{Q}_{k+1} = \mathbf{K}_{G_{k+1}}\mathbf{Q}_{k+1}\mathbf{\Lambda}_{k+1}$$

4. Find improved eigenvectors

$$\mathbf{X}_{k+1} = \mathbf{Y}_{k+1}\mathbf{Q}_{k+1}$$

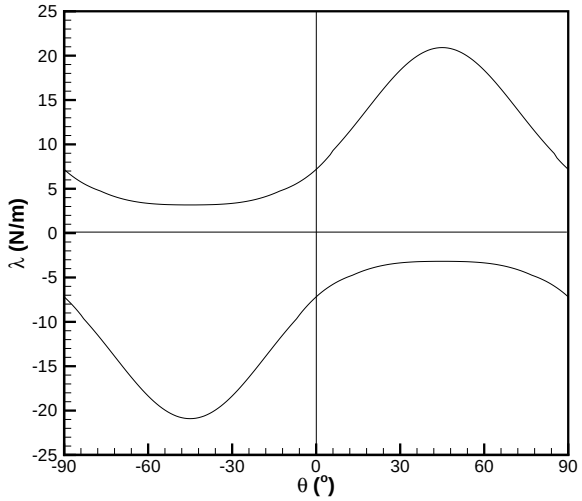

Figure 2. Critical load variation

As the above procedure progresses $\mathbf{\Lambda}_{k+1} \rightarrow \mathbf{\Lambda}$ and $\mathbf{X}_{k+1} \rightarrow \mathbf{\Phi}$, where $\mathbf{\Lambda}$ is a diagonal matrix storing the p least eigenvalues and $\mathbf{\Phi}$ is a matrix whose columns correspond to the associated p eigenvectors. Since $\mathbf{K}$ is positive-definite, so is the projected matrix $\mathbf{K}_{k+1}$. However, since $\mathbf{K}_G$ is indefinite, so is the projected matrix $\mathbf{K}_{G_{k+1}}$. Therefore, care should be exercised when solving the projected eigenproblem in step 3. The recommended generalized Jacobi method (Bathe, 1996) should be avoided because it assumes that both $\mathbf{K}_{k+1}$ and $\mathbf{K}_{G_{k+1}}$ are positive-definite. On the other hand, the standard Jacobi method (Bathe, 1996) can be employed with a few modifications. Since $\mathbf{K}_{k+1}$ is positive-definite it admits a Cholesky decomposition of the form $\mathbf{K}_{k+1} = \mathbf{S}_{k+1}\mathbf{S}_{k+1}^T$, where $\mathbf{S}_{k+1}$ is a lower triangular matrix. Inversion of $\mathbf{S}_{k+1}$ is not troublesome because the dimension of the projected eigenproblem is small. Hence, the projected eigenproblem can be recast in the form

$$\mathbf{S}_{k+1}^{-1}\mathbf{K}_{G_{k+1}}\mathbf{S}_{k+1}^{-T}\mathbf{Q}_{k+1} = \mathbf{Q}_{k+1}\mathbf{\Upsilon}_{k+1}$$

where $\mathbf{\Upsilon}_{k+1} = \mathbf{\Lambda}_{k+1}^{-1}$ and the eigenvalues of $\mathbf{S}_{k+1}^{-1}\mathbf{K}_{G_{k+1}}\mathbf{S}_{k+1}^{-T}$ can extracted using the standard Jacobi method.

A theoretical study of the Lanczos method applied to the eigenproblem described in Eq. (3) shows that the main contribution to the vector $\mathbf{x}_{i+1}$ is from the eigenvector $\boldsymbol{\phi}_i$, where $\boldsymbol{\phi}_i$ is associated with the ith least eigenvalue in absolute value. In other words, vector $\mathbf{x}_2$ is composed mainly of $\boldsymbol{\phi}_1$; vector $\mathbf{x}_3$ is composed mainly of $\boldsymbol{\phi}_2$; so on so forth, considering that $|\lambda_1| < |\lambda_2| < ... < |\lambda_n|$. Therefore, convergence to the least eigenvalues in absolute value and associated eigenvectors is anticipated. In practical applications, where the loadings can be applied in either direction, the effective buckling load corresponds to the least eigenvalue in absolute value. Hence, in these situations, the modified Lanczos algorithm proposed herein is of great applicability.

5. Acknowledgements

This work was partially financed by the Brazilian agency CNPq (grant no. 304642/2003-7).

6. References

1. Lanczos C. 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* 1950; **45**: 255-282.

2. Paige CC. The computation of eigenvalues and eigenvectors of very large sparse matrices, Ph.D. Thesis, University of London, 1971.

3. Paige CC. Computational variants of the Lanczos method for the eigenproblem. *Journal of the Institute of Mathematics and its Applications* 1972; **10**: 373-381.

4. Parlett BN, Scott DS. The Lanczos algorithm with selective orthogonalization. *Mathematics of Computation* 1979; **33**(145): 217-238

5. Ericsson T, Ruhe A. The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric eigenvalue problems. *Mathematics of Computation* 1980; **35**(152): 1251-1268.

6. Ramaswamy S. On the effectiveness of the Lanczos method for the solution of large eigenvalue problems. *Journal of Sound and Vibration* 1980; **73**(3):405-418.

7. Bathe K-J. *Finite Element Procedures*. Prentice Hall: New Jersey, 1996.

8. Zienkiewicz OC. *The Finite Element Method, Vol. 2: Solid and Fluid Mechanics, Dynamics and Non-linearity*. 4th ed., McGraw-Hill: London, 1991.

9. de Almeida SFM, Hansen JS. Natural frequencies of composite plates with thermal residual stresses. *International Journal of Solids and Structures* 2000; **36**(23):3517-3539.

7. Responsibility notice

The author is the only responsible for the printed material included in this paper.