

XIV Convegno Nazionale dell'Associazione Italiana
per l'Analisi delle Sollecitazioni
Catania 23-27 Settembre 1986

L'APPLICAZIONE DEL METODO DI LANCZOS
ALL'ANALISI DINAMICA DELLE STRUTTURE

Filippo Bertolino e Silvio Merazzi

Département de génie civil
Ecole polytechnique fédérale de Lausanne
CH - 1015 Lausanne, Suisse

Sommario

Vengono illustrati i problemi posti dal calcolo di alcuni autovalori ed autovettori di grandi matrici sparse e simmetriche derivanti dal MEF. Viene analizzato l'algoritmo di Lanczos e la sua implementazione nel codice di calcolo strutturale B2000. Si presentano alcuni esempi e confronti.

Abstract

The Lanczos method for extracting eigenvalues of large, sparse symmetric matrices arising from structural vibration problems is presented. An outline of the method and a brief description of its implementation in the B2000 modular FEM code are given. Some typical examples and comparisons are included in the report.

Introduzione

Data una matrice simmetrica $[A]$ di dimensione N , il problema standard agli autovalori consiste nel determinare uno scalare λ ed un vettore $x \neq 0$ tale che

$$[A]\{x\} = \lambda[I]\{x\} \quad (1)$$

Nei problemi strutturali si ha a che fare con il problema generalizzato:

$$[K]\{x\} = \omega^2[M]\{x\} \quad (2)$$

Nell'analisi dinamica è richiesto un numero prefissato di frequenze nell'intorno di uno shift s ed i relativi modi di vibrare. Per questi problemi la matrice di rigidità $[K]$ e la matrice di massa $[M]$ sono entrambe reali, simmetriche e semidefinite positive. Gli autovalori sono tutti positivi. I problemi di stabilità richiedono un prefissato numero di autovalori tra i più piccoli. In questo caso, la matrice di rigidità $[K]$ è reale, simmetrica, definita positiva e la matrice di rigidità differenziale $[M]$ è una matrice simmetrica indefinita. Gli autovalori possono essere positivi o negativi.

Tra i metodi che godono di maggiore popolarità, ricordiamo quello di Householder, il metodo del determinante, il subspace method. Mentre il primo è del tutto inadeguato a risolvere i problemi con grandi matrici sparse, il metodo del determinante è consigliabile solo se la banda delle matrici da trattare è molto stretta. Il subspace method è invece molto efficiente ed è ancora oggetto di numerose ricerche: oltre alla sua forma base, oggi sono disponibili diverse tecniche di accelerazione.

Nel 1950 Lanczos [1] presentò il metodo che oggi porta il suo nome. Per lungo tempo tale algoritmo fu dimenticato a causa del suo strano comportamento in presenza degli errori di arrotondamento e solo recentemente molti ricercatori ne hanno rivalutato l'utilità. Esso può essere usato sia per risolvere i sistemi di equazioni lineari [2],[16] sia per il calcolo di alcune copie (λ, x) del problema (1). La matrice $[A]$ non deve necessariamente essere definita positiva. Il numero di operazioni necessarie per ottenere una soluzione dipende in gran parte dalle caratteristiche della matrice e dalla precisione richiesta: spesso per il calcolo degli autovalori sono necessari meno di $3\sqrt{N}$ passi.

In questo articolo ci proponiamo di esaminare i problemi posti da una sua efficiente implementazione in un programma di calcolo automatico.

Descrizione dell'algoritmo

L'algoritmo di Lanczos può essere descritto semplicemente come l'applicazione dell'ortogonalizzazione di Gram-Schmidt ai vettori di Krylov $\{k\}$

$$\{k\}_j = [\{q\}_1, [A]\{q\}_1, \dots, [A]^{j-1}\{q\}_1]$$

Bertolino, F. - Merazzi, S.

Benché a prima vista questa operazione sembri molto onerosa, in realtà essa è assai semplice: infatti ad ogni passo è necessario ortogonalizzare solo il vettore

$[A]^{j-1}\{q\}_1$ rispetto agli ultimi due vettori $\{q\}_{j-1}$ e $\{q\}_j$.

I vettori risultanti da questa operazione costituiscono una base ortogonale $[Q]_j$ del sottospazio di Krylov e vengono chiamati vettori di Lanczos. In questa base, la proiezione di $[A]$ in $[k]_j$ è rappresentata da una matrice tridiagonale simmetrica $[T]_j$:

$$[T]_j = [Q]_j^t [A] [Q]_j$$

dove $[Q]_j = [\{q\}_1, \{q\}_2, \{q\}_3, \dots, \{q\}_j]$.

Se $[T]_j = [S]_j [D]_j [S]_j^t$

rappresenta la trasformazione spettrale di $[T]_j$, allora

$$[D]_j = \text{diag}[\theta_1, \theta_2, \dots, \theta_j] \quad \text{e} \quad [Y] = [Q]_j [S]_j$$

sono le migliori approssimazioni di Rayleigh-Ritz della matrice $[A]$ a partire dal sottospazio $[Q]_j$ di Lanczos.

Nel caso fossero disponibili alcune soluzioni approssimate, piuttosto che partire con un vettore unitario o random $\{q\}_1$, è preferibile generare un vettore di partenza come combinazione lineare di tali vettori. Ciò accelera la convergenza ed è molto utile nei problemi adattativi.

L'algoritmo di Lanczos ha una proprietà assai attraente quando si tratta con grandi matrici sparse: la matrice $[A]$ vi entra solo attraverso il prodotto $[A]\{x\}$, quindi non viene modificata durante il calcolo. Nel problema generalizzato (2) è però consigliabile trasformare il problema come segue:

- 1) fattorizzare la matrice $[K]$ secondo Cholesky

$$[K] = [L][L]^t$$

quindi porre

$$\{y\} = [L]^t \{x\} \quad \text{e} \quad ([L]^{-1} [M] [L]^{-t}) \{y\} = \bar{\lambda} \{y\}.$$

- 2) Ad ogni passo moltiplicare $[A]\{q\}_j$ attraverso le seguenti operazioni:

$$[L]^t \{z\} = \{q\}_j \quad \text{sostituzione all'indietro}$$

$$[M]\{z\} = \{a\}$$

$$[L]\{x\} = \{a\} \quad \text{sostituzione in avanti}$$

L'inversione dell'operatore provoca la trasformazione dello spettro con un aumento della velocità di convergenza. Ciò non ostante si tratta di un'operazione molto costosa, specie se la matrice [M] non è diagonale, e quindi sono allo studio delle tecniche che permettano di ottimizzarla. Una di queste consiste nel risolvere il sistema col metodo di Lanczos preconditionato.

In generale come il processo di costruzione di $[T]_j$ e $[Q]_j$ procede, si ha una perdita di ortogonalità tra i vettori di Lanczos che possono anche diventare linearmente dipendenti. In questa situazione le corrispondenti matrici $[T]_j$ non sono più le proiezioni ortogonali della matrice [A] nell'associato spazio di Krylov. Ciò non ostante tra gli autovalori delle matrici $[T]_j$ si trovano alcuni tra gli autovalori della matrice [A], tra numerosi autovalori spurii [12].

La spiegazione di tale comportamento fu fornita da Paige [3] nel 1971. Fino ad allora l'unica strategia per stabilizzare l'algoritmo fu quella consistente nel ri-ortogonalizzare ad ogni passo i vettori di Lanczos [8],[9],[10],[11]. Paige [3],[4] ha mostrato che la causa della perdita di ortogonalità è l'azione combinata della convergenza di alcuni autovalori: l'algoritmo può essere stabilizzato incorporandovi la ri-ortogonalizzazione dell'ultimo vettore generato rispetto ad ogni "buon" vettore di Ritz, in quanto è nella loro direzione che si ha perdita di ortogonalità. Un vettore di Ritz $\{y\}$ è definito "buono" quando

$$\| [A]\{y\} - \theta\{y\} \| \leq \sqrt{\epsilon} \| [T]_j \| \quad (3)$$

dove ϵ indica la precisione relativa del calcolatore. Un aspetto positivo dell'algoritmo di Lanczos è che per il controllo della convergenza e della norma residua (3) non è richiesto il calcolo esplicito dei vettori di Ritz [12].

Un'altra strategia consiste nell'evitare ogni tipo di ri-ortogonalizzazione e nell'isolare nello spettro di $[T]_j$ gli autovalori spurii da quelli della matrice [A] [5],[6],[7]. Questa tecnica presenta dei vantaggi in termini di occupazione di memoria, ma richiede un numero maggiore di passi. Poiché ad ogni passo è necessario il calcolo del prodotto $[A]\{x\}$, l'operazione di gran lunga più onerosa nei nostri problemi, abbiamo concentrato la nostra attenzione sulle tecniche di ri-ortogonalizzazione controllata.

L'idea base è mantenere un certo livello di indipendenza tra i vettori di Lanczos, misurata da:

$$K_j = \| [I] - [Q]_j [Q]_j^t \|$$

Nel caso dell'ortogonalizzazione esatta si ha $K_j = \epsilon$. Parlett e Scott [14],[15] hanno mostrato che anche la semi-ortogonalità ($K_j = \sqrt{\epsilon}$) consente di stabilizzare il metodo.

Il problema è ora il calcolo di una stima economica di tale quantità. Due tecniche si sono imposte per la loro efficienza, la prima proposta da Parlett [13], la seconda da Simon [16]. Il metodo di Simon è più semplice, ma richiede più memoria ed un numero maggiore di prodotti interni nel processo di ortogonalizzazione; esso è per altro più adatto nella soluzione dei sistemi lineari $[A]\{x\} = \{b\}$.

Fin tanto che K_j è piú piccolo di $\sqrt{\epsilon}$, si procede con la formula ricorsiva di Lanczos. Seguendo Parlett e Scott [14], quando K_j supera la tolleranza, calcoliamo gli autopaia della matrice $[T]_j$ e controlliamo le norme residue (3) dei vettori di Ritz. Calcoliamo quelli che hanno raggiunto la convergenza ed i vettori $\{q\}_j$ e $\{q\}_{j+1}$ sono ortogonalizzati rispetto ad essi.

Poiché per il calcolo dei vettori di Ritz sono necessari tutti i vettori di Lanczos, questi devono essere conservati o nella memoria virtuale o su disco e richiamati sequenzialmente.

Gli autovalori multipli possono essere calcolati con la tecnica del restart. Si genera un vettore random ortogonale ai vettori di Ritz già calcolati e si riparte con un nuovo run. Il nuovo sottospazio deve essere ortogonale al precedente [17].

Se la memoria disponibile è particolarmente limitata, si può usare l'algoritmo in modo iterativo. Fissata, in base allo spazio disponibile, la dimensione massima LSUB del sottospazio, si procede al calcolo delle soluzioni desiderate. Se al passo LSUB l'obiettivo non è stato ancora raggiunto, si riparte con un nuovo vettore di partenza. Questo è scelto come combinazione lineare di alcuni tra i vettori di Ritz che non hanno ancora raggiunto la convergenza [14],[17].

L'implementazione

L'algoritmo è stato implementato nel programma strutturale B2000 [18]. Esso è costituito da una serie di programmi modulari fisicamente indipendenti, ognuno dei quali esegue un compito definito, come per esempio:

- input dei dati e della topologia;
- calcolo della seconda variazione dell'energia potenziale (matrice di rigidità, di stabilità, di massa, ...);
- assemblaggio delle matrici e dei vettori locali (tecnica delle sottostrutture);
- soluzione dei sistemi di equazioni lineari (tecniche a matrice sparsa o in forma skyline) con metodi diretti (Crout) o iterativi (gradiente coniugato, metodo di Lanczos);
- calcolo degli sforzi e delle reazioni;
- analisi dinamica (metodo di Stodola-Vianello, subspace method, metodo di Lanczos);
- post-processor compreso l'output grafico.

Il requisito più importante di un sistema modulare è la definizione precisa e coerente dei dati. Essi devono essere auto-descrittivi e trasparenti, cioè l'utente non deve preoccuparsi del loro tipo, dimensione e localizzazione. Per far questo il B2000 ricorre al sistema di gestione dei dati MEMCOM [19] che si incarica del loro trasferimento tra i files e la memoria. Esso prevede la manipolazione di insiemi di dati (vettori, matrici, iper-matrici), di tavole relazionali (in cui per ogni dato è descritto il nome, il tipo e la dimensione) ed insiemi di dati relazionali.

In oltre tale sistema consente la manipolazione simbolica dei dati, per esempio è possibile moltiplicare due matrici semplicemente indicandone il nome.

Esempi

Problema 1

Per studiare il comportamento dell'algoritmo in presenza d'autovalori multipli, consideriamo il calcolo delle prime 3 frequenze di una piastra piana quadrata in acciaio, di 16.0 inch. di lato e di 0.5 inch. di spessore, semplicemente appoggiata sui quattro lati (fig. 1). Modellizziamo la struttura con 16 elementi flessionali isoparametrici a 8 nodi: ne risultano 390 gradi di libertà. Dopo aver calcolato 3 autovalori distinti, l'algoritmo procede ad un controllo sulla molteplicità. Preso un vettore random ortogonale agli autovettori già calcolati, costruisce un nuovo sottospazio ortogonale al precedente e dopo pochi passi individua un autovalore multiplo. Ad un controllo successivo l'algoritmo termina. Sono state calcolate 4 soluzioni, una in più del necessario: poichè in generale non è possibile conoscere a priori il numero di autolavori multipli, prima di procedere al calcolo bisogna riservare nel data base uno spazio più grande di quello strettamente necessario; detto NRITZ il numero di soluzioni richiesto ed MXRITZ il numero massimo di soluzioni calcolabile, poniamo

$$MXRITZ = \text{MIN}(2*NRITZ, NRITZ+8).$$

Tabella 1 - Frequenze piastra quadrata

Modo	Frequenze [hz]	
	subspace method	Lanczos
1	356.18	356.07
2	909.56	909.42
3	909.56	909.42

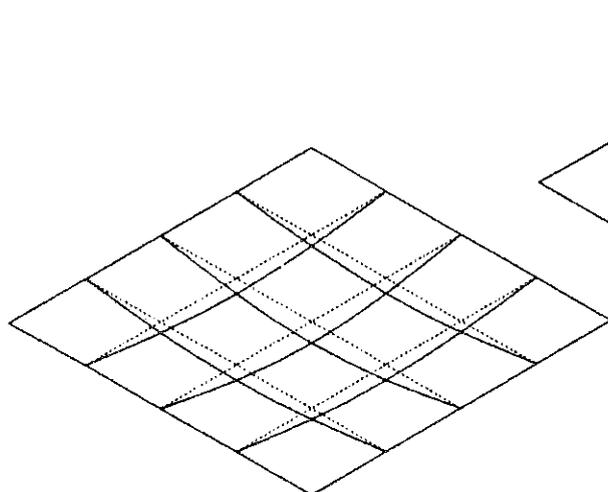


Fig. 1 - Geometria piastra, modo 1

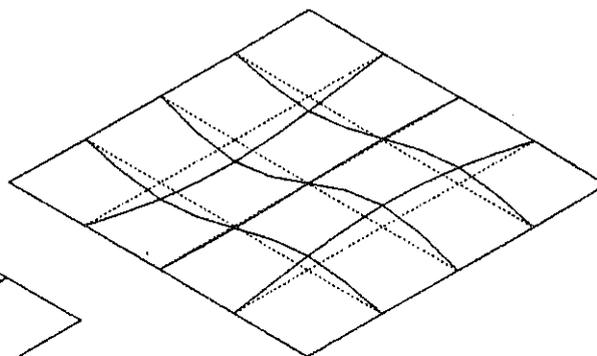


Fig. 2 - Modo 3

Problema 2

Consideriamo il calcolo dei primi 5 autovalori di un cilindro in acciaio, lungo 12.0 inch., di 3.0 inch. di raggio e di 0.01 inch. di spessore, incastato ai due estremi. Vista la simmetria, modellizziamo solo la parte indicata nella figura 3. Usiamo 64 elementi isoparametrici a 8 nodi di Ahmad: ne risultano 233 nodi e 1398 gradi di libertà. Confrontiamo i risultati del subspace method e dell'algoritmo di Lanczos: quest'ultimo risulta di oltre il 50% più veloce .

Tabella 2 - Frequenze cilindro

Modo	Frequenze [hz]	
	Subspace method	Lanczos
1	590.15	589.12
2	705.09	705.37
3	948.24	946.64
4	1056.70	1055.85
5	1331.80	1317.11

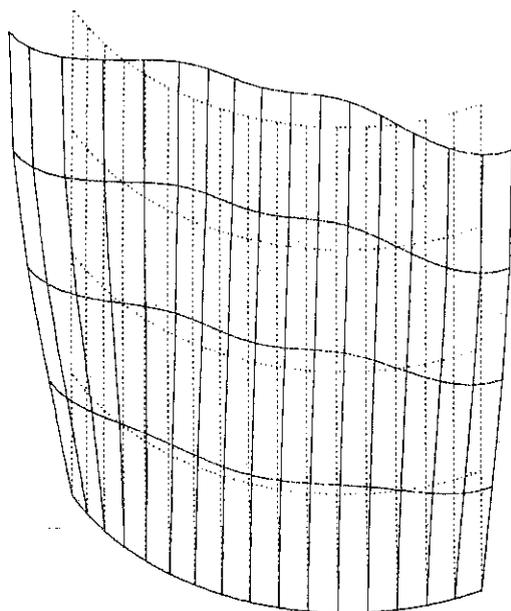


Fig. 3 - Geometria, modo 3

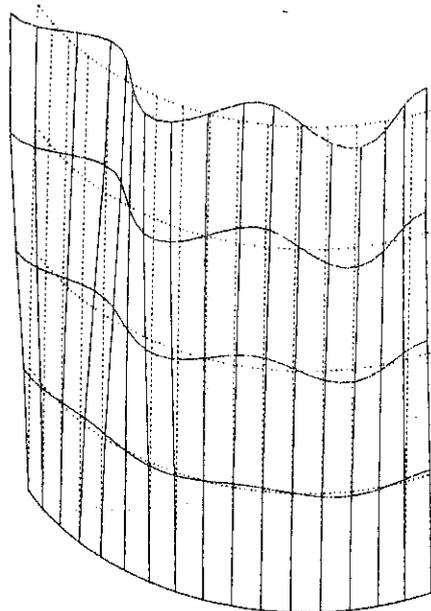


Fig. 4 - Modo 4

Problema 3

Calcoliamo le prime 10 frequenze di una passerella pedonale in acciaio (fig. 5). Si tratta di una costruzione dei primi anni del secolo in cui nel disegno di alcuni particolari costruttivi non fu preso in debito conto il problema delle vibrazioni. Oggi è necessario proporre delle soluzioni che non alterino l'aspetto di un'opera di particolare pregio storico ed architettonico.

Il modello è costituito da 480 travi, 276 nodi e 1078 gradi di libertà. L'algoritmo calcola 14 soluzioni in quanto individua un gruppo di 4 autovalori multipli.

Presentiamo un esempio sull'uso interattivo del programma B2000. Dopo aver letto i dati relativi alla geometria, ai vincoli ed ai materiali della passerella, il B2000 procede al calcolo delle matrici elementari ed all'assemblaggio delle matrici globali. Il modulo LAN legge le istruzioni in grassetto ed esegue l'analisi dinamica.

Bertolino, F. - Merazzi, S.

\$ RUN LAN

B2000 LANCZOS EIGENFREQUENCY SOLVER (LAN) VERSION 1.1

LAN-COMMAND ? CO=P.CO

DATA BASE FILE OPENED, FILE NAME= P.CO, UNIT=2

LAN-COMMAND ? FACTOR NEIG=10 LSUB=100 MULT

LAN-COMMAND ? GO

MAX. NUMBER OF RITZ VECTORS DEFINED BY MULTIPLICITY CHECK (18)
EIGENVALUE ERROR TOLERANCE CHANGED TO 3.7253E-09
LANCZOS ITERATION REQUIRED 117.15 CP-SEC.

FINAL "GOOD" EIGENVALUES

MODE	FREQUENCY	LAMBDA	OMEGA
1	2.96707	347.549	18.6427
2	3.19836	403.844	20.0959
3	3.34692	442.233	21.0293
4	3.34694	442.237	21.0294
5	3.34694	442.237	21.0294
6	3.34757	442.405	21.0334
7	4.42745	773.867	27.8185
8	4.86824	935.631	30.5881
9	6.80540	1828.38	42.7596
10	8.18533	2645.04	51.4299

RESIDUAL ERROR HAS BEEN COMPUTED ON NEIG= 10 EIGENVALUES:

1	6.861637E-13	2	5.525455E-13	3	5.852457E-11	4	6.825189E-10
5	4.158720E-09	6	1.371953E-12	7	1.081089E-11	8	2.959598E-12
9	7.042113E-10	10	3.801766E-11				

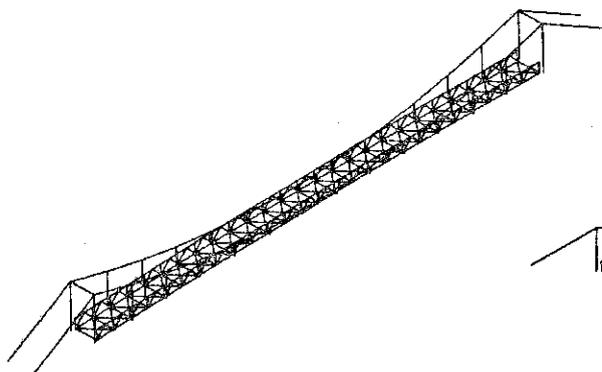


Fig. 5 - Geometria passerella



Fig. 6 - Modo di vibrazione n. 9

Conclusioni

In base alla nostra esperienza, l'algoritmo di Lanczos risulta più efficiente del subspace method sia in termini d'occupazione di memoria che in termini di tempo di calcolo. Sui calcolatori da noi utilizzati (VAX 785, APOLLO, CYBER 855) esso risulta più veloce di almeno il 50%.

Esso è particolarmente interessante nella soluzione dei problemi auto-adattativi, specie se implementato su un calcolatore vettoriale. D'altra parte la sua programmazione è più complicata ed il suo uso richiede una certa esperienza.

Le ricerche attuali si concentrano su due punti: da un lato, sull'applicazione di una serie di trasformazioni spettrali per mezzo di opportuni shift; dall'altro, sulla ottimizzazione del passaggio dal problema generalizzato (2) a quello standard (1).

Nomenclatura

[K]	Matrice di rigidezza
[M]	Matrice di massa
[L]	Matrice triangolare inferiore
s	Shift
ω^2	Autovalori del problema generalizzato
ϵ	Precisione relativa del calcolatore
j	j-esimo passo dell'algoritmo
[Q] _j	Sottospazio di Lanczos
[k] _j	Sottospazio di Krylov
{q} _j	j-esimo vettore di Lanczos
{y}	Vettore di Ritz
[Y]	Vettori di Ritz
[T] _j	Matrice tridiagonale simmetrica di ordine j.
[S] _j	Autovettori della matrice [T] _j
[D] _j	Autovalori della matrice [T] _j
N	Numero di variabili nel sistema globale
NEIG	Numero di soluzioni richiesto
NRITZ	Numero di soluzioni calcolato
MXRITZ	Numero massimo di soluzioni calcolabili
LSUB	Dimensione massima del sottospazio
K _j	Stima della perdita di ortogonalità tra i vettori di Lanczos

Bertolino, F. - Merazzi, S.

Bibliografia

- [1] C. LANCZOS, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators", J.Res.Nat. Bur.Standards, vol. 45, 1950, pp. 255-282.
- [2] C. LANCZOS, "Solution of systems of linear equations by minimized iterations", J.Res.Nat.Bur.Standards, vol. 49, 1952, pp. 33-53.
- [3] C.C. PAIGE, "The computation of eigenvalues and eigenvectors of very large sparse matrices", Ph.D. Thesis, University of London 1971.
- [4] C.C. PAIGE, "Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix", J.Inst.Maths.Applis., vol.18, 1976, pp. 341-349.
- [5] J. CULLUM & R.A. WILLOUGHBY, "Computing eigenvalues of very large symmetric matrices - An implementation of a Lanczos algorithm with no reorthogonalization", J. Comput. Phys. 44,329-358 (1981).
- [6] J.T. EDWARDS, D.C. LICCIARDELLO & D.J. THOULESS, "Use of the Lanczos method for finding complete sets of eigenvalues of large sparse symmetric matrices", J.Inst.Math.Applis (1979) 23,277-283.
- [7] B.N. PARLETT & J.K. REID, "Tracking the progress of the Lanczos algorithm for large symmetric eigenproblems", IMA J.Numer.Anal., vol. 1, 1981, pp. 135-155.
- [8] P.C. CHOWDHURY, "The truncated Lanczos algorithm for partial solution of the symmetric eigenproblem", Computer & Structures v.6,pp.439-446, 1976.
- [9] M. GERADIN, "On the Lanczos method for solving large structural eigenvalue problems", ZAMM 59, T 127 -T 129 (1979).
- [10] V.I. WEINGARTEN, R.K. RAMANATHAN & C.N. CHEN, "Lanczos eigenvalue algorithm for large structures on a minicomputer", Computer & Structures vol. 16, n. 1-4, pp. 253-257, 1983.
- [11] T. ERICSSON & A. RUHE, "The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric eigenvalue problems", Math. Comp., vol. 35, n. 152, 1980, pp. 1251-1268.
- [12] B.N. PARLETT, "The Symmetric Eigenvalue Problem", Prentice Hall, Englewood Cliffs, 1980.