p- 40

Contractor Report

NASA Grant #NAG-3-895

SAPNEW: Parallel Finite Element Code for Thin Shell Structures on the Alliant FX/80

(NASA-CR-190663) SAPNEW: PARALLEL FINITE ELEMENT CODE FOR THIN SHELL STRUCTURES ON THE ALLIANT FX/80 (Georgia Inst. of Tech.) 40 p

N93-10372

Unclas

G3/39 0115577

Manohar P. Kamat
Brian C. Watson

School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

February 1992

Contents

1. INTRODUCTION

This report summarizes the results of a research activity aimed at providing a finite element capability for analyzing turbo-machinery bladed-disk assemblies in a vector/parallel processing environment.

Analysis of aircraft turbo fan engines is very computationally intensive. Problems involving aeroelastic stability and response of bladed-disk assemblies in aircraft turbo fan engines are among the most difficult problems encountered. Complications in these studies arise from the small differences between individual blades known as mistuning. Previous researchers have come to believe that the static, flutter, and forced response of mistuned turbo-machinery blades can be studied by analyzing each blade separately in either a pure bending or a pure torsional motion.[1] However, with the development of thin blades with high sweep, it is necessary to model the coupled behavior. This requires a finite element analysis using shell elements, which is time consuming on a sequential computer. Concurrent (parallel) processing seems to offer the greatest promise for such an analysis.

The performance limit of modern day computers with a single processing unit has been estimated at 3 billions of floating point operations per second (3 gigaflops). In view of this limit of a sequential unit, performance rates higher than 3 gigaflops can be achieved only through vectorization and/or parallelization as on Alliant FX/80. Accordingly, the efforts of this critically needed research have been geared towards developing and evaluating parallel finite element methods for static and vibration analysis. A special purpose code, named with the acronym SAPNEW, performs static and eigen analysis of multi-degree-of-freedom blade models built-up from flat thin shell elements.

SAPNEW grew out of the well-known SAP IV and SAP V codes[2,3]. The flat thin shell element, as well as the beam element in SAPNEW were taken directly from the SAP IV and SAP V codes. These were integrated in a finite element code that uses a skyline storage scheme for the assembled mass and stiffness matrices[4] as well as efficient solution schemes for static and eigen analysis designed to accomodate this compact storage method.

The objective behind this concurrent code development on the Alliant FX/80 was to provide a stand alone capability for static and eigen analysis. The output of this program was designed to easily integrate into the input of another concurrent code, known by the acronym ASTROP, for aeroelastic studies[5]. A preprocessor, which accepts NASTRAN input decks and converts them to SAPNEW format, was added to make SAPNEW more user friendly and more readily used by researchers at NASA Lewis Research Center.

2. DESCRIPTION OF SAPNEW

SAPNEW is a finite element code for static and eigen analysis of three-dimensional, thin shell structures, particularly turbo-machinery blades. Structures may be modeled with triangular or quadrilateral flat elements with uncoupled in-plane and bending stiffnesses. Coupling between the in-plane and bending stiffnesses is achieved through assembling non-coplanar elements. Loading of the structure may be due to concentrated loads, normal pressure, thermal effects, uniform acceleration, and/or centrifugal acceleration.

Static Analysis

Linear static analysis may be performed on a model to generate deformation and stress information.

Eigen Analysis

Eigen value/vector analysis may be performed on a model to generate natural frequencies and mode shapes. This analysis may include geometric stiffening of the model due to applied loads and centrifugal effects.

Shell Element

Stiffness matrices

The primary modeling element of the SAPNEW program is a thin shell element. For details of the formulation of this element, consult reference [6]. A CST (constant strain triangular) element models the in-plane behavior. A CST element has six degrees of freedom. A quadrilateral element is formed by the assembly of four CST elements followed by a static condensation procedure to eliminate the interior node to leave eight degrees of freedom.

The bending behavior is modeled by a partially constrained assemblage of three LCCT (linear curvature compatible triangular) elements. Each LCCT element has ten degrees of freedom. Static condensation eliminates the internal node of the assemblage and the constraint of linearly varying curvature eliminates the mid-side degrees of freedom. The resulting triangular element (designated LCCT-9) has nine degrees of freedom. Normal twisting degrees of freedom are then added for the transformation to global coordinates, although no stiffnesses are associated with these degrees of freedom in the local coordinate system. The quadrilateral element is formed from an assembly of four LCCT-9 elements followed by static condensation to eliminate the internal node.

With the in-plane and bending properties combined, the resulting element has six degrees of freedom at each node (three displacements and three rotations).

In calculating the stiffness matrices, the program may (at user's option) use different constitutive (stress-strain) relationships for the in-plane and the bending behaviors. In this way, material properties typical of laminated composites may be simulated.

Mass matrix

The mass matrix for the thin shell element is formed using a lumped mass methodology. The total mass for the element is distributed evenly among the nodes and assigned to the displacement degrees of freedom. No values of rotary inertia are assigned to the rotation degrees of freedom.

Geometric stiffness matrices

The effect of in-plane stresses on the bending stiffnesses of an element is handled through the calculation of geometric stiffness matrices. Then, for initially stressed structures, or for analysis of structures subject to geometric non-linearities, the geometric stiffness matrices are scaled with the stress resultants and added to the element's stiffness matrix to create a "stressed element" stiffness matrix.

In calculating the geometric stiffness matrices, the program uses a linear interpolation for the normal displacement. Although this is a lower order of approximation than that used for the element stiffness matrix, this is consistent in an energy sense.

Auxiliary Elements

SAPNEW provides a three-dimensional beam element with twelve degrees of freedom and a two degree of freedom linear linear spring element as auxiliary elements. The intended use of these elements is for modeling elastic supports for the structure (e.g. to include the effects of an elastic rotor disk in a turbine blade analysis). Thus, these elements have not been optimized for concurrency and vectorization beyond automatic compiler optimizations and their use should be limited.

Centrifugal forces

SAPNEW calculates the effective load due to constant rotation using the lumped mass matrix previously described.

Multi-Point Constraints

In addition to fixed single-point constraints, SAPNEW allows constraints wherein one degree of freedom is determined by a linear combination of up to four other degrees of freedom. This allows semi-fixed supports, as well as rigid members to be modeled. Note that the degrees of freedom, upon which a multi-point constrained degree of freedom depends, may not themselves be multi-point constrained.

3. PARALLELIZATION OF SAPNEW

Because of the tremendous computational effort involved in performing an aeroelastic analysis on a bladed disk assembly, improvements in program performance are very important. Parallel and/or vector processing seems to provide the best hope for improved computing speed. For this reason, SAPNEW was intended for use on a parallel processing computer (e.g. the Alliant FX/80). Several aspects of the program were designed for improved parallel efficiency.

Element Generation

During the element generation phase, the program calculates the element stiffness matrices and element mass matrices. These calculations are independent and thus, are well suited to concurrent execution. SAPNEW does perform all shell element calculations in parallel.

Linear Equation Solution

Crout decomposition (LDL^T) or Cholesky decomposition (LL^T) (for positive definite systems) are well known direct methods for the solution of a linear system. These algorithms are popular partly because they can take advantage of a compact "skyline" storage scheme for the stiffness matrix, although there can be substantial fill-in below the skyline.

These methods were designed for sequential operation. However, careful examination of the algorithms shows that there are operations which can be performed concurrently. The LL^T algorithm is given in Figure 1.

For i = 1 to n

$$L_{ii} = \sqrt{K_{ii} - \sum_{k=1}^{i-1} L_{ik}^2}$$

For j=i+1 to n

$$L_{ji} = \frac{K_{ji} - \sum_{k=1}^{i-1} L_{ik} L_{jk}}{L_{ii}}$$

Next j

Next i

Figure 1. Cholesky decomposition algorithm.

The calculations in the inner loop (j-loop) in the LL^T algorithm are independent of each other. Thus, this loop can be executed concurrently. Note, however, that the number of tasks to be performed in this loop changes with i. As i gets close to n, there are fewer tasks to perform, and consequently, there is little benefit from parallelization at this point. This fact limits the parallel efficiency that this algorithm can achieve.

After the matrix is factored, the solution is obtained by first forward substituting to solve $[L]\{y\} = \{F\}$ and then back-substituting to solve $[L]^T\{q\} = \{y\}$. These substitutions are inherently sequential operations and further limit the application of parallel processing to this algorithm. Thus, it is desirable to explore alternate algorithms on parallel machines.

Element-by-element preconditioned conjugate gradient (EBE-PCG) algorithms have been advocated for use in parallel/vector environments as being superior to the LDL^T decomposition algorithm. The conjugate gradient algorithm involves generating a set of mutually conjugate direction vectors. The quadratic total potential energy function is then minimized successively along each direction. Using exact arithmetic, it can be shown[7] that this algorithm will require at most n iterations to find the solution for an n degree of freedom problem. This property makes the conjugate gradient algorithm attractive among iterative methods. A version of the conjugate gradient algorithm which exploits the inherent element-level parallelism of a finite element model has been proposed by Law[8].

Further improvements in the performance of the conjugate gradient algorithm can be achieved through preconditioning. Preconditioning consists of transforming the stiffness matrix with an approximation of its inverse. This approximation can be as simple as a diagonal matrix[9], or much more sophisticated, such as the element-by-element preconditioner proposed by Hughes.[10]

The element by element conjugate gradient algorithm has proven to be relatively efficient in taking advantage of a parallel computing environment. However, its cost effectiveness is highly problem dependent. For finite element problems which generate a stiffness matrix with a large mean bandwidth, the EBE-PCG is the method of choice. For problems with low mean bandwidths, or involving multiple load cases it was found that the EBE-PCG cannot match the performance of the LL^T decomposition algorithm[11].

Thus, the SAPNEW program can use either a parallelized LL^T algorithm or the EBE-PCG algorithm to solve the linear systems that it generates. However, for blade models (which are generally very ill-conditioned) the EBE-PCG method may fail due to machine round-off, and it is recommended that the decomposition algorithm be used.

Eigen Analysis

To calculate the eigenvalues and eigenvectors, SAPNEW uses the subspace iteration procedure. This procedure involves projecting the stiffness and mass matrices on a desired subspace. This process is, in fact, parallelizable over the dimension of the subspace. SAPNEW calculates the projected mass and stiffness matrices in parallel.

4. EVALUATION OF SAPNEW

Validation

To check the accuracy of the SAPNEW program, several static and dynamic analyses of rectangular plates were carried out for various aspect ratios and mesh-sizes.

Descriptions of models are listed in Table 1. The results of the static analysis are listed in Table 2. The results of the dynamic analysis are listed in Table 3. Finally, the results of the dynamic restart analysis are listed in Table 4.

Table 1. Description of models

Model no	1	2	3	4	5	6	7	8
Aspect ratio (b/a)	1.0	1.0	1.0	1.0	1.4	1.4	1.4	1.4
Mesh size	10x10	20x20	30x30	50x50	10x10	20x20	30x30	50x50
Total D.O.F	287	1167	2649	7409	287	1167	2649	7409
Mean bandwidth	30	61	96	156	30	61	96	156

Notes: boundary condition : simple supports on all four sides
plate length : a = 20.0 m
bending rigidity : 0.08333 N-m
mass density : 0.0001 kg
loading type
- Concentrated load applied at mid-point of plate. (F = 1.0 N)
- Uniform pressure load (p = 0.1 N/m^2)

Table 2. The results of static analysis

Aspect ratio of shell element	Loading type	Mesh size	Maximum deflection (mm)	theory (mm)	relative error (%)
1.0	F	10x10	55.007	55.903	1.60
		20x20	55.484		0.74
		30x30	55.623		0.50
		50x50	55.847		0.10
	p	10x10	764.31	782.65	2.34
		20x20	776.04		0.84
		30x30	779.51		0.41
		50x50	781.08		0.11
1.4	F	10x10	70.329	71.518	1.66
		20x20	71.050		0.65
		30x30	71.303		0.31
		50x50	71.374		0.20
	p	10x10	1333.4	1359.04	1.88
		20x20	1353.5		0.41
		30x30	1361.1		0.15
		50x50	1358.9		0.10

Notes: F : concentrated load at the mid-point of plate
p : uniform pressure load

Table 3. The results of the dynamic analysis

Model no.		Frequencies of modes (Hz)						
		1	2	3	4	5	6	7
1	C	4.5717	11.331	11.331	18.216	22.776	22.776	29.777
	T	4.5048	11.262	11.262	18.019	22.524	22.524	29.281
	E	1.5	0.6	0.6	1.1	1.1	1.1	1.7
2	C	4.5079	11.279	11.279	18.069	22.587	22.587	29.406
	T	4.5048	11.262	11.262	18.019	22.524	22.524	29.281
	E	0.06	0.15	0.15	0.28	0.27	0.27	0.4
3	C	4.5061	11.269	11.269	18.041	22.551	22.551	29.336
	T	4.5048	11.262	11.262	18.019	22.524	22.524	29.281
	E	0.02	0.06	0.06	0.12	0.1	0.1	0.18
4	C	4.5053	11.264	11.264	18.027	22.534	22.534	29.301
	T	4.5048	11.262	11.262	18.019	22.524	22.524	29.281
	E	0.01	0.02	0.02	0.04	0.04	0.04	0.68
5	C	3.4594	6.9313	10.291	13.208	19.564	20.845	27.752
	T	3.4016	6.8492	10.159	13.065	19.352	20.639	27.396
	E	1.7	1.2	1.3	1.1	1.1	1.0	1.3
6	C	3.4458	6.9176	10.230	13.143	19.352	20.701	27.451
	T	3.4016	6.8492	10.159	13.065	19.352	20.639	27.396
	E	1.3	1.0	0.7	0.6	0.8	0.3	0.2
7	C	3.4390	6.9245	10.230	13.104	19.448	20.680	27.451
	T	3.4016	6.8492	10.159	13.065	19.352	20.639	27.396
	E	1.1	1.1	0.7	0.3	0.5	0.2	0.2
8	C	3.4322	6.8971	10.169	13.130	19.390	20.680	27.478
	T	3.4016	6.8492	10.159	13.065	19.352	20.639	27.396
	E	0.9	0.7	0.1	0.5	0.2	0.2	0.3

Notes: C : calculated value
T : theoretical value (from reference [12])
E : relative error (%)

Test models

The models used for evaluating the SAPNEW program were typical propfan blades: SR5 and SR7L. The NTOS conversion program was used to convert a NASTRAN models of these blades to the SAPNEW data input format.

Figure 2. shows the geometry of the SR5 blade. Table 4. lists the statistics for this blade model. The SR5 test case consisted of determining the three lowest eigenvalues and their corresponding mode shapes using geometric stiffness generated by the static solution of the blade loaded by centrifugal forces. The SR5 blade model constructed using homegeneous and isotropic material properties.

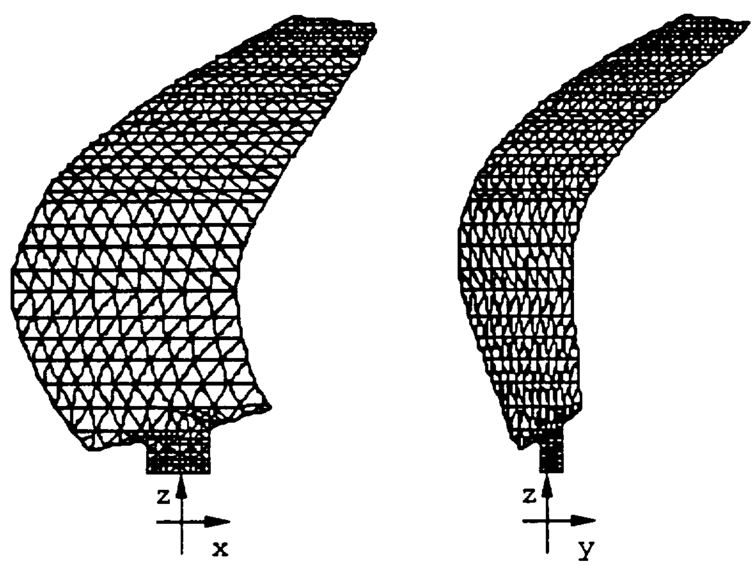


Figure 2. SR5 blade geometry.

Table 4. SR5 blade model statistics.

General:	
Types of elements	Triangular Thin Shell
Number of elements	702
Number of nodes	402
Number of degrees of freedom	2360
Stiffness Matrix:	
Number of working elements	321117
Maximum half-bandwidth	2008
Mean half-bandwidth	136

Figure 3. shows the geometry of the SR7L blade. Table 5. lists the statistics for this blade model. The SR7L test case consisted of determining the six lowest eigenvalues and their corresponding mode shapes using geometric stiffness generated by the static solution of the blade loaded by centrifugal forces. The SR7L blade model was constructed using material properties derived from classical plate analysis of laminated composite structures.

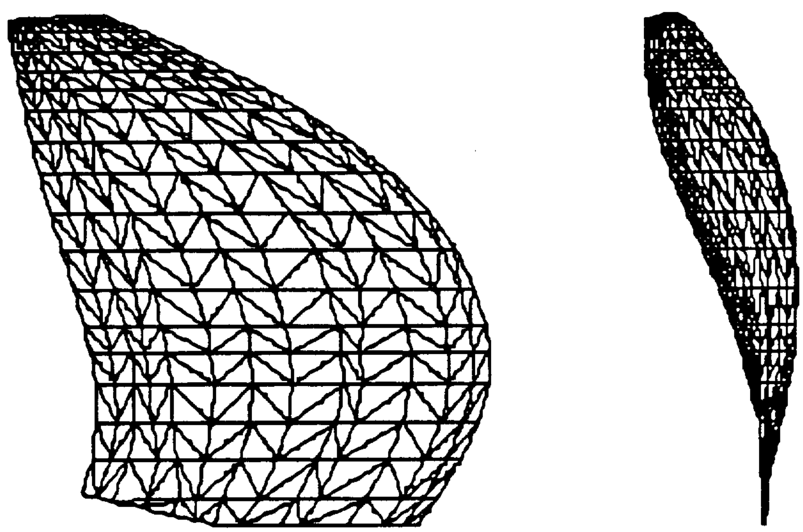


Figure 3. SR7L blade geometry.

Table 5. SR7L blade model statistics.

General:	
Types of elements	Triangular Thin Shell
Number of elements	449
Number of nodes	267
Number of degrees of freedom	1550
Stiffness Matrix:	
Number of working elements	208793
Maximum half-bandwidth	1474
Mean half-bandwidth	134

Results

The calculated natural frequencies for both blade models are given in Table 6. This table also presents the frequencies calculated by MSC/NASTRAN for comparison. The lowest mode frequency discrepancy between SAPNEW and MSC/NASTRAN is due to differences in the manner in which geometric stiffening is accounted for. For the geometric stiffness calculations, NASTRAN uses the same interpolation functions for normal displacements as were used in the bending stiffness calculations. SAPNEW uses a linear interpolation for the normal displacement. Although this is a lower order of approximation than that used for the element stiffness matrix, this is consistent in an energy sense.

Table 6. Blade model results.

(a.) SR5

Mode	Frequency (Hz)		
	SAPNEW	MSC/NASTRAN	Relative error (%)
1	174.60	151.32	15.38
2	287.41	281.11	2.24
3	563.16	586.33	-3.95

(b.) SR7L

Mode	Frequency (Hz)		
	SAPNEW	MSC/NASTRAN	Relative error (%)
1	51.34	43.52	17.98
2	90.50	94.40	-4.14
3	105.91	108.50	-2.39
4	149.82	147.08	1.87
5	175.52	182.47	-3.80
6	245.05	231.25	5.97

The times required by the SAPNEW program to run the test cases on the Alliant FX/80 for different code optimization options are given in Table 7. The corresponding speed-up values are listed in Table 8. and presented in Figure 4.

Table 7. Time results (All times in sec.).

	Number of Processors					
	1	2	3	4	5	6
Without Vectorization						
SR5	190.27	106.45	78.22	73.67	72.09	53.55
SR7L	233.44	124.73	88.56	71.92	70.21	54.69
With Vectorization						
SR5	105.26	63.31	50.31	47.24	46.28	41.05
SR7L	105.45	61.09	47.25	41.56	41.12	38.58

Table 8. Speedup results.

	Number of processors					
	1	2	3	4	5	6
Eigen Analysis only						
SR5	1.00	1.84	2.44	2.55	2.52	3.12
SR7L	1.00	1.89	2.59	3.04	3.01	3.31
Total Problem Run						
SR5	1.00	1.66	2.09	2.23	2.27	2.56
SR7L	1.00	1.73	2.23	2.54	2.56	2.73

Note : Total problem run includes: input, element formulation, static analysis, eigen analysis, and output.

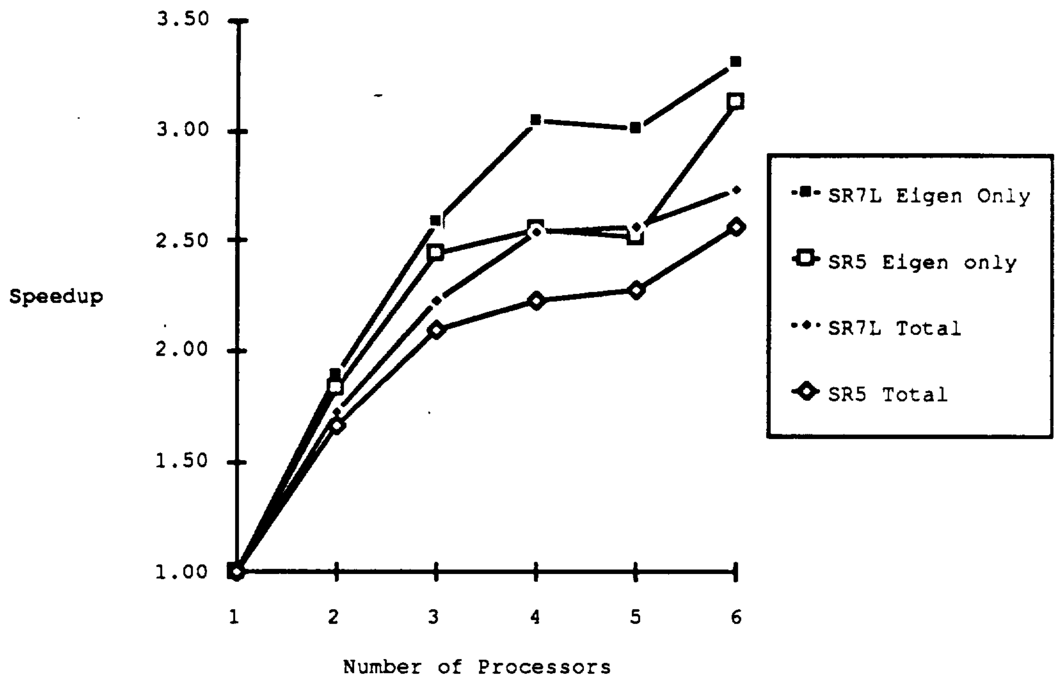


Figure 4. Speedup results.

The dips in the curves for the eigen analysis speedup are cause by the fact the there are six tasks for the SR5 test model and twelve tasks for the SR7L test model which are performed concurrently. The number of tasks is related to the number of modes to be found.

5. REFERENCES

1. Kaza, K.R.V. and Kielb, R.E. , " Flutter and Response of Mistuned Cascade in Incompressible Flow", *AIAA Journal*, vol. **20**, no.2, pp.1120-1127, 1982.
2. Bathe, K.J., Wilson, E.L., and Peterson, F.E., "SAP IV-- A Structural Analysis Program for Static and Dynamic Analysis of Linear Structural Systems", *EERC Report No. 73-11*, College of Engineering, University of California, Berkeley, June 1973.
3. Bathe, K.J., Ozdemir, H., and Wilson, E.L., "Static and Dynamic Geometric and Material Nonlinear Analysis", *SESM Report No. 74-4*, College of Engineering, University of California, Berkeley, February 1974.
4. Bathe, K.J., *Finite element procedures in engineering analysis,* Prentice-Hall company, pp.672-684, 1982.
5. Narayanan, G.V. and Kaza, K.R.V., "ASTROP2 Users' Manual: A Program for Aeroelastic Stability Analysis of Propfans", NASA TM 4304, August 1991.
6. Clough R.W. and Felippa, C.A.," A Refined Quadrilateral element for analysis of plate bending", *Proceedings 2'nd conference on matrix methods in structural mechanics,* Wright Peterson AFB, Ohio, 1968.
7. Hestenes, M.R., *Conjugate Direction Methods in Optimization,* Springer-Verlag, pp. 116-125, 1980
8. Law, K.H.,"Parallel Finite Element Solution Method", *Computer & Structures* vol.**23**, No.6, pp845-858, 1986.
9. Argyris, J. et al., "Τα Παντα Ρει", *Computer Methods in Applied Mechanics and Engineering,* vol.**51**, pp.289-365, 1985.
10. Hughes, T.J.R., et al. , "New Alternating Direction procedures in Finite Element Analysis based upon eBe Approximate Factorizations", *Computer Methods for Nonlinear solids and Structural Mechanics,* AMD vol.**54**, pp.75-109, 1983.
11. Watson, B.C. and Kamat, M.P., "A Study of Equation Solvers for Linear and Non-Linear Finite Element Analysis on Parallel Processing Computers", *Proceedings 33rd AIAA/ASME/ ASCE/AHS SDM Conference, Paper # 92-2478, Dallas, TX, April 1992, .*
12. Timoshenko S.P., Woinowsky-Krieger S. ,*Theory of plates and shells,* McGraw-Hill company, pp.140-158, 1959.

APPENDIX I. USER'S GUIDE FOR SAPNEW

File names

Executable file

The executable file is located on the Alliant FX/80 at NASA Lewis Research Center. The program name is **sapnew**. The program synopsis is as follows:

```
$ sapnew [-e|c|n] infln
```

The input file should be named *infln*.dat where *infln* is a user chosen file name prefix. The program will write its output into a file named *infln*.out .

-e This option will cause the program to use the element-by-element conjugate gradient algorithm to solve the linear system for static analysis. If the data file specifies dynamic analysis, this option has no effect. If the model has multi-point constraints, this option should not be used.

-c This option will cause the program to use the conjugate gradient algorithm on the assembled stiffness matrix to solve the linear system for static analysis. If the data file specifies dynamic analysis, this option has no effect.

-n This option causes the program to generate a data file for the ASTROP aeroelastic analysis program. This data will be written to a file named *infln*.nasty . If the input data specifies static analysis, this flag has no effect.

Source files

The source files are written in Alliant's FX/Fortran. This is an extension of Fortran/77 with directives to specify parallelization and vectorization. These directives appear as comments to standard Fortran. They are located on the Alliant FX/80 together with an associated Makefile. A short description of each module follows:

sapmain.f : main program code.
sapsubs.f : general subroutines.
saprecur.f : code to generate the shell element stiffness and mass matrices.
sapsolv.f : code for Cholesky decomposition of stiffness matrix
sapdyn.f : code for eigen analysis
sapecgm.f : code for element-by-element conjugate gradient algorithm
sapcgm.f : code for general conjugate gradient algorithm

Auxiliary files

Auxiliary files may be created by the program (at the user's option) for the possibility of restarting a dynamic analysis to calculate more eigen values/vectors.

modal.inf :	storage of modal information
stif.inf :	storage of assembled stiffness matrix
mass.inf :	storage of assembled mass matrix and the LM array

Sample data files

Sample data files for static and modal analysis of propfan blades (SR5 and SR7L) are available on the Alliant FX/80.

sr5.dat :	static analysis of an isotropic blade with centrifugal load
sr5dyn2.dat:	modal analysis of an isotropic blade with geometric stiffening due to centrifugal load.
sr7l.dat:	static analysis of a composite blade with centrifugal load. This model uses beam and spring elements to simulate an elastic support.
sr7ldyn2.dat:	modal analysis of a composite blade with geometric stiffening due to centrifugal load.

Input data file format

Static analysis

Title card	(section 1)
Control information card	(section 2)
Node information cards	(section 4)
Concentrated load information cards	(section 5)
Element information cards	(section 7)
Centrifugal load information cards	(section 8)
Load factor cards	(section 9)

Modal analysis

***Without** geometric stiffening*

Title card	(section 1)
Control information card	(section 2)
Dynamic control information card	(section 3)
Node information cards	(section 4)
Concentrated mass information cards	(section 6)
Element information cards	(section 7)

***With** geometric stiffening*

Title card	(section 1)
Control information card	(section 2)
Dynamic control information card	(section 3)
Node information cards	(section 4)
Concentrated load information cards	(section 5)
Element information cards	(section 7)
Centrifugal load information cards	(section 8)

Restarting the eigen value / vector analysis

Title card	(section 1)
Control information card	(section 2)
Dynamic control information card	(section 3)

1. Title card

Format	Description
A80	Title of analysis

2. Control information card

Format	Description
I5	Analysis code 0; Static analysis >0; Eigen analysis Analysis code - 1 = number of static solution iterations for geometric stiffness computation (E.g. Analysis code = 1 means eigen analysis with no geometric stiffening effect accounted for. Analysis code = 2 means eigen analysis with one static analysis to compute geometric sitffness matrices. Analysis code = 3 means eigen analysis with two static analysis iterations to compute geometric stiffness matrices etc.)
I5	Number of node points
I5	Number of element groups
I5	Number of load cases or modes Analysis code = 0; Load cases (not ineluding centrifugal load) Analysis code >0; Modes
I5	Flag for execution mode 0; Execute 1; Input data verification
I5	Flag for centrifugal load 0; No centrifugal loads 1; Use centrifugal loads

Note: If analysis code > 1 and centrifugal loading is not used, then one load case (with concentrated loads) is expected.

3. Dynamic control information card

Format	Description
F10.0	Cut-off frequency
	Default = 1.0×10^9
F10.0	Error tolerance in the subspace iteration procedure
	Default = 1.0×10^{-6}
I5	Maximum number of iterations
	Default = 16
I5	Flag for shifting
	0 ; Do not use shifting 1 ; Use shifting
F10.0	Shifting factor
I5	Flag for Sturm sequence check
	0 ; Do not check 1 ; Check
I5	Flag for printing the iteration procedure
	0 ; Do not print 1 ; Print
I5	Flag for restart execution
	0 ; Initial execution -1 ; Restart execution
I5	Flag for saving modal parameters
	0 ; Do not save 1 ; Save for the later usage

Notes: 1. Normally, the lowest eigenvalues are computed. Shifting can be used to find the closest eigenvalues to the specified shifting factor.
2. The Sturm sequence check can be used to insure that the desired eigenvalues were in fact the ones that were found.

4. Node information cards

Node information cards (one for each node)

Format	Description
I5	Node number
6I5	Boundary condition code for X, Y, Z, RX, RY, RZ directions 0; Free 1; Fixed >1; Constrained by Multi-Point-Constraint
F10.0	X-coordinate
F10.0	Y-coordinate
F10.0	Z-coordinate
I5	Node generation code

Note: Node generation may be used if some nodes are evenly spaced along some line segment. The node generation code is the increment in node number to be used for the generated nodes. For example, these input cards:

```
    8    0    0    0    0    0    0       0.0       0.0       0.0    2
   18    0    0    1    1    1    1      20.0       0.0      25.0    0
```

would generate the following nodes:

```
    8    0    0    0    0    0    0       0.0       0.0       0.0
   10    0    0    0    0    0    0       4.0       0.0       5.0
   12    0    0    0    0    0    0       8.0       0.0      10.0
   14    0    0    0    0    0    0      12.0       0.0      15.0
   16    0    0    0    0    0    0      16.0       0.0      20.0
   18    0    0    1    1    1    1      20.0       0.0      25.0
```

Note that the node number increment (Node generation code) is specified on the first card of this input pair.

Following all node information cards:

- *Multi-point constraint information cards (one for each multi-point constrained DOF)*

Format	Description		
I5	Node 1	}	DOF 1
I5	Direction 1=X, 2=Y, ..., 6=RZ		
F10.0	Coefficient 1	}	TR 1
I5	Node 2	}	DOF 2
I5	Direction 1=X, 2=Y, ..., 6=RZ		
F10.0	Coefficient 2	}	TR 2
I5	Node 3	}	DOF 3
I5	Direction 1=X, 2=Y, ..., 6=RZ		
F10.0	Coefficient 3	}	TR 3
I5	Node 4	}	DOF 4
I5	Direction 1=X, 2=Y, ..., 6=RZ		
F10.0	Coefficient 4	}	TR 4

Note: The constraint is formed as:

Constrained DOF = TR1*DOF1 + TR2*DOF2 + TR3*DOF3 + TR4*DOF4

5. Concentrated load information cards

(one set for each load case)

Load control card

Format	Description
I5	Load case number
I5	Number of loads in this load case

Concentrated load cards (one for each load)

Format	Description
I5	Node number at which the load is applied
I5	Code for the direction of the applied load
	1=X, 2=Y, ..., 6=RZ
F10.0	Magnitude of the applied load

6. Concentrated mass information cards

(one for each concentrated mass)

Format	Description
I5	Node number
F10.0	Mass in the x-dir.
F10.0	Mass in the y-dir.
F10.0	Mass in the z-dir.
F10.0	Inertia in the rx-dir.
F10.0	Inertia in the ry-dir.
F10.0	Inertia in the rz-dir.

Note: A blank card signals the end of the concentrated mass input data. Thus, even for no concentrated masses, a blank card must be present (for dynamic analysis).

7. Element information cards

Shell element control card

Format	Description
I5	Code for the element type
	1 ; shell element
I5	Number of shell elements
I5	Number of shell material property sets

Shell material property cards (a pair of cards for each shell material property set)

Format	Description
I5	Material property number
20X	
F10.0	Mass density
F10.0	Thermal expansion coefficient in the x-dir.
F10.0	Thermal expansion coefficient in the y-dir.
F10.0	Thermal expansion coefficient in the z-dir.

Format	Description
F10.0	C_{11} of the material coefficient matrix $[C_{ij}]$
F10.0	C_{12} of the material coefficient matrix $[C_{ij}]$
F10.0	C_{13} of the material coefficient matrix $[C_{ij}]$
F10.0	C_{22} of the material coefficient matrix $[C_{ij}]$
F10.0	C_{23} of the material coefficient matrix $[C_{ij}]$
F10.0	C_{33} of the material coefficient matrix $[C_{ij}]$

Note: The material coefficient matrix $[C_{ij}]$ should be as follows:

For isotropic materials:

Plane stress:

$$[C_{ij}] = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}$$

Plane strain:

$$[C_{ij}] = \frac{E}{(1+\nu)(1-2\nu)}\begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}$$

For orthotropic materials:

Plane stress:

$$[C_{ij}] = \frac{E_y}{1-n\nu_y{}^2}\begin{bmatrix} n & n\nu_y & 0 \\ n\nu_y & 1 & 0 \\ 0 & 0 & m(1-\nu_y{}^2) \end{bmatrix}$$

Plane strain:

$$[C_{ij}] = \frac{E_y}{(1+n\nu_y)(1-2n\nu_y)}\begin{bmatrix} 1-n\nu_y & n\nu_y & 0 \\ n\nu_y & 1-n\nu_y & 0 \\ 0 & 0 & \frac{m(1-n\nu_y)}{2} \end{bmatrix}$$

where
E : Young's modulus
G : shear modulus
ν : Poisson's ratio
n : E_x / E_y
m : G_x / G_y

Shell element load multiplier cards (5 cards)

Format	Description
4F10.0	pressure load multiplier factors

Format	description
4F10.0	thermal load multiplier factors

Format	description
4F10.0	x-acceleration multiplier factors

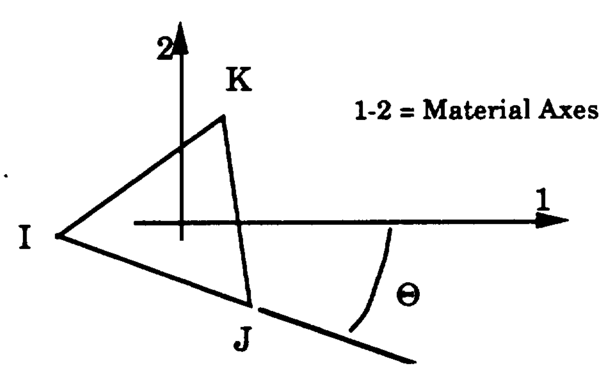
Format	description
4F10.0	y-acceleration multiplier factors

Format	description
4F10.0	z-acceleration multiplier factors

Note: The four multipliers for these loads form four different loading conditions. Within each loading condition, these values determine the relative amount of each load type (e.g. pressure to thermal loading). For each problem load case, these four loading conditions will be scaled (through a load factor card [section 9]) and superposed and then added to the load vector.

Shell element description card (one card for each shell element)

Format	Description
I5	Shell element number
I5	Node I
I5	Node J
I5	Node K
I5	Node L
I5	Mid-point node
I5	In-plane material property number
I5	Bending material property number
I5	Element generation code (See note 5. on next page)
F7.0	Thickness of the element
F7.0	Transverse pressure on the element
F7.0	Temperature of the element
F7.0	Temperature gradient accross the thickness of the element
F7.0	Theta (See Figure below)



Notes: 1. The elements must be consecutively numbered, and input in order.

2. If the element is triangular, node L and the mid-point node should be zero.

3. If the element is quadrilateral and the behavior at the mid-point needs to be known, the mid-point node should be specified. Otherwise, set this node to zero.

4. If the material is isotropic or the element is quadrilateral, then theta should be greater than 180.

5. Different in-plane and bending material properties are allowed so that laminated composite materials may be simulated. (This is similar to NASTRAN. However, unlike NASTRAN, this shell element does not include the transverse shear deformation.)

6. Automatic element geneneration can be used if the relative node numbers for some elements remain constant.

For example, the following input cards:

16	1	3	4	2	0	1	1	0	0.1	0.0	0.0	0.0	200.0
20	9	11	12	10	0	1	1	2	0.1	0.0	0.0	0.0	200.0

would generate the following elements:

16	1	3	4	2	0	1	1		0.1	0.0	0.0	0.0	200.0
17	3	5	6	4	0	1	1		0.1	0.0	0.0	0.0	200.0
18	5	7	8	6	0	1	1		0.1	0.0	0.0	0.0	200.0
19	7	9	10	8	0	1	1		0.1	0.0	0.0	0.0	200.0
20	9	11	12	10	0	1	1		0.1	0.0	0.0	0.0	200.0

Note that the node increment (element generation code) is specified on the second card in this pair.

Beam element control card

Format	Description
I5	Code for the element type
	2 ; beam element
I5	Number of beam elements
I5	Number of beam geometric property sets
I5	Number of beam fixed-end force sets
I5	Number of beam material property sets

Beam material property cards (one card for each beam material property set)

Format	Description
I5	Beam material property set number
F10.0	Young's modulus
F10.0	Poisson's ratio
F10.0	Mass density
F10.0	Weight per unit length

Beam geometric property cards (one card for each beam geometric property set)

Format	Description
I5	Geometric property set number
F10.0	Axial cross section area
F10.0	Cross section area for shear 1
F10.0	Cross section area for shear 2
F10.0	Torsion coefficient 'J'
F10.0	Second area moment for axis 1
F10.0	Second area moment for axis 2

Beam element load multiplier cards (3 cards)

Format	Description
4F10.0	x-acceleration load multiplier

Format	Description
4F10.0	y-acceleration load multiplier

Format	Description
4F10.0	z-acceleration load multiplier

Beam fixed end force cards (a pair of cards for each fixed-end force set)

<u>Format</u>

I5

6F10.0

<u>Format</u>

F15.0

5F10.0

Beam element description cards (one card for each beam element)

Format	Description
I5	Element number
I5	Node I
I5	Node J
I5	Node K
I5	Material property set number
I5	Geometric property set number
4I5	End loads
I6	End code for node I
I6	End code for node J

Note: The beam axis connects nodes I & J. The vector from node I to node K detemines the cross section axis 1

Spring element control card

Format	Description
I5	Code for the element type
	3 ; spring element
I5	Number of spring elements

Spring element data card (one for each element)

Format	Desription
I5	Node I
I5	Node J
I5	Direction code
	1=X, 2=Y, ..., 6=RZ
F10.0	Spring stiffness

8. Centrifugal load information card (only if centrifugal loading is used)

Format	Description
F10.0	X-component of spin axis vector
F10.0	Y-component of spin axis vector
F10.0	Z-component of spin axis vector
F10.0	Spin rate in radians/second
F10.0	Unit conversion factor

Note: Spin axis passes through coordinate system origin.

9. Load factor card (one for each load case (**not** centrifugal loading))

Format	Description
4F10.0	Element load factors

APPENDIX II. NTOS - A CONVERSION UTILITY

To make SAPNEW more convenient to use, a conversion utility named NTOS (Nastran TO Sapnew) was written. This utility changes the format of a NASTRAN input data deck to that used by SAPNEW. NTOS is located on the Alliant FX/80 at NASA Lewis Research Center. The procedure for using NTOS on the Alliant is as follows:

```
$ ntos <nasdatafile  >sapdatafile
```

where:

nasdatafile = NASTRAN input data filename

sapdatafile = SAPNEW input filename (must end in .dat)

The NTOS program only converts the BULK DATA section of the NASTRAN input data file. The user must manually edit the resulting SAPNEW file to include control information. (For example, the title card.) Following is a list of the NASTRAN bulk data cards which NTOS processes:

CBAR
CELAS1
CTRIA3
GRID
MAT1
MAT2
PBAR
PELAS
PSHELL

Any other cards in the bulk data deck will be ignored by NTOS. Thus the user must manually convert any other options. In particular, the user must manually add data cards for multi-point constraints, for centrifugal forces, and for any load cases that are desired.

The user must adjust the output of NTOS for either static or dynamic analysis. If dynamic analysis is desired, the dynamic control card must be entered manually (insert a blank line to accept control defaults).