



DEPARTMENT OF MECHANICAL ENGINEERING AND MECHANICS
SCHOOL OF ENGINEERING
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA 23508

(NASA-CR-176707) DEVELOPMENT OF COMPUTER PROGRAM NAS3D USING VECTOR PROCESSING FOR GEOMETRIC NONLINEAR ANALYSIS OF STRUCTURES
Final Report, 20 Sep. - 31 Dec. 1985 (Old Dominion Univ.) 44 p HC A03/EF A01 CSCL 09B G3/61 09000 N86-23320 Unclas

DEVELOPMENT OF COMPUTER PROGRAM NAS3D
USING VECTOR PROCESSING FOR
GEOMETRIC NONLINEAR ANALYSIS OF STRUCTURES

By

P. D. Mangaliri, Research Associate

R. Prabhakaran, Principal Investigator

Final Report

For the period September 20, 1985 to December 31, 1985

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Under
Master Contract NAS-1-17993
Task Authorization No. 20
Dr. Wolf Elber, Technical Monitor
MD-Fatigue & Fracture Branch



January 1986

DEPARTMENT OF MECHANICAL ENGINEERING AND MECHANICS
SCHOOL OF ENGINEERING
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA 23508

DEVELOPMENT OF COMPUTER PROGRAM NAS3D
USING VECTOR PROCESSING FOR
GEOMETRIC NONLINEAR ANALYSIS OF STRUCTURES

By

P. D. Mangalgi, Research Associate

R. Prabhakaran, Principal Investigator

Final Report

For the period September 20, 1985 to December 31, 1985

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Under
Master Contract NAS-1-17993
Task Authorization No. 20
Dr. Wolf Elber, Technical Monitor
MD-Fatigue & Fracture Branch

Submitted by the
Old Dominion University Research Foundation
P. O. Box 6369
Norfolk, Virginia 23508

January 1986

DEVELOPMENT OF COMPUTER PROGRAM NAS3D USING VECTOR PROCESSING
FOR GEOMETRIC NONLINEAR ANALYSIS OF STRUCTURES

By

P. D. Mangalgiri¹

Principal Investigator: Ramamurthy Prabhakaran²

ABSTRACT

An algorithm for vectorized computation of stiffness matrices of an 8-noded isoparametric hexahedron element for geometric nonlinear analysis was developed. This was used in conjunction with the earlier 2-D program GAMNAS to develop the new program NAS3D for geometric nonlinear analysis. A conventional, modified Newton-Raphson process is used for the nonlinear analysis. New schemes for the computation of stiffness and strain energy release rates is presented. The organization of the program is explained and some results on four sample problems are given. The study of CPU times showed that savings by a factor of 11-13 were achieved when vectorized computation was used for the stiffness instead of the conventional scalar one. Finally, the scheme of inputting data is explained in the Appendix.

¹Research Associate, Department of Mechanical Engineering and Mechanics, Old Dominion University, Norfolk, Virginia 23508.

²Professor, Department of Mechanical Engineering and Mechanics, Old Dominion University, Norfolk, Virginia 23508.

TABLE OF CONTENTS

	<u>PAGE</u>
ABSTRACT	ii
1. INTRODUCTION.....	1
2. THEORY.....	3
2.1 Formulation.....	3
2.2 Computation of Stiffness.....	6
2.3 Computation of Strain Energy Release Rates.....	14
3. DESCRIPTION AND ORGANIZATION OF PROGRAM NAS3D.....	15
4. SAMPLE PROBLEMS.....	17
4.1 Slender Beam.....	17
4.2 Double Cantilever Beam (DCB).....	20
4.3 Double Cracked Lap Shear Specimen.....	24
4.4 Cracked Lap Shear.....	26
5. CONCLUDING REMARKS.....	26
ACKNOWLEDGEMENT.....	29
REFERENCES.....	30
APPENDIX: INPUT DATA.....	31

LIST OF TABLES

<u>TABLE</u>	<u>PAGE</u>
1 CPU Times for Various Stages in the Nonlinear Analysis.....	18
2 Example of Slender Beam Axial stress in the Beam at Various Deflections.....	21
3 Example of Double Cantilever Beam. Comparison of Full and Reduced Integration Results for $P = 40$ B/in.....	23

LIST OF FIGURES

<u>FIGURE</u>	<u>PAGE</u>
1 Structure of arrays BV and BLV.....	10
2 Structure of vector SV and vector SUM replicated from SV.....	11

TABLE OF CONTENTS - Concluded

LIST OF FIGURES - Concluded

<u>FIGURE</u>	<u>PAGE</u>
3 Storage organization of the element stiffness matrix $[K^{(e)}]$..	12
4 Flowchart for main program.....	16
5 Example of slender beam. (a) configuration and mesh (b) nonlinear variation of stress with applied displacement.....	19
6 Example of double cantilever beam. (a) configuration and mesh of quarter beam (b) distribution of G along the crack front.....	22
7 Example of DCLS specimen (a) configuration and (b) FE mesh in y-z plane. The mesh division is repeated on each plane parallel to this plane to get 3-D mesh.....	25
8 Example of DCLS specimen. Total and individual strain energy release rates along the crack front in the linear solution.....	27
9 Example of CLS specimen. (a) configuration (b) G distributions along the crack front.....	28

1. INTRODUCTION

Many problems in the study of the behavior of composites require geometric nonlinear analyses due to large strains and displacements involved. Examples of these are the behavior of laminated composites under compression (buckling), with or without delamination, and deformation of lap shear geometries. Often, the two-dimensional (2-D) analysis is used successfully to study stress states and estimate fracture mechanics parameters in these problems. However, such an analysis proves inadequate in many cases where there exist complexities in the composite lay-up, geometry or form of delamination. A full three-dimensional (3-D) analysis is then required. Such an analysis, and particularly the nonlinear one, involves extensive computations and is normally prohibitively time-consuming with conventional computers and algorithms. The advent of vector processor computers, like the VPS-32 system at NASA Langley, has provided an economically feasible solution to the time-consuming operations of 3-D analyses. However, to fully utilize the fast processing capability of these computers in the vector form, new algorithms are required. These algorithms should be aimed specifically at taking advantage of vector processing by carrying out computations by building long vectors.

The two important time-consuming stages in the Finite Element (FE) analysis of structures are the computation of stiffness matrix and solution of resulting system of equations. The development of solution routines has received considerable attention in the past and already efficient vectorized solution routines are in operation on the VPS-32 system (Refs. 1, 2). Computation of stiffness in the vector form was studied by Noor and Hartley (Ref. 3) who developed an algorithm which builds and uses long vectors. Savings by a factor of 5 to 6 were obtained in CPU time when routines based

on these algorithms were used. Stiffness calculations based on these algorithms were effectively used in linear 3-D analysis by Raju et al. (Ref. 4). However, this has not been applied in the case of nonlinear analysis so far. Since the nonlinear stiffness calculation involves many more computations than the linear one, and since the nonlinear solution process requires repeated calculation of stiffnesses, the vectorization in this segment should lead to decreasing computation time. The present effort is directed towards this goal.

Development of a capability for general nonlinear analysis of structures with cracks or discontinuities is a complex, but worthwhile exercise. The present work is a step in this exercise and is basically aimed at analyzing laminated composites with delaminations under compression and lap shear geometries which involve large rotations during deformation. A good review of many efforts to estimate stresses and fracture mechanics parameters in these problems can be found in references 5-7. Almost all of this work except Lof's (Ref. 7) is based on 2-D FE analysis with plane stress or plane strain. It is now generally recognized that cracked lap-shear geometries have large rotations and require a geometric nonlinear analysis to get accurate estimates of strain energy release rate in various modes of fracture [6]. A 2-D nonlinear analysis was developed in an earlier effort at NASA Langley (Ref. 6) and resulted in the program GAMNAS for the case of plane stress and plain strain idealization (Ref. 8). Subsequently, a quasi-3-D linear analysis was developed for the case of a symmetric double cracked lap shear specimen (Ref. 9). This proved to be a good compromise between the accuracy of the results and the cost of the solution for the 3-D nature of the problem. However, this analysis involved superpositions and thus could not be extended to the case of nonlinear behavior.

In the present work, the program GAMNAS and the vectorized algorithm for linear stiffness computation used in earlier 3-D linear analysis are taken as the starting points. Routines are developed for vectorized computation of nonlinear stiffness using 8-noded isoparametric hexahedron element. These are used to develop a code for the 3-D geometric nonlinear analysis. The overall flow of the program follows closely that of GAMNAS. Some theoretical aspects which served as the basis for the development of the present program are presented. First, the general formulation of the nonlinear analysis is presented. Next, schemes of computation of stiffness and of extracting strain energy release rates are discussed. Then, the description and organization of the program NAS3D is given along with the study of computational times involved in various stages. Finally, the results of some simple problems are presented as test cases and examples. The details of input data required are given in the Appendix.

2. THEORY

A displacement based FE formulation is used in the present work. Such a formulation of a geometric nonlinear problem is given in Ref. 10. The salient features are given in this section. Further, schemes for computation of stiffness and strain energy release rates are also discussed.

2.1 Formulation

(i) Strain-Displacement Relations

A general definition of strains using Green's strain tensor is used to take into account the large displacements and rotations. This defines the strains as follows:

$$\epsilon_x = \frac{\partial u}{\partial x} + \frac{1}{2} \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial w}{\partial z} \right)^2 \right] \quad (1)$$

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} + \frac{\partial u}{\partial x} \cdot \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \cdot \frac{\partial v}{\partial y} + \frac{\partial w}{\partial x} \cdot \frac{\partial w}{\partial y}$$

other components being obtained by suitable permutation of subscripts. It may be noted that if the displacements are small, the general first order strain approximation is obtained by neglecting the quadratic terms.

(ii) Stress-Strain Relations

For small strains the stress-strain relations can be written in general form as

$$\{\sigma\} = [D] (\{\epsilon\} - \{\epsilon\}_0) + \{\sigma\}_0 \quad (2)$$

in which [D] is the usual set of the elastic constants and the subscript 0 refers to initial values. When the strains are large, a nonlinear stress-strain relationship for the material has to be used. This is not considered in the present formulation.

(ii) Equilibrium Equations

Equilibrium conditions have to be satisfied between internal and external generalized forces. If the displacements are prescribed in the usual manner of an FE analysis by a finite number of nodal parameters {a}, the necessary equilibrium conditions can be obtained by using principle of virtual work as when { } represents the sum of internal and external generalized forces and [B] is defined from the strain-displacement relations of Eq. (1) as

$$\{\psi(a)\} = \int_V [B]^T \{\sigma\} dV - \{f\} = 0. \quad (3)$$

Since the strains depend nonlinearly on displacements, the matrix B is now dependent on {a}. Also, the quadratic nature of Eq. (1) implies that [B] is linear with {a}. The stresses {σ} are linearly related to the strains. Hence, Eq. (3) results in cubic simultaneous equations.

(iv) Iterative Solution

The governing equations resulting from Eq. (3) are solved iteratively using modified Newton-Raphson process (Ref. 10). The basic Newton-Raphson method for the first load step may be outlined as follows:

1. Obtain a linear solution using the linear stiffness matrix $[K_0]$:

$$\{a_0\} = [K_0]^{-1} \{f\}; \quad (4)$$

2. Calculate residuals {ψ} with Eq. (3);
3. Check for convergence. Stop if {ψ} is sufficiently small;
4. Calculate tangential stiffness matrix $[K_T]$ as defined by the equation

$$d\{\psi\} = [K_T] d\{a\}; \quad (5)$$

5. Solve for correction to displacements:

$$\{\Delta a\} = -[K_T]^{-1} \{\Delta \psi\}; \quad (6)$$

6. Update the displacements:

$$\{a\} = \{a\} + \{\Delta a\} ; \quad (7)$$

7. Go to step 2.

When multiple load steps are used, only step 1 changes. After obtaining a converged solution for load step "i," the initial solution for the next load step is

$$\{a\}_{i+1} = \{a_i\} + [K_T]^{-1} \{\Delta f\}_i \quad (8)$$

where $\{\Delta f\}_i$ is the load increment after the i-th step.

In actual practice a modified Newton-Raphson process is used. In this process, the tangential stiffness is not calculated after every iteration but after a predetermined number of iterations. This saves considerable computation time.

2.2 Computation of Stiffness

A linear FE analysis requires the computation of stiffness matrix $[K_0]$ for an element in the following form (Ref. 10):

$$[K_0] = \int_V [B]^T [D] [B] dV \quad (9)$$

where $[B]$ is the matrix of strain-displacement relations obtained from Eq. (1) neglecting the nonlinear terms. In a nonlinear analysis one needs to evaluate a tangent stiffness matrix $[K_T]$ as defined by Eq. (6). On taking differentials from Eq. (3), $[K_T]$ can be obtained as

$$[K_T] = [K_0] + [K_L] + [K_\sigma] = [\bar{K}] + [K_\sigma] \quad (10)$$

where

$$[\bar{K}] = [K_0] + [K_L] = \int_V [\bar{B}]^T [D] [\bar{B}] dv \quad (11)$$

and

$$[K_\sigma] = \int_V [G]^T [M] [G] dv. \quad (12)$$

Here, $[\bar{B}]$, as defined in Eq. (3), incorporates the nonlinear strain-displacement relationship, $[G]$ contains the derivatives of the shape functions, and $[M]$ contains the stresses in the element. The matrix $[\bar{B}]$ is obtained as

$$[\bar{B}] = [B] + [B_L] \quad (13)$$

and

$$[B_L] = [A] \cdot [G] \quad (14)$$

where $[A]$ contains the derivatives of displacements. Thus,

$$[B]^T = \begin{bmatrix} N_x & 0 & 0 & 0 & N_z & N_y \\ 0 & N_y & 0 & N_z & 0 & N_x \\ 0 & 0 & N_z & N_y & N_x & 0 \end{bmatrix} \quad (15)$$

and

$$[B_L]^T = \begin{bmatrix} U_x N_x & U_y N_y & U_z N_z & U_z N_y + U_y N_z & U_z N_x + U_x N_z & U_y N_x + U_x N_y \\ V_x N_x & V_y N_y & V_z N_z & V_z N_y + V_y N_z & V_z N_x + V_x N_z & V_y N_x + V_x N_z \\ W_x N_x & W_y N_y & W_z N_z & W_z N_y + W_y N_z & W_z N_x + W_x N_z & W_y N_x + W_x N_z \end{bmatrix} \quad (16)$$

where N 's are vectors of the shape functions for the isoparametric 8-noded hexahedron and subscripts denote the derivatives. On carrying out multiplications in Eq. (13), it can be shown that $[K_\sigma]$ has the following structure:

$$[K_\sigma] = \begin{bmatrix} [H] & [0] & [0] \\ [0] & [H] & [0] \\ [0] & [0] & [H] \end{bmatrix} \quad (17)$$

where $[H]$ is defined by

$$[H] = \int_V [B_1]^T [S] [B_1] dv \quad (18)$$

where

$$[B_1]^T = [N_x, N_y, N_z] \quad \text{and} \quad [S] = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \sigma_y & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \sigma_z \end{bmatrix}. \quad (19/20)$$

The integrals are evaluated by numerical integration using Gaussian quadrature so that Eq. (12), for example, is evaluated as

$$[\bar{K}] = \sum_{i=1}^n w_i [B]_i^T [D]_i [B]_i |J|_i \quad (21)$$

Note that the right-hand sides of Eqs. (10), (12), and (19) have the same form, and similar logic can, therefore, be used in their evaluation. Hence, the algorithm given by Noor and Hartley (Ref. 3) for linear stiffness computation is also applicable in the case of the nonlinear stiffness computation. The method used in the present work follows very closely the algorithm of Noor and Hartley, and Ref. 3 can be seen for the details. The

computation of [K] proceeds as follows:

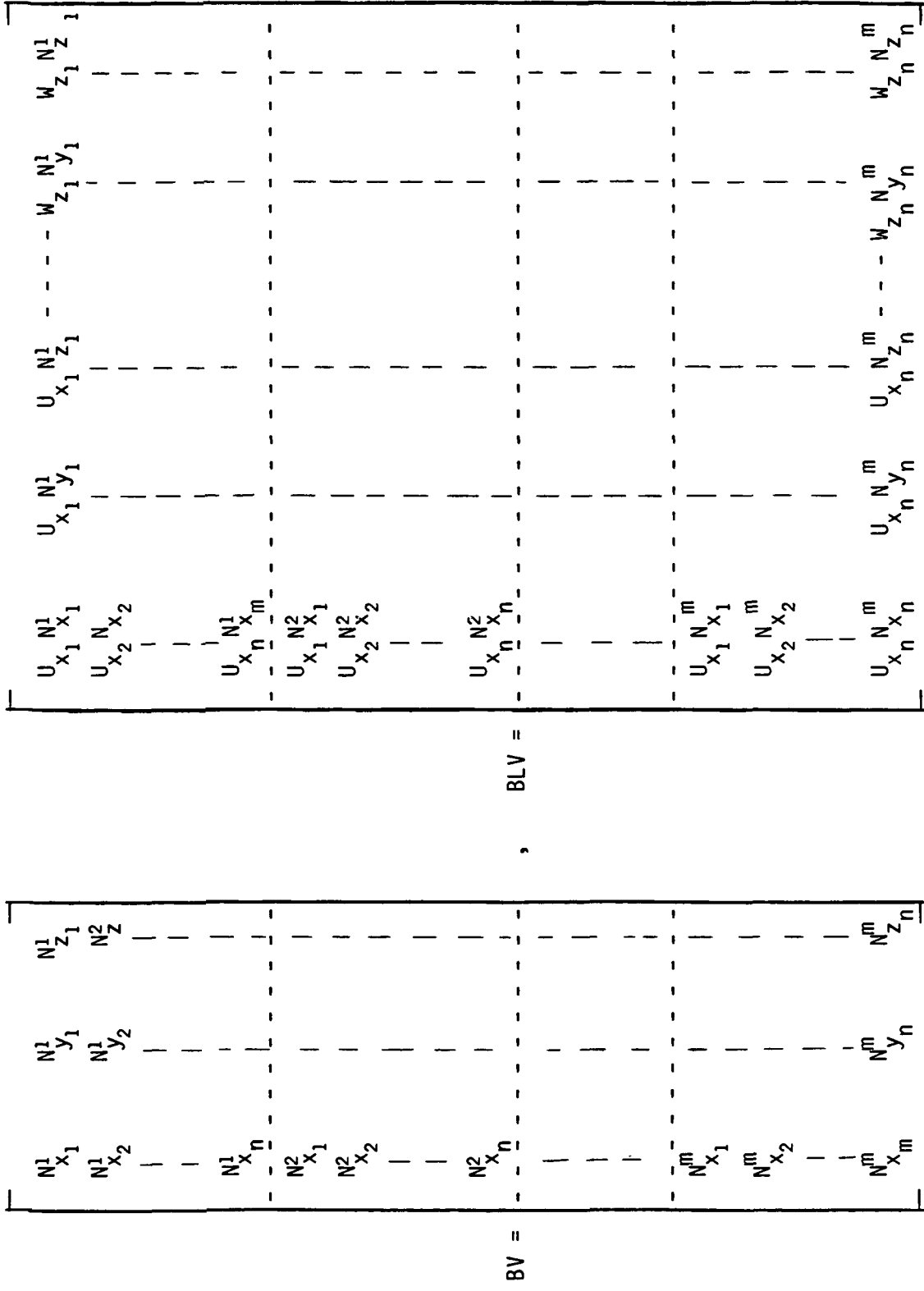
1. Long vectors of length $m \cdot n$ containing values of derivatives of m shape functions ($N_x, N_y, \text{etc.}$) at n integration points and products of these with derivatives of displacements ($u_x N_x, u_x N_y, \text{etc.}$) are formed. These vectors are stored in arrays BV and BLV as shown in Fig. 1. These now contain all the values required for matrices [B] and $[B_L]$ in Eqs. (16, 17). In the present program, the number of shape functions is $m=8$ and 3-point integration in each coordinate direction is used which gives $n=27$. Thus, vectors in BV and BLV have a length of 216.
2. The long vectors formed in step 1 are multiplied by the vector of weights and $|J|$.
3. Array SU is formed which contains the vectors in the product $[B]^T [D]$. The j -th column vector in SU is formed as the sum given by

$$\{SU\}_{ij} = \sum_{\ell} (BV)_{i\ell} * [D]_{\ell j}$$

where the index ℓ takes the values required by the strain-displacement matrix. A pointer matrix containing applicable values of 1 is formed for this purpose. The array SU has the same structure as BV and BLV and contains vectors of length mn . (See Fig. 2).

4. The nodal stiffness coefficients are now evaluated. For this purpose the element stiffness matrix is partitioned as shown in Fig. 3.
3. The independent blocks of the element stiffness matrix that occupy each column are formed simultaneously. The operations

(1) (2) (3) (1) (2) (3) (1) (2) (3) (26) (27)



$N_{x_j}^i$: x-derivative of i-th shape function at j-th integration point
 U_{x_j} : x-derivative of displacement u at j-th integration point. (m = 8, n = 27)

Fig. 1. Structure of arrays BV and BLV.

$$(SU)^T: \left[\begin{array}{cccc} s_1^i & s_2^i & \dots & s_n^i \\ s_1^2 & s_2^2 & \dots & s_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ s_1^m & s_1^m & \dots & s_n^m \end{array} \right]$$

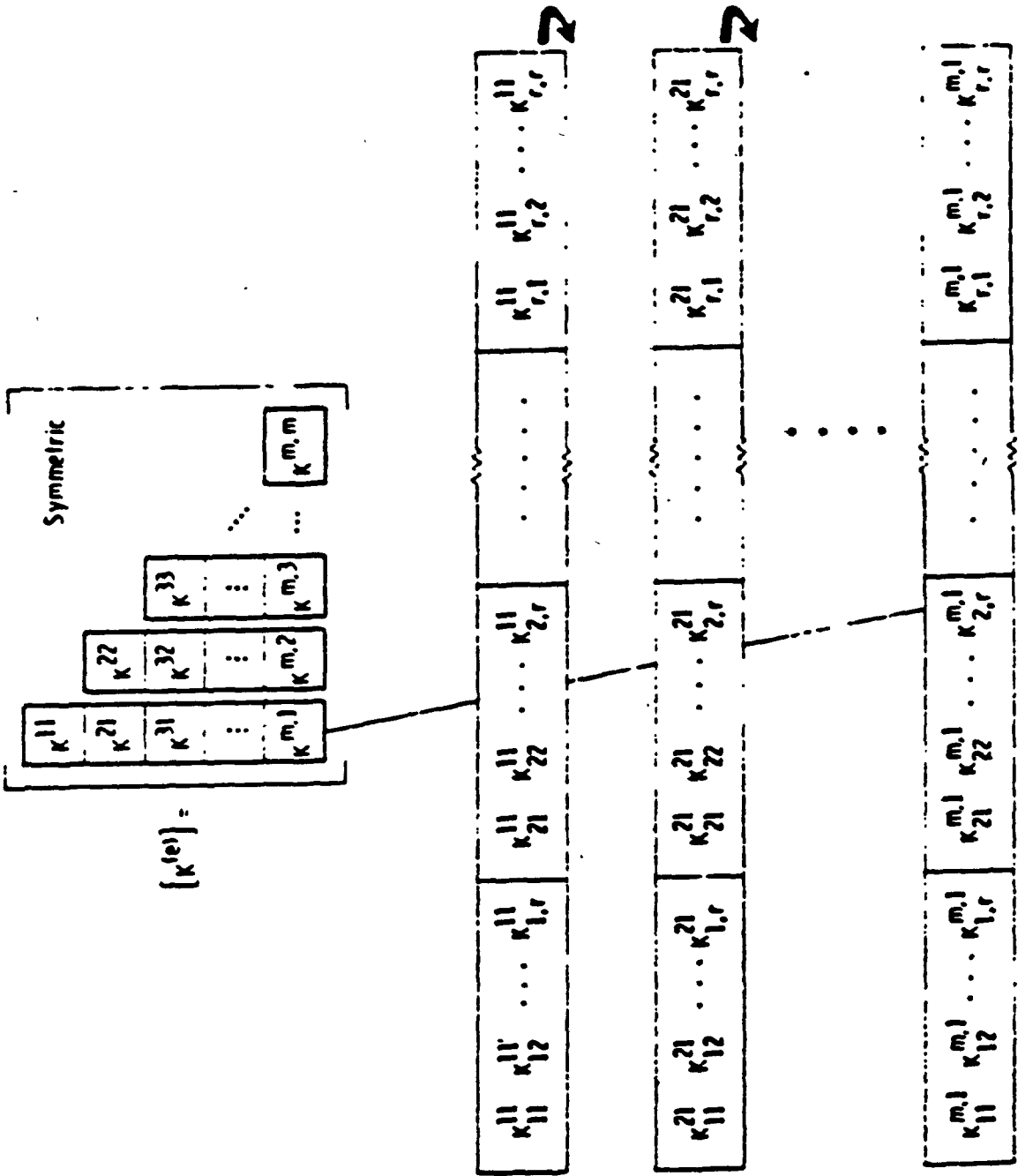
$$(SUM)^T: \left[\begin{array}{ccccccc} 1 & 2 & \dots & m & 2 & 3 & 4 \dots m \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array} \right]$$

$$\left[\begin{array}{ccc} \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \end{array} \right]$$

s_j^i : Element of SU corresponding to i-th shape function and j-th integration point.

$$i = \begin{bmatrix} s_1^i & s_2^i & s_3^i & \dots & s_n^i \end{bmatrix} \quad (m = 8, \quad n = 27)$$

Fig. 2. Structure of vector SU and vector SUM replicated from SU.



$k_{r,s}^{i,j} = (r,s)$ element in the (i,j) partition of the element stiffness matrix
Fig. 3. Storage organization of the element stiffness matrix $[K^{(e)}]$.

involved in the evaluation of the stiffness coefficients of the j -th column blocks (i.e. $[K_{ij}]$, $[K_{j,j+1}]$ - - - $[K_{j,m}]$) are as follows:

1. Array SUM is formed from the array SU by replicating j times the elements corresponding to the j -th shape function as shown in Fig. 2. The length of vectors in SUM is $nm(m+1)/2$.
2. Arrays BVJ and BLVJ are formed from BV and BLV, respectively, by replicating the values corresponding to j -th shape function $(m-j+1)$ times. Thus, the length of vectors in BVJ and BLVJ is now $nm(m+1)/2$. Array BV is used for the linear case. The sum of appropriate columns of BV and BLV as dictated by the strain-displacement relations is used for the nonlinear case.
3. The matrices $[K_{\ell j}]$ ($1=j$ to m) are formed by taking the products of vectors in SUM and BVJ or BLVJ. The elements of $[K_{ij}]$ are vectors of length $(m-j+1)n$. Once again, as in forming SU above, the strain-displacement relations decide which vectors from SUM would multiply each vector from BVJ or BLVJ. The same pointer matrix can be used for this purpose.
4. Stiffness coefficients $K_{\ell j}$ are then obtained by summing the n coefficients $K_{rs}^{(1)}$, $K_{rs}^{(2)}$ - - - $K_{rs}^{(3)}$ which occupy contiguous locations in memory.

Calculation of $[K_g]$ proceeds in the same manner with proper definitions of the $[B]$ and $[D]$ matrices. As seen from Eq. (18), only a smaller matrix $[H]$ needs to be calculated which then can be replicated to get $[K_r]$.

To make the element suitable for analyzing bending deformation use is made of reduced integration techniques (Ref. 10). In this technique, one or

more of shear strains are integrated by taking a lower order representation. For the present case of the isoparametric hexahedron, the lower order approximation makes the shear strains have a constant value in the element and can be easily incorporated in the computation of the stiffness matrix. For this purpose, the elements of the strain-displacement matrix which correspond to the strain being reduced-integrated are changed to their values at the centroid prior to the calculation of the stiffness. Similarly, to evaluate $[K_\sigma]$, the elements which multiply the shear stress corresponding to the shear strain being reduced-integrated are changed to their values at the centroid.

2.3 Computation of Strain Energy Release Rates

The strain energy release rates are calculated using a virtual crack extension technique similar to that reported in Ref. 11 and used in GAMNAS (Ref. 6, 8). This technique uses the forces transmitted across the crack tip to determine the energy release rates. The technique assumes that the change in geometry caused by a very small crack increment does not change the forces and displacements near the crack tip. The use of the technique requires that the mesh near the crack tip be uniform and symmetric about the crack plane. Because of the large displacements and rotations involved, the orientation of the crack plane is calculated in the deformed configuration in terms of the direction cosines of the normal to the crack increment plane, and the modes of fracture are redefined for this local system. The strain energy release rates are calculated for these modes of fracture. In calculating the energy, first the forces at the relevant nodes are reduced to force per unit length of the crack front (traction) using a strain energy equivalence. This is equivalent to inverting the process of obtaining the

consistent load vector for a given stress distribution. The energy obtained by multiplying these tractions by corresponding displacements is then divided by the length of the crack increment to get the strain energy release rates.

3. DESCRIPTION AND ORGANIZATION OF PROGRAM NAS3D

The program NAS3D is basically developed for 3-D geometric nonlinear analysis of lap shear geometries with cracks. An example of such a geometry would be a cracked lap shear specimen used for study of delaminations in composites or debonds in adhesively bonded joints. The program is written in FORTRAN VERSION 2 operating on CYBER 200 and VPS-32 systems at NASA Langley. It relies heavily on the vector processing capabilities of these machines for its speed and efficiency, NAS3D uses a simple 8-noded isoparametric hexahedron element. The iterative scheme for the nonlinear solution is a modified Newton Raphson process. It outputs the nodal displacements, stresses at the centroids of the elements, and reactions at the nodes. For the cracked configuration, it outputs the strain energy release rates along the crackfront in all the three modes as well as the total mean strain energy release rate.

The overall flow of NAS3D is similar to that of GAMNAS (Ref. 8). Even the material nonlinearity features of the GAMNAS are retained in the 3-D program, although this part has not been tested so far. This is done with a view to incorporate the material nonlinearity in the future. The flowchart of the program is shown in Fig. 4. Only one proportional load vector is input. The different load numbers (LOADNUM) refer to the scale factor by which the load vector is multiplied. For each new load, a linear incremental solution is obtained in the main program and then iterated in the

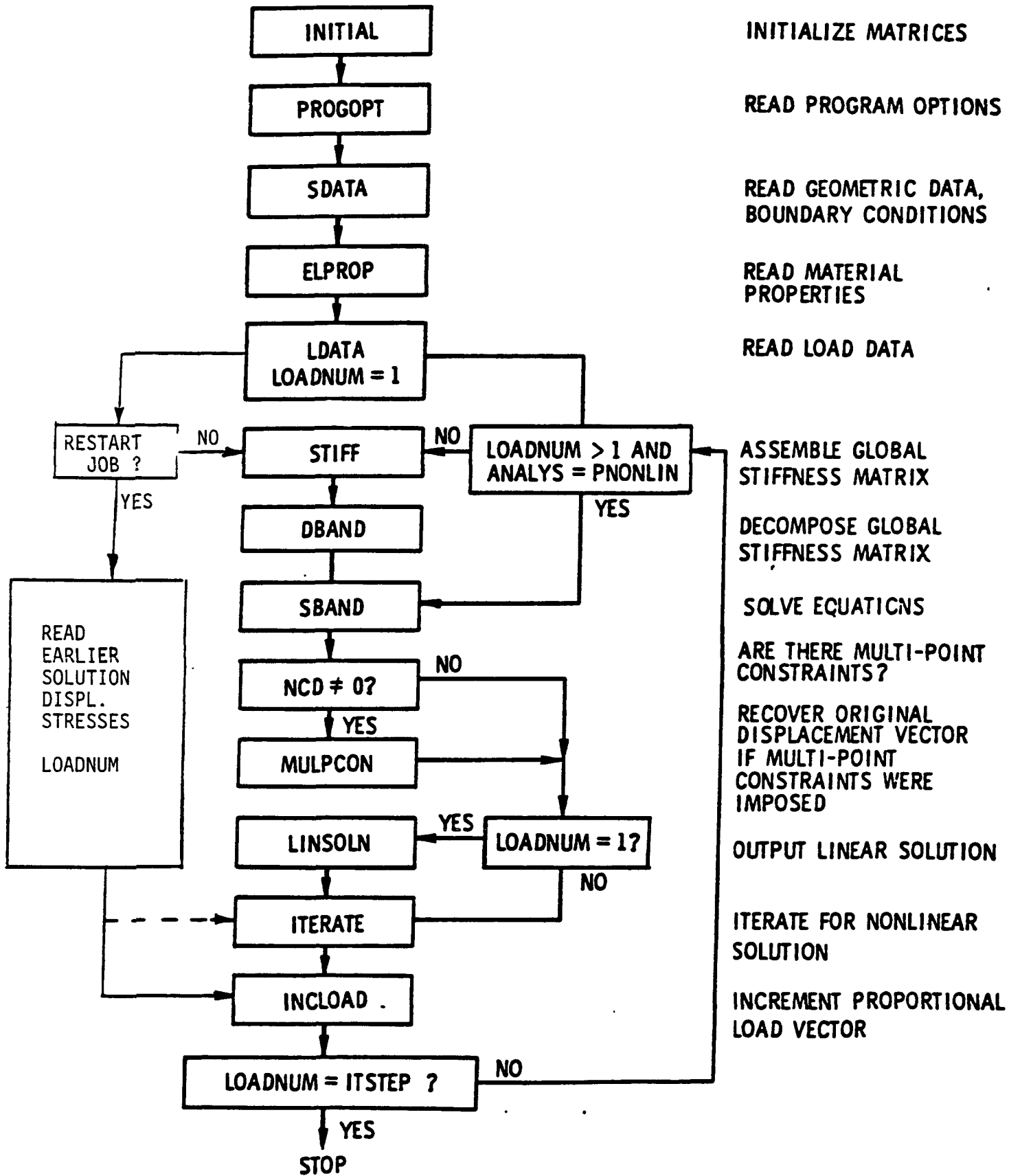


Fig. 4. Flowchart for main program.

routine ITERATE to obtain the incremental nonlinear solution. A restart option is included to facilitate the starting of the program from a solution to an intermediate load step. A description of the input data required for the program is given in the Appendix.

Table 1 gives the computation times obtained for various computations with NAS3D. Comparative times using conventional scalar algorithms are also given. The vector version of calculating stiffness matrix reduces the CPU time required by a factor 11 to 13 depending upon the choice of selective reduced integration. It may be noted here that the calculation of residuals and stresses is as time consuming as the stiffness calculation. Future efforts may be directed towards reducing the computations or computational times for the residuals.

4. SAMPLE PROBLEMS

In this section, some results obtained on four sample problems using NAS3D are given. These were used as test cases for the program and serve as useful examples.

4.1 Slender Beam

A slender beam of width 1", thickness .2" and length 20" is analyzed. The beam is supported at one end (see Fig. 5) and a transverse displacement w is applied at the center of the other end. The mesh and the configuration are shown in Fig. 5a. Although, the beam has no transverse stiffness initially (at no load), the geometric nonlinear effects stiffen the system as the transverse displacement increases. Figure 5b shows the calculated axial stress at the supported end of the rod at the mid-width and the edge. The results of 3-D FE analysis are shown by symbols and an exact solution obtained by the use of simple trigonometry for a rod with no bending stiffness

Table 1. CPU Times for Various Stages in the Nonlinear Analysis

Nodes = 8418, Elements = 6520, Bandwidth = 456, DOF = 25254

Operation		Time (seconds)
1.	Data Input	16.77
2.	Linear Stiffness and Assembly	
	(Scalar)	
	(Vector)	71.00
	Stiffness Time/Element (Vector)	.0096
3.	Nonlinear Tangent Stiffness and Assembly	
	(Scalar)	2180.00
	(Vector)	171.00
	Stiffness Time/Element (Scalar)	0.3200
	(Vector)	0.0251
4.	Stresses and Reactions (Vector)	140.0
5.	Residuals (Vector)	115.0
6.	Solution of Equations	
	(Scalar)	3933.0
	(Vector)	57.0

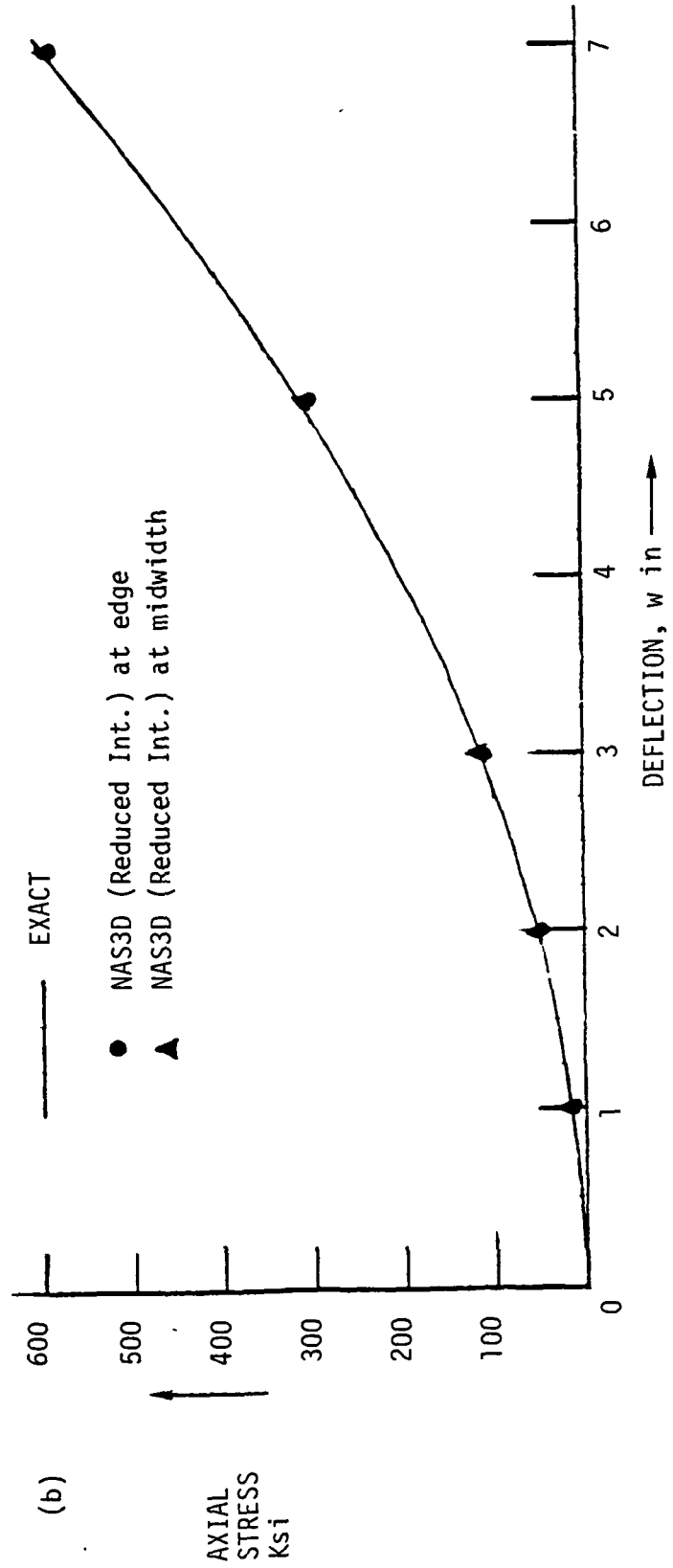
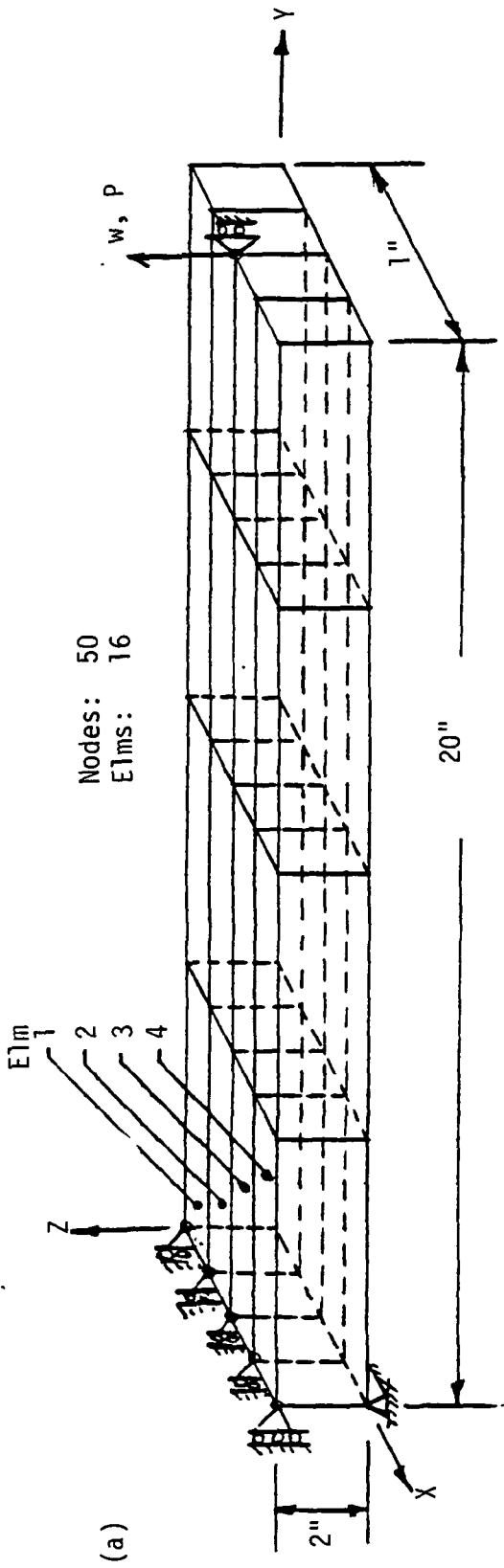


Fig. 5. Example of slender beam. (a) configuration and mesh, (b) nonlinear variation of stress with applied displacement.

is shown by the solid line. As seen from the figure, the FE results predict the nonlinearity very well. The 3-D results are close to the exact at the mid-width and only slightly lower at the edge. Table 2 gives the comparison of the stresses obtained with full and reduced integration. It is observed that at the low loads the reduced integration indicates some error. This shows that the reduced integration should be used judiciously and only where bending is involved.

4.2 Double Cantilever Beam (DCB)

Double cantilever beam geometries are extensively used in the study of fracture of composites and adhesive bonded joints. Figure 6 shows a symmetric DCB specimen with the FE mesh for the upper half which was analyzed. A load of 40 lb uniformly distributed over the 1" width was applied. The linear and nonlinear solutions were found to differ by an insignificant amount; for example, the error in tip deflection in the linear solution was less than 1%. The analysis was conducted using full integration, reduced integration of one shear strain (IREDU=1), and reduced integration of all shear strains (IREDU=3). The tip deflection w and average strain energy release rate G are compared with the strength of materials solution in Table 3. It is observed that the full integration performs poorly when bending is involved. The selective reduced integration allows the user to choose the proper reduced integration scheme suitable for a given problem. The difference in the two cases of reduced integration is not large here (Table 3) but it could be significant if the beam deforms in the width direction as well. In such a case, reduced integration of all shear strains may prove beneficial and necessary.

Figure 6b indicates the distribution of the strain energy release rate

Table 2. Example of Slender Beam Axial Stress in the Beam at Various Deflections.

Deflection w in	Axial stress (psi)				
	Reduced Integration		Full Integration		Theory
	Elm. 1	Elm. 2	Elm. 1	Elm. 2	
1	9.322	9.373	12.45	12.41	12.49
2	44.66	44.89	49.72	49.54	49.88
3	105.77	106.49	111.49	111.06	111.87
5	299.96	303.35	307.36	306.02	307.76
7	589.13	596.16	598.8	595.9	594.81

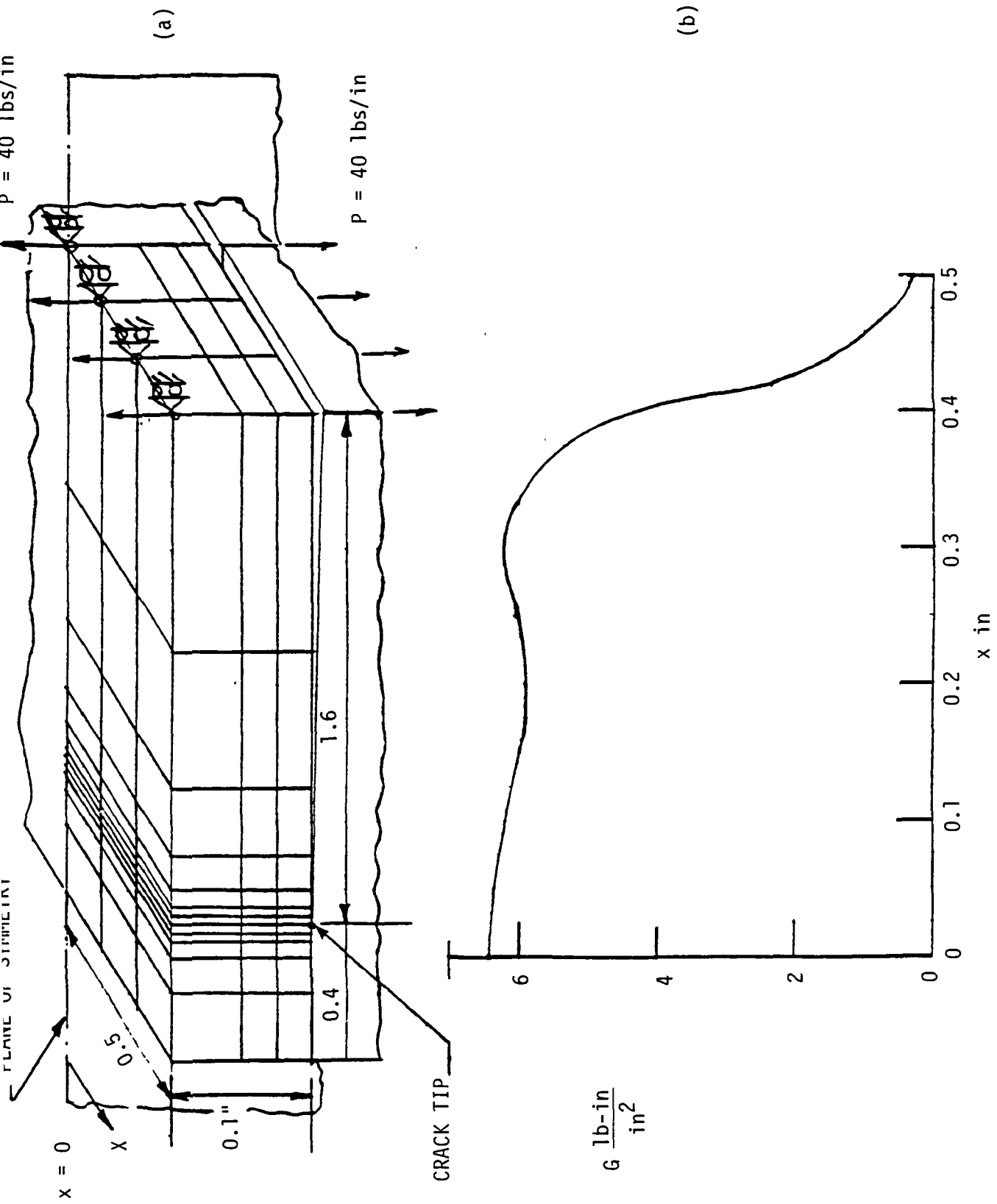


Fig. 6. Example of double cantilever beam. (a) configuration and mesh of quarter beam, (b) distribution of G along the crack front.

Table 3. Example of Double Cantilever Beam. Comparison of Full and Reduced Integration Results for $P = 40$ B/in.

Case	Tip Deflection w , in	Total Average strain energy release rate G lb. in/in ²
1. Full Integration	.0323	1.129
2. Reduced Integration of τ_{yz} only (IREDU=1)	0.06416	4.780
3. Reduced Integration of all Shear Strains (IREDU=3)	0.06595	4.928
4. Beam Theory	0.06540	4.915

along the crack-front. As should be expected the strain energy release rate has maximum value at the center and becomes negligibly small at the edge. Since the strain energy release rate decays to zero only in a small portion of the width at the edge, a finer mesh is necessary to get the accurate distribution of G . A finer mesh would result in a flatter curve in the central portion and a steeper decay at the edge.

4.3 Double Cracked Lap Shear Specimen

A symmetric double cracked lap shear specimen made of laminated composite is shown in Fig. 7a. A through-the-thickness delamination of length $a=1$ " exists symmetrically on either side of the specimen. Since the geometry is symmetric, not much of bending deformation is expected. Also, because of the symmetry, only one (upper) half of the specimen needs to be analyzed. NAS3D was used to carry out the 3-D analysis of this specimen. The mesh in the Y-Z plane used for the FE analysis is shown in Fig. 7b. The 3-D mesh is generated by repeating this mesh in every Y-Z plane for the subdivision in X-direction. Full integration was used. The results of the linear solution were compared with those obtained earlier by Raju (Ref. 9). The results of displacements and stresses were identical with Raju's for the identical meshes. The strain energy release rates along the crack-front obtained in these solutions are compared in Fig. 7b. Although the average values in the two solutions match some difference is observed in the actual distribution. This is due to the different techniques adopted to get energy per unit area from nodal forces and displacement. The present method of calculating consistent tractions and using them in G -calculation is expected to be more accurate than the earlier one.

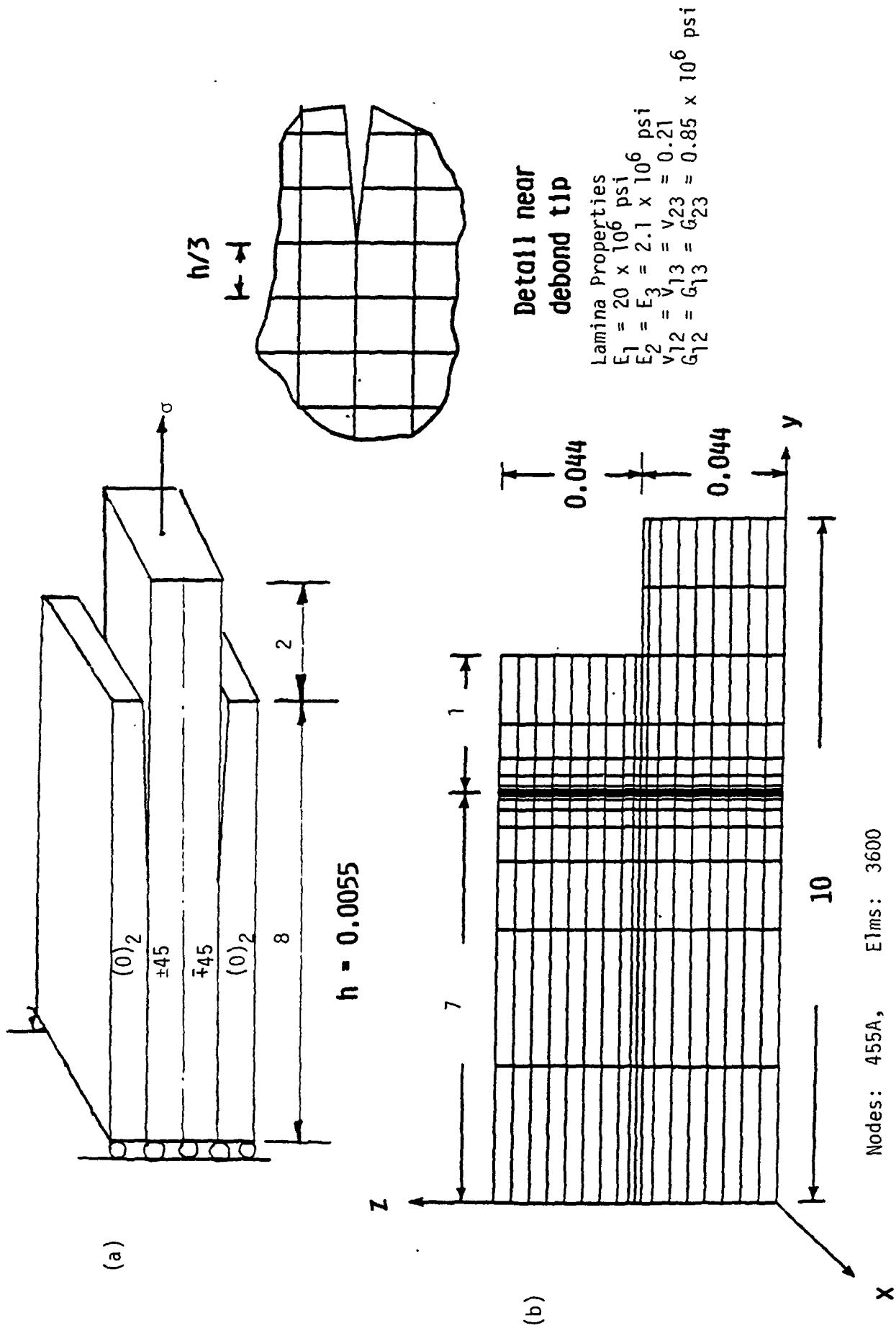


Fig. 7. Example of DCLS specimen, (a) configuration, and (b) FE mesh in y - z plane. The mesh division is repeated on each plane parallel to this plane to get 3-D mesh.

4.4 Cracked Lap Shear (CLS) Specimen

Cracked lap shear specimens are widely used in the study of mixed mode fracture related to debonds in adhesively bonded joints or delaminations in composites. One such specimen used for adhesive bond study is shown in Fig. 9a. The adherends are of equal thickness and made of aluminum with Young's modulus of 10500 ksi and Poisson ratio of .33. The adhesive Young's modulus is 280 ksi and Poisson ratio is 0.4. This specimen is one of the specimens used in a recent ASTM round robin effort on the analysis of CLS specimen (Ref. 7). The analysis was conducted using NAS3D for the case of debond length of 1". Reduced integration was used. The linear solution for $P=2500$ lb. was obtained. The distribution of strain energy release rates in various modes along the crackfront is shown in Fig. 9b. These results agree well with the results of an earlier 3-D analysis by Lof for the ASTM round robin (Ref. 7). However, the total average strain energy release rate is much higher than those obtained in nonlinear and linear 2-D analyses. This is surprising; but, it is expected that the nonlinear analysis will give the strain energy release rate comparable to the 2-D solutions. Further, to check the 3-D program plan strain conditions were simulated in the 3-D analysis by restraining the normal displacements of the edges of the specimen. The results of this analysis were within 2% of the 2-D plan strain linear analysis.

During the course of nonlinear analysis of this specimen, it was found that the initial load and the incremental load values required to obtain convergence were very small. Thus, nonlinear solutions for any load values of practical significance would require large computation time.

5. CONCLUDING REMARKS

An algorithm for computation of stiffnesses required in geometric

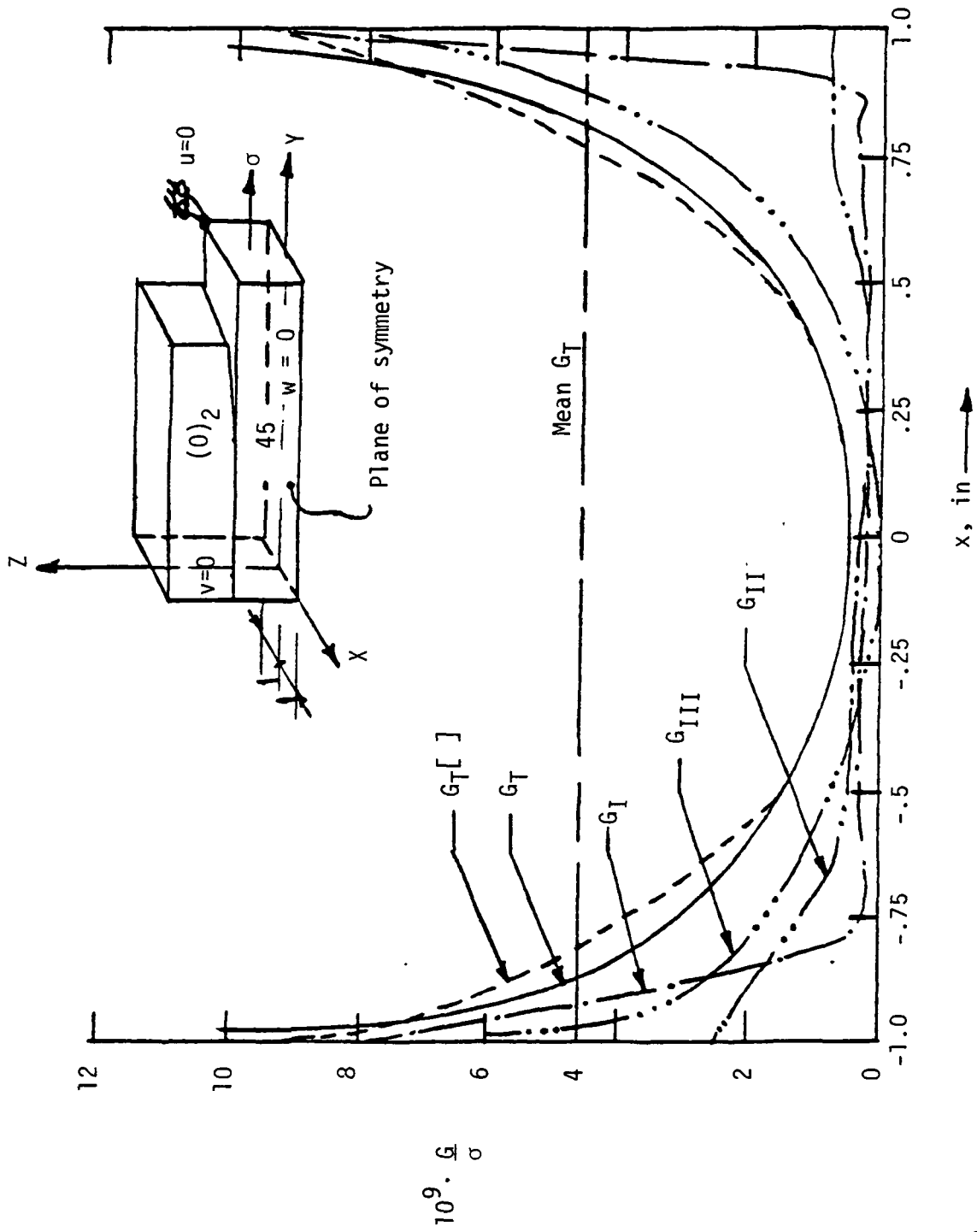
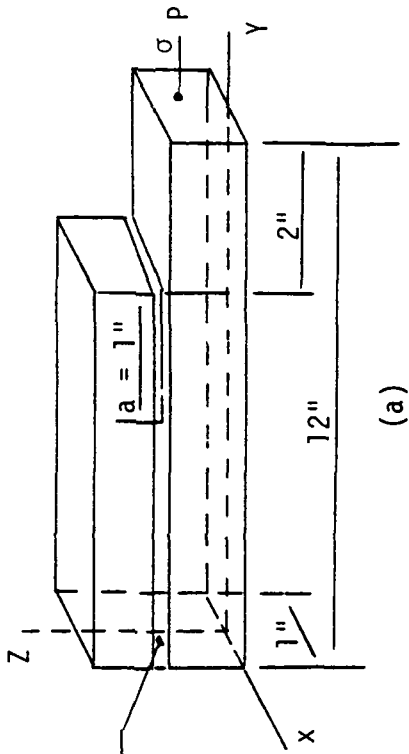


Fig. 8. Example of DCLS specimen. Total and individual strain energy release rates along the crack front in the linear solution.



Nodes: 8418
 E1ms: 6520

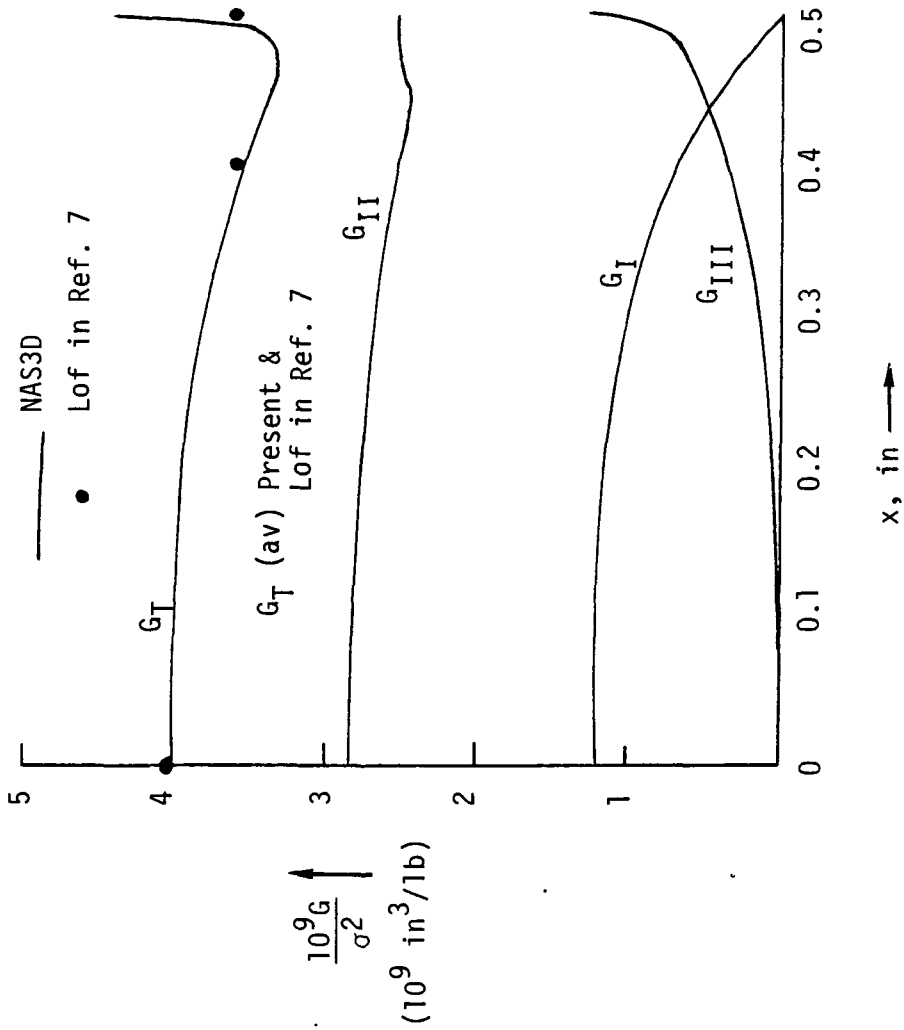


Fig. 9. Example of CLS specimen. (b) configuration, (b) G-distributions along the crack front.

nonlinear analysis was developed and used to generate the computer program NAS3D. Selective reduced integration of shear strains was incorporated to make the element suitable for bending. Strain energy release rate calculations were based on consistent tractions along the crack front. A provision for restarting the solution from an intermediate solution is also made. The code was tested by solving some simple problems having known solutions. Comparison of various computation times showed that savings by a factor of 11-13 were obtained in vectorized computation of stiffness over the conventional scalar computation. Calculation of residuals was found to be another time consuming step and future efforts may be directed to make this more efficient.

Successful use of any FE program for the structural analysis depends significantly on the ability of the analyst to predict qualitatively the response of the configuration. This insight, generally based on experience and simpler analyses, is particularly important for nonlinear analyses in which questions of convergence and uniqueness of solution and solution strategy have to be addressed. It is hoped that the present program coupled with some of this insight will prove to be a useful tool for the geometric nonlinear analysis.

ACKNOWLEDGEMENT

The author thanks Dr. I. S. Raju of Analytical Services and Materials Inc. and Mr. J. D. Whitcomb of NASA Langley Research Center for helpful discussions. The project was supported by NASA/Langley under Master Agreement NAS1-17993-20.

REFERENCES

1. J. J. Lambiotte Jr., The Solution of Linear Systems of Equations on a Vector Computer, Ph.D. Thesis, Univ. of Virginia, 1975.
2. J. M. Ortega and R. G. Voigt, "Solution of Partial Differential Equations on Vector and Parallel Computers," NASA CR 172500 (ICASE Report No. 85-1), 1985. Also, SIAM Review, Vol. 27, No. 2, 1985, pp. 149-334.
3. A. K. Noor and S. J. Hartley, "Evaluation of Element Stiffness Matrices on CDC STAR-100 Computer," Computers and Structures, Vol. 9, 1978, pp. 151-161.
4. I. S. Raju and J. C. Newman Jr., "Stress- Intensity Factors for a Wide Range of Semi-elliptical Surface Cracks in Finite Thickness Plates," Eng. Fracture Mech., Vol. 11, No. 4, 1979, pp. 817-829.
5. T. R. Brussat, S. T. Chiu and S. Mostovoy, "Fracture Mechanics of Structural Adhesive Bonds," AFML-TR-77-163, Air Force Materials Laboratory, 1977.
6. B. Dattaguru, R. E. Everett Jr., J. Whitcomb and W. S. Johnson, "Geometrically Nonlinear Analysis of Adhesively Bonded Joints," J. Eng. Mat. and Tech., Vol. 106, Jan. 1984, pp. 59-65.
7. W. S. Johnson, "Stress Analysis of the Cracked Lap Shear Specimen: An ASTM Round Robin," ASTM J. of Testing and Evaluation (forthcoming).
8. J. D. Whitcomb and B. Dattaguru, "User's Manual for GAMNAS -- Geometric and Material Nonlinear Analysis of Structures," NASA TM 85734, 1984.
9. I. S. Raju, B. Dattaguru and J. D. Whitcomb, "2-D, Quasi-3-D and 3-D analyses of composite joints," (forthcoming).
10. O. C. Zienkiewicz, The Finite Element Method, 3rd Ed., Tata-McGraw-Hill Pub. Co., New Delhi, 1979.
11. E. F. Rybicki and M. F. Kanninen, "A Finite Element Calculation of Stress Intensity Factors by a Modified Crack Closure Integral," Eng. Fracture Mech., Vol. 9, 1977, pp. 931-938.

APPENDIX

INPUT DATA

The required input data is described in this section. Where applicable, the maximum allowable values of the input parameters are noted.

<u>Card Set</u>	<u>Parameters</u>	<u>No. of Cards</u>	<u>Format</u>
1.	TITLE(I), I = 1,60 TITLE = TITLE OF PROBLEM	3	20A4
2.	OUTPUT, ANALYS, QUADRAT, ENERGY OUTPUT = Output option = XLONG for long output = SHORT for output (the nodal coordinates, element connectivity, and boundary conditions are not in the output) ANALYS = Type of analysis = XLINEAR for linear analysis = GNONLIN for geometrically nonlinear analysis = PNONLIN for materially nonlinear analysis = CNONLIN for combined geometric and material nonlinear analysis QUADRAT = Integration option = REDUC for reduced integration = XFULL for full integration ENERGY = Option for strain-energy release rate calculations	1	5A8

<u>Card Set</u>	<u>Parameters</u>	<u>No. of Cards</u>	<u>Format</u>
	= DO-G for G calculation = DONOJG for no G calculation		
3.	IREDUO	1	16I5
	IREDUO = Number of shear strains to be reduced integrated (1 for τ_{yz} , 2 for τ_{yz} and τ_{zx} , 3 for τ_{yz} , τ_{zx} and τ_{xy})		
4.	ITSTEP, NCYCLE, IMAX	1	3I5
	ITSTEP = Number of steps in the incremental loading minimum = 1, maximum = 30 NCYCLE = Number of iterations between updates of stiffness matrix IMAX = Maximum number of iterations allowed before terminating		
5.	ACCURACY	1	F10.3
	ACCURACY = Maximum residual allowed in converged solution as a fraction of applied load		
6.	NN, NE, NRN, NLX	1	16I5
	NN = Number of nodes in the FE model NE = Number of elements in the FE model NRN = Number of nodes with a restrained degree of freedom NLX = Number of nodes in the X-direction (direction normal to the 2-D mesh) (minimum = 1, maximum = 10)		
7.	X(I), I = 1,NLX	NLX/8 [†]	8F10.5
	X() = Coordinates in X direction for mesh division		
8.	<u>Nodal Coordinates</u> <u>x-coordinate</u>		

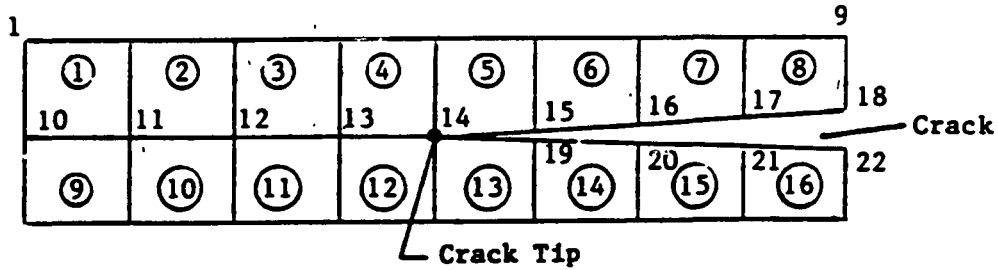
<u>Card Set</u>	<u>Parameters</u>	<u>No. of Cards</u>	<u>Format</u>
	XX, N(I) = 1,13 XX = coordinate XX = coordinate N() = list of nodes with coordinate XX *Input until all x-coordinates are specified. End x-coordinate data with a blank card.	*	E10.4, 13I5
	<u>y-coordinate</u>		
	XX, N(I), I = 1.13 *Similar to input of x-coordinates	*	E10.4, 13I5
9.	I, IN(I), JN(I), KN(I), LN(I)	NE	5I5
	I,IN,JN,KN,LN = Element number, four node numbers for element I. Nodes must be specified in a counterclockwise direction.		
10.	K, NRL (3*K-2), NRL (3*K-1), NRL (3*K)	NRN	4I5
	K = Node number NRL (3*K-2), NRL (3*K-1), NRL (3*K) = Constraints in X, Y and Z directions, respectively, at node K. 0 indicates no constraint 1 indicates constraint		

Note: Do not include degrees of freedom involved in

<u>Card Set</u>	<u>Parameters</u>	<u>No. of Cards</u>	<u>Format</u>
	multipoint constraints. Do include degrees of freedom with specified displacements.		
11.	IANC	1	16I5
	IANC = Number of additional nodal constraints on planes or lines.		
12.	XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX	3*	2F20.8
	XMIN, XMAX = Range of x-coordinate		
	YMIN, YMAX = Range of y-coordinates		
	ZMIN, ZMAX = Range of z-coordinates		
13.	ICONX, ICONY, ICONZ	1*	16I5
	ICONX = Restraint in x-direction		
	ICONY = Restraint in y-direction		
	ICONZ = Restraint in z-direction		
	(=1, restrained j = 0, free)		
	SKIP 14-17 IF ENERGY = DONOJG		
14.	INP	1	I5
	INP = Number of node sets used in virtual crack extension calculation (maximum = 15)		
15.	NEGCAL(I), I = 1, (INP+1)	$(INP+1)/16^\dagger$	16I5
	NEGCAL = Element numbers for elements in 2-D mesh contributing to the nodal forces required for virtual crack extension. (See example in sketch below. Element numbers are circled.)		

[†]Round off to next higher integer.

<u>Card Set</u>	<u>Parameters</u>	<u>No. of Cards</u>	<u>Format</u>
-----------------	-------------------	---------------------	---------------



IF INP = 3,

NEGCAL (1 to 4) = 2, 3, 4, 5

NFGCAL (1 to 3) = 14, 13, 12

NDGCAL (1 to 6) = 15, 19, 16, 20, 17, 21

16. NFGCAL(I), I = 1, INP INP/16[†] 16I5

NGGCAL(I) = Node numbers for nodes along which virtual crack extension forces are calculated.

List according to distance from crack tip, with the crack tip node as the first one.

17. NDGCAL(I), I = 1, (2*INP) 2*INP/16[†] 16I5

NDGCAL(I) = Node numbers for the nodes used to calculate cracking opening and sliding displacements

Repeat card sets 13-16 for each material group.

Maximum number of material groups = 10

End last group with blank card.

18. J, XMATER(J), ELASOPT 1 15, A8

J = Material group number

XMATER = Material type

= ELASTIC for linear stress-strain curve

= ELPLAST for elastic-perfectly plastic stress-strain

[†]Round off to next higher integer.

<u>Card Set</u>	<u>Parameters</u>	<u>No. of Cards</u>	<u>Format</u>
	curve = BILINEAR for bilinear stress-strain curve = RAMOSGO for Ramberg-Osgood stress-strain curve ELASOPT = YUNGMOD, when elastic constants of the material are input = FULDMAT, when [D] is input columnwise (Use ELASOPT = YUNGMOD with the present version)		
19.	EPROP(I), I = 1, 10	2	8E10.3
	EPROP = Contains the material elastic constants in the following order $E_1, E_2, E_3, G_{23}, G_{31}, G_{32}, \nu_{23}, \nu_{31}, \nu_{12}$, and θ where θ = Ply angle with respect to x-axis		
20.	YIELDS, ET, RO, ANM	1	5E10.3
	YIELDS = Yield stress ET = Tangent modulus for yielded bilinear material RO, ANM = Parameters defining Ramberg-Osgood stress-strain relation, $\epsilon = \frac{\sigma}{E} + \left(\frac{\sigma}{RO}\right)^{ANM}$ (a) If XMATER = ELASTIC, input YIELDS = ET = RO = 1.0×10^{21} , ANM = 10 (b) If XMATER = BILINEAR, input proper YIELDS and ET and set RO = ANM = 0.0 (c) If XMATER = RAMOSGO, input proper YIELDS, RO and ANM and set ET = 0.0		

<u>Card Set</u>	<u>Parameters</u>	<u>No. of Cards</u>	<u>Format</u>
21.	NEL1, NEL2, NELINC	*	3I5
	NEL1, NEL2, NELINC = Loop parameters used to define elements in material group NEL1 = First element NEL2 = Last element NELINC = Loop increment e.g., 1, 50, 20 defines elements 1, 21, and 31 to be in material group		
	*Repeat until all elements in group are defined.		
	End card set 16 by specifying NEL1 = NEL2 = NELINC = 0		
22.	DELLOAD(I) = 1, ITSTEP	ITSTEP/8 [†]	8F10.3
	DELLOAD(I) = Scale factor for proportional load vector for load step I. Always specify DELLOAD(I) = 1.0		
23.	NLN, NCD, NED	1	3I5
	NLN = Number of nodes with applied loads		
	NCD = Number of multipoint constraints, max = 15		
	NED = Number of specified displacements, max = 30		
24.	K, FX, FY, FZ	NLN	I5, 3F10.3
	K = Node number		
	FX, FY, FZ = Loads in x, y, and z directions, respectively		
25.	K, KDF, URD	NED	2I5, F10.3
	K = Node number		
	KDF = Displacement direction, specify 1 for x direction specify 2 for y direction		

[†]Round off to next higher integer.

<u>Card Set</u>	<u>Parameters</u>	<u>No. of Cards</u>	<u>Format</u>
		specify 3 for z direction	
	URD = Magnitude of displacement		
	SKIP 21-24 IF NCD = 0		
26.	NMPR(I), I = 1, NCD	NCD/16	16I5
	NMPR(I) = Number of degrees of freedom involved in the Ith multipoint constraint, max = 20		
27.	((ICDN(I,J),J=1, NMPR(I)), I = 1, NCD)	NCD sets	16I5
	(ICDN(I,J) = Jth degree of freedom involved in the Ith multipoint constraint		
28.	NZKV	1	I5
	NZKV = Number of multipoint constraints for which there is an applied load		
29.	NKV, ATOT		
	NKV, ATOT: ATOT is the non-zero load associated with the NKV set of constrained nodes		
30.	NPANS	1	I5
	NPANS = Number of planes/lines with applied uniform normal stress		
	SKIP 31 if NPANS = 0		
	Repeat 31-33 NPANS times.		
31.	NDIR, ASTR	1	I5, F20.8
	NDIR = Direction of applied stress (1 = x, 2 = y, 3 = z)		
	ASTR = Value of applied stress		

<u>Card Set</u>	<u>Parameters</u>	<u>No. of Cards</u>	<u>Format</u>
32.	XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX XMIN, XMAX = Range of x coordinate for the plane/line YMIN, YMAX = Range of y coordinate for the plane/line ZMIN, ZMAX = Range of z coordinate for the plane/line	3	2F20.8
33.	NI, NF, NPCHX, NPCHY, NPCHZ NI = Lowest node number in the plane/line NF = Highest node number in the plane/line NPCHX = Pitch in x-direction of node numbers NPCHY = Pitch in y-direction of node numbers NPCHZ = Pitch in z-direction for node numbers	1	16I5

34. IREDO 1 I5
IREDO = Restart parameter
(0 = regular job, 1 = restart job)

Note: The information required for restarting a job from an intermediate load step is written on TAPE7. This file should be saved and used for restarting the job. Thus, in the first run of the job IREDO is given as 0 and file on TAPE7 is saved. In the next run (for restarting at the point where the first run was over), IREDO is given as 1 and an earlier solution is read from TAPE7.