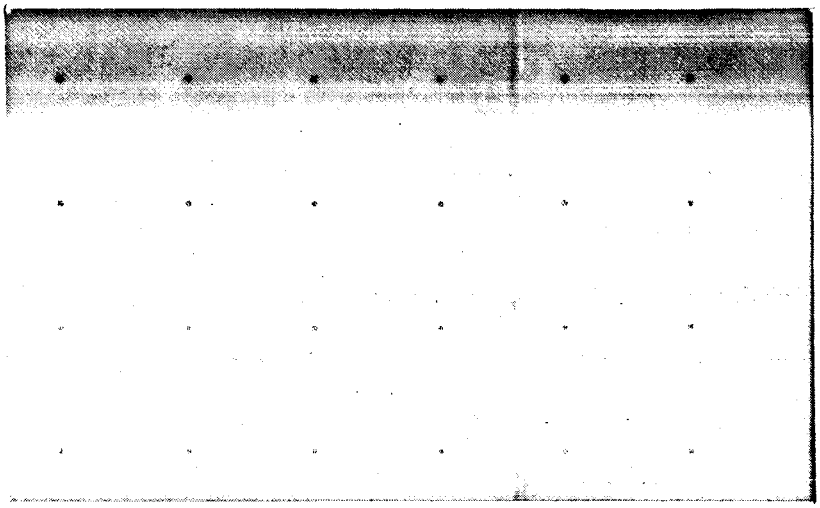
DAA/ LANGLEY

IN-39 P. 142

NAG1-215

CENTER FOR COMPUTER AIDED DESIGN

82561 R-FILE CR



College of Engineering
The University of Iowa
Iowa City, Iowa 52242

(NASA-CR-181101) DESIGN SENSITIVITY ANALYSIS WITH APPLICON IFAD USING THE ADJOINT VARIABLE METHOD (Iowa Univ.) 142 p Avail: NTIS HC A07/MF A01 CSCL 20K

N87-27220

Unclas
G3/39 0082561

Technical Report 84-17

DESIGN SENSITIVITY ANALYSIS WITH APPLICON IFAD USING THE ADJOINT VARIABLE METHOD

Marjorie C. Frederick and Kyung K. Choi

Center for Computer Aided Design
College of Engineering
The University of Iowa

December 1984

Research was partially supported by
NASA Grant NAG-1-215
and Applicon

ABSTRACT

A numerical method is presented to implement structural design sensitivity analysis using the versatility and convenience of existing finite element structural analysis program and the theoretical foundation in structural design sensitivity analysis. Conventional design variables, such as thickness and cross-sectional areas, are considered. Structural performance functionals considered include compliance, displacement, and stress. It is shown that calculations can be carried out outside existing finite element codes, using postprocessing data only. That is, design sensitivity analysis software does not have to be imbedded in an existing finite element code.

The finite element structural analysis program used in the implementation presented is IFAD. Feasibility of the method is shown through analysis of several problems, including built-up structures. Accurate design sensitivity results are obtained without the uncertainty of numerical accuracy associated with selection of a finite difference perturbation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

a_u	Bilinear form which is dependent on u
ℓ_u	Load linear form which is dependent on u
u	Design vector
ℓ	Design sensitivity vector; Gauss quadrature counter; Local coordinate system of a beam
ψ, ψ_i	Various constraint fuctionals
ψ', ψ_i'	Design sensitivites of various constraint functionals
δu	Perturbed design vector
h	Thickness of membrane; heigth of beam
t	Thickness of bending plate
b	Width of beam
E	Young's modulus
ν	Poisson's ratio
$\hat{D}(u)$	Flexural rigidity of plate
J	Torsional rigidity of beam; determinant of the Jacobian
W, W_ℓ	Gauss quadrature weight factor
G	Shear modulus
$\hat{\delta}$	Kroneker Delta
$\varepsilon, \varepsilon^{ij}$	Strain
σ, σ^{ij}	Stress
$\lambda, \lambda^{(i)}$	Adjoint variable vector
Ω	Domain of the system considered

Γ	Boundary of the system considered
$\sum$	Summation
z, z^i, z^i_ℓ	Displacement
F, F^i, F^i_ℓ	Applied body force
T, T^i	Applied traction
$\bar{z}, \bar{z}^i$	Virtual displacement
δh	Perturbation of h
δh	Perturbation of b
δt	Perturbation of t
m_p	Characteristic function
∂	Partial derivative
[ ]	Matrix
$\bar{d}$	Virtual displacement of an element
$z_{ii}, z_{xx}, z_{xx_\ell}$	Beam curvature due to z
$\bar{z}_{xx}, \bar{z}_{ii}$	Beam curvature due to $\bar{z}$
k	Element counter
γ	Material density
$\bar{\lambda}_{xx}, \lambda_{xx_\ell}$	Beam curavature due to $\bar{\lambda}$
L	Length of a beam element
A	Area of a triangle
ϕ	General function

CHAPTER I

INTRODUCTION

1.1 Purpose

To date there exists a wide variety of finite element structural analysis programs that are used as reliable tools for structural analysis. They give the designer pertinent information such as stresses, strains and displacements of the mechanical system being modeled. However, if this information reveals that the mechanical system does not meet specified constraint requirements, the designer must make intuitive guesses as to how to improve the design. If the mechanical system is complex, it becomes very difficult to decide what step must be taken to improve the design. There is however, substantial literature on the theory of design sensitivity analysis, which predicts the effect that structural design changes have on the performance of a mechanical system. Use of this technique has been primarily confined to papers in structural optimization literature.

The purpose of this work is to develop and implement structural design sensitivity analysis using the adjoint variable method that takes advantage of the versatility and convenience of an existing finite element structural analysis program and the theoretical foundation in structural design sensitivity analysis that is found in Ref. 1. The finite element program that will be used is IFAD [3]. It is developed by Applicon Inc. and has been provided to the Center for Computer Aided Design for the use in this study.

In order to check the feasibility of using the design sensitivity analysis technique with IFAD, an approximation of the differential ψ' of a structural performance measure ψ is made using the finite difference method. An appropriate design perturbation δu must be selected in order to insure accuracy of the perturbation $\Delta\psi$ of the constraint functional. If δu is too small, $\Delta\psi = \psi(u + \delta u) - \psi(u)$ may be inaccurate due to loss of significant digits in the difference. On the other hand, if δu is too large, $\Delta\psi$ will be influenced by nonlinearities and the differential approximation will be inaccurate. The feasibility check procedure is outlined in the flow chart of Fig. 1. Details of the calculations of the constraint functionals, the adjoint loads, and the design sensitivity vectors for each constraint functional are described in Chapter II, for different types of finite elements. The design sensitivity ψ' of the constraint functional is the scalar product of the design sensitivity vector ℓ and the design variable perturbation vector δu. If the design variable is constant throughout the finite element model of the mechanical system, this becomes a scalar multiplication. If the design sensitivity is an accurate prediction of the performance of the mechanical system due to a design change, it should be equivalent to the difference of the constraint functionals of the two finite element models, the original and the perturbed model.

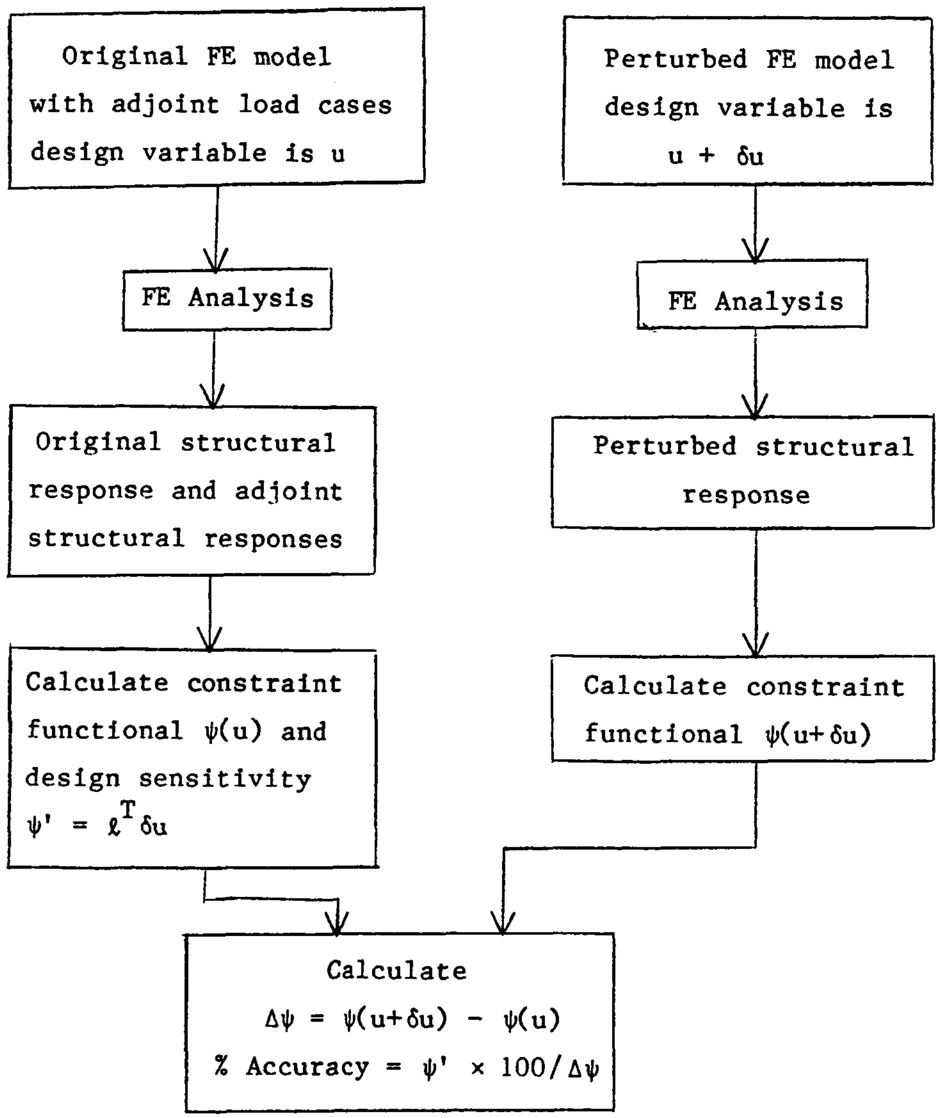


Figure 1. Flow Chart of Feasibility Check Procedure

1.2 Adjoint Variable Method

A number of methods could be used to implement structural design sensitivity analysis with an existing finite element code, but the most powerful is the adjoint variable method. This method can be implemented outside of an existing finite element code, using only postprocessing data. This is convienent, because the source code for most finite element programs is not readily available. If the code is available, less programming is involved. The same subroutines for the element shape functions used in the finite element model can be used in the design sensitivity analysis, since this method is dependent on element type. Generality is another factor that adds flexibility to the adjoint variable method. The code can be written to include basic design variables, constraints and loading conditions. This enables the designer to choose what design variables to modify to give the best design improvement.

The adjoint variable method can easily be used for complex mechanical systems that have more than one structural component. The details of this procedure are discussed in Section 2.2. Design sensitivity of a built-up structure is formed by combining the design sensitivities of each structural component. The only precaution that is necessary is in making sure that the interaction between the components is taken into account.

1.3 Adjoint Variable Method Results

The design sensitivity vector is the derivative of the constraint functional with respect to the design variables. It has the same number of components as there are elements in the finite element model. The magnitude of each component reflects how sensitive the element is to a change in design relative to the constraint functional. If the vector component is negative, the corresponding design variable should be decreased to increase ψ. Likewise, if the vector component is positive, the design variable should be increased to increase ψ. In addition, if the magnitude of the vector component is large, then the corresponding design variable will have a more substantial effect on design improvement.

When a designer uses a finite element structural analysis in design of a mechanical system, it is most likely that a number of program runs are necessary before a substantially improved design is obtained. With the aid of a design sensitivity vector, the designer will know what direction to take to improve the design most efficiently.

CHAPTER II

DESIGN SENSITIVITY ANALYSIS METHOD

2.1 Calculation Procedure for Structural Components

To implement the adjoint variable technique of design sensitivity analysis, the adjoint load for each constraint functional must be calculated. This procedure is developed in Ref. 1 using compliance, displacement, stress and natural frequency as constraint functionals. For the compliance functional, the adjoint equation is the same as the state equation. In this special case the adjoint system does not need to be solved. For the displacement functional the adjoint load is a unit point load acting at the point where the displacement constraint is imposed. To calculate the adjoint response it is necessary only to restart the finite element analysis with unit loads applied at varying points along the structure. For the stress functional the shape function of the structural component used in the finite element analysis must be known. This shape function is used to calculate the adjoint load for a stress constraint of a specific element of the structure. From this point the procedure is similar to the displacement functional, in that a restart of the finite element analysis must be completed using the adjoint loads of elements as other load cases.

The flow chart of Fig. 2 shows the overall process. This procedure is implemented after the structural response of the finite element

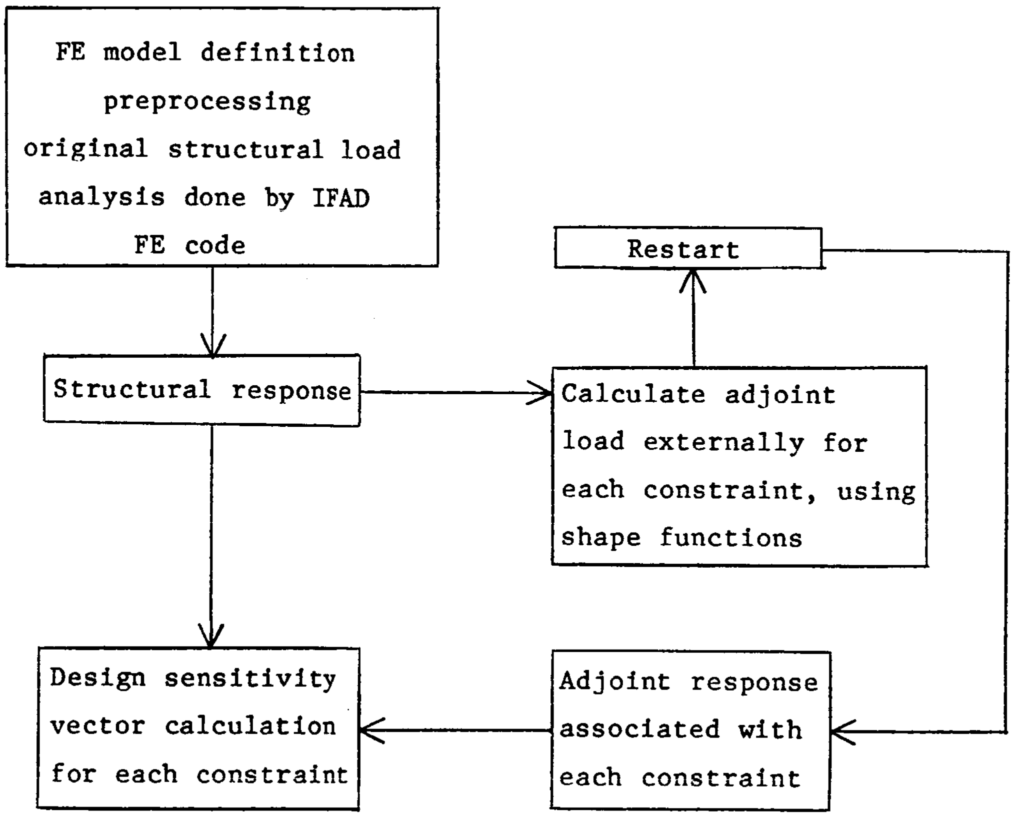


Figure 2. Flow Chart of Design Sensitivity Calculation Procedure

model due to the original load has been solved. The original structural response plus the adjoint response for each constraint is then utilized to calculate the design sensitivity vectors.

The following sections give detailed explanations of the calculation procedures and equations necessary for analyzing membranes, bending beams and bending plates.

2.1.1 Membranes

Consider a variable thickness thin elastic clamped solid, as shown in Fig. 3. The design variable is taken as the variable thickness $u = h(x)$ of the plate.

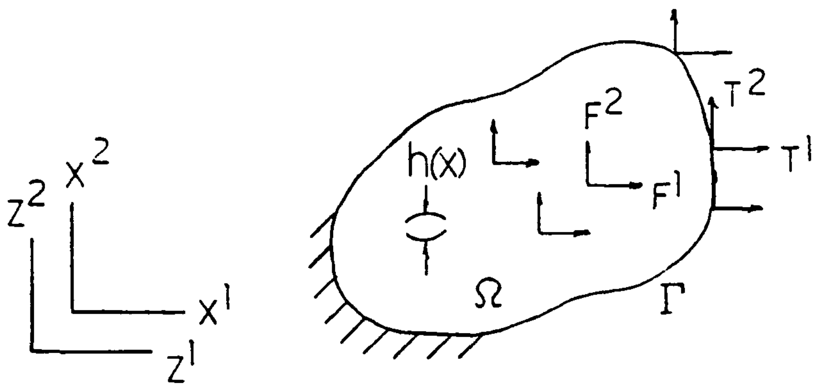


Figure 3. Clamped Elastic Solid of Variable Thickness h(x)

The energy bilinear form and the load linear form of the plane elasticity problem are given as [1]

$$a_u(z,\bar{z}) = \iint_\Omega h(x) \sum_{i,j=1}^{2} \sigma^{ij}(z)\varepsilon^{ij}(\bar{z})d\Omega \tag{1}$$

and

$$\ell_u(z,\bar{z}) = \iint_\Omega h(x)[\sum_{i=1}^{2} F^i \bar{z}^i] d\Omega + \int_\Gamma [\sum_{i=1}^{2} T^i \bar{z}^i] d\Gamma \tag{2}$$

where $z = [z^1, z^2]^T$ is the displacement, $F = [F^1, F^2]^T$ is the applied body force, $T = [T^1, T^2]^T$ is the traction, and $\sigma^{ij}(z)$ and $\varepsilon^{ij}(\bar{z})$ are the stress and strain fields associated with the displacement z and the virtual displacement $\bar{z}$ respectively. The state equation is given as [1]

$$a_u(z,\bar{z}) = \ell_u(\bar{z}) \tag{3}$$

for all kinematically admissible virtual displacement $\bar{z}$.

First consider the functional representing the compliance of the structure as

$$\psi_1 = \iint_\Omega h(x)[\sum_{i=1}^{2} F^i z^i] d\Omega + \int_\Gamma [\sum_{i=1}^{2} T^i z^i] d\Gamma \tag{4}$$

The first variation of Eq. (4) is

$$\psi_1' = \iint_\Omega [\sum_{i=1}^{2} F^i z^i] \delta h \, d\Omega + \iint_\Omega h[\sum_{i=1}^{2} F^i z^{i'}] d\Omega + \int_\Gamma [\sum_{i=1}^{2} T^i z^{i'}] d\Gamma \tag{5}$$

In order to eliminate the dependence on the state variable in Eq. (5), it is necessary to define the adjoint equation as [1]

$$a_u(\lambda,\bar{\lambda}) = \iint_\Omega h[\sum_{i=1}^{2} F^i \bar{\lambda}^i] d\Omega + \int_\Gamma [\sum_{i=1}^{2} T^i \bar{\lambda}^i] d\Gamma \tag{6}$$

for all kinematically admissible virtual displacement $\bar{\lambda}$. Since Eq. (6) is identical to Eq. (4) if $\lambda = z$ and $\bar{\lambda} = \bar{z}$, the adjoint equation does not need to be solved. Using the adjoint variable method of design sensitivity analysis gives [1]

$$\psi_1' = \iint_\Omega \Big[\sum_{i=1}^{2} F^i z^i\Big]\delta h\, d\Omega + \iint_\Omega \Big[\sum_{i=1}^{2} F^i \lambda^i - \sum_{i,j=1}^{2} \sigma^{ij}(z)\varepsilon^{ij}(\lambda)\Big]\delta h\, d\Omega$$

$$= \iint_\Omega \Big[2\sum_{i=1}^{2} F^i z^i - \sum_{i,j=1}^{2} \sigma^{ij}(z)\varepsilon^{ij}(z)\Big]\delta h\, d\Omega \qquad (7)$$

since $z = \lambda$ for the compliance functional.

To numerically integrate Eqs. (4) and (7), a two-point Gauss quadrature formula is used. The equations become

$$\psi_1 = \sum_{k=1}^{N} \{h_k[\sum_{\ell=1}^{2} \sum_{i=1}^{2} F_\ell^i z_\ell^i]W_\ell J + \sum_{i=1}^{2} T^i z^i\} \qquad (8)$$

and

$$\psi_1' = \sum_{k=1}^{N} \{\sum_{\ell=1}^{2} [\sum_{i=1}^{2} 2F_\ell^i z_\ell^i - \sum_{i,j=1}^{2} \sigma_\ell^{ij}(z)\varepsilon_\ell^{ij}(z)W_\ell]\}J\,\delta h_k \qquad (9)$$

respectively, where J is the Jacobian, N is the total number of elements, subscript ℓ is the counter for the number of Gauss points, subscript k is the counter for the element number, W is the weighting constant for the ℓth Gauss point, and supercript i is the direction of the force and the displacement.

Next consider the functional representing the displacement z at a discrete point $\hat{x}$ as

$$\psi_2 \equiv z(\hat{x}) = \iint_\Omega \hat{\delta}(x - \hat{x})z(x)d\Omega \qquad (10)$$

where $\hat{\delta}(x)$ is the Dirac measure in the plane, acting at the origin. The first variation of Eq. (10) is

$$\psi_2' = \iint_\Omega \hat{\delta}(x - \hat{x}) z'(x) d\Omega \tag{11}$$

The adjoint equation in this case is [1]

$$a_u(\lambda, \bar{\lambda}) = \iint_\Omega \hat{\delta}(x - \hat{x}) \bar{\lambda}(x) d\Omega \tag{12}$$

for all kinematically admissible virtual displacement $\bar{\lambda}$. This equation has a unique solution $\lambda^{(2)}$, where $\lambda^{(2)}$ is the plate displacement due to a unit point load acting at a point $\hat{x}$. Using the adjoint variable method of design sensitivity analysis gives

$$\psi_2' = \iint_\Omega \left[\sum_{i=1}^{2} F^i \lambda^{(2)^i} - \sum_{i,j=1}^{2} \sigma^{ij}(z) \varepsilon^{ij}(\lambda^{(2)}) \right] \delta h \, d\Omega \tag{13}$$

where $\lambda^{(2)}$ is the solution of Eq. (12).

For this constraint two equations must be solved. The adjoint load of Eq. (12) is just a unit load applied at a discrete point in the finite element model. All that is necessary is a restart of the model so that load cases of applied unit loads at various nodal points can be analyzed. The resulting strains due to the adjoint load are then used in calculating Eq. (13). Note that for each displacement constraint there is a different adjoint load.

With numerical techniques applied as in the compliance constraint case, Eq. (13) becomes

$$\psi_2' = \sum_{k=1}^{N} \left\{ \sum_{\ell=1}^{2} \left[\sum_{i=1}^{2} F_\ell^i \lambda_\ell^{(2)^i} - \sum_{i,j=1}^{2} \sigma_\ell^{ij}(z)\varepsilon_\ell^{ij}(\lambda^{(2)}) \right] W_\ell \right\} J \; \delta h_k \tag{14}$$

Finally consider the general functional representing a locally averaged stress on an element as

$$\psi_3 = \iint_\Omega g\big(\sigma(z)\big) m_p \, d\Omega \tag{15}$$

where m_p is a characteristic function defined on a finite element Ω_p as

$$m_p = \begin{cases} \dfrac{1}{\int_{\Omega_p} d\Omega} , & x \in \Omega_p \\ 0 , & x \notin \Omega_p \end{cases} \tag{16}$$

The first variation of Eq. (15) is

$$\psi_3' = \iint_\Omega \left[\sum_{i,j=1}^{2} \frac{\partial g}{\partial \sigma^{ij}} \sigma^{ij}(z') \right] m_p \, d\Omega \tag{17}$$

Replacing the variation in state z' by a virtual displacement $\bar{\lambda}$, the adjoint equation is obtained as [1]

$$a_u(\lambda,\bar{\lambda}) = \iint_\Omega \left[\sum_{i,j=1}^{2} \frac{\partial g}{\partial \sigma^{ij}} \sigma^{ij}(\bar{\lambda}) \right] m_p \, d\Omega \tag{18}$$

for all kinematically admissible virtual displacements $\bar{\lambda}$. Eq. (18) has a unique solution for a displacement field $\lambda^{(3)}$. Using the adjoint variable method of design sensitivity analysis gives

$$\psi_3' = \iint_\Omega \left[\sum_{i=1}^{2} F^i \lambda^{(3)^i} - \sum_{i,j=1}^{2} \sigma^{ij}(z)\varepsilon^{ij}(\lambda^{(3)}) \right] \delta h \, d\Omega \tag{19}$$

where $\lambda^{(3)}$ is the solution of Eq. (18).

With numerical techniques applied as in the displacement constraint case, Eqs. (15), (18), and (19) become

$$\psi_3 = \sum_{\ell=1}^{2} [g(\sigma_\ell(z))m_p \; W_\ell]J \tag{20}$$

$$\iint_\Omega [\sum_{i,j=1}^{2} \frac{\partial g}{\partial \sigma^{ij}} \sigma^{ij}(\bar{\lambda})]m_p \; d\Omega$$

$$= \sum_{\ell=1}^{2} [\sum_{i,j=1}^{2} \frac{\partial g}{\partial \sigma^{ij}} \sigma^{ij}(\bar{\lambda})]m_p \; W_\ell \; J \; \bar{d} \tag{21}$$

and

$$\psi_3' = \sum_{k=1}^{N} \{\sum_{\ell=1}^{2} \sum_{i=1}^{2} F_\ell^i \lambda_\ell^{(3)^i} - \sum_{i,j=1}^{2} \sigma_\ell^{ij}(z)\varepsilon_\ell^{ij}(\lambda^{(3)})]W_\ell\}J \; \delta h_k \tag{22}$$

respectively. The numerical calculation of the adjoint load is considerably more difficult in the stress functional case. The shape function of the element must be known so that $\sigma^{ij}(\bar{\lambda})$ can be calculated using the finite element technique [2]

$$\sigma = [E][B]d = [E]\varepsilon \tag{23}$$

where [E] is the elasticity matrix, [B] is the strain-displacement matrix, which relates to the element shape function, d is the displacement vector, and ε is the strain vector.

For a plane stress problem,

$$[E] = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \tag{24}$$

The adjoint load becomes

$$\iint_{\Omega} \left[\sum_{i,j=1}^{2} \frac{\partial g}{\partial \sigma^{ij}} \sigma^{ij}(\bar{\lambda}) \right] m_p \, d\Omega \qquad (25)$$

where F is the adjoint equivalent nodal force. Since m_p is a characteristic function on the finite element Ω_p, F acts only on the nodal points of element Ω_p.

After the adjoint load is calculated for various elements, a restart of the finite element model for each load case corresponding to each element adjoint load is made. The strains resulting from these adjoint loads are then used in calculating Eq. (22) for the sensitivity of each functional.

When principal stress is selected as the functional,

$$g_1 = (\sigma^{11} + \sigma^{22})/2 + \tau_{max} \qquad (26)$$

$$\tau_{max} = \{[(\sigma^{11} - \sigma^{22}/2]^2 + (\sigma^{12})^2\}^{1/2} \qquad (27)$$

and

$$\begin{aligned} \partial g_1/\partial\sigma^{11} &= \frac{1}{2} + \frac{1}{4}(\sigma^{11} - \sigma^{22})/\tau_{max} \\ \partial g_1/\partial\sigma^{22} &= \frac{1}{2} - \frac{1}{4}(\sigma^{11} - \sigma^{22})/\tau_{max} \\ \partial g_1/\partial\sigma^{12} &= \sigma^{12}/\tau_{max} \end{aligned} \qquad (28)$$

and when von Mises' stress is selected as the functional,

$$g_2 = [(\sigma^{11})^2 - \sigma^{11}\sigma^{22} + (\sigma^{22})^2 + 3(\sigma^{12})^2]^{1/2} \qquad (29)$$

and

$$\partial g_2/\partial\sigma_{11} = \frac{1}{2}\,(2\sigma^{11} - \sigma^{22})/g_2$$

$$\partial g_2/\partial\sigma^{22} = \frac{1}{2}\,(2\sigma^{22} - \sigma^{11})/g_2 \qquad (30)$$

$$\partial g_2/\partial\sigma^{12} = 3\sigma^{12}/g_2$$

2.1.2 Bending of Beams

Consider a cantilever beam with variable width and height and self weight, as shown in Fig. 4. The width and height are the design variables, $u = [b(x),h(x)]^T$.

The energy bilinear form and the load linear form of the beam are

$$a_u(z,\bar{z}) = \int_0^L E\,\frac{bh^3}{12}\,z_{xx}\bar{z}_{xx}\,dx \qquad (31)$$

and

$$\ell_u(\bar{z}) = -\int_0^L (F + \gamma\,bh)\bar{z}\,dx \qquad (32)$$

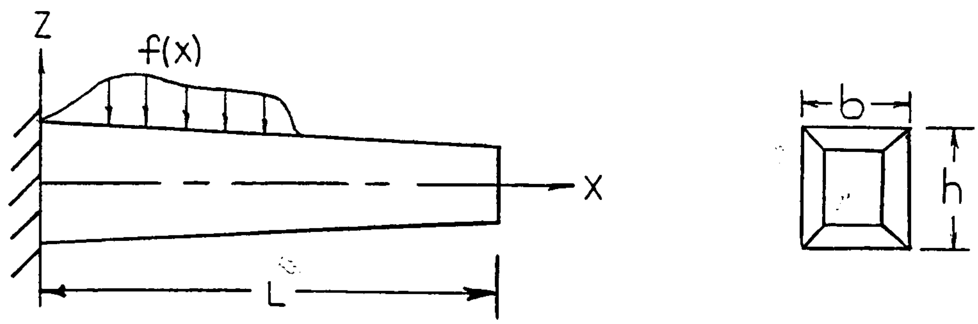


Figure 4. Cantilever Beam with Variable Width and Height

where γ is the weight density of the beam material, F is the distributed load, E is the modulus of elasticity of the beam material, $bh^3/12$ is the moment of inertia, $\bar{z}$ is the virtual displacement, z_{xx} is the beam curvature, and $\bar{z}_{xx}$ is the beam curvature due to the virtual displacement $\bar{z}$.

The negative sign in the load linear equation is due to the fact that the load is applied in the $-z$ direction.

The state equation is [1]

$$a_u(z,\bar{z}) = \ell_u(\bar{z}) \tag{33}$$

for all kinematically admissible virtual displacements $\bar{z}$.

First consider the functional representing the compliance of the structure as

$$\psi_4 = - \int_0^L (F + \gamma bh)z \, dx \tag{34}$$

The first variation of Eq. (34) is

$$\psi_4' = - \int_0^L (F + \gamma bh)z' \, dx - \int_0^L - h\gamma z \, dx \, \delta b - \int_0^L b\gamma z \, dx \, \delta h \tag{35}$$

To replace the variation in state z' by a virtual displacement $\bar{\lambda}$, the adjoint equation is defined as [1]

$$a_u(\lambda,\bar{\lambda}) = - \int_0^L (F + \gamma bh)\bar{\lambda} \, dx \tag{36}$$

for all kinematically admissible virtual displacements $\bar{\lambda}$. Since Eq. (36) is identical to Eq. (34), if $\bar{\lambda} = z$ the adjoint equation does not need to be solved. Using the adjoint variable method of design sensitivity analysis gives

$$\psi_4' = \int_0^L [-2\gamma hz - (Eh^3/12)(z_{xx})^2]dx \ \delta b$$
$$+ \int_0^L [-2\gamma bz - (3Ebh^2/12)(z_{xx})^2]dx \ \delta h \tag{37}$$

To numerically integrate Eqs. (34) and (37), a three-point Gauss quadrature formula is used. These equations become

$$\psi_4 = \sum_{k=1}^{N} \{ \sum_{\ell=1}^{3} [F_\ell + \gamma b_k h_k] z_\ell W_\ell \tag{38}$$

and

$$\psi_4' = \sum_{k=1}^{N} \{ \sum_{\ell=1}^{3} [-2\gamma h_k z_\ell - (Eh_k^2/12)(z_{xx_\ell})^2] W_\ell \} J \ \delta b_k$$
$$+ \sum_{k=1}^{3} \{ \sum_{\ell=1}^{3} [-2\gamma b_k z_\ell - (3E \ b_k h_k^2/12)(z_{xx_\ell})^2 \ W_\ell \} J \ \delta h_k \tag{39}$$

where N is the total number of elements, ℓ is the Gauss point counter, W is the weighting constant for the ℓth Gauss point and J is the Jacobian. The beam curvature z_{xx} is calculated using a cubic polynomial for the standard beam shape functions. Because the load is in the -z direction, it is necessary to change the sign of the local element y-rotation θ_y.

Next consider the functional representing the displacement z at a discrete point $\hat{x}$ as

$$\psi_5 \equiv z(\hat{x}) = \int_0^L \hat{\delta}(x - \hat{x}) z(x) dx \tag{40}$$

where $\hat{\delta}(x)$ is the Dirac measure at zero. The first variation of Eq. (40) is

$$\psi_5' = \int_0^L \hat{\delta}(x - \hat{x})z'(x)dx \tag{41}$$

The adjoint equation is defined as [1]

$$a_u(\lambda,\bar{\lambda}) = \int_0^L \hat{\delta}(x - \hat{x})\bar{\lambda}(x)dx \tag{42}$$

for all kinematically admissible virtual displacements $\bar{\lambda}$. Equation (42) has a unique solution $\lambda^{(5)}$, where $\lambda^{(5)}$ is the beam displacement due to a unit point load acting at a point $\hat{x}$. Using the adjoint variable method of design sensitivity analysis gives

$$\begin{aligned} \psi_5' = {} & \int_0^L [-h\gamma\lambda^{(5)} - (Eh^3/12)z_{xx}\lambda^{(5)}_{xx}]dx \ \delta b \\ & + \int_0^L [-b\gamma\lambda^{(5)} - (3Ebh^2/12)z_{xx}\lambda^{(5)}_{xx}]dx \ \delta h \end{aligned} \tag{43}$$

As in the membrane displacement constraint case, only Eq.(43) needs to be solved numerically. Using the three-point Gauss quadrature technique, Eq. (43) becomes

$$\begin{aligned} \psi_5' = {} & \sum_{k=1}^{N} \{\sum_{\ell=1}^{3} [-h_k\gamma\lambda^{(5)}_\ell - (Eh_k^3/12)z_{xx_\ell}\lambda^{(5)}_{xx_\ell}]W_\ell\}J \ \delta b_k \\ & + \sum_{k=1}^{N} \{\sum_{\ell=1}^{3} [=b_k\gamma\lambda^{(5)}_\ell - (3Eb_kh_k^2/12)z_{xx_\ell}\lambda^{(5)}_{xx_\ell}]W_\ell\}J \ \delta h_k \end{aligned} \tag{44}$$

Finally consider the functional representing the allowable stresses in the beam as

$$\psi_6 = \int_0^L - \frac{1}{2} hEz_{xx}m_p dx \tag{45}$$

where h/2 is the half-depth of the beam, and m_p is a characteristic function defined on a finite element dx_p as

$$m_p = \begin{cases} \dfrac{1}{\int_{dx_p} dx}, & x \in dx_p \\ 0, & x \notin dx_p \end{cases} \tag{46}$$

The first variation of Eq. (45) is

$$\psi_6' = \int_0^L \left[-\frac{1}{2} hE\ z'_{xx} m_p - \frac{1}{2} E\ z_{xx} m_p \delta h\right] dx \tag{47}$$

Replacing the variation in state z' by a virtual displacement $\bar{\lambda}$, the adjoint equation is defined as

$$a_u(\lambda, \bar{\lambda}) = -\int_0^L \frac{1}{2} hE\bar{\lambda}_{xx} m_p dx \tag{48}$$

for all kinematically admissible virtual displacements $\bar{\lambda}$. Equation (48) has a unique solution for a displacement field $\lambda^{(6)}$. Using the adjoint variable method of design sensitivity analysis gives

$$\begin{aligned} \psi_6' = & \int_0^L \left[-h\gamma\lambda^{(6)} - (Eh^3/12) z_{xx} \lambda^{(6)}_{xx}\right] dx\ \delta b \\ & + \int_0^L \left[-\frac{1}{2} Ez_{xx} m_p - b\gamma\lambda^{(6)} - (3Ebh^2/12) z_{xx} \lambda^{(6)}_{xx}\right] dx\ \delta h \end{aligned} \tag{49}$$

where $\lambda^{(6)}$ is the solution of Eq. (48).

With the three-point Gauss quadrature numerical integration technique, the integrals in Eqs. (45), (48), and (49) become

$$\psi_6 = \sum_{k=1}^{N} \left\{ \sum_{\ell=1}^{3} \left(-\frac{1}{2} h_k E\ z_{xx_\ell} m_p\right) W_\ell \right\} J \tag{50}$$

$$\int_0^L \frac{1}{2} E \, \bar{\lambda}_{xx} \, m_p \, dx$$

$$= \sum_{k=1}^{N} \{ \sum_{\ell=1}^{3} (-\frac{1}{2} E[B]_\ell \, m_p) W_\ell \} J \, \bar{d} \tag{51}$$

and

$$\psi_6' = \sum_{k=1}^{N} \{ \sum_{\ell=1}^{3} [-h_k \gamma \lambda_\ell^{(6)} - (E h_k^3/12) z_{xx_\ell} \lambda_{xx_\ell}^{(6)}] W_\ell \} J \, \delta b_k$$

$$+ \sum_{k=1}^{N} \{ \sum_{\ell=1}^{3} [-\frac{1}{2} E \, z_{xx_\ell} m_p - b_k \gamma \lambda_\ell^{(6)} \tag{52}$$

$$- (3E \, b_k h_k^2/12) z_{xx_\ell} \lambda_{xx_\ell}^{(6)}] W_\ell \} J \, \delta h_k$$

where [B] is the strain-displacement matrix, which is the second derivative of the shape functions.

2.1.3 Bending of Plates

Consider the clamped plate in Fig. 5 of variable thickness $u = t(x)$, with a distributed load $f(x)$ that consists of an externally applied pressure $F(x)$ and self weight, given by [1]

$$f(x) = F(x) + \gamma t(x) \tag{53}$$

where γ is the weight density of the material.

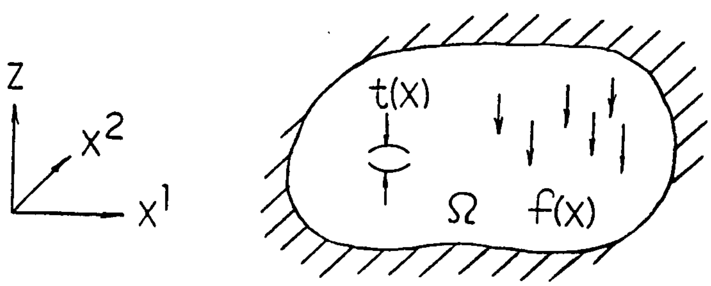


Figure 5. Clamped plate of variable thickness t(x)

For this design dependent loading, the energy bilinear form and the load linear form for the plate are given as [1]

$$a_u(z,\bar{z}) = \iint_\Omega \hat{D}(u)[z_{11}\bar{z}_{11} + z_{22}\bar{z}_{22} + \nu(z_{22}\bar{z}_{11} + z_{11}\bar{z}_{22})$$

$$+ 2(1 - \nu)z_{12}\bar{z}_{12}]d\Omega \tag{54}$$

and

$$\ell_u(\bar{z}) = \iint_\Omega [F + \gamma t]\bar{z}\, d\Omega \tag{55}$$

where $\hat{D}(u) = Et^3/[12(1-\nu^2)]$ is the flexural rigidity, E is Young's modulus, ν is Poisson's ratio, F is the externally applied pressure, and γ is the material density. The state equation is [1]

$$a_u(z,\bar{z}) = \ell_u(\bar{z}) \tag{56}$$

for all kinematically admissible virtual displacements $\bar{z}$.

First consider the functional representing the compliance of the structure as

$$\psi_7 = \iint_\Omega (F + \gamma t)z\, d\Omega \tag{57}$$

The first variation of Eq. (57) is

$$\psi_7' = \iint_\Omega [(F + \gamma t)z' + \gamma z\, \delta t]d\Omega \tag{58}$$

The adjoint equation is defined as

$$a_u(\lambda,\bar{\lambda}) = \iint_\Omega (F + \gamma t)\bar{\lambda}\, d\Omega \tag{59}$$

for all kinematically admissible displacements $\bar{\lambda}$. As in the previous cases (membranes and bending beams), Eq. (59) is identical to Eq. (57)

if $\bar{\lambda} = z$. Using the adjoint variable method of design sensitivity analysis gives

$$\psi_7' = \iint_\Omega \{2\gamma z - Et^2[z_{11}^2 z_{22}^2 + 2\nu z_{11} z_{22} + 2(1-\nu) z_{12}^2]/4\ (1-\nu^2)\}\delta t\ d\Omega$$

$$= \iint_\Omega [2\gamma z - \sum_{i,j=1}^{2} \sigma^{ij}(z)\varepsilon^{ij}(z)]\delta t\ d\Omega \qquad (60)$$

where $\sigma^{ij}(z)$ and $\varepsilon^{ij}(z)$ are the stress and strain of the extreme fiber, given as

$$\varepsilon^{ij} = -\frac{tz_{ij}}{2}\ , \qquad i,j=1,2 \qquad (61)$$

and

$$\sigma^{11} = -\frac{Et}{2(1-\nu^2)}(z_{11} + \nu z_{22})$$

$$\sigma^{22} = -\frac{Et}{2(1-\nu^2)}(z_{22} + \nu z_{11}) \qquad (62)$$

$$\sigma^{12} = -\frac{Et}{2(1+\nu)} z_{12}$$

To numerically integrate Eqs. (57) and (60), a one-point Gauss quadrature formula on a triangular element is used to correspond to the IFAD integration technique for this element. Equations (57) and (60) become

$$\psi_7 = \sum_{k=1}^{N} \{(F + \gamma t_k)z\ W\}J \qquad (63)$$

and

$$\psi_7' = \sum_{k=1}^{N} [2\gamma z - \sum_{i,j=1}^{2} \sigma^{ij}(z)\varepsilon^{ij}(z)]W\ J\ \delta t_k \qquad (64)$$

Next consider the functional representing the displacement z at a discrete point $\hat{x}$ as

$$\psi_8 \equiv z(\hat{x}) = \iint_\Omega \hat{\delta}(x - \hat{x})z(x)d\Omega \tag{65}$$

The first variation of Eq. (65) is

$$\psi_8' = \iint_\Omega \hat{\delta}(x - \hat{x})z'(x)dx \tag{66}$$

The adjoint equation is defined as

$$a_u(\lambda,\bar{\lambda}) = \iint_\Omega \hat{\delta}(x - \hat{x})\bar{\lambda}(x)d\Omega \tag{67}$$

for all kinematically admissible virtual displacements $\bar{\lambda}$. This equation has a unique solution $\lambda^{(8)}$, where $\lambda^{(8)}$ is the plate displacement due to a unit vertical load acting at a point $\hat{x}$. Using the adjoint variable method of design sensitivity analysis gives

$$\psi_8' = \iint_\Omega \{\gamma\lambda^{(8)} - Et^2[z_{11}\lambda_{11}^{(8)} + z_{22}\lambda_{22}^{(8)} + \nu(z_{11}\lambda_{22}^{(8)} + z_{22}\lambda_{11}^{(8)})$$

$$+ 2(1 - \nu)z_{12}\lambda_{12}^{(8)}]/4(1 - \nu^2)\}\delta t\, d\Omega$$

$$= \iint_\Omega [\gamma\lambda^{(8)} - \sum_{i,j=1}^{2} \sigma^{ij}(z)\varepsilon^{ij}(\lambda^{(8)})]\delta t\, d\Omega \tag{68}$$

The same numerical integration procedure is used as in the compliance case. Equation (68) becomes

$$\psi_8' = \sum_{k=1}^{N} [\gamma\lambda^{(8)} - \sum_{i,j=1}^{2} \sigma^{ij}(z)\varepsilon^{ij}(\lambda^{(8)})]W\, J\, \delta t_k \tag{69}$$

Finally, consider the functional representing a locally averaged stress in the plate as

$$\psi_9 = \iint_\Omega g(\sigma(z)) m_p \, d\Omega \tag{70}$$

where $g(\sigma(z))$ may be principal stress, von Mises' stress, or some other material failure criteria and m_p is a characteristic function defined on a finite element Ω_p as

$$m_p = \begin{cases} \dfrac{1}{\int_{\Omega p} d\Omega} , & x \in \Omega_p \\ 0 , & x \notin \Omega_p \end{cases} \tag{71}$$

The first variation of Eq. (70) is

$$\psi_9' = \iint_\Omega \left[\sum_{i,j=1}^{2} \frac{\partial g}{\partial \sigma^{ij}} \sigma^{ij}(z') \right] m_p \, d\Omega \tag{72}$$

The adjoint equation is defined as [1]

$$a_u(\lambda,\bar{\lambda}) = \iint_\Omega \left[\sum_{i,j=1}^{2} \frac{\partial g}{\partial \sigma^{ij}} \sigma^{ij}(\bar{\lambda}) \right] m_p \, d\Omega \tag{73}$$

for all kinematically admissible virtual displacement $\bar{\lambda}$. Using the adjoint variable method of design sensitivity analysis gives

$$\psi_9' = \iint_\Omega \left[\gamma\lambda^{(9)} - \sum_{i,j=1}^{2} \sigma^{ij}(z)\varepsilon^{ij}(\lambda^{(9)}) \right] \delta t \, d\Omega + \iint_\Omega \frac{\partial g}{\partial t} \delta t \, d\Omega \tag{74}$$

where $\lambda^{(9)}$ is the solution of Eq. (73). For principal stress the last term on the right of Eq. (74) becomes

$$\iint_\Omega \left[(\sigma^{11} + \sigma^{22} + 2\tau_{max})/(2t)\right]m_p \, d\Omega \tag{75}$$

where τ_{max} is defined in Eq. (27). For von Mises' stress, this term becomes

$$\iint_\Omega \frac{1}{t}\left[(\sigma^{11})^2 - \sigma^{11}\sigma^{22} + (\sigma^{22})^2 + 3(\sigma^{12})^2\right]^{1/2} m_p \,\delta t \, d\Omega \tag{76}$$

The IFAD thin shell element is the element that is used for plate bending. The membrane effects can be eliminated so that only the bending term exists. This element is a hybrid element which uses a derivative smoothing technique [4]. The IFAD code evaluates the normal and shear stresses at the centroid of the triangle, but Ref. 4 stipulates that the stresses at the midside nodes of the triangle give the most accurate results. The integration of a function ϕ from its mid-stresses is [4]

$$\int \phi dA = \frac{A}{3}\left[\phi(0, \tfrac{1}{2}, \tfrac{1}{2}) + \phi(\tfrac{1}{2}, \tfrac{1}{2}, 0) + \phi(\tfrac{1}{2}, \tfrac{1}{2}, 0)\right] \tag{77}$$

where A is the area of the triangle and $(0, \frac{1}{2}, \frac{1}{2})$, $(\frac{1}{2}, 0, \frac{1}{2})$ and $(\frac{1}{2}, \frac{1}{2}, 0)$ are the area coordinates of the mid-side nodes. To achieve the most accurate atresses possible, this numerical integration technique is applied on Eqs. (70), (73), and (74). These equations become

$$\psi_9 = \sum_{n=1}^{3} g(\sigma_n(z))m_p \frac{A}{3} \tag{78}$$

$$\iint_\Omega \left[\sum_{i,j=1}^{2} \frac{\partial g}{\partial \sigma^{ij}} \sigma^{ij}(\bar{\lambda})\right]m_p \, d\Omega = \sum_{n=1}^{3}\left[\frac{\partial g}{\partial \sigma_n^{11}}, \frac{\partial g}{\partial \sigma_n^{22}}, \frac{\partial g}{\partial \sigma_n^{12}}\right]$$

$$[E][B]_n \, m_p \, \frac{A}{3} \, \bar{d} = F^T \bar{d} \tag{79}$$

and

$$\psi_9' = \sum_{k=1}^{N} \left\{ \sum_{n=1}^{3} \left[\gamma \lambda_n^{(9)} - \sum_{i,j=1}^{2} \sigma_n^{ij}(z) \varepsilon_n^{ij}(\lambda^{(9)}) + \left(\frac{\partial g}{\partial t} \right)_n \right] \frac{A}{3} \right\} \delta t_k \tag{80}$$

where subscript n is the counter for the midside value. The term $(\partial g/\partial t)_n$ for principal and von Mises' stress is

$$[(\sigma_n^{11} + \sigma_n^{22} + \partial \tau_{max_n})/(2t_k)]m_p \tag{81}$$

and

$$\frac{1}{t_k} [(\sigma_n^{11})^2 - \sigma_n^{11} \sigma_n^{22} + (\sigma_n^{22})^2 + 3(\sigma_n^{12})^2]^{1/2} \, m_p \tag{82}$$

respectively.

To numerically calculate the adjoint load, the shape function of the element must be known so that $\sigma^{ij}(\bar{\lambda})$ can be calculated using the finite element technique described in Section 2.1.1, Eqs. (23) and (25).

2.2 Calculation Procedure for a Built-Up Structure

The foundation of the built-up structure design sensitivity analysis method is the structural component analysis developed in Section 2.1. A built-up structure consists of various structural components that interact with each other. This interaction is taken into account by generalizing the individual components such that twisting, bending, transverse shear terms, etc. are included in the formulation of the sensitivity vector. Coordinate system precautions need to be taken to insure that the constraints and the sensitivity vectors are calculated correctly. In general, if the calculations are performed at the local element coordinate system level, there will be no problem when components are oriented differently in the global coordinate system.

An example of a built-up structure using the structural components developed in the previous section would be a bending beam and plate problem, where a framework of beams could act as the supporting structure for the plate; i.e., a roof structure.

Figure 6 shows a built-up structure that has design variables $u=[b(x),h(x),t(x)]^T$, where b(x) is the width of the beams, h(x) is the heigth of the beams, and t(x) is the thickness of the plates.

It is assumed that the plates are welded along the length of the beams. This would infer that the appropriate components in the finite element analysis must be chosen to insure kinematic compatibility along the component boundaries.

The energy bilinear form of the system equation is just the sum of the plate and beam energy bilinear equations, with an additional beam torsion term given as

$$a_u(z,\bar{z}) = \sum \iint_\Omega \hat{D}(u)[z_{xx}\bar{z}_{xx} + z_{yy}\bar{z}_{yy} + \sum(z_{yy}\bar{z}_{xx} + z_{xx}\bar{z}_{yy})$$

$$+ 2(1 - \nu)z_{xy}\bar{z}_{xy}]d\Omega + \sum \int_0^L E_b \frac{bh^3}{12} z_{xx}\bar{z}_{xx}dx_\ell$$

$$+ \sum \int_0^L GJ\ z_{xy}\bar{z}_{xy}dx_\ell$$

$$= \sum a_u(z,\bar{z})_{plate} + \sum a_u(z,\bar{z})_{beam} + \sum \int_0^L GJ\ z_{xy}\ \bar{z}_{xy}\ dx_\ell \tag{83}$$

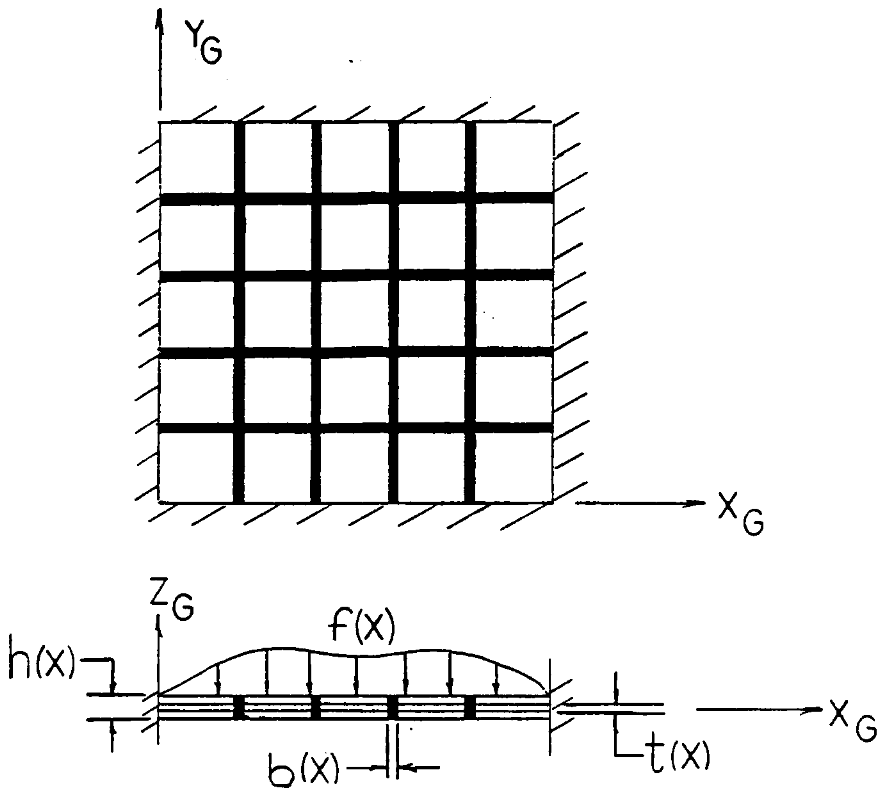


Figure 6. Roof Structure

where $a_u(z,z)_{plate}$ and $a_u(z,z)_{beam}$ are the energy bilinear forms of the plate, Eq. (54) and beam, Eq. (31), respectively, G is the modulus of rigidity, J is the torsional moment of inertia of the beam, and ℓ represents the local beam coordinate system, where x_ℓ runs along the length of the beam.

In Eq. (83) z_{xy} represents the beam torsion term. Since each structural component needs to be solved individually to make the analysis feasible with an existing finite element code, a relationship between z_{xy} and the beam rotation has to be used. This kinematic compatibility states that for the beam and plate system the term has to be equivalent to the relative angle of twist over an element length.

That is to say,

$$z_{xy} = (\theta^2 x_\ell - \theta^1 x_\ell)/L \quad (84)$$

where $\theta^2 x_\ell$ is the local element rotation at node 2 of the beam, $\theta^1 x_\ell$ is the local element rotation at node 1 of the beam, and L is the beam element length.

The load linear form of the system is

$$\ell_u (\bar{z}) = \iint_\Omega [F_p + \gamma_p t + \gamma_b bh]\bar{z}\ d\Omega \quad (85)$$

where F_p is the externally applied plate pressure, γ_p and γ_b are the material densities for the plates and beams respectively, and $\bar{z}$ is the virtual displacement. The state equation is [1]

$$a_u(z,\bar{z}) = \ell_u(\bar{z}) \quad (86)$$

for all kinematically admissible virtual displacements $\bar{z}$.

Since the energy bilinear form of the system equation is just the addition of each structural component's energy bilinear forms, the design sensitivity equation of the system turns out also to be an additive process. The generalized design sensitivity of the built-up structure is

$$\psi' = \psi'_b\ \delta b + \psi'_h\ \delta h + \psi'_t\ \delta t \quad (87)$$

The only necessary step to calculating this value is the reformulation of the beam to include the torsional term. The energy bilinear form of the beam component becomes

$$a_u(z,\bar{z}) = \int_0^L E(bh^3/12) z_{xx}\bar{z}_{xx} dx + \int_0^L GJ\ z_{xy}^2 dx \quad (88)$$

where Eq. (84) defines z_{xy}.

If the compliance, displacement, and stress functionals of the beam - Eqs. (34), (40), and (45), respectively - remain the same, the only additional term in the design sensitivity analysis [1] is due to the differentiation of the torsional terms in the energy bilinear equation with respect to the design variables b and h. This term is defined as

$$\int_0^L \left[\frac{\partial J}{\partial h} \delta h + \frac{\partial J}{\partial b} \delta b \right] G \, z_{xy} \, \lambda_{xy} \, dx \tag{89}$$

For a beam with a rectangular cross section [5]

$$J = bh^3 \left[\frac{1}{3} - 0.21(b/h)\left(1 - \frac{b^4}{12h^4}\right)\right] \tag{90}$$

and the derivatives with respect to the design variables are

$$\frac{\partial J}{\partial b} = \frac{h^3}{3} - 0.042b \left(h^2 + \frac{b^4}{4h^2}\right) \tag{91}$$

and

$$\frac{\partial J}{\partial b} = bh^2 - 0.42 \, b^2 \left(h - \frac{b^4}{12h^3}\right) \tag{92}$$

The compliance sensitivity of Eq. (37) becomes

$$\begin{aligned} \psi_4' = {} & \int_0^L \left[-2\gamma hz - (Eh^3/12)(z_{xx})^2 - \frac{\partial J}{\partial b} G(z_{xy})^2\right]dx \; \delta b \\ & + \int_0^L \left[-2\gamma bz - (3Ebh^2/12)(z_{xx})^2 - \frac{\partial J}{\partial h} G(z_{xy})^2\right]dx \; \delta h \end{aligned} \tag{93}$$

the displacement sensitivity of Eq. (43) becomes

$$\psi_5' = \int_0^L \left[-h\gamma\lambda^{(5)} - \frac{Eh^3}{12} z_{xx}\lambda_{xx}^{(5)} - \frac{\partial J}{\partial b} G\, z_{xy}\lambda_{xy}^{(5)}\right]dx\ \delta b$$
$$+ \int_0^L \left[-2\gamma b\lambda^{(5)} - \frac{3Ebh^2}{12} z_{xx}\lambda_{xx}^{(5)} - \frac{\partial J}{\partial h} G\, z_{xy}\lambda_{xy}^{(5)}\right]dx\ \delta h \qquad (94)$$

and the stress sensitivity of Eq. (49) becomes

$$\psi_6' = \int_0^L \left[-h\gamma\lambda^{(6)} \frac{Eh^3}{12} z_{xx}\lambda_{xx}^{(6)} - \frac{\partial J}{\partial b} G\, z_{xy}\lambda_{xy}^{(6)}\right]dx\ \delta b$$
$$+ \int_0^L \left[-\frac{1}{2} E\, z_{xx}m_p - b\gamma\lambda^{(6)} - \frac{3Ebh^2}{12} z_{xx}\lambda_{xx}^{(6)} - \frac{\partial J}{\partial h} Gz_{xy}\lambda_{xy}^{(6)}\right]dx\,\delta h \qquad (95)$$

when the torsional term is added, where $\lambda^{(5)}$ and $\lambda^{(6)}$ are the solutions to the adjoint Eqs. (42) and (48), respectively.

Caution has to be taken when the constraint functionals on the system are defined. When a von Mises' stress functional is specified for a particular plate element, the allowable beam bending stress term $-\int_0^L \frac{1}{2} E\, z_{xx}m_p d_x \delta h$ must be removed from Eq. (95), so that the design sensitivity of Eq. (87) is calculated only for the von Mises' stress functional. In the same way, if an allowable beam bending stress functional is specified for a particular beam element, the von Mises' stress term $\partial g/\partial t$ of Eq. (76) must be removed from Eq. (74), so that the design sensitivity of Eq. (87) is calculated only for the allowable beam bending stress functional. The application of this procedure is shown below for a von Mises' stress functional, a compliance functional and a displacement functional on a plate element and an allowable bending stress functional on a beam element of the built-up structure of Fig. 6.

The functional representing a von Mises' stress constraint on a plate element is

$$\psi_{10} = \iint_{\Omega} [\sigma_{xx}^{2} - \sigma_{xx}\sigma_{yy} + \sigma_{yy}^{2} + 3\tau_{xy}^{2}]^{1/2}\, m_p \, d\Omega \tag{96}$$

The adjoint load linear form is defined as

$$\begin{aligned}\iint_{\Omega} [\sum_{i,j=1}^{2} \frac{\partial g}{\partial \sigma^{ij}} \sigma^{ij}(\bar{\lambda})] m_p \, d\Omega \\ &= [\frac{\partial g}{\partial \sigma_{xx}}, \frac{\partial g}{\partial \sigma_{yy}}, \frac{\partial g}{\partial \tau_{xy}}][E][B]\, m_p \, d\Omega\, \bar{d} \\ &= F^{T} \bar{d}\end{aligned} \tag{97}$$

where $\partial g/\partial \sigma^{ij}$ is defined in Eq. (30) and F is the adjoint equivalent nodal force. Using Eq. (87) gives the design sensitivity of the roof structure as

$$\begin{aligned}\psi_{10}' = &\sum \iint_{\Omega} [\gamma\lambda^{(10)} - \sum_{i,j=1}^{2} \sigma^{ij}(z)\varepsilon^{ij}(\lambda^{(10)}) + \frac{\partial g}{\partial t}]\delta t \, d\Omega \\ &+ \sum \int_{0}^{L} [-h\gamma\lambda^{(10)} - \frac{Eh^{3}}{12} z_{xx}\lambda_{xx}^{(10)} - \frac{\partial J}{\partial b} G\, z_{xy}\, \lambda_{xy}^{(10)}]dx_{\ell}\, \delta b \\ &+ \sum \int_{0}^{L} [-b\gamma\lambda^{(10)} - \frac{3Ebh^{2}}{12} z_{xx}\lambda_{xx}^{(10)} - \frac{\partial J}{\partial h} G\, z_{xy}\, \lambda_{xy}^{(10)}]dx_{\ell}\, \delta h\end{aligned} \tag{98}$$

where subscript ℓ refers to the beam local coordinate system and $\partial g/\partial t$ is defined in Eq. (76).

The functional representing a compliance constraint on a plate element is

$$\psi_{11} = \iint_{\Omega} (F + \gamma t + \gamma bh) z \, d\Omega \tag{99}$$

where γt and γbh are the self weight of the plate and beam, respectively. The adjoint load linear form is defined as

$$a_u(\lambda, \bar{\lambda}) = \iint_{\Omega} (F + \gamma t + \gamma bh) \bar{\lambda} \, d\Omega \tag{100}$$

Since Eq. (100) is identical to Eq. (99) if $\bar{\lambda} = z$, the adjoint equation does not need to be solved. Using Eq. (87) gives the design sensitivity of the built-up structure as

$$\begin{aligned} \psi_{11}' = & \sum \iint_{\Omega} [2\gamma z - \sum_{i,j=1}^{2} \sigma^{ij}(z) \varepsilon^{ij}(\bar{z})] \delta t \, d\Omega \\ & + \sum \int_0^L [-2\gamma hz - Eh^3/12(z_{xx})^2 - \frac{\partial J}{\partial b} G(z_{xy})^2] dx_\ell \, \delta b \\ & + \sum \int_0^L [-2\gamma bz - (3Ebh^2/12)(z_{xx})^2 - \frac{\partial J}{\partial h} G(z_{xy})^2] dx_\ell \, \delta h \end{aligned} \tag{101}$$

where subscript ℓ refers to the beam local coordinate system.

The functional representing the displacement z at a discrete point $\hat{x}$ is

$$\psi_{12} \equiv z(\hat{x}) = \iint_{\Omega} \hat{\delta}(x - \hat{x}) z(x) d\Omega \tag{102}$$

The adjoint load linear form is defined as

$$a_u(\lambda, \bar{\lambda}) = \iint_{\Omega} \hat{\delta}(x - \hat{x}) \bar{\lambda}(x) d\Omega \tag{103}$$

for all kinematically admissible virtual displacements $\bar{\lambda}$. This equation has a unique solution $\lambda^{(12)}$, where $\lambda^{(12)}$ is the plate displacement due to a unit vertical load acting at a point $\hat{x}$. Using Eq. (87) gives the design sensitivity of the built-up structure as

$$\psi'_{12} = \sum \iint_\Omega [\gamma\lambda^{(12)} - \sum_{i,j=1}^{2} \sigma^{ij}(z)\varepsilon^{ij}(\lambda^{(12)})]$$

$$+ \sum \int_0^L [-h\gamma\lambda^{(12)} - \frac{Eh^3}{12} z_{xx}\lambda_{xx}^{(12)} - \frac{\partial J}{\partial b} Gz_{xy}\lambda_{xy}^{(12)}]dx_\ell \; \delta b$$

$$+ \sum \int_0^L [-b\gamma\lambda^{(12)} - \frac{3Ebh^2}{12} z_{xx}\lambda_{xx}^{(12)} - \frac{\partial J}{\partial h} Gz_{xy}\lambda_{xy}^{(12)}]dx_\ell \; \delta h \tag{104}$$

where subscript ℓ refers to the beam local coordinate system.

The functional representing an allowable bending stress on a beam element is

$$\psi_{13} = - \int_0^L \frac{1}{2} hEz_{xx} \, m_p \, dx \tag{105}$$

The adjoint load linear form is defined as

$$\int_0^L \frac{1}{2} hE\bar{\lambda}_{xx} \, m_p \, dx = \int_0^L \frac{1}{2} hE[B]m_p \, dx \, \bar{d} = F^T d \tag{106}$$

where F is the adjoint equivalent nodal force. Using Eq. (87) gives the design sensitivity of the built-up structures as

$$\psi'_{13} = \sum \iint_\Omega [\gamma\lambda^{(13)} - \sum_{i,j=1}^{2} \sigma^{ij}(z)\varepsilon^{ij}(\lambda^{(13)})]\delta t \; d\Omega$$

$$+ \sum \int_0^L [-h\gamma\lambda^{(13)} - \frac{Eh^3}{12} z_{xx}\lambda_{xx}^{(13)} - \frac{\partial J}{\partial b} G \, z_{xy}\lambda_{xy}^{(13)}]dx_\ell \; \delta b$$

$$+ \sum \int_0^L \left[-b\gamma\lambda^{(13)} - \frac{1}{2} Ez_{xx}m_p - \frac{3Ebh^2}{12} z_{xx}\lambda_{xx}^{(13)} \right.$$

$$\left. - \frac{\partial J}{\partial h} G \; z_{xy}\lambda_{xy}^{(13)} \right] dx_{\ell} \; \delta h \qquad (107)$$

where subscript ℓ refers to the beam local coordinate system.

CHAPTER III

NUMERICAL EXAMPLES

The design sensitivity of a constraint functional ψ is the differential ψ' of the constraint functional. In order to make sure the design sensitivity is accurate, an approximation $\Delta\psi$ is made using the finite difference method. It is very important that an appropriate perturbed design variable δu is selected. If δu is too small, the change in the constraint functional $\Delta\psi$ may be inaccurate due to losses of significant digits. If δu is too large, $\Delta\psi$ will be influenced by nonlinearities in the constraint functional, which in turn will cause an inaccurate design sensitivity prediction.

In all the following examples, perturbations in design of 1% and/or 5% are used, with the exception of the plate bending and built up structure examples. Because of the nonlinearity characteristics of built-up structural response, the perturbation of 0.1% was used.

3.1 Membranes

The finite element membrane model in Fig. 7 is a simple plane elastic solid that is restrained at one end and loaded with a distributed tensile load at the other end. It contains 80 isoparametric elements (IFAD plane stress element type 1104), 289 nodal points, and 560 degrees-of-freedom, with the design variable being the variable thickness $u = h(x)$. The material property constraints, Young's modulus,

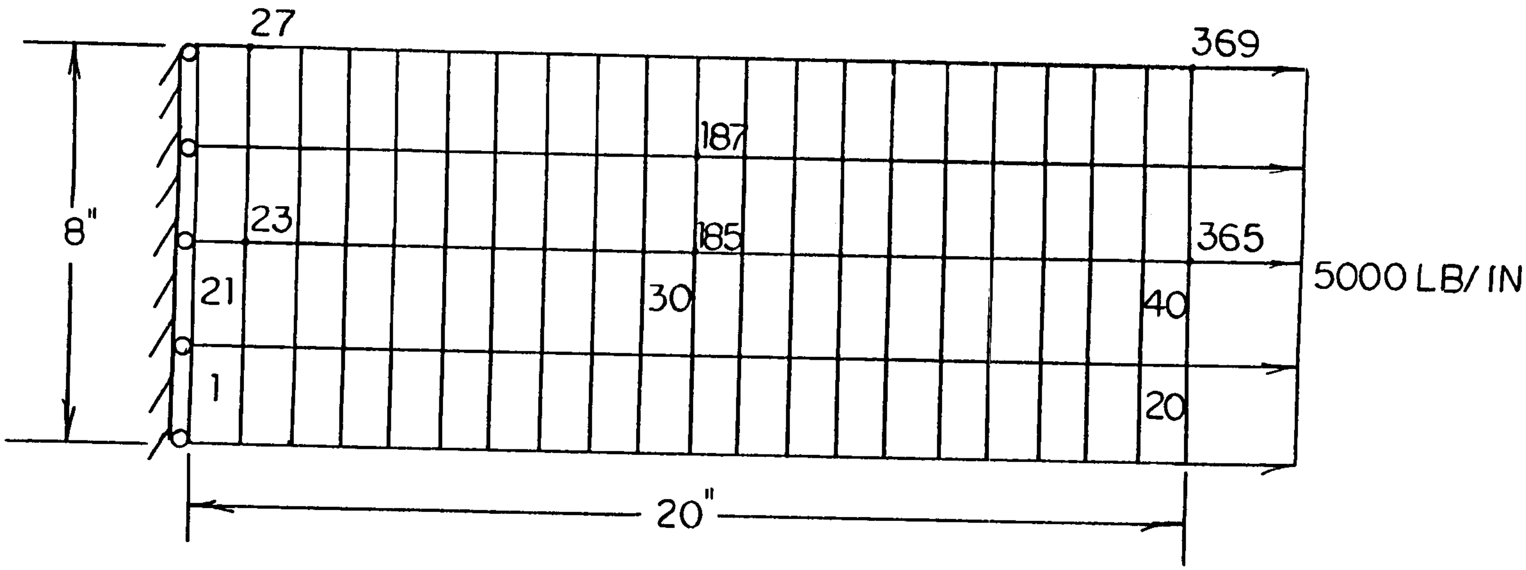


Figure 7. Plane Elastic Solid Finite Element Model

and Poission's ratio are given as $E = 3x10^7$ psi and $\nu = 0.3$, respectively. Each finite element is discretized so that a constant thickness of $h = 0.5$ in. is used in order to simplify the sensitivity calculation.

The compliance sensitivity results are shown in Table 1, where $\Delta\psi_1 = \psi_1(h + \delta h) - \psi(h)$ and ψ' and is the predicted value calculated from Eq. (9), with design perturbations of $\delta h = 0.01h$ and $\delta h=0.05h$. The percent accuracy of the sensitivity prediction is calculated using $\psi_1' \times 100/\Delta\psi_1$.

Table 1. Membrane Design Sensitivity Check for Compliance

δh	$\psi_1(h)$	$\psi_1(h+\delta h)$	$\Delta\psi_1$	ψ_1'	$\psi_1'\times 100/\Delta\psi_1$
0.01h	265.302	262.676	-2.627	-2.653	101.0
0.05h	265.302	252.668	-12.632	-13.265	105.0

Several discrete points shown in Fig. 6 are selected to check accuracy of the design sensitivity of the displacement functional of Eq. (14). In order to calculate this equation, the strain ε^{ij} due to the adjoint load is needed. Since the adjoint load is just a unit point load at point $\hat{x}$, acting in the direction of the displacement, a restart of the finite element analysis is all that is needed. For every node direction, there is a separate load case that produces a strain ε^{ij}, which in turn is used to calculate sensitivity of displacement. Design sensitivity predictions and differences, with $\delta h = 0.05h$ are given in Table 2.

Table 2. Design Sensitivity Check for Displacement

Node No.	Dir	$\psi_2(h)$	$\psi_2(h+\delta h)$	$\Delta\psi_2$	ψ'_2	% Accuracy
23	x1	2.974E-04	2.832E-04	-1.416E-04	-1.487E-05	105.0
27	x1	4.058E-04	3.865E-04	-1.932E-05	-2.029E-05	105.0
27	x2	-2.248E-04	-2.141E-04	1.071E-05	1.124E-05	105.0
185	x1	3.298E-03	3.141E-03	-1.571E-04	-1.649E-04	105.0
187	x1	3.299E-03	3.142E-03	-1.571E-04	-1.650E-04	105.0
187	x2	-2.014E-04	-1.918E-04	0.959E-05	1.007E-05	105.0
365	x1	6.633E-03	6.317E-03	-3.158E-04	-3.316E-04	105.0
369	x1	6.633E-03	6.317E-03	-3.158E-04	-3.316E-04	105.0
369	x2	-4.000E-04	-3.809E-04	1.905E-05	2.000E-05	105.0

To check the stress constraint sensitivity of Eq. (22), the equivalent nodal force of the adjoint load on the right of Eq. (25) has to be calculated so that $\varepsilon_{\ell}^{ij}(\lambda^{(3)})$ is known for each constrained element. This is accomplished by a restart of the original IFAD model, with each adjoint load for an element being a separate loading case. Design sensitivity results for principal and von Mises' stress functionals are given in Table 3 for several finite elements. The perturbations are $\delta h = 0.01h$ and $\delta h = 0.05h$ for von Mises' stress and $\delta h = 0.05h$ for principal stress.

The design sensitivity calculation is performed using double precision accuracy. Since the finite difference approximation in Tables 1, 2, and 3 are no smaller than two significant digits of the actual constraint functional, loss of significant digits is minimal. Therefore, the design perturbation is not to small. With all three

Table 3. Design Sensitivity Check for Stress

(a) von Mises' Stress with $\delta h = 0.01h$

El. No.	$\psi_3(h)$	$\psi_3(h + \delta h)$	$\Delta\psi_3$	ψ_3'	$(\psi_3'/\Delta\psi_3 \times 100)\%$
1	9888.882	9790.973	-97.910	-98.889	101.0
10	9989.932	9891.022	-98.910	-99.899	101.0
20	9999.982	9900.972	-99.010	-100.000	101.0
21	8752.850	8666.188	-86.662	-87.528	101.0
30	10024.586	9925.333	-99.253	-100.246	101.0
40	9999.853	9900.845	-99.008	-100.000	101.0

(b) von Mises' Stress with $\delta h = 0.05h$

El. No.	$\psi_3(h)$	$\psi_3(h + \delta h)$	$\Delta\psi_3$	ψ_3'	$(\psi_3'/\Delta\psi_3 \times 100)\%$
1	9888.882	9417.984	-470.899	-494.444	105.0
10	9989.932	9514.222	-475.711	-499.497	105.0
20	9999.982	9523.793	-476.189	-499.998	105.0
21	8752.850	8336.048	-416.802	-437.642	105.0
30	10024.586	9547.225	-477.361	-501.230	105.0
40	9999.853	9523.670	-476.183	-499.993	105.0

(c) Principal Stress with $\delta h = 0.05h$

El. No.	$\psi_3(h)$	$\psi_3(h + \delta h)$	$\Delta\psi_3$	ψ_3'	$(\psi_3'/\Delta\psi_3 \times 100)\%$
1	10582.770	10078.829	-503.941	-529.138	105.0
10	9987.061	9511.487	-475.574	-499.353	105.0
20	10000.013	9523.823	-476.191	-500.001	105.0
21	9660.279	9200.266	-460.013	-483.014	105.0
30	10012.976	9536.617	-476.808	-500.649	105.0
40	9999.987	9523.797	-476.189	-500.000	105.0

constraint functionals, the design sensitivity results compared to the finite difference approximation are excellent. This infers that the design perturbation is not too large as to cause significant nonlinearity effects in the calculation of design sensitivity.

It is interesting to note that in Tables 1, 2, and 3 the finite difference approximation is nearly 1% of the constraint functional when $\delta h = 0.01h$ and nearly 5% of the constraint functional when $h = 0.05h$. The results also show that as δh approaches zero, $\psi'/\Delta\psi$ approaches one.

3.2 Bending of Beams

A cantilevered beam finite elment model shown in Fig. 8 is loaded with a constant distributed force $f(x) = 0.03$ lb/in. along the entire length of the beam. It contains 20 IFAD beam elements of type 0501, each 3" in length, 121 nodal points, and 40 degrees-of-freedom, with design variables $u = [b(x),h(x)]^T$, the width and height of the beam. In Fig. 8, the element numbers are along the top of the beam and the node numbers are along the bottom of the beam. The beam has a rectangular cross-section with constant width and heigth $b = 0.5$ in. and $h = 0.75$ in., respectively. This gives the moment of inertia $I_y = 0.01758$ in.^{4} and the cross-sectional area $A_c = 0.375$ in^2. The material property constants, Young's modulus and Poisson's ratio are $E = 3\text{x}10^7$ psi and $\nu = 0.3$, respectively. Self weight is included in the analysis.

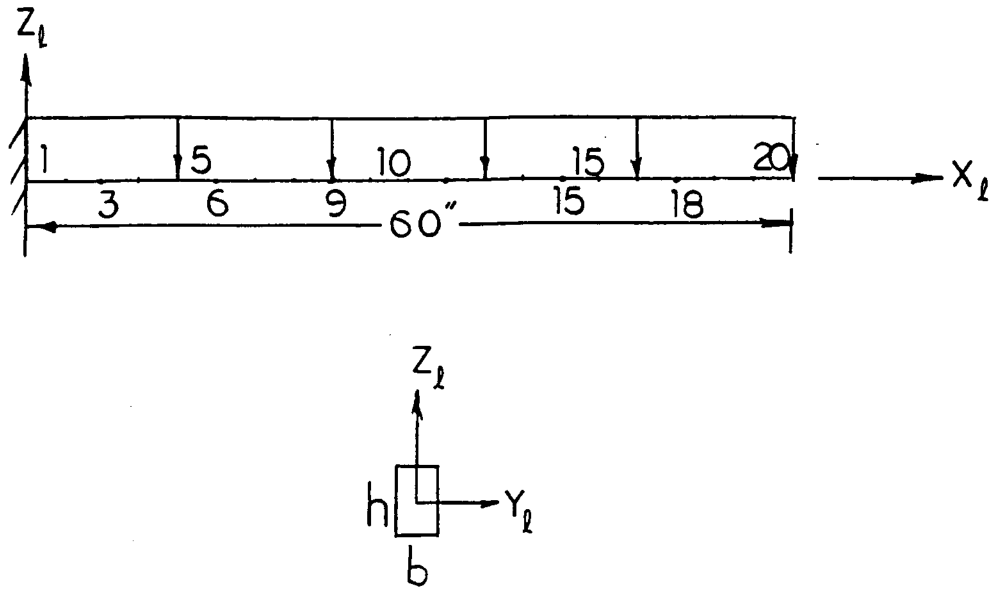


Figure 8. Cantilever Beam Finite Element Model

The compliance sensitivity results are shown in Table 4, where $\Delta\psi_4 = \psi_4(u + \delta u) - \psi_4(u)$ and ψ_4' is the predicted value calculated from Eq. (39), with design perturbations of $\delta b = 0.05b$, $\delta h = 0.05h$ and $\delta b = 0.01b$, $\delta h = 0.01h$.

Table 4. Beam Design Sensitivity Check for Compliance

δh	δb	$\psi_4(u)$	$\psi_4(u+\delta u)$	$\Delta\psi_4$	ψ_4'	% Accuracy
0.05h	0.05b	1.3684	1.3153	-0.0531	-0.0601	113.1
0.01h	0.01b	1.3684	1.3566	-0.0118	-0.0120	102.0

These results show that nonlinearities in the compliance of the cantilevered beam are highly evident. If too large a perturbation in design is chosen, the sensitivity will be inaccurate.

Several discrete points along the beam are selected to check the accuracy of design sensitivity of the displacement functional of Eq. (44). In order to calculate this equation, the beam curvature due to the adjoint load is needed. Since the adjoint load is just a unit point load at the point $\hat{x}$ acting in the -z direction, a restart of the finite element analysis is all that is needed. Displacement results are shown in Table 5 for design perturbations of $\delta b = 0.05b$, $\delta h = 0.05h$ and $\delta b = 0.01b$, $\delta h = 0.01h$.

In both Tables 5(a) and 5(b), results show that the design sensitivity compared to the finite difference approximation is good, with the exception of node 3. Since the finite diffenrence approximation is not too small in comparison to the constraint functional, loss of significant digits is not a valid reason for this inconsistency. The accuracy decreases as the node location approaches the restrained end of the beam. This is most likely due to the restraining effect, which causes rigidity in the beam and in turn gives smaller deflections.

To check the stress constraint sensitivity of Eq. (52), the equivalent nodal force of the adjoint load on the right side of Eq. (51) has to be calculated so that the curvature $\lambda^{(6)}_{xx_\ell}$ is known for each constraint element. This is accomplished by a restart of the orginal IFAD model, with each adjoint load for each element being a separate loading case. Allowable bending stress results for several finite elements are shown in Table 6 for a design perturbation of $\delta h = 0.05h$ and $\delta b = 0.05b$.

Table 5. Beam Design Sensitivity Check for Displacement

(a) $\delta b = 0.01b$ and $\delta h = 0.01h$

Node	$\psi_5(u)$	$\psi_5(u+\delta u)$	$\Delta\psi_5$	ψ_5'	$\psi_5' \times 100/\Delta\psi_5$
3	7.8353E-03	7.6473E-03	-1.8804E-04	-1.6082E-04	85.5
6	4.4162E-02	4.3102E-00	-1.0604E-03	-9.9992E-04	94.2
9	1.0181E-01	9.9365E-02	-2.4451E-03	-2.3554E-03	96.3
12	1.7317E-01	1.6901E-01	-4.1590E-03	-4.0441E-03	97.2
15	2.5229E-01	2.4623E-01	-6.0594E-03	-539211E-03	97.7
18	3.3493E-01	3.2686E-01	-8.0447E-03	-7.8836E-03	97.9
21	4.1857E-01	4.0852E-01	-1.0054E-02	-9.8701E-03	98.2

(b) $\delta b = 0.05b$ and $\delta h = 0.05h$

Node	$\psi_5(u)$	$\psi_5(u+\delta u)$	$\Delta\psi_5$	ψ_5'	$\psi_5' \times 100 \Delta\psi_5$
3	7.8353E-03	6.9743E-03	-8.6100E-04	-8.0412E-04	93.4
6	4.4162E-02	3.9306E-02	-4.8555E-03	-4.9959E-04	102.9
9	1.0181E-01	9.0619E-02	-1.1120E-02	-1.1777E-02	105.2
12	1.7317E-01	1.5412E-01	-1.9043E-02	-2.0220E-02	106.2
15	2.5229E-01	2.2454E-01	-2.7745E-02	-2.9605E-02	106.7
18	3.3493E-01	2.9810E-01	-3.6835E-02	-3.9418E-02	107.0
21	4.1857E-01	3.7253E-01	-4.6034E-02	-4.9350E-02	107.2

Table 6. Beam Design Sensitivity Check for Stress

El.	$\psi_6(u)$	$\psi_6(u+\delta u)$	$\Delta\psi_6$	ψ_5'	$\psi_6' \times 100/\Delta\psi_6$
1	4932.767	4609.263	-323.504	-349.234	108.0
5	3110.255	2906.201	-204.055	-219.226	107.4
10	1420.622	1327.375	- 93.287	- 99.103	106.2
15	385.008	359.664	- 25.344	- 26.076	102.9
20	3.295	3.067	- 0.228	- 0.147	64.7

Near the end of the beam, where the allowable bending stress is near zero, the design sensitivity decreases significantly. Since the design sensitivity is the derivative of the constraint functional, and the derivative is physically interpreted as the slope of the beam, this decrease can be attributed to the large increase in slope at the free end of the beam.

3.3 Bending of Plates

The clamped plate finite element model shown in Fig. 9 is uniformly loaded with a pressure $f(x) = -1.5$ lb/in. in the z direction. Since the model is symmetric along two planes, only one quarter of it needs to be analyzed and symmetric boundary conditions need to be applied. The quarter model contains 100 IFAD triangular thin shell elements of type 1601, with only the bending terms active. It has 61 nodal points and 140 degrees-of-freedom.

The design variable is the plate thickness $u = t(x)$, and the material property constants, Young's Modulus and Poisson's ratio, are E 30.5×10^6 psi and $\nu = 0.3$, respectively. The constant plate thickness is $t = 0.4$ in. and the self-weight of the plate is neglected.

Compliance sensitivity results are shown in Table 7, where $\Delta\psi_7 = \psi_7(t + \delta t) - \psi_7(t)$ and ψ_7' is the predicted value calculated from Eq. (60), with design perturbations of $\delta t = 0.01t$ and $\delta t = 0.05t$.

Both perturbations for the compliance constraint functional give good correlation between the design sensitivity and the finite difference approximation. This implies that a five percent change in thickness is acceptable when making design improvements.

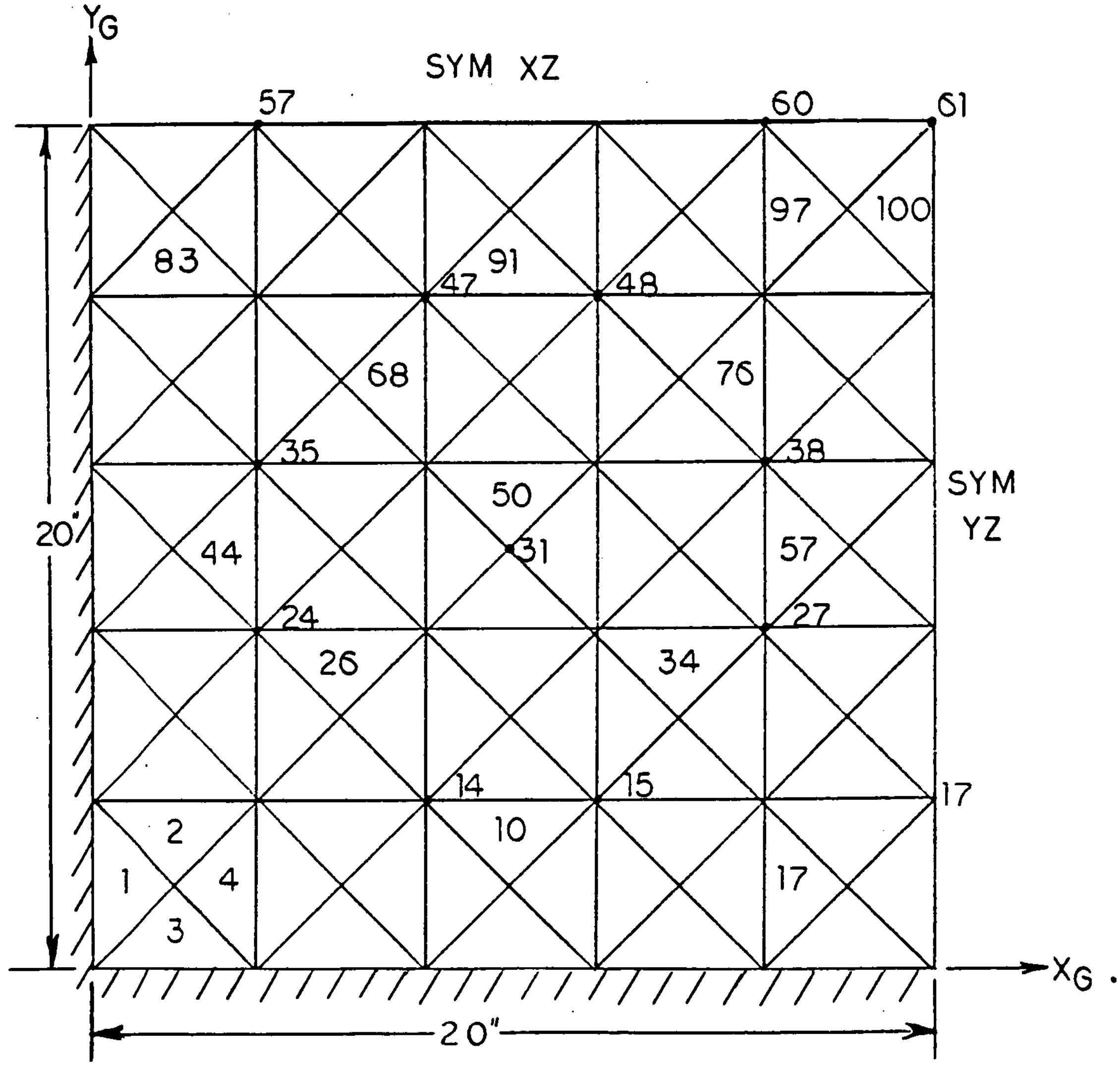


Figure 9. Bending Plate Finite Element Model

Table 7. Plate Design Sensitivity Check for Compliance

δt	$\psi_7(t)$	$\psi_7(t+\delta t)$	$\Delta\psi_7$	ψ_7'	$\psi_7' \times 100/\Delta\psi_7$
.05t	4.9625	4.2868	-0.6757	-0.7204	106.6
.01t	4.9625	4.8165	-0.1459	-0.1441	98.7

Several discrete points in Fig. 8 are selected to check accuracy of the design sensitivity of the displacement functional in Eq. (69). In order to calculate this equation, just as in the membrane case, the strain $\varepsilon^{ij}(\lambda^{(8)})$ due to the adjoint load is needed. A restart of the finite element analysis, using a unit point load in the -z direction for each selected point as a separate load case, accomplishes this task. Some displacement results are shown in Table 8 for a design perturbation of δt = 0.01t.

Table 8. Plate Design Sensitivity Check for Displacement

Node	$\psi_8(t)$	$\psi_8(t+\delta t)$	$\Delta\psi_8$	ψ_8'	$\psi_8' \times 100/\Delta\psi_8$
14	1.9033E-03	1.8473E-03	-5.5976E-05	-5.2596E-05	94.0
15	3.0497E-03	2.9600E-03	-8.9692E-05	-8.5144E-05	94.9
24	1.9033E-03	1.8473E-03	-5.5976E-05	-5.2596E-05	94.0
27	1.1258E-02	1.0927E-02	-3.3109E-04	-3.2620E-04	98.5
31	9.7772E-03	9.4897E-03	-2.8754E-04	-2.8309E-04	98.4
35	3.0497E-03	2.9600E-03	-8.9692E-05	-8.5144E-05	94.9
38	1.8450E-02	1.7908E-02	-5.4262E-04	-5.3957E-04	99.4
47	1.1258E-02	1.0927E-02	-3.3109E-04	-3.2620E-04	98.5
48	1.8450E-02	1.7908E-02	-5.4262E-04	-5.3957E-04	99.4
57	4.2639E-03	3.9443E-03	-1.1952E-04	-1.1410E-04	95.5
60	2.5084E-02	2.4366E-02	-7.3773E-04	-7.3688E-04	99.9
61	2.6942E-02	2.6150E-02	-7.9237E-04	-7.9223E-04	100.0

The design sensitivity results of Table 8 seem to substantiate the fact that nodes near a clamped edge give a somewhat less accurate prediction of the design sensitivity. However, these results are still considered good, since there is less than 15% deviation from the approximate finite difference result. Since the finite difference result is only an approximation, it is reasonable to say that the design sensitivity prediction is good every where along the plate.

A number of elements are selected to check the design sensitivity for von Mises' stress in Eq. (80). Before this an be evaluated, the nodal force of the adjoint load on the right side of Eq. (79) has to be calculated so that the strain $\varepsilon^{ij}(\lambda^{(9)})$ is known for each constraint element. This is accomplished by a restart of the original IFAD model with each adjoint load for each element being a separate loading case. Von Mises' stress results are shown in Table 9 for a design perturbation of $\delta t = 0.001t$.

Recall that the design sensitivity and the constraint functional was calculated using an integration technique where the stresses were calculated at the midside nodes of the triangular element. This integration technique was used so as to get the best design sensitivity results as possible for the finite elements. This is the technique used because the element is a hybrid element [4]. The results in Table 9 are excellent when this technique is used. These results further substantiate the fact that as design perturbation approaches zero, $\psi/\Delta\psi$ approaches one.

Table 9. Plate Design Sensitivity Check for von Mises' Stress

El. No.	$\psi_9(t)$	$\psi_9(t+\delta t)$	$\Delta\psi_9$	ψ_9'	$\psi_9' \times 100/\Delta\psi_9$
1	172.1862	171.8423	-0.3439	-0.3445	100.17
2	752.7383	751.2351	-1.5032	-1.5049	100.11
3	172.1861	171.8423	-3.4386	-3.4446	100.17
4	752.7383	751.2351	-1.5032	-5.0495	100.11
10	1442.8840	1140.0025	-2.8815	-2.8841	100.09
17	2457.3522	2452.4447	-4.9074	-4.9125	100.10
25	1149.0097	1146.7151	-2.2947	-2.2970	100.11
26	1331.9565	1329.2966	-2.6600	-2.6624	100.10
27	1149.0097	1146.7151	-2.2946	-2.2970	100.11
28	1331.9565	1329.2966	-2.6600	-2.6662	100.09
34	973.8975	971.9526	-1.9450	-1.9465	100.08
44	1442.8840	1440.0025	-2.8815	-2.8841	100.09
49	1284.1558	1281.5913	-2.5645	-2.5667	100.08
50	1278.5066	1275.9534	-2.5532	-2.5554	100.09
51	1274.1558	1281.5913	-2.5645	-2.5667	100.08
52	1278.5066	1275.9531	-2.5532	-2.5554	100.09
57	1110.0286	1107.8118	-2.2168	-2.2188	100.09
68	973.8975	971.9526	-1.9449	-1.9465	100.08
73	1470.1247	1417.2887	-2.8360	-2.8384	100.08
74	1629.1926	1625.9391	-3.2536	-3.2563	100.08
75	1420.1247	1417.2887	-2.8360	-2.8384	100.08
76	1629.1926	1625.9391	-3.2536	-3.2561	100.08
83	2457.3522	2452.4447	-4.9074	-4.9125	100.10
91	1110.0286	1107.8118	-2.2168	-2.2188	100.09
97	1884.4503	1880.6870	-3.7633	-3.7660	100.07
98	2010.1163	2006.1021	-4.0143	-4.0170	100.07
99	1884.4503	1880.6870	-3.7633	-3.7660	100.07
100	2010.1663	2006.1021	-4.0143	-4.0170	100.07

3.4 Built-Up Structure

A built-up structure that uses both beams and plates is shown in Fig. 10. Since it is symmetric along two planes, only a quarter is modeled. The built-up structure is clamped on two edges, with symmetric boundary conditions applied along the other two edges. The plates and the beams are considered to be welded. A uniform pressure $f(x) = -1.5$ $lb/in.^2$ is applied on the top surface of the plates. The model contains 100 IFAD triangular thin shell elements of type 1601, with only the bending terms active, and 20 IFAD beam elements of type 0501. There are 61 nodal points and 140 degrees-of-freedom. There are three design variables, beam width, beam height, and plate thickness. The material constants, Young's modulus and Poisson's ratio, for both the beams and the plates are $E = 30.5 \times 10^6$ psi and $\nu = 0.3$, respectively. Self weight is neglected and the initial design variables are $b = 0.5$ in., $h = 0.75$ in., and $t = 0.4$ in.

Compliance sensitivity results are shown in Table 10, where $\Delta\psi_{11} = \psi_{10}(u + \delta u) - \psi_{10}(u)$ and ψ'_{11} is the predicted design sensitivity, using Eq. (101) with a 1% design perturbation for all the design variables.

Several discrete points in Fig. 10 are selected to check the accuracy of design sensitivity of the displacement functional in Eq. (104). To calculate this equation, the strain $\varepsilon^{ij}(\lambda)^{(12)}$ due to the adjoint load is obtained by doing a restart of the finite element analysis. A unit point load in the -z direction for each selected point

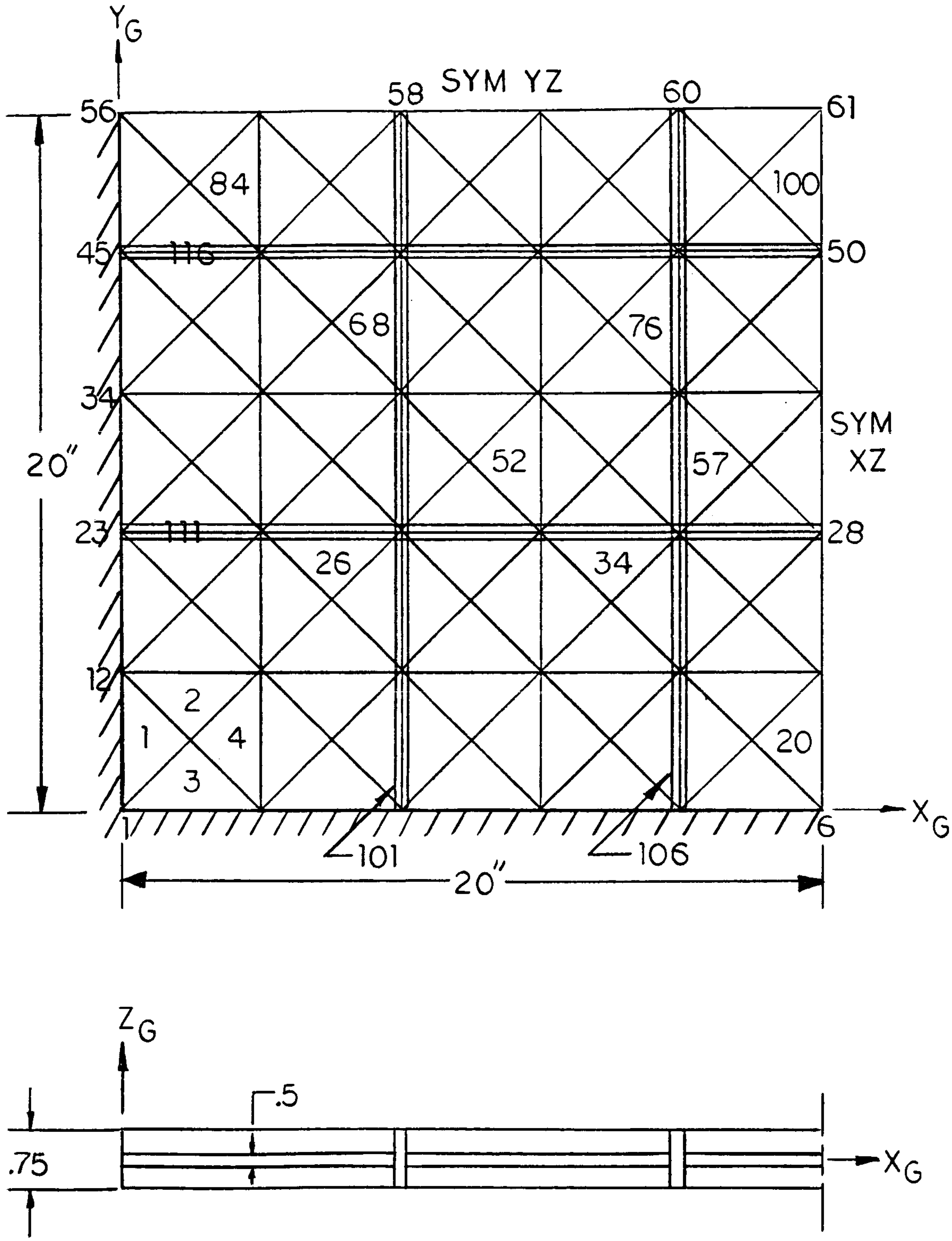


Figure 10. Built-up Structure Finite Element Model

Table 10. Built-up Structure Design Sensitivity Check for Compliance with $\delta h = 0.01h$ $\delta b = 0.01b$, and $\delta t = 0.01t$

$\psi_{11}(u)$	$\psi_{11}(u+\delta u)$	$\Delta\psi_{11}$	ψ'_{11}	$\psi'_{11}\times 100/\Delta\psi_{11}$
3.7200	3.6017	-0.1183	-0.1189	100.5

is specified as a separate load case. The design sensitivity results for these selected points are shown in Table 11, where a 1% perturbation for each design variable is used.

Table 11. Built-up Structure Design Sensitivity Check for Displacement with $\delta h = 0.01h$, $\delta b = 0.01b$, and $\delta t = 0.01t$

Node	$\psi_{12}(u)$	$\psi_{12}(u+\delta u)$	$\Delta\psi_{12}$	ψ'_{12}	$\psi'_{12}\times 100/\Delta\psi_{12}$
13	4.768E-04	4.616E-04	-1.520E-05	-1.453E-05	95.6
15	2.294E-06	2.222E-03	-7.283E-05	-6.750E-05	92.7
17	3.064E-03	2.966E-03	-9.723E-05	-9.152E-05	94.1
25	4.125E-03	3.994E-03	-1.313E-04	-1.255E-04	95.6
27	5.421E-03	8.153E-03	-2.680E-04	-2.621E-04	97.8
35	2.294E-03	2.221E-03	-7.283E-05	-6.750E-05	92.7
37	1.095E-02	1.060E-02	-3.484E-04	-3.448E-04	99.0
39	1.485E-02	1.438E-02	-4.724E-04	-4.737E-04	100.3
47	8.421E-03	8.153E-03	-2.680E-04	-2.621E-04	97.8
49	1.755E-02	1.699E-02	-5.580E-04	-5.621E-04	100.7
55	1.955E-02	1.892E-02	-6.215E-04	-6.296E-04	101.3
57	3.064E-03	2.966E-03	-9.723E-05	-9.152E-05	94.1
59	1.485E-02	1.438E-02	-4.724E-04	-4.737E-04	101.3
61	2.025E-02	1.961E-02	-6.439E-04	-6.535E-04	101.5

Both the compliance and displacement design sensitivity results show good correlation with the approximated finite difference. This indicates that the method of design sensitivity is a good method for predicting the response of a built-up structure for these constraint functionals. This is substantiated by the fact that correlation between the design sensitivities and the finite difference approximations for the built-up structure in Tables 10 and 11 are not that different than those in Tables 7 and 8 where plate bending results are shown.

To check the stress constraint sensitivity of Eq. (98), the equivalent nodal force of the adjoint load on the right of Eq. (97) has to be calculated so that $\varepsilon^{ij}(\lambda)^{(10)}$ is known for each constrained element. A restart of the original IFAD built-up structure model is made, with each adjoint load for a finite element being a separate loading case. The design sensitivity results for the von Mises' stress functional are given in Table 12, with design perturbation $\delta t = 0.001t$.

The design sensitivity results compared to the finite difference approximations fluctuate more than would be expected, considering the excellent correlation of the von Mises' stress design sensitivites for the bending plate in Table 9. It doesn't appear that the problem is caused by the design perturbation being too small, because the finite difference approximation is no smaller than three significant digits of the actual constraint functional. It is possible that the nonlinear response of the built-up structure is partially the cause of the inconsistencies in the design sensitivities, but not the whole problem, since the design perturbation is quite small.

It is most likely that the beam contribution to the built-up structure is the major influence. The beam seems to be more influenced by nonlinearities, as shown in Tables 4, 5, and 6. It is most probable that if a more complex beam element, such as a cubic beam, was used, design sensitivity results would improve. Unfortunately the IFAD code does not currently support this type of beam, so this cannot be verified in this study.

Table 12. Built-up Structure Design Sensitivity Check for von Mises' Stress with $\delta t = 0.001t$, $\delta b = 0.001b$, and $\delta h = 0.001h$

Element	$\psi_{10}(t)$	$\psi_{10}(t+\delta t)$	$\Delta\psi_{10}$	ψ'_{10}	$\psi'_{10}\times 100/\Delta\psi_{10}$
1	127.8318	127.5490	-0.2828	-0.3339	118.1
2	549.7750	548.5339	-1.2411	-1.2230	98.5
3	127.8318	127.5490	-0.2828	-0.3339	118.1
4	549.7750	548.5339	-1.2411	-1.2230	98.5
10	1074.7414	1072.3202	-2.4212	-2.7653	114.2
17	1828.5601	1824.4411	-4.1190	-4.2965	104.3
25	875.4135	873.4610	-1.9525	-2.3201	118.1
26	985.6012	983.3797	-2.2215	-2.6663	120.0
27	875.4135	873.4610	-1.9525	-2.3201	118.1
28	985.6012	983.3747	-2.2215	-2.6663	120.0
34	735.6485	734.0066	-1.6419	-2.2807	138.9
44	1074.7414	1072.3202	-2.4212	-2.7653	114.2
50	962.8944	960.7381	-2.1563	-2.4538	113.8
51	944.5642	942.4288	-2.1354	-2.7966	131.0
57	804.5913	802.7656	-1.8257	-2.2406	122.7
68	735.6485	734.0066	-1.6420	-2.2807	138.9
73	1074.1010	1071.6982	-2.4028	-2.7000	112.6
74	1243.6633	1240.8853	-2.7780	-3.2605	117.4
75	1074.1010	1071.6982	-2.4028	-2.7060	112.6
76	1243.6633	1240.8853	-2.7780	-3.2605	117.4
83	1828.5601	1824.4411	-4.1190	-4.2965	104.3
91	804.5913	802.7656	-1.8257	-2.2406	122.7
97	1401.1703	1398.0144	-3.1559	-3.2171	101.9
98	1517.4522	1514.0511	-3.4011	-3.6964	108.7
99	1401.1703	1398.0144	-3.1559	-3.2171	101.9
100	1517.4522	1514.0511	-3.4011	-3.6964	108.7

CHAPTER IV

CONCLUSIONS

The results of this study indicate that it is feasible to implement the theoretical design sensitivity analysis of Ref. 1 with an existing finite element code. Calculations of the design sensitivites can be accomplished outside of the finite element code using only the postprocessing data. In addition, the results show that accurate design sensitivity can be predicted without the uncertainty of numerical accuracy associated with the selection of a finite difference perturbation. However, results also indicate that the integration technique used in the calculation of the design sensitivity and the knowledge of the exact finite element shape functions used for each finite element in the finite element code is important in getting the most accurate design sensitivities possible.

Results of the built-up structure indicate that design sensitivities of the built-up structure cannot be any more accurate than the accuracy of the individual components. Care must be taken to use the best integration techniques and the same shape functions as the finite element analysis for the individual components, if at all possible.

REFERENCES

1. Haug, E. J., Choi, K. K., and Komkov, V., Structural Design Sensitivity Analysis, Academic Press, New York, N. Y., to appear, 1985.

2. Cook, Robert D., Concepts and Applications of Finite Element Analysis, 2nd Ed., John Wiley and Sons, Inc., New York, 1981.

3. "Applicon IFAD User's Guide," Applicon, Inc. Cleveland, Ohio, 1983.

4. Razzaque, Abdur, "Program for Triangular Bending Elements with Derivative Smoothing," Int. Journal for Num. Methods in Eng., Vol 6, 1973, pp. 333-343.

5. Roark, Raymond J., Formulas for Stress and Strain, 3rd Ed., McGraw-Hill, New York, N.Y., 1954.

APPENDIX

DESIGN SENSITIVITY IFAD PROGRAM

```
      PROGRAM SENSIT
CP*************************************************************
CP*
CP* SENSIT: THE MAIN PROGRAM FOR CALCULATING DESIGN SENSITIVITY
CP*
CP*************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*       'SENSIT' IS THE MAIN PROGRAM FOR THE DESIGN SENSITIVITY
CP*       VECTOR CALCULATION. IT CHECKS THE ACCURACY OF THE
CP*       SENSITIVITY TO THE FINITE DIFFERENCE METHOD IF THE
CP*       TWO FINITE ELEMENT ANALYSES ( THE ORIGINAL ANALYSIS
CP*       AND THE PERTURBED ANALYSIS ) ARE ALREADY CREATED.
CP*       MAX. OF 500 ELEMENTS. IF MORE ELEMENTS ARE NEEDED THE
CP*       SVECTR.MON COMMON BLOCK FILE NEEDS TO CHANGED.
CP*
CP*************************************************************
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
      INCLUDE '[AEGSDR.INC] SVECTR.MON'
C
      EQUIVALENCE (NDAT(46),NELM),(NDAT(181),IPNAME)
C
      DIMENSION PS1B(2),IPNAME(2),IPNAM1(2),IPNAM2(2)
      CHARACTER YESNO*1
C
C
      NT = 0
      ISAC = 0
      LCS = 1
      NLC = 1
C
C  ASK FOR FINITE ELEMENT MODEL FILE NAME
C
      PRINT *,' '
      PRINT *,'ENTER THE ORIGINAL PROBLEM NAME (1-8 CHARS)'
      READ(5,1009) IPNAM1
C
C  ASK FOR CONSTRAINT TYPE
C
      PRINT *,' '
      PRINT *,'ENTER CONSTRAINT TYPE: 1 = COMPLIANCE'
      PRINT *,'                       2 = DISPLACEMENT'
      PRINT *,'                       3 = STRESS'
      READ(5,1006) ICT
      IF(ICT.NE.3)  GO TO 20
      PRINT *,' '
      PRINT *,'ENTER STRESS TYPE: 1 = PRINCIPAL'
      PRINT *,'                   2 = VON MISES'
      PRINT *,'                   3 = BEAM ALLOWABLE'
      READ(5,1006) IST
C
20    PRINT *,'CALCULATING ADJOINT LOADS (Y/N) ?'
      READ(5,1000) YESNO
      IF(YESNO.EQ.'Y') ISAC = 1
      IF(ISAC.EQ.1) GO TO 40
C
C  ASK FOR THE PERTURBED FINITE ELEMENT NAME
C
      PRINT *,' '
```

```
      PRINT *,'ENTER THE PERTURBED PROBLEM NAME (1-8 CHARS)'
      READ(5,1009) IPNAM2
C
C  OPEN AN OUTPUT FILE
C
      OPEN(UNIT=10,NAME='RESULT.DAT',TYPE='NEW')
      WRITE(10,1007) ICT
      IF(ICT.NE.3)  GO TO 40
      WRITE(10,1008) IST
C
C  GET LOAD CASE START AND END NOS.
C
40    PRINT *,' '
      PRINT *,'ENTER NO. OF LOAD CASES TO BE PROCESSED'
      READ(5,1006) NLC
      PRINT *,'ENTER FIRST LOAD CASE NO.'
      READ(5,1006) LCS
50    LCS = LCS - 1
C
C  GET CONSTRAINED ELEMENTS FOR STRESS CALCULATION
C
      IF(ICT.NE.3)  GO TO 90
      PRINT *,' '
      IF(ISAC.EQ.1)  GO TO 60
      PRINT 2000, NLC
      GO TO 70
C
C  GET ELEMENTS FOR ADJOINT LOAD CALCULATION
C
60    PRINT 2001, NLC
70    DO 80 NE=1,NLC
80    READ(5,1006)  ICE(NE)
90    DO 800 NC=1,NLC
      IF(ISAC.EQ.1)  GO TO 100
      WRITE(10,1010)  ICE(NC)
C
C  GET SENSITIVITY VECTOR
C
100   IF(NT.EQ.1)  GO TO 110
      IPNAME(1) = IPNAM1(1)
      IPNAME(2) = IPNAM1(2)
      GO TO 120
110   IPNAME(1) = IPNAM2(1)
      IPNAME(2) = IPNAM2(2)
120   CALL GETSEN(PSIB,NT,NELM,IPNAME)
      IF(ISAC.EQ.1) GO TO 910
      IF(NT.GT.1)  GO TO 200
      GO TO 100
C
C  CALC. CHANGE IN PLATE THICKNESS, CHANGE IN BEAM WIDTH AND DEPTH
C
200   NT = 0
      DT = DABS(TM(2)-TM(1))
      DB = DABS(BW(2)-BW(1))
      DH = DABS(BH(2)-BH(1))
      TB = DABS(FB(2)-FB(1))
C
C  CALCULATE THE PERCENT ACCURACY
C
      DPSIBDB = 0.0
      DO 300 I=1,NELM
```

```
          DPSIBDB = DPSIBDB+DPSIT(I)*DT+DPSIB(I)*DB
     *              +DPSIH(I)*DH+DPSITB(I)*TB
300   CONTINUE
      PNUM = DPSIBDB*100
400   PDEN = PSIB(2)-PSIB(1)
      IF(PDEN.EQ.0)  GO TO 801
      PACCUR = PNUM/PDEN
C
      WRITE(10,*) ' '
      WRITE(10,1004) DT,DB,DH,TB
      WRITE(10,1005) PDEN
      WRITE(10,1002) DPSIBDB
      WRITE(10,1003) PACCUR
800   CONTINUE
C
      GO TO 900
C
801   PRINT *, 'DPSI(B)*DB = 0'
C
900   CLOSE(10)
910   CONTINUE
C
C
1000  FORMAT(A)
1001  FORMAT(F8.5)
1002  FORMAT(1X,'DPSI(B)*DELTAB=',E16.8)
1003  FORMAT(1X,'PERCENT ACCURACY=',F16.8)
1004  FORMAT(1X,'CHANGE IN MEMBRANE THICKNESS =',F8.5,/,1X,'CHANGE
     * IN BEAM WIDTH =',F8.5,/,1X,'CHANGE IN BEAM DEPTH =',F8.5,
     */,1X,'CHANGE IN BENDING PLATE THICKNESS =',F8.5)
1005  FORMAT(1X,'PSI(B+DB) - PSI(B) = ',F16.8)
1006  FORMAT(I4)
1007  FORMAT(1X,'***CONSTRAINT TYPE =',I4)
1008  FORMAT(/,1X,'***STRESS TYPE =',I4)
1009  FORMAT(2A4)
1010  FORMAT(1X,'CONSTRAINT ELEMENT IS ',I4)
2000  FORMAT(1X,'ENTER',I4,' CONSTRAINT ELEMENTS, FOLLOW EACH BY
     * A RETURN')
2001  FORMAT(1X,'ENTER',I4,' ELEMENTS THAT ARE TO HAVE AN ADJOINT
     * LOAD CALCULATED. FOLLOW EACH BY A RETURN.')
C
C
      END
```

```
      SUBROUTINE AL16(X,Y,C,AL,TB,THK,IT)
CP**************************************************************
CP*
CP* AL16: ADJOINT LOAD CALCULATION FOR TRIANGULAR ELEMENT 1601
CP*
CP**************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*       'AL16' CALCULATES THE ADJOINT LOADS FOR THE TRIANGULAR
CP*       PLATE BENDING ELEMENT. AL = [C]*[B]*MP  WHERE
CP*       [C] IS THE DERIVATIVE OF THE STRESS FUNCTION VECTOR
CP*       TIMES THE ELASTICITY MATRIX. SINCE XFAD CALCULATES
CP*       STRESS RESULTANTS FIRST, [AL] MUST BE MULTIPLIED BY
CP*       PLATE THICKNESS DIVIDE BY 2.
CP*       [B] IS A 3X9 MATRIX IN COLUMNS 4,5 & 6 OF [W].  MP
CP*       IS THE CHARACTERISTIC FUNCTION THAT IS 1/AREA OF THE
CP*       ELEMENT THAT IS CONSTRAINED AND ZERO FOR ALL THE OTHER
CP*       ELEMENTS.
CP*
CP**************************************************************
CP*
CP*   X      THE LOCAL ELEMENT X COORDINATE
CP*   Y      THE LOCAL ELEMENT Y COORDINATE
CP*   C      MATRIX [C] = [DG]*[E]*T/2   3X3 MATRIX
CP*   AL     ADJOINT LOAD VECTOR
CP*   TB     TRANSFORMATION MATRIX
CP*   THK    MATERIAL THICKNESS
CP*   IT     MIDSIDE NODE COLUMN LOCATOR
CP*
CP**************************************************************
C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
C
      EQUIVALENCE (NDAT(14),IPR)
C
      DIMENSION X(3),Y(3),GPTS(3,3),XL(6),YL(6),W(18,7),C(3),
     *          F(9),AL(18),R(6,3),TB(6,6),RP(6,3)
C
      DATA GPTS/0.0D0,.5D0,.5D0,  .5D0,0.0D0,.5D0,  .5D0,.5D0,0.0D0/
C
C*** INITIALIZE VARIABLES
C
      DO 10 I=1,9
10    F(I) = 0.0D0
      DO 12 I=1,18
12    AL(18) = 0.0D0
      DO 14 I=1,6
        DO 14 J=1,3
          R(I,J) = 0.0D0
14    CONTINUE
C
      AREA = EUTRIA(X,Y)
      XMP = 1.D0/AREA
C
C*** GET LOCAL X AND Y COORDINATES
C
      CALL MOVESP(XL,X,3*IPR)
      CALL MOVESP(YL,Y,3*IPR)
       CALL SF1501(XL,YL,GPTS(1,IT),W,EM)
```

```
      DO 50 M=1,9
        GASH = 0.0D0
        DO 40 J=1,3
          GASH = GASH+C(J)*W(M,J+3)
40      CONTINUE
        F(M) = F(M)+GASH*XMP
50    CONTINUE
C
C** ROTATE ADJOINT LOAD VECTOR TO GLOBAL COORDINATE SYSTEM
C
      N = 1
      DO 60 MM=1,3
        R(3,MM) = F(N)
        R(4,MM) = F(N+1)
        R(5,MM) = F(N+2)
        N = N+3
60    CONTINUE
      CALL UMXAB(TB,R,RP,6,3,6)
      N1 = 0
      DO 80 M=1,3
        DO 70 M1=1,6
          AL(M1+N1) = RP(M1,M)
70      CONTINUE
        N1 = N1+6
80    CONTINUE
      DO 90 N2 = 1,18
        AL(N2) = AL(N2)*THK/2.0D0
90    CONTINUE
C
C
      RETURN
      END
```

```
      FUNCTION AREAQ(X,Y)
CP*************************************************************
CP*
CP* AREAQ: CALCULATES THE AREA OF A STRAIGHT SIDED FOUR OR
CP*        EIGHT NODE ELEMENT.
CP*
CP*************************************************************
CP*
CP*   X    GLOBAL X COORDINATES
CP*   Y    GLOBAL Y COORDINATES
CP*
CP*************************************************************
C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIPN.MON'
      INCLUDE '[AEGSDR.INC] ELEDES.MON'
C
      DIMENSION X(4),X1(3),X2(3),Y(4),Y1(3),Y2(3),Z(4),BUF(100)
C
      DATA IREF/1/
C
C**** CALCULATES THE AREA OF A QUADRALATERAL
C
      IF(NUNPE.EQ.8)  GO TO 20
C
C  FOUR NODED ELEMENT
C
        X1(1) = X(1)
        X1(2) = X(2)
        X1(3) = X(4)
        Y1(1) = Y(1)
        Y1(2) = Y(2)
        Y1(3) = Y(4)
        A1 = EUTRIA(X1,Y1)
        X2(1) = X(2)
        X2(2) = X(3)
        X2(3) = X(4)
        Y2(1) = Y(2)
        Y2(2) = Y(3)
        Y2(3) = Y(4)
        A2 = EUTRIA(X2,Y2)
      GO TO 30
C
C  EIGHT NODED ELEMENT WITH STRAIGHT SIDES
C
20      X1(1) = X(1)
        X1(2) = X(3)
        X1(3) = X(5)
        Y1(1) = Y(1)
        Y1(2) = Y(3)
        Y1(3) = Y(5)
        A1 = EUTRIA(X1,Y1)
        X2(1) = X(1)
        X2(2) = X(5)
        X2(3) = X(7)
        Y2(1) = Y(1)
        Y2(2) = Y(5)
        Y2(3) = Y(7)
        A2 = EUTRIA(X2,Y2)
C
30    AREAQ = A1+A2
```

```
C
      GO TO 900
C
807   PRINT 877, IERR
C
877   FORMAT(1X,'ACCELC RETURNED WITH ERROR', I4)
900   CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE COMP(PSIB,NT,NELM)
CP**************************************************************
CP*
CP* COMP: BRANCHES TO THE APPROPRIATE ELEMENT TYPE TO CALCULATE
CP*
CP**************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*       'COMP' BRANCHES TO THE APPROPRIATE ELEMENT TYPE TO
CP*       CALCULATE THE COMPLIANCE AND THE COMPLIANCE SENSIT-
CP*       IVITY VECTOR.
CP*
CP**************************************************************
C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACC1PN.MON'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
      INCLUDE '[AEGSDR.INC] ELEDES.MON'
      INCLUDE '[AEGSDR.INC] SVECTR.MON'
      COMMON/LCSDES/DLCS(90)
C
      EQUIVALENCE (NDAT(97),IDBS),(NDAT(98),IDBL)
C
      DIMENSION DATN(50),PSIBB(500),PSIB(2),CFBUF(6),PSIBTB(500)
C
      DATA IREF/1/
C
      PSIB16 = 0.0
      PSIBC5 = 0.0
      SDPSIT = 0.0
      SDPSIB = 0.0
      SDPSIH = 0.0
      SDPSTB = 0.0
C
C  SET LOAD CASE NUMBER FOR COMPLIANCE
C
      L1 = LCS + NC
C
C
C  SETUP POINTERS
C
      CALL ACCELM(1,IPNELM,IDBS,1,0,IERR)
      IF(IERR.NE.0)  GO TO 800
      CALL ACCFES(1,IPNFES,IDBL,1,L1,0,0,IERR)
      IF(IERR.NE.0)  GO TO 801
      CALL ACCCND(1,IPNCND,IDBL,1,L1,0,0,IERR)
      IF(IERR.NE.0)  GO TO 802
      CALL ACCLCS(1,IPNLCS,IDBL,1,0,IERR)
      IF(IERR.NE.0)  GO TO 805
      CALL ACCNOD(1,IPNNOD,IDBS,1,0,IERR)
      IF(IERR.NE.0)  GO TO 806
      CALL ACCELC(1,IPNELC,IDBS,1,0,0,IERR)
      IF(IERR.NE.0)  GO TO 807
      CALL ACCMAT(1,IPNMAT,IDBS,1,0,0,IERR)
      IF(IERR.NE.0)  GO TO 808
      CALL ACCEPR(1,IPNEPR,IDBS,1,0,0,IERR)
      IF(IERR.NE.0)  GO TO 809
      CALL ACCEEN(1,IPNEEN,IDBL,1,L1,0,0,IERR)
      IF(IERR.NE.0)  GO TO 810
C
```

```
C  LOOP THROUGH THE ELEMENTS
C
      DO 100 I=1,NELM
        IF(I.GT.1)  GO TO 50
C
C     GET INTERNAL LOAD CASE NUMBER
C
        CALL ACCLCS(2,IPNLCS,L1,2,DLCS,IERR)
        IF(IERR.NE.0)  GO TO 805
        ILCN = DLCS(21)
C
C     GET ELEMENT DESCRIPTORS
C
50      CALL ACCELM(2,IPNELM,I,2,IED,IERR)
        IF(IERR.NE.0)  GO TO 800
C
C     BRANCH TO THE APPROPRIATE ELEMENT TYPE
C
        IF(ITYP.EQ.11)  CALL COMP11(NT,I,ILCN)
        IF(ITYP.EQ.5)  CALL COMP05(NT,I,ILCN,PSIBB)
        IF(ITYP.EQ.16)  CALL CP16(NT,I,ILCN,PSIBTB)
C
C
        PSIB16 = PSIB16+PSIBTB(I)
        PSIBC5 = PSIBC5+PSIBB(I)
        SDPSIT = SDPSIT+DPSIT(I)
        SDPSIB = SDPSIB+DPSIB(I)
        SDPSIH = SDPSIH+DPSIH(I)
        SDPSTB = SDPSTB+DPSITB(I)
100   CONTINUE
C
      IF(NT.GT.1)  GO TO 720
      WRITE(10,859)
      DO 710 I=1,NELM
710   WRITE(10,857) I,DPSIT(I),DPSIB(I),DPSIH(I),DPSITB(I)
      WRITE(10,861) SDPSIT,SDPSIB,SDPSIH,SDPSTB
C
720   IF(ITYP.EQ.11)  CALL CPSI11(PSIBT,ILCN,IREF)
C
      PSIB(NT) = PSIBT + PSIBC5 + PSIB16
C
      WRITE(10,858) PSIB(NT)
      PRINT 858, PSIB(NT)
C
C  CLEAN-UP EVERYTHING
C
      CALL ACCELM(4,IPNELM,0,0,0,IERR)
      IF(IERR.NE.0)  GO TO 800
      CALL ACCFES(4,IPNFES,0,0,0,0,0,IERR)
      IF(IERR.NE.0)  GO TO 801
      CALL ACCCND(4,IPNCND,0,0,0,0,0,IERR)
      IF(IERR.NE.0)  GO TO 802
      CALL ACCLCS(4,IPNLCS,0,0,0,IERR)
      IF(IERR.NE.0)  GO TO 805
      CALL ACCNOD(4,IPNNOD,0,0,0,IERR)
      IF(IERR.NE.0)  GO TO 806
      CALL ACCELC(4,IPNELC,0,0,0,0,IERR)
      IF(IERR.NE.0)  GO TO 807
      CALL ACCMAT(4,IPNMAT,0,0,0,0,IERR)
      IF(IERR.NE.0)  GO TO 808
      CALL ACCEPR(4,IPNEPR,0,0,0,0,IERR)
```

```
      IF(IERR.NE.0)  GO TO 809
      CALL ACCEEN(4,IPNEEN,0,0,0,0,0,IERR)
      IF(IERR.NE.0)  GO TO 810
C
      GO TO 820
C
C  WRITE ERROR MESSAGES TO THE SCREEN
C
800   PRINT 870, IERR
      GO TO 820
801   PRINT 871, IERR
      GO TO 820
802   PRINT 872, IERR
      GO TO 820
805   PRINT 875, IERR
      GO TO 820
806   PRINT 877, IERR
      GO TO 820
807   PRINT 878, IERR
      GO TO 820
808   PRINT 879, IERR
      GO TO 820
809   PRINT 876, IERR
      GO TO 820
810   PRINT 880, IERR
      GO TO 820
C
C
820   CONTINUE
C
C
857   FORMAT(I3,4X,4(E16.8,4X))
858   FORMAT(1X,'PSIB=',E16.8)
859   FORMAT(1X,/,1X,'EN',6X,'SENSITIVITY 1',7X,'SENSITIVITY R',
     *7X,'SENSITIVITY H',6X,'SENSITIVITY TB')
861   FORMAT(1X,/,1X,'TOTAL=',4(E16.8,4X))
862   FORMAT(1X,'ELEMENT ',I4)
870   FORMAT(1X,'ACCELM RETURNED WITH ERROR ',I4)
871   FORMAT(1X,'ACCFES RETURNED WITH ERROR ',I4)
872   FORMAT(1X,'ACCCND RETURNED WITH ERROR ',I4)
875   FORMAT(1X,'ACCLCS RETURNED WITH ERROR ',I4)
876   FORMAT(1X,'ACCEPR RETURNED WITH ERROR ',I4)
877   FORMAT(1X,'ACCNOD RETURNED WITH ERROR ',I4)
878   FORMAT(1X,'ACCELC RETURNED WITH ERROR ',I4)
879   FORMAT(1X,'ACCMAT RETURNED WITH ERROR ',I4)
880   FORMAT(1X,'ACCEEN RETURNED WITH ERROR ',I4)
1004  FORMAT(I4)
C
C
C
      RETURN
      END
```

```
      SUBROUTINE COMPO5(NT,I,ILCN,PSIBB)
CP*************************************************************
CP*
CP* COMPO5: CALCULATES COMPLIANCE AND SENSITIVITY FOR A BEAM
CP*
CP*************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*      'COMPO5' CALCULATES THE COMPLIANCE AND THE DESIGN
CP*      SENSITIVITY FOR A 1-D BEAM IN BENDING, WITH AN
CP*      APPLIED ELEMENT FORCE IN #/IN.  SELF WEIGHT IS
CP*      NEGLECTED.  BEAM TORSION HAS BEEN ADDED.
CP*
CP*************************************************************
CP*
CP*   NT       COUNTER FOR FINITE DIFFERENCE
CP*   I        EXTERNAL ELEMENT NO. BEING PROCESSED
CP*   ILCN     INTERNAL LOAD CASE NO.
CP*   NELM     TOTAL NO. OF ELEMENTS
CP*
CP*************************************************************

C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIPN.MON'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
      INCLUDE '[AEGSDR.INC] ELEDES.MON'
      INCLUDE '[AEGSDR.INC] SVECTR.MON'
C
      EQUIVALENCE (NDAT(14),IP)
C
      DIMENSION GPLW(3),PSIBB(500),DDSHPF(12),X(3),Y(3),Z(3),
     *          SHPF(12),SBUF(200),DAIN(50),CFBUF(6),EQBUF(200),
     *          BUF(100),CD(2,6),WTW(3),C(6,2),CDL(2,6),T(3,3),
     *          TB(6,6),COOR(3,3)
C
      DATA GPLW/-.77459667D0, .0D0, .77459667D0/
      DATA WTW/ .55555556D0, .88888889D0, .55555556D0/
      DATA  KT/3/,IREF/1/,MPT/1/
C
      PSIBB(I) = 0.0
      DPSIBG = 0.0
      DPSIHG = 0.0
      IF(I.GT.1)  GO TO 50
C
C  GET APPLIED FORCE IN #/IN.
C
        PRINT *, ' '
        PRINT *, 'ENTER APPLIED FORCE IN #/IN. UNITS'
        READ(5,1001) AF
C
C  GET APPLIED DISTRIBUTED MOMENT
C
        PRINT *,' '
        PRINT *,'ENTER APPLIED DISTRIBUTED MOMENT'
        READ(5,1001) AM
C
C  GET AREA MOMENT OF INERTIA ABOUT Y-AXIS
C
50      CALL ACCEPR(2,IPNEPR,IPTAB,0,BUF,LEN,IERR)
```

```
      IF(IERR.NE.0)  GO TO 809
      YI = BUF(5)
      H = 2.DO*BUF(9)
      B = 2.DO*BUF(10)
      BW(NT) = B
      BH(NT) = H
C
C    GET WEIGHT DENSITY AND MODULUS OF ELASTICITY
C
      GAMMA = 0.0D0
      E = BUF(5)
      V = BUF(7)
      G = E/(2.DO*(1.DO+V))
C
C
C    GET DISPLACEMENTS AT ELEMENT ENDS
C
      DO 150 J=1,NUNPE
        CALL ACCCND(2,IPNCND,INTNN(J),1,ILCN,CFBUF,LEN,IERR)
        IF(IERR.NE.0)  GO TO 802
        DO 150 K=1,NDOF
          CD(J,K) = CFBUF(K)
150   CONTINUE
C
C**********************************************************
C  EVALUATE DISPLS. AND CURVATURE AT THE GAUSS POINT USING
C  SHAPE FUNCTIONS - ONE PT. FOR CURV., THREE PT. FOR DISPL
C**********************************************************

C  GET X, Y, AND Z OF ELEMENT NODES
C
      CALL ACCELC(2,IPNELC,KINT,IREF,BUF,LEN,IERR)
      IF(IERR.NE.0)  GO TO 807
      M = 1
      DO 200 J=1,9,3
      K = J+1
      L = J+2
      X(M) = BUF(J)
      Y(M) = BUF(K)
      Z(M) = BUF(L)
      M = M+1
200   CONTINUE
      DO 210 J=1,3
        COOR(1,J) = X(J)
        COOR(2,J) = Y(J)
        COOR(3,J) = Z(J)
210   CONTINUE
C
C  FORM THE ELEMENT LOCAL COORDINATE SYSTEM
C
      IN3 = INTNN(3)
      CALL EUBTM(IN3,BETA,COOR,T,IERR)
      CALL ZEROSP(TB,36*IP)
      DO 220 J=1,3
        DO 220 K=1,3
          TB(J,K) = T(J,K)
          TB(J+3,K+3) = T(J,K)
220   CONTINUE
      CALL UMXABT(TB,CD,C,6,2,6)
      DO 230 J=1,2
        DO 230 K=1,6
```

```
            CDL(J,K) = C(K,J)
230     CONTINUE
C
C  CALCULATE ELEMENT LENGTH
C
        DX = X(2)-X(1)
        DY = Y(2)-Y(1)
        DZ = Z(2)-Z(1)
        EL = DSQRT(DX*DX+DY*DY+DZ*DZ)
        IF(DY.EQ.0.AND.DZ.EQ.0)  GO TO 236
        DO 235 J=1,NUNPE
          CDL(J,4) = -CDL(J,4)
235     CONTINUE
C
C  CHANGE LOCAL Y-ROTATION FROM POSITIVE TO NEGATIVE
C  IF BEAM LIES ALONG X GLOBAL AXIS
C
236     IF(DX.LT.0.001.AND.DX.GT.-0.001)  GO TO 246
        DO 240  J=1,NUNPE
          CDL(J,5) = -CDL(J,5)
240     CONTINUE
C
246     F = -AF - GAMMA*B*H
C
C  CALCULATE THE TWISTING ANGLE
C
        WXY = DABS((CDL(2,4)-CDL(1,4))/EL)
C
C  EVALUATE SHAPE FUNCTIONS FOR DISPL. - THREE POINT QUADRATURE
C
        B2 = B*B
        B3 = B2*B
        B4 = B3*B
        H2 = H*H
        H3 = H2*H
        WRITE(10,1002) I
        WRITE(10,1004) E,G,V
        WRITE(10,1005) H,B
C
        DO 300 K=1,3
          PSI = GPLW(K)
          CALL EU3DSB(PSI,SHPF,DDSHPF,2,EL)
          W = (SHPF(3)*CDL(1,3)+SHPF(5)*CDL(1,5)
     *        +SHPF(9)*CDL(2,3)+SHPF(11)*CDL(2,5))
C
C  EVALUATE SHAPE FUNCTIONS FOR CURV. - THREE POINT QUADRATURE
C
          WXX = (DDSHPF(3)*CDL(1,3)+DDSHPF(5)*CDL(1,5)
     *          +DDSHPF(9)*CDL(2,3)+DDSHPF(11)*CDL(2,5))
          WRITE(10,860) K,W,WXX,WXY
C
          IF(NT.GT.1)  GO TO 250
C
C  CALCULATE SENSITIVITY VECTORS
C
          PJB = H3/3.D0-.42D0*B*(H2+B4/(4.D0*H2))
          PJH = B*H2-.42D0*B2*(H-B4/(12.D0*H3))
          DPSIBG = DPSIBG+(-2*GAMMA*H*W-(E*H3/12)*WXX*WXX-
     *             PJB*G*WXY*WXY)*WTW(K)*(EL/2.D0)
          DPSIHG = DPSIHG+(-2*GAMMA*B*W-(3*E*B*H2/12)*WXX*WXX-
     *             PJH*G*WXY*WXY)*WTW(K)*(EL/2.D0)
```

```
C
C  CALCULATE PSI(B) - INTEGRAL OF  FORCE*DISPLACEMENT
C
250       PSIBB(I) = PSIBB(I) + (F*W+AM*WXY)*WTW(K)*(EL/2.D0)
300     CONTINUE
        WRITE(10,1003) DPSIBG,DPSIHG
C
        IF(NT.GT.1)  GO TO 820
        DPSIB(I) = DPSIBG
        DPSIH(I) = DPSIHG
C
        GO TO 820
C
C  WRITE ERROR MESSAGES TO THE SCREEN
C
802     PRINT 872, IERR
        GO TO 820
807     PRINT 878, IERR
        GO TO 820
808     PRINT 879, IERR
        GO TO 820
809     PRINT 876, IERR
        GO TO 820
C
C
820     CONTINUE
C
C
851     FORMAT(/,1X,'BEAM WIDTH B=',F8.5,2X,'BEAM DEPTH=',F8.5,2X
       *,'E=',E9.3,2X,'IYY=',E9.3,2X,'GAMMA=',F6.5,'APPLIED FORCE
       *=',F8.5)
855     FORMAT(1X,'NODE=',I2,2X,'X=',E12.5,2X,'Y=',E12.5,2X,'Z=',
       *E12.5,2X,'RX=',E12.5,2X,'RY=',E12.5,2X,'RZ=',E12.5)
860     FORMAT(1X,'GP=',I2,4X,'W=',E11.5,4X,'WXX=',E11.5,4X,
       *'WXY=',E11.5)
870     FORMAT(1X,'ACCELM RETURNED WITH ERROR ',I4)
872     FORMAT(1X,'ACCCND RETURNED WITH ERROR ',I4)
876     FORMAT(1X,'ACCEPR RETURNED WITH ERROR ',I4)
878     FORMAT(1X,'ACCELC RETURNED WITH ERROR ',I4)
879     FORMAT(1X,'ACCMAT RETURNED WITH ERROR ',I4)
1001    FORMAT(E12.5)
1002    FORMAT(1X,'ELEMENT =',I4)
1003    FORMAT(1X,'DPSIBG=',E12.5,4X,'DPSIHG=',E12.5)
1004    FORMAT(1X,'E=',E12.5,4X,'G=',E12.5,4X,'V=',E12.5)
1005    FORMAT(1X,'HEIGHT=',E12.5,4X,'WIDTH=',E12.5)
C
C
C
        RETURN
        END
```

```
      SUBROUTINE COMP11(NT,I,ILCN)
CP**************************************************************
CP*
CP* COMP11: CALCULATES COMPLIANCE AND SENSIT. FOR PLANE STRESS
CP*
CP**************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*       'COMP11' CALCULATES THE COMPLIANCE AND THE DESIGN
CP*       SENSITIVITY OF THE FOUR AND EIGHT NODE PLANE STRESS
CP*       ELEMENT IN TRACTION WITHOUT SELFWEIGHT.
CP*
CP**************************************************************
CP*
CP*   NT      COUNTER FOR FINITE DIFFERENCE
CP*   I       EXTERNAL ELEMENT NO. BEING PROCESSED
CP*   ILCN    INTERNAL LOAD CASE NO.
CP*
CP**************************************************************
C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIPN.MON'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
      INCLUDE '[AEGSDR.INC] ELEDES.MON'
      INCLUDE '[AEGSDR.INC] SVECTR.MON'
C
      DIMENSION X(8),Y(8),Z(8),SHPF(8),GPL(2,4),DATN(50),
     *          BUF(100),PSIB(2),SBUF(50),CFBUF(6),EQBUF(50),
     *          DSHPGX(8),DSHPGY(8),BF(4,48),SE(500),DSHPL(2,8),
     *          SIGMA(6,4),EPSLN(6,4)
C
      DATA  GPL/2*-.57735027, .57735027,-.57735027,
     *          2*.57735027,-.57735027, .57735027/
      DATA  KT/3/,IREF/1/
C
      SE(I) = 0.
      IF(NT.GT.1)  GO TO 350
C
C    GET ELEMENT STRESSES AND STRAINS
C
        CALL ACCFES(2,IPNFES,KINT,IREF,ILCN,SBUF,LEN,IERR)
        IF(IERR.NE.0)  GOTO 801
        LOC = LEN - 1
        M = 1
        DO 50 K=1,NSVAL
          SIGMA(1,K) = SBUF(M)
          SIGMA(2,K) = SBUF(M+1)
          SIGMA(3,K) = SBUF(M+3)
          M = M+4
50      CONTINUE
        DO 60 K=1,NSVAL
          EPSLN(1,K) = SBUF(M)
          EPSLN(2,K) = SBUF(M+1)
          EPSLN(3,K) = SBUF(M+3)
          M = M+4
60      CONTINUE
C
C  GET X AND Y FOR JACOBIAN EVALUATION
C
        CALL ACCELC(2,IPNELC,KINT,IREF,BUF,LENB,IERR)
```

```
      IF(IERR.NE.0)  GO TO 807
      M = 1
      DO 200 L=1,LENB,3
        J = L+1
        K = L+2
        X(M) = BUF(L)
        Y(M) = BUF(J)
        Z(M) = BUF(K)
        M = M+1
200   CONTINUE
C
C
C  CALCULATE FORCES AT THE GAUSS POINTS
C
      DO 250 L=1,NDOF
        DO 250 K=1,NSVAL
          BF(K,L) = 0.0
250   CONTINUE
C
C  LOOP OVER THE GAUSS POINTS
C
      DO 300 K=1,NSVAL
        PSI = GPL(1,K)
        ETA = GPL(2,K)
C
C  EVALUATE SHAPE FUNCTIONS AT THE GAUSS POINTS
C
        IF(ISTYP.EQ.2)  CALL EU2DLQ(PSI,ETA,KT,SHPF,DSHPL,
     *                  DSHPGX,DSHPGY,DETJ,X,Y,IERR)
        IF(ISTYP.EQ.4)  CALL EU2DPQ(PSI,ETA,KT,SHPF,DSHPL,
     *                  DSHPGX,DSHPGY,DETJ,X,Y,IERR)
        IF(IERR.NE.0)  GOTO 809
300   CONTINUE
      WRITE(10,855)
      DO 320 K=1,NSVAL
320   WRITE(10,854) K, (SIGMA(J,K),J=1,NSIG)
      WRITE(10,860)
      DO 330 K=1,NSVAL
330   WRITE(10,854) K, (EPSLN(J,K),J=1,NSIG)
      DO 340 J=1,NSIG
        DO 340 K=1,NSVAL
          SE(I) = SE(I) + SIGMA(J,K)*EPSLN(J,K)*DETJ
340   CONTINUE
C  CALCULATE SENSITIVITY VECTOR
C
      DPSIT(I) = - SE(I)
350   GO TO 820
C
C  WRITE ERROR MESSAGES TO THE SCREEN
C
801   PRINT 871, IERR
      GO TO 820
807   PRINT 878, IERR
      GO TO 820
809   PRINT 876, IERR
      GO TO 820
C
C
820   CONTINUE
C
```

```
C
854    FORMAT(1X,I2,2X,3(E16.8,2X))
855    FORMAT(1X,'GP',5X,'SIGMAX(GP)',8X,'SIGMAY(GP)',8X,
      *'SIGMAXY(GP)')
860    FORMAT(1X,'GP',5X,'EPSLNX(GP)',8X,'EPSLNY(GP)',8X,
      *'EPSLNXY(GP)')
871    FORMAT(1X,'ACCFES RETURNED WITH ERROR ',I4)
876    FORMAT(1X,'EU2DPQ RETURNED WITH ERROR ',I4)
878    FORMAT(1X,'ACCELC RETURNED WITH ERROR ',I4)
C
C
C
       RETURN
       END
```

```
      SUBROUTINE CP16(NT,I,ILCN,PSIBTB)
CP*************************************************************
CP*
CP*  CP16: BRANCHES TO THE APPROPRIATE ELEMENT SUBTYPE
CP*
CP*************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*       'CP16' BRANCHES TO THE APPROPRIATE ELEMENT SUBTYPE
CP*       TO CALCULATE THE COMPLIANCE AND COMPLIANCE DESIGN
CP*       SENSITIVITY OF THE PLATE BENDING ELEMENT 16.
CP*       NOTE: THIS DOES NOT TAKE INTO ACCOUNT ANY MEMBRANE
CP*       STIFFNESS.
CP*
CP*************************************************************
CP*
CP*   NT        COUNTER FOR FINITE DIFFERENCE
CP*   I         EXTERNAL ELEMENT NO. BEING PROCESSED
CP*   ILCN      INTERNAL LOAD CASE NO.
CP*   PSIBTB    COMPLIANCE OF A BENDING PLATE, USED FOR
CP*             CALCULATING THE FINITE DIFFERENCE
CP*
CP*************************************************************
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ELFDES.MON'
      INCLUDE '[AEGSDR.INC] SVECTR.MON'
C
      DIMENSION PSIBTB(500)
C
C** BRANCH TO THE APPROPRIATE ELEMENT SUBTYPE
C
      IF(ISTYP.EQ.1)  CALL CP1601(NT,I,ILCN,PSIBTB)
      IF(ISTYP.EQ.2)  CALL CP1602(NT,I,ILCN,PSIBTB)
C
      RETURN
      END
```

```
      SUBROUTINE CP1601(NT,I,ILCN,PSIBTB)
CP************************************************************
CP*
CP* CP1601: CALCULATES COMPLIANCE AND DESIGN SENSITIVITY
CP*
CP************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*      'CP1601' CALCULATES COMPLIANCE AND DESIGN SENSITIVITY
CP*      FOR A TRIANGULAR BENDING ELEMENT.
CP*
CP************************************************************
CP*
CP*   NT       COUNTER FOR FINITE DIFFERENCE
CP*   I        EXTERNAL ELEMENT NO. BEING PROCESSED
CP*   ILCN     INTERNAL LOAD CASE NO.
CP*   PSIBTB   COMPLIANCE, USED FOR CALCULATING FINITE DIFFERENCE
CP*
CP*************************************************************
C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIPN.MON'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
      INCLUDE '[AEGSDR.INC] ELEDES.MON'
      INCLUDE '[AEGSDR.INC] SVECTR.MON'
C
      DIMENSION X(3),Y(3),PSIB(2),SBUF(100),CPE(500),EF(3,6),
     *          SIGMA(6),EPSLN(6),BUF(100),CFBUF(6),EQBUF(100),
     *          PSIBTB(500),CD(3,6),Z(3)
C
      DATA  KT/3/,IREF/1/,MPT/1/
C
      CPE(I) = 0.
      PSIBTB(I) = 0.
C
C    GET PROPERTIES
C
      CALL ACCEPR(2,IPNEPR,IPTAB,0,BUF,LEN,IERR)
      IF(IERR.NE.0) GO TO 809
      PB(NT) = BUF(25)
      IF(NT.GT.1)  GO TO 65
C
C    GET ELEMENT STRESSES AND STRAINS
C
        CALL ACCFES(2,IPNFES,KINT,IREF,ILCN,SBUF,LEN,IERR)
        IF(IERR.NE.0)  GOTO 801
        M = 1
        DO 50 J=1,NDOF
          SIGMA(J) = SBUF(J)
50      CONTINUE
        M = 7
        DO 60 J=1,NDOF
          EPSLN(J) = SBUF(M)
          M = M+1
60      CONTINUE
C
C  GET DISPLACEMENTS FOR PSI CALCULATION
C
65      DO 70 J=1,NUNPE
          CALL ACCCND(2,IPNCND,INTNN(J),1,ILCN,CFBUF,LEN,IERR)
```

```
          IF(IERR.NE.0)  GO TO 802
          DO 70 K=1,NDOF
            CD(J,K) = CFBUF(K)
70      CONTINUE
C
C  GET EQUIVALENT FORCES AT THE ELEMENT NODES FOR PSI CALC.
C
        CALL ACCEEN(2,IPNEEN,KINT,IREF,ILCN,EQBUF,LEN,IERR)
        IF(IERR.NE.0)  GO TO 808
        M = 1
        DO 80 J=1,NUNPE
          DO 80 K=1,NDOF
            EF(J,K) = EQBUF(M)
            M = M+1
80      CONTINUE
        IF(NT.GT.1)  GO TO 350
C
C  GET THE JACOBIAN
C
        CALL ACCELC(2,IPNELC,KINT,IREF,BUF,LENB,IERR)
        IF(IERR.NE.0)  GO TO 807
        M = 1
        DO 200 J=1,LENB,3
          X(M) = BUF(J)
          Y(M) = BUF(J+1)
          Z(M) = BUF(J+2)
          M = M+1
200     CONTINUE
        DETJ = EUTRIA(X,Y)
C
C
C    CALCULATE SENSITIVITY VECTOR
C
        DO 340 J=1,3
          CPE(I) = CPE(I) + SIGMA(J)*EPSLN(J)*DETJ
340     CONTINUE
        DPSITB(I) = - CPE(I)
C
C  CALCULATE PSI(B) - INTEGRAL OF FORCE*DISPLACEMENT IN Z
C
350   DO 400 J=1,NUNPE
        PSIBTB(I) = PSIBTB(I) + EF(J,3)*CD(J,3)
400   CONTINUE
      GO TO 820
C
C  WRITE ERROR MESSAGES TO THE SCREEN
C
801   PRINT 871, IERR
      GO TO 820
802   PRINT 872, IERR
      GO TO 820
807   PRINT 877, IERR
      GO TO 820
808   PRINT 879, IERR
      GO TO 820
809   PRINT 880,  IERR
      GO TO 820
C
C
820   CONTINUE
C
```

```
C
852   FORMAT(1X,'NODE',6X,'DISP X',8X,'DISP Y',8X,'DISP '
     *        ,8X,'ROT X',9X,'ROT Y',9X,'ROT Z')
853   FORMAT(1X,I4,6(2X,E12.4))
871   FORMAT(1X,'ACCFES RETURNED WITH ERROR ',I4)
872   FORMAT(1X,'ACCCND RETURNED WITH ERROR ',I4)
877   FORMAT(1X,'ACCELC RETURNED WITH ERROR ',I4)
879   FORMAT(1X,'ACCEEN RETURNED WITH ERROR ',I4)
880   FORMAT(1X,'ACCEPR RETURNED WITH ERROR ',I4)
C
C
C
      RETURN
      END
```

```
      SUBROUTINE CP1602(NT,I,ILCN,PSIBTB)
CP*************************************************************
CP*
CP* CP1602: CALCULATES COMPLIANCE AND DESIGN SENSITIVITY
CP*
CP*************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*       'CP1602' CALCULATES COMPLIANCE AND DESIGN SENSITIVITY
CP*       FOR A FOUR NODE BENDING ELEMENT.
CP*
CP*************************************************************
CP*
CP*   NT        COUNTER FOR FINITE DIFFERENCE
CP*   I         EXTERNAL NO. BEING PROCESSED
CP*   ILCN      INTERNAL LOAD CASE NO.
CP*   PSIBTB    COMPLIANCE
CP*
CP*************************************************************

C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIPN.MON'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
      INCLUDE '[AEGSDR.INC] ELEDES.MON'
      INCLUDE '[AEGSDR.INC] SVECTR.MON'
C
      DIMENSION X(4),Y(4),PSIB(2),SBUF(100),CPE(500),EF(4,6),Z(4),
     *          SIGMA(6,4),EPSLN(6,4),BUF(100),CFBUF(6),
     *          EQBUF(100),PSIBTB(500),CD(4,6)
C
      DATA  KT/3/,IREF/1/,MPT/1/
C
      CPE(I) = 0.
      PSIBTB(I) = 0.
C
C    GET PROPETIES
C
      CALL ACCEPR(2,IPNEPR,IPTAB,0,BUF,LEN,IERR)
      IF(IERR.NE.0)  GO TO 809
      PB(NT) = BUF(25)
      IF(NT.GT.1)  GO TO 65
C
C    GET ELEMENT STRESSES AND STRAINS
C
        CALL ACCFES(2,IPNFES,KIND,IREF,ILCN,SBUF,LEN,IERR)
        IF(IERR.NE.0)  GOTO 801
        M = 1
        DO 50 K=1,NSVAL
          J = M+1
          L = M+2
          SIGMA(1,K) = SBUF(M)
          SIGMA(2,K) = SBUF(J)
          SIGMA(3,K) = SBUF(L)
          M = M+6
50      CONTINUE
        DO 60 K=1,NSVAL
          J = M+1
          L = M+2
          EPSLN(1,K) = SBUF(M)
```

```
          EPSLN(2,K) = SBUF(J)
          EPSLN(3,K) = SBUF(L)
          M = M+6
60      CONTINUE
C
C  GET DISPLACEMENTS FOR PSI CALCULATION
C
65      DO 70 J=1,NUNPE
          CALL ACCCND(2,IPNCND,INTNN(J),1,ILCN,CFBUF,LEN,IERR)
          IF(IERR.NE.0)  GO TO 802
          DO 70 K=1,NDOF
            CD(J,K) = CFBUF(K)
70      CONTINUE
C
C  GET EQUIVALENT FORCES AT THE ELEMENT NODES FOR PSI CALC.
C
        CALL ACCEEN(2,IPNEEN,KINT,IREF,ILCN,EQBUF,LEN,IERR)
        IF(IERR.NE.0)  GO TO 808
        M = 1
        DO 80 J=1,NUNPE
          DO 80 K=1,NDOF
            EF(J,K) = EQBUF(M)
            M = M+1
80      CONTINUE
        IF(NT.GT.1)  GO TO 350
C
C  GET THE JACOBIAN
C
        CALL ACCELC(2,IPNELC,KINT,IREF,BUF,LENB,IERR)
        IF(IERR.NE.0)  GO TO 807
        M = 1
        DO 200 J=1,LENB,3
          X(M) = BUF(J)
          Y(M) = BUF(J+1)
          Z(M) = BUF(J+2)
          M = M+1
200     CONTINUE
        DETJ = AREAQ(X,Y)
        DETJ = DETJ/4.D0
C
C
C     CALCULATE SENSITIVITY VECTOR
C
        DO 340 J=1,3
          DO 340 K=1,NSVAL
            CPE(I) = CPE(I) + SIGMA(J,K)*EPSLN(J,K)*DETJ
340     CONTINUE
        DPSITB(I) = - CPE(I)
C
C  CALCULATE PSI(B) - INTEGRAL OF FORCE*DISPLACEMENT IN Z
C
350   DO 400 J=1,NUNPE
        PSIBTB(I) = PSIBTB(I) + EF(J,3)*CD(J,3)
400   CONTINUE
      GO TO 820
C
C  WRITE ERROR MESSAGES TO THE SCREEN
C
801   PRINT 871, IERR
      GO TO 820
802   PRINT 872, IERR
```

```
      GO TO 820
807   PRINT 877, IERR
      GO TO 820
808   PRINT 879, IERR
      GO TO 820
809   PRINT 880, IERR
      GO TO 820
C
C
820   CONTINUE
C
C
852   FORMAT(1X,'NODE',6X,'DISP X',8X,'DISP Y',8X,'DISP '
     *        ,8X,'ROT X',9X,'ROT Y',9X,'ROT Z')
853   FORMAT(1X,I4,6(2X,E12.4))
871   FORMAT(1X,'ACCFES RETURNED WITH ERROR ',I4)
872   FORMAT(1X,'ACCCND RETURNED WITH ERROR ',I4)
877   FORMAT(1X,'ACCELC RETURNED WITH ERROR ',I4)
879   FORMAT(1X,'ACCEEN RETURNED WITH ERROR ',I4)
880   FORMAT(1X,'ACCEPR RETURNED WITH ERROR ',I4)
C
C
C
      RETURN
      END
```

```
      SUBROUTINE CPSI11(PSIBT,IL1,IREF)
C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIFN.MON'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
      INCLUDE '[AEGSDR.INC] FLEDES.MON'
C
      DIMENSION DATN(50),CFBUF(6)
C
C  CALCULATE PSI(B) FOR PLATE - FORCE*DISPLACEMENT
C
      PRINT *,' '
      PRINT *,'ENTER NODE WHERE LOAD IS APPLIED'
      READ(5,1004) NEXT
      PRINT *,' '
      PRINT *,'ENTER LOAD DIRECTION(1:X,...,6:RZ)'
      READ(5,1004) LDIR
C
C  GET INTERNAL NODE NUMBER
C
      CALL ACCNOD(2,IFNNOD,NEXT,2,DATN,IERR)
      IF(IERR.NE.0)  GO TO 806
      NINT = DATN(4)
C
C  GET DISPLACEMENT AT NODE
C
      CALL ACCCND(2,IFNCND,NINT,IREF,IL1,CFBUF,LEN,IERR)
      IF(IERR.NE.0)  GO TO 802
      DISP = CFBUF(LDIR)
C
      PSIBT = 40000*DISP
C
C  WRITE ERROR MESSAGES
C
802   PRINT 872, IERR
      GO TO 820
806   PRINT 877, IERR
      GO TO 820
C
820   CONTINUE
C
C
872   FORMAT(1X,'ACCCND RETURNED WITH ERROR ',I4)
877   FORMAT(1X,'ACCNOD RETURNED WITH ERROR ',I4)
1004  FORMAT(I4)
C
C
      RETURN
      END
```

```
      SUBROUTINE DISP(FSIB,NT,NELM)
CP*************************************************************
CP*
CP* DISP: BRANCHES TO THE APPROPRIATE ELEMENT TYPE FOR THE
CP*
CP*************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*        'DISP' BRANCHES TO THE APPROPRIATE ELEMENT TYPE FOR
CP*        CALCULATION OF THE DISPLACEMENT CONSTRAINT AND THE
CP*        DISPLACEMENT SENSITIVITY VECTOR.
CP*
CP*************************************************************
CP*
CP*   FSIB       DISPLACEMENT AT X
CP*   NT         COUNTER FOR FINITE DIFFERENCE
CP*   NELM       TOTAL NO. OF ELEMENTS
CP*
CP*************************************************************
C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIPN.MON'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
      INCLUDE '[AEGSDR.INC] ELEDES.MON'
      INCLUDE '[AEGSDR.INC] SVECTR.MON'
C
      EQUIVALENCE (NDAT(97),IDBS),(NDAT(98),IDBL)
C
      DIMENSION DATN(50),FSIB(2),CFBUF(6)
C
      DATA IREF/1/
C
      SDFSTB = 0.0
      SDFSIT = 0.0
      SDFSIB = 0.0
      SDFSIH = 0.0
      IF(NT.GT.1)  GO TO 10
C
C
C  SET ADJOINT LOAD CASE NUMBER
C
      L2 = LCS + NC
C
C  SET ORIGINAL LOAD CASE NUMBER
C
10    L1 = 1
C
C
C  SETUP POINTERS
C
      CALL ACCELM(1,IPNELM,IDBS,1,0,IERR)
      IF(IERR.NE.0)  GO TO 800
      CALL ACCFES(1,IPNFES,IDBL,1,L1,0,0,IERR)
      IF(IERR.NE.0)  GO TO 802
      CALL ACCCND(1,IPNCND,IDBL,1,L1,0,0,IERR)
      IF(IERR.NE.0)  GO TO 802
      CALL ACCNOD(1,IPNNOD,IDBS,1,0,IERR)
      IF(IERR.NE.0)  GO TO 806
      CALL ACCELC(1,IPNELC,IDBS,1,0,0,IERR)
      IF(IERR.NE.0)  GO TO 807
```

```
      CALL ACCMAT(1,IPNMAT,IDBS,1,0,0,IERR)
      IF(IERR.NE.0)  GO TO 808
      CALL ACCEPR(1,IPNEPR,IDBS,1,0,0,IERR)
      IF(IERR.NE.0)  GO TO 809
C
C  LOOP THROUGH THE BUFFERS TO GET STRESSES, STRAINS ,MOMENTS,ETC.
C
      DO 100 I=1,NELM
C
C  GET ELEMENT DESCRIPTORS
C
        CALL ACCELM(2,IPNELM,I,IREF,IED,IERR)
        IF(IERR.NE.0)  GO TO 800
C
        IF(NT.GT.1)  GO TO 720
C
C    BRANCH TO THE APPROPRIATE ELEMENT TYPE
C
        IF(ITYP.EQ.11)  CALL DISP11(NT,I,L1,L2,IL1)
        IF(ITYP.EQ.5)  CALL DISP05(NT,I,L1,L2,IL1)
        IF(ITYP.EQ.16)  CALL DP16(NT,I,L1,L2,IL1)
C
C
        SDPSIT = SDPSIT+DPSIT(I)
        SDPSIB = SDPSIB+DPSIB(I)
        SDPSIH = SDPSIH+DPSIH(I)
        SDPSTB = SDPSTB+DPSITB(I)
100   CONTINUE
C
      WRITE(10,859)
      DO 710 I=1,NELM
710   WRITE(10,857) I,DPSIT(I),DPSIB(I),DPSIH(I),DPSITB(I)
      WRITE(10,861) SDPSIT,SDPSIB,SDPSIH,SDPSTB
C
C
C  CALCULATE PSI(B) - ORIGINAL DISPL AT NODE WHERE ADJT LOAD APPLIED
C
720   PRINT *, ' '
      PRINT *, 'ENTER NODE NUMBER WHERE ADJT LOAD IS APPLIED'
      READ(5,1004) NEXT
      LDIR = 3
C
C  GET INTERNAL NODE NUMBER
C
      CALL ACCNOD(2,IPNNOD,NEXT,2,DATN,IERR)
      IF(IERR.NE.0)  GO TO 806
      NINT = DATN(4)
C
C  GET DISPLACEMENT AT NODE
C
      CALL ACCCND(2,IPNCND,NINT,IREF,IL1,CFBUF,LEN,IERR)
      IF(IERR.NE.0)  GO TO 802
      DIS = CFBUF(LDIR)
C
      PSIB(NT) = -DIS
C
      WRITE(10,858)  PSIB(NT)
      PRINT 858, PSIB(NT)
C
      IF(NT.EQ.1)  GO TO 730
      WRITE(10,*)
```

```
      WRITE(10,856) NEXT
C
C
C  CLEAN-UP EVERYTHING
C
730   CALL ACCELM(4,IPNELM,0,0,0,IERR)
      IF(IERR.NE.0)  GO TO 800
      CALL ACCFES(4,IPNFES,0,0,0,0,0,IERR)
      IF(IERR.NE.0)  GO TO 801
      CALL ACCCND(4,IPNCND,0,0,0,0,0,IERR)
      IF(IERR.NE.0)  GO TO 802
      IF(NT.GT.1)  GO TO 750
      CALL ACCLCS(4,IPNLCS,0,0,0,IERR)
      IF(IERR.NE.0)  GO TO 805
750   CALL ACCNOD(4,IPNNOD,0,0,0,IERR)
      IF(IERR.NE.0)  GO TO 806
      CALL ACCELC(4,IPNELC,0,0,0,0,IERR)
      IF(IERR.NE.0)  GO TO 807
      CALL ACCMAT(4,IPNMAT,0,0,0,0,IERR)
      IF(IERR.NE.0)  GO TO 808
      CALL ACCEPR(4,IPNEPR,0,0,0,0,IERR)
      IF(IERR.NE.0)  GO TO 809
C
      GO TO 820
C
C  WRITE ERROR MESSAGES TO THE SCREEN
C
800   PRINT 870, IERR
      GO TO 820
801   PRINT 871, IERR
      GO TO 820
802   PRINT 872, IERR
      GO TO 820
805   PRINT 875, IERR
      GO TO 820
806   PRINT 877, IERR
      GO TO 820
807   PRINT 878, IERR
      GO TO 820
808   PRINT 879, IERR
      GO TO 820
809   PRINT 876, IERR
      GO TO 820
C
C
820   CONTINUE
C
C
856   FORMAT(1X,'***ADJOINT LOAD IS APPLIED AT NODE',I4)
857   FORMAT(I3,4X,4(E16.8,4X))
858   FORMAT(1X,'PSIB=',E16.8)
859   FORMAT(1X,/,1X,'EN',6X,'SENSITIVITY T',7X,'SENSITIVITY H',
     *7X,'SENSITIVITY H',6X,'SENSTIVITY TH')
861   FORMAT(1X,/,1X,'TOTAL=',4(E16.8,4X))
862   FORMAT(1X,'ELEMENT ',I4)
870   FORMAT(1X,'ACCELM RETURNED WITH ERROR ',I4)
871   FORMAT(1X,'ACCFES RETURNED WITH ERROR ',I4)
872   FORMAT(1X,'ACCCND RETURNED WITH ERROR ',I4)
875   FORMAT(1X,'ACCLCS RETURNED WITH ERROR ',I4)
876   FORMAT(1X,'ACCEPR RETURNED WITH ERROR ',I4)
877   FORMAT(1X,'ACCNOD RETURNED WITH ERROR ',I4)
```

```
878   FORMAT(1X,'ACCELC RETURNED WITH ERROR ',I4)
879   FORMAT(1X,'ACCMAT RETURNED WITH ERROR ',I4)
1004  FORMAT(I4)
2000  FORMAT(A)
C
C
C
      RETURN
      END
```

```
      SUBROUTINE DISP05(NT,I,L1,L2,IL1)
CP************************************************************
CP*
CP* DISP05: CALCULATES THE DISPL. DESIGN SENSITIVITY OF A BEAM
CP*
CP************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*       'DISP05' CALCULATES THE DISPLACEMENT CONSTRAINT DESIGN
CP*       SENSITIVITY OF A 1-D BEAM IN BENDING, WITH AN APPLIED
CP*       ELEMENT FORCE IN #/IN.  SELF WEIGHT IS NEGLECTED.
CP*       TORSION IS ACTIVE.
CP*
CP************************************************************
CP*
CP*   NT      COUNTER FOR FINITE DIFFERENCE
CP*   I       EXTERNAL ELEMENT NO. BEING PROCESSED
CP*   L1      ORIGINAL EXTERNAL LOAD CASE NO.
CP*   L2      EXTERNAL ADJOINT LOAD CASE NO.
CP*   IL1     INTERNAL LOAD CASE NO. OF ORIGINAL LOAD
CP*
CP************************************************************
C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIPN.MON'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
      INCLUDE '[AEGSDR.INC] ELEDES.MON'
      INCLUDE '[AEGSDR.INC] SVECTR.MON'
      COMMON/LCSDES/DLCS(90)
C
      EQUIVALENCE (NDAT(14),IP),(NDAT(98),IDBL)
      DIMENSION X(3),Y(3),Z(3),DDSHPF(12),DATN(50),BUF(100),
     *          SHPF(12),CFBUF(6),GPLW(3),ALD(2,6),CD(2,6),WTW(3),
     *          C(6,2),D(6,2),T(3,3),TB(6,6),CDL(2,6),ALDL(2,6),
     *          COOR(3,3)
C
      DATA  GPLW/-.77459667, .00000000, .77459667/
      DATA WTW/ .555555556, .88888889, .55555556/
      DATA  KT/3/,IREF/1/,MPT/1/
C
      DPSIBG = 0.0
      DPSIHG = 0.0
      IF(I.GT.1)  GO TO 50
C
C  REQUEST APPLIED FORCE IN LOAD/LENGTH
C
        PRINT *,' '
        PRINT *,'ENTER APPLIED LOAD IN [FORCE/LENGTH]'
        READ(5,1001) AF
C
C  GET AREA MOMENT OF INERTIA ABOUT Y-AXIS
C
50      CALL ACCEPR(2,IPNEPR,IPTAB,0,BUF,LEN,IERR)
        IF(IERR.NE.0)  GO TO 809
        YI = BUF(5)
        H = 2*BUF(9)
        B = 2*BUF(10)
        BH(NT) = H
        BW(NT) = B
C
```

```
C     GET WEIGHT DENSITY AND MODULUS OF ELASTICITY
C
         CALL ACCMAT(2,IPNMAT,NMAT,MPT,BUF,LEN,IERR)
         IF(IERR.NE.0)  GO TO 808
         GAMMA = 0.0D0
         E = BUF(5)
         V = BUF(7)
         G = E/(2.D0*(1.D0+V))
C
         IF(I.GT.1)  GO TO 60
C
C   GET INTERNAL LOAD CASE NUMBER FOR ORIGINAL LOAD
C
60       CALL ACCLCS(1,IPNLCS,IDBL,1,0,IERR)
         IF(IERR.NE.0)  GO TO 805
         CALL ACCLCS(2,IPNLCS,L1,2,DLCS,IERR)
         IF(IERR.NE.0)  GO TO 805
         IL1 = DLCS(21)
C
C     GET DISPLACEMENTS AT ELEMENT ENDS
C
        CALL ACCCND(1,IPNCND,IDBL,1,L1,0,0,IERR)
        IF(IERR.NE.0)  GO TO 802
        DO 70 J=1,NUNPE
           CALL ACCCND(2,IPNCND,INTNN(J),1,IL1,CFBUF,LEN,IERR)
           IF(IERR.NE.0)  GO TO 802
           DO 70 K=1,NDOF
             CD(J,K) = CFBUF(K)
70       CONTINUE
C
C   GET INTERNAL LOAD CASE NUMBER FOR ADJOINT LOAD
C
         CALL ACCLCS(1,IPNLCS,IDBL,1,0,IERR)
         IF(IERR.NE.0)  GO TO 805
         CALL ACCLCS(2,IPNLCS,L2,2,DLCS,IERR)
         IF(IERR.NE.0)  GO TO 805
         ILCN = DLCS(21)
C
C   GET DISPLACEMENTS AT ELEMENT ENDS FOR ADJOINT LOAD
C
         CALL ACCCND(1,IPNCND,IDBL,1,L2,0,0,IERR)
         IF(IERR.NE.0)  GO TO 802
         DO 100 J=1,NUNPE
           CALL ACCCND(2,IPNCND,INTNN(J),1,ILCN,CFBUF,LEN,IERR)
           IF(IERR.NE.0)  GO TO 802
           DO  100 K=1,NDOF
             ALD(J,K) = CFBUF(K)
100      CONTINUE
C
C**********************************************************
C  EVALUATE DISPLS. AND CURVATURE AT THE GAUSS POINT USING
C  SHAPE FUNCTIONS - ONE PT. FOR CURV., THREE PT. FOR DISPL
C**********************************************************

C  GET X, Y, AND Z OF ELEMENT NODES
C
        CALL ACCELC(2,IPNELC,KINT,IREF,BUF,LEN,IERR)
        IF(IERR.NE.0)  GO TO 807
        M = 1
        DO 200 J=1,9,3
        K = J+1
```

```
      L = J+2
      X(M) = BUF(J)
      Y(M) = BUF(K)
      Z(M) = BUF(L)
      M = M+1
200   CONTINUE
      DO 210 J=1,3
        COOR(1,J) = X(J)
        COOR(2,J) = Y(J)
        COOR(3,J) = Z(J)
210   CONTINUE
C
C  FORM THE ELEMENT LOCAL COORDINATE SYSTEM FOR DISPLACEMENTS
C
      IN3 = INTNN(3)
      CALL EUBTM(IN3,BETA,COOR,T,IERR)
      CALL ZEROSP(TB,36*IP)
      DO 220 J=1,3
        DO 220 K=1,3
          TB(J,K) = T(J,K)
          TB(J+3,K+3) = T(J,K)
220   CONTINUE
      CALL UMXABT(TB,CD,C,6,2,6)
      DO 230 J=1,2
        DO 230 K=1,6
          CDL(J,K) = C(K,J)
230   CONTINUE
      CALL UMXABT(TB,ALD,D,6,2,6)
      DO 232 J=1,2
        DO 232 K=1,6
          ALDL(J,K) = D(K,J)
232   CONTINUE
C
C  CALCULATE ELEMENT LENGTH
C
      DX = X(2)-X(1)
      DY = Y(2)-Y(1)
      DZ = Z(2)-Z(1)
      EL = DSQRT(DX*DX+DY*DY+DZ*DZ)
C
C  CHANGE LOCAL Y-ROTATION FROM POSITIVE TO NEGATIVE IF
C  BEAM LIES ALONG X GLOBAL AXIS
C
      IF(DX.LT.0.001.AND.DX.GT.-0.001)  GO TO 246
      DO 240 J=1,NUNPE
        CDL(J,5) = -CDL(J,5)
        ALDL(J,5) = -ALDL(J,5)
240   CONTINUE
C
246   F = -AF - GAMMA*B*H
C
C  CALCULATE THE TWISTING ANGLES
C
      WXY = DABS((CDL(2,4)-CDL(1,4))/EL)
      AWXY = DABS((ALDL(2,4)-ALDL(1,4))/EL)
C
C  EVALUATE SHAPE FUNCTIONS FOR DISPL. - THREE POINT QUADRATURE
C
      B2 = B*B
      B3 = B2*B
      B4 = B3*B
```

```
          H2 = H*H
          H3 = H2*H
C
          DO 300 K=1,3
            PSI = GPLW(K)
            CALL EU3DSB(PSI,SHPF,DDSHPF,2,EL)
            W = (SHPF(3)*CDL(1,3)+SHPF(5)*CDL(1,5)+SHPF(9)*
     *          CDL(2,3)+SHPF(11)*CDL(2,5))
            AW = (SHPF(3)*ALDL(1,3)+SHPF(5)*ALDL(1,5)+SHPF(9)*
     *           ALDL(2,3)+SHPF(11)*ALDL(2,5))
C
C  EVALUATE SHAPE FUNCTIONS FOR CURV. - THREE POINT QUADRATURE
C
            WXX = (DDSHPF(3)*CDL(1,3)+DDSHPF(5)*CDL(1,5)+
     *            DDSHPF(9)*CDL(2,3)+DDSHPF(11)*CDL(2,5))
            AWXX = (DDSHPF(3)*ALDL(1,3)+DDSHPF(5)*ALDL(1,5)+
     *             DDSHPF(9)*ALDL(2,3)+DDSHPF(11)*ALDL(2,5))
C
C  CALCULATE SENSITIVITY VECTORS
C
            PJB = H3/3.D0-.42D0*B*(H2+B4/(4.D0*H2))
            PJH = B*H2-.42D0*B2*(H-B4/(12.D0*H3))
            DPSIBG = DPSIBG+(-GAMMA*H*AW-(E*H3/12)*AWXX*WXX-
     *               PJB*G*WXY*AWXY)*WTW(K)*(EL/2.D0)
            DPSIHG = DPSIHG+(-GAMMA*B*AW-(3*E*B*H2/12)*AWXX*WXX-
     *               PJH*G*WXY*AWXY)*WTW(K)*(EL/2.D0)
300    CONTINUE
       DPSIB(I) = DPSIBG
       DPSIH(I) = DPSIHG
C
       GO TO 820
C
C  WRITE ERROR MESSAGES TO THE SCREEN
C
802    PRINT 872, IERR
       GO TO 820
805    PRINT 875, IERR
       GO TO 820
807    PRINT 878, IERR
       GO TO 820
808    PRINT 879, IERR
       GO TO 820
809    PRINT 876, IERR
       GO TO 820
C
C
820    CONTINUE
C
C
851    FORMAT(/,1X,'BEAM WIDTH B=',F8.5,2X,'BEAM DEPTH=',F8.5,2X
     *,'E=',E9.3,2X,'IYY=',E9.3,2X,'GAMMA=',F6.5,2X,'APPLIED
     *FORCE=',F8.5)
855    FORMAT(1X,'NODE=',I2,2X,'X=',E12.5,2X,'Y=',E12.5,2X,'Z=',
     *E12.5,2X,'RX=',E12.5,2X,'RY=',E12.5,2X,'RZ=',E12.5)
860    FORMAT(1X,'GP=',I2,4X,'W=',E11.5,4X,'WXX=',E11.5,4X,'AW=',
     *E11.5,4X,'AWXX=',E11.5)
864    FORMAT(1X,'NODE=',I2,2X,'AX=',E12.5,2X,'AY=',E12.5,2X,
     *'AZ=',E12.5,2X,'ARX=',E12.5,2X,'ARY=',E12.5,2X,'ARZ=',
     *E12.5)
872    FORMAT(1X,'ACCCND RETURNED WITH ERROR ',I4)
875    FORMAT(1X,'ACCLCS RETURNED WITH ERROR ',I4)
```

```
876    FORMAT(1X,'ACCEPR RETURNED WITH ERROR ',I4)
878    FORMAT(1X,'ACCELC RETURNED WITH ERROR ',I4)
879    FORMAT(1X,'ACCMAT RETURNED WITH ERROR ',I4)
1001   FORMAT(E12.5)
C
C
       RETURN
       END
```

```
      SUBROUTINE DISP11(NT,I,L1,L2,IL1)
CP************************************************************
CP*
CP* DISP11: CLACULATES THE DISPL. SENSIT. FOR PLANE STRESS
CP*
CP************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*       'DISP11' CALCULATES THE DISPLACEMENT CONSTRAINT DESIGN
CP*       SENSITIVITY FOR THE FOUR AND EIGHT NODE PLANE STRESS
CP*       ELEMENT WITH TRACTION, SELF WEIGHT NOT INCLUDED.
CP*
CP************************************************************
CP*
CP*   NT      COUNTER FOR FINITE DIFFERENCE
CP*   I       EXTERNAL ELEMENT NO. BEING PROCESSED
CP*   L1      ORIGINAL EXTERNAL LOAD CASE NO.
CP*   L2      EXTERNAL ADJOINT LOAD CASE NO.
CP*   IL1     ORIGINAL INTERNAL LOAD CASE NO.  RETURNED TO DISP.FOR
CP*
CP************************************************************
C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIPN.MON'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
      INCLUDE '[AEGSDR.INC] ELEDES.MON'
      INCLUDE '[AEGSDR.INC] SVECTR.MON'
      COMMON/LCSDES/DLCS(90)
C
      EQUIVALENCE (NDAT(98),IDBL)
C
      DIMENSION SHPF(8),GPL(2,4),BUF(100),X(8),Y(8),Z(8),
     *          DATN(50),PSIB(2),SBUF(50),DSHPGX(8),DSHPGY(8),
     *          DSHPL(2,8),SIGMA(6,4),EPSLN(6,4),BF(4,48),SE(500)
C
      DATA  GPL/2*-.57735027, .57735027,-.57735027,
     *          2*.57735027,-.57735027, .57735027/
      DATA  KT/3/,IREF/1/,MPT/1/
C
        JJ = L1
C
C     GET INTERNAL LOAD CASE NUMBER
C
10      CALL ACCLCS(1,IPNLCS,IDBL,1,0,IERR)
        IF(IERR.NE.0)  GO TO 805
        CALL ACCLCS(2,IPNLCS,JJ,2,DLCS,IERR)
        IF(IERR.NE.0)  GO TO 805
        ILCN = DLCS(21)
C
C     SETUP POINTER FOR STRESS-STRAIN BUFFER
C
        CALL ACCFES(1,IPNFES,IDBL,1,ILCN,0,0,IERR)
        IF (IERR.NE.0)  GO TO 801
        IF(JJ.EQ.L1)  IL1 = ILCN
        SE(I) = 0.0
C
C     GET ELEMENT STRESSES AND STRAINS
C
        CALL ACCFES(2,IPNFES,KINT,IREF,ILCN,SBUF,LEN,IERR)
        IF(IERR.NE.0)  GOTO 801
```

```
        LOC = LEN - 1
        M = 1
        IF (JJ.EQ.L2)  GO TO 55
        DO 50 K=1,NSVAL
          SIGMA(1,K) = SBUF(M)
          SIGMA(2,K) = SBUF(M+1)
          SIGMA(3,K) = SBUF(M+3)
          M = M+4
50      CONTINUE
        GO TO 65
55      M = 17
        DO 60 K=1,NSVAL
          EPSLN(1,K) = SBUF(M)
          EPSLN(2,K) = SBUF(M+1)
          EPSLN(3,K) = SBUF(M+3)
          M = M+4
60      CONTINUE
65      IF (JJ.EQ.L2)  GO TO 100
        JJ = L2
        GO TO 10
C
C  GET X AND Y FOR JACOBIAN EVALUATION
C
100     CALL ACCELC(2,IPNELC,KINT,IREF,BUF,LENB,IERR)
        IF(IERR.NE.0)  GO TO 807
        M = 1
        DO 200 L=1,LENB,3
          X(M) = BUF(L)
          Y(M) = BUF(L+1)
          Z(M) = BUF(L+2)
          M = M+1
200     CONTINUE
C
C  CALCULATE FORCES AT THE GAUSS POINTS
C
        DO 250 L=1,NDOF
          DO 250 K=1,NSVAL
            BF(K,L) = 0.0
250     CONTINUE
C
C  LOOP OVER THE GAUSS POINTS
C
        DO 300 K=1,NSVAL
          PSI = GPL(1,K)
          ETA = GPL(2,K)
C
C  EVALUATE SHAPE FUNCTIONS AT THE GAUSS POINTS
C
          IF(ISTYP.EQ.2)  CALL EU2DLQ(PSI,ETA,KT,SHPF,DSHPL,
     *                    DSHPGX,DSHPGY,DETJ,X,Y,IERR)
          IF(ISTYP.EQ.4)  CALL EU2DPQ(PSI,ETA,KT,SHPF,DSHPL,
     *                    DSHPGX,DSHPGY,DETJ,X,Y,IERR)
          IF(IERR.NE.0)  GOTO 809
300     CONTINUE
        DO 340 J=1,NSIG
          DO 340 K=1,NSVAL
            SE(I) = SE(I) + SIGMA(J,K)*EPSLN(J,K)*DETJ
340     CONTINUE
C
C  CALCULATE SENSITIVITY VECTOR
C
```

```
        DPSIT(I) = - SE(I)
700   CONTINUE
C
      GO TO 820
C  WRITE ERROR MESSAGES TO THE SCREEN
C
801   PRINT 871, IERR
      GO TO 820
805   PRINT 875, IERR
      GO TO 820
807   PRINT 878, IERR
      GO TO 820
809   PRINT 876, IERR
      GO TO 820
C
C
820   CONTINUE
C
C
854   FORMAT(1X,I2,2X,3(E16.8,2X))
855   FORMAT(1X,'GP',5X,'SIGMAX(GP)',8X,'SIGMAY(GP)',8X,
     *'SIGMAXY(GP)')
860   FORMAT(1X,'GP',5X,'EPSLNX(GP)',8X,'EPSLNY(GP)',8X,
     *'EPSLNXY(GP)')
871   FORMAT(1X,'ACCFES RETURNED WITH ERROR ',I4)
875   FORMAT(1X,'ACCLCS RETURNED WITH ERROR ',I4)
876   FORMAT(1X,'EU2DPQ RETURNED WITH ERROR ',I4)
878   FORMAT(1X,'ACCELC RETURNED WITH ERROR ',I4)
C
C
      RETURN
      END
```

```
      SUBROUTINE DP16(NT,I,L1,L2,IL1)
CP*****************************************************************
CP*
CP*  CP16: BRANCHES TO THE APPROPRIATE ELEMENT SUBTYPE
CP*
CP*****************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*      'CP16' BRANCHES TO THE APPROPRIATE ELEMENT SUBTYPE
CP*      TO CALCULATE THE DISPLACEMENT DESIGN SENSITIVITY OF
CP*      THE PLATE BENDING ELEMENT 16.
CP*      NOTE: THIS DOES NOT TAKE INTO ACCOUNT ANY MEMBRANE
CP*      STIFFNESS.
CP*
CP*****************************************************************
CP*
CP*   NT      COUNTER FOR FINITE DIFFERENCE
CP*   I       EXTERNAL ELEMENT NO. BEING PROCESSED
CP*   L1      ORIGINAL EXTERNAL LOAD CASE NO.
CP*   L2      EXTERNAL ADJOINT LOAD CASE NO.
CP*   IL1     ORIGINAL INTERNAL LOAD CASE NO. - RETURNED VALUE
CP*
CP*****************************************************************
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ELEDES.MON'
      INCLUDE '[AEGSDR.INC] SVECTR.MON'
C
C** BRANCH TO THE APPROPRIATE ELEMENT SUBTYPE
C
      IF(ISTYP.EQ.1)  CALL DP1601(NT,I,L1,L2,IL1)
      IF(ISTYP.EQ.2)  CALL DP1602(NT,I,L1,L2,IL1)
C
      RETURN
      END
```

```
      SUBROUTINE DP1601(NT,I,L1,L2,IL1)
CP***********************************************************
CP*
CP* DP1601: CALCULATES THE DISPLACEMENT CONSTRAINT DESIGN
CP*
CP***********************************************************
CP*
CP* DESCRIPTION:
CP*
CP*       'DP1601' CALCULATES THE DISPLACEMENT CONSTRAINT DESIGN
CP*       SENSITIVITY VECTOR FOR A TRIANGULAR PLATE BENDING
CP*       ELEMENT.
CP*
CP***********************************************************
CP*
CP*   NT      COUNTER FOR FINITE DIFFERENCE
CP*   I       EXTERNAL ELEMENT NO. BEING PROCESSED
CP*   L1      ORIGINAL EXTERNAL LOAD CASE NO.
CP*   L2      EXTERNAL ADJOINT LOAD CASE NO.
CP*   IL1     ORIGINAL INTERNAL LOAD CASE NO. - RETURNED VALUE
CP*
CP***********************************************************
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIPN.MON'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
      INCLUDE '[AEGSDR.INC] ELFDES.MON'
      INCLUDE '[AEGSDR.INC] SVECTR.MON'
      COMMON/LCSDES/DLCS(90)
C
      EQUIVALENCE (NDAT(98),IDBL)
C
      DIMENSION SIGMA(6),EPSLN(6),SBUF(100),BUF(100),CPE(500),
     *          X(3),Y(3),Z(3)
C
      DATA  KT/3/,IREF/1/
C
        JJ = L1
        CPE(I) = 0.0
C
C*** GET INTERNAL LOAD CASE NUMBER
C
10      CALL ACCLCS(1,IPNLCS,IDBL,1,0,IERR)
        IF(IERR.NE.0)  GO TO 805
        CALL ACCLCS(2,IPNLCS,JJ,2,DLCS,IERR)
        IF(IERR.NE.0)  GO TO 805
        ILCN = DLCS(21)
C
C*** GET PROPERTIES
C
        CALL ACCEPR(2,IPNEPR,IPTAB,0,BUF,LEN,IERR)
        IF(IERR.NE.0)  GO TO 809
        PB(NT) = BUF(25)
C
C*** SETUP POINTER FOR STRESS-STRAIN BUFFER
C
        CALL ACCFES(1,IPNFES,IDBL,1,ILCN,0,0,IERR)
        IF (IERR.NE.0)  GO TO 801
        IF(JJ.EQ.L1)  IL1 = ILCN
C
C*** GET ELEMENT STRESSES AND STRAINS
C
```

```
      CALL ACCFES(2,IPNFES,KINT,IRFF,ILCN,SBUF,LEN,IERR)
      IF(IERR.NE.0)  GOTO 801
      IF (JJ.EQ.L2)  GO TO 55
      DO 50 K=1,NDOF
        SIGMA(K) = SBUF(K)
50    CONTINUE
      GO TO 65
55    M = 7
      DO 60 K=1,NDOF
        EPSLN(K) = SBUF(M)
        M = M+1
60    CONTINUE
65    IF (JJ.EQ.L2)  GO TO 100
      JJ = L2
      GO TO 10
C
C*** GET THE JACOBIAN
C
100   CALL ACCELC(2,IPNELC,KINT,IREF,BUF,LENB,IERR)
      IF(IERR.NE.0)  GO TO 807
      M = 1
      DO 200 J=1,LENB,3
        K = J+1
        LL = J+2
        X(M) = BUF(J)
        Y(M) = BUF(K)
        Z(M) = BUF(LL)
        M = M+1
200   CONTINUE
C
      DETJ = EUTRIA(X,Y)
C
C*** CALCULATE SENSITIVITY VECTOR
C
      DO 340 J=1,3
        CPE(I) = CPE(I) + SIGMA(J)*EPSLN(J)*DETJ
340   CONTINUE
      DPSITB(I) = -CPE(I)
C
    GO TO 820
C
C*** WRITE ERROR MESSAGES TO THE SCREEN
C
801 PRINT 871, IERR
    GO TO 820
805 PRINT 875, IERR
    GO TO 820
807 PRINT 878, IERR
    GO TO 820
809 PRINT 879, IERR
    GO TO 820
C
820 CONTINUE
C
854 FORMAT(1X,I2,2X,3(E16.8,2X))
855 FORMAT(1X,'GP',5X,'SIGMAX(GP)',8X,'SIGMAY(GP)',8X,
   *'SIGMAXY(GP)')
860 FORMAT(1X,'GP',5X,'EPSLNX(GP)',8X,'EPSLNY(GP)',8X,
   *'EPSLNXY(GP)')
871 FORMAT(1X,'ACCFES RETURNED WITH ERROR ',I4)
875 FORMAT(1X,'ACCLCS RETURNED WITH ERROR ',I4)
```

```
878   FORMAT(1X,'ACCELC RETURNED WITH ERROR ',I4)
879   FORMAT(1X,'ACCEPR RETURNED WITH ERROR ',I4)
C
C
      RETURN
      END
```

```
      SUBROUTINE DP1602(NT,I,L1,L2,IL1)
CP*************************************************************
CP*
CP* DP1602: CALCULATES THE DISPLACEMENT CONSTRAINT DESIGN
CP*
CP*************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*      'DP1602' CALCULATES THE DISPLACEMENT CONSTRAINT DESIGN
CP*      SENSITIVITY VECTOR FOR A FOUR NODE PLATE BENDING
CP*      ELEMENT.
CP*
CP*************************************************************
CP*
CP*   NT      COUNTER FOR FINITE DIFFERENCE
CP*   I       EXTERNAL ELEMENT NO. BEING PROCESSED
CP*   L1      ORIGINAL EXTERNAL LOAD CASE NO.
CP*   L2      EXTERNAL ADJOINT LOAD CASE NO.
CP*   IL1     ORIGINAL INTERNAL LOAD CASE NO. - RETURNED VALUE
CP*
CP*************************************************************
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIPN.MON'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
      INCLUDE '[AEGSDR.INC] ELEDFS.MON'
      INCLUDE '[AEGSDR.INC] SVECTR.MON'
      COMMON/LCSDES/DLCS(90)
C
      EQUIVALENCE (NDAT(98),IDBL)
C
      DIMENSION SIGMA(6,4),EPSLN(6,4),SBUF(100),BUF(100),CPE(500),
     *          X(4),Y(4),Z(4)
C
      DATA  KT/3/,IREF/1/
C
        JJ = L1
        CPE(I) = 0.0
C
C*** GET PROPERTIES
C
        CALL ACCEPR(2,IPNEPR,IPTAB,0,BUF,LEN,IERR)
        IF(IERR.NE.0)  GO TO 809
        PB(NT) = BUF(25)
C
C*** GET INTERNAL LOAD CASE NUMBER
C
10      CALL ACCLCS(1,IPNLCS,IDBL,1,0,IERR)
        IF(IERR.NE.0)  GO TO 805
        CALL ACCLCS(2,IPNLCS,JJ,2,DLCS,IERR)
        IF(IERR.NE.0)  GO TO 805
        ILCN = DLCS(21)
C
C*** SETUP POINTER FOR STRESS-STRAIN BUFFER
C
        CALL ACCFES(1,IPNFES,IDBL,1,ILCN,0,0,IERR)
        IF (IERR.NE.0)  GO TO 801
        IF(JJ.EQ.L1)  IL1 = ILCN
C
C*** GET ELEMENT STRESSES AND STRAINS
C
```

```
      CALL ACCFES(2,IPNFES,KINT,IREF,ILCN,SBUF,LEN,IERR)
      IF(IERR.NE.0)  GOTO 801
      M = 1
      IF (JJ.EQ.L2)  GO TO 55
      DO 50 K=1,NSVAL
        J = M+1
        L = M+2
        SIGMA(1,K) = SBUF(M)
        SIGMA(2,K) = SBUF(J)
        SIGMA(3,K) = SBUF(L)
        M = M+6
50    CONTINUE
      GO TO 65
55    M = 25
      DO 60 K=1,NSVAL
        J = M+1
        L = M+2
        EPSLN(1,K) = SBUF(M)
        EPSLN(2,K) = SBUF(J)
        EPSLN(3,K) = SBUF(L)
        M = M+6
60    CONTINUE
65    IF (JJ.EQ.L2)  GO TO 100
      JJ = L2
      GO TO 10
C
C*** GET THE JACOBIAN
C
100   CALL ACCELC(2,IPNELC,KINT,IREF,BUF,LENB,IERR)
      IF(IERR.NE.0)  GO TO 807
      M = 1
      DO 200 J=1,LENB,3
        K = J+1
        LL = J+2
        X(M) = BUF(J)
        Y(M) = BUF(K)
        Z(M) = BUF(LL)
        M = M+1
200   CONTINUE
C
      DETJ = AREAQ(X,Y)
      DETJ = DETJ/4.D0
C
C*** CALCULATE SENSITIVITY VECTOR
C
      DO 340 J=1,3
        DO 340 K=1,NSVAL
          CPE(I) = CPE(I) + SIGMA(J,K)*EPSLN(J,K)*DETJ
340   CONTINUE
      DPSITB(I) = -CPE(I)
C
    GO TO 820
C
C*** WRITE ERROR MESSAGES TO THE SCREEN
C
801 PRINT 871, IERR
    GO TO 820
805 PRINT 875, IERR
    GO TO 820
807 PRINT 878, IERR
    GO TO 820
```

```
809   PRINT 879, IERR
      GO TO 820
C
820   CONTINUE
C
854   FORMAT(1X,I2,2X,3(E16.8,2X))
855   FORMAT(1X,'GP',5X,'SIGMAX(GP)',8X,'SIGMAY(GP)',8X,
     *'SIGMAXY(GP)')
860   FORMAT(1X,'GP',5X,'EPSLNX(GP)',8X,'EPSLNY(GP)',8X,
     *'EPSLNXY(GP)')
871   FORMAT(1X,'ACCFES RETURNED WITH ERROR ',I4)
875   FORMAT(1X,'ACCLCS RETURNED WITH ERROR ',I4)
878   FORMAT(1X,'ACCELC RETURNED WITH ERROR ',I4)
879   FORMAT(1X,'ACCEFR RETURNED WITH ERROR ',I4)
C
C
      RETURN
      END
```

```
      SUBROUTINE EU3DSB(PSI,SHPF,DDSHPF,KT,EL)
CP*****************************************************************
CP*
CP* EU3DSB: BEAM SHAPE FUNCTIONS
CP*
CP*****************************************************************
CP*
CP*   PSI        GAUSS POINT LOCATION. THE CENTER OF THE BEAM
CP*              IS ZERO.
CP*   SHPF       STANDARD BEAM SHAPE FUNCTIONS - RETURNED VALUE
CP*   DDSHPF     SECOND DERIVATIVE OF THE SHAPE FUNCTIONS
CP*               - RETURNED VALUE
CP*   KT         FLAG FOR RETURNING DDSHPF
CP*                    =1  ONLY RETURN SHPF
CP*                    =2  RETURN BOTH SHPF AND DDSHPF
CP*   EL         ELEMENT LENGTH
CP*
CP*****************************************************************
C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
C
      DIMENSION SHPF(12),DDSHPF(12)
C
      EL2 = EL*EL
      EL3 = EL2*EL
      X = PSI + (EL/2)
      X2 = X*X
      X3 = X2*X
C
C  CALCULATE THE SHAPE FUNCTIONS
C
      SHPF(1) = 1-X/EL
      SHPF(2) = 1-(3*X2/EL2)+(2*X3/EL3)
      SHPF(3) = SHPF(2)
      SHPF(4) = SHPF(1)
      SHPF(5) = X-(2*X2/EL)+(X3/EL2)
      SHPF(6) = SHPF(5)
      SHPF(7) = X/EL
      SHPF(8) = (3*X2/EL2)-(2*X3/EL3)
      SHPF(9) = SHPF(8)
      SHPF(10) = SHPF(7)
      SHPF(11) = (-X2/EL)+(X3/EL2)
      SHPF(12) = SHPF(11)
C
      IF(KT.EQ.1)  GO TO 900
C
C  CALCULATE THE SECOND DERIVATIVE OF THE SHAPE FUNCTIONS
C
      DDSHPF(1) = 0.0D0
      DDSHPF(2) = (-6/EL2)+(12*X/EL3)
      DDSHPF(3) = DDSHPF(2)
      DDSHPF(4) = 0.0D0
      DDSHPF(5) = (-4/EL)+(6*X/EL2)
      DDSHPF(6) = DDSHPF(5)
      DDSHPF(7) = 0.0D0
      DDSHPF(8) = (6/EL2)-(12*X/EL3)
      DDSHPF(9) = DDSHPF(8)
      DDSHPF(10) = 0.0D0
      DDSHPF(11) = (-2/EL)+(6*X/EL2)
      DDSHPF(12) = DDSHPF(11)
C
```

```
900   CONTINUE
C
C
      RETURN
      END
```

```
      SUBROUTINE GETSEN(PSIB,NT,NELM,IPNAME)
CP*****************************************************************
CP*
CP* GETSEN: BRANCHES TO THE APPROPRIATE CONSTRAINT TYPE.
CP*
CP*****************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*       'GETSEN' BRANCHES TO THE APPROPRIATE CONSTRAINT TYPE.
CP*       THE AVAILABLE CONSTRAINT TYPES ARE:
CP*                 COMP - COMPLIANCE
CP*                 DISP - DISPLACEMENT
CP*                 STRESS
CP*
CP*****************************************************************
CP*
CP*    PSIB           THE CONSTRAINT VALUE IS RETURNED FOR FINITE
CP*                   DIFFERENCE EVALUATION
CP*    NT             COUNTER FOR THE FINITE DIFFERENCE
CP*    NELM           TOTAL NUMBER OF ELEMENTS IN FINITE ELEMENT
CP*                   MODEL. THIS IS IN IFAD DATA BASE, AND IS
CP*                   RETRIEVED WHEN DATA BASE IS OPENED.
CP*    IPNAME         NAME OF FINITE ELEMENT ANALYSIS - CHARACTER
CP*                   INTERGER CONVERSION.
CP*
CP*****************************************************************
C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIFN.MON'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
      INCLUDE '[AEGSDR.INC] ELEDES.MON'
      INCLUDE '[AEGSDR.INC] SVECTR.MON'
C
      COMMON /MEMARY/ IARRY(60000)
      CHARACTER NAME*8,A*4,B*4
C
      DIMENSION IPNAME(2),IDBPTR(4),IERDEV(2),PSIB(2)
C
      DATA ISIZE/60000/
C
      NT = NT+1
C
60    CALL ACCFND(7,IPNAME,1,0,0,IERR)
      IF(IERR.NE.0)  GO TO 800
C
C  OPEN THE IFAD DATABASE
C
      CALL INENTR(ISIZE,IERDEV,IPNAME,IDBPTR,3,ISTAT)
      IF (ISTAT.NE.0)  GO TO 810
      A = CHAR(IPNAME(1))
      B = CHAR(IPNAME(2))
      NAME = A//B
C
C  BRANCH TO THE APPROPRIATE CONSTRAINT ROUTINE
C
100   IF(ICT.EQ.1)  CALL COMP(PSIB,NT,NELM)
      IF(ICT.EQ.2)  CALL DISP(PSIB,NT,NELM)
      IF(ICT.EQ.3)  CALL STRESS(PSIB,NT,NELM,NAME)
      GO TO 820
C
```

```
C  WRITE ERROR MESSAGES TO THE SCREEN
C
800    PRINT 875, IERR
       GO TO 820
810    PRINT 876, IERR
       GO TO 820
C
C
C  CLOSE IFAD DATABASE
C
820    CALL INEXIT(IDBPTR,IERR)
       IF(IERR.NE.0)  GO TO 850
C
       GO TO 900
C
850    PRINT 877,IERR
C
900    CONTINUE
C
C  FORMAT STATEMENTS
C
875    FORMAT(1X,'ACCFND RETURNED WITH ERROR ',I4)
876    FORMAT(1X,'INENTR RETURNED WITH ERROR ',I4)
877    FORMAT(1X,'INEXIT RETURNED WITH ERROR ',I4)
1000   FORMAT(2A4)
1001   FORMAT(F8.5)
1004   FORMAT(A6)
C
C
C
       RETURN
       END
```

```
      SUBROUTINE LC16(X,Y,Z,XL,YL,ZL,TB)
CP*****************************************************************
CP*
CP* LC16: TRANSFORMS GLOBAL COORD TO LOCAL COORD FOR '1601'
CP*
CP*****************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*       'LC16' TRANSFORMS THE TRIANGULAR GLOBAL COORDINATES
CP*       OF ELEMENT 1601 TO LOCAL COORDINATES
CP*
CP*****************************************************************
CP*
CP*   X      GLOBAL X COORDINATE
CP*   Y      GLOBAL Y COORDINATE
CP*   Z      GLOBAL Z COORDINATE
CP*   XL     LOCAL X COORDINATE
CP*   YL     LOCAL Y COORDINATE
CP*   ZL     LOCAL Z COORDINATE
CP*   TB     TRANSFORMATION MATRIX
CP*
CP*****************************************************************
C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIPN.MON'
      INCLUDE '[AEGSDR.INC] ELEDES.MON'
C
      DIMENSION X(3),Y(3),Z(3),TB(6,6),COORL(3,3),XL(3),YL(3),
     *          ZL(3),COOR(3,3),V12(3),V13(3),VN(3),T(3,3)
C
      CHARACTER STYPE*4
C
      DATA IREF/1/,ITYPE/1/
C
C*** INTIALIZE VARIABLES
C
C
      DO 40 I=1,3
        COOR(1,I) = X(I)
        COOR(2,I) = Y(I)
        COOR(3,1) = Z(I)
40    CONTINUE
C
C*** GET THE VECTORS PARALLEL TO THE 1-2 AND 1-3 SIDES
C
      CALL UMVEC(COOR(1,1),COOR(1,2),V12,IERR)
      CALL UMVEC(COOR(1,1),COOR(1,3),V13,IERR)
C
C  OBTAIN NORMAL V12 X V13
C
      CALL UMVCRS(V12,V13,VN,0,IERR)
C
C  OBTAIN LOCAL TRANSFORMATION MATRIX [T]
C
      CALL EUTSCS(VN,T,IERR)
C
C  OBTAIN THE NODAL TRANSFORMATION MATRIX AS AN ASSEMBLFDGE OF [T]
C
      DO 50 K=1,3
        DO 50 J=1,3
```

```
          TB(J,K) = T(J,K)
          TB(J+3,K+3) = T(J,K)
50      CONTINUE
C
C   GET LOCAL COORDINATES
C
        CALL UMXATB(T,COOR,COORL,3,3,3)
        DO 60 I=1,3
          XL(I) = COORL(1,I)
          YL(I) = COORL(2,I)
          ZL(I) = COORL(3,I)
60      CONTINUE
C
        RETURN
        END
```

```
      SUBROUTINE SRSTO5(ILCN)
CP***************************************************************
CP*
CP* SRSTO5: CALCULATES THE ADJOINT LOAD FOR THE BEAM ELEMENT
CP*
CP***************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*       'SRSTO5' CALCULATES THE ADJOINT LOAD FOR BENDING
CP*       STRESS IN A 1-DIMENSIONAL BEAM
CP*       ELEMENT AND CREATES A RESTART FILE SO THAT A RESTART
CP*       OF THE FINITE ELEMENT MODEL CAN BE MADE.  THE
CP*       RESULTING DISPLACEMENTS CAN THEN BE USED TO CALCULATE
CP*       THE STRESS CONSTRAINT DESIGN SENSITIVITY.
CP*****   THIS ROUTINE ONLY WORKS FOR BEAMS LYING IN THE
CP*       X-GLOBAL OR Y-GLOBAL PLANE.
CP*
CP***************************************************************
CP*
CP*   ILCN    INTERNAL LOAD CASE NO. OF ORIGINAL LOAD
CP*
CP***************************************************************
C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIFN.MON'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
      INCLUDE '[AEGSDR.INC] ELEDES.MON'
      INCLUDE '[AEGSDR.INC] SVECTR.MON'
C
      DIMENSION X(2),Y(2),Z(2),DDSHPF(12),DATN(50),BUF(100),
     *          SHPF(12),CFBUF(6),GPLW(3),ALGP(12),CD(2,6),WTW(3),
     *          IDIR(2),AL(12)
C
      DATA GPLW/-.774596669241483D0, .0D0,
     *            .774596669241483D0/
      DATA WTW/ .555555555555556D0, .888888888888889D0,
     *           .555555555555556D0/
      DATA  KT/3/,IREF/1/,MPT/1/
C
C  INITIALIZE VARIABLES
C
      ND = NDOF*NUNPE
      DO 10 J=1,ND
        AL(J) = 0.0
10    CONTINUE
C
C  GET DEPTH AND WIDTH OF THE BEAM
C
        CALL ACCEPR(2,IPNEPR,IPTAB,0,BUF,LEN,IERR)
        IF(IERR.NE.0)  GO TO 809
        H = 2*BUF(9)
        B = 2*BUF(10)
C
C    GET MODULUS OF ELASTICITY
C
        CALL ACCMAT(2,IPNMAT,NMAT,MPT,BUF,LEN,IERR)
        IF(IERR.NE.0)  GO TO 808
        E = BUF(5)
C
C  GET X, Y, AND Z OF ELEMENT NODES
```

```
C
         CALL ACCELC(2,IFNELC,KINT,IREF,BUF,LEN,IERR)
         IF(IERR.NE.0)  GO TO 807
         M = 1
         DO 200 J=1,6,3
         K = J+1
         L = J+2
         X(M) = BUF(J)
         Y(M) = BUF(K)
         Z(M) = BUF(L)
         M = M+1
200      CONTINUE
C
C  CALCULATE ELEMENT LENGTH
C
         DX = X(2)-X(1)
         DY = Y(2)-Y(1)
         DZ = Z(2)-Z(1)
         EL = DSQRT(DX*DX+DY*DY+DZ*DZ)
C
         XMP = 1.DO/EL
C
         DO 300 K=1,3
           PSI = GPLW(K)
           CALL EU3DSB(PSI,SHPF,DDSHPF,2,EL)
C
C  CALCULATE ADJOINT LOAD
C
         ALGP(3) = -.5DO*H*E*XMP*DDSHPF(3)
         ALGP(9) = -.5DO*H*E*XMP*DDSHPF(9)
         IF(DX.EQ.0.AND.DZ.EQ.0)  GO TO 250
         ALGP(4) = 0.0DO
         ALGP(5) = -.5DO*H*E*XMP*DDSHPF(5)
         ALGP(10) = 0.0DO
         ALGP(11) = -.5DO*H*E*XMP*DDSHPF(11)
         GO TO 255
250      ALGP(4) = -.5DO*H*E*XMP*DDSHPF(5)
         ALGP(5) = 0.0DO
         ALGP(10) = -.5DO*H*E*XMP*DDSHPF(11)
         ALGP(11) = 0.0DO
C
C  SUM ADJOINT LOAD OVER GAUSS POINTS
C
255      DO 260 J=1,ND
           AL(J) = AL(J) + ALGP(J)
260      CONTINUE
300    CONTINUE
C
       N = -1
       DO 400 J=1,NUNPE
         DO 350 K=1,NDOF
           IF(AL(K+J+N).EQ.0.0)  GO TO 350
           WRITE(11,864) IEXTNN(J),K,AL(K+J+N)
350      CONTINUE
         N = N+5
400    CONTINUE
C
C
       GO TO 820
C
C  WRITE ERROR MESSAGES TO THE SCREEN
```

```
C
802    PRINT 872, IERR
       GO TO 820
807    PRINT 878, IERR
       GO TO 820
808    PRINT 879, IERR
       GO TO 820
809    PRINT 876, IERR
       GO TO 820
C
C
820    CONTINUE
C
C
860    FORMAT(1X,'GP=',I2,4X,'W=',E11.5,4X,'WXX=',E11.5,4X,'AW=',
      *E11.5,4X,'AWXX=',E11.5)
864    FORMAT(2(I4,2X),E16.8)
872    FORMAT(1X,'ACCCND RETURNED WITH ERROR ',I4)
876    FORMAT(1X,'ACCEPR RETURNED WITH ERROR ',I4)
878    FORMAT(1X,'ACCELC RETURNED WITH ERROR ',I4)
879    FORMAT(1X,'ACCMAT RETURNED WITH ERROR ',I4)
1001   FORMAT(E12.5)
C
C
C
       RETURN
       END
```

```
      SUBROUTINE SRST11(ILCN,IE)
CP****************************************************************
CP*
CP* SRST11: CALCULATES THE ADJOINT LOAD FOR THE PLANE STRESS EL
CP*
CP****************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*       'SRST11' CALCULATES THE ADJOINT LOADS FOR A FOUR AND
CP*       EIGHT NODE PLANE STRESS ELEMENT AND THEN CREATES A
CP*       RESTART FILE SO THAT THE FINITE ELEMENT MODEL CAN BE
CP*       RESTARTED SO THAT THE RESULTING STRESSES AND STRAINS
CP*       CAN BE THEN USED TO CALCULATED THE STRESS CONSTRAINT
CP*       DESIGN SENSITIVITY.
CP*
CP****************************************************************
CP*
CP*  ILCN     INTERNAL LOAD CASE NO. OF ORIGINAL LOAD
CP*  IE       EXTERNAL ELEMENT NO. CONSTRAINED
CP*
CP****************************************************************
C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIPN.MON'
      INCLUDE '[AEGSDR.INC] CNIL.MON'
      INCLUDE '[AEGSDR.INC] ELEDES.MON'
      INCLUDE '[AEGSDR.INC] SVECTR.MON'
C
      DIMENSION SHPF(8),GPL(2,4),SBUF(50),BUF(100),
     *          DSHPGX(8),DSHPGY(8),DSHPL(2,8),X(8),Y(8),Z(8),
     *          DG(3),ALGP(16),AL(16),E(3,3),D(9),C(3),B(3,16),
     *          EMT(1)
C
      DATA  GPL/2*-.57735027, .57735027,-.57735027,
     *          2* .57735027,-.57735027, .57735027/
      DATA  KT/3/,IREF/1/,MPT/1/,ITYPE/1/
C
C** INITIALIZE VARIABLES
C
        ND = NUNPE*NDOF
        DO 10 J=1,ND
10      AL(J) = 0.
        IF(IE.GT.1) GO TO 15
C
C** GET X AND Y FOR JACOBIAN EVALUATION
C
15      CALL ACCELC(2,IPNELC,KINT,IREF,BUF,LENB,IERR)
        IF(IERR.NE.0)  GO TO 807
        M = 1
        DO 20 L=1,LENB,3
          J = L+1
          K = L+2
          X(M) = BUF(L)
          Y(M) = BUF(J)
          Z(M) = BUF(K)
          M = M+1
20      CONTINUE
        AREA = AREAQ(X,Y)
        XMP = 1.DO/AREA
C
```

```
C** CALCULATE ADJOINT LOAD
C
C*** LOOP OVER GAUSS POINTS
C
        IC = 0
        M = 1
        DO 80 KK=1,NSVAL
          J = M+1
          L = M+3
C
C*** GET DERIVATIVES OF STRESS FUNCTION G
C
C*** VON MISES: G=(SIGX**2+SIGY**2-SIGX*SIGY+3*SIGXY**2)**.5
C
          IF(IC.GT.0)  GO TO 25
          CALL ACCFES(2,IFNFES,KINT,IREF,ILCN,SBUF,LEN,IERR)
          IF(IERR.NE.0)  GO TO 801
25        IF(IST.EQ.1)  GO TO 30
          VMS = (SBUF(M)**2+SBUF(J)**2-SBUF(M)*SBUF(J)+3*
     *            SBUF(L)**2)**.5
          DG(1) = .5*(2*SBUF(M)-SBUF(J))/VMS
          DG(2) = .5*(2*SBUF(J)-SBUF(M))/VMS
          DG(3) = (3*SBUF(L))/VMS
          GO TO 40
C
C*** PRINCIPAL STRESS
C
30        TMAX = ((.5*(SBUF(M)-SBUF(J)))**2+SBUF(L)**2)**.5
          DG(1) = .5+.25*(SBUF(M)-SBUF(J))*(1/TMAX)
          DG(2) = .5-.25*(SBUF(M)-SBUF(J))*(1/TMAX)
          DG(3) = SBUF(L)*(1/TMAX)
C
40        M = M+4
          IC = IC+1
          PSI = GPL(1,KK)
          ETA = GPL(2,KK)
          IF(ISTYP.EQ.2)  CALL EU2DLQ(PSI,ETA,KT,SHPF,DSHPL,
     *                    DSHPGX,DSHPGY,DETJ,X,Y,IERR)
          IF(ISTYP.EQ.4)  CALL EU2DPQ(PSI,ETA,KT,SHPF,DSHPL,
     *                    DSHPGX,DSHPGY,DETJ,X,Y,IERR)
          IF(IERR.NE.0)  GO TO 809
C
C*** GET ELASTICITY MATRIX E
C
          CALL EU2DSS(D,TMP,EMT,NMAT,ITYMAT,ITYPE)
          N = 1
          DO 50 JJ=1,3
            LL = N+1
            LLL = N+2
            E(JJ,1) = D(N)
            E(JJ,2) = D(LL)
            E(JJ,3) = D(LLL)
            N = N+3
50        CONTINUE
C
C*** GET STRAIN-DISPLACEMENT MATRIX B
C
          CALL SD11(B,DSHPGX,DSHPGY,NUNPE)
C
C*** CALCULATE [DG]*[E]*[B]*MP
C
```

```
          CALL UMXAB(DG,E,C,1,3,3)
          CALL UMXAB(C,B,ALGP,1,ND,3)
         DO 60 JJ=1,ND
60        ALGP(JJ) = ALGP(JJ)*XMP
C
C*** SUM ADJOINT LOAD OVER GAUSS POINTS (INTEGRATE OVER ELEM)
C
          DO 70 JJ=1,ND
70        AL(JJ) = AL(JJ)+ALGP(JJ)*DETJ
80      CONTINUE
C
C** WRITE ADJOINT LOAD TO RESTART FILE
C
        N = -1
        DO 95 J=1,NUNPE
          DO 90 K=1,NDOF
            WRITE(11,2004) IEXTNN(J),K,AL(K+J+N)
90        CONTINUE
          N = N+1
95      CONTINUE
C
      GO TO 820
C  WRITE ERROR MESSAGES TO THE SCREEN
C
801   PRINT 871, IERR
      GO TO 820
807   PRINT 878, IERR
      GO TO 820
809   PRINT 876, IERR
      GO TO 820
C
C
820   CONTINUE
C
C
871   FORMAT(1X,'ACCFES RETURNED WITH ERROR ',I4)
876   FORMAT(1X,'SHAPE FUNCTION ROUTINE RETURNED WITH ERROR ',
     *I4)
878   FORMAT(1X,'ACCELC RETURNED WITH ERROR ',I4)
2001  FORMAT(A)
2004  FORMAT(1X,I4,1X,I2,1X,E16.9)
C
C
      RETURN
      END
```

```
      SUBROUTINE SRST16(ILCN,IE)
CP************************************************************
CP*
CP* SRST16: CALCULATES THE ADJOINT LOADS FOR A TRI. BENDING EL
CP*
CP************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*       'SRST16' CALCULATES THE ADJOINT LOADS FOR A TRIANGULAR
CP*       PLATE BENDING ELEMENT AND CREATES A RESTART FILE SO
CP*       THAT THE FINITE ELEMENT MODEL CAN BE RESTARTED. THE
CP*       RESULTING STRESSES AND STRAINS ARE THEN USED IN THE
CP*       STRESS CONSTRAINT DESIGN SENSITIVITY CALCULATION IN
CP*       THE 'ST1601' SUBROUTINE.  STRESS TYPES 1=PRINCIPAL
CP*                                              2 = VON MISES
CP*
CP************************************************************
CP*
CP*   ILCN     INTERNAL LOAD CASE NO. OF THE ORIGINAL LOAD
CP*   IE       EXTERNAL CONSTRAINED ELEMENT NO.
CP*
CP************************************************************
C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIPN.MON'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
      INCLUDE '[AEGSDR.INC] ELEDFS.MON'
      INCLUDE '[AEGSDR.INC] SVECTR.MON'
C
      DIMENSION X(3),Y(3),Z(3),BUF(50),AL(18),SBUF(50),DG(3),
     *          CG(3),XL(3),YL(3),ZL(3),C(3),D(9),EMI(27),E(3,3),
     *          TB(6,6),SIG(3,3),ALM(18)
C
C
      DATA  IREF/1/,ITYPE/1/
C
        DO 10 J=1,18
          AL(J) = 0.0D0
10      CONTINUE
C
C** GET X, Y AND Z
C
15      CALL ACCELC(2,IPNELC,KINI,IREF,BUF,LENB,IERR)
        IF(IERR.NE.0)  GO TO 807
        M = 1
        DO 20 L=1,LENB,3
          J = L+1
          K = L+2
          X(M) = BUF(L)
          Y(M) = BUF(J)
          Z(M) = BUF(K)
          M = M+1
20      CONTINUE
C
C** GET LOCAL COORDINATE SYSTEM
C
        CALL LC16(X,Y,Z,XL,YL,ZL,TB)
C
C* GET ELASTICITY MATRIX [E]
C
```

```
         CALL EU2DSS(D,IMP,EMT,NMAT,ITYMAT,ITYPE)
         N=1
         DO 25 JJ=1,3
           E(JJ,1) = D(N)
           E(JJ,2) = D(N+1)
           E(JJ,3) = D(N+2)
           N = N+3
25       CONTINUE
C
C  GET THE THICKNESS OF THE PLATE
C
         CALL ACCEPR(2,IPNEPR,IPTAB,0,BUF,LEN,IERR)
         IF(IERR.NE.0)  GO TO 809
         THK = BUF(25)
C
C** GET STRESSES AT THE MIDSIDE NODES OF TRIANGLE
C
         CALL SSMD16(XL,YL,ILCN,SIG)
C
C** CALCULATE THE ADJOINT LOAD WHERE Q=[DG]*[E]*[B]*MP*T/2
C** AL(18) = (TAREA/3)*(Q(0,.5,.5)+Q(.5,0,.5)+Q(.5,.5,0)
C
         TAREA = EUTRIA(X,Y)
         DO 80 IT=1,3
C
C*** CALCULATE [DG] THE DERIVATIVE VECTOR
C
         IF(IST.EQ.1)  GO TO 30
C
C*** VON MISES: G=SQRT(SIGX**2+SIGY**2-SIGX*SIGY+3*SIGXY**2)
C
         VMS = DSQRT(SIG(IT,1)**2+SIG(IT,2)**2-SIG(IT,1)*SIG(IT,2)+3*
     *            SIG(IT,3)**2)
         DG(1) = .5D0*(2.D0*SIG(IT,1)-SIG(IT,2))/VMS
         DG(2) = .5D0*(2.D0*SIG(IT,2)-SIG(IT,1))/VMS
         DG(3) = (3.D0*SIG(IT,3))/VMS
         GO TO 40
C
C** PRINCIPAL STRESS
C
30       TMAX = DSQRT((.5D0*(SIG(IT,1)-SIG(IT,2)))**2+SIG(IT,3)**2)
         DG(1) = .5D0+.25D0*(SIG(IT,1)-SIG(IT,2))/TMAX
         DG(2) = .5D0-.25D0*(SIG(IT,1)-SIG(IT,2))/TMAX
         DG(3) = SIG(IT,3)/TMAX
C
C*** CALCULATE [C] = [DG] * [E]
C
40       CALL UMXAB(DG,E,C,1,3,3)
C
C*** CALCULATE [AL] = [C] * [B] * T/2 * XMP
C
           CALL AL16(XL,YL,C,ALM,IB,THK,IT)
           DO 50 K=1,18
             AL(K) = AL(K) + (TAREA/3.0D0)*ALM(K)
50         CONTINUE
80       CONTINUE
C
C** WRITE ADJOINT LOAD TO RESTART FILE
C
        N = -1
        DO 95 J=1,NUMPE
```

```
          DO 90 K=1,NDOF
            IF(AL(K+J+N).EQ.0.0D0)  GO TO 90
            WRITE(11,2004) IEXTNN(J),K,AL(K+J+N)
90         CONTINUE
           N = N+5
95       CONTINUE
C
      GO TO 820
C  WRITE ERROR MESSAGES TO THE SCREEN
C
801   PRINT 871, IERR
      GO TO 820
807   PRINT 878, IERR
      GO TO 820
809   PRINT 879, IERR
      GO TO 820
C
C
820   CONTINUE
C
C
871   FORMAT(1X,'ACCFES RETURNED WITH ERROR ',I4)
878   FORMAT(1X,'ACCELC RETURNED WITH ERROR ',I4)
879   FORMAT(1X,'ACCEPK RETURNED WITH ERROR ',I4)
2001  FORMAT(A)
2002  FORMAT(1X,I4,2X,'DISP= ',E16.9)
2004  FORMAT(1X,I4,1X,I2,1X,E16.9)
C
C
      RETURN
      END
```

```
      SUBROUTINE SSMD16(X,Y,ILCN,SIG)
CP*************************************************************
CP*
CP* SSMD16: CALCULATES STRESSES AT THE MIDSIDE NODES OF A TRI.
CP*
CP*************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*      'SSMD16' CALCULATES STRESSES AT THE MIDSIDE NODES OF
CP*      TRIANGLE 1601. [SIG] = [E]*[B]*[DIS]*T/2 IS THE
CP*      FINITE ELEMENT FORMULA FOR CALCULATING THE STRESS.
CP*      THE STRAIN-DISPLACEMENT MATRIX [B] IS LOCATED IN
CP*      COLUMNS 4,5, AND 6 OF THE [W] MATRIX.[E] IS THE
CP*      ELASTICITY MATRIX, 3X3 FOR ISOTROPIC MATERIAL, AND
CP*      [DIS] IS THE DISPLACEMENT VECTOR, WHERE THE DISPLS.
CP*      ARE TAKEN AT THE NODES OF THE TRIANGLE.  BECAUSE
CP*      IFAD CALCULATES THE STRESS RESULTANTS, THE T/2 TERM
CP*      MUST BE INCLUDED IN THIS CALCULATION.
CP*
CP*************************************************************
CP*
CP*   X       THE LOCAL ELEMENT X COORDINATE
CP*   Y       THE LOCAL ELEMENT Y COORDINATE
CP*   ILCN    THE INTERNAL LOAD CASE NUMBER FOR THE ORIGINAL
CP*           ORIGINAL MODEL
CP*   SIG     THE MIDSIDE NODE STRESSES - RETURNED VALUE
CP*
CP*************************************************************
C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIPN.MON'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
      INCLUDE '[AEGSDR.INC] ELEDES.MON'
C
      EQUIVALENCE (NDAT(14),IPR)
C
      DIMENSION X(3),Y(3),GPTS(3,3),W(18,7),CFBUF(7),CD(3,7),
     *          DIS(9),D(9),E(3,3),EMT(27),BUF(100),EB(3,9),
     *          SIG(3,3),XL(6),YL(6)
C
      DATA GPTS/0.0D0,.5D0,.5D0,  .5D0,0.0D0,.5D0,  .5D0,.5D0,0.0D0/
      DATA ITYPE/1/
C
C** GET DISPLACEMENTS
C
      DO 10 J=1,NUNPE
        CALL ACCCND(2,IPNCND,INTNN(J),1,ILCN,CFBUF,LEN,IERR)
        IF(IERR.NE.0)  GO TO 802
        DO 10 K=1,LEN
          CD(J,K) = CFBUF(K)
10    CONTINUE
      M = 1
      DO 20 J=1,NUNPE
        DIS(M) = CD(J,3)
        DIS(M+1) = CD(J,4)
        DIS(M+2) = CD(J,5)
        M = M+3
20    CONTINUE
C
C**  GET ELASTICITY MATRIX
```

```
C
      CALL EU2DSS(D,TMP,EMT,NMAT,1,YMAT,ITYPE)
      N = 1
      DO 30 JJ=1,3
        E(JJ,1) = D(N)
        E(JJ,2) = D(N+1)
        E(JJ,3) = D(N+2)
        N = N+3
30    CONTINUE
C
C** GET MATERIAL THICKNESS
C
      CALL ACCEPR(2,IPNEPR,IPTAB,0,BUF,LEN,IERR)
      IF(IERR.NE.0)  GO TO 809
      THK = BUF(25)
C
C** GET LOCAL X AND Y COORDINATES
C
      CALL MOVESP(XL,X,3*IPR)
      CALL MOVESP(YL,Y,3*IPR)
C
C** CALCULATE STRESSES AT THE MIDSIDE NODES
C
      DO 100 INT=1,3
C
C*** GET SHAPE FUNCTION AT MIDSIDE NODE
C
        CALL SF1501(XL,YL,GPTS(1,INT),W,EB)
C
C*** [SIG] = [E]*[B]*[DIS]*T/2
C
        DO 50 M=1,9
          DO 50 K=1,3
            GASH = 0.0D0
            DO 40 J=1,3
              GASH = GASH + E(K,J)*W(M,J+3)
40          CONTINUE
            EB(K,M) = GASH
50      CONTINUE
        DO 60 I1 = 1,3
          SIG(INT,I1) = 0.0D0
          DO 60 M2 = 1,9
            SIG(INT,I1) = SIG(INT,I1)+EB(I1,M2)*DIS(M2)
60      CONTINUE
        DO 70 I2=1,3
          SIG(INT,I2) = SIG(INT,I2)*THK/2.0D0
70      CONTINUE
100   CONTINUE
      GO TO 820
C
C
C** WRITE ERROR MESSAGES TO THE SCREEN
C
802   PRINT 872, IERR
      GO TO 820
809   PRINT 879, IERR
      GO TO 820
C
C
820   CONTINUE
C
```

```
C
872   FORMAT(1X,'ACCCND RETURNED WITH ERROR ',I4)
879   FORMAT(1X,'ACCEPR RETURNED WITH ERROR ',I4)
C
C
      RETURN
      END
```

```
      SUBROUTINE SNMD16(X,Y,ILCN,EPN)
CP************************************************************
CP*
CP* SNMD16: CALCULATES STRAINS AT THE MIDSIDE NODES OF A TRI.
CP*
CP************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*      'SNMD16' CALCULATES STRAINSES AT THE MIDSIDE NODES OF
CP*      TRIANGLE 1601.  THE FORMULAS FOR STRAIN ARE:
CP*
CP*                     EX = 1/E*(SIGX-V*SIGY)
CP*                     EY = 1/E*(SIGY-V*SIGX)
CP*                     EXY = SIGXY/G
CP*
CP************************************************************
CP*
CP*   X       THE LOCAL ELEMENT X COORDINATE
CP*   Y       THE LOCAL ELEMENT Y COORDINATE
CP*   ILCN    THE INTERNAL LOAD CASE NUMBER FOR THE ADJOINT
CP*           LOAD
CP*   EPN     THE MIDSIDE NODE STRAINS - RETURNED VALUE
CP*
CP************************************************************
C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIPN.MON'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
      INCLUDE '[AEGSDR.INC] ELEDES.MON'
C
      DIMENSION X(3),Y(3),GPTS(3,3),W(18,7),CFBUF(7),LD(3,7),
     *           DIS(9),EMT(27),BUF(100),EPN(3,3),SIG(3,3)
C
      DATA MPT/1/
C
C** GET CONSTANTS
C
      CALL ACCMAT(2,IPNMAT,NMAT,MPT,BUF,LEN,IERR)
      IF(IERR.NE.0)  GO TO 808
      E = BUF(5)
      V = BUF(7)
      G = E/(2.D0*(1.D0+V))
C
C** GET STRESSES AT THE MIDSIDE NODES
C
      CALL SSMD16(X,Y,ILCN,SIG)
C
C** CALCULATE STRAINS AT THE MIDSIDE NODES
C
      DO 100 IT=1,3
        EPN(IT,1) = 1/E*(SIG(IT,1)-V*SIG(IT,2))
        EPN(IT,2) = 1/E*(SIG(IT,2)-V*SIG(IT,1))
        EPN(IT,3) = SIG(IT,3)/G
100   CONTINUE

      GO TO 820
C
C
C** WRITE ERROR MESSAGES TO THE SCREEN
C
```

```
808    PRINT 878, IERRR
       GO TO 820
C
C
820    CONTINUE
C
C
878    FORMAT(1X,'ACCMAT RETURNED WITH ERROR ',I4)
C
C
       RETURN
       END
```

```
      SUBROUTINE STRESO5(NT,I,L,FSIBB)
CP*************************************************************
CP*
CP* STRESO5: CALC. STRESS CONSTRAINT AND SENSIT. FOR A BEAM
CP*
CP*************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*       'STRESO5' CALCULATES THE STRESS CONSTRAINT AND THE
CP*       DESIGN SENSITIVITY FOR THE 1-D BEAM ELEMENT IN
CP*       BENDING.  INCLUDES TORSION.
CP*
CP*************************************************************
CP*
CP*   NT      COUNTER FOR FINITE DIFFERENCE
CP*   I       EXTERNAL ELEMENT NO. FOR ELEMENT BEING PROCESSED
CP*   L       LOAD CASES - EXTERNAL
CP*   FSIBB   STRESS CONSTRAINT - RETURNED VALUE
CP*
CP*************************************************************
C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIPN.MON'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
      INCLUDE '[AEGSDR.INC] ELEDES.MON'
      INCLUDE '[AEGSDR.INC] SVECTR.MON'
      COMMON/LCSDES/DLCS(90)
C
      EQUIVALENCE (NDAT(14),IP),(NDAT(98),IDBL)
C
      DIMENSION X(3),Y(3),Z(3),DDSHPF(12),DATN(50),L(2),BUF(100),
     *          SHPF(12),CFBUF(6),GPLW(3),ALD(2,6),CD(2,6),WTW(3),
     *          FSIBB(500),C(6,2),D(6,2),T(3,3),TB(6,6),CDL(2,6),
     *          ALDL(2,6),COOR(3,3)
C
      DATA  GPLW/-.774596669241483D0, .0D0,
     *             .774596669241483D0/
      DATA WTW/ .555555555555556D0, .888888888888889D0,
     *           .555555555555556D0/
      DATA  KT/3/,IREF/1/,MPT/1/
C
      DPSIBG = 0.0D0
      DPSIHG = 0.0D0
      FSIBB(I) = 0.0D0
      IF(I.GT.1)  GO TO 50
C
C  GET AREA MOMENT OF INERTIA ABOUT Y-AXIS
C
50      CALL ACCEPR(2,IPNEPR,IPTAB,0,BUF,LEN,IERR)
        IF(IERR.NE.0)  GO TO 809
        YI = BUF(5)
        H = 2*BUF(9)
        B = 2*BUF(10)
        BH(NT) = H
        BW(NT) = B
C
C    GET WEIGHT DENSITY AND MODULUS OF ELASTICITY
C
        CALL ACCMAT(2,IPNMAT,NMAT,MPT,BUF,LEN,IERR)
        IF(IERR.NE.0)  GO TO 808
```

```
         GAMMA = 0.0D0
         E = 30500000.D0
         V = .3D0
         G = E/(2.D0*(1.D0+V))
C
C   GET INTERNAL LOAD CASE NUMBER FOR ORIGINAL LOAD
C
60       CALL ACCLCS(1,IPNLCS,IDBL,1,0,IERR)
         IF(IERR.NE.0)  GO TO 805
         CALL ACCLCS(2,IPNLCS,L(1),2,DLCS,IERR)
         IF(IERR.NE.0)  GO TO 805
         IL1 = DLCS(21)
C
C     GET DISPLACEMENTS AT ELEMENT ENDS
C
        DO 70 J=1,NUNPE
           CALL ACCCND(2,IPNCND,INTNN(J),1,IL1,CFBUF,LEN,IERR)
           IF(IERR.NE.0)  GO TO 802
           DO 70 K=1,NDOF
             CD(J,K) = CFBUF(K)
70       CONTINUE
C
C   BYPASS ADJOINT LOAD CALCULATION
C
         IF(NT.GT.1)  GO TO 150
C
C   GET INTERNAL LOAD CASE NUMBER FOR ADJOINT LOAD
C
         CALL ACCLCS(1,IPNLCS,IDBL,1,0,IERR)
         IF(IERR.NE.0)  GO TO 805
         CALL ACCLCS(2,IPNLCS,L(2),2,DLCS,IERR)
         IF(IERR.NE.0)  GO TO 805
         ILCN = DLCS(21)
C
C   GET DISPLACEMENTS AT ELEMENT ENDS FOR ADJOINT LOAD
C
         CALL ACCCND(1,IPNCND,IDBL,1,ILCN,0,0,IERR)
         IF(IERR.NE.0)  GO TO 802
         DO 100 J=1,NUNPE
           CALL ACCCND(2,IPNCND,INTNN(J),1,ILCN,CFBUF,LEN,IERR)
           IF(IERR.NE.0)  GO TO 802
           DO  100 K=1,NDOF
             ALD(J,K) = CFBUF(K)
100      CONTINUE
C
C***********************************************************
C   EVALUATE DISPLS. AND CURVATURE AT THE GAUSS POINT USING
C   SHAPE FUNCTIONS - ONE PT. FOR CURV., THREE PT. FOR DISPL
C***********************************************************

C   GET X, Y, AND Z OF ELEMENT NODES
C
150      CALL ACCELC(2,IPNELC,KINT,IREF,BUF,LEN,IERR)
         IF(IERR.NE.0)  GO TO 807
         M = 1
         DO 200 J=1,9,3
         K = J+1
         LL = J+2
         X(M) = BUF(J)
         Y(M) = BUF(K)
         Z(M) = BUF(LL)
```

```
        M = M+1
200     CONTINUE
        DO 210 J=1,3
          COOR(1,J) = X(J)
          COOR(2,J) = Y(J)
          COOR(3,J) = Z(J)
210     CONTINUE
C
C  FORM THE ELEMENT LOCAL COORDINATE SYSTEM FOR DISPLACEMENTS
C
        IN3 = INTNN(3)
        CALL EUBTM(IN3,BETA,COOR,T,IERR)
        CALL ZEROSP(TB,36*IP)
        DO 220 J=1,3
          DO 220 K=1,3
            TB(J,K) = T(J,K)
            TB(J+3,K+3) = TB(J,K)
220     CONTINUE
        CALL UMXABT(TB,CD,C,6,2,6)
        CALL UMXABT(TB,ALD,D,6,2,6)
        DO 230 J=1,2
          DO 230 K=1,6
            CDL(J,K) = C(K,J)
            ALDL(J,K) = D(K,J)
230       CONTINUE
C
C  CALCULATE ELEMENT LENGTH
C
        DX = X(2)-X(1)
        DY = Y(2)-Y(1)
        DZ = Z(2)-Z(1)
        EL = DSQRT(DX*DX+DY*DY+DZ*DZ)
C
C  CHANGE LOCAL Y-ROTATION FROM POSITIVE TO NEGATIVE IF
C  BEAM LIES ALONG THE X GLOBAL AXIS
C
        IF(DX.LT.0.001.AND.DX.GT.-0.001)  GO TO 245
        DO 240 J=1,NUNPE
          CDL(J,4) = -CDL(J,4)
          CDL(J,5) = -CDL(J,5)
          ALDL(J,3) = -ALDL(J,3)
240     CONTINUE
C
C  CALCULATE THE TWISTING ANGLES
C
245     WXY = DABS((CDL(2,4)-CDL(1,4))/EL)
        AWXY = DABS((ALDL(2,4)-ALDL(1,4))/EL)
C
C  EVALUATE SHAPE FUNCTIONS FOR DISPL. - THREE POINT QUADRATURE
C
        B2 = B*B
        B3 = B2*B
        B4 = B2*B2
        H2 = H*H
        H3 = H2*H
C
        DO 300 K=1,3
          PSI = GPLW(K)
          CALL EU3DSB(PSI,SHPF,DDSHPF,2,EL)
          W = (SHPF(3)*CDL(1,3)+SHPF(5)*CDL(1,5)+SHPF(9)*
     *        CDL(2,3)+SHPF(11)*CDL(2,5))
```

```
            AW = (SHPF(3)*ALDL(1,3)+SHPF(5)*ALDL(1,5)+SHPF(9)*
     *          ALDL(2,3)+SHPF(11)*ALDL(2,5))
C
C  EVALUATE SHAPE FUNCTIONS FOR CURV. - THREE POINT QUADRATURE
C
            WXX = (DDSHPF(3)*CDL(1,3)+DDSHPF(5)*CDL(1,5)+
     *            DDSHPF(9)*CDL(2,3)+DDSHPF(11)*CDL(2,5))
            AWXX = (DDSHPF(3)*ALDL(1,3)+DDSHPF(5)*ALDL(1,5)+
     *             DDSHPF(9)*ALDL(2,3)+DDSHPF(11)*ALDL(2,5))
C
C  CALCULATE SENSITIVITY VECTORS
C
            XMP = 1.D0/EL
            IF(NT.GT.1)  GO TO 250
            STERM = .5D0*XMP*E*WXX
            PJB = H3/3.D0-.42D0*B*(H2+B4/(4.D0*H2))
            PJH = B*H2-.42D0*B2*(H-B4/(12.D0*H3))
            DPSIBG = DPSIBG+(-GAMMA*H*AW-(E*H3/12.D0)*AWXX*WXX-
     *               PJB*G*WXY*AWXY)*WTW(K)*(EL/2.D0)
            IF(I.NE.ICE(NC))  GO TO 247
            DPSIHG = DPSIHG+(-GAMMA*B*AW-(3.D0*E*B*H2/12.D0)*AWXX*WXX
     *             -STERM-PJH*G*WXY*AWXY)*WTW(K)*(EL/2.D0)
            GO TO 250
247         DPSIHG = DPSIHG+(-GAMMA*B*AW-(3.D0*E*B*H2/12.D0)*AWXX
     *               *WXX-PJH*G*WXY*AWXY)*WTW(K)*(EL/2.D0)
C
250       IF(I.NE.ICE(NC))  GO TO 300
C
C  CALCULATE PSI(B) - INTEGRAL OF STRESS FUNCTION FOR ELEMENT
C
          STRESS = -.5D0*H*E*WXX*XMP
          PSIBB(I) = PSIBB(I) + STRESS*WTW(K)*(EL/2.D0)
300     CONTINUE
        IF(NT.GT.1)  GO TO 820
        DPSIB(I) = DPSIBG
        DPSIH(I) = DPSIHG
C
        GO TO 820
C
C  WRITE ERROR MESSAGES TO THE SCREEN
C
802     PRINT 872, IERR
        GO TO 820
805     PRINT 875, IERR
        GO TO 820
807     PRINT 878, IERR
        GO TO 820
808     PRINT 879, IERR
        GO TO 820
809     PRINT 876, IERR
        GO TO 820
C
C
820     CONTINUE
C
C
850     FORMAT(1X,'***ADJOINT LOAD IS APPLIED AT ELEMENT',I4)
851     FORMAT(/,1X,'BEAM WIDTH B=',F8.5,2X,'BEAM DEPTH=',F8.5,2X
     *,'E=',E9.3,2X,'IYY=',E9.3,2X,'GAMMA=',F6.5,'APPLIED FORCE
     *=',F8.5)
855     FORMAT(1X,'NODE=',I2,2X,'X=',E12.5,2X,'Y=',E12.5,2X,'Z=',
```

```
     *E12.5,2X,'RX=',E12.5,2X,'RY=',E12.5,2X,'RZ=',E12.5)
860   FORMAT(1X,'GP=',I2,4X,'W=',E11.5,4X,'WXX=',E11.5,4X,'AW=',
     *E11.5,4X,'AWXX=',E11.5)
864   FORMAT(1X,'NODE=',I2,2X,'AX=',E12.5,2X,'AY=',E12.5,2X,
     *'AZ=',E12.5,2X,'ARX=',E12.5,2X,'ARY=',E12.5,2X,'ARZ=',
     *E12.5)
872   FORMAT(1X,'ACCCND RETURNED WITH ERROR ',I4)
875   FORMAT(1X,'ACCLCS RETURNED WITH ERROR ',I4)
876   FORMAT(1X,'ACCEPR RETURNED WITH ERROR ',I4)
878   FORMAT(1X,'ACCELC RETURNED WITH ERROR ',I4)
879   FORMAT(1X,'ACCMAT RETURNED WITH ERROR ',I4)
1001  FORMAT(E12.5)
1004  FORMAT(I4)
C
C
      RETURN
      END
```

```
      SUBROUTINE STRES11(NT,I,L,PSIBT)
CP*****************************************************************
CP*
CP* STRES11: CALC. STRESS CONSTRAINT AND SENSIT. - PLANE STRESS
CP*
CP*****************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*      'STRES11' CALCULATES THE STRESS CONSTRAINT AND THE
CP*      DESIGN SENSITIVITY FOR A FOUR OR AND EIGHT NODE PLANE
CP*      STRESS ELEMENT WITH TRACTION, SELF WEIGHT NOT INCLUDED
CP*
CP*****************************************************************
CP*
CP*   NT      COUNTER FOR FINITE DIFFERENCE
CP*   I       EXTERNAL ELEMENT NO. BEING PROCESSED
CP*   L       EXTERNAL LOAD CASE NOS.
CP*   PSIBT . STRESS CONSTRAINT
CP*
CP*****************************************************************
C
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIPN.MON'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
      INCLUDE '[AEGSDR.INC] ELEDES.MON'
      INCLUDE '[AEGSDR.INC] SVECTR.MON'
      COMMON/LCSDES/DLCS(90)
C
      EQUIVALENCE (NDAT(98),IDBL)
C
      DIMENSION X(8),Y(8),Z(8),SHPF(8),GPL(2,4),L(2),ILCN(2),
     *          DATN(50),BUF(100),PSIBT(500),SBUF(50),CFBUF(6),
     *          BF(4,48),EMI(1),DSHPGX(8),EQBUF(50),DSHPGY(8),
     *          DSHPL(2,8),SIGMA(6,4),EPSLN(6,4),B(3,16),SE(500),
     *          DG(3),ALGP(16),AL(16),E(3,3),D(9),C(3),PSIBPE(500)
      CHARACTER YESNO*1
C
      DATA  GPL/2*-.57735027, .57735027,-.57735027,
     *          2*.57735027,-.57735027, .57735027/
      DATA  KT/3/,IREF/1/,MP1/3/,ITYPE/1/
C
C
      SE(I) = 0.0
      PSIBPE(I) = 0.0
      IF(I.NE.1)  GO TO 100
10      DO 50 JJ=1,2
C
C     GET INTERNAL LOAD CASE NUMBER
C
120     CALL ACCLCS(1,IPNLCS,IDBL,1,0,IERR)
        IF(IERR.NE.0)  GO TO 805
        CALL ACCLCS(2,IPNLCS,L(JJ),2,DLCS,IERR)
        IF(IERR.NE.0)  GO TO 805
        ILCN(JJ) = DLCS(21)
50      CONTINUE
C
C     SETUP POINTER FOR STRESS-STRAIN BUFFER
C
100     DO 165  JJ=1,2
        CALL ACCFES(1,IPNFES,IDBL,1,ILCN(JJ),0,0,IERR)
```

```
      IF (IERR.NE.0)  GO TO 801
C
C    GET ELEMENT STRESSES AND STRAINS
C
      CALL ACCFES(2,IPNFES,KINT,IREF,ILCN(JJ),SBUF,LEN,IERR)
      IF(IERR.NE.0)  GOTO 801
      M = 1
      IF (JJ.EQ.2)  GO TO 140
      DO 130 K=1,NSVAL
        J = M+1
        LL = M+3
        SIGMA(1,K) = SBUF(M)
        SIGMA(2,K) = SBUF(J)
        SIGMA(3,K) = SBUF(LL)
        M = M+4
130   CONTINUE
      GO TO 160
140   M = 17
      DO 150 K=1,NSVAL
        J = M+1
        LL = M+3
        EPSLN(1,K) = SBUF(M)
        EPSLN(2,K) = SBUF(J)
        EPSLN(3,K) = SBUF(LL)
        M = M+4
150   CONTINUE
160   IF (JJ.EQ.2)  GO TO 170
      IF(NT.GT.1)  GO TO 170
165   CONTINUE
C
C  GET X AND Y FOR JACOBIAN EVALUATION
C
170   CALL ACCELC(2,IPNELC,KINT,IREF,BUF,LENB,IERR)
      IF(IERR.NE.0)  GO TO 807
      M = 1
      DO 200 J=1,LENB,3
        K = J+1
        LL = J+2
        X(M) = BUF(J)
        Y(M) = BUF(K)
        Z(M) = BUF(LL)
        M = M+1
200   CONTINUE
      AREA = AREAQ(X,Y)
      XMP = 1.D0/AREA
C
C  CALCULATE FORCES AT THE GAUSS POINTS
C
      DO 250 J=1,NDOF
        DO 250 K=1,NSVAL
          BF(K,J) = 0.0
250   CONTINUE
C
C  LOOP OVER THE GAUSS POINTS
C
      DO 300 K=1,NSVAL
        PSI = GPL(1,K)
        ETA = GPL(2,K)
C
C  EVALUATE SHAPE FUNCTIONS AT THE GAUSS POINTS
C
```

```
          IF(ISTYP.EQ.2)  CALL EU2DLQ(PSI,ETA,KI,SHPF,DSHPL,
     *                    DSHPGX,DSHPGY,DETJ,X,Y,IERR)
          IF(ISTYP.EQ.4)  CALL EU2DPQ(PSI,ETA,KT,SHPF,DSHPL,
     *                    DSHPGX,DSHPGY,DETJ,X,Y,IERR)
          IF(IERR.NE.0)  GOTO 809
300     CONTINUE
        WRITE(10,855)
        DO 320 K=1,NSVAL
320     WRITE(10,854) K, (SIGMA(J,K),J=1,NSIG)
        IF(NT.GT.1)  GO TO 345
        WRITE(10,860)
        DO 330 K=1,NSVAL
330     WRITE(10,854) K, (EPSLN(J,K),J=1,NSIG)
        DO 340 J=1,NSIG
          DO 340 K=1,NSVAL
            SE(I) = SE(I) + SIGMA(J,K)*EPSLN(J,K)*DETJ
340     CONTINUE
C
C  CALCULATE SENSITIVITY VECTOR
C
        DPSIT(I) = - SE(I)
C
345     IF(I.NE.ICE(NC))  GO TO 820
C
C* CALCULATE PSI(B) - INTEGRAL OF STRESS FUNCTION G FOR ELEMENT
C
        IF(IST.EQ.1) GO TO 360
        DO 350 K=1,NSVAL
        VMS = (SIGMA(1,K)**2+SIGMA(2,K)**2-SIGMA(1,K)*
     *        SIGMA(2,K)+3*SIGMA(3,K)**2)**.5
350     PSIBPE(I) = PSIBPE(I) + VMS*DETJ
        GO TO 380
360     DO 370 K=1,NSVAL
          TMAX = ((.5*(SIGMA(1,K)-SIGMA(2,K)))**2+SIGMA(3,K)**2)
     *           **.5
370     PSIBPE(I) = PSIBPE(I)+(.5*(SIGMA(1,K)+SIGMA(2,K))+TMAX)
     *              *DETJ
380     PSIBT(I) = PSIBPE(I)*XMP
C
C
      GO TO 820
C  WRITE ERROR MESSAGES TO THE SCREEN
C
800   PRINT 870, IERR
      GO TO 820
801   PRINT 871, IERR
      GO TO 820
805   PRINT 875, IERR
      GO TO 820
807   PRINT 878, IERR
      GO TO 820
809   PRINT 876, IERR
      GO TO 820
C
C
820   CONTINUE
C
C
850   FORMAT(1X,'***ADJOINT LOAD IS APPLIED AT ELEMENT',I4)
851   FORMAT(1X,'TYPE OF STRESS IS ',A4)
854   FORMAT(1X,I2,2X,3(E16.8,2X))
```

```
855   FORMAT(1X,'GP',5X,'SIGMAX(GP)',8X,'SIGMAY(GP)',8X,
     *'SIGMAXY(GP)')
860   FORMAT(1X,'GP',5X,'EPSLNX(GP)',8X,'EPSLNY(GP)',8X,
     *'EPSLNXY(GP)')
870   FORMAT(1X,'ACCELM RETURNED WITH ERROR ',I4)
871   FORMAT(1X,'ACCFES RETURNED WITH ERROR ',I4)
875   FORMAT(1X,'ACCLCS RETURNED WITH ERROR ',I4)
876   FORMAT(1X,'EU2DFQ RETURNED WITH ERROR ',I4)
878   FORMAT(1X,'ACCELC RETURNED WITH ERROR ',I4)
1004  FORMAT(I4)
2001  FORMAT(A4)
C
      RETURN
      END
```

```
      SUBROUTINE ST1601(NT,I,L,PSIBTB)
CP***********************************************************
CP*
CP* ST1601: DESIGN SENSITIVITY VECTOR FOR A BENDING PLATE
CP*
CP**************************************************************
CP*
CP* DESCRIPTION:
CP*
CP*     . 'ST1601' COMPUTES THE DESIGN SENSITIVITY VECTOR FOR
CP*       A TRIANGULAR BENDING PLATE ELEMENT.
CP*
CP**************************************************************
CP*
CP*   NT       COUNTER FOR FINITE DIFFERENCE
CP*   I        EXTERNAL ELEMENT NO. BEING PROCESSED
CP*   L        EXTERNAL LOAD CASE NOS.
CP*   PSIBTB   STRESS CONSTRAINT FOR BENDING PLATE
CP*
CP***************************************************************
      INCLUDE '[AEGSDR.INC] IMPLIC.SPC'
      INCLUDE '[AEGSDR.INC] ACCIPN.MON'
      INCLUDE '[AEGSDR.INC] CNTL.MON'
      INCLUDE '[AEGSDR.INC] ELEDES.MON'
      INCLUDE '[AEGSDR.INC] SVECTR.MON'
      COMMON/LCSDES/DLCS(90)
C
      EQUIVALENCE (NDAT(98),IDBL)
C
      DIMENSION X(3),Y(3),Z(3),SIG(3,3),EPN(3,3),L(2),ILCN(2),
     *          SE(500),BUF(100),PSIBTB(500),SBUF(50),PSIBPE(500)
C
      DATA  KT/3/,IREF/1/,MPI/1/,ITYPE/1/
C
C
      SE(I) = 0.0D0
      PSIBTB(I) = 0.0D0
      PSIBPE(I) = 0.0D0
C
20    DO 50 JJ=1,2
C
C    GET INTERNAL LOAD CASE NUMBER
C
        CALL ACCLCS(1,IPNLCS,IDBL,1,0,IERR)
        IF(IERR.NE.0)  GO TO 805
        CALL ACCLCS(2,IPNLCS,L(JJ),2,DLCS,IERR)
        IF(IERR.NE.0)  GO TO 805
        ILCN(JJ) = DLCS(21)
C
C  GET PROPERTIES
C
        CALL ACCEPR(2,IPNEPR,IPTAB,0,BUF,LEN,IERR)
        IF (IERR.NE.0) GO TO 809
        PB(NT)=BUF(25)
        IF(NT.GT.1)  GO TO 100
50    CONTINUE
C
C  GET X AND Y FOR JACOBIAN EVALUATION
C
100     CALL ACCELC(2,IPNELC,KINT,IREF,BUF,LENB,IERR)
        IF(IERR.NE.0)  GO TO 807
```

```
          M = 1
          DO 110 J=1,LENB,3
            K = J+1
            LL = J+2
            X(M) = BUF(J)
            Y(M) = BUF(K)
            Z(M) = BUF(LL)
            M = M+1
110       CONTINUE
C
C     GET STRESSES FROM ORIGINAL LOAD CASE
C
          CALL SSMD16(X,Y,ILCN(1),SIG)
          IF(NT.GT.1)  GO TO 120
C
C     GET STRAINS FROM ADJOINT LOAD
C
          CALL SNMD16(X,Y,ILCN(2),EPN)
C
120       AREA = EUTRIA(X,Y)
          XMP = 1.DO/AREA
C
C     START INTEGRATION LOOP
C
          DO 400 IT=1,3
C
          VMS = DSQRT(SIG(IT,1)**2+SIG(IT,2)**2-SIG(IT,1)*SIG(IT,2)
     *          +3*SIG(IT,3)**2)
          TMAX = DSQRT((.5*(SIG(IT,1)-SIG(IT,2)))**2+SIG(IT,3)**2)
          IF(NT.GT.1)  GO TO 345
          IF (I.NE.ICE(NC)) GO TO 320
          IF (IST.EQ.1) GO TO 300
          IF (IST.GT.2)  GO TO 320
C
C     CALCULATE VON MISES STRESS SENSITIVITY TERM
C
          SE(I)=SE(I)+(AREA/3.DO)*VMS*XMP/PB(NI)
          GO TO 320
C
C     CALCULATE PRINCIPAL STRESS SENSITIVITY TERM
C
300       SE(I)=SE(I)+(AREA/3.DO)*(.5DO*(SIG(IT,1)+SIG(IT,2)+2
     *          *TMAX))*XMP/PB(NI)
C
C     CALCULATE THE SENSITIVITY VECTOR
C
320       DO 340 J=1,3
            SE(I) = SE(I) - (AREA/3.DO)*SIG(IT,J)*EPN(IT,J)
340       CONTINUE
          DPSITB(I) =  SE(I)
C
345       IF(I.NE.ICE(NC))  GO TO 400
C
C* CALCULATE PSI(B) - INTEGRAL OF STRESS FUNCTION G FOR ELEMENT
C
          IF(IST.EQ.1) GO TO 360
          PSIBTB(I) = PSIBTB(I) + (AREA/3.DO)*VMS*XMP
          GO TO 400
360       PSIBTB(I) = PSIBTB(I)+(AREA/3.DO)*(.5*(SIG(IT,1)+SIG(IT,2))
     *                +TMAX)*XMP
400     CONTINUE
```

```
C
C
      GO TO 820
C  WRITE ERROR MESSAGES TO THE SCREEN
C
800   PRINT 870, IERR
      GO TO 820
801   PRINT 871, IERR
      GO TO 820
805   PRINT 875, IERR
      GO TO 820
807   PRINT 878, IERR
      GO TO 820
809   PRINT 876, IERR
      GO TO 820
C
C
820   CONTINUE
C
C
850   FORMAT(1X,'***ADJOINT LOAD IS APPLIED AT ELEMENT',I4)
851   FORMAT(1X,'***TYPE OF STRESS IS ',A4)
854   FORMAT(1X,I2,2X,3(E16.8,2X))
855   FORMAT(1X,'GP',5X,'SIGMAX(GP)',8X,'SIGMAY(GP)',8X,
     *'SIGMAXY(GP)')
860   FORMAT(1X,'GP',5X,'EPSLNX(GP)',8X,'EPSLNY(GP)',8X,
     *'EPSLNXY(GP)')
862   FORMAT(1X,'ELEMENT ',I4)
870   FORMAT(1X,'ACCELM RETURNED WITH ERROR ',I4)
871   FORMAT(1X,'ACCFES RETURNED WITH ERROR ',I4)
875   FORMAT(1X,'ACCLCS RETURNED WITH ERROR ',I4)
876   FORMAT(1X,'ACCEPR RETURNED WITH ERROR ',I4)
878   FORMAT(1X,'ACCELC RETURNED WITH ERROR ',I4)
1004  FORMAT(I4)
2001  FORMAT(A4)
C
      RETURN
      END
```

```
C
C  ELEMENT ATTRIBUTES COMMON
C
      COMMON/ELEDES/IED(50),INTNN(32),IEXTNN(32),IDLD(4),IAF(32)
      EQUIVALENCE (IED(1),ITYP)     , (IED(2),ISTYP)   , (IED(3),NUNPE),
     *            (IED(4),NDOF)     , (IED(5),MAXDOF)  , (IED(6),NUELL),
     *            (IED(7),ILUMP)    , (IED(8),IACTV)   ,
     *            (IED(9),NRPT)     , (IED(10),KEXT)   ,
     *            (IED(11),KINT)    , (IED(12),KNEXT)  , (IED(13),KLINT),
     *            (IED(14),IESM)    , (IED(15),NINT)   , (IED(16),INCOPT),
     *            (IED(17),ISDP)    , (IED(18),IBOUN)  , (IED(19),NOC),
     *            (IED(20),IMATP)   , (IED(21),IMATC)  , (IED(22),MREL),
     *            (IED(23),NUMREL)  , (IED(26),NMAT)   , (IED(27),IPTAB),
     *            (IED(28),ISPTAB)  , (IED(29),BETA)   ,
     *            (IED(31),NSVAL)   , (IED(32),NSIG)   , (IED(43),KSIG )


      COMMON/SVECTR/DPSIT(500),DPSIB(500),DPSIH(500),DPSITB(500),
     *               TM(2),BW(2),BH(2),PB(2),ICT,ISAC,LCS,NLC,NC,
     *               IST,ICE(200)


C
C*** CURRENT IPNT POINTER STORAGE FOR ACCESS ROUTINE CALLS
C
      COMMON/ACCIPN/IPNELM(2),IPNNOD(2),IPNNEC(2),IPNEKM(2),IPNEMM(2),
     *               IPNEDK(2),IPNELC(2),IPNENF(2),IPNBER    ,IPNMAT    ,
     *               IPNNCF(2),IPNAFL(3),IPNEEN(2),IPNAND(2),IPNEFR(2),
     *               IPNNSF(2),IPNBMF(2),IPNFES(2),IPNLCS(2),IPNSTK    ,
     *               IPNRES    ,IPNASD    ,IPNCLF(2),IPNEPR    ,IPNMPR    ,
     *               IPNLMI(2),IPNBLK    ,IPNCNU(2),IPNNDF(2),IPNSGM(2),
     *               IPNNRF    ,IPNTEM    ,IPNISS(2),IPNSTM


C
C*** GLOBAL CONTROL PARAMETERS
C
      COMMON /CNTL/ NETY(200),NDAT(300),NLINE,NWID4,NWID1,NAUX,
     *               IHEADR(132),IAUX(33,5)
```