Reduced Complexity Structural Modeling for Automated Airframe Synthesis

Prabhat Hajela
Department of Engineering Sciences
University of Florida
Gainesville, Florida

Prepared for
Ames Research Center
under Contract NCA2-IR240-401
May 1987
ABSTRACT

The present report documents a procedure for the optimum sizing of wing structures that is based on representing the built-up finite element assembly of the structure by equivalent beam models. The reduced-order beam models are computationally less demanding in an optimum design environment which dictates repetitive analysis of several trial designs. The design procedure is implemented in a computer program that requires geometry and loading information to create the wing finite element model and its equivalent beam model, and provides a rapid estimate of the optimum weight obtained from a fully stressed design approach applied to the beam. The synthesis procedure is demonstrated for representative conventional-cantilever and joined wing configurations.

INTRODUCTION

Automated design synthesis programs provide a significant capability for assessing new concepts in aircraft design. Such concepts invariably entail a multidisciplinary synthesis environment that is characterized by complex analysis codes for various participating disciplines. Since optimum design involves repetitive analysis, the computational costs can be significant, particularly if no effort is made to substitute approximating strategies in lieu of more detailed analyses. The optimum synthesis scenario for the present work resulted from studies directed at the optimum weight evaluation of the joined wing. The joined wing (Ref. 1) is a general concept that seeks aerodynamic and structural advantages by replacing the horizontal tail in a conventional airplane design by a forward swept wing that is joined to the front wing at the tip. The resulting truss-like structure is claimed to have higher stiffness and a significant potential for structural weight savings. References 2 and 3 document the results of studies that primarily examined the sensitivity of key geometric parameters on the optimum weight of the joined wing design. In both of these studies, a finite element analysis capability was employed in conjunction with a nonlinear programming based optimization algorithm to determine the mathematical optima. The computational requirements for these solutions were substantial, thereby precluding other combinations of geometric parameters from consideration. Although these studies were successful in establishing preliminary trends of the optimum weight, the approach of using detailed finite element models with mathematical programming based optimization algorithms was considered inappropriate for a more detailed study. Such a detailed multidisciplinary synthesis study would include optimization for aerodynamics and stability/control in addition to the structural performance.

The purpose of this study was to formulate a procedure for optimum structural design with limitations on computational requirements enforced by a multidisciplinary design environment. The strategy adopted for this task was to replace the built-up finite element model of the wing structure by a lower order beam framework model that would simulate the strength and stiffness characteristics of the former with a minimum loss in accuracy. Subsequent sections of this report describe the approach in greater detail, including its numerical implementation into a synthesis program. An annotated listing of the fortran programs and the related data files can be obtained as an appendix to this report.
THEORETICAL BACKGROUND

The configuration of an automated synthesis procedure requires careful consideration in the selection of the analysis and optimization capabilities. These programs must incorporate approximating strategies to reduce the overall computational effort and at the same time must retain any peculiar characteristics of the problem. This is particularly true in the case of the joined wing which has rather unique displacement and stiffness characteristics. The present section develops the theoretical concepts that form the basis for the design strategy proposed in this report.

Joined Wing Structural Analysis

The joined wing configuration results in a stiff load carrying structure which has been shown to yield lower weights than the conventional wing-tail design. The potential for weight reduction is most simply explained by a 'tilted-truss' visualization of the fore-aft wing combination as seen in Figure 1. The front and aft wings form a truss with the primary load carrying plane inclined to the horizontal by an angle which is determined by the dihedral angle of the wings. The aerodynamic loads can be resolved into the inplane and out-of-plane components. The load component perpendicular to the plane of the truss tends to concentrate material on the upper surface of the leading edge and the lower surface on the trailing edge of the wings. The effective beam depth is thus determined by the chord length as opposed to the thickness profile for conventional wings.

The structural joint between the front and aft wings is also critical in determining the optimum weight of a configuration. It is an area of stress concentration and its own rigidity in bending and torsion determines the material distribution on the front and aft wings. Furthermore, the location of the joint along the span also influences the structural weight and the optimal material distribution. The formulation of a mathematical design model should therefore give special consideration to these characteristics.

Wing Finite Element Modeling

The finite element models for the conventional and joined wing configurations studied in this effort were built-up models with axial rod elements and quadrilateral membrane elements. The plan view and a typical cross section are shown in Figures 2 and 3. Such a single cell representation is considered appropriate in the preliminary design studies for which the model is intended. The consistency of the design and analysis models is also an important consideration in this exercise. At least two chordwise panels are essential in the design model to allow for unsymmetrical distribution of material that is predicted by the tilted bending-axis hypothesis. For an improvement in the stress and displacement results, the analysis model can have any even number of panels in the chordwise direction. The number of spanwise stations is at the discretion of the user and is generally selected to keep the panel aspect ratio close to unity. Ribs, modeled by quadrilateral membrane elements were added to the built-up structure at a specified number of locations.

The joint between the front and aft wings was modeled by a framework of
beam elements as shown in Figure 4. The beam sectional properties were assigned numerical values to generate a structure that was extremely rigid in extension, bending and twisting deformations. The aerodynamic loads are specified as an array of forces at the leading and trailing edges of the structure. A unique feature of this study is the automated generation of the finite element model and will be discussed in a later section.

Beam Representation of Wing Models

A typical finite element model of a joined wing with a relatively coarse mesh has more than seven hundred degrees of freedom. Repetitive analysis in an optimization exercise with such a large model is prohibitive from a computational standpoint. The approach adopted in the present work replaces the detailed built-up models by equivalent beam models in the design loop.

A cantilever beam has deflection and stiffness characteristics that are very similar to a conventional cantilevered wing. The wing can be regarded as a tapered plate with one end built-in and the other free. The deflection of this plate in its primary bending mode can be represented by a cantilever beam with appropriately matched moment of inertia characteristics. An exact value of the wing moment of inertia cannot be used for the beam as it would result in an artificially stiff structure. This difference is attributed to the phenomenon of shear lag. If one considers the upper and lower surfaces of the wing model as flange elements, shear lag is the description of the state in which the flange strains decrease asymptotically when moving away from the web section. Hence, the bending stiffness computed using the full width of the flange for the moment of inertia would result in a conservative estimate. A reduced flange width should be used and this is dependent on the geometry of the web and flange, wing span, boundary conditions and the bending load distribution. The beam should therefore have a moment of inertia equal to that of the wing multiplied by a reduction factor to account for wide flange effects. In the present exercise, this factor was computed numerically by a process of matching the response of the built-up model to the simplified beam model.

The cross sectional properties of the wing that are represented on the equivalent beam model are the moments of inertia $I_{xx}$ and $I_{yy}$, the product of inertia $I_{xy}$, and the torsional constant $J$ (see Figure 4). The volume of the material per unit length of the wing span is introduced as an additional variable to establish a weight relation between the beam and wing models. The torsional constant $J$ is computed for a thin-walled closed section by the Bredt formula (Ref. 4)

$$J = \frac{4A^2}{ds}$$  \hspace{1cm} (1)

The product of inertia term is essential to accommodate the unsymmetrical bending that is present in conventional swept wings and the joined wing configurations. The beam cross section to which these sectional properties are attributed is shown in Figure 5. The five sectional properties described above can be expressed in terms of an equal number of independent wall thicknesses of the cross section. The choice of this cross section was primarily dictated by the anticipated unsymmetrical material distribution in the design of the joined wing. The wall thicknesses, updated during the design process, can result in either a symmetrical or an unsymmetrical cross
The deflections of the equivalent beam structure under the applied loads were computed from a finite element program. The specification of a special beam cross section precluded the computation of element stresses in the same finite element program. These stresses are required for the strength sizing of the beam and were computed in a post-processing program using the following unsymmetrical bending stress relationship (Ref. 5)

$$\sigma_{xx} = -\left(\frac{M_y y z + M_z y z}{I_{yy}^z} - \frac{M_y y z + M_z y z}{I_{zz}^y}\right)$$  \hspace{1cm} (2)$$

Here $M_y$ and $M_z$ are the bending moments about the $y$ and $z$ axes, respectively. The distances $y$ and $z$ are measured in a centroidal axes system shown in Figure 6. The bending moments $M_y$ and $M_z$ are computed from the moment-curvature relations

$$M_y = -EI_y \frac{d^2 w}{dx^2}$$  \hspace{1cm} (3)$$

$$M_z = -EI_z \frac{d^2 v}{dx^2}$$  \hspace{1cm} (4)$$

where $v$ and $w$ are the deflections in the $y$ and $z$ directions, respectively. The curvatures in equations (3) and (4) were obtained from the displacement field by a central finite difference approximation (see Fig. 7)

$$\frac{d^2 w}{dx^2} \bigg|_i = \frac{(w_{i+1} - 2w_i + w_{i-1})/(\Delta x)^2}{(5)$$

$$\frac{d^2 v}{dx^2} \bigg|_i = \frac{(v_{i+1} - 2v_i + v_{i-1})/(\Delta x)^2}{(6)$$

for the station at the root, the boundary conditions at a fixed support can be invoked to obtain the approximation

$$w_{-1} = w_1$$  \hspace{1cm} (7)$$

$$v_{-1} = v_1$$

To obtain better approximations for the second order derivatives described above, the step size $\Delta x$ was reduced by increasing the number of nodes in the beam model. The same effect can also be achieved by using an interpolated
polynomial obtained from the displacement corresponding to a coarser grid model. An approach such as the present one allows the specification of an arbitrary cross section and can be used in conjunction with any finite element displacement analysis program. It can also be used with a displacement field obtained from a classical Galerkin or a Rayleigh Ritz type solution.

Optimum Structural Design

There are several options available to size the wing and the equivalent beam structures for minimum weight and a prescribed structural strength. The general mathematical problem statement for this problem can be written as

Minimize \( W(\mathbf{d}) \) \hspace{1cm} (8)

Subject to \( g_j(\mathbf{d}) < 0 \) \hspace{1cm} j = 1,2,\ldots,m \hspace{1cm} (9)

and \( d^l_i < d_i < d^u_i \) \hspace{1cm} i = 1,2,\ldots,n \hspace{1cm} (10)

Here \( W \) is typically the structural weight; \( \mathbf{d} \) is a vector of design variables with prescribed lower and upper bounds on its components given as \( d^l_i \) and \( d^u_i \), respectively. The inequality constraints \( g_j \) can be used to prescribe bounds on strength and nodal displacements. This approach can be integrated into the present design strategy with minor modifications but is computationally demanding in the presence of a large number of design variables and constraints. An alternative strategy referred to as the fully-stressed design approach was implemented instead. This approach is based on the hypothesis that a strength governed design is optimal when all elements are stressed to their maximum permissible limits. The assumption is valid for singly loaded structures that do not have multiple load paths (Ref. 6). The built-up wing finite element model is a redundant structure and cannot be strictly considered as optimum in the fully stressed state. The beam model, however, is considered a good candidate for the fully stressed design philosophy. In previous work it has been shown that the fully stressed design strategy provides a good first estimate of the optimum weight for even mildly redundant structures.

A stress ratio algorithm was implemented in the present work where the \( i \)-th wall thickness at the \( j+1 \)-th iteration is given by

\[ t_{i}^{j+1} = t_{i}^{j} \frac{\sigma_i^j}{\sigma_{\text{all}}} \] \hspace{1cm} (11)

The allowable strengths in compression and tension are taken to be equal in the above approach. Bending stresses were recovered from six locations on the cross section and these are labeled one to six in Figure 5. The vertical sections 1-6 and 3-4 were kept equal in the design process. This equality was enforced after each thickness had been obtained independently from equation 11, with the greater value of thickness assigned to the two sections. Each element was sized on the basis of the maximum stress on the element. In the present exercise the stresses are recovered at six locations with each location corresponding to an extremity for an element. More stress recovery
points can be introduced with an insignificant addition in computational time.

Convergence in the stress ratio algorithm is very rapid in the initial stages of the design. When approaching close to the optimum, the design iterations illustrate a 'tail-like' characteristic. Approaches which combine a gradient based search algorithm with such a stress ratio approach have been proposed and will be examined in future work. The other drawback in this approach is the lack of constraints to limit the displacements at nodal locations. This can be circumvented by following up the stress ratio sizing by a redesign based on the energy level in each element with the objective of forcing the element energies to comparable values for all elements in the structure.

**COMPUTER IMPLEMENTATION**

A stated objective of the present work was to generate an automated synthesis procedure for airframe structures suitable for adaptation in a multidisciplinary design environment. In particular, the program was to interact with aerodynamic design codes that were in turn driven by external optimization programs. Thus, the generation of the wing finite element model, its reduction to an equivalent beam model and the subsequent optimum design of the beam had to be completely automated. *Engineering Analysis Language* (EAL, see Ref. 7) was used for all structural analysis in the present task. The fortran programs that automatically generate runstreams for various segments of the program are currently written for EAL. However, these programs can be adapted for other finite element environments with minor modifications. The organization and execution of these program is controlled in the Command Language feature on DEC systems. A flowchart depicting the order of execution is shown in Figure 8. The primary function of each module is discussed next and the corresponding input/output requirements are detailed in the appendix.

**COORDS:**

This program provides an automated finite element modeling capability for conventional and joined wings. The user provides input information pertaining to the type of structure, semi-span, root and tip chords, thickness ratio, 1/4-chord sweep and the dihedral angle. Additional information is also provided on the number of chordwise and spanwise stations, sizes of elements, and the number of ribs in the model. The program then generates a finite element model of the structural box using axial rod elements for stringers and quadrilateral membrane elements for the plate sections. This model includes complete description of nodal co-ordinates, element connectivity and distribution of nodal loads. In its present form, this information is available as an EAL input runstream. Suitable modifications of format statements can adapt this program for other finite element codes. The program additionally generates data files that provide input data for programs executed later in the optimization sequence. In particular, these files transfer information related to wing geometry, loads, cross sectional geometry and element sizes.

**MOMNT:**

The cross sectional properties of the wing finite element model are
computed in this program. At each of the span stations specified by the user, the sectional moments and products of inertia, torsional constant and mass per unit span are computed and data files generated to transfer this information to a program which generates an equivalent beam model with the same sectional properties.

BEAM:

For the built-up finite element model of the wing created in program COORDS, this program creates an equivalent finite element model with beam elements located at the wing elastic axis. For a conventional wing, the equivalent model is a cantilevered beam with section properties equal to those obtained from MOMNT. The number of beam elements used in modeling this equivalent beam is identical to the number of spanwise stations entered in COORDS. The modeling is similar for the joined wing configuration with the exception that there is an equivalent beam for each of the front and aft wings and the joint between the beams is simply modeled as a common node. The two equivalent beams are built-in at the root and permit the joint to be located arbitrarily along the span. Element connectivity, load specifications and other execution runstream are in context of EAL but can be modified for other finite element programs.

SHLAG:

This program is used to provide the correction required in the wing sectional properties before they are transferred to the beam model. The moment of inertia about the primary bending axis of the wing would result in a stiffer beam because of shear lag effects. This program reads the maximum displacements, \( W \), of the wing and beam models and defines a constant \( \rho \) where

\[
\rho = \frac{\left(\frac{W_{\text{max}}}{\text{beam}}\right)}{\left(\frac{W_{\text{max}}}{\text{wing}}\right)} \cdot \frac{(EI)_{\text{wing}}}{(EI)_{\text{beam}}}
\]

The chord on the beam and wing structural box were kept the same and the height of the beam section was changed to account for the shear lag effects. If the moment of inertia corresponding to the thin sidewalls in the beam is neglected, the bending rigidity is proportional to the square of the depth, \( d \)

\[
(EI) \sim d^2
\]

\[
d_{\text{wing}} = \sqrt{\rho} \cdot d_{\text{beam}}
\]

Numerical evaluation with test cases shows this to be a reasonable assumption. The effect of wing sweep and dihedral was also incorporated in this program.

BSAP:

The section properties of a general beam section with five independent wall thicknesses as shown in Figure 5 are computed in this program. All section properties are computed about a centroidal axis which is also computed
in this segment. Since the finite element program EAL needs section properties about the principal axes system for a section, additional computations are necessary to transform centroidal properties to principal axes properties. The orientation of the principal axes system with respect to the global axes system is necessary to complete element coordinate axes definitions and was therefore computed in this program. In addition, the location of the six stress recovery points in terms of y-z coordinates change with each iteration and were computed here.

MODISP:

This program is identical to BEAM except that it is configured to double the number of elements from the previous beam finite element model. This was necessary to increase the number of nodal points at which the displacements were computed so as to enhance the quality of the finite difference approximations for curvature. The section properties for each element were transferred from the BSAP program and used here to create an EAL runstream for the equivalent beam.

STRESS:

The bending stresses necessary in the resizing algorithm were computed in this program. The beam displacements are read in from a data file and used to compute the curvatures and hence also the components of the sectional bending moment. At the stress recovery points obtained in BSAP, the stresses were computed using equation 2. These stresses were placed in an output file to be accessed by the design optimization program.

FSD:

This is the computer implementation of the fully stressed design strategy discussed in an earlier section. The wall thicknesses of each element are transferred to this program as are the stresses computed in program STRESS. The stress ratio algorithm (Eq. 11) is used to resize the wall thicknesses based upon the most recently computed stresses and a user specified allowable stress. The weight computed in three consecutive passes of the sizing algorithm is used to terminate the optimization iterations based upon a user specified value for the relative change in the weight.

In the sizing algorithm there are two items of which a user should be aware. The vertical walls of the beam section are kept equal and the largest stress of the four corners of the section is used to determine this thickness. Additionally, lower and upper bounds are imposed on the thickness of the walls and these stem from two considerations. The wall thicknesses must be kept such that they are physically meaningful dimensions within constraints imposed by fixed width and depth of the section. Furthermore, the wall thicknesses should not be so large as to create a conflict with the thin wall assumptions used in computing the beam sectional properties.

NUMERICAL RESULTS

The structural resizing procedure described in the preceding sections was validated through a sequence of test problems consisting of both the joined and the conventional wing configurations. The primary objective of the present study was to establish trends on the deflection characteristics and
the optimum structural weight as predicted by the equivalent beam models, and to compare these trends with those obtained from a fully stressed design of a built-up finite element model of the wing. The geometry parameters considered in this validation study include the wing sweep and dihedral angle, and the spanwise location of the joint between the front and aft wings for the joined wing configuration. The numerical results for the various test cases are summarized in Tables 1 - 4. The shear lag effects described in an earlier section are also dependent upon the geometry of the configuration. The variation of these influences with sweep and dihedral were established by a series of numerical experiments and the trends were mapped into cubic spline functions for the purpose of interpretation for intermediate values. These spline functions are valid for sweep angle variations between 10° and 30° and dihedral angle variations between 4° and 20°.

Table 1 lists the optimum structural weights for a conventional cantilever wing with a dihedral angle of 4° and various sweep angles. The wing span and the root and tip chords were held to constant values as the sweep angle was varied. An increase in structural weight is expected with increasing sweep angle and is clearly indicated by both the wing and the equivalent beam models. The material distribution on the beam was similar to a conventional wing with maximum material located at the root section. Results for a similar parametric variation of the sweep angle for similar loading of a joined wing configuration are shown in Table 2. The front and aft wings were identical and have a dihedral of ±4°, respectively. The optimum weight of the beam model displays the same qualitative trend as the built-up finite element model. Furthermore, the deflection characteristics of the two models also display the same behavior, with the maximum displacement occurring at about 70% of the semi-span. Consequently, the material distribution along the span also displays similar qualitative trends.

The influence of the dihedral angle on the optimum weight is illustrated in Table 3. With increasing dihedral angle, the tilted-truss effect of the joined wing structure becomes more predominant. The effective depth of the beam is increased and provides for a reduction in the structural weight. The effect of material concentration at the upper leading edge and the lower trailing edge of the wing structural box was present in the built-up finite element model and was clearly evident in the equivalent unsymmetrical beam cross section obtained from a fully-stressed design of the beam model. The results presented are for a ±25° sweep of the quarter chord lines of the front and aft wings.

Table 4 demonstrates the influence of moving the joint between the front and the aft wings inboard from the tip. A decrease in structural weight is demonstrated in both the built-up wing models and the equivalent beam representations. These results are for a dihedral of ±4° and a quarter chord sweep of ±25°, respectively.

CONCLUSIONS AND RECOMMENDATIONS

This report describes a procedure for optimum structural design in a preliminary design environment where computational efficiency is a primary consideration. Rapid estimates of the optimum structural weight of wing structures for a specified load are obtained by the automated synthesis of representative beam models which have considerably fewer degrees of freedom. The underlying design criterion for minimum weight is to stress each element
to its ultimate load carrying capacity. The most significant advantages of the proposed procedure are

a) automation of the design process to make the synthesis procedure easily adaptable in a multidisciplinary design environment. This includes an automated creation of all finite element models required in the process

b) considerable savings in computational resources over the conventional approach of optimizing detailed built-up models of the wing structure.

Although the automated process provides a reasonable strategy for preliminary design, additional effort is required to enhance its effectiveness as a robust design tool. These modifications can be summarized as follows

a) Prescribing bounds on nodal displacements in addition to constraints on strength. In the present approach, this can be obtained by creating a sizing ratio based on the strain energy within the element

b) The potential of the present approach can be extended further by adding the ability to recover element sizes of the built-up finite element of the wing from the optimized sections of the beam. In the present model, the five independent membrane element thicknesses can be related to the five optimized values of the sectional properties through nonlinear relations. Reasonable qualitative estimates of these dimensions can be obtained from the optimized beam and used as input to a nonlinear equation solver to recover more precise values.
ACKNOWLEDGEMENTS

The author gratefully acknowledges the contributions of graduate student J. L. Chen during the course of this study. The author would also like to thank Mr. W. H. Reuter and J. R. Dennison for their assistance in the programming effort involved in the project.

REFERENCES


<table>
<thead>
<tr>
<th>Case #</th>
<th>1/4 chord sweep A</th>
<th>Wing FEM Weight-lbs</th>
<th>Beam FEM Weight - lbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10°</td>
<td>1056.10</td>
<td>960.75</td>
</tr>
<tr>
<td>2</td>
<td>15°</td>
<td>1064.26</td>
<td>947.16</td>
</tr>
<tr>
<td>3</td>
<td>20°</td>
<td>1112.41</td>
<td>1052.31</td>
</tr>
<tr>
<td>4</td>
<td>25°</td>
<td>1216.82</td>
<td>1118.14</td>
</tr>
<tr>
<td>5</td>
<td>30°</td>
<td>1254.27</td>
<td>1171.11</td>
</tr>
</tbody>
</table>

Table 1. Numerical results for an elliptically loaded conventional cantilever wing.

(semi-span loads = 30,000 lbs, b = 450", c_R = 60", c_T = 24", δ = 4°, t_R = 0.12)
Table 2. Numerical results for elliptically loaded joined wings with joint located at wing tip. (semispan load=30,000lbs)

<table>
<thead>
<tr>
<th>Case #</th>
<th>1/4 chord sweep Λ</th>
<th>Wing FEM Weight - lbs</th>
<th>Beam FEM Weight - lbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10°</td>
<td>707.87</td>
<td>761.65</td>
</tr>
<tr>
<td>2</td>
<td>15°</td>
<td>802.03</td>
<td>859.77</td>
</tr>
<tr>
<td>3</td>
<td>20°</td>
<td>891.55</td>
<td>922.66</td>
</tr>
<tr>
<td>4</td>
<td>25°</td>
<td>958.62</td>
<td>967.15</td>
</tr>
<tr>
<td>5</td>
<td>30°</td>
<td>1101.57</td>
<td>1098.67</td>
</tr>
</tbody>
</table>

Front wing : b=450", c_R=60", c_T=24", δ=+4°, t_R=0.12
Aft wing : b=450", c_R=60", c_T=24", δ=-4°, t_R=0.12
Table 3. Numerical results for elliptically loaded joined wings with joint located at wing tip. (semispan load=30,000lbs)

<table>
<thead>
<tr>
<th>Case #</th>
<th>Dihedral Angle $\delta$</th>
<th>Wing FEM Weight - lbs</th>
<th>Beam FEM Weight - lbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4°</td>
<td>958.60</td>
<td>967.15</td>
</tr>
<tr>
<td>2</td>
<td>8°</td>
<td>826.67</td>
<td>863.55</td>
</tr>
<tr>
<td>3</td>
<td>12°</td>
<td>753.49</td>
<td>787.63</td>
</tr>
<tr>
<td>4</td>
<td>16°</td>
<td>657.43</td>
<td>678.90</td>
</tr>
<tr>
<td>5</td>
<td>20°</td>
<td>595.37</td>
<td>618.87</td>
</tr>
</tbody>
</table>

Front wing : $b=450''$, $c_r=60''$, $c_T=24''$, $\Lambda=+25^\circ$, $t_R=0.12$
Aft wing : $b=450''$, $c_r=60''$, $c_T=24''$, $\Lambda=-25^\circ$, $t_R=0.12$
Table 4. Numerical results for elliptically loaded joined wings with joint located inboard from front wing tip. (semispan load = 30,000 lbs)

Front wing: \( b = 450" \), \( c_R = 60" \), \( c_T = 24" \), \( \Lambda = +25^\circ \), \( t_R = 0.12 \)
Aft wing: \( c_R = 60" \), \( c_T = 60" \), \( \Lambda = -25^\circ \), \( t_R = 0.12 \)

* \( B1 \) and \( B2 \) are the front and aft wing semi-spans, respectively.
Figure 1. The fore-aft wings in a joined wing configuration can be idealized as a planar truss with the plane of the truss inclined to the horizontal.
Figure 2. Finite element model of the wing structural box with eight stringer elements.
Figure 3. A beam element grid joint between the front and aft structural boxes. Each beam is rigid in bending and twisting deformations.

Figure 4. Cross sectional properties of wing computed for typical section with six stringer elements. $\bar{A}$ is the cross sectional area at enclosed by the section at a spanwise station.
Figure 5. Unsymmetrical beam section used to model the wing cross sectional properties.

Figure 6. General cross section depicting the y-z axes system used in beam bending stress computations.
Figure 7. Grid point nomenclature for the finite difference representation of beam deformations and curvature.
Figure 8. Flow chart depicting the organization of the optimum design procedure.
APPENDIX

This appendix documents the input/output file specifications and an annotated listing of all fortran programs and pertinent data files used in the optimum synthesis procedure. The function of each of these programs is described in the report. Table A-l summarizes the data files used for input/output functions in each of the major program segments.

The annotated listings are self explanatory and can be used as a guide if program modifications are attempted. Numerical constants that are hardwired into the programs and cannot be altered by input data specifications are identified in these listings.
<table>
<thead>
<tr>
<th>PROGRAM</th>
<th>INPUT FILES</th>
<th>OUTPUT FILES</th>
</tr>
</thead>
<tbody>
<tr>
<td>COORDS</td>
<td>IGO.DAT, THICK.DAT, WING.DAT</td>
<td>INPUT.DAT, BEAM1.DAT, SL.DAT, BCSD.DAT, THICK.DAT WING.COM</td>
</tr>
<tr>
<td>MOMNT</td>
<td>INPUT.DAT</td>
<td>BEAM2.DAT</td>
</tr>
<tr>
<td>MODISP</td>
<td>TRANSFR.DAT, BCSD1.DAT</td>
<td>MODE.COM</td>
</tr>
<tr>
<td>BEAM</td>
<td>BEAM1.DAT, BEAM2.DAT</td>
<td>BEAM.COM, TRANSFR.DAT</td>
</tr>
<tr>
<td>SHLAG</td>
<td>KAPPA.DAT, WING.DAT, FOROO3.DAT, FOROO4.DAT</td>
<td>KAPPA.DAT</td>
</tr>
<tr>
<td>BSAP</td>
<td>SKIN.DAT, BCSD.DAT, KAPPA.DAT, GEOM.DAT</td>
<td>INERT.DAT, BCSD1.DAT</td>
</tr>
<tr>
<td>STRESS</td>
<td>TRANSFR.DAT, INERT.DAT, FOROO4.DAT, BCSD.DAT</td>
<td>STRESS.DAT</td>
</tr>
<tr>
<td>FSD</td>
<td>WING.DAT, KAPPA.DAT, ALST.DAT, SKIN.DAT, BCSD.DAT, FOROO4.DAT, STRESS.DAT</td>
<td>SKIN.DAT, FSDIF.DAT</td>
</tr>
<tr>
<td>WFSD</td>
<td>WING.DAT, FOROO2.DAT, ALST.DAT, THICK.DAT, FOROO1.DAT</td>
<td>THICK.DAT, FSDIF.DAT</td>
</tr>
</tbody>
</table>
$SET VERIFY
$ NSTOP:==0
$ LOOP1:
$ RUN COORDS
$ SET NOVERIFY
$ @WING
$ SET VERIFY
$ DEL XXX.*;*
$ RUN MOMNT
$ RUN BEAM
$ SET NOVERIFY
$ @BEAM
$ SET VERIFY
$ DEL FOR001.DAT;*.FOR002.DAT;*.XXX.*;*
$ DEL BEAM2.DAT;*
$ RUN SHLAG
$ SET VERIFY
$ ITER:==1
$ NFLAG:==0
$ LOOP1:
$ RUN BSAP
$ RUN MODISP
$ SET NOVERIFY
$ @MODE
$ SET VERIFY
$ DEL XXX.*;*
$ RUN STRESS
$ RUN FSD
$ APPEND FOR057.DAT FSDSS
$ DEL FOR00*.*;*
$ DEL MODE.COM;*
$ DEL STRESS.DAT;*
$ DEL INERT.DAT;*
$ ITER=ITER+1
$ IF NFLAG.EQ.1 THEN GOTO EXT1
$ IF ITER.EQ.20 THEN GOTO EXT1
$ GOTO LOOP1
$ EXT1:
$ EXIT
| 5.000000E-02 | 5.000000E-02 | 5.000000E-02 | 5.000000E-02 | 5.000000E-02 |
| 5.000000E-02 | 5.000000E-02 | 5.000000E-02 | 5.000000E-02 | 5.000000E-02 |
| 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| 0.500000 | 0.500000 | 0.500000 | 0.500000 | 0.500000 |
| 0.500000 | 0.500000 | 0.500000 | 0.500000 | 0.500000 |
| 0.500000 | 0.500000 | 0.500000 | 0.500000 | 0.500000 |
| 0.500000 | 0.500000 | 0.500000 | 0.500000 | 0.500000 |
| 0.500000 | 0.500000 | 0.500000 | 0.500000 | 0.500000 |
| 0.500000 | 0.500000 | 0.500000 | 0.500000 | 0.500000 |
| 0.500000 | 0.500000 | 0.500000 | 0.500000 | 0.500000 |
| 0.500000 | 0.500000 | 0.500000 | 0.500000 | 0.500000 |
| 0.500000 | 0.500000 | 0.500000 | 0.500000 | 0.500000 |
| 0.500000 | 0.500000 | 0.500000 | 0.500000 | 0.500000 |
| 0.500000 | 0.500000 | 0.500000 | 0.500000 | 0.500000 |
| 0.500000 | 0.500000 | 0.500000 | 0.500000 | 0.500000 |

5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
0.500000 0.500000
beam wall thickness—five independent thicknesses for each spanwise station.

starting estimates of weights
JOINED

450. 60. 24. 6 10 10 15. 6. .12 0.7
450. 60. 24. 6 10 10 15. 6. .12 0.7
60000.

b, cr, ct, nstr,nsta,nrib,sweep,dihedral, thickness ratio, chordwise taper

6 - NSTR

.001 - convergence tolerance
35000.00 - allowable stress

20 - NSTA+NSTA2

20.0 10.0 - Arbitrary values of starting weights

ALST.DAT

GEOM.DAT
### VARIABLES

**NUM** = TOTAL NO. OF NODES.
**NRT** = TOTAL NO. OF RIB STATIONS.
**NTYP1** = FIRST FOUR LETTERS OF WING TYPE (JOIN or SING).
**NSTA** = NO. OF STATIONS ON FORWARD BEAM.
**NSTA2** = NO. OF STATIONS ON AFT BEAM.
**NRIB** = NO. OF RIB STATIONS ON FORWARD WING.
**NRIB2** = NO. OF RIB STATIONS ON AFT WING.
**NCOM** = NODE NO. AT WHICH THE WINGS ARE JOINED (COMMON NODE).
**XCOM, YCOM, ZCOM** = X, Y, AND Z COORDINATES OF COMMON NODE.
**B** = SEMI-SPAN OF FORWARD WING.
**B2** = SEMI-SPAN OF AFT WING.
**XISTA(I)** = X COORDINATE OF ITH RIB STATION.
**RLOAD(I)** = LOAD @ Ith STATION ON BEAM.
**RMOMNT(I)** = MOMENT ABOUT Ith STATION ON BEAM.
**XNODE(I)** = X COORDINATE OF Ith NODE ON BEAM.
**YNODE(I)** = Y COORDINATE OF Ith NODE ON BEAM.
**ZNODE(I)** = Z COORDINATE OF Ith NODE ON BEAM.
**SKIN(I)** = SKIN THICKNESSES OF WING TORQUE BOX.
**SECT(I,J)** = SECTION PROPERTIES OF WING (BETWEEN RIBS).

### PROGRAM BEGINS

```fortran
DIMENSION XNODE(30), SECT(25,5), XISTA(50), RLOAD(50)
DIMENSION RMOMNT(50), SKIN(50)
DIMENSION YNODE(30), ZNODE(30)

OPEN(UNIT=10, NAME='BEAM.COM', STATUS='NEW')
OPEN(UNIT=11, NAME='BEAM1.DAT', STATUS='OLD')
OPEN(UNIT=12, NAME='BEAM2.DAT', STATUS='OLD')
OPEN(UNIT=2, NAME='TRNSFR.DAT', STATUS='NEW')

READ(11,1005) NTYP1, NTYP2
1005 FORMAT(A4,A2)
WRITE(2,1006) NTYP1
1006 FORMAT(A4)
READ(11,*) NSTA, NRIB, B
WRITE(2,*) NSTA, NRIB, B
NRT=NRIB
SPAN=B
IF(NTYP1 .EQ. 'JOIN') THEN
  READ(11,*) NSTA2, NRIB2, B2
  WRITE(2,*) NSTA2, NRIB2, B2
  NRT=NRIB+NRIB2
```

---

A-7
SPAN=B2
END IF

SHIFT BOTH BEAMS BACK & DOWN SO THAT THE ORIGIN OF THE FORWARD BEAM IS AT THE POINT(0,0,0).

NUM=NSTA+1
IF(NTYP1 .EQ. 'JOIN') NUM=NUM+NSTA2
WRITE(2,*),NUM
DO 75 I=1,NSTA+1
READ(11,*),XNODE(I),YNODE(I),ZNODE(I),RLOAD(I),RMOMNT(I)
IF(I.EQ.1)THEN
  DY=YNODE(1)
  DZ=ZNODE(1)
ENDIF
YNODE(I)=YNODE(I)-DY
ZNODE(I)=ZNODE(I)-DZ
WRITE(2,*),XNODE(I),YNODE(I),ZNODE(I)
IF(XNODE(I) .LT. SPAN)THEN
  NCOM=I
  XCOM=XNODE(I)
  YCOM=YNODE(I)
  ZCOM=ZNODE(I)
ENDIF
CONTINUE

SAME FOR AFT WING.

IF(NTYP1 .NE. 'JOIN')GO TO 125
DO 100 I=NSTA+2,NUM
READ(11,*),XNODE(I),YNODE(I),ZNODE(I),RLOAD(I),RMOMNT(I)
YNODE(I)=YNODE(I)-DY
ZNODE(I)=ZNODE(I)-DZ
WRITE(2,*),XNODE(I),YNODE(I),ZNODE(I)
CONTINUE

READ IN THE RIB LOCATIONS.

DO 200 J=1,NRIB
READ(11,*),XISTA(J)
WRITE(2,*),XISTA(J)
CONTINUE
IF(NTYP1 .NE. 'JOIN')GO TO 250
DO 225 J=NRIB+1,NRT
READ(11,*),XISTA(J)
WRITE(2,*),XISTA(J)
CONTINUE

READ IN THE SECTION PROPERTIES.

DO 300 I=1,NRIB
READ(11,*),SKIN(I)
READ(12,*)(SECT(I,JJ),JJ=1,5)
CONTINUE
IF(NTYP1 .NE. 'JOIN')GO TO 350
DO 325 I=NRIB+1,NRT
READ(11,*),SKIN(I)
READ(12,*)(SECT(I,JJ),JJ=1,5)
BEGIN CONSTRUCTING E.A.L. COMMAND FILE.

WRITE(10,1050)NUM
FORMAT('$RUN DUAI:[PXH]EAL'/
$' 'DUAL[: ]XXX.' /
$ ' ONLINE=C'/
$ ' OUTF=2'/
$ ' XQT TAB'/
$ ' START 'I2,' 6'/
$ ' MATC:1 10.5+6 .3 .1'/
$ ' JLOC')

DO 400 I=1,NUM
WRITE(10,2000) I,XNODE(I),YNODE(I),ZNODE(I)
400 CONTINUE
WRITE(10,2500)
FORMAT('CON' /
$ ' ZERO 1,2,3,4,5 : I')
IF(NTYPI .NE. 'JOIN')GO TO 417
WRITE(10,2510)NUM
FORMAT(' ZERO 1,2,3,4,5 : ',I2)
WRITE(10,2520)
FORMAT(' MREF' /
$ ' 1 1 3 1. 0.' /
$ ' BA')

MOMENT OF INERTIA, AREA, AND TORSIONAL STIFFNESS AT EACH RIB STATION(NSECT) FOR FORWARD WING.

DO 500 I=1,NRIB
SECT(I,3)=SECT(I,3)+SKIN(I)
WRITE(10,3000)(I,(SECT(I,J),J=I,4))
500 CONTINUE
SAME FOR AFT WING.

IF(NTYPI .NE. 'JOIN')GO TO 575
DO 520 I=NRI B+1,NRT
SECT(I,3)=SECT(I,3)+SKIN(I)
WRITE(10,3000)(I,(SECT(I,J),J=I,4))
520 CONTINUE
575 CONTINUE
WRITE(10,3500)
FORMAT(*XQT ELD'/
$ 'E21'/
$ 'NSECT= 1')

NSECTS & CONNECTIVITY FOR FORWARD WING.

NSECT=1
ISTART=1
IFIN=2
WRITE(10,4500)ISTART,IFIN
DO 600 I=2,NSTA
IF (XNODE(I).EQ.XISTA(NSECT)) THEN
NSECT=NSECT+1
WRITE(10,4000)NSECT
FORMAT( ' NSECT=',I2)
ENDIF
ISTART=I
IFIN=I+1
WRITE(10,4500)ISTART,IFIN
FORMAT(1X,2(I3,1X))
600 CONTINUE

NSECTS & CONNECTIVITY FOR AFT WING.
C
IF(NTYP1 .NE. 'JOIN')GO TO 675
NSECT=NSECT+1
WRITE(10,4000)NSECT
WRITE(10,4500)NCOM,NSSTA+2
NSECT=NSECT+1
DO 625 I=NSTA+2,NSTA+NSTA2
IF(XNODE(I) .EQ. XISTA(NSECT)) THEN
WRITE(10,4000)NSECT
NSECT=NSECT+I
ENDIF
ISTART=I
IFIN=I+I
WRITE(10,4500)ISTART,IFIN
CONTINUE
CONTINUE
WRITE(10,5000)
FORMAT( '*XQT E'/
RESET G=386.'/
*XQT TOPO'/
*XQT EKS'/
*XQT K'/
*XQT INV'/
*XQT AUS'/
SYSVEC: APPL FORC')

DO 700 I=2,NSTA
WRITE(10,5500)I,RLOAD(I)
FORMAT(' I=3:J=',I2,':',F9.2)
WRITE(10,5570)I,RMOMNT(I)
FORMAT(' I-4:J=',I2,':',F9.2)
WRITE(2,5575)I,RLOAD(I),RMOMNT(I)
CONTINUE

DO 710 I=NSTA+2,NSTA+NSTA2
WRITE(10,5500)I,RLOAD(I)
WRITE(10,5570)I,RMOMNT(I)
WRITE(2,5578)I,RLOAD(I),RMOMNT(I)
CONTINUE
CLOSE(UNIT=2)
WRITE(10,6000)
FORMAT( '*XQT SSOL'/

A-10
$ '*XQT DCU' '*OUTF=4'
$ ' PRINT 1 STATIC DISPLACEMENTS'
$ '*XQT EXIT')
STOP
END
**B.S.A.P.-BEAM SECTION ANALYSIS PROGRAM**

This program calculates the moments of inertia, torsional constant, and cross-sectional area of each beam section given the skin thicknesses.

---

**VARIABLES**

NSECT = total no of sections on beam.
BB = base dimension of upper & lower skin elements.
ATOT(I) = total skin area of Ith cross-section.
AREA = area enclosed by skin.
XX = x-coordinate of centroid w/respect to ref axis.
YY = y-coordinate of centroid w/respect to ref axis.
RIXX(I) = moment of inertia of Ith cross-section.
RIYY(I) = moment of inertia of Ith cross-section.
RXXY(I) = product of inertia about centroidal axis.
TCNST(I) = torsional constant of Ith cross-section.

DIMENSION XBAR(100,6), YBAR(100,6), RIX(100,6), RIY(100,6)
DIMENSION YPT(100,4), ZPT(100,4)
DIMENSION T(100,6), A(100,6), B(100), H(100), DST(6)
DIMENSION PHI(50), ATOT(50), TCNST(50)
DIMENSION DIX(100,6), DIY(100,6)

OPEN(UNIT=37, NAME='EAPPA.DAT', STATUS='OLD')
READ(37, *) RKAP, SLCON, SLCON1
OPEN(UNIT=1, NAME='SKIN.DAT', STATUS='OLD')
OPEN(UNIT=4, NAME='BCSD1.DAT', STATUS='OLD', ERR=11)
CONTINUE
OPEN(UNIT=3, NAME='OUT2.DAT', STATUS='NEW')
OPEN(UNIT=2, NAME='GEOM.DAT', STATUS='OLD')
OPEN(UNIT=8, NAME='BCSD.DAT', STATUS='OLD')
OPEN(UNIT=9, NAME='INERT.DAT', STATUS='NEW')

READ IN ALL INPUT DATA.

READ(2,*) NSECT
DO 10 I=1, NSECT
READ(8,*) B(I)
H(I) = RKAP * B(I) / 0.65  structural box is 65% of the chord
READ(1,*) (T(I,J), J=1,5)
CONTINUE
REWIND 4
CLOSE(UNIT=8)

BEGIN CALCULATIONS FOR EACH BEAM SECTION.

DO 500 I=1, NSECT
BB = B(I) / 2.0 - T(I,1)
WRITE(3, 1010) I
1010 FORMAT( '/ ', ' PROPERTIES OF BEAM SECTION NO* ', I2, ' ', 1X,$ 33(', '=') )
CALCULATE INDIVIDUAL AREA'S WHICH MAKE UP Ith BEAM SECTION.

\[
\begin{align*}
A(I,1) &= T(I,1) \times H(I) \\
A(I,2) &= T(I,2) \times BB \\
A(I,3) &= T(I,3) \times BB \\
A(I,4) &= A(I,1) \\
A(I,5) &= T(I,4) \times BB \\
A(I,6) &= T(I,5) \times BB
\end{align*}
\]

CALCULATE THE TOTAL CROSS-SECTIONAL AREA.

\[
\begin{align*}
SUM &= 0.0 \\
&DO 25 J = 1, 6 \\
&SUM = SUM + A(I,J) \\
&CONTINUE \\
&ATOT(I) = SUM \\
&WRITE(3,1025) ATOT(I) \\
1025 &FORMAT(/, 2X, ' TOTAL SKIN AREA = ', F15.5)
\end{align*}
\]

LOCATE CENTROIDS OF SKIN AREA'S W/RESPECT TO REFERENCE AXIS

\[
\begin{align*}
XBAR(I,1) &= -(B(I) - T(I,1))/2.0 \\
XBAR(I,2) &= -BB/2.0 \\
XBAR(I,3) &= BB/2.0 \\
XBAR(I,4) &= -XBAR(I,1) \\
XBAR(I,5) &= XBAR(I,3) \\
XBAR(I,6) &= XBAR(I,2)
\end{align*}
\]

LOCATE CENTROIDS OF SKIN AREA'S W/RESPECT TO NEW AXIS

\[
\begin{align*}
XBAR(I,J) &= XBAR(I,J) - XX \\
YBAR(I,J) &= YBAR(I,J) - YY
\end{align*}
\]

FIND THE CENTROID OF THE TOTAL CROSS-SECTION.

\[
\begin{align*}
SUM1 &= 0.0 \\
SUM2 &= 0.0 \\
&DO 50 J = 1, 6 \\
&SUM1 = SUM1 + A(I,J) \times XBAR(I,J) \\
&SUM2 = SUM2 + A(I,J) \times YBAR(I,J) \\
&CONTINUE \\
&XX = SUM1 / ATOT(I) \\
&YY = SUM2 / ATOT(I)
\end{align*}
\]

FIND THE CENTROID OF THE TOTAL CROSS-SECTION.

\[
\begin{align*}
RIX(I,1) &= (T(I,1) \times H(I)^3)/12.0 \\
RIX(I,2) &= (BB \times T(I,2)^3)/12.0 \\
RIX(I,3) &= (BB \times T(I,3)^3)/12.0 \\
RIX(I,4) &= RIX(I,1) \\
RIX(I,5) &= (BB \times T(I,4)^3)/12.0 \\
RIX(I,6) &= (BB \times T(I,5)^3)/12.0
\end{align*}
\]
RIX(I,5)=(BB*T(I,4)**3.0)/12.0
RIX(I,6)=(BB*T(I,5)**3.0)/12.0

RIY(I,1)=(H(I)*T(I,1)**3.0)/12.0
RIY(I,2)=(T(I,2)*BB**3.0)/12.0
RIY(I,3)=(T(I,3)*BB**3.0)/12.0
RIY(I,4)=RIY(I,1)
RIY(I,5)=(T(I,4)*BB**3.0)/12.0
RIY(I,6)=(T(I,5)*BB**3.0)/12.0

USE PARALLEL-AXIS THEOREM TO TRANSFER MOMENTS TO CENTROID.

RIXX=0.0
RIYY=0.0
DO 100 J=1,6
RIX(I,J)=RIX(I,J)+A(I,J)*(YBAR(I,J)**2.0)
RIY(I,J)=RIY(I,J)+A(I,J)*(XBAR(I,J)**2.0)
RIXX=RIXX+RIX(I,J)
RIYY=RIYY+RIY(I,J)
100 CONTINUE
WRITE(3,1050)RIXX
1050 FORMAT(2X,'IXX = ',F15.5)
WRITE(3,1051)RIYY
1051 FORMAT(2X,'IYY = ',F15.5)

CALCULATE THE PRODUCT OF INERTIA ABOUT CENTROIDAL AXIS.

RIXY=0.0
DO 105 J=1,6
RIXY=RIXY+A(I,J)*XBAR(I,J)*YBAR(I,J)
105 CONTINUE
WRITE(3,1070)RIXY
1070 FORMAT(2X,'IXY = ',F15.5)
RIXX=RIXX*SLCON
RIYY=RIYY*SLCON
WRITE(9,*),RIXX,RIYY,RIXY

CALCULATE THE POLAR MOMENT OF INERTIA ABOUT CENTROID.

RIZZ=RIXX+RIYY
WRITE(3,1074)RIZZ
1074 FORMAT(2X,'IZZ = ',F15.5)

CALCULATE TORSIONAL CONSTANT FOR ITH SECTION.

AREA=B(I)*H(I)-ATOT(I)
DST(1)=H(I)/T(I,1)
DST(2)=BB/T(I,2)
DST(3)=BB/T(I,3)
DST(4)=DST(1)
DST(5)=BB/T(I,4)
DST(6)=BB/T(I,5)
SUM=0.0
DO 170 J=1,6
SUM=SUM+DST(J)
170 CONTINUE
TCNST(I)=4.0*(AREA**2.0)/SUM
WRITE(3,1075)TCNST(I)
1075 FORMAT(2X,'TORSIONAL CONSTANT = ',F15.5)
DETERMINE ANGLE OF ROTATION OF PRINCIPAL AXIS.

\[
\phi(I) = \left( \frac{180.0}{\pi} \right) \times 0.5 \times \arctan \left( \frac{2.0 \times R_{xy}}{(R_{xx} - R_{yy}) / 2.0} \right)
\]

WRITE (3, 1078) PHI(I)

\[
\text{FORMAT}(2X, ' PHI = ', F15.5, ' DEGREES')
\]

MAXIMUM & MINIMUM MOMENTS OF INERTIA ABOUT PRINCIPAL AXIS.

\[
R_{\text{max}} = R_{zz} / 2.0 + \sqrt{(R_{xy})^2 + \left(\frac{(R_{xx} - R_{yy})}{2.0}\right)^2}
\]

WRITE (3, 1080) RIMAX

\[
\text{FORMAT}(2X, ' I1(\text{MAX}) = ', F15.5)
\]

\[
R_{\text{min}} = R_{zz} / 2.0 - \sqrt{(R_{xy})^2 + \left(\frac{(R_{xx} - R_{yy})}{2.0}\right)^2}
\]

WRITE (3, 1081) RIMIN

\[
\text{FORMAT}(2X, ' I2(\text{MIN}) = ', F15.5)
\]

CALCULATE THE DERIVATIVES OF THE TOTAL MOMENT OF INERTIA ABOUT X\text{\_}/Y AXIS' W/RESPECT TO THE SKIN THICKNESSES.

\[
D_{x1}(I, 1) = H(I)^3 / 6.0 + 2.0 \times R_{xy} \times H(I)
\]

\[
D_{x2}(I, 1) = B \times (T(I, 2)^2 / 4.0 + B \times (R_{xx}^2)
\]

\[
D_{x3}(I, 1) = B \times (T(I, 3)^2 / 4.0 + B \times (R_{xx}^2)
\]

\[
D_{x4}(I, 1) = B \times (T(I, 4)^2 / 4.0 + B \times (R_{xx}^2)
\]

\[
D_{x5}(I, 1) = B \times (T(I, 5)^2 / 4.0 + B \times (R_{xx}^2)
\]

\[
D_{y1}(I, 1) = H(I) \times (T(I, 1)^2 / 2.0 + 2.0 \times H(I) \times (R_{xx}^2)
\]

\[
D_{y2}(I, 1) = (B \times 3 / 12.0 + B \times (R_{xx}^2)
\]

\[
D_{y3}(I, 1) = D_{y2}(I, 1)
\]

\[
D_{y4}(I, 1) = D_{y3}(I, 1)
\]

\[
D_{y5}(I, 1) = D_{y4}(I, 1)
\]

XB = ABS(XBAR(I, 1))

\[
\phi(I) = \left( \frac{\pi}{180.0} \right) \times \phi(I)
\]

SP = SIN(\phi(I))

CP = COS(\phi(I))

Y2 = ABS(YBAR(I, 2))

Y3 = ABS(YBAR(I, 3))

Y5 = ABS(YBAR(I, 5))

Y6 = ABS(YBAR(I, 6))

\[
Y_{pt1}(I, 1) = -XB \times CP + Y2 \times SP
\]

\[
Y_{pt2}(I, 2) = -XB \times SP + Y3 \times CP
\]

\[
Y_{pt3}(I, 3) = -XB \times SP - Y5 \times CP
\]

\[
Y_{pt4}(I, 4) = -XB \times CP - Y6 \times SP
\]

\[
Z_{pt1}(I, 1) = +XB \times SP + Y2 \times CP
\]

\[
Z_{pt2}(I, 2) = +XB \times SP + Y3 \times CP
\]

\[
Z_{pt3}(I, 3) = +XB \times SP - Y5 \times CP
\]

\[
Z_{pt4}(I, 4) = +XB \times SP - Y6 \times CP
\]

WRITE (4, *) I, RIMAX, RIMIN, ATOT(I), TCNST(I), PHI(I)

CONTINUE

DO 501 I = 1, NSECT

WRITE (9, *) ATOT(I), TCNST(I)

WRITE (4, *) B(I)

CONTINUE

WRITE (4, *) ((YPT(I, J), I = 1, NSECT), J = 1, 4)

WRITE (4, *) ((ZPT(I, J), I = 1, NSECT), J = 1, 4)

CLOSE (UNIT = 3)

CLOSE (UNIT = 4)

CLOSE (UNIT = 9)

STOP

END

A-15
**FINITE ELEMENT GENERATOR FOR WING WITH GIVEN NUMBER OF STATIONS AND STRINGERS**

DIMENSION XCOR(500), YCOR(500), ZCOR(500)
DIMENSION CHORD(100), RLOAD(100,50), ISTA(50), ISTA2(50)
DIMENSION AREA(50,50), THICK(50,50)
DIMENSION BLOAD(50), XISTA(50), RMOMNT(50)
DIMENSION SKIN(50), VX(50)

OPEN(UNIT=13, NAME='IGO.DAT', STATUS='OLD')
OPEN(UNIT=15, NAME='WING.DAT', STATUS='OLD')

READ(13,*) IGO
READ(15,5) NTYP1, NTYP2
WRITE(6,10) NTYP1, NTYP2

READ IN DATA FOR FORWARD WING.
READ(15,*) B, CR, CT, NSTR, NSTA, NRIB, RLAM, GAM, TR, RR

IF (NTYP1 .NE. 'JOIN') THEN
    WRITE(6,20) B, CR, CT, NSTR, NSTA, NRIB, RLAM, GAM, TR
    GO TO 50
ELSE
    WRITE(6,35)
    FORMAT(/,1X, 'FORWARD WING'/1X, 12('=',CHAR(32)))
    WRITE(6,20) B, CR, CT, NSTR, NSTA, NRIB, RLAM, GAM, TR
END IF

READ IN DATA FOR AFT WING.
READ(15,*) B2, CR2, CT2, NSTR2, NSTA2, NRIB2, RLAM2, GAM2, TR2, RR

WRITE(6,45)

WRITE(6,20) B2, CR2, CT2, NSTR2, NSTA2, NRIB2, RLAM2, GAM2, TR2

**GENERATE COORDINATES FOR FORWARD WING**

CR = .65*CR
CT = .65*CT
IEND = NSTR*(NSTA+1)
IREND = IEND
NSTP1 = NSTA - 1
RLAM = RLAM * 3.14159 / 180.
GAM  = GAM  * 3.14159 / 180.

C
C DENOTE THE ORIGINATION OF THE 1/4 CHORD LINE
C AS THE ORIGIN OF THE CARTESIAN COORDINATE SYSTEM
C
C DETERMINE COORDINATES OF TERMINUS OF 1/4 CHORD LINE
C
XEND = B
YEND = -B * TAN(RLAM)

C
C DETERMINE COORDINATES OF BOX VERTICES AT ROOT AND TIP
C
XRLE = 0.
XRTE = 0.
YRLE = 0.25 * CR
YRTE = -0.75 * CR
XTLE = B
XTTE = B
YTLE = YEND + 0.25 * CT
YTTE = YEND - 0.75 * CT

C
C OBTAIN SLOPES OF LE AND TE FOR Z = CONST PLANE
C
SLOPLE = (YTLE - YRLE) / B
SLOPTE = (YTTE - YRTE) / B

C
C GENERATE X-COORDINATES OF ALL NODES FOR FORWARD WING
C
IF (NTYP1 .NE. 'JOIN') THEN
  NSTA2 = NSTA
  B2 = B
END IF
DO 85 I = 1, NSTR
  XCOR(I) = 0.0
  JEND = NSTR * (NSTA2 + 1)
DO 100 I = NSTR + 1, JEND
  XCOR(I) = ((I - 1) / NSTR) * B2 / NSTA2
CONTINUE
85
DO 100 I = 1, NSTR
  XCOR(I) = (K / NSTR) * (B - B2) / (NSTA - NSTA2) + XCOR(JEND)
100 CONTINUE
IF (NSTA .EQ. NSTA2) GO TO 102
DO 101 I = JEND + 1, IEND
  K = (I + NSTR) - (JEND + 1)
  XCOR(I) = (K / NSTR) * (B - B2) / (NSTA - NSTA2) + XCOR(JEND)
101 CONTINUE
102 CONTINUE

C
C Y-COORDINATES OF TE AND LE'S
C
II = (NSTR / 2) + 1
K = 1 - NSTR
KK = II - NSTR
DO 150 I = 1, NSTP1
  K = K + NSTR
  KK = KK + NSTR
  K1 = K + 1
  KK1 = KK + 1
  YCOR(K) = YRTE + SLOPTE * XCOR(K)
  YCOR(KK) = YRLE + SLOPLE * XCOR(KK)
  YCOR(K1) = YCOR(K)
  YCOR(KK1) = YCOR(KK)

A-17
CONTINUE

DETERMINE CHORD LENGTH AT EACH STATION

I1=1-NSTR
I2=(NSTR/2)+2-NSTR
DO 200 I=1,NSTPI
I1=I1+NSTR
I2=I2+NSTR
CHORD(I)=ABS(YCOR(I2)-YCOR(I1))
CONTINUE

Y-COORDINATES OF ANY INTERMEDIATE POINTS

IF(NSTR.LE.4)GO TO 275
I1=(NSTR/2)+1-NSTR
I2=(NSTR/2)+2-NSTR
NCRD=(NSTR/2)-1
DO 250 I=1,NSTPI
I1=I1+NSTR
I2=I2+NSTR
DO 260 J=I,NCRD-I
YCOR(I1-J)=YCOR(I1)-CHORD(I).*(1./NCRD)*FLOAT(J)
YCOR(I2+J)=YCOR(I1-J)
CONTINUE

Z-COORDINATES OF NODE POINTS
ASSUME THAT THE DIHEDRAL ANGLE IS DEFINED
BETWEEN THE CENTER LINE AND THE HORIZONTAL PLANE

ZRCL=0.5*CR*TR/.65
ZTCL=ZRCL+B*TAN(GAM)

GENERATE CORNER COORDINATES FIRST

RR IS INPUT FROM WING.DAT, BUT HAS NO MEANING IF ONLY 4 STRINGERS
IF(NSTR.EQ.4)RR=1.0
ZRLSC AND ZRLSC ARE LOWER SURFACE Z-COR OF CORNERS

ZRLSC=ZRCL-RR*0.5*TR*CHORD(1)/.65
ZTLC=ZTCL-RR*0.5*TR*CHORD(NSTPI)/(.65)
SLPLS=(ZTLC-ZRLSC)/B

Z-COORDINATES OF TE AND LE

II=(NSTR/2)+2
E=1-NSTR
KK=II-NSTR
DO 280 I=1,NSTPI
E=E+NSTR
KK=KK+NSTR
ZCOR(K)=ZRLSC+SLPLS*XCOR(K)
ZCOR(KK)=ZCOR(K)
ZCOR(K+1)=ZCOR(K)+RR*TR*CHORD(I)/.65
ZCOR(KK-1)=ZCOR(K+1)
CONTINUE
C C TERMINATE IF ONLY FOUR STRINGERS
C
IF (NSTR .EQ. 4) GO TO 295
I1 = (NSTR/2+1)-NSTR
I2 = (NSTR/2+2)-NSTR
DO 290 I = I1, NSTP1
I1 = I1 + NSTR
I2 = I2 + NSTR
DO 285 J = 1, NCRD-1
ZCOR (I1-J) = ZCOR (I1) + TR*CHORD (I)/.65*(1.-RR)*0.5
ZCOR (I2+J) = ZCOR (I2) - TR*CHORD (I)/.65*(1.-RR)*0.5
285 CONTINUE
290 CONTINUE
295 CONTINUE
IF (NTYPI .NE. 'JOIN') GO TO 310

C***********************************************************
C*   GENERATE COORDINATES FOR AFT WING  *
C***********************************************************

CR2 = .65*CR2
CT2 = .65*CT2
IEND2 = NSTR2*(NSTA2+1)
IREND = IEND + IEND2
RLAM2 = RLAM2*3.14159/180.
GAM2 = GAM2*3.14159/180.

C GENERATE X-COORDINATES FOR REAR WING

IRTIP = IEND + NSTR2
DO 303 I = IEND+1, IRTIP
XCOR (I) = XCOR (JEND)
DO 304 I = IRTIP+1, IREN
D = I-IEND-1
XCOR (I) = XCOR (IRTIP) - (D/NSTR2)*B2/NSTA2
304 CONTINUE

C DETERMINE COORDINATES OF TERMINUS OF 1/4 CHORD LINE
C ON AFT WING.
YENDT = YCOR (JEND-NSTR+1) - 0.07*CT2/.65-.25*CT2
YENDR = YENDT - B2*TAN (RLAM2)

C DETERMINE COORDINATES OF BOX VERTICES AT ROOT & TIP
C FOR AFT WING.
YTLE = YENDT + 0.25*CT2
YTTE = YENDT - 0.75*CT2
YRLE = YENDR + 0.25*CR2
YRTE = YENDR - 0.75*CR2

C OBTAIN SLOPES OF LE AND TE FOR AFT WING
SLOPLE = (YRLE-YTLE)/B2
SLOPTE = (YRTE-YTTE)/B2

C GENERATE Y-COORDINATES OF LE AND TE
II = (NSTR2/2)+1+IEND
K = 1-NSTR2+IEND
KK = II-NSTR2
DO 305 I=1,NSTA+1
K = K+NSTR2
KK = KK+NSTR2
K1 = K+1
KK1 = KK+1
YCOR(K) = YTTE+(B2/NSTA2)*(I-1)*SLOPTE
YCOR(KK) = YTTE+(B2/NSTA2)*(I-1)*SLOPLE
YCOR(K1) = YCOR(K)
YCOR(KK1) = YCOR(KK)
CONTINUE

Determine chord length at each station

II = 1-NSTR2+IEND
I2 = (NSTR2/2)+2-NSTR2+IEND
NSTP2 = NSTA+2+NSTA
DO 306 I=NSTA+2,NSTP2
II = II+NSTR2
I2 = I2+NSTR2
CHORD(I) = ABS(YCOR(I2)-YCOR(II))
CONTINUE

Generate y-coordinates at intermediate points

IF (NSTR2 .LE. 4) GO TO 307
II = (NSTR2/2)+1-NSTR2+IEND
I2 = (NSTR2/2)+2-NSTR2+IEND
NCRD = (NSTR2/2)-1
DO 307 I=NSTA+2,NSTP2
II = II+NSTR2
I2 = I2+NSTR2
DO 308 J=I,NCRD-I
YCOR(II-J) = YCOR(II)-CHORD(I)*(1./NCRD)*FLOAT(J)
YCOR(I2+J) = YCOR(I2-J)
CONTINUE
CONTINUE

Generate z-coordinates of aft wing

ZTCL = (ZCOR(JEND-NSTR+2)+ZCOR(JEND-NSTR+1))/2.0
ZRCL = ZTCL+B2*TAN(GAM2)

Generate corner coordinates first

RR IS INPUT FROM WING.DAT, BUT HAS NO MEANING IF ONLY 4 STRINGERS
IF (NSTR2 .EQ. 4) RR=1.0
ZRLSC AND ZTLSC ARE LOWER SURFACE Z-COR OF CORNERS
ZTLSC = ZTCL-RR*0.5*TR2*CHORD(NSTA+2)/.65
ZRLSC = ZRCL-RR*0.5*TR2*CHORD(NSTP2)/.65
SLPLS = (ZRLSC-ZTLSC)/B2

Generate z-coordinates of te and le

II = (NSTR2/2)+2+IEND
K = 1-NSTR2+IEND
KK = II-NSTR2
DO 309 I=1,NSTA2+1
K = K+NSTR2
KK = KK+NSTR2
ZCOR(K) = ZTLSC+SLPLS*(B2/NSTA2)*(I-1)
ZCOR(KK) = ZCOR(K)
ZCOR(K+1) = ZCOR(K)+RR*TR2*CHORD(I+NSTA+1)/.65
ZCOR(KK-1) = ZCOR(K+1)
309 CONTINUE
C
C BYPASS IF ONLY FOUR STRINGERS
C
IF (NSTR2 .EQ. 4) GO TO 310
II = (NSTR2/2+1)-NSTR2+IEND
I2 = (NSTR2/2+2)-NSTR2+IEND
NCRD = (NSTR2/2)-1
DO 310 I=1,NSTA2+1
II=II+NSTR2
I2=I2+NSTR2
DO 311 J=I,NCRD-1
ZCOR(II-J)=ZCOR(II)+TR2*CHORD(I+NSTP1)/0.65*(I.-RR)*0.5
ZCOR(I2+J)=ZCOR(I2)-TR2*CHORD(I+NSTP1)/0.65*(1.-RR)*0.5
311 CONTINUE
310 CONTINUE
C
C PRINT OUT NODAL COORDINATES OF WING
C
C***********************************************************************
C***********************************************************************
C***********************************************************************
C***********************************************************************
C***********************************************************************
G
OPEN(UNIT=9,NAME='WING.COM',STATUS='NEW')
WRITE(9,2220)IREND
2220 FORMAT('$RUN DUAL:[PXH]EAL'/'$''DUAL1:[]XXX.'/'$ '*OUTF-1'/'$ '*ONLINE=0'/'$ '*XQT TAB'/'$ 'START ',I3,',4 5 6'/'$ 'MATC'/' 1 10.5+6 .3 .I'/'$ ' 2 10.5+6 .3 .I'/'$ ' JLOC'')
C
C WRITE IN JOINT LOCATIONS FOR ALL NODES.
C
DO 500 I=1,IREND
WRITE(9,2500)I,XCOR(I),YCOR(I),ZCOR(I)
2500 FORMAT(1X,I3,1X,3(F9.2,1X))
500 CONTINUE
C
C WRITE CONSTRAINT DEFINITION(S) [TWO IF JOINED WING]
C
WRITE(9,2550)NSTR
2550 FORMAT(' CON 1'' ZERO 1,2,3:1.,12)
IF(NTYP1 .NE. 'JOIN')GO TO 502
I = IREND-(NSTR2-1)
WRITE(9,2555)I,IREND

A-21
CONNECT FORWARD & AFT WINGS.

IF(NTYP1 .EQ. 'JOIN') THEN
WRITE(9,1118)
1118 FORMAT(' MREF'/1 1 1 I. I.')
ENDIF

CALCULATE TOTAL NO# RIB STATIONS.

IF(NTYP1 .NE. 'JOIN')NRIB2=0
NRT = NRIB + NRIB2
DO 503 I=1,NRIB
DO 504 J=1,NSTR
504 AREA(I,J)=0.5
503 CONTINUE
IF(NTYP1.EQ.'JOIN') THEN
DO 505 I=NRIB+I,NRT
DO 506 J=1,NSTR2
506 AREA(I,J)=0.5
505 CONTINUE
ENDIF

IF(IGO.GT.1) THEN
OPEN(UNIT=4,NAME='THICK.DAT',STATUS='OLD')
ENDIF

INITIALIZE RIB/SKIN THICKNESS' & STRINGER AREA'S FOR 1ST RUN ONLY.

IF(IGO .GT. 1) GO TO 530
DO 511 I=1,NRT
THICK(I,1) = 0.05
511 CONTINUE
DO 515 I=1,NRIB
DO 514 J=1,NSTR
THICK(I+I,J) = 0.05
514 CONTINUE
515 CONTINUE
WEIGHT=t0.0
IF(NTYP1 .NE. 'JOIN') GO TO 530
DO 525 I=NRIB+I,NRT
DO 520 J=1,NSTR2
THICK(I+I,J) = 0.05
520 CONTINUE
525 CONTINUE

READ IN RIB/SKIN THICKNESS' & STRINGER AREA'S [IGO.1]

IF(IGO .EQ. 1) GO TO 590
READ(4,*)(THICK(1,JK),JK=1,NRT)
DO 550 I=1,NRIB
READ(4,*)(THICK(I+1,JK),JK=1,NSTR)
550 CONTINUE
IF(NTYP1 .NE. 'JOIN') GO TO 580
DO 575 I=NRIB+1,NRT
READ(4,*)(THICK(I+1,JK),JK=1,NSTR2)
575 CONTINUE
READ(4,*)WGT1,WGT2
CLOSE(UNIT=4,DISP='DELETE')
CONTINUE
OPEN(UNIT=4,NAME='THICK.DAT',STATUS='NEW')
WRITE(4,*)(THICK(1,JP),JP=1,NRT)
DO 507 IJJ=1,NRIB
WRITE(4,*)(THICK(IJJ+I,K),K=1,NSTR)
IF(NTYP1.EQ.'JOIN')THEN
  DO 508 IJJ=NRI+I,NRT
  WRITE(4,*)(THICK(IJJ+I,K),K=1,NSTR2)
  ENDIF
WRITE(4,*)WGT1,WGT2
CLOSE(UNIT=4)
CONTINUE

WRITE IN E23 SECTION PROPERTIES.
WRITE(9,*)' BC'
ISUB = 1
DO 610 I=1,NRIB
  DO 600 J=1,NSTR
    WRITE(9,2600)ISUB,AREA(I,J)
    ISUB = ISUB + 1
  CONTINUE
600 CONTINUE
610 CONTINUE
2600 FORMAT(1X,I3,1X,F6.2)
IF(NTYP1 .NE. 'JOIN')GO TO 650
DO 630 I=NRI+1,NRT
  DO 620 J=1,NSTR2
    WRITE(9,2600)ISUB,AREA(I,J)
    ISUB = ISUB + 1
  CONTINUE
620 CONTINUE
630 CONTINUE

BEAM ELEMENT GRIDWORK TO JOIN FRONT AND REAR WINGS
WRITE(9,*)' BA'
WRITE(9,1117)
1117 FORMAT(' GIVN 1 10000. 0. 10000. 0. 10. 10000. ') rigid beam
 CONTINUE

WRITE IN SHELL SECTION PROPERTIES.
WRITE(9,*)' SA'
DO 675 I=1,NRT
  WRITE(9,2620)I,THICK(I,I)
CONTINUE
2620 FORMAT(1X,I3,1X,F7.4)
ISUB = NRT + 1
DO 685 I=NRI+1,NRT
  DO 680 J=1,NSTR
    WRITE(9,2620)ISUB,THICK(I+I,J)
    ISUB = ISUB + 1
  CONTINUE
680 CONTINUE
685 CONTINUE
IF(NTYP1 .NE. 'JOIN')GO TO 700
DO 695 I=NRI+1,NRT
  DO 690 J=1,NSTR2
    WRITE(9,2620)ISUB,THICK(I+I,J)
    ISUB = ISUB + 1
  CONTINUE
690 CONTINUE
CONTINUE

WRITE(9,2573)

FORMA(*XQT ELD*)
WRITE(9,*),'E41'
WRITE(9,*),'NMAT=2'

LOCATE NODES OF AFT QUADRANT ON 1ST RIB STATION.

II1=NSTR+1
II2=NSTR*2
II3=NSTR+3
II4=II1+1

GENERATE RIBS FOR FORWARD WING.

K=0
NINC=NSTA/NRIB
DO 701 I=1,NSTA,NINC
K=K+1
NSECTION=N
WRITE(9,4001)NSECTION
FORMAT(' NSECTION=',I2)
WRITE(9,4025)II1,II2,II3,II4
FORMAT(IX,4(13.1X))
ISTRA(K)=II1
M1=NSTR/2-2
IJ1=II2
IJ2=IJ1-1
IJ3=IJ2+1
IJ4=IJ3
DO 705 J=1,M1
WRITE(9,4025)IJ1,IJ2,IJ3,IJ4
IJ1=IJ1-1
IJ2=IJ1-1
IJ3=IJ2-1
IJ4=IJ3-1
CONTINUE

INCREMENT TO NEXT RIB STATION.

II1=II1+(NSTR*NINC)
II2=II2+(NSTR*NINC)
II3=II3+(NSTR*NINC)
II4=II4+(NSTR*NINC)
IF(K.EQ.NRIB)GO TO 707
CONTINUE

IF(NTYP1 .NE. 'JOIN')GO TO 712

LOCATE NODES OF AFT QUADRANT ON TIP RIB STATION.

II1=IEND+1
II2=II1+NSTR2-1
II3=II1+2
II4=II1+1

GENERATE RIBS FOR AFT WING.

K=0
NINC=1
DO 709 I=1,NSTA2,NINC
IF(I.GT.1)NINC=NSTA2/NRIB2
K=K+1
NSECT=K+NRIB
WRITE(9,4026)NSECT
4026 FORMAT(’NSECT=’,I2)
WRITE(9,4027)II1,II2,II3,II4
4027 FORMAT(1X,4(I3,1X))
ISTA2(K)=II1
M1=NSTR2/2-2
IJ1=II2
IJ2=IJ1-1
IJ3=II3+1
IJ4=II3
DO 710 J=1,M1
WRITE(9,4027)IJ1,IJ2,IJ3,IJ4
IJ1=IJ1-1
IJ2=IJ1-1
IJ3=IJ2-1
IJ4=IJ3-1
710 CONTINUE
C
C INCREMENT TO NEXT RIB STATION.
C
II1=II1+(NSTR2*NINC)
II2=II2+(NSTR2*NINC)
II3=II3+(NSTR2*NINC)
II4=II4+(NSTR2*NINC)
IF(K.EQ.NRT)GO TO 712
709 CONTINUE
712 CONTINUE
C
C GENERATE SKIN MEMBRANE FOR FORWARD WING.
C
C LOCATE NODES ON BOTTOM AFT SKIN PANEL [BOTTOM PANEL IF NSTR=4]
C
J1=1
J2=NSTR+1
J3=2*NSTR
J4=NSTR
C
NSECT=NSECT+1
DO 714 J=1,NSTR
WRITE(9,3041)NSECT,J1,J2,J3,J4
3041 FORMAT(1X,’NSECT=’,I2,/,1X,4(I3,1X))
NSECT=NSECT+1
J1=J
J2=NSTR+J
J3=J2+1
J4=J1+1
714 CONTINUE
C
C COMPLETE THE REST OF THE WING.
C
3050 FORMAT(1X,4(I3,1X))
K=1
KEND=NRIB+1
C
C-25
IF(NRIB .EQ. NSTA) KEND = NRIB
DO 720 I = 2, KEND
  J1 = ISTA(K)
  J2 = ISTA(K) + NSTR
  J3 = J2 + (NSTR - 1)
  J4 = J3 - NSTR
WRITE(9, 3040) NSECT
3040 FORMAT( ' NSECT=', I3)
  K = K + 1
NSECT = NSECT + 1
IF(I .EQ. NRIB) ISTA(K + 1) = IEND - (NSTR - 1)
ITEMP = ISTA(K) - 1
  DO 719 II = ISTA(K - 1), ITEMP, NSTR
    WRITE(9, 3050) J1, J2, J3, J4
      J1 = J1 + NSTR
      J2 = J2 + NSTR
      J3 = J3 + NSTR
      J4 = J4 + NSTR
719    CONTINUE
DO 715 J = 1, NSTR - 1
  I1 = ISTA(K - 1) + J - 1
  I2 = I1 + NSTR
  I3 = I2 + 1
  I4 = I1 + 1
WRITE(9, 3040) NSECT
NSECT = NSECT + 1
  DO 717 KK = ISTA(K - I), ITEMP, NSTR2
    WRITE(9, 3050) I1, I2, I3, I4
      I1 = I1 + NSTR
      I2 = I2 + NSTR
      I3 = I3 + NSTR
      I4 = I4 + NSTR
717    CONTINUE
715    CONTINUE
720    CONTINUE
C
C GENERATE SKIN MEMBRANE FOR AFT WING.
C
C
IF(NTYP1 .NE. 'JOIN') GO TO 729
3054 FORMAT(1X, 4(I3, 1X))
   K = 1
KEND = NRIB2 - 1
DO 728 I = 1, KEND
  J1 = ISTA2(K)
  J2 = ISTA2(K) + NSTR2
  J3 = J2 + (NSTR2 - 1)
  J4 = J3 - NSTR2
WRITE(9, 3053) NSECT
3053 FORMAT( ' NSECT=', I3)
   K = K + 1
NSECT = NSECT + 1
ITEMP = ISTA2(K) - 1
  DO 725 J = ISTA2(K - 1), ITEMP, NSTR2
    WRITE(9, 3054) J1, J2, J3, J4
      J1 = J1 + NSTR2
      J2 = J2 + NSTR2
      J3 = J3 + NSTR2
      J4 = J4 + NSTR2
725    CONTINUE
728    CONTINUE
729    CONTINUE
CONTINUE
DO 726 J=1,NSTR2-1
I1=ISTA2(K-1)+J-1
I2=I1+NSTR2
I3=I2+1
I4=I1+1
WRITE(9,3053)NSECT
NSECT=NSECT+1
DO 727 II=ISTA2(K-1),ITEMP,NSTR2
WRITE(9,3054)II,I2,I3,I4
I1=I1+NSTR2
I2=I2+NSTR2
I3=I3+NSTR2
I4=I4+NSTR2
CONTINUE
727 CONTINUE
726 CONTINUE
728 CONTINUE
C
C GENERATE SKIN PANELS FOR ROOT SECTION.
C
J1=IREND-(2*NSTR2-1)
J2=J1+NSTR2
J3=J2+(NSTR2-1)
J4=J3-NSTR2
C
I1=J1
DO 730 J=I1,I1+NSTR2-1
WRITE(9,3018)NSECT,J1,J2,J3,J4
FORMAT(' NSECT=',I2,/,2(IX,13))
CONTINUE
730 CONTINUE
729 CONTINUE
C
C GENERATE BAR ELEMENTS FOR FORWARD WING.
C
WRITE(9,2575)
FORMAT(' E23'/IX,' NMAT=I')
NSEC=NSTR
DO 731 JJ=1,NSTR
J1=JJ
J2=JJ+NSTR
WRITE(9,3019)JJ,J1,J2
FORMAT(' NSECT=',I2,/,2(IX,13))
CONTINUE
731 CONTINUE
KEND=NRIB+1
IF(NRIB.EQ.NSTA)KEND=NRIB
DO 734 K=2,KEND
IF(K.EQ.NRIB)ISTA(K+I)=IEND-(NSTR-I)
DO 733 J=1,NSTR
DO 732 I=ISTA(K-1),ISTA(K)-1,NSTR
IF(I.EQ.ISTA(K-1))THEN
NSEC=NSEC+1
WRITE(9,3022)NSECT
FORMAT(' NSECT=',I2)
GENERATE BAR ELEMENTS FOR AFT WING.

IF(NTYP1 .NE. 'JOIN')GO TO 750
KEND2=NRIB2+1
IF(NRIB2 .EQ. NSTA2)KEND2=NRIB2
DO 745 K=I,KEND2-1
IF(K .EQ. NRIB2)ISTA2(K+I)=IREND-(NSTR2-1)
DO 742 J=1,NSTR2
   DO 740 I=ISTA2(K),ISTA2(K+I)-I,NSTR2
      IF(I .EQ. ISTA2(K))THEN
         NSEC=NSEC+1
         WRITE(9,3030)NSEC
         FORMAT('NSEC=',I3)
      END IF
   ISTART=I+6+(K-1)
   ITERM=ISTART+NSTR2
   WRITE(9,3031)ISTART,ITERM
   FORMAT(IX,2(1X))
   CONTINUE
CONTINUE
CONTINUE
II=IREND-2+NSTR2+1
I2=II+NSTR2-1
DO 746 J=II,I2
   I1=J
   J2=J+NSTR2
   NSEC=NSEC+1
   WRITE(9,3030)NSEC
   WRITE(9,3031)J1,J2
CONTINUE
IF(NTYP1.EQ. 'JOIN')THEN
   WRITE(9,1119)
   FORMAT(' E21 ',/,' NSECT=',I3)
   J1=ISTA2(1)+NSTR2/2
   J2=J1+1
   J3=JEND-NSTR+1
   J4=J3+1
   WRITE(9,3025)J1,J4
   WRITE(9,3025)J2,J3
   WRITE(9,3025)J1,J3
   WRITE(9,3025)J2,J4
   WRITE(9,3025)J1,J2
   WRITE(9,3025)J3,J4
ENDIF
WRITE(9,4050)
FORMAT('*XQT E'/
 $ ' 'RESET G=386.'/
 $ ' *XQT TOPO'/
 $ ' *XQT EKS'/
 $ ' *XQT K' 'RESET SPDP=2' ' *XQT INV' ' *XQT AUS'/
 $ ' SYSVEC:APPL FORC')
APPLY SPANWISE LOAD DISTRIBUTION TO FORWARD WING.

RLOAD(1,1)=0.0
RLOAD(2,1)=0.0
READ(15,*)WT
W1=WT/2.0
IF(NTYP1 .EQ. 'JOIN')W1=0.7*W1
PI=3.1416
Q=(2.0*W1)/(PI*B**2)
VX(1)=0.0
NODE=NSTR+2
DO 760 I=1,NSTA
VX(I+1)=Q*(XCOR(NODE)*SQRT(B**2-XCOR(NODE)**2)
$ +B**2)*ASIN(XCOR(NODE)/B))
W=VX(I+1)-VX(I)
   DO 755 J=1,2
      CC=0.65
      IF(J .EQ. I)CC=0.35
      RLOAD(I+1,J)=W*CC
      WRITE(9,5000)NODE,RLOAD(I+1,J)
      FORMAT('I=3: J=',I3,':',F7.2)
      NODE=NODE+(NSTR/2-1)
   CONTINUE
  NODE=NODE+2
  CONTINUE
FACT1=RLOAD(NSTA+1,1)/(NSTA-1)
FACT2=RLOAD(NSTA+1,2)/(NSTA-1)
DO 765 I=2,NSTA
   RLOAD(I,1)=RLOAD(I,1)+FACT1
   RLOAD(I,2)=RLOAD(I,2)+FACT2
CONTINUE
RLOAD(NSTA+1,1)=0.0
RLOAD(NSTA+1,2)=0.0
DO 770 I=1,NSTA+1
CONTINUE

APPLY SPANWISE LOAD DISTRIBUTION TO AFT WING.

IF(NTYP1 .NE. 'JOIN')GO TO 780
N=NSTA+NSTA2+2
RLOAD(N,1)=0.0
RLOAD(N,2)=0.0
W2=0.3*WT/2.0
Q=(2.0*W2)/(PI*B2**2)
VX(1)=0.0
NODE=IREND-2*(NSTR2-1)
K=NSTA+NSTA2+1
DO 772 I=1,NSTA2
   VX(I+1)=Q*(XCOR(NODE)*SQRT(B2**2-XCOR(NODE)**2)
$ +(B2**2)*ASIN(XCOR(NODE)/B2))
   W=VX(I+1)-VX(I)
   DO 771 J=1,2
      CC=0.65
      IF(J .EQ. I)CC=0.35
      RLOAD(K,J)=W*CC
      WRITE(9,5000)NODE,RLOAD(K,J)
      NODE=NODE+(NSTR2/2-1)
   CONTINUE
  NODE=NODE-2*(NSTR2-1)

application of loads on the front and aft wings.
replace segment by read statement for aerodynamic loads generated in an external program.
CALL UP STRESSES AND STATIC DISPLACEMENTS

WRITE(9,5020)
5020 FORMAT('XQT SSOL'/
       "$'OUTF=2'$/
       "$'XQT ES'$/
       "$'E41'$/
       "$'OUTF=3'$/
       "$'XQT DCU'$/
       "$'PRINT 1 STAT DISP'$/
       "$'XQT EXIT'$/
       CLOSE(UNIT=9)

DATA FOR FORWARD WING.

OPEN(UNIT=2,NAME='INPUT.DAT',STATUS='NEW')
WRITE(2,5031)NTYP1,NTYP2
5031 FORMAT(A4,A2)
WRITE(2,5032)NRIB
5032 FORMAT(1X,I2)
ISTRT=1
DO 792 J=1,NRIB
WRITE(2,5032)NSTR
J1=0
IF(J.GT.1)ISTRT=ISTA(J-1)
IFIN=ISTRT+NSTR-1
DO 790 I=ISTRT,IFIN
J1=J1+1
WRITE(2,5040)YCOR(I),ZCOR(I),AREA(J,J1),THICK(J+1,J1)
5040 FORMAT(' L'.2X.3(F8.2,1X),FS.5)
CONTINUE
CONTINUE
C
SAME INFORMATION FOR AFT WING
C
IF(NTYP1 .NE. 'JOIN')GO TO 800
WRITE(2,5032)NRIB2
DO 798 J=2,NRIB2+1
WRITE(2,5032)NSTR2
JJ=0
J1=J+NRIB-1
ISTRT=ISTA2(J)
IF(J.EQ.NRIB2+1)ISTRT=ISTA2(NRIB2)+NSTR2
IFIN=ISTRT+NSTR2-1
DO 795 I=ISTRT,IFIN
JJ=JJ+1
WRITE(2,5040)YCOR(I),ZCOR(I),AREA(JI,JJ),THICK(JI+1,JJ)
795 CONTINUE
798 CONTINUE
800 CONTINUE
CLOSE(UNIT=2)

C
C
C*---------------------------------------------------------------
C** GENERATE DATA FOR BEAM FINITE ELEMENT PROGRAM **
C*---------------------------------------------------------------
C
OPEN (UNIT=11, NAME='BEAM1.DAT', STATUS='NEW')
WRITE (11,5050)NTYP1,NTYP2
5050 FORMAT (A4,A2)
WRITE (11,5060)NSTA,NRIB,B
5060 FORMAT (IX,2(I3,IX),F7.2)
DD=1
TSKVOL=0
K=2
IF (NTYP1 .EQ. 'JOIN') THEN
  DIST=B2/NSTA2
  NDUM=NSTA2
ELSE
  DIST=B/NSTA
  NDUM=NSTA
ENDIF
DO 803 I=1,NSTA
IF (B .EQ. B) GO TO 801
IF (I .GT. NDUM) DIST=(B-B2)/(NSTA-NSTA2)
CONTINUE
801 CONTINUE
DO 804 J=1,NSTR
C1SUB=I*NSTR+J
C1=YCOR(I*NSTR+J)
C2=YCOR(I*NSTR+J+1)
C3=YCOR((I-1)*NSTR+J)
C4=YCOR((I-1)*NSTR+J+1)
C1Z=ZCOR(C1SUB)
C2Z=ZCOR(C1SUB+1)
C3Z=ZCOR(C1SUB-NSTR)
C4Z=ZCOR(C1SUB-NSTR+1)
IF (C1SUB .EQ. NSTR*(I+1)) THEN
  C2=YCOR(C1SUB-NSTR+1)
  C4=YCOR(C1SUB-(2*NSTR)+1)
  C2Z=ZCOR(C1SUB-NSTR+1)
  C4Z=ZCOR(C1SUB-(2*NSTR)+1)
ENDIF
RLEN=(XCOR(C1SUB)-XCOR(C1SUB-NSTR))**2+
$ (ZCOR(C1SUB)-ZCOR(C1SUB-NSTR))**2
IF (C1SUB .EQ. NSTR*(I+1) .OR. C1SUB .EQ. (NSTR/2)+1) +
$ \text{IF} (\text{NSTR} \cdot \text{I}) \text{THEN} \\
\text{RLEN} = (\text{XCOR(C1SUB)} - \text{XCOR(C1SUB-NSTR)})^2 + \\
(\text{YCOR(C1SUB)} - \text{YCOR(C1SUB-NSTR)})^2 \\
\text{ENDIF} \\
\text{RLEN} = \text{SQRT}(\text{RLEN}) \\
\text{SKAREA} = \text{RLEN} \cdot 0.5 \cdot \\
(\text{SQRT((C2-C1)^2+(C2Z-C1Z)^2)} + \text{SQRT((C3-C4)^2+} \\
(C3Z-C4Z)^2)) \cdot \\
\text{IF(J.EQ.NSTR) LL=1} \\
\text{SKVOL} = \text{SKAREA} \cdot \text{THICK(K, LL)} \\
\text{TSKVL=TSKVL+SKVOL} \\
\text{CONTINUE} \\
\text{IF(C1SUB.EQ.IEND) SKIN(K-1)=TSKVL/DIST} \\
\text{IF(C1SUB.NE.ISTA(K-1)+NSTR-1) DD=DD+1} \\
\text{DIST=DIST*DD} \\
\text{IF(C1SUB.EQ.ISTA(K-1)+NSTR-1) THEN} \\
\text{SKIN(K-1)=TSKVL/DIST} \\
\text{TSKVL=0} \\
K=K+1 \\
DD=1 \\
\text{ENDIF} \\
\text{CONTINUE} \\
\text{IF(NTYP1.NE.'JOIN') GO TO 815} \\
\text{C FOR THE AFT WING} \\
K=K+1 \\
\text{WRITE}(11,5060) NSTA2, NRIB2, B2 \\
DD=1 \\
\text{TSKVL=0} \\
\text{DO 810 I=1,NSTA2} \\
\text{DIST=B2/NSTA2} \\
\text{DO 806 J=IEND+1,IEND+NSTR2} \\
\text{C1SUB=I*NSTR2+J} \\
\text{C1=YCOR(I*NSTR2+J)} \\
\text{C2=YCOR(I*NSTR2+J+1)} \\
\text{C3=YCOR((I-1)*NSTR2+J)} \\
\text{C4=YCOR((I-1)*NSTR2+J+1)} \\
\text{C1Z=ZCOR(C1SUB)} \\
\text{C2Z=ZCOR(C1SUB+1)} \\
\text{C3Z=ZCOR(C1SUB-NSTR2)} \\
\text{C4Z=ZCOR(C1SUB-NSTR2+1)} \\
\text{IF(C1SUB.EQ.IEND+NSTR2*(I+1)) THEN} \\
\text{C2=YCOR(C1SUB-NSTR2+1)} \\
\text{C4=YCOR(C1SUB-(2*NSTR2)+1)} \\
\text{C2Z=ZCOR(C1SUB-NSTR2+1)} \\
\text{C4Z=ZCOR(C1SUB-(2*NSTR2)+1)} \\
\text{ENDIF} \\
\text{RLEN=(XCOR(C1SUB)-XCOR(C1SUB-NSTR2))^2+} \\
(\text{ZCOR(C1SUB)-ZCOR(C1SUB-NSTR2)})^2 \\
\text{IF(C1SUB.EQ.IEND+NSTR2*I+1.OR.C1SUB.EQ.} \\
(\text{NSTR2/2+1)+IEND+NSTR2*I) THEN} \\
\text{RLEN=(XCOR(C1SUB)-XCOR(C1SUB-NSTR2))^2+} \\
(\text{YCOR(C1SUB)-YCOR(C1SUB-NSTR2)})^2 \\
\text{ENDIF} \\
\text{RLEN=SQRT(RLEN)} \\
\text{SKAREA=RLEN} \cdot 0.5 \cdot \\
(\text{SQRT((C2-C1)^2+(C2Z-C1Z)^2)} + \text{SQRT((C3-C4)^2+} \\
(C3Z-C4Z)^2))
LL=J+1-IEND
IF(J-IEND.EQ.NSTR2)LL=1
SKVOL=SKAREA*THICK(K-2,LL)
TSKVOL=TSKVOL+SKVOL

806 CONTINUE
IF(CISUB.EQ.IREND)SKIN(K-2)=TSKVOL/DIST
IF(CISUB.NE.ISTA2(K-1-NSTA)+NSTR2-1)DD=DD+1
DIST=DIST*DD
IF(CISUB.EQ.ISTA2(K-1-NSTA)+NSTR2-1)THEN
SKIN(K-2)=TSKVOL/DIST
TSKVOL=0
K=K+1
DD=1
ENDIF
CONTINUE
810 CONTINUE
815 CONTINUE

C GENERATE BEAM LOADS
C
ISUB=(NSTR+2)/2
DO 817 I=1,NSTA+1
BLOAD(I)=RLOAD(I,1)+RLOAD(I,2)
RMOMNT(I)=RLOAD(I,1)*(-0.5*CHORD(I))+RLOAD(I,2)*0.5*CHORD(I)
$ IF(XCOR(ISUB).EQ.B2)THEN
  NCOM=I
  YCOM=YCOR(ISUB)
  ZCOM=ZCOR(ISUB)
ENDIF
WRITE(11,5070)XCOR(ISUB),YCOR(ISUB),ZCOR(ISUB),BLOAD(I),RMOMNT(I)
$ 5070 FORMAT(1X,3(F9.2,2X),F10.2,2X,F9.2)
ISUB=ISUB+NSTR
CONTINUE
817 CONTINUE
IF(NTYP1.NE.'JOIN')GO TO 820
ISUB=IEND+(NSTR2+2)/2
DY=ABS(YCOM-YCOR(ISUB))
DZ=ABS(ZCOM-ZCOR(ISUB))
ISUB=IEND+3*NSTR2/2+1
DO 818 I=1,NSTA2
J=I+NSTA+1
BLOAD(J)=RLOAD(J+1,1)+RLOAD(J+1,2)
RMOMNT(J)=RLOAD(J+1,1)*(-0.5*CHORD(J+2))+RLOAD(J+1,2)*0.5*CHORD(J+2)
$ YCOR(ISUB)=YCOR(ISUB)+DY
ZCOR(ISUB)=ZCOR(ISUB)+DZ
WRITE(11,5070)XCOR(ISUB),YCOR(ISUB),ZCOR(ISUB),BLOAD(J),RMOMNT(J)
$ 5070 FORMAT(1X,3(F9.2,2X),F10.2,2X,F9.2)
ISUB=ISUB+NSTR2
CONTINUE
818 CONTINUE
820 CONTINUE
DO 825 J=1,NRIB
ISTA(J)=XCOR(ISTA(J))
WRITE(11,5080)ISTA(J)
8080 FORMAT(1X,F9.2)
CONTINUE
IF(NTYP1.NE.'JOIN')GO TO 831
DO 829 J=1,NRIB2
I=J+NRIB
a-33
XISTA(I) = XCOR(ISTA2(J))
WRITE(11, 5080) XISTA(I)
CONTINUE
DO 832 I = 1, NRT
WRITE(11, 5080) SKIN(I)
CONTINUE
CLOSE(UNIT=11)

C
C***************************************************************
C**
C**    GENERATE DATA FOR SHEAR LAG PROGRAM
C**
C***************************************************************
C
OPEN(UNIT=8, NAME='SL.DAT', STATUS='NEW')
WRITE(8, 5081) NTYP1
FORMAT(A4)
WRITE(8, *) NSTR, NSTA
IF(NTYP1 .EQ. 'JOIN') THEN
WRITE(8, *) NSTR2, NSTA2
ENDIF
CLOSE(UNIT=8)

C
C***************************************************************
C**
C**    GENERATE DATA FOR BEAM SECTION ANALYSIS
C**    PROGRAM [BSAP.FOR].
C**
C***************************************************************
C
OPEN(UNIT=12, NAME='BCSD.DAT', STATUS='NEW')
J = 1 + NSTR
DO 836 I = 1, NSTA
IF(XISTA(I) .EQ. XCOR(J)) THEN
WRITE(12, *) CHORD(I)
ENDIF
J = J + NSTR
CONTINUE
IF(NTYP1 .NE. 'JOIN') GO TO 836
J = IEND + 1
DO 837 I = NSTA + 1, NSTA2 + NSTA
IF(XISTA(I) .EQ. XCOR(J)) THEN
WRITE(12, *) CHORD(I + 2)
ENDIF
J = J + NSTR2
CONTINUE
STOP
END
DIMENSION STRESS(100,6)
DIMENSION T(100,5),TN(100,5)
DIMENSION UT1(100),UT2(100)
DIMENSION B(100)
DIMENSION DD(100)

READ THE NUMBER OF ELEMENTS ON BEAM
NSTA & NATA2

OPEN (UNIT=1,NAME='WING.DAT',STATUS='OLD')
OPEN(UNIT=64,NAME='KAPPA.DAT',STATUS='OLD')
READ(64,*)RKAP,SLCON,SLCON1
   READ(1,2) NTYP1
   FORMAT(A4)
   READ(1,* )BSP,CR,CT,NSTR,NST1,NRIB,RLAM,GAM,TR
   IF(NTYP1.EQ. 'JOIN') THEN
      READ(1,* )BSP2,CR2,CT2,NSTR2,NST1,NRIB2,RLAM2,GAM2,TR2
   ENDIF
   N=NST1
   IF(NTYP1.EQ. 'JOIN')N=NST1-NST2
   CLOSE(UNIT=1)

READ THE NUMBER OF POINT TO EVALUATE THE STRESS
DURING THE FULLY STRESSED DESIGN
N1 (IN EACH ELEMENT'S CROSS-SECTION)
READ THE CONVERGENCE CRITERIA EP
READ THE ALLOWABLE STRESS VALUE ALSTR(I)

OPEN (UNIT=2,NAME='ALST.DAT',STATUS='OLD')
READ(2,* ) N1
READ(2,* )EP
READ(2,* ) ALSTR
CLOSE(UNIT=2)

READ THE CURRENT DESIGN VARIABLE (THICKNESS)
T(I,J)

OPEN (UNIT=3,NAME='SKIN.DAT',STATUS='OLD')
   DO 4 I=1,N
      READ(3,* ) (T(I,J),J=1,5)
      4 CONTINUE
   READ(3,* )W1,W2
   CLOSE(UNIT=3,DISP='DELETE')

READ THE WIDTH OF BEAM FOR EACH ELEMENT
& COMPUTE THE UPPER BOUND OF THICKNESS

OPEN(UNIT=9,NAME='BCSD.DAT',STATUS='OLD')
   DO 5 I=1,N
      READ(9,* )B(I)
      UT1(I)=0.45*B(I)
      UT2(I)=0.45*RKAP*B(I)/0.65 -- Structural box is 65% of cho
CLOSE(UNIT=9)

C READ THE CURRENT STRESS FROM EAL OUTPUT
C
OPEN (UNIT=8, NAME='FOR004.DAT', STATUS='OLD')
READ(8,8) WEIGHT
8
FORMAT(11(/).10X,E11.5)
WEIGHT=WEIGHT*386.
CLOSE(UNIT=8)

C READ THE STRESS FROM STRESS.DAT
C
OPEN(UNIT=10, NAME='STRESS.DAT', STATUS='OLD')
DO 17 I=1,N
READ(10,*)(STRESS(I.J),J=1,N1)
17 CONTINUE

C FULLY STRESSED DESIGN STEPS
C
DO 20 I1=1,N
DO 19 J=1,N1
DD(J)=ABS(STRESS(I1.J))
19 CONTINUE

DDA=(DD(1)+DD(6))/(2.*ALSTR)
T1=T(I1.1)*DDA
DDA=(DD(3)+DD(4))/(2.*ALSTR)
T2=T(I1.1)*DDA
IF(T1.LT.T2)T1=T2
TN(I1,1)=T1
DDA=DD(1)
IF(DD(2).GE.DDA)DDA=DD(2)
DDA=DDA/ALSTR
TN(I1,2)=T(I1.2)*DDA
DDA=DD(2)
IF(DD(3).GE.DDA)DDA=DD(3)
DDA=DDA/ALSTR
TN(I1,3)=T(I1.3)*DDA
DDA=DD(4)
IF(DD(5).GE.DDA)DDA=DD(5)
DDA=DDA/ALSTR
TN(I1,4)=T(I1.4)*DDA
DDA=DD(5)
IF(DD(6).GE.DDA)DDA=DD(6)
DDA=DDA/ALSTR
TN(I1,5)=T(I1.5)*DDA

C CHECK THE UPPER Bound OF THICKNESS
C
IF(TN(I1,1).GT.UT1(I1)) TN(I1,1)=UT1(I1)
DO 14 I2=2,5
IF(TN(I1.I2).GT.UT2(I1)) TN(I1,12)=UT2(I1)
14 CONTINUE

C CHECK THE Minimum ALLOWABLE THICKNESS=0.05 (IN)
C
DO 15 I3=1,5
IF(TN(I1.I3).LT.0.05) TN(I1,I3)=0.05
15 CONTINUE
20 CONTINUE

A-36
C RENEW DESIGN VARIABLE
C
OPEN (UNIT=3, NAME='SKIN.DAT', STATUS='NEW')
DO 30 J=1,N
  WRITE(3,25) (TN(J,I), I=1,5)
25  FORMAT(1X,(F12.6,1X))
30  CONTINUE
  WRITE(3,*), WEIGHT, W1
CLOSE(UNIT=3)

C PRINTOUT THE ITERATION INFORMATION
C
OPEN (UNIT=5, NAME='FSDIF.DAT', STATUS='OLD', ERR=31)
CONTINUE
DO 38 I2=1,N
  WRITE(5,35) I2, (T(I2,J), TN(I2,J), J=1,5)
35    FORMAT(1X, 'ELEMENT #', I3, '/', (1X, 'T(OLD)=', F12.6, 
             1X, 'T(NEW)=', F12.6, '/))
  WRITE(5,36) (STRESS(I2,J), J=1,N1)
36    FORMAT(/, 1X, 'STRESS=', /, (9X, E12.3, /))
CONTINUE
  WRITE(5,39) WEIGHT, W1
OPEN (UNIT=54, NAME='FOR057.DAT', STATUS='OLD')
  WRITE(54,*), WEIGHT, W1
WRITE(6,*), WEIGHT, W1
39    FORMAT(/, 1X, 'OBJECTIVE FUNCTION( WEIGHT )=', F12.6, 
            'LBF')
CLOSE(UNIT=5)

C CHECK THE CONVERGENCE OF FULLY STRESSED DESIGN
C
D=(WEIGHT-W1)/WEIGHT
DN=(WEIGHT-W2)/WEIGHT
D1=ABS(D)
D2=ABS(DN)
IF(D1.GT.EP.OR.D2.GT.EP) GO TO 50

C EXIT FROM FULLY STRESSED DESIGN ITERATION LOOP
C
CALL LIB$DO_COMMAND
  '*(@ FSDEXIT')
GO TO 50
50 CONTINUE
STOP
END
**MODE FOR** - THIS PROGRAM CREATES AN E. A. L. COMMAND FILE FOR A BEAM WHICH IS IDENTICAL TO THE ONE CREATED BY BEAM.FOR EXCEPT THAT THERE ARE DOUBLE THE NUMBER OF BEAM ELEMENTS

**DIMENSION** RMAX(100), RMIN(100), ATOT(100), TCNST(100)

* , XNODE(100), YNODE(100), ZNODE(100), RLOAD(100)

* , RMOMNT(100), BEE(100), H(100), YPT(100,4), ZPT(100,4)

* , PHI(100), XISTA(100)

* , N(IO0)

OPEN(UNIT=1, NAME='MODE.COM', STATUS='NEW')
OPEN(UNIT=2, NAME='TRNSFR.DAT', STATUS='OLD')
OPEN(UNIT=3, NAME='BCSD1.DAT', STATUS='OLD')

**READ IN DATA**

READ(2,1000) NTYP1

1000 FORMAT(A4)

READ(2,*) NSTA, NRIB, B
NRT=NRIB
IF(NTYP1.EQ. 'JOIN') THEN
READ(2,*) NSTA2, NRIB2, B2
NRT=NRIB+NRIB2
ENDIF

READ(2,*) NUM
NUM=NUM*2-1
DO 100 I=1, NUM, 2
READ(2,*) XNODE(I), YNODE(I), ZNODE(I)
IF(XNODE(I).EQ.B2) NCOM=I
CONTINUE

100 CONTINUE
DO 101 I=2, NUM-1, 2
XNODE(I)=(XNODE(I-1)+XNODE(I+1))/2.
YNODE(I)=(YNODE(I-1)+YNODE(I+1))/2.
ZNODE(I)=(ZNODE(I-1)+ZNODE(I+1))/2.
IF(NTYP1.EQ. 'JOIN') GO TO 102
GO TO 101

102 IF(I.EQ.2*NRIB+2) THEN
XNODE(I)=(XNODE(I+1)+XNODE(NCOM))/2.
YNODE(I)=(YNODE(I+1)+YNODE(NCOM))/2.
ZNODE(I)=(ZNODE(I+1)+ZNODE(NCOM))/2.
ENDIF

101 CONTINUE
DO 105 I=1, NRT+NRT, 2
READ(2,*) XISTA(I)

105 CONTINUE
NRT1=2*NRT
DO 125 I=1, NRT1-1, 2
I1=I+1
READ(3,*) II, RMAX(I), RMIN(I), ATOT(I), TCNST(I), PHI(I)
RMAX(I1)=RMAX(I)

A-38
RIMIN(I1)=RIMIN(I)
ATOT(I1)=ATOT(I)
TCNST(I1)=TCNST(I)
PHI(I1)=PHI(I)
125 CONTINUE
DO 126 I=1,NRT
READ(3,*) BEE(I)
126 CONTINUE
NLOAD=NSTA+NSTA2-2
IF(NSTA2.EQ.0) NLOAD=NLOAD+1
DO 127 I=1,NLOAD
READ(2,*)N(I),RLOAD(I),RMOMNT(I)
127 CONTINUE
DO 128 I=1,NLOAD
N(I)=N(I)+I
IF(I.GE.NSTA)N(I)=N(I)+1
128 CONTINUE
C C BEGIN WRITING E. A. L. COMMAND FILE
C WRITE(1,1020)NUM
1020 FORMAT('RUN DUAI:[PXH]EAL'/
  'DUAL:[]XXX.'/
  'ONLINE=0'/
  'OUTF=6'/
  'XQT TAB'/
  'START ',I2,',6'/
  'MATC:1 10.5+6 .3 .1'/
  'JLOC'
C DO 130 I=1,NUM
WRITE(1,1025)I,XNODE(I),YNODE(I),ZNODE(I)
1025 FORMAT(1X,I2,3(1X,F9.2))
130 CONTINUE
WRITE(1,1050)
1050 FORMAT('CON 1'/
  ZERO 1,2,3,4,5 : 1')
IF(NTYP1.NE. 'JOIN')GO TO 135
WRITE(1,1055)NUM
1055 FORMAT('ZERO 1,2,3,4,5 : ',I2)
135 CONTINUE
C C WRITE MREF'S FOR EACH BEAM SECTION
C WRITE(1,*)'MREF'
NREF=NRT1.
DO 140 I=1,NREF
SIGN=1.0
IF(PHI(I) .LT. 0.) SIGN=-1.
COSN=ABS(SIN(PHI(I)))
WRITE(1,1060)I,SIGN,COSN
1060 FORMAT(IX,12,1X,'2',IX,'3',IX,F8.4,1X,F8.4)
140 CONTINUE
WRITE(1,*)'BA'
C C WRITE BEAM SECTION PROPERTIES
C [ASSUME HEIGHT=65% BASE]
C DO 210 I=1,NRT1
WRITE(1,1078)I,RIMAX(I),RIMIN(I),ATOT(I),TCNST(I)
*  PHI(I)
1078  FORMAT( ' GIVN ', I3, 1X, F10.4, ' 0.0 ', F10.4, ' 0.0 ',
' F10.4,1X,F10.4,' 0.0 ',1X,' 0.0 ',' 0.0 ',F10.4)

210  CONTINUE
    WRITE(1,2000)
2000  FORMAT( 'XQT ELD/', ' E21' )

C  WRITE BEAM CONNECTIVITY
C
    NSECT=1
    WRITE(1,4000)NSECT
    WRITE(1,4001)1
    ISTRT=1
    IFIN=2
    WRITE(1,4500)ISTRT,IFIN
    DO 600 I=2,NSTA+NSTA
        NSECT=NSECT+1
        WRITE(1,4000)NSECT
        WRITE(1,4001)NSECT
    4000  FORMAT( ' NSECT=', I2)
    4001  FORMAT( ' NREF=', I2)
        ISTRT=I
        IFIN=I+1
        WRITE(1,4500)ISTRT,IFIN
    4500  FORMAT(1X,2(I3,1X))
    CONTINUE

C  NSECTS & CONNECTIVITY FOR AFT WING.
C
    IF(NTYPI.NE. 'JOIN')GO TO 350
    NSECT=NSECT+1
    WRITE(1,4000)NSECT
    WRITE(1,4001)NSECT
    WRITE(1,4500)NCOM,NSTA+NSTA+NSTA+2
    NSECT=NSECT+1
    DO 325 I=NSTA+NSTA+2,2*(NSTA+NSTA2)
        WRITE(1,4000)NSECT
        WRITE(1,4001)NSECT
        NSECT=NSECT+1
        ISTRT=I
        IFIN=I+1
        WRITE(1,4500)ISTRT,IFIN
    325  CONTINUE
    350  CONTINUE
C
    WRITE(1,2020)
2020  FORMAT( 'XQT E/
        * RESET G=356.*/
        "XQT TOPO"/
        "XQT EKS"/
        "XQT K"/
        "XQT INV"/
        "XQT AUS"/
        " SYSVEC: APPL FORC")

C  APPLY LOADS & MOMENTS TO BEAM.
C
    DO 400 I=1,NLOAD
WRITE(1,3000)N(I),RLOAD(I)
3000 FORMAT(' I=3: J=',I2,':',F9.2)
WRITE(1,3001)N(I),RMOMNT(I)
3001 FORMAT(' I=4: J=',I2,':',F9.2)
400 CONTINUE
WRITE(1,6100)NUM
6100 FORMAT(6000)
WRITE(1,6000)
6000 FORMAT(6000)
CLOSE(UNIT=1)
STOP
END
THIS PROGRAM WILL CALCULATE THE CENTROIDS MOMENTS OF INERTIA AND ELASTIC AXES OF A WING BOX STRUCTURE.

DIMENSION AR(500), XYC(500, 2), RIX(50), RIY(50), RIXY(50)
DIMENSION DS(50), T(50), RJ(50), ATO(50), ARSK(500), SH(200, 5)

OPEN(UNIT=1, NAME='INPUT.DAT', STATUS='OLD')

READ(1, 5) NTYP1, NTYP2
FORMAT(A4, A2)
READ(1, *) NRIB
DO 4000 II = 1, NRIB
RIXT = 0.0
RIYT = 0.0
RIXYT = 0.0
DST = 0.0
READ(1, *) NSTR
READ(1, *) ((SH(I, J), J = 1, 5), I = 1, NSTR)
DO 10 I = 1, NSTR
T(I) = SH(I + 1, 5)
IF(I.EQ.NSTR) T(I) = SH(1, 5)
TEMP = I + 1
IF(I.EQ.NSTR) TEMP = 1
DS(I) = SQRT((SH(I, 3) - SH(TEMP, 3))**2 + (SH(I, 2) - SH(TEMP, 2))**2)
$ / T(I)
XYC(I, 1) = (SH(I, 1) + SH(TEMP, 1)) / 2.
XYC(I, 2) = (SH(I, 2) + SH(TEMP, 2)) / 2.
DST = DST + DS(I)
10 CONTINUE
CALL AREA(SH, NSTR, AR, AT, ARSK, DS, T)
CALL CG(SH, NSTR, AR, XYC, XBAR, YBAR, AT, ARSK)
CALL MOMENTS(SH, AR, XYC, RIXC, RIYC, RIXT, RIYT, RIXY, NSTR, AT, XBAR, YBAR, DS, T, DST, ARSK)
RIX(II) = RIXT
RIY(II) = RIYT
RIXY(II) = RIXYT
DO 1 I = 1, NSTR
AT = AT - ARSK(I)
1 CONTINUE
ATO(II) = AT
A = (SH(3, 2) - SH(2, 2))**2 + (SH(3, 3) - SH(NSTR, 3)) + (SH(2, 3) - SH(1, 3)) + (ABS(SH(3, 2) - SH(NSTR/2, 2)) * ABS(SH(3, 3) - SH(NSTR, 3)))
$ RJ(II) = 4 * A**2 / DST
4000 CONTINUE

SAME CALCULATIONS FOR AFT WING.
DO 12 I=1,NSTR2
TEMP=I+1
IF(I.EQ.NSTR2) TEMP=1
T(I)=SH(TEMP,5)
DS(I)=SQRT((SH(I,3)-SH(TEMP,3))**2+(SH(I,2)-SH(TEMP,2))**2)/T(I)
XYC(I,1)=(SH(I,2)+SH(TEMP,2))/2.0
XYC(I,2)=(SH(I,3)+SH(TEMP,3))/2.0
DST=DST+DS(I)
12 CONTINUE
CALL AREA(SH,NSTR2,AR,AT,ARSK,DS,T)
CALL CG(SH,NSTR2,AR,XYC,XBAR,YBAR,AT,ARSK)
CALL MOMENTS(SH,AR,XYC,RIxC,RIyC,RIxT,RIyT,RIxyT,
$ NSTR2,AT,XBAR,YBAR,DS,T,DST,ARSK)
RIx(II)=RIxT
RIy(II)=RIyT
RIxy(II)=RIxyT
DO 40 I=1,NSTR2
40 CONTINUE
AT=AT-ARSK(I)
ATO(II)=AT
A=(SH(3,2)-SH(2,2))**2+(SH(3,3)-SH(NSTR2,3))+(SH(2,3)
$ -SH(1,3))+(ABS(SH(3,2)-SH(NSTR2/2,2))*ABS(SH(3,3)-
$ SH(NSTR2,3))
RJ(II)=4.0*A**2/DST
4001 CONTINUE
26 CONTINUE
OPEN(UNIT=12,NAME='BEAM2.DAT',STATUS='NEW')
NDUM=NRIB
IF(NTYP1.EQ.'JOIN') NDUM=NRIB+NRIB2
DO 225 I=1,NDUM
WRITE(12,6002)RIX(I),RIY(I),ATO(I),RJ(I),RIXY(I)
6002 FORMAT(1X,5(F13.3,E10.2))
225 CONTINUE
STOP
END
C
C SUBROUTINE AREA
C
C
SUBROUTINE AREA(SH,N,AR,AT,ARSK,DS,T)
DIMENSION SH(200,5),AR(500),ARSK(500),DS(50),T(50)
AT=0
DO 40 I=1,N
AR(I)=SH(I,4)
ARSK(I)=DS(I)*T(I)**2
40 AT=AT+AR(I)+ARSK(I)
RETURN
END
C
C SUBROUTINE CG
SUBROUTINE CG(SH,N,AR,XYC,XBAR,YBAR,AT,ARSK)
DIMENSION SH(200,5),AR(500),XYC(500,2),ARSK(500)
SUM1=0.
SUM2=0.
DO 80 I=1,N
   SUM1=SUM1+SH(I,2)*AR(I)+XYC(I,1)*ARSK(I)
   SUM2=SUM2+SH(I,3)*AR(I)+XYC(I,2)*ARSK(I)
CONTinue
XBAR=SUM1/AT
YBAR=SUM2/AT
RETURN
END

SUBROUTINE MOMENTS

SUBROUTINE MOMENTS(SH,AR,XYC,RIXC,RIYC,RIXT,RIYT,RIXYT,N,AT,XBAR,YBAR,DS,T,DST,ARSK)
DIMENSION SH(200,5),AR(500),XYC(500,2),RIX(10),RIY(10),RIXY(10)
DIMENSION DS(50),T(50),ARSK(500)
DO 15 I=1,N
   THETA=0.
   IF(I.EQ.1) GO TO 99
   IF(NTYP1.EQ. 'JOIN') THEN
      IF(I.EQ.5) GO TO 99
      GO TO 90
   ENDIF
   IF(I.EQ.4) GO TO 99
   CONTINUE
   RIOY=I./12.*T(I)*(DS(I)*T(I))**3+(AR(I)**2)/(4.*3.14159)
   RIOX=I./12.*(DS(I)*T(I))*T(I)**3+(AR(I)**2)/(4.*3.14159)
   GO TO 89
99   RIOY=I./12.*(DS(I)*T(I))*T(I)**3+(AR(I)**2)/(4.*3.14159)
   RIOX=I./12.*T(I)*(DS(1)*T(1))**3+(AR(I)**2)/(4.*3.14159)
   CONTINUE
   UC=SH(I,2)-XBAR
   UCS=XYC(I,1)-XBAR
   VCS=XYC(I,2)-YBAR
   VC=SH(I,3)-YBAR
   CALL MOMAXIS(THETA,SH,AR,RIOX,RIOY,UC,VC,I,RIX,RIY,RIXY,$ RIXT,RIYT,RIXYT,VCS,UCS,ARSK)
CONTINUE
RETURN
END

SUBROUTINE MOMAXIS

SUBROUTINE MOMAXIS(THETA,SH,AR,RIOX,RIOY,UC,VC,I,RIX,RIY,RIXY,$ RIXT,RIYT,RIXYT,VCS,UCS,ARSK)
DIMENSION SH(200,5),AR(500),RIX(10),RIY(10),RIXY(10),ARSK(500)
RIX(I)=RIOX+AR(I)*VC**2+ARSK(I)*VCS**2
RIY(I)=RIOY+AR(I)*UC**2+ARSK(I)*UCS**2
RIXY(I)=AR(I)*UC*VC+ARSK(I)*UCS*VCS
RIXT=RIXT+RIX(I)
RIYT = RIYT + RIY(I)
RIXYT = RIXYT + RIXY(I)
RETURN
END
**PROGRAM TO REDUCE SHEAR LAG EFFECTS**

C**

DIMENSION WGDISP(500), BMDISP(100)
DIMENSION WGDISPI(500), BMDISPI(100)

OPEN(UNIT=1, NAME='WING.DAT', STATUS='OLD')
OPEN(UNIT=7, NAME='FOR003.DAT', STATUS='OLD')
OPEN(UNIT=8, NAME='FOR004.DAT', STATUS='OLD')
OPEN(UNIT=37, NAME='KAPPA.DAT', STATUS='OLD')
READ(37,*) RKAP, SLCON, SLCONI
REWIND 37

READ THE DISPLACEMENTS OF ALL
NODGES ON WING & STORE IN ARRAY[WGDISP(I)].

READ(1,2) NTYP1
2 FORMAT(A4)
READ(1,*) BSP, CR, CT, NSTR, NSTA, NRIB, RLAM, GAM, TR
RLAMD=RLAM
RLAM=3.14159*RLAM/180.
GAMD=GAM
GAM=3.14159*GAM/180.
IEND=NSTR*(NSTA+1)
IREND=IEND
IF(NTYP1.EQ. 'JOIN') THEN
    READ(1,*) BSP2, CR2, CT2, NSTR2, NSTA2, NRIB2, RLAM2, GAM2, TR2
    IREND=IEND+NSTR2*(NSTA2+1)
ENDIF
READ(7,100) (WGDISPI(I), WGDISP(I), I=1, IREND)
100 FORMAT(//////,(21X,E12.5,E12.5))

READ THE DISPLACEMENTS OF NODES ON BEAM
& STORE IN ARRAY[BMDISP(I)]

INODE=NSTA+1
IF(NTYP1.EQ. 'JOIN') INODE=NSTA+NSTA2+1
READ(8,105) (BMDISPI(I), BMDISP(I), I=1, INODE)
105 FORMAT(//////,(21X,E12.5,E12.5))
CLOSE(UNIT=7, DISP='DELETE')
CLOSE(UNIT=8, DISP='DELETE')

FIND THE MAXIMUM DISPLACEMENT ON BEAM

BMAX=ABS(BMDISP(I))
BMAX1=ABS(BMDISPI(I))
DO 6 I=1, INODE
   IF(ABS(BMDISPI(I)).LT.BMAX1) GO TO 66
   BMAX1=ABS(BMDISPI(I))
6 CONTINUE
   IF(ABS(BMDISP(I)).LT.BMAX) GO TO 6
   BMAX=ABS(BMDISP(I))
6 CONTINUE

FIND THE MAXIMUM DISPLACEMENT ON WING

A-46
WMAX = ABS(WGDISP(1))
WMAX1 = ABS(WGDISP1(1))
DO 8 I=1,IREND
IF(ABS(WGDISP(I)) .LT. WMAX) GO TO 8
WMAX = ABS(WGDISP(I))
CONTINUE
IF(ABS(WGDISP(I)) .LT. WMAX1) GO TO 88
WMAX1 = ABS(WGDISP1(I))
CONTINUE

CALCULATE THE SHEAR LAG CONSTANT

SLCON1 = 1.0
SLCON = BMAX / WMAX
RPAR = 0.925 * TR / (SQRT(SLCON) * COS(RLAM))
SLCON = 1.
IF(NTYPE .EQ. 'JOIN') THEN
CALL FACT(SINV, RLAMD, GAMD)
SLCON1 = SINV * BMAX1 / WMAX1
SLCON = SINV * BMAX / WMAX
RPAR = TR
ENDIF
WRITE (37, *) RPAR, SLCON, SLCON1
CLOSE (UNIT=1)
STOP
END

SUBROUTINE FACT(SINV, RLAMD, GAMD)

C THIS PROGRAM PROVIDES AN INTERPOLATION FOR THE CORRECTIVE
C FACTOR USED TO ACCOUNT FOR SHEAR LAG EFFECTS -- INTERPOLATION
C IS BASED ON CUBIC SPLINE FITS. AUGUST 1985

C IF(GAMD.LT.4.) GAMD = 4.
IF(GAMD.GE.20.) GAMD = 19.95
IF(RLAMD.LT.10.) RLAMD = 10.
IF(RLAMD.GE.30.) RLAMD = 29.95

C BEGIN INTERPOLATION

IF(GAMD.GE.4.0.AND.GAMD.LT.8.0) THEN
C1 = -9.8437E-2
C2 = 0.0
C3 = -8.7891E-4
YY = 2.2
XI = 4.0
GO TO 200
ENDIF

IF(GAMD.GE.8.0.AND.GAMD.LT.12.0) THEN
C3 = -2.8320E-3
C2 = -1.0547E-2
C1 = -0.1406
YY = 1.75
XI = 8.0
GO TO 200
ENDIF

IF(GAMD.GE.12.0.AND.GAMD.LT.16.0) THEN
C3 = -2.9297E-4

A-47
C2=-2.3437E-2
C1=-8.9063E-2
YY=1.2
XI=12.0
GO TO 200
ENDIF
IF(GAMD.GE.16.0.AND.GAMD.LT.20.0)THEN
C3=-1.6602E-3
C2=1.9922E-2
C1=8.4375E-2
YY=1.20
XI=16.0
GO TO 200
ENDIF
CONTINUE
FNG=((C3*(GAMD-XI)+C2)*(GAMD-XI)+C1)*(GAMD-XI)+YY
IF((RLAMD.GE.10.0.AND.RLAMD.LT.15.0)THEN
C3=-8.7143E-4
C2=0.0
C1=-4.8214E-2
YY=3.25
XI=10.0
GO TO 300
ENDIF
IF((RLAMD.GE.15.0.AND.RLAMD.LT.20.0)THEN
C3=2.3571E-3
C2=-1.3071E-2
C1=-0.1136
YY=2.9
XI=15.0
GO TO 300
ENDIF
IF((RLAMD.GE.20.0.AND.RLAMD.LT.25.0)THEN
C3=2.5571E-3
C2=-2.2286E-2
C1=-6.75E-2
YY=2.3
XI=20.0
GO TO 300
ENDIF
IF((RLAMD.GE.25.0.AND.RLAMD.LT.30.0)THEN
C3=1.0714E-3
C2=-1.6071E-2
C1=-3.6429E-2
YY=2.2
XI=25.0
GO TO 300
ENDIF
CONTINUE
FNR=((C3*(RLAMD-XI)+C2)*(RLAMD-XI)+C1)*(RLAMD-XI)+YY
SINV=FNG*FNR/2.2
RETURN
END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

J. L. CHEN/ P. HAJELA APRIL 1985

* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

STRESS.FOR PROGRAM TO COMPUTE THE STRESS
FROM THE DISPLACEMENT RESULTS

* * * * * * * * * * * * * * * * * * * * * * * * * * * * *

DIMENSION RIXX(100), RIYY(100), RIXY(100), DISP1(100), DISP2(100)
DIMENSION B(100), YCORD(100), ZCORD(100)
DIMENSION YM(100), ZM(100), STRESS(100,6)
DIMENSION XCORD(100), YP(100,6), ZP(100,6)
DIMENSION TDISP1(100), TDISP2(100)
DIMENSION ATOT(100), RJ(100), NJOIN(100), RLOAD(100), RMONT(100)
DIMENSION V(100)
DIMENSION TORQUE(100), SHEAR(100,6), DIS(100,6)
OPEN(UNIT=1, NAME='INERT.DAT', STATUS='OLD')
OPEN(UNIT=2, NAME='FOR004.DAT', STATUS='OLD')
OPEN(UNIT=3, NAME='TRNSFR.DAT', STATUS='OLD')
OPEN(UNIT=4, NAME='STRESS.DAT', STATUS='NEW')
OPEN(UNIT=5, NAME='BCSD.DAT', STATUS='OLD')
OPEN(UNIT=37, NAME='KAPPA.DAT', STATUS='OLD')
READ(37, *) RKAP, SLCON, SLCON1
READ(3, 9) NTYP1

9 FORMAT(A4)
READ(3, *) NSTA, NRIB, B1
N=NSTA
IF(NTYP1.EQ. 'JOIN') THEN
READ(3, *) NSTA2, NRIB2, B2
N=NSTA+NSTA2
ENDIF
READ(3, *) NUM

1 READ THE MOMENT OF INERTIA
C RIXX, RIYY, RIXY

READ(1, *) (RIXX(I), RIYY(I), RIXY(I), I=1,N)
READ(1, *) (ATOT(I), RJ(I), I=1,N)

C READ THE DISPLACEMENT VALUES
C
READ(2, 5) NN
FORMAT(18(/), I2)
READ(2, 10)(DISP1(I), DISP2(I), I=1,N+1)

10 FORMAT(21X, E12.5, E12.5)

C READ THE NODE'S X-COORDINATES
C
DO 20 I=1,N1
READ(3, *) XCORD(I), YCORD(I), ZCORD(I)
IF(ABS(XCORD(I)-B2).LE.0.001) NCOM=I
20 CONTINUE

C READ THE CROSS-SECTION PROPERTY
C & COMPUTE THE POSITION TO EVALUATE THE STRESS
C
READ(8, *) (B(I), I=1,N)
DO 25 I=1,N
YP(I,1)=-B(I)/2.

C

A-49
ZP(I,1)=RKAP*B(I)/(0.65*2.0)
YP(I,2)=0.
ZP(I,2)=RKAP*B(I)/(2.*0.65)
YP(I,3)=-YP(I,1)
ZP(I,3)=ZP(I,1)
YP(I,4)=-YP(I,1)
ZP(I,4)=-ZP(I,1)
YP(I,5)=0.
ZP(I,5)=-ZP(I,1)
YP(I,6)=YP(I,1)
ZP(I,6)=-ZP(I,1)

CONTINUE
DO 26 I=1,N
DO 26 J=1,6
DIS(I,J)=SQRT((ABS(YP(I,J)))**2+(ABS(ZP(I,J)))**2)

CONTINUE
C COMPUTE THE CURVATURE OF THE BEAM'S DEFLECTION
C
DO 40 I=1,N
IF(I.EQ.NSTA+1)THEN
  J1=NCOM
  GIS=(XCORD(I+I)-XCORD(J1))**2 + (YCORD(I+I)-YCORD(J1))**2
  $+ (ZCORD(I+I)-ZCORD(J1))**2
  GO TO 99
ENDIF
GIS=(XCORD(I+1)-XCORD(I))**2 + (YCORD(I+1)-YCORD(I))**2
$+ (ZCORD(I+1)-ZCORD(I))**2

99  CONTINUE
H=(S@RT(GIS))/2.
I1=I+1
IF(I.EQ.1)GO TO 30
IF(I.GE.NSTA+1) I1=I*2+1
IF(NTYPI.EQ. 'JOIN') THEN
  IF(I.EQ.N)GO TO 31
ENDIF
TDISP1(I)=(DISP1(I-1)-2.*DISP1(I1)+DISP1(I1+I))/(H**2)
TDISP2(I)=(DISP2(I-1)-2.*DISP2(I1)+DISP2(I1+I))/(H**2)
GO TO 40
I1=I*2
30  TDISP1(I)=2.*(DISP1(I1+1)-DISP1(I))/H**2
TDISP2(I)=2.*(DISP2(I1+1)-DISP2(I))/H**2
IF(I.LE.NSTA)GO TO 40
IF(NTYPI.EQ. 'JOIN')THEN
TDISP1(I)=-TDISP1(I)
TDISP2(I)=-TDISP2(I)
ENDIF
40  CONTINUE
C
C COMPUTE THE MOMENT MY,MZ
C
E=10.5E+6
DO 50 I=1,N
YM(I)=E*RIXX(I)*TDISP2(I)
ZM(I)=E*RIYY(I)*TDISP1(I)
50  CONTINUE
C
C COMPUTE THE SHEAR STRESS
C
READ(3,*)(XCORD(I),I=1,N)
K=NSTA-1
IF(NTYP1.EQ. 'JOIN')K=NSTA+NSTA2-2
READ(3,*) (NJOIN(I), RLOAD(I), RMONT(I), I=1, K)
TRMONT=0.
TRLOAD=0.
DO 70 I=1, 5
    TRMONT=TRMONT+RMONT(I)
70 TRLOAD=TRLOAD+RLOAD(I)
    V(1)=TRLOAD
    TORQUE(1)=TRMONT
DO 80 I=2, NSTA
    V(I)=V(I-1)-TRLOAD(I-1)
80 TORQUE(I)=TORQUE(I-1)-RMONT(I-1)
IF(NTYP1.EQ. 'JOIN')THEN
    V(7)=RLOAD(5)+RLOAD(4)
70 TORQUE(7)=RMONT(5)+RMONT(4)
    V(8)=RLOAD(6)+V(7)
80 TORQUE(8)=RMONT(6)+TORQUE(7)
    V(9)=RLOAD(7)+V(8)
80 TORQUE(9)=RMONT(7)+TORQUE(8)
    V(10)=RLOAD(8)+V(9)
70 TORQUE(10)=RMONT(8)+TORQUE(9)
ENDIF
DO 90 I=1, N
85 SHEAR(I, J)= (TORQUE(I)*DIS(I, J))/RJ(I)
90 V(I)=1.5*V(I)/ATOT(I)
C
C COMPUTE THE STRESS IN EACH POSITION
C
DO 60 I=1, N
60 DD=((YM(I)*RIXY(I)-ZM(I)*RIXX(I))*YP(I, J)+
    *(ZM(I)*RIXY(I)-YM(I)*RIYY(I))*ZP(I, J))/
    *(RIXX(I)*RIYY(I)-RIXY(I)**2)
DI=(ABS(DD))**2
STRESS(I, J)=SQRT(DI)
CONTINUE
WRITE(4,*) (STRESS(I, K), K=1, 6)
CONTINUE
CLOSE(UNIT=1)
CLOSE(UNIT=2)
CLOSE(UNIT=3)
CLOSE(UNIT=4)
CLOSE(UNIT=6)
STOP
END
DIMENSION STRESS(200,20)
DIMENSION T(200,20),TN(200,20)
DIMENSION PSTRI(200,20),PSTR2(200,20)
DIMENSION DD(200)

READ THE NUMBER OF SKIN ELEMENTS ON WING
NSTA & NATA2

OPEN (UNIT=1, NAME='WING.DAT', STATUS='OLD')
READ(I,2) NTYP1,NTYP2
2 FORMAT(A4,A2)
READ(I,*) B,CR,CT,NSTR,NSTA,NRIB,RLAM,GAM,TR
IF(NTYP1.EQ. 'JOIN') READ(I,*) B2,CR2,CT2,NSTR2,NSTA2.
*NRIB2,RLAM2,GAM2,TR2
N=NSTA
NRT=NRT-NRIB
IF(NTYP1.EQ. 'JOIN') N=NSTA+NSTA2
IF(NTYP1.EQ. 'JOIN') NRT=NRT+NRIB2
CLOSE(UNIT=1)

READ THE NUMBER OF POINT TO EVALUATE THE STRESS
DURING THE FULLY STRESSED DESIGN
N1 (IN EACH ELEMENT'S CROSS-SECTION)

READ THE CONVERGENCE CRITERIA EP
OPEN (UNIT=2, NAME='ALST.DAT', STATUS='OLD')
READ(2,*) N1
READ(2,*) EP
READ(2,*) ALSTR
CLOSE(UNIT=2)

READ THE CURRENT DESIGN VARIABLE (THICKNESS)
T(I,J)

OPEN (UNIT=3, NAME='THICK.DAT', STATUS='OLD')
READ(3,*)(T(I,I),I=1,NRT)
DO 4 I=1,NRIB
READ(3,*)(T(I+I,J),J=1,NSTR)
4 CONTINUE
IF(NTYP1.EQ. 'JOIN') GO TO 5
DO 1 I=NRIB+1,NRT
READ(3,*)(T(I+I,J),J=1,NSTR2)
1 CONTINUE
5 READ(3,*) W1,W2
CLOSE(UNIT=3, DISP='DELETE')

READ THE CURRENT STRESS FROM EAL OUTPUT

OPEN (UNIT=8, NAME='FOR001.DAT', STATUS='OLD')
OPEN(UNIT=18, NAME='FOR002.DAT', STATUS='OLD')
IF(NTYP1.EQ. 'JOIN') GO TO 6
READ(8,10) WEIGHT
10 FORMAT(22(/),25X,E12.6)
READ(18,101)N3
FORMAT(140(/),20X,I2,/)  
DO 1020 I=1,N  
DO 1020 J=1,NSTR  
K=(I-1)*NSTR+J  
IF(K.EQ.4)GO TO 105  
IF(K.EQ.12)GO TO 105  
IF(K.EQ.20)GO TO 105  
IF(K.EQ.28)GO TO 105  
IF(K.EQ.36)GO TO 105  
IF(K.EQ.44)GO TO 105  
IF(K.EQ.52)GO TO 105  
READ(18,103)PSTR1(I,J),PSTR2(I,J)  
FORMAT(45X,E9.3,E9.3,////)  
GO TO 102  
READ(18,104)PSTR1(I,J),PSTR2(I,J)  
FORMAT(45X,E9.3,E9.3,10(/))  
GO TO 102  
READ(18,107)PSTR1(I,J),PSTR2(I,J)  
FORMAT(45X,E9.3,E9.3)  
CONTINUE  
CONTINUE  
READ(8,8)WEIGHT  
FORMAT(24(/),25X,E12.6)  
READ(18,111)N3  
FORMAT(275(/),20X,I2,/)  
DO 112 I=1,N  
K=NSTR  
IF(I.GT.NSTA)K=NSTR2  
DO 112 J=1,K  
KI=(I-I)*NSTR+J  
IF(I.GT.NSTA)KI=NSTA*NSTR+(I-NSTA-I)*NSTR2+J  
IF(KI.EQ.12)GO TO 115  
IF(KI.EQ.20)GO TO 115  
IF(KI.EQ.28)GO TO 115  
IF(KI.EQ.36)GO TO 115  
IF(KI.EQ.44)GO TO 115  
IF(KI.EQ.52)GO TO 115  
IF(KI.EQ.64)GO TO 115  
IF(KI.EQ.72)GO TO 115  
IF(KI.EQ.80)GO TO 115  
IF(KI.EQ.88)GO TO 115  
IF(KI.EQ.96)GO TO 115  
IF(KI.EQ.104)GO TO 115  
IF(KI.EQ.112)GO TO 115  
READ(18,103)PSTR1(I,J),PSTR2(I,J)  
GO TO 112  
READ(18,104)PSTR1(I,J),PSTR2(I,J)  
GO TO 112  
READ(18,107)PSTR1(I,J),PSTR2(I,J)  
CONTINUE  
CONTINUE  
CLOSE(UNIT=8)  
C  
C COMPUTE THE VON-MISES STRESS CRITERIA  
C  
DO 110 I=1,N  
J=NSTR
IF(I.GT.NSTA) J=NSTR2
DO 110 K=1,J
VMC=(ABS(PSTRI(I,K)))**2+(ABS(PSTR2(I,K)))**2
   *PSTR1(I,K)*PSTR2(I,K)
VMC=ABS(VMC)
STRESS(I,K)=SQRT(VMC)
110 CONTINUE
C
C FULLY STRESSED DESIGN STEPS
C
DO 20 II=1,NSTA
DO 19 J=1,NSTR
   DD(J)=ABS(STRESS(II,J))
19 CONTINUE
DO 60 K=1,NSTR
   DDD=DD(K)/ALSTR
   TN(II+1,K)=T(II+1,K)*DDD
60 CONTINUE
20 CONTINUE
IF(NTYPI.EQ. 'JOIN')THEN
   DO 70 I=NSTA+1,N
      DO 65 J=1,NSTR2
         DD(J)=ABS(STRESS(I,J))
      65 CONTINUE
      DO 68 K=1,NSTR2
         DDD=DD(K)/ALSTR
         TN(I+1,K)=T(I+1,K)*DDD
      68 CONTINUE
70 CONTINUE
ENDIF
C
C CHECK THE UPPER BOUND OF THICKNESS=3.0
C
DO 15 I=1,N
   IF(TN(I+1,2).LT.TN(I+1,5)) TN(I+1,2)=TN(I+1,5)
   IF(TN(I+1,2).GT.TN(I+1,5)) TN(I+1,5)=TN(I+1,2)
I3=NSTR
   IF(I.GT.NSTA)I3=NSTR2
   DO 15 I2=1,I3
      IF(TN(I+1,I2).GT.3.0) TN(I+1,I2)=3.0
      IF(TN(I+1,I2).LT.0.03) TN(I+1,I2)=0.03
15 CONTINUE
C
C RENEW DESIGN VARIABLE
C
OPEN (UNIT=3,NAME= 'THICK.DAT',STATUS= 'NEW')
WRITE(3,*)(T(I,I),I=1,NRT)
DO 30 J=1,N
   K=NSTR
   IF(J.GT.NSTA)K=NSTR2
      WRITE(3,25) (TN(J+1,I),I=1,K)
   25 FORMAT(1X,(F12.6,1X))
30 CONTINUE
WRITE(3,*)W2,WEIGHT
CLOSE(UNIT=3)
C
C PRINTOUT THE ITERATION INFORMATION

lower and upper bounds on skin thickness.
C
OPEN (UNIT=55, NAME='FSDIF.DAT', STATUS='OLD', ERR=31)
WRITE(57, *) W2, WEIGHT
31  CONTINUE
DO 38 I2=1, N+1
  K=NSTR
  IF (I2.EQ.1) K=NRT
  IF (I2.GT. NSTA+1) K=NSTR2
    WRITE(55, 35) I2, (T(I2, J), TN(I2, J), J=1, K)
35    FORMAT(1X, 'ELEMENT #', I3, /, (1X, 'T(OLD)=', F12.6,
                    1X, 'T(NEW)=', F12.6, /))
    IF (I2.EQ.1) GO TO 38
    WRITE(55, 36) (STRESS(I2-1, J), J=1, K)
36    FORMAT(/, 1X, 'STRESS=', /, (9X, E12.3, /))
    CONTINUE
    WRITE(55, 39) W2, WEIGHT
39    FORMAT(/, 1X, 'OBJECTIVE FUNCTION( WEIGHT )=', F12.6,
                      *'LEF')
CLOSE(UNIT=55)

C  
C  CHECK THE CONVERGENCE OF FULLY STRESSED DESIGN
C
D=(WEIGHT-W1)/W1
DB=(WEIGHT-W2)/W2
D2=ABS(DB)
D1=ABS(D)
IF (D1.GT.EP OR D2.GT.EP) GO TO 50

C  
C  EXIT FROM FULLY STRESSED DESIGN ITERATION LOOP
C
CALL LIB$DO_COMMAND
*('@ FSDEXIT')
50  CONTINUE
STOP
END
Reduced Complexity Structural Modeling for Automated Airframe Synthesis

Prabhat Hajela
University of Florida, Gainesville, FL

Ames Research Center
Moffett Field, CA 94035

National Aeronautics and Space Administration
Washington, DC 20546

Point of Contact: Steve C. Smith, Ames Research Center, M/S 227-2
Moffett Field, CA 94035, (415)694-5856

The present report documents a procedure for the optimum sizing of wing structures that is based on representing the built-up finite element assembly of the structure by equivalent beam models. The reduced-order beam models are computationally less demanding in an optimum design environment which dictates repetitive analysis of several trial designs. The design procedure is implemented in a computer program that requires geometry and loading information to create the wing finite element model and its equivalent beam model, and provides a rapid estimate of the optimum weight obtained from a fully stressed design approach applied to the beam. The synthesis procedure is demonstrated for representative conventional-cantilever and joined wing configurations.

Optimization, Structures, Structural Analysis, Joined Wing

Unclassified

Unclassified

Unlimited - Unclassified

Subject Category - 01