AD-A094 770   AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 20/4
CALCULATION OF AIRLOADS FOR A FLEXIBLE WING VIA NASTRAN.(U)
DEC 80 L E CHRISINGER
UNCLASSIFIED AFIT/GAE/AA/80D-2 NL

1 OF 1
AD A
094770

END
DATE
FILMED
3-81
DTIC

14

AFIT/GAE/AA/80D-2

1 Dec 80

Accession For	
NTIS GRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	

Dist	Avail and/or Special
A	

9 Master's thesis

DTIC
ELECTE
FEB 10 1981
S D
F

CALCULATION OF AIRLOADS FOR A FLEXIBLE WING VIA NASTRAN.

THESIS

AFIT/GAE/AA/80D-2 10 Lance E. Chrisinger
Captain USAF

12 78

23 JAN 1981

Approved for public release; distribution unlimited

APPROVED FOR PUBLIC RELEASE AFR 190-17.

Laurel A. Lampela
LAUREL A. LAMPELA, 2Lt, USAF
Deputy Director, Public Affairs

Air Force Institute of Technology (ATC)
Wright-Patterson AFB, OH 45433

012[illegible]

81 2 09 095

AFIT/GAE/AA/80D-2

CALCULATION OF AIRLOADS FOR A FLEXIBLE WING VIA NASTRAN

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

Lance E. Chrisinger, B.S.A.E.
Captain USAF

Graduate Aerospace Engineering

December 1980

Approved for public release; distribution unlimited

Acknowledgments

The potential capability for using the NASTRAN program to calculate flexible wing airloads has previously existed. This report describes the process necessary in using the program for these calculations. I owe thanks to my advisor Captain Hugh C. Briggs for his guidance with this project. I also extend my appreciation to Mr. J. R. Johnson for his help with the NASTRAN program.

A very special thanks is extended to my wife, Donna, for her understanding, patience, and help, without which this work could not have been completed.

Finally, a special debt of gratitude is given to Charlotte Eschmann who typed this thesis.

Lance E. Chrisinger

List of Figures

List of Tables

List of Symbols

Symbols	Definition
A	Element area
AJJL	Aerodynamic influence matrix
AOAP	Angle-of-attack parameters
ASET	Analysis set
C_p	Coefficient of pressure
ΔC_p	Jump in coefficient of pressure across lifting surface
ΔC_{p_i}	Elemental vector of ΔC_p's at the element nodes
D1JK	Steady down wash matrix
DMAP	Direct matrix abstraction program
DZ	Attachment flexibility parameter
ELOOP	Loop convergence parameter
F_i	Consistent elemental load vector
GTAK	Surface spline interpolation matrix
L_i	Interpolation functions for three noded element
P	Pressure
PG	Global load vector
q	Dynamic pressure
QKKK	Vector of ΔC_p values for aerodynamic points
SASEL	Diagonal matrix of elemental areas
TTTTT	Load transformation matrix
X-Y-Z	Axis coordinates

Abstract

This thesis describes the use of the NASTRAN program (Level 17.0) in the calculation of flexible wing airloads and stresses. The problems of interfacing the aerodynamic and structural models are discussed with assumptions needed to solve them. Two different methods of transferring the aerodynamic forces into a structural load vector are presented.

A Direct Matrix Abstraction Program was developed for use in these calculations. This sequence is limited to the use of one doublet-lattice panel for the aerodynamic model and requires several new data blocks for execution. Although these data blocks were input with the bulk data deck to test the sequence, a new preliminary module has been constructed to build them internally.

The results of this investigation show that NASTRAN can be used in calculating the airloads and stresses for a flexible wing. This new sequence has extended the capability of NASTRAN to allow for the application of internally generated steady airloads to the structural model.

Contents

CALCULATION OF AIRLOADS FOR A FLEXIBLE WING VIA NASTRAN

I Introduction

The calculation of flexible aircraft airloads and structural stresses requires the interfacing of both structural and aerodynamic theories. The NASTRAN structural analysis program includes both of these types of theories. The purpose of this study was to demonstrate the feasibility of using this program for these calculations.

The NASTRAN program uses a finite element structural model. A wide variety of element types are available, thus allowing the construction of simple to very complex models. The aerodynamic theory chosen for this study uses doublet-lattice panels to model the lifting surfaces. Other aerodynamic theories are available for use in the NASTRAN program but were not considered.

A new instruction sequence known as a Direct Matrix Abstraction Program (DMAP) was developed for use within NASTRAN to calculate flexible aircraft airloads and structural stresses. This new DMAP sequence is limited to a structural model consisting of only a wing. The corresponding aerodynamic model is limited to a single aerodynamic panel to model the lifting surface. This new sequence extends the capability of the NASTRAN program by creating the ability to apply the internally generated steady airloads to the structural model. The potential to do this has existed since the inclusion of aerodynamic theories in the NASTRAN program.

The calculation of flexible wing airloads by this DMAP sequence is an iterative process. The initial airloads are calculated based upon a rigid wing shape at an angle-of-attack. These airloads are applied to the structural model and the deflections calculated. The deflected structure causes the airloads to change, which will in turn change the deflection of the structure. This cyclic process is continued until the incremental changes in the airloads and the displacements are negligible.

The wing model chosen to demonstrate the new DMAP sequence was first tested for solution convergence of the cyclic process. The test was performed using the rigid format for static analysis interfaced with the Woodward aerodynamic program. The results of this test were also used as an alternative solution to check the reasonableness of the solution generated by the new DMAP sequence.

The potential users for this new DMAP sequence are the Structures Branch of the Flight Dynamics Lab (AFFDL/FB) and San Antonio Air Logistic Center (SAALC). The SAALC has had a NASTRAN structural model of half of a T-37 aircraft since

1976. This model was intended to be used to calculate stresses developed by an aerodynamic loading of the model. A workable automated method of generating airloads for this model has not been found. The AFFDL/FB has developed two programs to generate loads for a wing structural model from airloads calculated by an aerodynamic program. Both organizations can use the new DMAP sequence.

II Background

A current computer program for calculating the loads on a flexible aircraft is Flexloads. This program includes several theories to calculate aerodynamic forces for velocities ranging from subsonic to supersonic. These aerodynamic theories are the Woodward Finite Element Subsonic-Supersonic Steady State Theory, Doublet-Lattice Steady and/or Unsteady Theory, Steady and/or Unsteady Modified Newtonian Theory, and finally Subsonic and Supersonic Kernal Function Theory. The structures are modeled in this program as slender beams or as built-up plates. As a result, Flexloads cannot use all the different elements available with NASTRAN.

The AFFDL/FB has developed two programs to generate equivalent loads for a NASTRAN wing model from airloads calculated by an aerodynamic program. These two programs are SPLOADS and BEAMPROG. These programs handle the interfacing problem of applying airloads to a structural model. These programs could be used as part of the iterative process necessary in the calculation of flexible aircraft airloads. However, another program would have to be developed to introduce structural deformation effects into the calculation of airloads before this process could be employed.

III Interfacing Problems

The problems of interfacing an aerodynamic wing model with its corresponding structural model are discussed in this section. The two models solve two different problems, so the mesh sizes used for the two models will be different, which causes many of the interfacing problems. The interfacing is required for the application of airloads to a structural model and for the introduction of structural deformation into the calculation of airloads. In the following these problems are discussed in general and as they specifically apply to the new DMAP sequence.

The structural models of most wings only include the main structural elements, the wing box. This means the planform of the aerodynamic model is larger than that of the structural model. Because of this difference, the aerodynamic model will generate loads for which the corresponding structural elements do not exist. These airloads have to be handled as special cases for translation and application to the structural model. The difference in planform sizes causes problems in the calculation of new airloads based on structural deformation. The structural deformation changes the local slope of an aerodynamic box which is used in the calculation of the down wash. However, the deformed structure necessary for this calculation does not exist in areas for the aerodynamic model lying outside the wing box. A simple solution to this difference in planform sizes is to force the two to be the same. As a result the structural model must include the secondary wing structure. Since this secondary wing structure only transfers the loads to the wing box, this structure does not have to be modeled accurately except in size of the planform.

Even when these planforms are the same, the aerodynamic forces have to be transformed into a load vector for the structural model. The doublet-lattice method used in NASTRAN generates a delta pressure coefficient ΔC_p which is the difference between the pressure on the upper and lower surfaces at the aerodynamic points for each box. For simplicity the new DMAP sequence applies the load to the upper surface of the structural model. Assumptions on the ΔC_p distribution are required to transform the aerodynamic forces into a load vector. Two methods based upon a different set of assumptions for the distribution have been implemented.

In the first method the ΔC_p value at the structural element centroid is found by a surface spline interpolation and then assumed constant over the element. Because ΔC_p is constant, multiplication by the dynamic pressure and the element area will yield the total load. For basically parallelogram shaped elements, one quarter of this total load is now distributed to each of the element's nodes. This is done for each membrane element on the upper surface, resulting in a global expression for the load vector.

In the second transformation method more variation in ΔC_p distribution is accounted for. After finding interpolated corner values for each element, a bilinear variation is assumed for the distribution over two triangular subregions. See Figure 1.

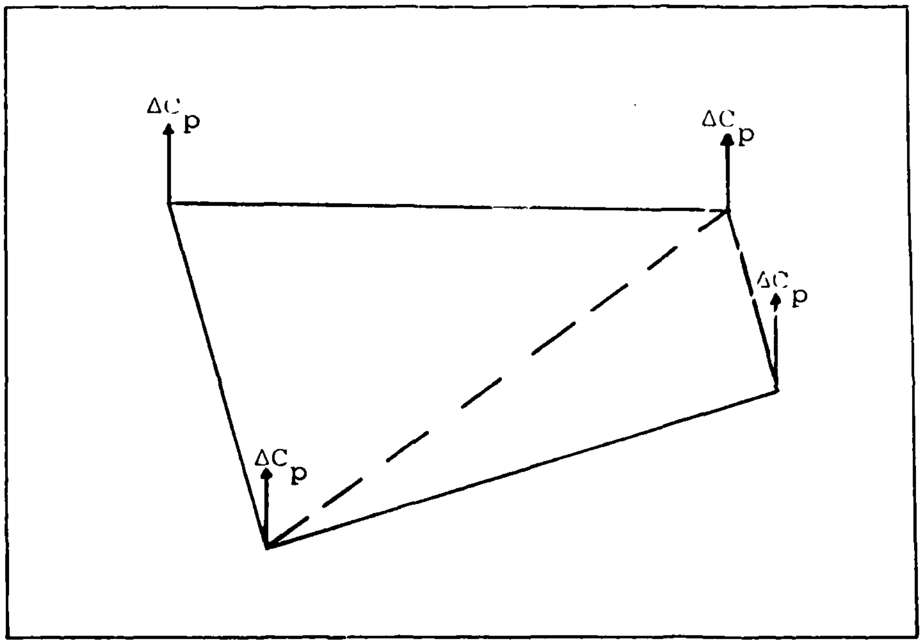

Figure 1. Division of the Upper Surface Quadralateral Membrane Elements into Triangular Elements.

The bilinear pressure over each of the triangular subelements is assumed to be

$$P(X, Y) = q\ L_i\ \Delta C_{p_i}$$

where q is the dynamic pressure, L_i are the shape functions for a three noded triangular element, and ΔC_{p_i} is a vector of nodal values of ΔC_p. The equivalent nodal forces are calculated as usual by:

$$F_i = \int_A PL_i dA = \int_A qL_j \, C_{p_i} L_i d_A$$

When integrated this becomes

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix} = \frac{qA}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \begin{Bmatrix} C_{p_1} \\ C_{p_2} \\ C_{p_3} \end{Bmatrix}$$

These subelement expressions are now assembled into a global expression for the load vector.

The doublet-lattice method calculates the aerodynamic forces normal to the panel. When the wing is at an angle-of-attack the orientation of the lift vector and the normal force vector will differ by this angle. The new DMAP sequence uses the normal force instead of lift in calculating the coefficients of lift, pitching moment and rolling moment. The resulting errors in these coefficients will be minimal, if small angles-of-attacks are used.

IV Procedures

An intermediate complexity wing structural model was obtained from AFFDL/FB to demonstrate the new DMAP sequence. A description of the model used is contained in Appendix B. The wing model was first tested by using the Woodward program USSAERO to generate the airloads used in the iterative process. This test checked the model for solution convergence and provided an alternative solution. The new DMAP sequence was then used to resolve the problem.

This section covers first the interfaced NASTRAN-USSAERO sequence and next the new DMAP sequence. The new sequence is explained relative to functions it performs, new data blocks it requires, and in how it would use the new module.

Interfaced NASTRAN-USSAERO Sequence

The static analysis rigid format of NASTRAN and USSAERO are interfaced in a manual execution of the iterative process. The output from USSAERO is used to create constant pressure loads on upper surface structural elements and then the displacements from NASTRAN are used to change the camber surface for USSAERO in this process.

The doublet-lattice method in NASTRAN is a lifting surface theory, therefore USSAERO which includes thickness effects approximates the thin sheet by using the planar option and the smallest allowable wing thickness. USSAERO generates values for pressure coefficient C_p at the center of each aerodynamic box for both the upper and lower surfaces. A further explanation of USSAERO is contained in Reference 8.

The mesh size for the aerodynamic model was chosen to avoid interface problems by exactly matching that for the structural model. By this matching the need to transform the airloads for structural application is avoided. The total elemental load now relates to the aerodynamic force produced by the corresponding box.

The boundary conditions for the static analysis were chosen to take out the rigid angle-of-attack displacements, leaving only total deformation in the looping process. To accomplish this, all degrees-of-freedom for the wing root grid points were fixed. For each loop pass, the new total deformation were calculated from the changing total airloads.

Two conversion programs were required to interface NASTRAN and USSAERO for the iterative process. Because of the matched meshes, these essentially just changed the output from one into the proper format for input into the other. The first program converts the ΔC_p to pressure loads. To do this the delta coefficient of pressure for each box is found and then multiplied by the dynamic pressure q. These pressure loads are used to create PLOAD2 cards for use in the NASTRAN bulk data deck. The second program converts the total deformation to a specification of a new camber surface for the aerodynamic model. The structural displacements are averaged to find a midsurface, which also defines the camber surface used by USSAERO. The position of these two programs in the iterative process is shown in Figure 2.

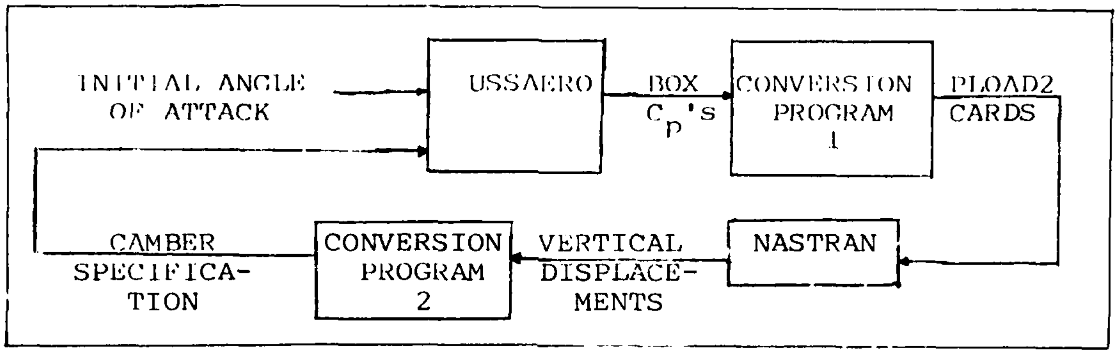

Figure 2. INTERFACED NASTRAN-USSAERO Sequence

This manual execution of the iterative process is inefficient for the following reasons. The equations have to be reformed with every loop pass when only the right hand side of the equations change. Next, the process generates large amounts of intermediate data that is not used. Also, the precision of the intermediate solutions are decreased, because the data passed from one program to the next must be truncated to specific formated field sizes. Finally, the process requires sequential jobs to be run with each pass, requiring excessive amounts of user time.

The DMAP Sequence

The new DMAP sequence calculates the airloads and structural stresses for a flexible wing model. The airloads used in this sequence are calculated by the doublet-lattice method in NASTRAN. By using a reduced frequency of approximately zero, this unsteady method reduces to a steady vortex-lattice method. Values of ΔC_p are produced for the center of the quarter-chord for each box. A further explanation of this aerodynamic method is found in References 2, 3 and 6.

The new DMAP sequence is explained by major functions in the following section. The total capability has previously existed for NASTRAN to calculate the flexible wing airloads, except for the ability to transform the steady airloads into a structural load vector. To test the new DMAP sequence, the data blocks necessary to build this transformation matrix were input directly with the bulk data deck. A discussion of these new data blocks follows the explanation of the sequence. A new module was designed that will build this transformation matrix internally. The functions of the new module are explained in the section following the data block descriptions.

Several restrictions are required on models used with the new DMAP sequence as it presently exists. The planforms for the aerodynamic and structural models are required to be the same. This is necessary, as discussed in Section III, so that the airloads can be applied to corresponding structural elements without consideration of special cases.

Next the aerodynamic model is restricted to one doublet-lattice panel to simplify bookkeeping. Finally, an analysis set must be specified that contains only the Z-direction degree-of-freedom for the wing grid points exclusive of the wing root. This limitation is required for correct calculation of the airloads.

Functions of the New DMAP Sequence. The new DMAP sequence will be described here in general terms by the major functions it performs. The complete listing for the DMAP sequence is contained in Appendix A. The sequence consists largely of a combination of rigid formats used for static analysis and flutter analysis. The sequence accomodates any of the structural element types available for use with the static analysis rigid format, but does not allow the use of the down wash factors due to extra points available in the flutter analysis rigid format.

The main functions of the new DMAP sequence will be presented in sequential order. The first portion of the sequence sets up the tables necessary to describe the structural model, including the element connection table (ECT) and the grid point list (GPL). These tables are used in the assembly of the unconstrained global stiffness and mass matrices for the system.

The constraints for subcase one are now applied to these global matrices. These constraints are constructed so as to leave only a rigid body pitch mode. The constrained matrices are then partitioned down to the analysis set specified by the ASET card.

A real eigenvalue analysis is performed on the analysis set to calculate the displacements due to this rigid pitch mode. The mass matrix is only used in this sequence for the eigenvalue analysis, therefore a lumped mass matrix is sufficiently accurate. This rigid pitch mode shape is now multiplied by an input parameter AOAP to specify displacements for an initial angle-of-attack. This procedure was necessary because the NASTRAN program does not have the capability to set the model at an initial angle-of-attack. The camber is now added to the displacements due to angle-of-attack.

The next function is the formation of the transformation matrix GTKA which transforms the displacements to values of slope at the aerodynamic points. These slopes are used to specify the strengths for the steady down wash vector D1JK. The aerodynamic influence matrix AJJL for the doublet-lattice method is calculated next. The inverse of this matrix when multiplied by the down wash vector, yields the values for ΔC_p at each of the aerodynamic points. These ΔC_p values are for the total pressure differential across the wing.

The next functions of the sequence are those the new module will perform when incorporated into NASTRAN. The transformation matrix TSTA is calculated to transform the vector of ΔC_p values to an unconstrained global load vector PG. This load vector will only contain non-zero forces for the Z-direction degree-of-freedom for the upper surface grid points. The matrices necessary for the calculation of the coefficients of lift, pitching moment, and rolling moment are also built in this section of the sequence.

The boundary conditions are now changed to those specified by subcase two, which constrains all the degrees-of-freedom for the wing root grid points. The matrix data blocks necessary for a static deflection calculation are setup, with this boundary condition. The displacements calculated will be from the midplane of the wing pitched at an angle-of-attack. These matrices are partitioned down to the same analysis set used to calculate the airloads. By doing this, the transformation matrices TSTA and GTKA previously calculated can be used in the iterative loop to calculate the flexible wing airloads.

The next function of the sequence is the loop to calculate the flexible wing's displacements, static load vector and airloads by the iterative process. This loop calculates an incremental displacement vector due to the initial load vector. This incremental displacement vector is then used to calculate an incremental airload vector which is transformed to an incremental load vector. The iterative process is continued until the norm of the incremental displacement vector is less than a prescribed input parameter (ELOOP). The incremental displacements, incremental load vectors and incremental airloads are summed with each pass through the loop. These sums give the final displacement of the analysis set, the final load vector and the final airloads for the flexible wing.

The next part of the sequence recovers the complete solution for the analysis set and the omitted degrees-of-freedom. From the complete solution, the stresses are now calculated for any elements specified in subcase two of the case control deck.

The coefficients of lift, pitching moment and rolling moment due to both the final airloads and the final load vector are calculated next. These coefficients are used as a check on errors induced when the airloads are transformed to a load vector. The final section of the sequence contains error messages called when a specific error is found during the execution of the sequence.

New Data Blocks and Parameters. The data blocks required by the new DMAP sequence were created and input directly with the bulk data deck. All of the input matrices except the camber vector will be created by the new module when it is incorporated into NASTRAN. Some extra steps were required in the sequence to process these input matrices and will be removed when the module is added.

The new parameters that are needed for this sequence are listed with a brief explanation in Table I. A list of the new input matrices for the sequence is given in Table II. The major function of these matrices is the transformation of the aerodynamic forces to the structural load vector PG.

This load vector can be created by one of the two methods discussed in Section III. The first method generates the transformation matrix by

$$[\mathrm{TSTA}] = q\,[\mathrm{TTTTT}]\,[\mathrm{SASEL}]\,[\mathrm{GTAKT}]^{T}$$

for use in

$$\{\mathrm{PG}\} = [\mathrm{TSTA}]\,\{\mathrm{QKKK}\}$$

which PG is the unconstrained load vector and QKKK is the vector of ΔC_p values at the aerodynamic points. The transformation matrix GTAK for this method interpolates the QKKK matrix to a vector of ΔC_p values at the structural element centroids.

The second method generates the transformation matrix TSTA in a slight different way. The matrix is now

$$[\mathrm{TSTA}] = q\,[\mathrm{TTSAS}]\,[\mathrm{GTAKT}]^{T}$$

where the matrix TTSAS is the assembled global interpolation matrix for a consistent load vector for triangular elements and the matrix GTAKT interpolates the same QKKK matrix used

Table I. List of New Parameters.

NAME	DESCRIPTIONS OF PARAMETERS
AOAP	Angle of attack parameter, used to set angle of attack; Complex (AOAP,0.0)
QC	Dynamic pressure, used for calculation of loads; Complex (Q,0.0)
Q	Dynamic pressure, used in calculation of aerodynamic coefficients; Real
RC	Reference chord length, used in calculation of aerodynamic coefficients; Real
WAREA	Planform area of the structural model; Real
ICAMB	Flag for camber vector, only input if camber to be specified; Integer >0
ELOOP	Used to specify convergence criteria for the norm of the incremental displacements Real >0 or as a flag to specify only rigid ΔCp calculations Real <0

Table II. List of New Matrices

NAME	DESCRIPTIONS OF MATRICES
GTAKT	Surface spline interpolation matrix for transformation of ΔC_p's at the aero points to ΔC_p's at structural points; real, general
SASEL	Contains areas ofthe surface membranes to which loads are applied; real, diagonal
TTTTT	Transformation matrix to distribute 1/4 of the load on the membranes to their nodes; real, general
TTSAS	Assembled global interpolation matrix for consistent load vector, triangular elements, replaces SASEL and TTTTT for linear pressure distribution; real, general
CAMBER	Contains camber surface displacements for the analysis set grid points; real, vector
SAS	A matrix containing one's where areas are in SKJ matrix, imaginary part zero; complex, general
AIDMIT	Unit vector of length equal to number of aerodynamic boxes; real, vector
VGLS	A vector with ones in the proper positions, so that when multiplied by PG it sums the loads; real, vector
VGMS	A matrix with ones in the proper positions, so that when multiplied by PG it creates a vector of the non-zero loads; real, general
VTMA	Contains the moment arms for the aerodynamic pitching moment about the X-axis; real, vector

Table II. List of New Matrices (Cont.)

NAME	DESCRIPTIONS OF MATRICES
VTRA	Contains the moment arms for the aerodynamic rolling moment about the Y-axis; real, vector
VTM	Contains the moment arms for the pitching moment due to the transformed loads about the Y-axis; real, vector
VTR	Contains the moment arms for the rolling moment due to the transformed loads about the X-axis; real, vector

in the first method to a vector of nodal ΔC_p values. The equation for the unconstrained consistent load vector is

$$\{PG\} = [TSTA] \{QKKK\}$$

where PG and QKKK are defined the same as in the first method.

Formation of the interpolation matrix GTAKT is the same for both methods. The NASTRAN subroutines source decks used in the surface spline interpolation were used to build this matrix in a temporary external program. By using the same subroutines the transposed interpolation matrix will be the same as the one generated by the module when incorporated. Since the number of significant figures is limited to the size of the data fields used for input, the large field bulk data cards should be used.

The camber vector listed on Table II is used to specify the camber surface of the wing, when required. This vector contains the Z-direction camber displacements for the upper surface grid points in the analysis set. Zeros are entered for the lower surface grid points. In addition, if the wing is twisted this is added into the camber vector in a similar manner.

Module Description. The new module when incorporated into the NASTRAN program will be designated by the name GIAS. This module was not incorporated into the NASTRAN program because the link-editor routine for the program was not functional. This module will build internally all of the matrices listed in Table II, except for the camber vector. The module will be called in the new DMAP sequence prior to the calculation of the initial ΔC_p values. The exact position of this new module and the steps it will replace in the sequence are shown in Appendix A. The fortran source program for this module is contained in Appendix C. This source program has not been tested with NASTRAN, but it should function correctly when incorporated.

The following restrictions are required when using this module. Since only the upper surfaces is loaded, the structural model is restricted to CQMEM2 membrane elements on that surface. Also, the aerodynamic model is restricted to one doublet-lattice panel which must lay in the X-Y plane.

The DMAP calling sequence for this module will be

```
GIAS    ACPT,TOSEL,ECTA,GPLA,BGPA/SAS,
        GTAK,TTTTT,SASEL,VGLS,VGMS,VTMA,
        VTRA,VTM,VTR,AIDMIT/$
```

The five data blocks before the first slash mark are the required input tables for this module. These input tables are listed in Table III with a short description. The eleven output data blocks listed between the slash marks are the same matrices described in Table II.

Table III. Input Tables for the Module

NAME	DESCRIPTION
ACPT	Aerodynamic connection property table contains the value for the total number of aero boxes.
TOSEL	Sorted list of CQMEM2 elements to which loads will be applied, in ascending order.
ECTA	Element connection table for both structural elements and aero boxes.
GPLA	Grid point numbers listed in order of appearance in the BGPA table.
BGPA	List of coordinates for both structural grid points and grid points of aero boxes.

The TOSEL table is specifically required by the new module and must be input with the bulk data deck. This table lists the CQMEM2 element identification numbers for the elements on the upper surface. These are the elements the aerodynamic force will be applied to when transformed to a load vector. These numbers must be listed in order from lowest to highest.

This module builds the data blocks necessary for load application by method one. For this method, the interpolation matrix GTAK is built by a surface spline through the aerodynamic points located at the center of the quarter chord of each box and the centroids of the membrane elements. This module also builds the data blocks necessary for the calculation of the aerodynamic coefficients. All of these data blocks are built with the origin as the reference point for calculating the coefficients.

V Results

The new DMAP sequence was successful in calculating the airloads and stresses for a flexible wing. Several sample cases were run and are presented in this section. Results were obtained for two different aerodynamic models and for two methods of airload transformation. These are compared with the solution obtained by the NASTRAN-USSAERO sequence. Comparisons of rigid wing ΔC_p distribution, errors caused by the assumption used for load transformation, wing tip deflections, and the convergence of the iterative processes are presented. A summary of the cases compared are contained in Table IV.

The doublet-lattice panel is an unsteady thin airfoil aerodynamic model, that for zero frequency reduces to a steady vortex lattice panel. The new DMAP sequence only uses the steady part of this panel, so this part was checked against USSAERO for accuracy. The chordwise ΔC_p values calculated for a rigid wing are shown in Figure 3. The steady part calculates lower values of ΔC_p near the leading edge and slightly higher near the trailing edge. The ΔC_p distribution for the steady part is close to that for USSAERO and is thus acceptable for use.

The chordwise ΔC_p distributions assumed for load transformation are compared with the aerodynamic distribution on

Table IV. Program Set-Ups.

CASE NUMBER	TYPE OF SEQUENCE USED	AERODYNAMIC MODELS USED*			ASSUMPTION USED FOR AIRLOAD APPLICATION	MACH NUMBER	ANGLE OF ATTACK (DEGREES)
		NUMBER OF AERODYNAMIC BOXES					
		CHORDWISE	SPANWISE	TOTAL			
1	NASTRAN-USSAERO	4	5	20	ΔC_p at center constant over membrane element	.2	2
2	New DMAP	4	5	20	ΔC_p at center constant over membrane element	.2	2
3	New DMAP	7	9	63	ΔC_p at center constant over membrane element	.2	2
4	New DMAP	7	9	63	Bi-linear ΔC_p distribution over membrane element	.2	2

*Appendix B contains figures showing division points used to define the size of boxes relative to the structural model.

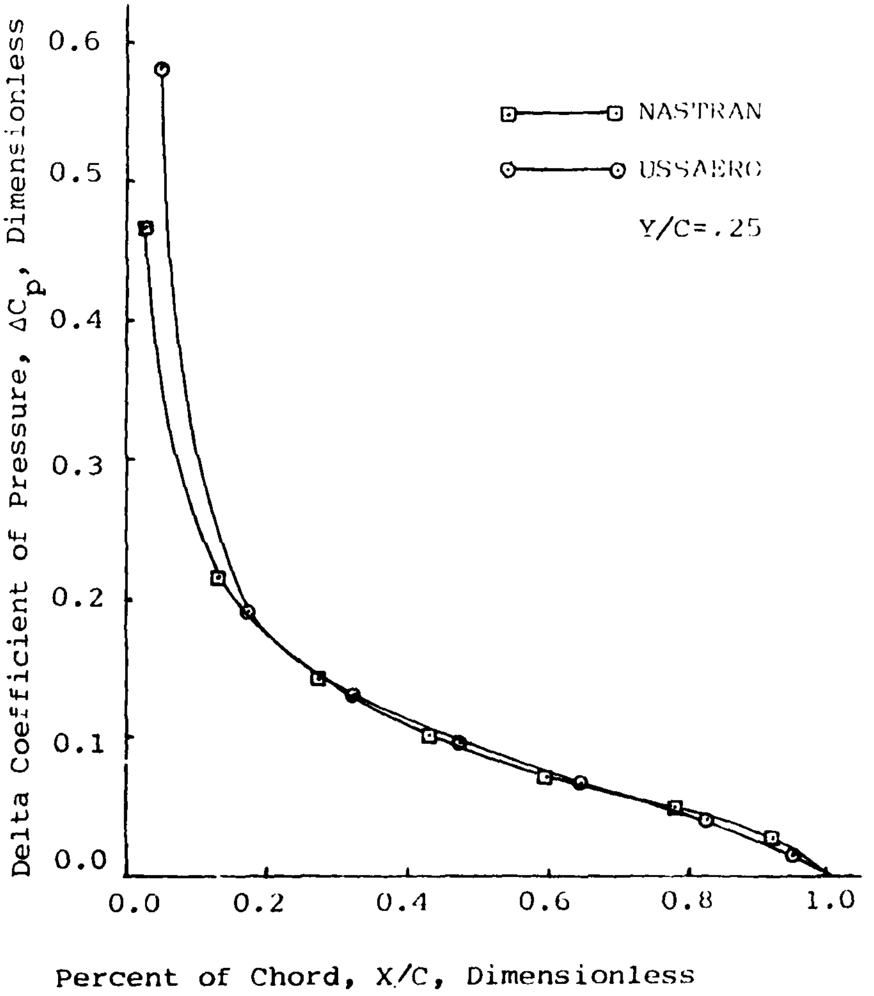

Figure 3. Rigid Wing ΔC_p Comparison.

Figure 4. The constant distribution assumed for the equivalent loading in the first three cases caused the load to be shifted aft as shown for case 3 on this figure. The linear distribution assumed for the fourth case performs significantly better in this respect. Similar changes in the spanwise ΔC_p distributions are noted in Figure 5. The linear distribution for the spanwise direction transferred higher loads to the coarsely meshed tip region due to extrapolation.

This redistribution of pressure over the wing causes errors in the transformed coefficients of lift, pitching moment and rolling moment. To illustrate this, these coefficients were calculated for the four cases for both aerodynamic and transformed forces and are shown in Table V. Included is the percentage difference induced by the load transformation. The linear pressure distribution induced the smallest errors for the load transformation.

The final deflected shape of the tip chords due to the flexible wing airloads are plotted for all the cases on Figure 6. All cases showed a similar chord shape, but the total displacements due to spanwise bending is different. This is caused by the combination of aerodynamic mesh size and load transformation method, which created varied wing center of pressure locations. It can be seen that, for this model, the elasticity effects are related to twist and camber changes.

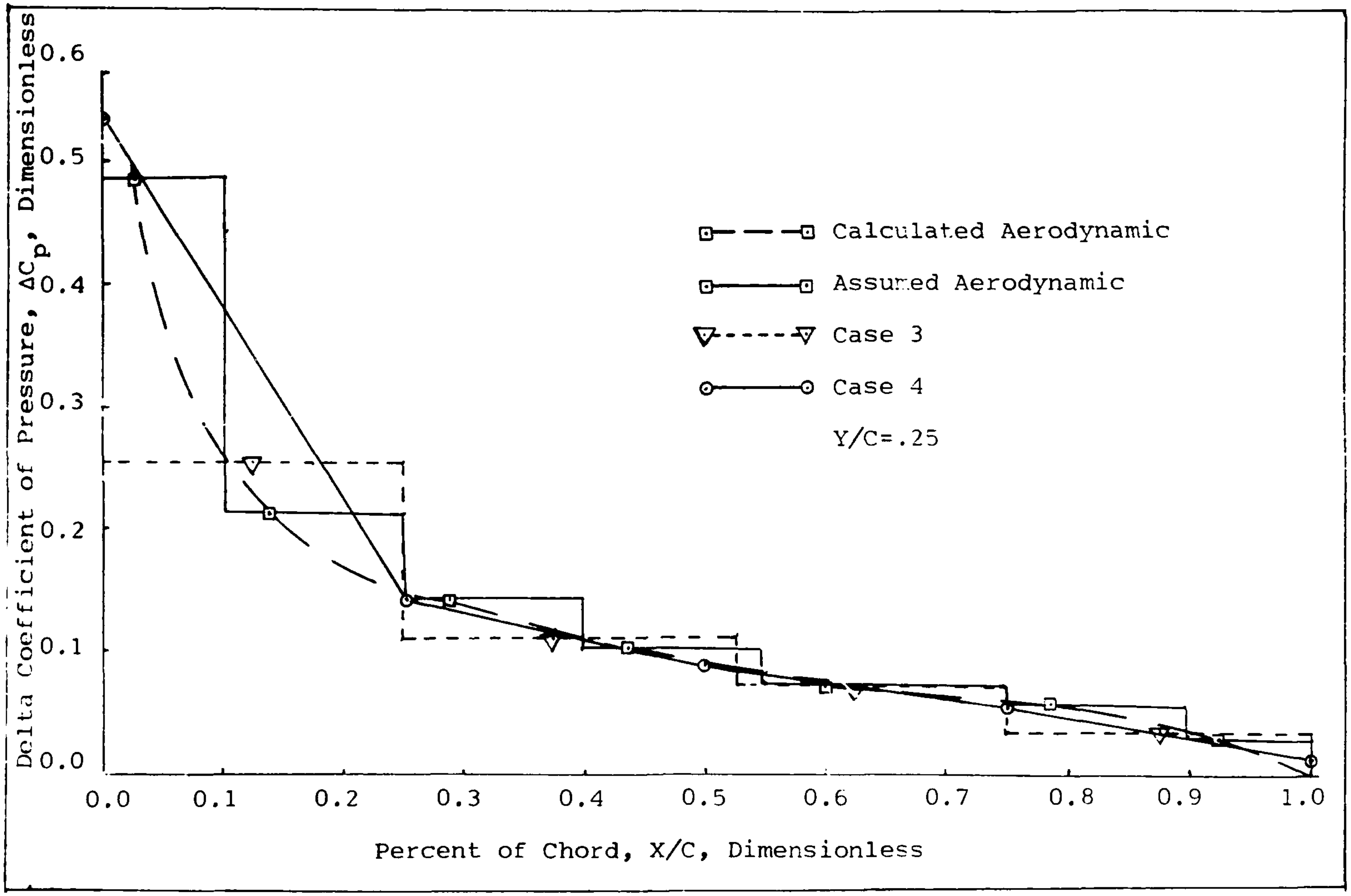

Figure 4. Chordwise Comparison of ΔC_p Distributions.

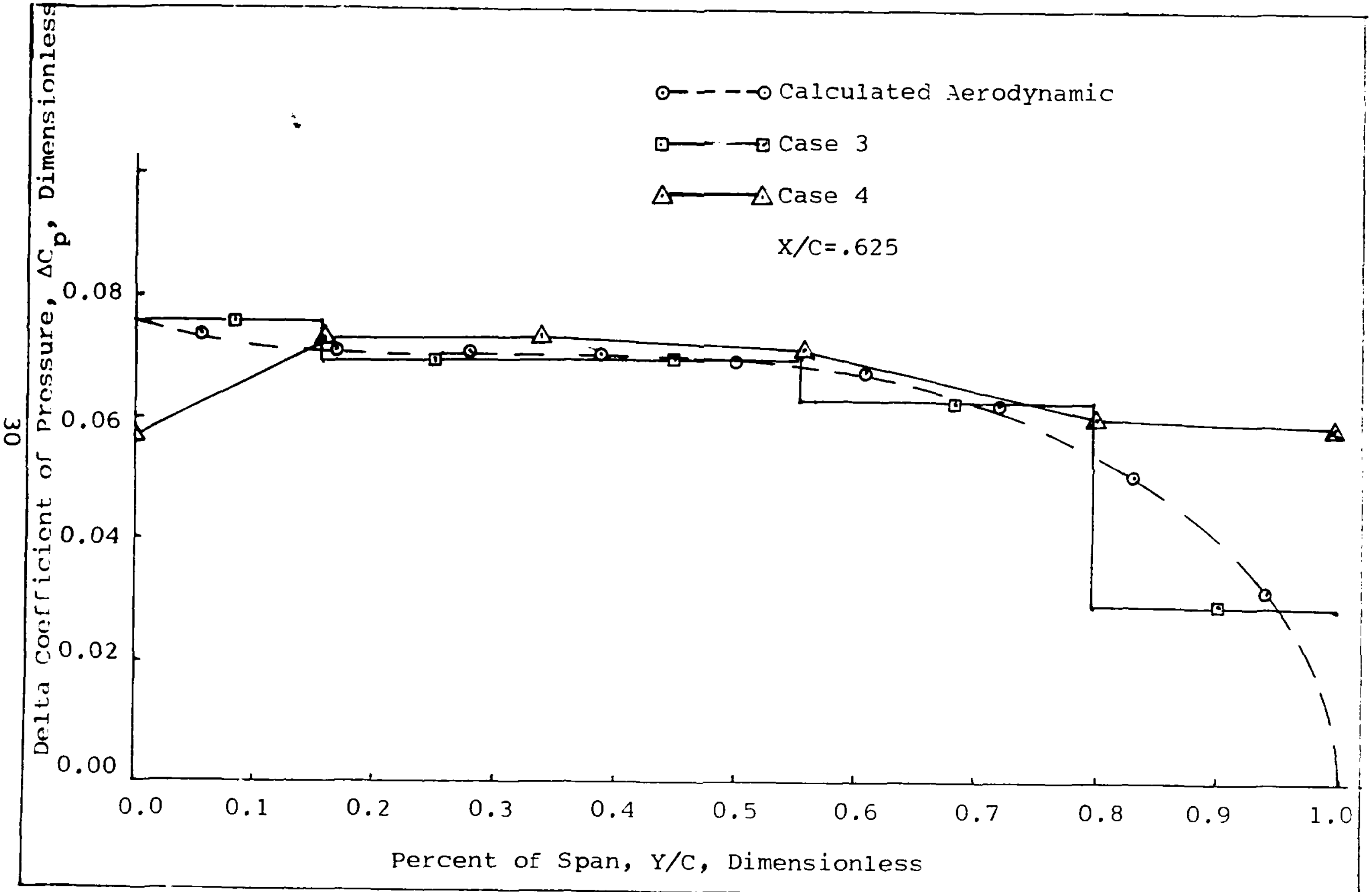

Figure 5. Spanwise Comparison of ΔC_p Distributions.

Table V. Comparison of Aerodynamic Coefficient

	COEFFICIENTS*	CASE** 1	CASE 2	CASE 3	CASE 4
COEFFICIENTS DUE TO FLEXIBLE AIRLOADS	C_L	.13764	.131823	.13246	.13625
	C_M	-.10995	-.103377	-.1027	-.10596
	C_R		.136603	.13508	.13932
COEFFICIENTS DUE TO TRANSFORMED LOADS	C_L		.11389	.11328	.14424
	C_M		-.09636	-.09457	-.11191
	C_R		.11845	.11469	.14161
*PERCENTAGE**** ERROR IN COEFFICIENTS DUE TO LOAD TRANSFORMATION	ERROR IN C_L		-15.74	-16.94	+ 5.54
	ERROR IN C_M		- 7.28	- 8.60	+ 5.32
	ERROR IN C_R		-15.07	-17.58	+ 1.62

*All the moments are calculated about the origin.

**Transformed loads were not available for the sequence used for this case.

***For cases 2, 3 and 4 the percentage error in the coefficients was calculated as follows

$$\text{ERROR} = \frac{\text{TRANSFORMED} - \text{AERODYNAMIC}}{\text{TRANSFORMED}} \times 100\%$$

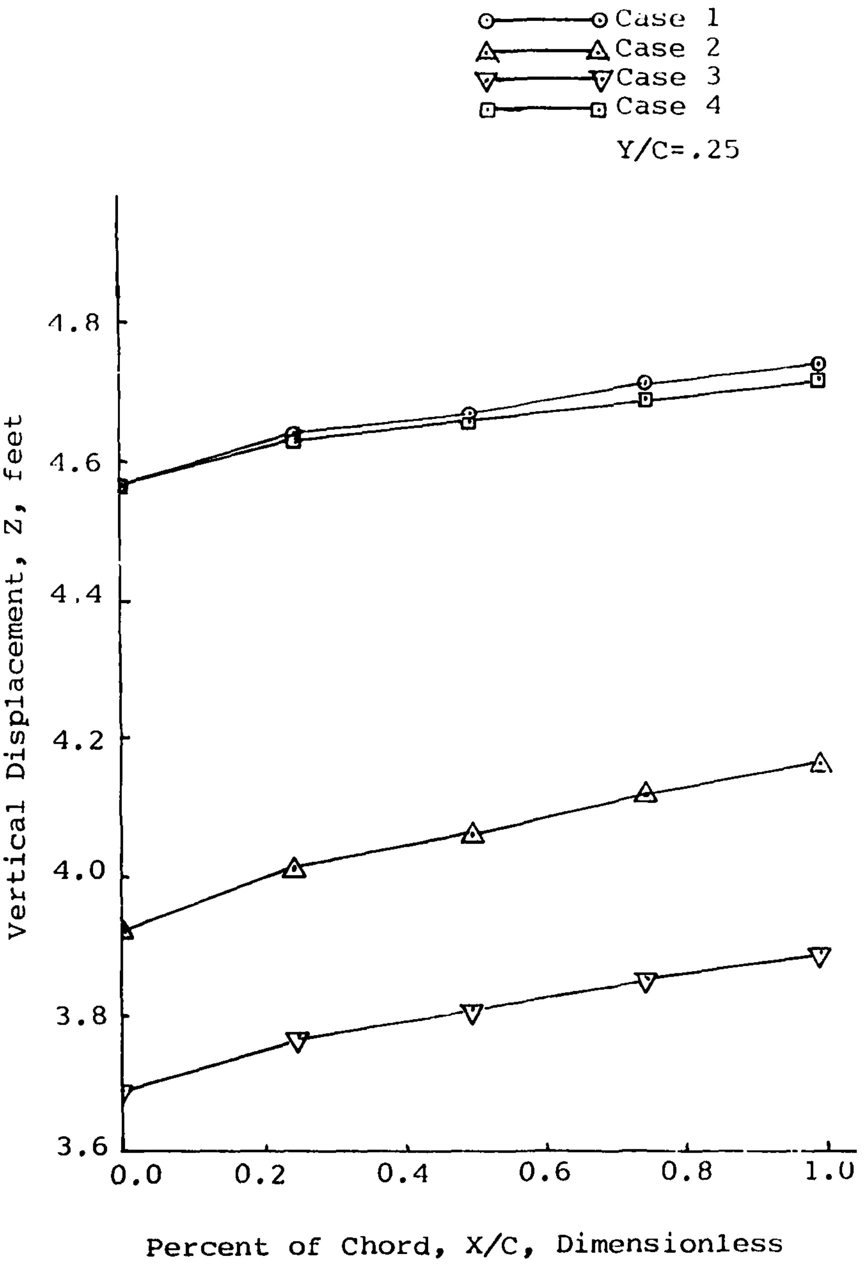

Figure 6. Vertical Wing Tip Deflections.

The solution convergence for the four cases is illustrated in Table VI. The new DMAP sequence used the norm of the incremental displacement vector as the convergence check, while the change in lift coefficient was used for the interfaced NASTRAN-USSAERO sequence. The iterative process converged slightly faster for the finer aerodynamic mesh used for cases three and four. The bilinear pressure distribution of case three. Each loop pass in the iterative process used in the new DMAP sequence required approximately twenty seconds of central processor time and forty seconds of input/output time for this particular model.

Table VI. Convergence of the Four Cases.

LOOP	CASE 1 C_L	CASE 2* NORM**	CASE 3* NORM**	CASE 4* NORM**
1	.14192	18.2335	16.428	20.069
2	.13853	1.627	.74882	.92205
3	.13668	.0788	.10017	.09327
4	.13799	.0289	.01048	.0092
5	.13673	.0003264	.00118	.000837
6	.13802	.0004375	.000128	.000094
7	.13762	.000031		
8	.13764			
MAXIMUM CHANGE IN z DISPLACEMENT FOR LAST PASS		$7.81(10^{-6})$	$-3.184(10^{-6})$	$-2.347(10^{-6})$

*ELOOP = 0.0002 for cases 2, 3 and 4.
**The norm shown for each loop are the incremental values.

VI Discussion and Recommendations

The new DMAP sequence has shown that with some refinements it can be a very useful method of calculating flexible aircraft airloads. The sequence could be modified to allow restart capability for large jobs and also modified to allow solutions for multiple angles-of-attack. The methods of transforming the airloads could be improved for quadralateral plate elements by using a bilinear ΔC_p distribution over the element. This would allow for the calculation of consistent moments in the load vector and would better represent the actual airloads.

Having a module in the program to set upthe required matrices will make the sequence easier to use, the data blocks will not have to be created separately and then input with the bulk data. This module could be set up to use any of the load transformation methods depending on the requirements and the model used. The module could also be used to set the initial angle-of-attack, and would thus eliminate the need for a real eigenvalue analysis.

The surface spline interpolation used in these transformations needs to be looked at to set how the aerodynamic mesh can be specified to make the interpolation of ΔC_p distribution more accurate. The linear attachment flexibility parameter DZ should be checked also, relaxation of this parameter should smooth the interpolated surface.

The new sequence currently uses the doublet-lattice aerodynamic method. Although this method can be used for transonic speeds, the results produced are not accurate. The other aerodynamic methods and the interference bodies need to be looked at, to see if they can be used in this sequence. This would allow for subsonic modeling of the whole airplace and the use of wing models for supersonic speeds.

Bibliography

1. AFWAL-TR-80-3036, Vol. II. Flexible Airframe Design Loads Update for Wing/Body Blending and Lifting Body Concepts, Fort Worth, TX: General Dynamics, March 1980.

2. AFFDL-TR-71-5, Part II. Subsonic Unsteady Aerodynamics for General Configurations, Long Beach, CA: Douglas Aircraft Co., Aircraft Division, April 1972.

3. Albano, Edward and William P. Rodden. "A Doublet-Lattice Method for Calculating Lift Distributions on Oscillating Surfaces in Subsonic Flows", AIAA Journal, Vol. 7, pp. 279-285 (February 1969).

4. NASA SP-223(04). The NASTRAN Programer's Manual, Level 17.0. Washington, D.C.: Scientific and Technical Information Division, National Aeronautics and Space Administration, December 1977.

5. NASA SP-222(04). The NASTRAN User's Manual, Level 17.0. Washington, D.C.: Scientific and Technical Information Division, National Aeronautics and Space Administration, December 1977.

6. NASA SP-221(04). The NASTRAN Theoretical Manual, Level 17.0. Washington, D.C.: Scientific and Technical Information Division, National Aeronautics and Space Administration, December 1977.

7. NASTRAN System Programming. Silver Springs, MD: MSN Software Services, 1978.

8. Woodward, F.A. An Improved Method for the Aerodynamic Analysis of Wing-Body-Tail Configurations in Subsonic and Supersonic Flow., Bellevue, WA: Analytical Methods, Inc., May 1973.

APPENDIX A

List of the New DMAP Sequence

STATMENT SEQUENCE FOR A FLEXIBLE WING MODEL

```
        LEVEL 2.0 NASTRAN DMAP COMPILER - SOURCE LISTING


OPTIONS IN EFFECT:  GO       ERR=2   NOLIST  NODECK  NOREF   NOOSCAR

     1  BEGIN     $

     2  $*************************************************************************

     2  $

     2  $          THIS SECTIO COMMON TO RIGID FORMATS 1 AND 10

     2  $

     2  GP1       GEOM1,GEOM2,/GPL,EQEXIN,GPDT,CSTM,BGPDT,SIL/V,N,LUSET/  V,N,
                  NOGPDT $

     3  SAVE      LUSET,NOGPDT $

     4  COND      ERROR1,NOGPDT $

     5  GP2       GEOM2,EQEXIN/ECT $

     6  PARAML    PCDB//C,N,PRES/C,N,/C,N,/C,N,/V,N,JUMPPLOT $

     7  GP3       GEOM3,EQEXIN,GEOM2/,GPTT/V,N,NOGRAV $

     8  TA1       ECT,EPT,BGPDT,SIL,GPTT,CSTM/EST,GEI,GPECT,/V,N,LUSET/   V,N,
                  NOSIMP/C,N,1/V,N,NOGENL/V,N,GENEL $

     9  SAVE      NOGENL,NOSIMP,GENEL $

    10  COND      ERROR1,NOSIMP $

    11  PARAM     //C,N,ADD/V,N,NOKGGX/C,N,1/C,N,0 $

    12  PARAM     //C,N,ADD/V,N,NOMGG /C,N,1/C,N,0 $

    13  EMG       EST,CSTM,MPT,DIT,GEOM2,/KELM,KDICT,MELN,MDICT,,/V,N,NOKGGX/ V,
                  N,NOMGG/C,N,/C,N,/C,N,/C,Y,COUPMASS/C,Y,CPBAR/C,Y,CPROD/  C,Y,
                  CPQUAD1/C,Y,CPQUAD2/C,Y,CPTRIA1/C,Y,CPTRIA2/C,Y,CPTUBE/ C,Y,
                  CPQDPLT/C,Y,CPTRPLT/C,Y,CPTRBSC $

    14  SAVE      NOKGGX,NOMGG $

    15  COND      JMPKGGX,NOKGGX $

    16  EMA       GPECT,KDICT,KELM/KGGX,GPST $

    17  LABEL     JMPKGGX $
```

STATMENT SEQUENCE FOR A FLEXIBLE WING MODEL

```
    LEVEL 2.0 NASTRAN DMAP COMPILER - SOURCE LISTING


18  COND      ERROR1,NOMGG $

19  EMA       GPECT,MDICT,MELM/MGG,/C,N,-1/C,Y,WTMASS=1.0 $

20  COND      LGPWG,GRDPNT $

21  GPWG      BGPDT,CSTM,EQEXIN,MGG/OGPWG/V,Y,GRDPNT=-1/C,Y,WTMASS $

22  OFP       OGPWG,,,,,//V,N,CARDNO $

23  LABEL     LGPWG $

24  EQUIV     KGGX,KGG/NOGENL $

25  COND      LBL11,NOGENL $

26  SMA3      GEI,/KGGY/V,N,LUSET/V,N,NOGENL/C,N,-1 $

27  ADD       KGGX,KGGY/KGG $

28  LABEL     LBL11 $

29  PARAM     //C,N,MPY/V,N,NSKIP/C,N,0/C,N,0 $

30  PARAM     //C,N,SUB/V,N,DESINT/C,N,0/C,N,1 $

31  PURGE     BBBBB,D1JE,D2JE/DESINT $

32  PARAM     //C,N,MPY/V,N,NOCAMBER/C,N,1/C,Y,ICAMB=-1 $

33  PURGE     CAMBER/NOCAMBER $

34  PARAM     //C,N,ADD/V,N,TRANS/C,N,1/C,N,0 $

35  JUMP      SETUP $

36  LABEL     SETUP $

37  COND      LBL4B,TRANS $

38  $********************************************************* ************

38  $

38  $         THIS SECTIO COMMON TO RIGID FORMATS 10

38  $

38  GP4       CASECC,GEOM4,EQEXIN,GPDT,BGPDT,CSTM/RG,,USET,ASET/V,  N,LUSET/
              V,N,MPCF1/V,N,MPCF2/V,N,SINGLE/V,N,OMIT/V,N,REACT/V,N,NSKIP/V,
```

STATMENT SEQUENCE FOR A FLEXIBLE WING MODEL

```
     LEVEL 2.0 NASTRAN DMAP COMPILER - SOURCE LISTING

                      N,REPEAT/V,N,NOSET/V,N,NOL/V,N,NOA/C,Y,SUBID $
 39  SAVE             MPCF1,MPCF2,SINGLE,OMIT,REACT,NSKIP,REPEAT,NOSET,NOL,NOA $
 40  GPSP             GPL,GPST,USET,SIL/OGPST/V,N,NOGPST $
 41  SAVE             NOGPST $
 42  COND             LBL4,NOGPST $
 43  OFP              OGPST,,,,,//V,N,CARDNO $
 44  LABEL            LBL4 $
 45  EQUIV            KGG,KNN/MPCF1/MGG,MNN/MPCF1 $
 46  COND             LBL2,MPCF1 $
 47  MCE1             USET,RG/GM $
 48  MCE2             USET,GM,KGG,MGG,,/KNN,MNN,, $
 49  LABEL            LBL2 $
 50  EQUIV            KNN,KFF/SINGLE/MNN,MFF/SINGLE $
 51  COND             LBL3,SINGLE $
 52  SCE1             USET,KNN,MNN,,/KFF,KFS,,MFF,, $
 53  LABEL            LBL3 $
 54  EQUIV            KFF,KAA/OMIT/ MFF,MAA/OMIT $
 55  PURGE            GO/OMIT $
 56  COND             LBL5,OMIT $
 57  PARAM            //C,N,PREC/V,N,PREC $
 58  VEC              USET/V/C,N,F/C,N,O/C,N,A $
 59  PARTN            KFF,V,/KOO,,KOA,KAAB $
 60  DECOMP           KOO/LOO,UOO/C,N,1/C,N,0/V,N,MIND/V,N,DET/V,N,NDET/V,N,SING $
 61  SAVE             MIND,DET,NDET,SING $
 62  FBS              LOO,UOO,KOA/GO/C,N,1/C,N,-1/V,N,PREC/V,N,PREC $
```

STATMENT SEQUENCE FOR A FLEXIBLE WING MODEL

LEVEL 2.0 NASTRAN DMAP COMPILER - SOURCE LISTING

```
 63  MPYAD      KOA,GO,KAAB/KAA/C,N,1/C,N,1/C,N,1/V,N,PREC $

 64  SMP2       USET,GO,MFF/MAA $

 65  LABEL      LBL5 $

 66  COND       LBL6,REACT $

 67  RBMG1      USET,KAA,MAA/KLL,KLR,KRR,MLL,MLR,MRR $

 68  RBMG2      KLL/LLL/ $

 69  RBMG3      LLL,KLR,KRR/DM $

 70  RBMG4      DM,MLL,MLR,MRR/MR $

 71  LABEL      LBL6 $

 72  DPD        DYNAMICS,GPL,SIL,USET/GPLD,SILD,USETD,TFPOOL,,,,,EED,EQDYN/V,
                N,LUSET/V,N,LUSETD/V,N,NOTFL/V,N,NODLT/V,N,NOPSDL/V,N,NOFRL/V,
                N,NONLFT/V,N,NOTRL/V,N,NOEED/C,N,123/V,N,NOUE $

 73  SAVE       LUSETD,NOUE,NOEED $

 74  COND       ERROR2,NOEED $

 75  EQUIV      GO,GOD/NOUE/GM,GMD/NOUE $

 76  READ       KAA,MAA,MR,DM,EED,USET,CASECC/LAMA,PHIA,MI,OEIGS/C,N,MODES/V,N,
                NEIGV $

 77  SAVE       NEIGV $

 78  OFP        OEIGS,LAMA,,,,//V,N,CARDNO $

 79  COND       ERROR4,NEIGV $

 80  MTRXIN     CASECC,MATPOOL,EQDYN,,TFPOOL/K2PP,M2PP,B2PP/V,N,LUSETD/V,N,
                NOK2PP/V,N,NOM2PP/V,N,NOB2PP $

 81  SAVE       NOK2PP,NOM2PP,NOB2PP $

 82  EQUIV      M2PP,M2DD/NOA/B2PP,B2DD/NOA/K2PP,K2DD/NOA $

 83  GKAD       USETD,GM,GO,,,,,K2PP,M2PP,B2PP/,,,GMD,GOD,K2DD,M2DD,B2DD/C,N,
                CMPLEV/C,N,DISP/C,N,MODAL/C,N,0.0/C,N,0.0/C,N,0.0/V,N,NOK2PP/V,
                N,NOM2PP/V,N,NOB2PP/V,N,MPCF1/V,N,SINGLE/V,N,OMIT/V,N,NOUE/C,
                N,-1/C,N,-1/ C,N,-1/C,N,-1 $

 84  GKAM       USETD,PHIA,,LAMA,DIT,M2DD,B2DD,K2DD,CASECC/MHH,BHH,KHH,AAAAA/V,
```

STATMENT SEQUENCE FOR A FLEXIBLE WING MODEL

```
     LEVEL 2.0 NASTRAN DMAP COMPILER - SOURCE LISTING

                   N,NOUE/C,Y,LMODES=0/C,Y,LFREQ=0./C,Y,HFREQ=0./V,N,  NOM2PP/V,N,
                   NOB2PP/V,N,NOK2PP/V,N,NONCUP/V,N,FMODE/C,Y,KDAMP $

 85  SAVE          NONCUP,FMODE

 86  ADD           AAAAA,CAMBER/PHIDH/C,Y,AOAP=(1.0,0.0)/C,Y,BETA=(1.0,0.0) $

 87  APD           EDT,EQDYN,ECT,BGPDT,SILD,USETD,CSTM,GPLD/EQAERO,ECTA,BGPA,SILA,
                   USETA,SPLINE,AERO,ACPT,FLIST,CSTMA,GPLA,SILGA/V,N,NK/V,N,NJ/V,
                   N,LUSETA/V,N,BOV $

 88  SAVE          NK,NJ,LUSETA,BOV $

 89  PARAM         //C,N,MPY/V,N,PFILE/C,N,0/C,N,1 $

 90  COND          SKPPLT,JUMPPLOT $

 91  PARAM         //C,N,MPY/V,N,PLTFLG/C,N,0/C,N,1 $

 92  PLTSET        PCDB,EQAERO,ECTA/PLTSETA,PLTPARA,GPSETSA,ELSETSA/V,N,NSIL1/V,N,
                   JUMPPLOT $

 93  SAVE          NSIL1,JUMPPLOT $

 94  PRTMSG        PLTSETA // $

 95  COND          SKPPLT,JUMPPLOT $

 96  PLOT          PLTPARA,GPSETSA,ELSETSA,CASECC,BGPA,EQAERO, ,,,,/PLOTX2/V,N,
                   NSIL1/V,N,LUSETA/V,N,JUMPPLOT/V,N,PLTFLG/V,N,PFILE $

 97  SAVE          PFILF,JUMPPLOT,PLTFLG $

 98  PRTMSG        PLOTX2 // $

 99  LABEL         SKPPLT $

100  COND          ERROR2,NOEED $

101  GI            SPLINE,USET ,CSTMA,BGPA,SIL , ,GM,GO/GKA/V,N,NK/V,N,LUSET $

102  PARAM         //C,N,ADD/V,N,DESTRY/C,N,0/C,N,1/ $

103  AMG           AERO,ACPT/AJJL,SKJ,D1JK,D2JK/V,N,NK/V,N,NJ/V,N,DESTRY/ $

104  SAVE          DESTRY $

105  PARAM         //C,N,SUB/V,N,IWEO/C,N,0/C,N,1 $

106  $
```

```
STATMENT SEQUENCE FOR A FLEXIBLE WING MODEL


     LEVEL 2.0 NASTRAN DMAP COMPILER - SOURCE LISTING


106  $
106  $************************************************************
106  $
106  $          THE MODULE WILL REPLACE THE FOLLOWING STATMENTS
106  $
106  $          THE DMAP INSTRUCTION FOR THE MODULE WILL BE
106  $
106  $   GIAS      ACPT,TOSEL,ECTA,GPLA,BGPA/SAS,GTAK,VGLS,VGMS,VTMA,VTRA,
106  $             VTM,VTR,AIDMIT/$
106  $
106  EQUIV     EEE,SAS/THEO $
107  TRNSP     GTAKT/GTAK $
108  $
108  $          THIS ENDS THE SECTION TO BE REPLACED BY THE MODULE
108  $
108  $************************************************************
108  $
108  EQUIV     DLJK,CPPM/DESINT $
109  ADD       SAS,CPPM/TMA/C,Y,ALPHA=(1.0,0.0)/C,Y,BETA=(-1.0,0.0) $
110  MPYAD     TMA,SKJ,/TTAA/C,N,1/C,N,1/C,N,0/C,N,1 $
111  MPYAD     TTTTT,SASEL,/TTSAS/C,N,0/C,N,1/C,N,0/C,N,1 $
112  MPYAD     TTSAS,GTAK,/TSTAP/C,N,0/C,N,1/C,N,0/C,N,1 $
113  ADD       TSTAP,BBBBB/TSTA/C,Y,QC/C,Y,BETA=(0.0,0.0) $
114  MPYAD     AIDMT,TTAA,/ATMA/C,N,1/C,N,1/C,N,0/C,N,1 $
115  MPYAD     TTAA,VTRA,/T[illegible]A/C,N,0/C,N,1/C,N,0/C,N,1 $
```

STATMENT SEQUENCE FOR A FLEXIBLE WING MODEL

```
     LEVEL 2.0 NASTRAN DMAP COMPILER - SOURCE LISTING

116  MPYAD     TTAA,VTMA,/TVEMA/C,N,0/C,N,1/C,N,0/C,N,1 $

117  PARAM     //C,N,ADD/V,N,XQHHL/C,N,1/C,N,0 $

118  AMP       AJJL,SAS,D1JK,D2JK,GTKA,PHIDH,D1JE,D2JE,USETD,AERO/QHHLI,QKHLI,
               QHJLI/V,N,NOUE/V,N,XQHHL/V,Y,GUSTAERO=-1 $

119  SAVE      XQHHL $

120  MPYAD     GPPM,QKHLI,/QKKKI/C,N,1/C,N,1/C,N,0/C,N,1 $

121  MPYAD     TSTA,QKKKI,/PG/C,N,0/C,N,1/C,N,0/C,N,1 $

122  PARAMR    //C,N,MPY/V,N,RCAREA/V,Y,RC/V,Y,WAREA $

123  PARAMR    //C,N,MPY/V,N,QAREA/V,Y,WAREA/V,Y,Q $

124  PARAMR    //C,N,MPY/V,N,QRCAREA/V,N,RCAREA/V,Y,Q $

125  PARAM     //C,N,MPY/V,N,TRANS/C,N,1/C,N,-1 $

126  COND      LBL10,TRANS $

127  LABEL     LBL48 $

128  $*************************************************************

128  $

128  $         THIS SECTIO COMMON TO RIGID FORMATS 1

128  $

128  GP4       CASECC,GEOM4,EQEXIN,GPDT,BGPDT,CSTM/RG,YS,USET,ASET/V,N,LUSET/
               V,N,MPCF1/V,N,MPCF2/V,N,SINGLE/V,N,OMIT/V,N,REACT/V,N,NSKIP/V,
               N,REPEAT/V,N,NOSET/V,N,NOL/V,N,NOA/C,Y,SUBID $

129  SAVE      MPCF1,MPCF2,SINGLE,OMIT,REACT,NSKIP,REPEAT,NOSET,NOL,NOA $

130  COND      ERROR3,NOL $

131  PARAM     //C,N,AND/V,N,NOSR/V,N,SINGLE/V,N,REACT $

132  PURGE     KRR,KLR,QR,DM/REACT/GM/MPCF1/GO,KOO,LOO,PO,UOOV,RUOV/OMIT/PS,
               KFS,KSS/SINGLE/QG/NOSR $

133  COND      LBL14,GENEL $

134  GPSP      GPL,GPST,USET,SIL/OGPST/V,N,NOGPST $
```

```
STATMENT SEQUENCE FOR A FLEXIBLE WING MODEL


     LEVEL 2.0 NASTRAN DMAP COMPILER - SOURCE LISTING


135  SAVE      NOGPST $

136  COND      LBL14,NOGPST $

137  OFP       OGPST,,,,,// $

138  LABEL     LBL14 $

139  EQUIV     KGG,KNN/MPCF1 $

140  COND      LBL15,MPCF2 $

141  MCE1      USET,RG/GM $

142  MCE2      USET,GM,KGG,,,/KNN,,, $

143  LABEL     LBL15 $

144  EQUIV     KNN,KFF/SINGLE $

145  COND      LBL16,SINGLE $

146  SCE1      USET,KNN,,,/KFF,KFS,KSS,,, $

147  LABEL     LBL16 $

148  EQUIV     KFF,KAA/OMIT $

149  COND      LBL17,OMIT $

150  SMP1      USET,KFF,,,/GO,KAA,KOO,LOO,,,,, $

151  LABEL     LBL17 $

152  EQUIV     KAA,KLL/REACT $

153  COND      LBL18,REACT $

154  RBMG1     USET,KAA,/KLL,KLR,KRR,,, $

155  LABEL     LBL18 $

156  RBMG2     KLL/LLL $

157  COND      LBL19,REACT $

158  RBMG3     LLL,KLR,KRR/DM $

159  LABEL     LBL19 $
```

STATMENT SEQUENCE FOR A FLEXIBLE WING MODEL

```
     LEVEL 2.0 NASTRAN DMAP COMPILER - SOURCE LISTING


160  REPT      SETUP,1 $

161  COPY      PG/PGI/DESINT $

162  PURGE     ULVI,UOOVI/IWEQ $

163  PURGE     PSI,QRI/IWEQ $

164  $

164  $**************************************************************************

164  $

164  $         ITERATIVE LOOP FOR FLEXIBLE WING CALCULATIONS

164  $

164  JUMP      LOOPTOP $

165  LABEL     LOOPTOP $

166  COND      FINIS,ELOOP $

167  SSG2      USET,GM,YS,KFS,GO,DM,PG/QR,PO,PS,PL $

168  SSG3      LLL,KLL,PL,LOO,KOO,PO/PHIOH,UOOV,RULV,RUOV/V,N,OMIT/
               V,Y,IRES=-1/V,N,NSKIP/V,N,EPSI $

169  SAVE      EPSI $

170  COND      LBL20,IRES $

171  MATGPR    GPL,USET,SIL,RULV//C,N,L $

172  MATGPR    GPL,USET,SIL,RUOV//C,N,O $

173  LABEL     LBL20 $

174  ADD       PHIOH,ULVI/ULVT $

175  ADD       PS,PSI/PSTT $

176  ADD       UOOV,UOOVI/UOOVT $

177  COND      LBL30,REACT $

178  ADD       QR,QRI/QRT $

179  LABEL     LBL30 $
```

STATMENT SEQUENCE FOR A FLEXIBLE WING MODEL

```
     LEVEL 2.0 NASTRAN DMAP COMPILER - SOURCE LISTING

180  PARAM    //C,N,ADD/V,N,XQHHL/C,N,1/C,N,0 $

181  AMP      AJJL,SAS,D1JK,D2JK,GTKA,PHIDH,D1JE,D2JE,USETD,AERO/QHHL,QKHL,
              QHJL/V,N,NOUE/V,N,XQHHL/V,Y,GUSTAERO $

182  SAVE     XQHHL $

183  MPYAD    SPPM,QKHL,/QKKK/C,N,1/C,N,1/C,N,0/C,N,1 $

184  MPYAD    TSTA,QKKK,/PG/C,N,0/C,N,1/C,N,0/C,N,1 $

185  ADD      PG,PGI/PGT $

186  ADD      QKKK,QKKKI/QKKKT $

187  MPYAD    PHIDH,PHIDH,/TNORV/C,N,1/C,N,1/C,N,0/C,N,1 $

188  PARAML   TNORV//C,N,DMI/C,N,1/C,N,1/V,N,SQVN $

189  PARAMR   //C,N,SQRT/V,N,VNSR/V,N,SQVN $

190  PRTPARM  //C,N,0/C,N,VNSR $

191  PARAMR   //C,N,LT/V,N,CHLCOF/V,N,VNSR/C,Y,ELOOP////V,N,FLAG $

192  COND     LBL50,FLAG $

193  PARAM    //C,N,ADD/V,N,IWEQ/C,N,0/C,N,1 $

194  PURGE    ULVI,PSI,UOOVI/IWEQ $

195  COND     LBL31,REACT $

196  PURGE    QRI/IWEQ $

197  COPY     QRT/QRI/DESINT $

198  LABEL    LBL31 $

199  COPY     QKKKT/QKKKI/DESINT $

200  COPY     ULVT/ULVI/DESINT $

201  COPY     PGT/PSI/DESINT $

202  COPY     PSTT/PSI/DESINT $

203  COPY     UOOVT/UOOVI/DESINT $

204  REPT     LOOPTOP,10 $
```

STATMENT SEQUENCE FOR A FLEXIBLE WING MODEL

```
     LEVEL 2.0 NASTRAN DMAP COMPILER - SOURCE LISTING


205  LABEL     LBL50 $

206  $*************************************************************************

206  $

206  $         THIS SECTION SIMILAR TO RIGID FORMAT 1

206  $         FOR  THE RECOVERY OF TOTAL DISPLACEMENTS AND STRESSES

206  $

206  MATPRN    OKKKT,ULVT,,,// $

207  PARAM     //C,N,ADD/V,N,NSKIP/C,N,0/C,N,1 $

208  JUMP      RECOVERY $

209  LABEL     RECOVERY $

210  SDR1      USET,PGT,ULVT,UOOVT,YS,GO,GM,PSTT,KFS,KSS.QRT/UGV,PGG,QG/
               V,N,NSKIP/C,N,STATICS $

211  PARAM     //C,N,ADD/V,N,NSKIP/C,N,1/C,N,1 $

212  REPT      RECOVERY,1 $

213  GPFDR     CASECC,UGV,KELM,KDICT,ECT,EQEXIN,GPECT,PGG,QG/ONRGY1,OGPFB1/
               C,N,STATICS $

214  OFP       ONRGY1,OGPFB1,,,,// $

215  SDR2      CASECC,CSTM,MPT,DIT,EQEXIN,SIL,GPTT,EDT,BGPDT,,QG,UGV,EST,
               XYCDB,PGG/OPG1,OQG1,OUGV1,OES1,OEF1,PUGV1/C,N,STATICS/V,N,
               NOSORT2=-1 $

216  SAVE      NOSORT2 $

217  COND      LBL21,NOSORT2 $

218  SDR3      OUGV1,OPG1,OQG1,OEF1,OES1,/OUGV2,OPG2,OQG2,OEF2,OES2, $

219  OFP       OUGV2,OPG2,OQG2,OEF2,OES2,//V,N,CARDNO $

220  SAVE      CARDNO $

221  XYTRAN    XYCDB,OPG2,OQG2,OUGV2,OES2,OEF2/XYPLTT/C,N,TRAN/C,N,PSET/V,N,
               PFILE/V,N,CARDNO $

222  SAVE      PFILE,CARDNO $
```

```
STATMENT SEQUENCE FOR A FLEXIBLE WING MODEL


     LEVEL 2.0 NASTRAN DMAP COMPILER - SOURCE LISTING


223  XYPLOT    XYPLTT// $

224  JUMP      DPLOT $

225  LABEL     LBL21 $

226  OFP       OUGV1,OPG1,OQG1,OEF1,OES1,//V,N,CARDNO $

227  SAVE      CARDNO $

228  COND      P2,JUMPPLOT $

229  LABEL     DPLOT $

230  PLOT      PLTPAR,GPSETS,ELSETS,CASECC,BGPDT,EQEXIN,SIL,PUGV1,,GPECT,OES1/
               PLOTX2/V,N,NSIL/V,N,LUSET/V,N,JUMPPLOT/V,N,PLTFLG/V,N,PFILE $

231  SAVE      PFILE $

232  PRTMSG    PLOTX2// $

233  LABEL     P2 $

234  $**********************************************************************

234  $

234  $         THIS SECTION CALCULATES THE AERODYNAMIC COEFFICIENTS

234  $

234  MPYAD     ATMA,QKKKT,/TFACP/C,N,0/C,N,1/C,N,0/C,N,1 $

235  PARAML    TFACP//C,N,DMI/C,N,1/C,N,1/V,N,CLAA $

236  PARAMR    //C,N,DIV/V,N,CLACP/V,N,CLAA/V,Y,WAREA $

237  MPYAD     TVEMA,QKKKT,/TMACP/C,N,1/C,N,1/C,N,0/C,N,1 $

238  MPYAD     TVERA,QKKKT,/TRACP/C,N,1/C,N,1/C,N,0/C,N,1 $

239  PARAML    TMACP//C,N,DMI/C,N,1/C,N,1/V,N,ACMA $

240  PARAML    TRACP//C,N,DMI/C,N,1/C,N,1/V,N,ACRA $

241  PARAMR    //C,N,DIV/V,N,CMACP/V,N,ACMA/V,N,PCAREA $

242  PARAMR    //C,N,DIV/V,N,CRACP/V,N,ACRA/V,N,PCAREA $

243  MPYAD     VGLS,PGT,/TFSGP/C,N,1/C,N,1/C,N,0/C,N,1 $
```

STATMENT SEQUENCE FOR A FLEXIBLE WING MODEL

```
     LEVEL 2.0 NASTRAN DMAP COMPILER - SOURCE LISTING


244  PARAML    TFSGP//C,N,DMI/C,N,1/C,N,1/V,N,ACLS $

245  PARAMR    //C,N,DIV/V,N,CLSGP/V,N,ACLS/V,N,QAREA $

246  MPYAD     VGMS,PGT,/FMSGP/C,N,1/C,N,1/C,N,0/C,N,1 $

247  MPYAD     VTM,FMSGP,/TMSGP/C,N,1/C,N,1/C,N,0/C,N,1 $

248  MPYAD     VTR,FMSGP,/TRSGP/C,N,1/C,N,1/C,N,0/C,N,1 $

249  PARAML    TMSGP//C,N,DMI/C,N,1/C,N,1/V,N,QACMS $

250  PARAML    TRSGP//C,N,DMI/C,N,1/C,N,1/V,N,QACRS $

251  PARAMR    //C,N,DIV/V,N,CMSGP/V,N,QACMS/V,N,QRCAREA $

252  PARAMR    //C,N,DIV/V,N,CRSGP/V,N,QACRS/V,N,QRCAREA $

253  PRTPARM   //C,N,0/C,N,CLACP $

254  PRTPARM   //C,N,0/C,N,CMACP $

255  PRTPARM   //C,N,0/C,N,CRACP $

256  PRTPARM   //C,N,0/C,N,CLSGP $

257  PRTPARM   //C,N,0/C,N,CMSGP $

258  PRTPARM   //C,N,0/C,N,CRSGP $

259  JUMP      FINIS $

260  LABEL     ERROR2 $

261  PRTPARM   //C,N,-2/C,N,FLUTTER $

262  LABEL     ERROR1 $

263  PRTPARM   //C,N,-1/C,N,FLUTTER $

264  LABEL     ERROR4 $

265  PRTPARM   //C,N,-4/C,N,FLUTTER $

266  LABEL     ERROR3 $

267  PRTPARM   //C,N,-3/C,N,STATICS $

268  LABEL     FINIS $
```

APPENDIX B

Model Descriptions

APPENDIX B

The structural model used to test the DMAP sequence consisted of rods CROD, quadralateral membranes CQMEM2, and shear panel CSHEAR elements. The CROD's provide tension, compression and torsional stiffness. The CQMEM2 elements have a finite inplane stiffness but do not resist bending and the CSHEAR elements resist tangential forces, but do not resist normal forces. The elements used in this model do not handle rotational degrees of freedom, so these were constrained out of the solution. The numbered grid points are shown in Figure 7, illustrating the relative shape of the model. The elements were shrunk by 50 percent to show their relavtive positions in Figure 8.

The aerodynamic mesh size used for the panel with twenty boxes is shown in Figure 9 versus the projected mesh of the structural model. The sixty-three box panel is shown in Figure 10. The aerodynamic models used the reflected wing in the calculation of aerodynamic forces.

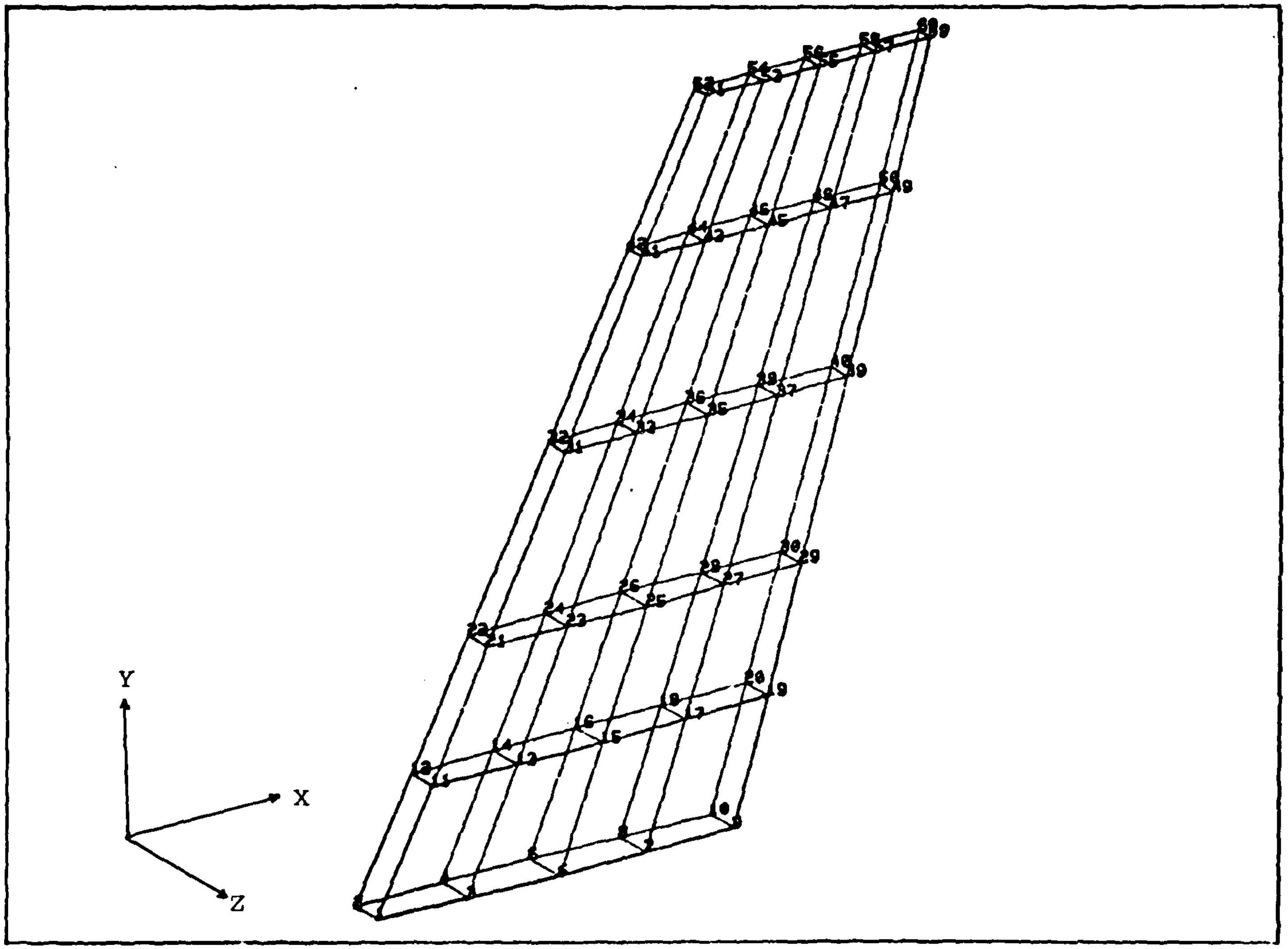

Figure 7. Graphic Depiction of Wing Model.

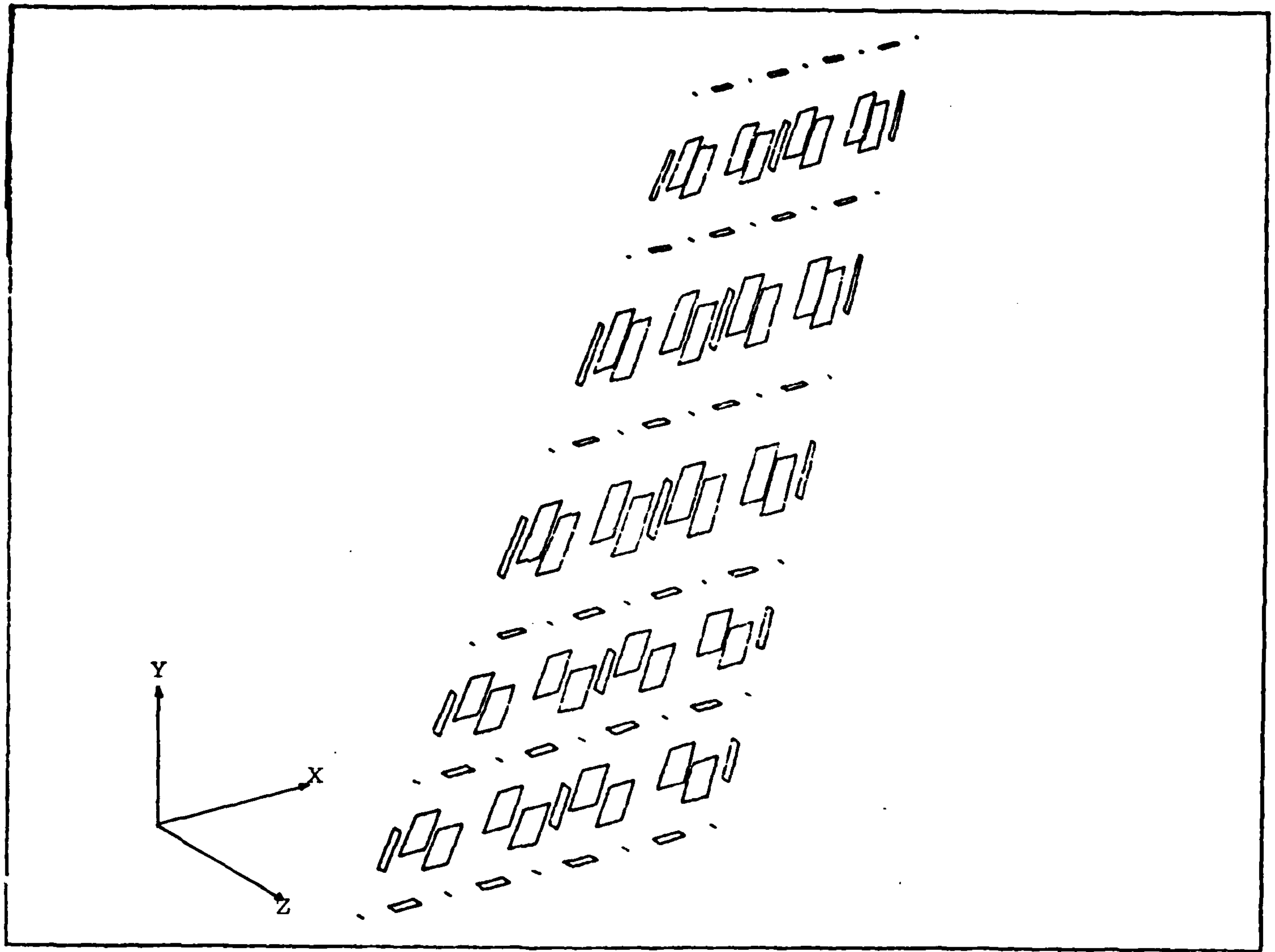

Figure 8. Graphic Depiction of Element Connection.

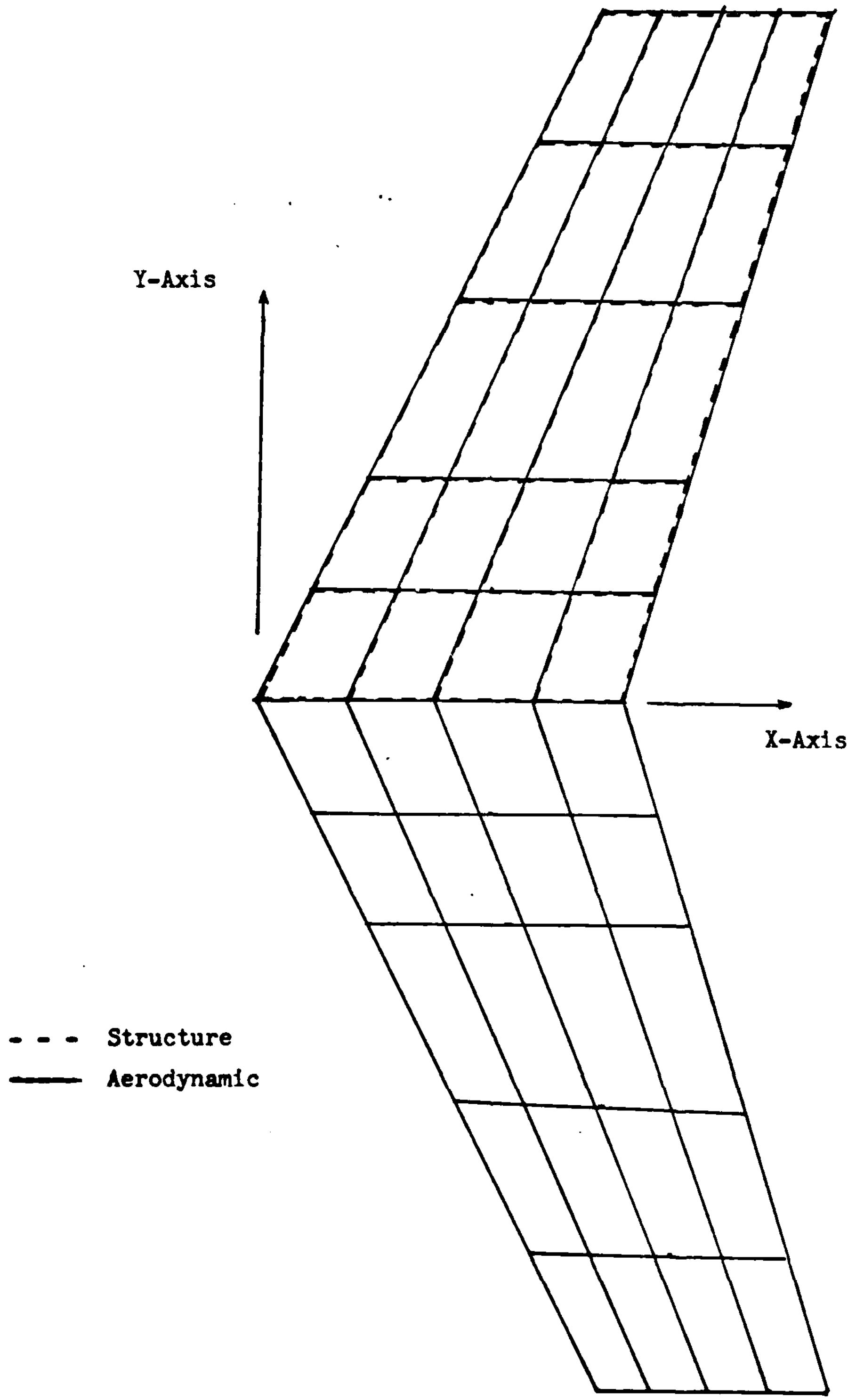

Figure 9. Projected View of Structural and 20-Box Aerodynamic Model.

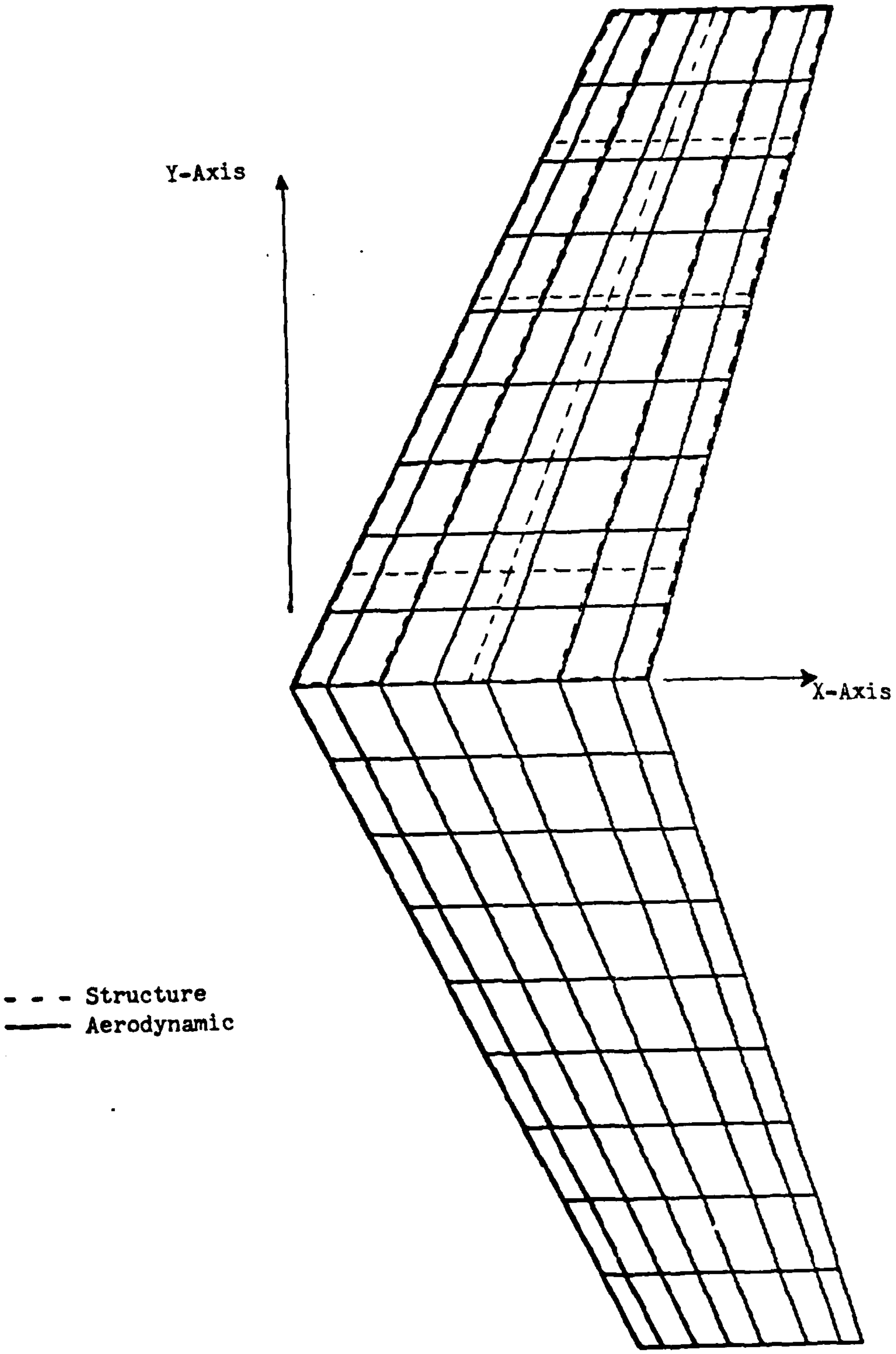

Figure 10. Projected View of Structural and 63-Box Aerodynamic Model.

APPENDIX C

Source Program for Module

```
      SUBROUTINE  GIAS
      INTEGER SAS,GTAK,SASEL,TTTTT,VGLS,VGMS,VTMA,VTM,VTR,VTRA,AIDMIT
      INTEGER ACPT,TOSEL,ECTA,GPLA,BGPA,BUF1,BUF2,BUF3,CORSZ
      INTEGER  TRAIL,MCBA,MCBB,TYPIN,TYPOUT
      DIMENSION  TRAIL(7),MCBA(7),MCBB(7),Z(1),IZ(1),NAME(2)
      EQUIVALENCE (IZ(1),Z(1))
      COMMON  //Z
      COMMON  /SYSTEM/IBUF
      COMMON  /PACKX/TYPIN,TYPOUT,IROW,NROW,NM
      NM=1
      DATA NAME/4HGIAS,4HREAD/
      DATA  ACPT,TOSEL,ECTA,GPLA,BGPA/101,102,103,104,105/
      DATA  SAS,GTAK,TTTTT,SASEL,VGLS,VGMS,VTMA,VTRA,VTM,VTR,AIDMIT/
     *  201,202,203,204,205,206,207,208,209,210,211/
      LCORE=CORSZ(Z,A)
      BUF1=LCORE-IBUF+1
      IF (BUF1.LE.0) GO TO 500
C     AN "N" IS BEFORE THE DATA BLOCK NAME SPECIFIES THE FIRST
C     ADDRESS OF THE BLOCK IN OPEN CORE
C
C     AN "L" BEFORE THE DATA BLOCK NAME SPECIFIES A PARAMETER
C     WHICH IS THE LENGTH OF THE DATA BLOCK IN OPEN CORE
C
      NACPT=1
      CALL GOPEN(ACPT,Z(BUF1),0)
      CALL READ(ACPT,Z(NACPT),6,1,M)
C     NI  IS THE NUMBER OF AERO BOXES AND POINTS
      NI=Z(NACPT+3)
      CALL CLOSE(ACPT,1)
      TRAIL(1)=TOSEL
      CALL RDTRL(TRAIL)
C     ND  IS THE NUMBER OF STRUCTURAL ELEMENTS LOADED AND ALSO
C     THE NUMBER OF DEPENDENT STRUCTURAL POINTS
C
      ND=TRAIL(2)
      NTOSEL=NACPT+5
      CALL GOPEN(TOSEL,Z(BUF1), 0)
      CALL READ (TOSEL,Z(NTOSEL),ND,1,M)
      CALL CLOSE(TOSEL,1)
      TRAIL(1)=GPLA
      CALL RDTRL(TRAIL)
      LGPLA=TRAIL(2)
      LBGPA=4*LGPLA
      CALL GOPEN(GPLA,Z(BUF1),0)
      NGPLA=NTOSEL+ND
      CALL READ(GPLA,Z(NGPLA),LGPLA,1,M)
      CALL CLOSE(GPLA,1)
      NBGPA=NGPLA+LGPLA
      CALL GOPEN(BGPA,Z(BUF1),0)
      CALL READ(BGPA,Z(NBGPA),LBGPA,1,M)
      CALL CLOSE(BGPA,1)
      NCORE=BUF1-1
```

```
C     CHECK TO SEE IF ENOUGH OPEN CORE IS AVAILABLE
      IF (NCORE.LT.(NBGPA+LBGPA)) GO TO 500
      ICOMEM2=5308 IS THE INTERNAL ADDRESS OF COMEM2 ELEMENTS
C     THE FOLLOWING 5 STATMENTS LOCATE THE RECORD FOR THESE
C     ELEMENTS IN THE ECTA TABLE
      CALL GOPEN(ECTA,Z(BUF1),0)
      ICOMEM2=5308
      NECTA=NBGPA+LBGPA
    1 CALL READ (ECTA,Z(NECTA),3,0,M)
      IF (IZ(NECTA).EQ.ICOMEM2) GO TO 2
      CALL READ(ECTA,Z(NECTA),1,1,M)
      GO TO 1
    2 TRAIL(1)=SIL
      CALL FDTRL(TRAIL)
      NDOF=TRAIL(2)
      NGP=NDOF/6
      BUF2=BUF1-IBUF
      BUF3=BUF2-IBUF
      NCORE=BUF3-1
      LECTA=7
      NND=NECTA+LECTA
      LND=2*ND
      NTEX=NND+LND
      LTEX=NGP
      NTEY=NTEX+LTEX
      LTEY=NGP
      NSASEL=LTEY+NTEY
      LSASEL=ND
      NTTTTT=NSASEL+LSASEL
      LTTTTT=NDOF
      NVGLS=NTTTTT+LTTTTT
      LVGLS=NDOF
      NVGMS=NVGLS+LVGLS
      LVGMS=NDOF*NGP
      ICOREN=NVGMS+LVGMS
      IF(NCORE.LT.ICOREN) GO TO 500
      FLAG=-10000.0
      DO 9 J=1,NDOF
      Z(NTEX+J-1)=FLAG
      Z(NTEY+J-I)=FLAG
    9 CONTINUE
      DO 10 J=1,LVGMS
      Z(NVGMS+J-1)=0.0
   10 CONTINUE
      DO 7 J=1,LVGLS
      Z(NVGLS+J-1)=0.0
    7 CONTINUE
      CALL GOPEN(SASEL,Z(BUF2),1)
      CALL GOPEN(TTTTT,Z(BUF3),1)
      TYPIN=1
      TYPOUT=1
      IROW=1
```

```
      MCBA(1)=SASEL
      MCBA(2)=0
      MCBA(3)=LSASEL
      MCBA(4)=3
      MCBA(5)=1
      MCBA(6)=0
      MCBA(7)=0
      MCBB(1)=TTTTT
      MCBB(2)=0
      MCBB(3)=LTTTTT
      MCBB(4)=2
      MCBB(5)=1
      MCBB(6)=0
      MCBB(7)=0
C     LOOP 3 CALCULATES THE FOLLOWING DATA BLOCKS SASEL,
      TTTTT,VGLS,AND VGMS
      DO 3 I=1,NO
    4 CALL READ(ECTA,Z(NECTA),7,0,M)
      IF(IZ(NECTA).EQ.IZ(NTOSEL+I-1)) GO TO 5
      GO TO 4
    5 IG1=IZ(NECTA+2)
      IG2=IZ(NECTA+3)
      IG3=IZ(NECTA+4)
      IG4=IZ(NECTA+5)
      IGP1=0
      IGP2=0
      IGP3=0
      IGP4=0
      DO 6 J=1,LGPLA
      K=NGPLA+J-1
      IF(IG1.EQ.IZ(K)) IGP1=NGPLA+(J-1)*4
      IF(IG2.EQ.IZ(K)) IGP2=NGPLA+(J-1)*4
      IF(IG3.EQ.IZ(K)) IGP3=NGPLA+(J-1)*4
      IF(IG4.EQ.IZ(K)) IGP4=NGPLA+(J-1)*4
    6 CONTINUE
      IF(IGP1.EQ.0) GO TO 500
      IF(IGP2.EQ.0) GO TO 500
      IF(IGP3.EQ.0) GO TO 500
      IF(IGP4.EQ.0) GO TO 500
      Z(NVGLS+5*IG1-3)=1.0
      Z(NGVLS+5*IG2-3)=1.0
      Z(NVGLS+5*IG3-3)=1.0
      Z(NVGLS+5*IG4-3)=1.0
      Z(NVGMS+5*IG1-3+(IG1-1)*NDOF)=1.0
      Z(NVGMS+5*IG2-3+(IG2-1)*NDOF)=1.0
      Z(NVGMS+5*IG3-3+(IG3-1)*NDOF)=1.0
      Z(NVGMS+5*IG4-3+(IG4-1)*NDOF)=1.0
      X1=Z(NBGPA+IGP1+1)
      Y1=Z(NGPA+IGP1+2)
      X2=Z(NGPA+IGP2+1)
      Y2=Z(NGPA+IGP2+2)
      X3=Z(NGPA+IGP3+1)
```

```
      Y3=Z(NGPA+IGP3+2)
      X4=Z(NGPA+IGP4+1)
      X4=Z(NGPA+IGP4+2)
      A=.5*((X3-X1)*(Y4-Y2)+(Y3-Y1)*(X4-X2))
      NROW=ND
      DO 8 J=1,ND
    8 Z(NSASEL+J-1)=0.0
      Z(NSASEL+I-1)=1
      CALL PACK(Z(NSASEL),SASEL,MCBA)
      NROW=LTTTTT
      DO 11 J=1,LTTTTT
   11 Z(NTTTTT+J-1)=1.0
      Z(NTTTTT+6*IG1-3)=.25
      Z(NTTTTT+6*IG2-3)=.25
      Z(NTTTTT+6*IG3-3)=.25
      Z(NTTTTT+6*IG4-3)=.25
      CALL PACK(Z(NTTTTT),TTTTT,MCBB)
      Z(NND+2*I-1)=(X1+X2+X3+X4)/4.
      Z(NND+2*I)=(Y1+Y2+Y3+Y4)/4.
      Z(NTEX-1+IG1)=X1
      Z(NTEX-1+IG2)=X2
      Z(NTEX-1+IG3)=X3
      Z(NTEY-1+IG4)=X4
      Z(NTEY-1+IG1)=Y1
      Z(NTEY-1+IG2)=Y2
      Z(NTEY-1+IG3)=Y3
      Z(NTEY-1+IG4)=Y4
    3 CONTINUE
      CALL CLOSE(ECTA,1)
      CALL CLOSE(SASEL,1)
      CALL CLOSE(TTTTT,1)
      CALL WRTTRL(MCBA)
      CALL WRTTRL(MCBB)
      MCBA(1)=VGLS
      MCBA(2)=0
      MCBA(3)=LVGLS
      MCBA(4)=2
      MCBA(5)=1
      MCBA(6)=0
      MCBA(7)=0
      MCBB(1)=VGMS
      MCBB(2)=0
      MCBB(3)=NDOF
      MCBB(4)=2
      MCBB(5)=1
      MCBB(6)=0
      MCBB(7)=0
      NROW=LVGLS
      CALL GOPEN(VGLS,Z(BUF1),1)
      CALL PACK(Z(NVGLS),VGLS,MCBA)
      CALL CLOSE(VGLS,1)
      CALL GOPEN(VGMS,Z(BUF2),1)
```

```
      DO 12 J=1,NGP
      CALL PACK(Z(NVGMS+(J-1)*NTOF),VGMS,MCBB)
   12 CONTINUE
      CALL CLOSE(VGMS,1)
      CALL WRTTRL(MCBA)
      CALL WRTTRL(MCBB)
      CALL GOPEN(VTM,Z(BUF1),1)
      CALL GOPEN(VTR,Z(BUF1),1)
      NTEXR=NBASEL
      NTEYR=NTEXR+NGP
      LTEXR=NGP
      LTEYR=NGP
      NC=1
C     LOOP 13 CALCULATES THE VTM AND VTR MATRICIES
      DO 13 J=1,NGP
      IF(Z(NTEX+J-1).EQ.FLAG) GO TO 13
      Z(NTEXR+NC-1)=Z(NTEX+J-1)
      Z(NTEYR+NC-1)=Z(NTEY+J-1)
      NC=NC+1
   13 CONTINUE
      NROW=NC
      MCBA(1)=VTM
      MCBA(2)=0
      MCBA(3)=NROW
      MCBA(4)=2
      MCBA(5)=1
      MCBA(6)=0
      MCBA(7)=0
      MCBB(1)=VTR
      MCBB(2)=0
      MCBB(3)=NROW
      MCBB(4)=2
      MCBA(5)=1
      MCBB(6)=0
      MCBB(7)=0
      CALL PACK(Z(NTEXR),VTM,MCBA)
      CALL PACK(Z(NTEYR),VTR,MCBB)
      CALL CLOSE(VTM,1)
      CALL CLOSE(VTR,1)
      CALL WRTTRL(MCBA)
      CALL WRTTRL(MCBB)
      MIL=100000
      L=1
C     NCB EQUALS THE NUMBER OF CHORDWISE BOXES
C     NSB EQUALS THE NUMBER OF SPANWISE BOXES
      NCB=Z(NADPT+5)
      NSB=Z(NADPT+2)
      DO 14 J=1,NSB
      DO 14 I=1,NCB
      IA1=MIL+L
      IA2=MIL+L+1
      IA3=MIL+L+NCB
```

```
      IA4=NIL+L+NDB+1
      IAP1=0
      IAP2=0
      IAP3=0
      IAP4=0
      DO 15  KK=1,LGPLA
      K=NGPLA+KK-1
      IF(IA1.EQ.IZ(K))   IAP1=NGPLA+(KK-1)*4
      IF(IA2.EQ.IZ(K))   IAP2=NGPLA+(KK-1)*4
      IF(IA3.EQ.IZ(K))   IAP3=NGPLA+(KK-1)*4
      IF(IA4.EQ.IZ(K))   IAP4=NGPLA+(KK-1)*4
   15 CONTINUE
      IF(IAP1.EQ.0) GO TO 500
      IF(IAP2.EQ.0) GO TO 500
      IF(IAP3.EQ.0) GO TO 500
      IF(IAP4.EQ.0) GO TO 500
      X1=Z(NGPLA+IAP1+1)
      X2=Z(NGPLA+IAP2+1)
      X3=Z(NGPLA+IAP3+1)
      X4=Z(NGPLA+IAP4+1)
      Y1=Z(NGPLA+IAP1+2)
      Y2=Z(NGPLA+IAP2+2)
      Y3=Z(NGPLA+IAP3+2)
      Y4=Z(NGPLA+IAP4+2)
      NNI=NTEX
C     THESE ARE THE COORDINATES OF AERO POINTS AT THE
C     ONE QUARTER CHORD POINT OF THE BOX
      Z(NNI+2*L-1)=(X1+X3+(X2+X4-X3-X1)/4.)/2.
      Z(NNI+2*L)=(Y1+Y2+Y3+Y4)/4.
      L=L+1
   14 CONTINUE
      MCBA(1)=GTAK
      MCBA(2)=0
      MCBA(3)=NI
      MCBA(4)=2
      MCBA(5)=1
      MCBA(6)=0
      MCBA(7)=0
      NGTAK=NNI+NI
      KT=0
      KX=0
      KY=0
      KD=0
      DZ=0
      NSCR=((NI+3)**2+3*(NI+3)+NI*ND+NI*(NI+3))
C     THIS CHECKS OPEN CORE TO SEE IF ENOUGH IS AVAILABLE
      IF(NCORE.LT.(NSCR+NGTAK)) GO TO 500
      CALL SSPLIN(NI,Z(NNI),ND,Z(NND),KX,KY,KD,KT,DZ,Z(NGTAK),NSCR,
     *  ISING)
      BUF1=LCORE-IBUF+1
      BUF2=BUF1-IBUF
      NCORE=BUF2-1
```

```
      IF(NCORE.LT.(NGTAK+NI*ND)) GO TO 500
      CALL GOPEN(GTAK,Z(BUF1),1)
      NROW=ND
      DO 16 J=1,ND
      CALL PACK(Z(NGTAK+(J-1)*ND),GTAK,MCBA)
   16 CONTINUE
      CALL CLOSE(GTAK,1)
      CALL WRTTRL(MCBA)
      CALL GOPEN(VTMA,Z(BUF1),1)
      CALL GOPEN(VTRA,Z(BUF2),1)
      NTEX=NGTAK
      LTEX=2*NI
      NTEY=NTEX+LTEX
C     LOOP 17 CALCULATES THE VTMA AND VTRA MATRICIES
      DO 17 I=1,NI
      Z(NTEX+I-1)=Z(NNI+2*I-1)
      Z(NTEY+I-1)=Z(NNI+2*I)
   17 CONTINUE
      MCBA(1)=VTMA
      MCBA(2)=0
      MCBA(3)=NROW
      MCBA(4)=2
      MCBA(5)=1
      MCBA(6)=0
      MCBA(7)=0
      MCBB(1)=VTRA
      MCBB(2)=0
      MCBB(3)=NROW
      MCBB(4)=2
      MCBB(5)=1
      MCBB(6)=0
      MCBB(7)=0
      CALL PACK(Z(NTEX),VTMA,MCBA)
      CALL PACK(Z(NTEY),VTRA,MCBB)
      CALL CLOSE(VTMA,1)
      CALL CLOSE(VTRA,1)
      CALL WRTTRL(MCBA)
      CALL WRTTRL(MCBB)
      NAIDMIT=1
      DO 18 I=1,NI
      Z(NAIDMIT+I-1)=1.0
   18 CONTINUE
      CALL GOPEN(AIDMIT,Z(BUF1) ,1)
      MCBA(1)=AIDMIT
      MCBA(2)=0
      MCBA(3)=NI
      MCBA(4)=2
      MCBA(5)=1
      MCBA(6)=0
      MCBA(7)=0
      NROW=NI
      CALL PACK(Z(NAIDMIT),AIDMIT,MCBA)
```

```
      CALL CLOSE (AIDMTT,1)
      CALL WRTTRL(MCBA)
      CALL GOPEN(SAS,Z(BUF1),1)
      TYPOUT=3
      NSAS=1
      NROW=2*NI
      NTL=NI*NROW
      DO 19 I=1,NTL
      Z(NSAS)=0.0
   19 CONTINUE
      L=1
      DO 20 I=1,NI
      Z(NSAS+L+I-2)=1.0
      Z(NSAS+L+I-1)=1.0
   20 CONTINUE
      MCBA(1)=SAS
      MCBA(2)=0
      MCBA(3)=NROW
      MCBA(4)=2
      MCBA(5)=3
      MCBA(6)=0
      MCBA(7)=0
      L=1
      DO 21 I=1,NI
      CALL PACK(Z(NSAS+L-1),SAS,MCBA)
      L=L+NROW
   21 CONTINUE
      CALL CLOSE(SAS,1)
      CALL WRTTRL(MCBA)
      RETURN
  500 N=-8
      FILE=0
      CALL MESAGE(N,FILE,NAM)
      END
```

VITA

Lance Eliot Chrisinger was born on 20 August 1952 in Munich, Germany. He graduated from high school in Winfield, Iowa in 1970 and attended Iowa State University. Following graduation in 1974 with a Bachelor of Science degree in Aerospace Engineering, he received a commission in the USAF through the ROTC program. He then entered Undergraduate Navigator training and received his wings in April 1975. He served as a F-4 Weapons System Officer at Holloman AFB, New Mexico and at Keflavik, Iceland until entering the School of Engineering, Air Force Institute of Technology, in June 1979.

Permanent address: P. O. Box 32
Winfield, Iowa 52659

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GAE/AA/80D-2	2. GOVT ACCESSION NO. AD-A094770	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) CALCULATION OF AIRLOADS FOR A FLEXIBLE WING VIA NASTRAN		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Lance E. Chrisinger		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology(AFIT-EN) Wright-Patterson AFB, OH 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE December, 1980
		13. NUMBER OF PAGES 67
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES Approved for public release;IAW190-17

FREDRIC C. LYNCH, Major, USAF

Director of Public Affairs

30 DEC 1980

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

NASTRAN

Flexible Airloads

Airload Application

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis describes the use of the NASTRAN program (Level 17.0) in the calculation of flexible wing airloads and stresses. The problems of interfacing the aerodynamic and structural models are discussed with assumptions needed to solve them. Two different methods of transferring the aerodynamic forces into a structural load vector are presented. A Direct Matrix Abstraction Program was developed for use in these calculations. This

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) sequence is limited to the use of one doublet-lattice panel for the aerodynamic model and requires several new data blocks for execution. Although these data blocks were input with the bulk data deck to test the sequence, a new preliminary module has been constructed to build them internally. The results of this investigation show that NASTRAN can be used in calculating the airloads and stresses for a flexible wing. This new sequence has extended the capability of NASTRAN to allow for the application of internally generated steady airloads to the structural model.

LMED

8