NASA-CR-204515

WING WEIGHT OPTIMIZATION UNDER AEROELASTIC LOADS SUBJECT TO STRESS CONSTRAINTS

Rakesh K. Kapania, J. Issac, D. Macmurdy
Aerospace and Ocean Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA, 24061-0203

and

Guru P. Guruswamy
NASA Ames Research Center
Moffett Field, CA, 98435

Contract Number: **NCC2-5036**

Abstract

A minimum weight optimization of the wing under aeroelastic loads subject to stress constraints is carried out. The loads for the optimization are based on aeroelastic trim. The design variables are the thickness of the wing skins and planform variables. The composite plate structural model incorporates first-order shear deformation theory, the wing deflections are expressed using Chebyshev polynomials and a Rayleigh-Ritz procedure is adopted for the structural formulation. The aerodynamic pressures provided by the aerodynamic code at a discrete number of grid points is represented as a bilinear distribution on the composite plate code to solve for the deflections and stresses in the wing. The lifting-surface aerodynamic code FAST is presently being used to generate the pressure distribution over the wing. The envisioned ENSAERO/Plate is an aeroelastic analysis code which combines ENSAERO version 3.0 (for analysis of wing-body configurations) with the composite plate code.

Introduction

There is a constant effort ongoing at NASA Ames Research Center to develop ENSAERO as an aeroelastic research and analysis tool. This requires coupling of structural analysis modules to the ENSAERO code to perform static/dynamic aeroelastic analysis and optimization. The envisioned ENSAERO/Plate is an aeroelastic analysis code

1

which combines ENSAERO version 3.0 (for analysis of wing-body configurations) with a composite plate model developed by Kapania and Lovejoy for the analysis of thick, skew trapezoidal laminated plates.

The composite plate model incorporates first-order shear deformation theory so that it can be used to model thick wings, where shear deformation effects are important. The wing deflections are expressed using Chebyshev polynomials and a Rayleigh-Ritz procedure is adopted for the structural formulation. The model is capable of solving for the transverse and inplane displacements and rotations. However, only the transverse deflection of the plate model due to the pressure distribution over the wing (obtained at discrete grid points from an aerodynamic code) is desired.

The plate code is set up in such a way that it can also be run independently without running the aerodynamic code. For example, the number of eigensolutions required can be specified to obtain natural frequencies and modes of the composite plate. The aerodynamic pressure distribution over the plate can be calculated using any independent aerodynamic code and input to the plate code for performing structural analysis providing deflections and stresses in the plate. The aerodynamic pressures provided by the aerodynamic model at a discrete number of grid points is represented as a bi-linear distribution on the composite plate structural model to solve for the deflections of the plate.

The aerodynamic model supplies two main pieces of information required to generate the generalized force vector for use with the equivalent plate structural code, namely the surface grid locations and the pressure coefficients at those locations. The structural model supplies the displacement shape functions upon which the generalized forces are based. With this information, the elements of the generalized force vector are generated by calculating the work done by these non-conservative pressure forces in going through the wing displacements. Once the generalized force vector is constructed, the governing equations for the static problem are solved for the generalized displacements from which the actual displacement field is computed. The transformation of the CFD surface grid

2

coordinates and the bi-linear pressure representation that is adopted are discussed in the subsequent sections.

First, the code is validated for the wing deflections obtained when the pressure distribution over the wing is represented using a bilinear distribution using know pressure values at the four corner grid points of each bilinear panel. The stresses at various locations in the wing skins due to a known load distribution are calculated and validated. A minimum weight optimization of the wing under aeroelastic loads subject to stress constraints is carried out. The loads for the optimization are based on aeroelastic trim. The yield criterion proposed by Hill for anisotropic materials is used for the stress constraints in performing the optimization. The design variables are the thickness of the wing skins and planform variables. The lifting-surface aerodynamic code FAST is presently being used to generate the pressure distribution over the wing. The next step will be the coupling of ENSAERO with the plate code to generate the static aeroelastic loads.

Structural model

The structural model is based on a first-order shear deformation theory developed by Lovejoy and Kapania which incorporates transverse shear.

The plate displacements u, v and w are given by

$$u = u^o(x, y, t) + z\phi_x(x, y, t)$$

$$v = v^o(x, y, t) + z\phi_y(x, y, t)$$

$$w = w^o(x, y, t)$$

where u^o, v^o and w^o are midplane displacements and ϕ_x and ϕ_y are rotations about the y and x axes, respectively.

The original (x, y) coordinate system is transformed to the (η, ξ) coordinate system, and a Rayleigh-Ritz formulation using Chebyshev polynomials to represent the plate displacements is adopted. The coordinate transformation is accomplished by

$$x(\eta, \xi) = \sum_{i=1}^{4} N_i(\eta, \xi) \, x_i$$

3

$$y(\eta, \xi) = \sum_{i=1}^{4} N_i(\eta, \xi) \, y_i$$

where

$$N_i(\eta, \xi) = \frac{1}{4}(1 + \eta\eta_i)(1 + \xi\xi_i)$$

and (x_i, y_i) and (η_i, ξ_i) are the coordinates of the four corner points of the wing in the original and transformed coordinate systems respectively. Note that the domain of the transformed coordinates (η, ξ) ranges from -1 to 1 in both directions.

The mid-plane displacements and rotations are represented in a series as

$$u^o = \sum_{i=0}^{I} \sum_{j=0}^{J} R_{ij}(t) T_i(\eta) T_j(\xi)$$

$$v^o = \sum_{k=0}^{K} \sum_{l=0}^{L} S_{kl}(t) T_k(\eta) T_l(\xi)$$

$$w^o = \sum_{m=0}^{M} \sum_{n=0}^{N} P_{mn}(t) T_m(\eta) T_n(\xi)$$

$$\phi_x = \sum_{p=0}^{P} \sum_{q=0}^{Q} X_{pq}(t) T_p(\eta) T_q(\xi)$$

$$\phi_y = \sum_{r=0}^{R} \sum_{s=0}^{S} Y_{rs}(t) T_r(\eta) T_s(\xi) \qquad -1 \leq \eta, \xi \leq 1$$

and the Chebyshev polynomials are given by

$$T_0(\psi) = 1$$

$$T_1(\psi) = \psi$$

$$T_i(\psi) = 2\psi T_{i-1} - T_{i-2} \qquad -1 \leq \psi \leq 1$$

The plate boundary conditions are handled by using springs of appropriate magnitude at the boundaries. For modeling cantilever wings, linear and rotational springs of large magnitude are placed at the root to satisfy the clamped wing boundary condition.

The wing displacements are obtained from the solution of a set of equations given by

$$[K]\{q\} = \{F\}$$

4

where $[K]$ is the stiffness matrix and the vector $\{q\}$ contains coefficients of the polynomial representing the displacement functions, i.e.,

$$\{q\} = (R_{00}, R_{01}, ... R_{ij}; S_{00}, ... S_{kl}; P_{00}, ... P_{mn}; X_{00}, ... X_{pq}; Y_{00}, ... Y_{rs})^T$$

Bilinear pressure representation

The wing pressure distribution over a quadrilateral panel, in general can be expressed as an interpolant of known values at specific locations on the panel. The bilinear interpolation scheme is adopted here for approximating the pressures over a panel. For the bilinear interpolation method, trapezoidal panels are placed between sets of four known discrete pressures obtained from the CFD grid points, and the pressure at any interior point of that panel is calculated using the bilinear interpolation functions. The pressures are given in terms of panel local coordinates which has a domain ranging from -1 to 1 in both coordinate directions.

For a square panel defined as $(u, v) = (-1, 1)$, and the values at the four corners of the panel as b_{00}, b_{01}, b_{10} and b_{11}, the interpolated value p at any point (u, v) is given by:

$$p(u,v) = \frac{1}{4}[(1-u)(1-v) \quad (1-u)(1+v) \quad (1+u)(1-v) \quad (1+u)(1+v)] \begin{Bmatrix} b_{00} \\ b_{01} \\ b_{10} \\ b_{11} \end{Bmatrix}$$

Alternatively

$$p(u,v) = \{R\}^T \{a\}$$

where $\{R\}$ is the interpolation vector, and $\{a\}$ is actually the true pressures at the corner points of the bilinear panel under consideration.

The contributions from all the bilinear pressure panels are summed up to assemble the global force vector which are functions of the Chebyshev polynomial-based displacement shape functions, the geometry of the aerodynamic grid on the wing surface and the bilinear pressure distribution. The pressures on the wing are applied only to the transverse deflection degrees of freedom of the plate. The bilinear panels are integrated using Gaussian Quadrature in each direction.

Transformation of CFD surface grid coordinates of the wing

The aerodynamic pressures from the aerodynamic code are output at discrete points on the CFD grid. The transformed coordinates (η, ξ) of the CFD surface grid for a wing for $(-1 \leq \eta, \xi \leq 1)$ have to be calculated so that the discrete aerodynamic pressures at the physical coordinates (x, y) of the wing can be converted into generalized aerodynamic forces based on the Ritz functions in the structural plate model. In order to carry out this transformation, Murthi & Valliapan's Inverse Mapping Routines are used which calculates the local coordinate (η, ξ) of a point (x, y) where (η, ξ) are defined from -1 to 1 in each direction. This is done conceptually by drawing a straight line from one corner of the domain in (x, y) through the point of interest. In (η, ξ)-space this is a parabola of known equation form. If the parabola is defined over the entire possible -1 to 1 value of either ξ or η then a line search is conducted to find the precise point of interest in (η, ξ). (At least one of the four corners of the domain can be used to choose such a line, if necessary by interchanging the axes and renumbering the nodes).

Force vector generation for composite plate code

The aerodynamic model supplies two main pieces of information required to generate the generalized force vector for use with the equivalent plate code: (a) the surface grid locations and (b) the pressure coefficients at those locations. The structural model supplies the displacement shape functions upon which the genralized forces are based.

For the static problem, we have

$$[K]\{C\} = \{Q\}$$

where $[K]$ is the stiffness matrix, $\{C\}$ is the vector of generalized displacements and $\{Q\}$ is the generalized force vector. The generalized force vector terms can be written as:

$$Q_i = \int\int_\Omega p(x, y)\gamma_i(x, y)dxdy$$

7

where Q_i is the generalized force term corresponding to the *i-th* displacement shape function $\gamma_i(x, y)$, $p(x, y)$ is the pressure field on the surface of the structure and Ω is the wing surface area.

The generalized displacements C_i are related to the actual displacement field over the wing $w(\eta, \xi)$ by

$$w(\eta, \xi) = \sum_i \gamma_i(\eta, \xi) \, C_i$$

where $-1 \leq \eta, \xi \leq 1$, η and ξ are the transformed coordinates for the wing and the γ_i's are Chebyshev polynomials.

Analysis

The aeroelastic response of the wing was obtained in an iterative fashion. The pressure distribution on the wing is first obtained from FAST by assuming the wing to be rigid and having an angle of attack of 1°. The pressure distribution thus obtained is imposed on the wing and the generalized forces for the plate code are calculated for this loading. The elastic wing displacements output by the plate code are superimposed on the rigid wing displacements and a new pressure distribution on the wing is obtained. This pressure distribution is then used to calculate the new displacements. The total lift on the wing is calculated, and a new trim angle of attack is obtained by dividing the total required lift by the current calculated lift and multiplying by the current trim angle of attack. This process is repeated till a converged value of the trim angle of attack is achieved for the wing. The optimization is carried out about this aeroelastic trim position.

Stress calculations

Once the unknown coefficients of the polynomial representing the displacements u, v, w and rotations ϕ_x, ϕ_y are obtained the strains ϵ_x, ϵ_y, γ_{xy}, γ_{xz} and γ_{yz} can be calculated by differentiation. The stresses σ_x, σ_y, σ_{xy}, σ_{xz} and σ_{yz} are related to the strains through the lamina transformed stiffnesses. The ply stresses in the material direction (1-2) can then be evaluated as shown:

$$
\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{Bmatrix} = \begin{bmatrix} m^2 & n^2 & 0 & 0 & 0 & 2mn \\ n^2 & m^2 & 0 & 0 & 0 & -2mn \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & m & -n & 0 \\ 0 & 0 & 0 & n & m & 0 \\ -mn & mn & 0 & 0 & 0 & (m^2 - n^2) \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{Bmatrix}
$$

where $m = cos\theta$, $n = sin\theta$ and θ is the angle of fiber orientation in the ply.

The stresses are evaluated at various locations on the wing surface and the stress constraint (for the optimization) is evaluated at each of these locations. Hill's criterion for anisotropic materials given by

$$
\frac{\sigma_1^2}{X^2} - \frac{\sigma_1 \sigma_2}{X^2} + \frac{\sigma_2^2}{Y^2} + \frac{\sigma_{12}^2}{S^2} + \frac{\sigma_{13}^2}{P^2} + \frac{\sigma_{23}^2}{Q^2} \leq 1
$$

is used which is an extension of von Mises' isotropic yield criterion and X, Y, S, P and Q can be regarded as failure strengths.

This criterion is adopted to put constraints on the stresses. The stress constraint is evaluated at various locations on the wing and the maximum value of the constraint is evaluated. The maximum value should be less than unity for the stress constraint not to be violated.

Model validation

In order to validate the bilinear pressure representation, the whole wing surface was idealized as a single bilinear panel and unit pressures were input at the four corners of the panel (see Figure 1), thus simulating a uniform unit pressure load over the wing. The wing is a $10m$ X $1m$ rectangular wing with $t = 0.2m$, $E = 1.512e9$ N/m^2 and $G = 5.815e8$ N/m^2. The deflections obtained from the plate code (with Poisson ratio $\nu = 0.0$) are compared with a beam-theory solution. The tip deflection from the plate code is $1.241e\text{-}3m$ as compared to the beam theory solution of $1.240e\text{-}3m$ calculated using

$$
w = \frac{q_o L^4}{8EI}
$$

9

with $q_o = 1 \ N/m^2$.

Then a linearly varying pressure load from the root to the tip of the wing was input on the rectangular wing (see Figure 2), by using the panel corner values and the deflections obtained are compared with beam-theory solution. The tip deflection from the plate code is 3.309e-4m as compared to the beam theory solution of 3.307e-4m calculated using

$$w = \frac{q_o L^5}{30EI}$$

with $q_o = 1/10 \ N/m/m$.

In order to validate the stresses predicted by the shear-deformable plate code, the maximum strains and stresses at the wing root are calculated (again with Poisson ratio $\nu = 0.0$ for comparison purposes with beam theory solution) and tabulated in Table 1 for a unit pressure loading over the wing. The strains and stresses calculated by using Timoshenko beam theory (first-order shear deformation theory) are shown in Table 2. The strains by Timoshenko beam theory are given by

$$\epsilon_x = z \ \frac{q_o L^2}{2EI}$$

$$\gamma_{xz} = \frac{q_o L}{\kappa G A}$$

where κ is the shear correction factor and A is the area of cross-section. The results from the plate code agree very well with the beam solution.

Optimization results

This is an effort to obtain a minimum weight design of the wing under aeroelastic loads subject to stress constraints. It should be noted that more constraints can be added to this optimization problem if required. The gradients required for the optimization are computed through a sensitivity analysis using automatic differentiation (using ADIFOR).

Since the optimization is performed about the aeroelastic trim position, at every new design point the trim position has to be found. First, for a given wing planform, the wing thickness is used as a design variable with constraints on the stresses. The initial configuration of the wing is as given: AR=3.0, Area=12.0m^2, TR=0.5, Sweep=15°. The initial thickness of the wing was t=0.25m. The rigid body displacements of the wing for a rigid angle of attack of 1° is input to FAST to generate the pressure coefficients on the wing for $M = 0.6$. The dynamic pressure value of 5000 N/m^2 is chosen and the total lift on the wing is calculated by summing up the pressures on the bilinear panels. The elastic deflected shape is superimposed with the rigid displacements and the new lift is calculated. The new angle of attack required to generate a lift of $2(10)^6$ N was found to be 3.839^o. The rigid and deflected shapes at this angle of attack are summed up for the next iteration and the new lift is found to have converged to $2(10)^6$ N. The stress and their gradients have to be evaluated using the loads at this aeroelastic trim position for the optimization. Note that if the design variables are changed, the aeroelastic trim position to provide the required lift needs to be reevaluated with the new wing parameters.

The optimization problem can be stated as follows:

Minimize

$$W_w = \rho_m Stg$$

subject to the constraints

$$\frac{\sigma_1^2}{X^2} - \frac{\sigma_1\sigma_2}{X^2} + \frac{\sigma_2^2}{Y^2} + \frac{\sigma_{12}^2}{S^2} + \frac{\sigma_{13}^2}{P^2} + \frac{\sigma_{23}^2}{Q^2} \leq 1$$

where W_w is the weight of the wing, ρ_m is the density of the material, S is the area of the wing, t is the thickness, g is the acceleration due to gravity. The planform is kept a constant, so thickness is the only design variable and minimizing the weight amounts to minimizing the thickness.

Method of solution

There are several methods to perform this nonlinear constrained optimization. The gradient projection method is based on projecting the search direction into the subspace tangent to the active constraints. In the method of feasible directions, the concept is to stay in the feasible domain, move in a direction which reduces the objective function and stay away from the constraint boundaries. The program by Vanderplaats, CONMIN is an implementation of the method of feasible directions. Another method which uses successive quadratic programming is implemented in the IMSL subroutine, NCONG. The method, based on the iterative formulation and solution of quadratic programming subproblems, uses a quadratic approximation of the Lagrangian and linearization of the constraints.

The optimization problem can be written as

$$Min \ f(\mathbf{x})$$

$$subject \ to \quad g_j(\mathbf{x}) = 0, \quad for \ j = 1, \ldots, m_e$$

$$g_j(\mathbf{x}) \geq 0, \quad for \ j = m_e + 1, \ldots, m$$

We seek the direction $\mathbf{d}$ as the solution of the following quadratic programming problem:

$$Min \ \frac{1}{2}\mathbf{d}^T\mathbf{B}_k\mathbf{d} + \nabla f(\mathbf{x}_k)^T\mathbf{d}$$

$$subject \ to \quad \nabla g_j(\mathbf{x}_k)^T\mathbf{d} + g_j(\mathbf{x}_k) = 0, \quad j = 1, \ldots, m_e$$

$$\nabla g_j(\mathbf{x}_k)^T\mathbf{d} + g_j(\mathbf{x}_k) \geq 0, \quad j = m_e + 1, \ldots, m$$

where $\mathbf{B}_k$ is a positive definite approximation of the Hessian, and $\mathbf{x}_k$ is the current iterate. If $\mathbf{d}_k$ is the solution to the subproblem, a line search is used to find the new point $\mathbf{x}_{k+1}$,

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda\mathbf{d}_k, \quad \lambda \in (0, 1]$$

$\mathbf{B}_k$ is updated according to the modified BFGS formula.

Results

The optimization problem was solved using the IMSL routine NCONG. The wing has one layer of 0° laminated Graphite/Epoxy ($T300/N5208$) with the following material properties: $E_1 = 181$ x 10^9 Pa, $E_2 = 10.3$ x 10^9 Pa, $\nu_{12} = 0.28$, $G_{12} = 7.17$ x 10^9 Pa and $\rho_m = 1600$ kg/m^3. The wing has an $AR = 3.0$, $Area = 12m^2$, $TR = 0.5$ and $Sweep = 15^o$

The gradients of the aeroelastic constraints are obtained by performing sensitivity analysis of the aeroelastic response with respect to the planform parameters using the automatic differentiation package, ADIFOR. The results of the optimization for each iteration are given in Table 3. The stress constraints and their gradients are calculated each time at a design point and the optimization routine is invoked separately each time which provides the new design point for the next iteration. At the first iteration, the wing thickness dropped from $t = 0.25m$ to $t = 0.15m$ which was the lower bound of thickness. The value of the stress constraint was 0.693. The lower bound was then extended to 0.10 and in the next iteration, a value of $t = 0.133048m$ was returned. The stress constraint value was 1.1703 which clearly violated the constraint. In the next iteration, the optimization routine was unsuccessful in its line search from the infeasible region because it needed more than one function evaluation and could not find the new design point in one iteration. So the design space was shrinked with the lower bound at 0.14 and in the next iteration the optimum value of $t = 0.138m$ was obatined with the stress constraint value at 0.99776. Since more function evaluations were needed in the infeasible region and a line search was unsuccessful, it is felt that fitting a response surface to the constraint function would be worthwhile and would be required if the plate code were linked to ENSAERO.

Next the planform parameters, aspect ratio and area were also added as design variables. The new objective function is

$$S\,t \; + \; \frac{p}{(AR - AR_{lb})}$$

13

where S is the area and t is the wing thickness, AR is the aspect ratio with AR_{lb} as its lower bound. The penalty parameter p was added to maximize the difference between the aspect ratio and its lower bound. A value of $p = 10$ is used.

The starting values of $AR = 3.0$, $Area = 12m^2$ and $t = 0.2m$ were chosen. In the third iteration, at values of $AR = 6.6$, $Area = 10.45m^2$ and $t = 0.15m$, the stress constraint value was 1.445 which violated the stress constraint. Again, from this infeasible design space, the optimization routine could not provide a new design point since it required more function evaluations in the line search. However when the penalty parameter p was increased to $p = 100$, a new design point was found with $AR = 7.0$, $Area = 10.27m^2$ and $t = 0.22m$. It should be noted that there are combinations of S and t that can produce the minimum weight design and the solution is non-unique. Also note that this second optimization was not strictly performed about the aeroelastic trim position since only the rigid wing displacements were used to estimate the trim condition. With only the stress constraint prevailing, the wing thickness plays a more important role than the planform variables. One can add more constraints which makes the planform variables drive the optimization.

Computational resources

The structures and aerodynamic codes were run on crunch (R8000) computer. The lifting-surface flutter analysis code FAST was run at zero reduced frequency to generate the pressures at discrete points on the wing surface and required only a few seconds of computational time. This might significantly change when a CFD code like ENSAERO is incorporated. The structures code which is a global model based on a Rayleigh-Ritz formulation takes approximately a minute of computational time to generate the deflections and stresses in the wing. It should be noted that 8 terms were used for the Chebyshev polynomials representing the wing deflections in the x and y directions. The trim angle of attack calculations converged within 3 iterations in all the cases, which means three runs of the structural code and three runs of the aerodynamic code were required.

Once the aeroelastic trim position is determined, the pressures from the aerodynamic code at this angle of attack is input to the structural code to calculate the deflections. These deflections are then superimposed on the rigid displacements to generate the new pressures and the structural code is executed one more time. The stresses, the stress-constraints and their gradients are evaluated. ADIFOR (automatic differentiation) was used to determine the derivatives of the deflections and the stresses and the pressures from FAST. The ADIFOR generated augmented code takes approximately 8 minutes of computational time for the structures code, whereas it takes only a few seconds for the aerodynamic code. Since the optimizations were performed externally at each design point with only the function values, constraint values and their gradients input at that particular design point, estimation of the next design point takes only a few seconds.

Concluding remarks

A plate code based on first order shear-deformation theory which can provide deflections and stresses has been prepared which can read in aerodynamic information such as pressures from arbitrary grid points. A bilinear representation of the pressures over the wing is used for the pressure distribution by idealizing the wing to be made up of bilinear pressure panels with the corner values being provided by discrete aerodynamic pressure data. The generalized forces due to the pressure over the wing are calculated and the wing deflections are calculated using a Rayleigh-Ritz solution. The stresses at various locations of the wing are calculated. A minimum weight optimization of the wing is carried out under aeroelastic loads subject to stress constraints. The loads for the optimization are based on aeroelastic trim. Since the aerodynamic code FAST was used to generate the pressure data, it was possible to use automatic differentiation ADIFOR to generate derivative information. If a code such as ENSAERO is used, it might be worthwhile to resort to response surfaces to approximate the constraint function.

References:

1. Lovejoy, Andrew E., "Natural Frequencies and an Atlas of Mode Shapes for Generally Laminated, Thick, Skew, Trapezoidal Plates," M.S Theis, Virginia Polytechnic Institute and State University, 1994.

2. Eldred, Lloyd, B., "Sensitivity Analysis of the Static Aeroelastic Response of a Wing," Ph.D Dissertation, Virginia Polytechnic Institute and State University, 1993.

Appended to the report is a previous report with a description of how the plate code fits into ENSAERO. Also note that the aspect ratio is defined over the half wing instead of the full wing as stated in the appended report.
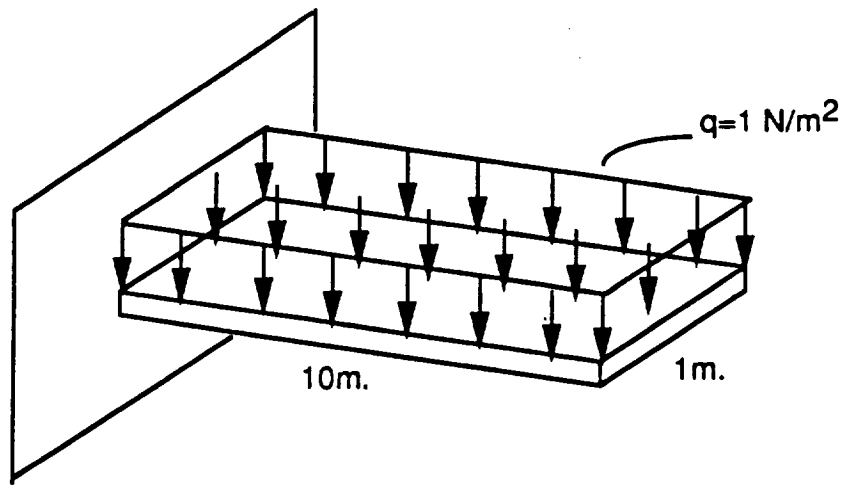
q=1 N/m^2

10m.   1m.

Figure 1. Uniform loading
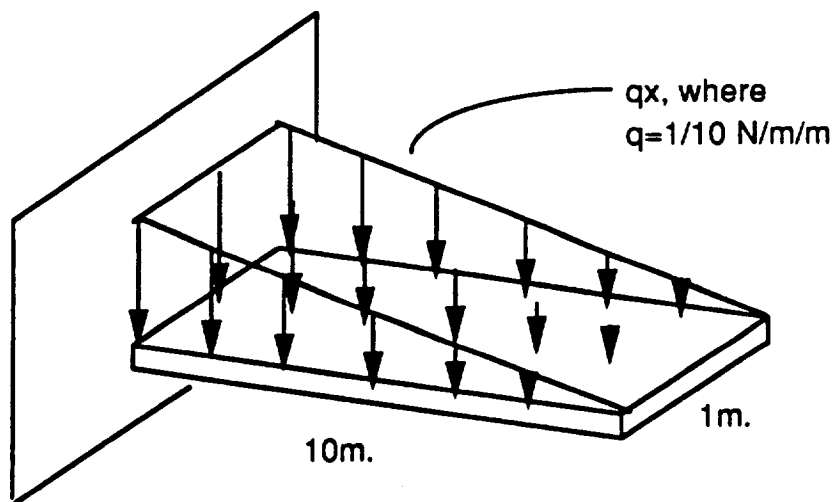

qx, where
q=1/10 N/m/m

10m.   1m.

Figure 2. Linearly varying load

Table 1. Maximum strains and stresses for unit pressure load

Strains	Values	Stresses	Values
ϵ_1	0.	σ_1	0.
$\epsilon_2 = \epsilon_x$	-4.9603e-6	$\sigma_2 = \sigma_x$	-7500.0016
$\gamma_{23} = \gamma_{xx}$	1.04545e-7	$\sigma_{23} = \sigma_{xx}$	60.793
γ_{13}	0.	σ_{13}	0.
γ_{12}	0.	σ_{12}	0.

Table 2. Strains and stresses using Timoshenko beam theory

Strains	Values	Stresses	Values
ϵ_x	-4.9603e-6	σ_x	-7500.
γ_{xx}	1.0454e-7	σ_{xx}	60.793

Table 3. Optimization results for the minimum weight wing design

Iteration	Trim angle of attack	Stress constraint	Thickness (m)
1	3.839°	7.91e-2	0.25
2	2.540°	0.693	0.15
3	2.145°	1.170	0.133048
4	2.313°	0.937	0.14
5	2.266°	0.997	0.138

ENSAERO/Plate

PRELIMINARY USER'S MANUAL

Table of Contents

I. Introduction

This document is a draft user's/programmer's guide to ENSAERO/Plate. ENSAERO/Plate is an aeroelastic analysis code created by combining ENSAERO version 3.0 (for analysis of wing-body configurations) with a composite plate model developed by A. Lovejoy for the analysis of thick, skew, trapezoidal laminated plates.

Included in this manual are descriptions of the routines added to ENSAERO for the plate model, a description of the input variables to ENSAERO, and notes on how to modify, compile and run ENSAERO/Plate on the Cray C90 Eagle located at NASA Ames Research Center. Also included are notes on the methodology used to link the aerodynamic and structural models.

II. Editing, Compiling and Running ENSAERO/Plate

Editing ENSAERO/Plate

Since there is a constant effort ongoing at NASA Ames Research Center to develop ENSAERO as an aeroelastic research and analysis tool, a bookkeeping method for handling the many separate programming efforts is required. Dr. Guruswamy has chosen to use the update utility present on the NASA Cray supercomputers, called **nupdate** on eagle.nas.nasa.gov.

nupdate is used to combine a modification file, containing any corrections, changes and new routines to be added to the ENSAERO code, with a file containing the last baseline version of the code as an indexed object library. The utility creates a compilable FORTRAN source code from these two files.

The format of the **nupdate** utility command to create ENSAERO/Plate is as follows:

nupdate -p ensv20opl -i crensplate.mod -c ensplate -a f -f -o sq

This takes the baseline code (contained in *ensv20opl*) combines it with the modification file (*crensplate.mod*) and creates a file called *ensplate.f*. The *-o sq* option writes the sequence numbers (the identifying numbers **nupdate** uses to label the lines of code) beyond column 72 in the fortran file (so they will be ignored when compiling the code).

For more information on the utility, type **man nupdate** on eagle.

The new code for adding the composite plate model to ENSAERO is contained in the file *crensplate.mod*. This file contains corrections to ENSAERO's wing-body version and adds the composite plate model (while also removing the former finite element model information, so as not to create any conflicts between the structural models). All new code should be added at the tail end of this file. NOTE: Changes to the base code (*ensv20opl*) happen in the order listed in the mod-file.

Compiling ENSAERO/Plate

Once **nupdate** has been used to create the FORTRAN source for ENSAERO/Plate, the following command can be used to create the executable file:

cf77 -Zv -lSimsl -Wf"-a static" ensplate.f

This compiles the file *ensplate.f* and creates a file named *a.out*. The options on the command line are:

-Zv - This option vectorizes math operations (this option may no longer be needed, but it certainly does not harm the compiled code)

-lSimsl - This option links the compiler to the IMSL Library math subroutines, which are used in ENSAERO/Plate to solve the structural governing equations and to find the natural frequencies and mode shapes of composite plates.

-Wf"-a static" - This option forces memory to be allocated in a static manner. Without this option, the compiler attempts to dynamically allocate memory and this causes errors when running ENSAERO/Plate.

I recommend storing *a.out* to the CSF archive storage on eagle, using a recognizable name. The CSF storage is reached by changing directories to $CSF, which is defined on eagle as your personal CSF directory. As an example, I usually move the executable to a CSF directory using:

cp a.out $CSF/working/ensplate.exe

Running ENSAERO/Plate

ENSAERO/Plate runs on the Cray supercomputers are executed as batch jobs. The batch run is controlled by a run script file, and the input data for ENSAERO/Plate is contained in a control data file. In addition, the code will require a surface grid file in PLOT3D/FAST format describing the aerodynamic body, and if the code is starting from a previous case, a restart file.

The run script should be submitted to the cray using the **qsub** command. For example:

qsub -lM 16MW -lT 300 run.raeplate

This command submits a run script file named *run.raeplate* to the queue, and requests 16 MWords of system memory for the job and sets a limit of 300 CPU seconds for execution. Jobs under 300 seconds run in the debug queue, and typically return much faster (from 5 minutes or less to an hour under extremely heavy system use) than jobs over 300 seconds in time, which will take 1/2 a day to overnight to return.

The status of your jobs can be checked with the following command:

qstat -au *username* - Gives the status of all the user's jobs
qstat -a   - Gives the status of all jobs on the system

The run script for ENSAERO/Plate sets up a scratch directory, copies the control data file and all needed data files to the correct input units and executes the job. When the run is complete, the restart file is written to storage and other output files are returned to the user's directory.

In addition, two summary files are sent by the system, with names such as (for the example run script above) *run.rae.e36071* and *run.rae.o36071* where the first portion of the name is the name of the run script file (truncated if needed) and the number refers to the system job number assigned when the job was executed. This number always increases (and resets every 100,000th job) so can be used to differentiate between separate runs.

These summary files include job accounting data (such as CPU time used) and the "*.eXXXXX*" file contains any run-time error messages encountered. The errors are traced to their source, although the line numbers indicated seem to be generated by the compiler rather than matching the original Fortran. Still, the calling routine will be identified, as will the type of error.

The main items to edit in the runscript are the input and output filenames and whether or not a restart file is needed.

Example Run Script

```
# RUN SCRIPT FOR ENSAERO VERSION 3.0 Wingbody (5/94)
#
#    Execute with qsub -lT (# of Seconds) -lM (#)MW [SCRIPTFILE]
#    E.G. - qsub -lT 300 -lM 8MW runcase1
#
#
#    Set CPU's used to 1
set echo
setenv NCPUS 1
cd $HOME/wingbody
#
#    Assign Scratch Space
#
srfs -r 20MW $BIGDIR
cd $BIGDIR
#
#    Copy Input Decks to Scratch Directory
#
cp $HOME/wingbody/raeplate.dat raeplate.dat
#cp $HOME/wingbody/swb_fsintf.dat_c90-new fort.18
#cp $HOME/wingbody/swb_finele.dat_c90-new fort.19
cp $HOME/wingbody/rae_newgrid.dat fort.20
#
#    Start Job Accounting
#
ja
# SSD FILES
env FILENV=ssddefl assign -a $BIGDIR/fort.41 -s u fort.41
env FILENV=ssddefl assign -a $BIGDIR/fort.42 -s u fort.42
env FILENV=ssddefl assign -a $BIGDIR/fort.43 -s u fort.43
env FILENV=ssddefl assign -a $BIGDIR/fort.44 -s u fort.44
env FILENV=ssddefl assign -a $BIGDIR/fort.45 -s u fort.45
env FILENV=ssddefl assign -a $BIGDIR/fort.46 -s u fort.46
env FILENV=ssddefl assign -a $BIGDIR/fort.47 -s u fort.47
env FILENV=ssddefl assign -a $BIGDIR/fort.48 -s u fort.48
env FILENV=ssddefl assign -a $BIGDIR/fort.49 -s u fort.49
#
#get restart file from CSF
#
#cp $CSF/r06a01.res fort.21
cp $CSF/s06a01.res fort.21
#
#    Get ENSAERO Executable file
#
cp $CSF/working/ensplate.exe a.out
chmod u+x a.out
echo "Executable retrieved from CSF Storage"
env FILENV=ssddefl hpm a.out <raeplate.dat> raeplate.out
ja -s
```

```
#
# save restart file on CSF
#
cp fort.22 $CSF/s06a01.res
chmod 600 $CSF/s06a01.res
#
#   Copy result files to home directory
#
cd $HOME/wingbody
cp $BIGDIR/raeplate.out raeplate.out
cp $BIGDIR/fort.11 raeplate.11
#cp $BIGDIR/fort.21 raeplate.21
cp $BIGDIR/fort.31 raeplate.31
cp $BIGDIR/fort.34 raeplate.34
cp $BIGDIR/fort.37 raeplate.37
cp $BIGDIR/fort.38 raeplate.38
#
#   Release Scratch Space
#
srfs -r 0MW $BIGDIR
#
#   END OF SCRIPT
#
```

ENSAERO/Plate Output Files

Using the script above, several output files are written to the user's home directory. The main output file (named *raeplate.out* for the above script) is the default (unit 6) output from the code and contains a variety of information about the run. The other output units are returned with names like *raeplate.11* where the number after the dot refers to the unit and the name is that of the control input data file.

The following table lists the input/ouput files used by ENSAERO/Plate and describes the information contained in each:

I/O UNIT NUMBER	DESCRIPTION
1-3	Reserved for ENSAERO Output files (Pressure, Grid, Q-File)
5	Control Input Data File
6	Main Output File
11	Convergence History
12	Diagnostic File
21	Input Restart File
22	Output Restart File
31	Wing Tip Deflection History
33	Surface Information?
34	Pressure Output (Cl, Cm data)
35,36	Reserved for ENSAERO
41-49	Reserved for SSD

Restart Files

To allow ENSAERO/Plate to continue from any point in a run, the code writes a large data file called a restart file at the end of every run. This file contains all the needed information on the state of the system at the final iteration.

The restart file is a binary file containing the following information:

(a) The deflected grid position (in PLOT3D format)
(b) The aerodynamic flux vector at every point in the grid (also in PLOT3D format)
(c) If the code is used aeroelastically -
> (i) Structural displacement/velocity/acceleration vectors and related state
> information
>> (ii) Decomposed structural matrices (Stiffness/Mass/Damping)
>> (iii) Structural/Aerodynamic interface information (Aerodynamic Matrix,
>> Coordinate transformation data, etc.)

The restart file is used by copying the file into input unit 21 and setting the IREAD flag to 1 if restarting from a different case (such as static aeroelastic from a rigid aerodynamic only case) or 2 if restarting from the same type of run. The user can judiciously change certain control input variables (such as time step) but should avoid altering too many input quantities when restarting.

As a final warning, in aeroelastic cases, the structural model will not be generated again if restarting from a similar case, and therefore the plate model is not accessed. Instead, all the needed information is read from the restart file, and cannot be changed. If the plate model must be regenerated, use an IREAD = 1 value in the input deck.

III. ENSAERO/Plate Input File Format

The following table lists the input quantities for ENSAERO/Plate. In general, all the variables listed between each comment line are to be read on a single line (see example input deck after table for a demonstration.)
The comment lines must be read, although whatever is on the line is ignored.

Variable	Type	Description
TITLE	HOLL	Problem Description in words.
		<COMMENT LINE>
IREAD	INT	Restart Flag: 0 - Start from free stream conditions 1 - Restart from different conditions 2 - Restart from identical conditions
NBLK	INT	Number of Blocks in CFD Grid
		<COMMENT LINE>
JGRD	INT	Number of CFD grid points in x-direction
KGRD	INT	Number of CFD grid points in y-direction
LGRD	INT	Number of CFD grid points in z-direction
		<COMMENT LINE>
ITASK	INT	0 - Steady Rigid Case 1 - Start Unsteady Rigid Case from Steady Rigid Case 2 - Unsteady Forced Motion Case 3 - Static Aeroelastic Case 4 - Dynamic Aeroelastic Case
		<COMMENT LINE>
FSMACH	REAL	Mach Number
GAMMA	REAL	Ratio of Specific Heats
RE	REAL	Reynold's Number
ALP	REAL	Angle of Attack
TINF	REAL	T_∞ for temperature model
		<COMMENT LINE>
DIS2	REAL	Second Order Dissipation for Central Diff. Solution (Set to 0 for Upwind Algorithm)
DIS4	REAL	Fourth Order Dissipation for Central Diff. Solution (Set to 0 for Upwind Algorithm)
		<COMMENT LINE>
IVIS	INT	Global Viscous Option (0 - Inviscid; 1- Viscous)
		<Then Read The Next Two Variables NBLK Times>
IVISK	INT	K-direction Viscous Option (0 - Inviscid; 1- Viscous) for n-th CFD Grid Block
IVISL	INT	L-direction Viscous Option (0 - Inviscid; 1- Viscous) for n-th CFD Grid Block
		<COMMENT LINE>
ITURB	INT	Turbulance Model Flag (0 - Off; 1 - On) [Baldwin-Lomax Model]
		<COMMENT LINE>
IDSM	INT	Degani-Schiff Modification to Turbulance Model (0 - Off; 1 - On)
		<COMMENT LINE>
NSTART	INT	Starting Time Step
NSTOP	INT	Ending Time Step
		<COMMENT LINE>
IPRGRD	INT	Frequency (# Iterations between Output) of grid output to unit 2

IPRPRE	INT	Frequency (# Iterations between Output) of pressure output to unit 6
IPRPLT	INT	Frequency (# Iterations between Output) of grid and q-vector output to units 2,3
IPRAER	INT	Frequency (# Iterations between Output) of aeroelastic output to units 6,13
		<COMMENT LINE>
JBDYLE	INT	J-direction Grid Line at Body Leading Edge (Fuselage Nose)
JBDYTR	INT	J-direction Grid Line at Body Trailing Edge (Fuselage Tail)
JWNGLE	INT	J-direction Grid Line at Wing Leading Edge
JWNGTR	INT	J-direction Grid Line at Wing Trailing Edge
KROOT1	INT	K-direction Grid Line at Wing Root (Top of Wing)
KROOT2	INT	K-direction Grid Line at Wing Root (Bottom of Wing)
		<COMMENT LINE>
IWHOLE	INT	Volume Grid Option - 0 - Read PLOT3D Format Surface Grid from Unit 20 and generate volume grid 1 - Read volume grid data from unit 29
ZFIR	INT	First L-direction Grid Spacing (Near Surface)
ZSTR	INT	Exponential Stretching Factor for L-direction grid.
		<COMMENT LINE>
REDFRE	REAL	Reduced frequency based on chord
PHAA	REAL	Phase Lag in degrees for pitch motion
PAXIS	REAL	Axes for pitch motion
PHAD	REAL	Phase Lag in degrees for plunge motion
FORAMP	REAL	Scale factor for Fourier coefficients
PRAT	REAL	Pitch rate for ramp motion
RAMANG	REAL	Maximum ramp angle of attack
MTYPE	INT	Flag to control motion type 0 - Steady case 1 - Unsteady sinusoidal case 2 - Unsteady ramp case
MODINP	INT	Flag to control modal data type 1 - Rigid forced motion data 2 - Modal forced motion data
NSPC	INT	Number of time steps per cycle (set = 5 for steady rigid case) [larger value = smaller time step taken]
		<If ITASK < 3 and MTYPE = 1 then read the variables marked *>
		<COMMENT LINE>*
TIPTWS*	REAL	Tip twist in degrees
TIPDIS*	REAL	Tip displacement in % root chord
BODDISN*	REAL	Body displacement (nose)
BODDIST*	REAL	Body displacement (tail)
		<IF ITASK > 2 THEN READ STRUCTURAL DATA (AEROELASTIC CASES) FROM THIS POINT FORWARD>
		<COMMENT LINE>
DYNPRE	REAL	Dynamic pressure of the free stream (physical units)
FREVEL	REAL	Free stream velocity (physical units)
PHYLEN	REAL	Root chord length in physical units
		<COMMENT LINE>
NTERMS	INT	Number of Chebyshev polynomials to be used in each coordinate direction for plate [Lovejoy ran into numerical difficulties with values greater than 8]

| NUMEIG	INT	Number of Eigensolutions sought (Also controls case type for structures)- -2 - Override ENSAERO Pressures with input distribution -1 - Run Plate Load Case (from input pressure distribution) 0 - Run Aeroelastic Load Case (ENSAERO Aeroelastic runs) >0 - Solve Eigenproblem for this many natural frequencies and mode shapes.
INRATE	INT	Number of Aerodynamic solution steps between structural equation solution (for ENSAERO Static Aeroelastic Cases. Helps stabilize numerical solution of aerodynamics).
		<COMMENT LINE>
COPRE(I=1,4)	REAL	Bi-linear pressure input on plate (used when NUMEIG = -1). 4 values, given at each corner. (Physical Units) [1 - Root TE, 2 - Tip TE, 3 - Tip LE, 4 - Root LE]
NETAPT	INT	Number of output points in eta-direction.
NXSIPT	INT	Number of output points in xsi-direction.
		<COMMENT LINE>
IBMAT(I=1,4)	INT	Plate boundary condition array (Root/Tip/LE/TE BC's) 1 - Free edge 2 - Simply supported edge 3 - Clamped edge [Therefore a wing would have the input values 3,1,1,1]
		<COMMENT LINE>
X(4)	REAL	Root Leading Edge x-coordinate (physical length units)
Y(4)	REAL	Root Leading Edge y-coordinate (physical length units)
BETA	REAL	c/4 line sweep angle in degrees.
ARATIO	REAL	Aspect Ratio (Based on whole wing)
TPRRATIO	REAL	Taper Ratio
		<COMMENT LINE>
NLAY	INT	Number of Lamina (layers) in plate
		<COMMENT LINE>
RESP1	INT	If = 1. sets all lamina thicknesses to be the same.
		<COMMENT LINE>
RESP2	INT	If = 1, then all lamina are of the same material.
		<COMMENT LINE>
		<For NLAY layers (I=1,NLAY), read the following lamina information, marked **
		<COMMENT LINE>**
ANGIN(I)**	REAL	Lamina angle with respect to the c/4 line of the plate (degrees).
		<If I=1 or if RESP1 not = 1 then read the next variable>
		<COMMENT LINE>**
THICKLAY(I)**	REAL	Lamina thickness (physical units).
		<If I=1 or if RESP2 not = 1 then read the material property array, each value on a separate line>
		<COMMENT LINE>**
PROP(1,I)**	REAL	E1 - Modulus in primary material direction
PROP(2,I)**	REAL	E2 - Modulus in-plane of plate orthogonal to direction 1
PROP(3,I)**	REAL	E3 - Modulus out-of-plane of the plate (through the thickness of the lamina)
PROP(4,I)**	REAL	G12 - Shear modulus (In-plane of plate)
PROP(5,I)**	REAL	G13 - Shear modulus
PROP(6,I)**	REAL	G23 - Shear modulus
PROP(7,I)**	REAL	υ_{12} - Poisson's ratio between direction 1 and 2.
PROP(8,I)**	REAL	υ_{13} - Poisson's ratio between direction 1 and 3.
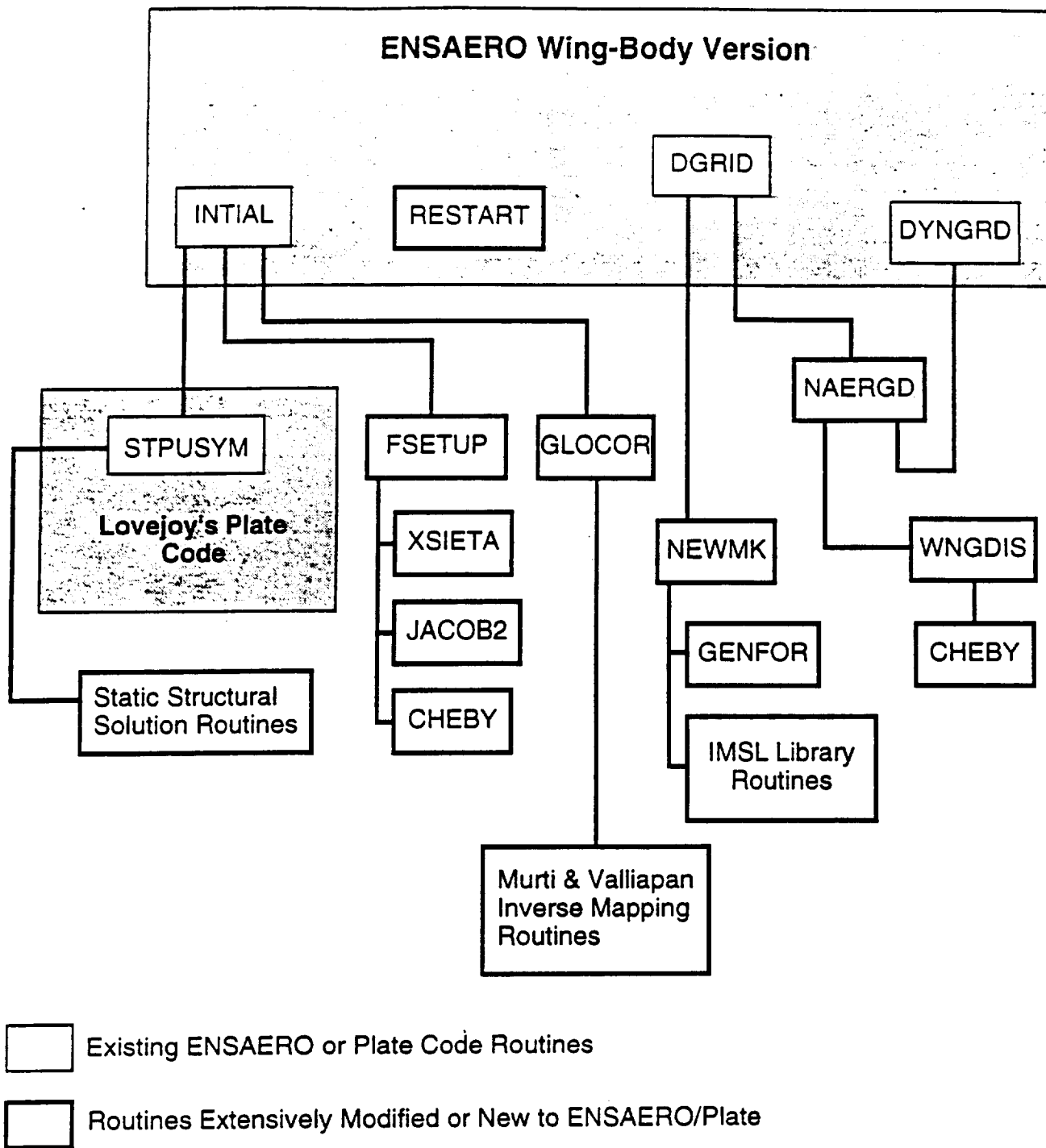
Figure 1. Structure of ENSAERO/Plate Showing Structural/Aerodynamic Interface

PROP(9,I)**	REAL	υ_{23} - Poisson's ratio between direction 2 and 3.
PROP(10,I)**	REAL	Density

Example Input Deck

```
SAMPLE INPUT FOR BLENDED WING-BODY CONFIGURATION (RAE RESEARCH WING)
RESTART 0(START) 1(STEADY TO UNSTEADY) 2(RESTART UNSTEADY), NUMBER OF ZONES
 2,                    1
GRID JGRD  KGRD  LGRD FOR THE LARGEST ZONE
     115,  81,  35
TASK  0 = STD, 1= UNS FROM STD, 2 = UNS, 3= STATIC AERO, 4= DYNAMIC AERO.
     3
FLOW VARIABLES  FSMACH,  GAMMA,  RE,     ALP,  TINF
          0.60,   1.4,  1.50E+06, 1.0,  -288.15
DISSIPATIONS   DIS2    DIS4
          0.25,   0.01
VISCOUS OPTIONS GLOBAL, (K-DIRECTION, L-DIRECTION, N=1, NBLK)
          0,      0,      0
TURBULANCE OPTION 0 = OFF, 1 = ON
          0
MODIFIED BALDWIN LOMAX
          0
TIME STEPS     START STOP
          201, 1200
PRINT FLAGS IPRGRD(grid)  IPRPRE (pre), IPLTFRE(plot), IAERPRE(AERO)
        3000,     100,    3000,     100
GRID DEPENDENT INDICES: J-BDY-LE, J-BDY-TR, J-WNG-LE, J-WNG-TR, K-ROOT1, KROOT2
           15,    115,    35,    85,   13,   69
ZGRID DATA IWHOLE, ZFIR, ZSTR 1.1
          0,   0.025, 1.15
REDFRE,PHAA,PAXIS, PHAD, FORAMP,  PRAT, RAMANG, MTYPE, MODINP NSPC(S=5)
0.5,  0.0, 2.5371, 0.0, 0.00837758, 0.01, 5.0,   0,   0,   100
DYNPRE FREVEL  PHYLEN
534.06 670.2  0.69579167
NTERMS NUMEIG INRATE
 8     0    5
CORNER PRESSURE ARRAY  NETAPT NXSIPT
 1.0 1.0 1.0 1.0   10  10
IBMAT (Edge BC's)  Root/Tip/LE/TE
 3  1  1  1
Xrtle  Yrtle BETA(c/4) ARATIO  TPRRATIO
1.815903 0.25 35.685335 5.2865765 0.359303
NLAY (Number of Lamina)
 1
RESP1
 1
RESP2
 1
ANGIN
0.0
THICKLAY (was 0.069579)
 0.03479
PROP Array (Structural Mat'l Data)
 1.512E9
```

1.512E9
1.512E9
5.815E8
5.815E8
5.815E8
0.3
0.3
0.3
5.366

Notes on Selecting Input Values/ Answers to Commonly Asked Questions

The following series of short topics will attempt to answer the most basic questions about how to set up ENSAERO/Plate runs. This is to help the user know which input data values should be considered when attempting to run the code in a certain way (for example: a static aeroelastic run).

WHAT DIMENSIONS SHOULD MY INPUT QUANTITIES BE IN?

All aerodynamic data should be non-dimensional. The free stream velocity, dynamic pressure and root chord length can be used to change any of the data back into physical units.

The Structural input should all be in consistent units. This means kg-N-m for the metric system and slugs-lb-ft for the English system. This is so no conversion constants are required. Avoid lb-mass, inches, etc. since they would require internal conversion to arrive at a consistent set of units.

PHYLEN is the reference length. It is the wing root chord distance (in general) and is used to scale all the aerodynamic geometry data. It is also used by the plate code as the distance between corners 1 and 4 of the trapezoidal plate.

HOW ARE YOUR ASPECT RATIO, TAPER RATIO, ETC. DEFINED?

The plate geometry is calculated from the following input quantities:

ARATIO - Defined as twice the plate span $(Y(3)-Y(4))$ squared divided by twice the plate area. This is a sort of wing-only based aspect ratio (it ignores the wetted area /additional span located inside the fuselage)

PHYLEN - Defined as $(X(1)-X(4))$, or the root chord length of the plate. (Physical units)

TPRRATIO - The taper ratio, or tip chord/root chord ($(X(2)-X(3))/(X(1)-X(4))$).

BETA - The quarter chord line sweep angle in degrees.

$X(4)$ and $Y(4)$ - The location of the Wing Root Leading Edge in physical units.

Note: The X and Y values indicated above refer to the 4 corners of the plate, where 1 is the root trailing edge, 2 the tip trailing edge, 3 the tip leading edge and 4 the root leading edge. This is Lovejoy's numbering convention.

SELECTING THE 'TIME' STEP

Generally the user will control the time step size by setting a single input parameter: NSPC. This parameter is used by ENSAERO to generate the computational step size for static cases, and

generates a true time step for dynamic cases. NSPC is inversely proportional to the time step (i.e. a small NSPC is a large time step). A value of NSPC=5 is good for most rigid cases.

For static aeroelastic cases, especially transonic cases, a larger NSPC may be needed, especially when starting the static aeroelastic case from a rigid aerodynamics-only solution. NOTE: while NSPC can be adjusted at the beginning of a restart, it is unwise to ever *increase* NSPC. If the solution is converging well, then decreasing the time step is unnecessary. If the calculation is appearing to have convergence difficulties, these will not, in general, be helped by reducing step size, since the solution already includes numerical problems. It is better in that case to start over with the higher NSPC choice, rather than continue the bad run.

WHEN HAVE I CONVERGED A STEADY-STATE RUN?

Steady-state runs (rigid or static aeroelastic cases) are considered converged when the residual calculation (output unit 11) drops 3 to 4 orders of magnitude. A slight oscillation may be seen about a general down trend when looking at the numbers in the file. This is acceptable as long as any transients (such as when restarting from a different case) die out quickly.

RUNNING A RIGID AERODYNAMIC CASE

A rigid aerodynamic case is usually run as an initial starting point for an aeroelastic run. By supplying a good initial guess for the flowfield to the static aeroelastic run, convergence is improved, and the rigid and flexible results may be compared later.

To run a rigid aerodynamic run, select IREAD = 0 (no restart) and ITASK = 0 (rigid steady-state aerodynamics). An NSPC value of 5 should be good for this run. Convergence should occur in less than a thousand iterations for most cases.

If an Unsteady rigid case is needed, it will be restarted from the steady rigid case just described by setting IREAD=1 and ITASK=1, and decreasing the step size by increasing NSPC to 100+ will be required. "Convergence" no longer applies in unsteady cases, since the calculation is made through time.

RUNNING A STATIC AEROELASTIC CASE

To run a static aeroelastic case, it is best to start from a steady-state rigid aerodynamic solution of the same type. Set the IREAD parameter to 1 (restart from different case) and the ITASK parameter to 3 (Static Aeroelastic run). NSPC should be around 10-20 for most cases although initially it may be necessary for it to be set higher to handle the transient changes in the aerodynamics caused by the initial structural motion (perhaps as high as 100).

Convergence of static aeroelastic cases is highly dependent upon the flow conditions. In the transonic regime, it can be very difficult to converge the combined aerodynamic/structural system, since shock motion and flow discontinuities are hard to resolve when the grid is in motion. Therefore, the parameter INRATE has been added to the structural input to control the number of aerodynamic iterations between solutions of the structural deflections. A setting of INRATE=1 means the structural equations are solved every iteration of the aerodynamic model and the grid moved accordingly. For INRATE=5, the grid will be held in a steady position for 5 aerodynamic iterations, allowing the aerodynamic model to steady any small transients caused by the last grid motion, then the structural equations are solved, the grid moved, and the aerodynamic model goes another 5 iterations. It will be possible to see the effect when looking at the residual calculations in output unit 11. Every INRATE steps, the residual should increase slightly, then decrease until the next movement of the grid.

RUNNING LOVEJOY'S PLATE CODE WITHOUT RUNNING ENSAERO

As long as all the necessary input data for ENSAERO is present, Lovejoy's code can be run by setting NUMEIG > 0. This tells the plate code to find NUMEIG number of eigensolutions for natural frequencies and modes of the composite plate. The code will not return to ENSAERO after the solution. The eigensolution will be output to the main output file (unit 6).

To solve an input bi-linear pressure distribution on the composite plate, set NUMEIG = -1 and input the pressure values (in physical units) at each of the four corners into the COPRE array. The code will not return to ENSAERO after solution. The output from this case will be in the main output file (unit 6).

OVERRIDING THE AERODYNAMIC MODEL WITH AN INPUT PRESSURE DISTRIBUTION

It is possible to use the simple pressure distribution given by the COPRE input array to override the calculated aerodynamic pressures in ENSAERO/Plate. This can be done by setting IREAD = 1 (Restart from a different case), ITASK = 3 (Static aeroelastic run) and setting NUMEIG = -2 (Override option). Set the number of iterations for the case to a low value (such as 5 or 10) since the structural solution will be that of the wing under the given bi-linear distribution. (Just make sure that INRATE allows the structural model to be solved and that the code goes past the first iteration).

The given pressure distribution will be used to set the deflection of the grid, but does not change the aerodynamic pressure data. This was done so the aerodynamic model could proceed without blowing up because of incorrect pressure information.

This option was included to help validate the methods used to solve static aeroelastic problems.

HOW MANY ITERATIONS CAN I GET AWAY WITH IN THE DEBUG QUEUE?

The author's experience is that less than 100 iterations are possible for a full static aeroelastic model under the 300 CPU second limit required to get in the debug queue. 80 iterations is about the limit. For a steady rigid aerodynamic case, 100 iterations may be possible.

If a full run is going to be done from scratch, run about 1000 iterations in the main queue (a time limit of 10800 seconds should be plenty) and wait for the job to return. It is far faster.

HOW MANY JOBS CAN I QUEUE ON EAGLE AT ONCE?

This varies from time to time as NASA Ames changes the queue system to accomodate new users or to find a better system for allowing more runs by different users to go through. While in the past it was possible to submit multiple jobs, currently users are limited to a single debug job and a single job in the main queue simultaneously. This may change in the future with little notice.

IV. Notes on Structural Model Subroutines

The following are subroutines which have either been extensively modified or added to the wing-body version of ENSAERO to create ENSAERO/Plate. The routines can be divided into roughly 5 groups:

(A) ENSAERO routines
(B) Lovejoy's composite plate code (modified, including added static plate loading solution routines)
(C) Murti's & Valliapan's Inverse Isoparametric Mapping Routines
(D) Structural Governing Equation Solver with Force/Displacement routines
(E) IMSL Library routines for solution of eigenproblems and linear systems of equations. (These are not listed, but are noted where called.)

ENSAERO Routines

INTIAL

Reads ENSAERO input data

Structural Code Calls: STPUSYM, GLOCOR, FSETUP

DGRID

Handles motion of the configuration-adaptive CFD grid.

Structural Code Calls: NEWMK, NAERGD

DYNGRD

Generates deflected volume grid according to surface grid deflections generated by NAERGD

Called by: NAERGD

RESTART

Stores and retrieves restart file information for ENSAERO. The data is stored as the combined PLOT3D format grid and q-vector files. Any necessary structural data is appended at the end of the fluid data.

Lovejoy's Plate Code

STPUSYM

This routine was the main program for Lovejoy's composite plate code. It has been modified and rewritten as a subroutine to ENSAERO. It controls generation of the stiffness and mass matrices of an input composite plate model.

Called by: INTIAL
Calls: GEOMETRY, ABD, MASSKST, BOUNDARY, FREQ

GEOMETRY

Calculates basic plate geometry.

Called by: STPUSYM
Calls:

ABD

Calculates Laminate A-B-D matrix values

Called by: STPUSYM
Calls: INPUT, Q_CALCULATE, QBAR_CALCULATE, ABD_CALCULATE, OUTPUT

INPUT

Reads in Lamina(layer) properties for the composite plate.

Called by: ABD
Calls:

Q_CALCULATE

Creates Lamina stiffness information. (See Lovejoy's thesis for additional details)

Called by: ABD
Calls:

QBAR_CALCULATE

Converts Lamina (layer) stiffness information to Laminate (plate) local coordinates.

Called by: ABD
Calls: ZERO2

ABD_CALCULATE

Assembles A-B-D terms for Laminate.

Called by: ABD
Calls: ZERO

ZERO

Sets very small (relatively) terms in the plate material matrix to zero.

Called by: ABD_CALCULATE
Calls:

ZERO2

Sets very small terms (relatively) in the lamina material matrix to zero.

Called by: QBAR_CALCULATE
Calls:

OUTPUT

Writes A,B, and D matrices for the laminate to an output file.

Called by: ABD
Calls:

MASSKST

Creates mass and stiffness matrices for unrestrained laminated plate. Also transforms coordinates to the solution domain.

Called by: STPUSYM
Calls:

BOUNDARY

Sets plate edge boundary conditions. The IBMAT Array indicates which of three possible conditions to apply to each edge (Clamped, Free, or Simply Supported). This information is used to generate "spring" stiffnesses which are later added to the unrestrained stiffness matrix to create a non-singular matrix which approximately (to a very close degree) matches the edge conditions (rather than following the Finite Element approach of eliminating known displacements from the structural system of equations).

Called by: STPUSYM
Calls:

FREQ

Solves the Eigenproblem for natural frequencies and modes of the plate. Also assembles the "restrained" stiffness matrix (Adds "Spring" stiffness matrix [BC's] to the unrestrained plate stiffness matrix).

Called by: STPUSYM
Calls: DGVLSP, DGVCSP (Both are IMSL Library calls) and SFPRE, OUTEVAL, OUTEVEC

OUTEVAL

Outputs eigenvalue (natural frequency) of the plate.

Called by: FREQ
Calls:

OUTEVEC

Outputs eigenvectors (mode shapes) of the plate.

Called by: FREQ
Calls:

SFPRE

Calculates pressure matrix for static equilibrium solution of plate.

Called by: FREQ
Calls: CHEBY, KUFSLV

KUFSLV

Solves static equilibrium of plate given a simple pressure distribution.

Called by: SFPRE
Calls: LFTSF, LFSSF (IMSL Library routines) and FPLATE, UPLATE

FPLATE

Generates generalized force vector on plate from given pressure data.

Called by: KUFSLV
Calls:

UPLATE

Outputs displacement data for plate on a uniform (η, ξ) grid.

Called by: KUFSLV
Calls: CHEBY, XYPT

XYPT

Gives X, Y given ETA, XSI, Corner Coordinates

Called by: UPLATE
Calls:

Murti's & Valliapan's Inverse Mapping Routines

CXY2RS

Calculates the local coordinate (η, ξ) of a point (x,y) where (η, ξ) are defined from -1 to 1 in each direction. This is done conceptually by drawing a straight line from one corner of the domain in (x,y) through the point of interest. In (η, ξ)-space this is a parabola of known equation form. If the parabola is defined over the entire possible -1 to 1 value of either ξ or η then a line search is conducted to find the precise point of interest in (η, ξ). (At least one of the 4 corners of the domain can be used to choose such a line, if necessary by interchanging the axes and renumbering the nodes).

Called by: GLOCOR
Calls: TRANSF, BISECT

TRANSF

Renumbers nodes and interchanges axes if necessary to define the line search over a -1 to 1 range.

Called by: CXY2RS
Calls:

BISECT

Determines the $\left(\xi, \eta\right)$ coordinate by bisectioning the defined line

Called by: CXY2RS
Calls: CALNQ, FIND, MVSHAP

FIND

Finds the corresponding value of eta for a given set of input data: XSI, XX(2,9).

Called by: BISECT
Calls:

CALNQ

Calculates nodal quantity $\left(\Phi_x, \Phi_y\right)$ for a given shape function H in an element with a variable number of nodes.

Called by: BISECT
Calls:

MVSHAP

Shape Function Routine for Murti&Valliapan's Inverse Isoparametric Mapping Routines.

Called by: BISECT
Calls:

Force/Displacement Solution Routines

GLOCOR

Finds the transformed coordinates (η, ξ) of the CFD surface grid for a wing. Reads in the grid data from the restart file (written by GENGRD) then arranges the data as input for calling Murti's and Valliapan's mapping routines. *NOTE: THE MAPPING ROUTINES AND LOVEJOY'S PLATE CODE DO NOT USE THE SAME AXES NOTATION. THE NODES MUST BE RENUMBERED AND THE AXES FLIPPED TO BE CONSISTENT WITH THE PLATE CODE NOTATION.*

Called by: INTIAL
Calls: CXY2RS

FSETUP

Sets up the Aerodynamic Matrix (the matrix which converts discrete aerodynamic pressure data into generalized forces based on the Ritz functions of Lovejoy's Plate Code.)

Called by: INTIAL
Calls: XSIETA, CHEBY, JACOB2

CHEBY

Evaluates a Chebyshev polynomial of order I at a given point X.

Called by: FSETUP, WNGDIS, SFPRE, UPLATE
Calls:

JACOB2

Evaluates the Jacobian of the Global transformed coordinate system to the local bilinear coordinate system mapping.

Called by: FSETUP
Calls:

NEWMK

Solves the static or dynamic equilibrium equations for the structural model using forces generated by the Aerodynamic model. The Newmark algorithm is used for dynamic aeroelastic problems.

Called by: DGRID
Calls: LFTSF, LFSSF (IMSL Library Routines) and GETFOR, DISCHK

GETFOR

Generates the Plate Generalized Force Vector from the Aerodynamic Model Pressures.

Called by: NEWMK
Calls:

WNGDIS

Calculates Wing grid deflections from the structural governing equation solution and passes them to NAERGD. *NOTE: THE GRID IS DEFLECTED BASED ON THE CHANGE IN DISPLACEMENT FROM THE LAST GRID POSITION. THE ORIGINAL GRID INFORMATION IS NOT RETAINED. IF IT IS NEEDED IT MUST BE RE-READ FROM THE GRID OR RESTART FILES.*

Called by: NAERGD
Calls: CHEBY

NAERGD

Replaces ENSAERO's AERGRD routine. Calculates the change in the wingbody surface grid and calls the volume grid generator.

Called by: DGRID
Calls: WNGDIS, DYNGRD

XSIETA

Calculates (η, ξ) coordinate of bi-linear Panel given the local (u,v) coordinate and the values of the 4 corners.

Called by: FSETUP
Calls:

DISCHK

Checks output displacement vector and filters out relatively small result terms.

Called by: NEWMK
Calls:

V. Force Vector Generation for Composite Plate Code

AERODYNAMIC MODEL:

The Aerodynamic Model (ENSAERO) supplies two main pieces of information required to generate the generalized force vector for use with Lovejoy's equivalent plate code: A) the Surface Grid locations and B) the pressure coefficients at those locations.

STRUCTURAL MODEL:

The Structural Model supplies the displacement shape functions upon which the generalized forces are based.

GOVERNING STRUCTURAL EQUATIONS

For the static problem the governing equations of motiion may be written as:

$$[K]\{C\} = \{F\}$$

where $[K]$ is the stiffness matrix; $\{C\}$ is the vector of generalized displacements and $\{F\}$ is the generalized force vector. Following Eldred, the generalized force terms may be written as

$$Q_i = \iint_\Omega p(x,y)\gamma_i(x,y)dxdy$$

where Q_i is the force term corresponding to the i-th displacement shape function $\gamma_i(x,y)$, and $p(x,y)$ is the pressure field on the surface of the structure. Ω is the wing surface area.

The generalized displacements C_i are related to the actual displacement field by

$$\omega(x,y) = \sum_i \gamma_i(x,y) \cdot C_i$$

GLOBAL COORDINATE TRANSFORMATION

For convenience, I will use the global coordinate transformation that Eldred (and Lovejoy as well) used, to the (η, ξ) system described by

$$x(\eta, \xi) = \sum_{i=1}^{4} N_i(\eta, \xi) \cdot x_i$$

$$y(\eta, \xi) = \sum_{i=1}^{4} N_i(\eta, \xi) \cdot y_i$$

where

$$N_i = \frac{1}{4}(1 + \xi\xi_i)(1 + \eta\eta_i)$$

and (x_i, y_i) and (η_i, ξ_i) are the coordinates of the four corner points of the wing in their respective coordinate systems. Note the domain of the transformed coordinates ranges from -1 to 1 in both directions.

BILINEAR PRESSURE REPRESENTATION

Again following Eldred, the wing pressure distribution may in general be represented as

$$p(\eta, \xi) = \sum_j \beta^j(\eta, \xi) \cdot a^j$$

where the β^j are chosen interpolation functions and the α^j are the generalized coefficients. For the bilinear interpolation method, trapezoidal panels are placed between sets of four known discrete pressures, and the pressure at any interior point of that panel is given (LOCALLY) by

$$p(u,v) = \frac{1}{4} \cdot \begin{bmatrix} (1-u)(1-v) \\ (1-u)(1+v) \\ (1+u)(1-v) \\ (1+u)(1+v) \end{bmatrix}^T \cdot \begin{Bmatrix} b_{00} \\ b_{01} \\ b_{10} \\ b_{11} \end{Bmatrix} = \{R\}^T \cdot \{a\}$$

Note the pressures are given in terms of panel local coordinates (u,v) which, like the (η, ξ) global system has a domain ranging from -1 to 1 in both coordinate direction-s. The $\{a\}$ terms are the discrete pressures at the four panel corners.

DISPLACEMENT SHAPE FUNCTIONS

The displacement shape functions γ_i in Lovejoy's plate code are the Chebyshev polynomials. They are given for computational ease in the (η, ξ) coordinate system described above. The shape functions are

$$\gamma_i(\eta, \xi) = T_j(\eta) \cdot T_k(\xi)$$

where i is an index which is a function of j and k, and j and k are the order of the Chebyshev polynomials. The Chebyshev polynomials are

$$T_0(s) = 1$$
$$T_1(s) = s$$
$$T_i(s) = 2sT_{i-1}(s) - T_{i-2}(s) \text{ for } i > 1$$

GENERALIZED FORCE VECTOR EQUATIONS

The generalized force terms Q_i given above may now be written for our case more specifically as

$$Q_i = \sum_l A_{il} \cdot a_l$$

where a_l are the surface pressures given at the ENSAERO aerodynamic surface grid points and A_{il} are terms in what I shall call the aerodynamic matrix, $[A]$. The aerodynamic matrix terms are created from the integration of the pressure field over the wing surface area. For the bilinear pressure representation, and substituting in the Chebyshev polynomial-based displacement shape functions, leads to the following equation

$$[A] = \sum_{j=0}^{j_{max}} \sum_{k=0}^{k_{max}} \sum_{PANELS} \int_{-1}^{1}\int_{-1}^{1} \{R(u,v)\} \cdot T_j(\eta) \cdot T_k(\xi) \cdot |J_1(\eta,\xi)| \cdot |J_2(u,v)| \cdot du\,dv$$

where $\{R(u,v)\}$ is the bilinear interpolation function vector, T_j and T_k are the Chebyshev polynomials, $|J_1(\eta,\xi)|$ is the Jacobian of the transformation between the original coordinates (x,y) and the transformed global coordinates (η, ξ), and $|J_2(u,v)|$ is the Jacobian of the transformation between the transformed global coordinates (η, ξ) and the local panel coordinates (u,v). Note that the matrix $[A]$ is independent of the discrete pressures, and is solely a function of the number of displacement shape functions chosen and the geometry of the ENSAERO aerodynamic grid on the wing surface. Note also that it will have a size given (roughly) by the number of displacement shape functions multiplied by the number of wing surface grid points.

The integral for each panel will generate 4 terms, which are assembled into the global aerodynamic matrix by adding the result to the column location corresponding to the correct node of the panel and the row indicating the displacement shape function considered.

INTEGRATION OF THE BILINEAR PANELS

The Bilinear panels shall be integrated by Gaussian Quadriture, using n=2 points in each direction for a total of 4 evaluations of the integrand of the above equation for each node of each panel per each shape function. Luckily, this overhead must be performed only once per run, after which the generalized pressures may be updated by a simple matrix multiplication of the discrete surface pressures.

ASSEMBLY OF THE GLOBAL FORCE VECTOR $\{F\}$

The pressures on the wing only are applied to the translational degrees of freedom of the plate. By assuming a flat plate, they may additionally be assumed to act only in the vertical direction (a thin airfoil assumption). It will be convenient to separate the upper and lower surfaces of the wing into separate domains for the purposes of generating the aerodynamic matrix given above.

REFERENCES

ELDRED:

Eldred, Lloyd B., "Sensitivity Analysis of the Static Aeroelastic Response of a Wing," Ph.D. Dissertation, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, February 1993.

LOVEJOY:

Lovejoy, Andrew E., Preliminary Notes on M.S. Thesis, 1994.