JOURNAL OF COMPUTATIONAL PHYSICS **118**, 73–85 (1995)

Application of Alternative Multidisciplinary Optimization Formulations to a Model Problem for Static Aeroelasticity

GREGORY R. SHUBIN

Boeing Computer Services, P.O. Box 24346, M/S 7L-21, Seattle, Washington 98124-0346

Received February 17, 1994

A new model problem for static aeroelasticity is introduced and used to illustrate several alternative approaches for formulating multidisciplinary design optimization problems. The alternatives are distinguished by the kind of analysis problem feasibility that is maintained at each optimization iteration. In the familiar "multidisciplinary feasible" approach, the full multidisciplinary analysis problem is solved at each iteration of the optimizer. At the other end of the spectrum is the "all-at-once" approach where none of the individual analysis discipline equations is guaranteed to be satisfied until optimization convergence. In between lie other possibilities that amount to enforcing feasibility of the individual analysis disciplines at each optimization iteration, but allowing the coupling between the disciplines to be incorrect until optimization convergence. Results are given for these three approaches applied to the new model. In general, delaying feasibility until optimality reduces the total amount of computing work.

1. INTRODUCTION

The purpose of this paper is to apply three alternative approaches for multidisciplinary design optimization (MDO), introduced in [1], to a model problem for static aeroelasticity. By MDO we mean the coupling of two or more analysis disciplines with numerical optimization. The reader unfamiliar with this relatively new field may wish to consult the proceedings of three recent symposia [2–4] on MDO.

The new model for aeroelasticity was introduced in [5], and much of the material in that unpublished report is repeated here for completeness. The physical situation is depicted in Fig. 1. The model is an extension of the one-dimensional duct flow model, previously employed in studying aerodynamic optimization [6–8], to allow for flexibility in the walls of the duct, thus introducing "structural" effects. Since the solution for the flow is influenced by changes in duct shape due to this flexibility and since the displacement of the wall is caused by pressures generated by the flow, the model exhibits simple static aeroelastic behavior. By analogy with aeroelasticity for a flexible aircraft wing, we refer to the analysis disciplines involved as aerodynamics (or flow) and structures.

We emphasize that our goal is to highlight the trade-offs among the various MDO formulations in the context of a concrete, but rather simple, model problem. Because the model investigated here is radically simpler than real MDO problems, we cannot hope to reach any general conclusions about which (if any) of the formulations would be appropriate for real MDO problems. However, we do examine many questions that should be asked in making such a determination.

The outline of the paper is as follows. In Section 2, we present both the analysis and design problems for the flexible duct model. In Section 3 we review the multidisciplinary optimization formulations. In Section 4 we present results obtained by applying the formulations to the model. In Section 5 we make some concluding remarks.

2. FLEXIBLE DUCT MODEL PROBLEM

2.1. *Continuous Analysis Problem*

We now present the continuous analysis formulation for the flexible duct model. Here and in what follows, the aerodynamics (or flow) discipline is denoted by subscript 1, and the structures discipline by subscript 2.

In [6] we showed how the steady flow of an inviscid fluid in a duct of variable cross sectional area $A(\xi) + d(\xi)$, governed by the Euler equations, can (under certain circumstances) be reduced to the single nonlinear ordinary differential equation

$$w_1 \equiv f_\xi + g = 0, \tag{1}$$

where

$$f(v) \equiv v + \overline{H}/v, \quad g(v, \xi) \equiv \frac{A_\xi + d_\xi}{A + d}(\overline{\gamma} v - \overline{H}/v),$$

$v(\xi)$ is the fluid velocity, ξ is distance along the duct, and $\overline{\gamma}$ and $\overline{H}$ are given constants. Here, the subscript ξ means differentiation with respect to ξ. The specified boundary values $v(0)$ and $v(1)$ are chosen so that the (weak) solution of (1) contains a shock. The derivation of this part of the model and much more detail about it can be found in [6].

In the above formulation, the area function A is thought of as the specified "aerodynamic" shape of the duct, not including

0021-9991/95 $6.00

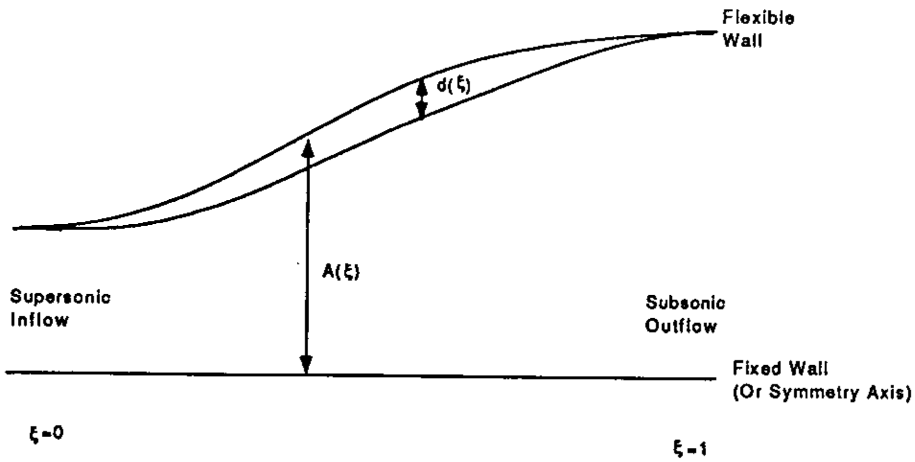

FIG. 1. Model problem geometry.

flexibility effects. The area function d gives the change in aerodynamic shape due to flexibility of the upper wall. Note that $d < 0$ in Fig. 1. We assume that the area change d, which is like a "displacement" in structural analysis, is governed by the beam equation (from beam theory)

$$w_2 \equiv [sd_{\xi\xi}]_{\xi\xi} - p(v) = 0. \tag{2}$$

Here, $s(\xi)$ is the given beam stiffness, usually designated as EI in the engineering literature, where E is Young's modulus and I is the cross sectional area of the beam. The forcing function p is analogous to pressures in aerodynamics and in reality turns out to be a function of both $v(\xi)$ and $A(\xi)$ for the model (1). However, to simplify our model, we take p to be only a function of the velocity v, namely $p(v) = -v$. We assume that the boundary conditions on (2) correspond to those of a cantilevered beam, namely, $d(0) = 0$, $d_\xi(0) = 0$, $d(1) = 0$, $d_\xi(1) = 0$.

2.2. *Discrete Analysis Problem*

Equations (1) and (2) are assumed to be discretized on two different computational grids, called the aerodynamic (or flow) grid and the structures grid.

We first consider the discretization of (1). Let the ξ-coordinate be discretized by a uniform, cell-centered "aerodynamic" grid with centers at $\xi_m = (m - 1/2)\Delta\xi_m$, $\Delta\xi_m = 1/M$, where M is the number of unknown (interior) grid values. Let V_m represent a piecewise constant approximation to v on each grid cell. Then, a conservative difference scheme for (1) is given by

$$W_{1,m} \equiv \mathcal{F}_{m+1/2} - \mathcal{F}_{m-1/2} + (\Delta\xi_m)\mathcal{G}_m = 0 \tag{3}$$

for $m = 1, ..., M$. Here the source term $\mathcal{G}_m$ involves V and A, defined on the flow grid, and the displacements D which will subsequently be defined on the structures grid; we discuss the evaluation of this term later. The boundary conditions on V are $W_{1,0} = V_0 - v(0) = 0$ and $W_{1,M+1} = V_{M+1} - v(1) = 0$.

To complete the discretization of (1) it remains to prescribe the fluxes $\mathcal{F}_{m+1/2}$ as functions of V_m and V_{m+1}. Three such prescriptions, $\mathcal{F}^{G}$, $\mathcal{F}^{EO}$, and $\mathcal{F}^{AV}$, corresponding to the Godunov, Engquist-Osher, and artificial viscosity methods for numerically approximating hyperbolic conservation laws, are given in [6]. The trade-offs between the three choices are also discussed in [6]; here, for simplicity, we use only the artificial viscosity flux, defined as $\mathcal{F}^{AV}_{m+1/2} = \frac{1}{2}[f(V_{m+1}) + f(V_m) - (V_{m+1} - V_m)]$. By making this choice we have sidestepped some issues regarding AAO optimization involving difference schemes of low continuity; see [6]

Next we consider the discretization of (2). Let the ξ-coordinate be discretized by a uniform, point-centered "structures" grid $\xi_n = n\Delta\xi_n$, $\Delta\xi_n = 1/(N + 1)$, where N is the number of unknown interior grid values. Let D_n represent an approximation to d at each grid point and $S_n = s(\xi_n)$. Then a simple finite difference approximation for (2) is given by

$$W_{2,n} \equiv \delta^2[S_n\delta^2D_n] - (\Delta\xi_n)^4P_n = 0, \tag{4}$$

for $n = 2, 3, ..., N - 1$, where $\delta^2D_n \equiv D_{n+1} - 2D_n + D_{n-1}$ is the usual centered, second difference operator. The evaluation of P_n on the structures grid involves V_m on the flow grid; we discuss its evaluation later. The boundary conditions are $W_{2,0} = D_0 = 0$, $W_{2,N+1} = D_{N+1} = 0$, $W_{2,1} = -3D_0 + 4D_1 - D_2 = 0$, and $W_{2,N} = D_{N-1} - 4D_N + 3D_{N+1} = 0$. The latter two conditions are second-order accurate, one-sided approximations of the derivative boundary conditions on d.

Once the discretization has been made, we are faced with solving a system of nonlinear algebraic equations. Let $U_1 \equiv [V_0, V_1, ..., V_{M+1}]$ be the vector of unknown analysis variables (velocities) on the aerodynamics grid, and $U_2 \equiv [D_0, D_1, ..., D_{N+1}]$ be the vector of unknown analysis variables

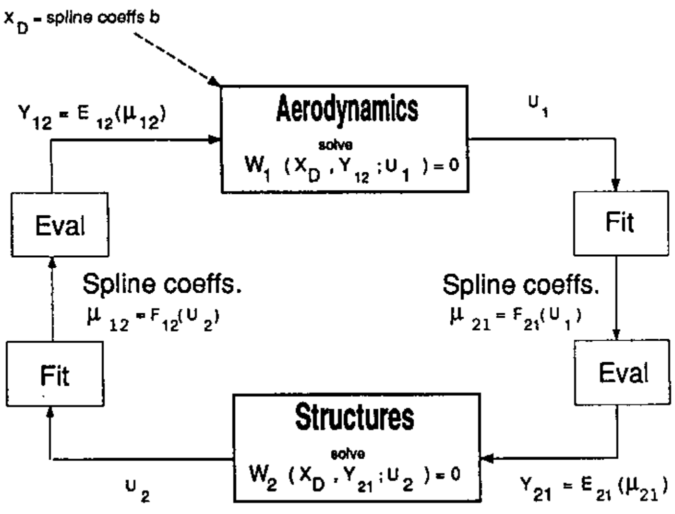

FIG. 2. Multidisciplinary feasibility is achieved when the aerodynamics equations are satisfied, the structures equations are satisfied, and the interdisciplinary mapping is correct.

(displacements) on the structures grid. The system we need to solve is

$$W_1(X_D, Y_{12}; U_1) = 0,$$
$$W_2(X_D, Y_{21}; U_2) = 0, \quad (5)$$

$$Y_{12} - G_{12}(U_2) = 0,$$
$$Y_{21} - G_{21}(U_1) = 0. \quad (6)$$

We call (5), (6) the multidisciplinary analysis, or MDA, system. Here, W_1 represents the system of discrete flow equations $W_{1,m}$, $m = 0, \ldots, M + 1$ and W_2 represents the system of discrete structures equations $W_{2,n}$, $n = 0, \ldots, N + 1$. X_D represent physical parameters like spline coefficients defining the duct area A and structural stiffness s; these are fixed for the analysis problem, but later they are possible optimization variables for the design problem. (In this paper, we examine a design problem in which only A is controlled by design variables and s is fixed.) Our notation here mimics that introduced in [1].

The terms $G_{12}(U_2)$ and $G_{21}(U_1)$ in (6) represent the evaluation of the displacements D on the flow grid, and the pressures P on the structures grid, respectively. We call these *interdisciplinary mappings,* since they map the outputs of one discipline into the inputs to another discipline. In practice, each of these mappings is likely to be the composition of two other mappings, $G_{ij} = E_{ij}(F_{ij}(U_j))$; see Fig. 2. Here $\mu_{ij} = F_{ij}(U_j)$ is a "fit" of the output of discipline j to obtain, say, some spline coefficients μ_{ij}. Note the convention that double subscripts ij denote a "to–from" relationship. The fit F_{ij} may be either an interpolation or an approximation. In practical applications it is likely to be the latter, since this reduces the amount of information transmitted between disciplines and, for the "individual discipline feasible" formulation of MDO (described later), thereby reduces the number of design sensitivity calculations required in optimization [1]. The mapping E_{ij} is an evaluation of the spline representation from discipline j into a form suitable for use by discipline i. Both F and E may contain additional computations that, for example, calculate loads from pressures. They could also both depend on the design variables X_D, although we do not assume such dependence here.

In this particular flexible duct example, the interdisciplinary mappings serve the simple purpose of moving back and forth between the flow grid and the structures grid. Specifically, the evaluation of the term $\mathcal{G}_m$ in (3), which approximates $g(v, \xi)$ in (1), proceeds as follows. The factor $(\bar{\gamma} v - \bar{H}/v)$ is evaluated simply as $\bar{\gamma} V_m - \bar{H}/V_m$. It remains to approximate $(A_\xi + d_\xi)/(A + d)$. We assume that the area function A is given by a piecewise cubic spline described in the B-spline basis, with coefficients b (that are design variables later), and that D_n from the solution of (4) has also been fit with a piecewise cubic spline with coefficients μ_{12} in the manner described above for interdisciplinary mappings. Then we can either evaluate $(A_\xi + d_\xi)/(A + d)$ at ξ_m on the flow grid directly from the two splines, or we can evaluate A_m and D_m from the splines, and employ a difference formula to approximate A_ξ and d_ξ; we do the latter. Similarly, the evaluation of P_n in (4) is obtained by fitting a spline with coefficients μ_{21} to $-V_m$ on the flow grid and evaluating the spline at ξ_n on the structures grid.

In discussing the formulations in Section 3, we will say that *individual discipline feasibility* has been achieved when (5) is solved, irrespective of where the inputs Y from the other disciplines come from. (They may be guessed or estimated, for example.) We will say that we have *multidisciplinary feasibility* if, in addition to (5) the interdisciplinary coupling equations (6) are satisfied. In other words, multidisciplinary feasibility means that the MDA system is satisfied. In terms of Fig. 2, multidisciplinary feasibility means that equilibrium has been reached in the graph representing the flow of information. The observation that it is possible to have individual discipline feasibility without multidisciplinary feasibility is key to the IDF formulation, presented later.

To achieve multidisciplinary feasibility, we need a method to solve the MDA system of nonlinear algebraic equations (5), (6). Two obvious methods are fixed point iteration and Newton's method. In fixed point iteration, we (for example) guess U_2, solve $W_1 = 0$ for U_1, use these values of U_1 to solve $W_2 = 0$ for U_2, and then iterate. This assumes that we have individual discipline solvers for $W_1 = 0$ (flow) and $W_2 = 0$ (structures) already available. For our model, we assume that the flow solver uses Newton's method with iteration matrix $\partial W_1/\partial U_1$, and the structures solver simply forms and solves the linear discrete structures equations with coefficient matrix $\partial W_2/\partial U_2$.

In Newton's method for the coupled equations (5), (6), we need to form and solve the following linear system:

$$\begin{bmatrix} \frac{\partial W_1}{\partial U_1} & \frac{\partial W_1}{\partial U_2} \\ \frac{\partial W_2}{\partial U_1} & \frac{\partial W_2}{\partial U_2} \end{bmatrix} \begin{bmatrix} \Delta U_1 \\ \Delta U_2 \end{bmatrix} = \begin{bmatrix} -W_1 \\ -W_2 \end{bmatrix}. \tag{7}$$

The off-diagonal blocks $\partial W_1/\partial U_2$ and $\partial W_2/\partial U_1$ represent the coupling from structures to flow, and from flow to structures, respectively. Note that, to compute these, it is necessary to differentiate through the interdisciplinary mappings G. This means that the fitting and evaluation processes that ordinarily constitute these mappings must be chosen to be differentiable. Later, we call this the "Newton-grid" method since it uses Newton's method to solve for the grid-based variables U_i. Also mentioned later is the occasional need to damp the Newton iteration to get it to converge.

A third, less obvious option for solving (5), (6) is to solve simultaneously, using (say) Newton's method, for the spline coefficients μ_{12} and μ_{21} defining the interdisciplinary mappings. In this option, the equations we need to solve are

$$\begin{aligned} C_{12} &= \mu_{12} - F_{12}(U_2(X_D, E_{21}(\mu_{21}))) = 0, \\ C_{21} &= \mu_{21} - F_{21}(U_1(X_D, E_{12}(\mu_{12}))) = 0. \end{aligned} \tag{8}$$

Here the U_j are thought of as solution operators. For use later, we define $\bar{\mu}_{ij} \equiv F_{ij}(U_j(X_D, E_{ji}(\mu_{ji})))$ as the overall operator that provides the spline coefficients from a discipline's (output) fitter as a function of the coefficients to the discipline's (input) evaluator. Again, X_D is fixed for now but we include it for reference later. We then rewrite (8) as

$$\begin{aligned} C_{12} &= \mu_{12} - \bar{\mu}_{12} = 0, \\ C_{21} &= \mu_{21} - \bar{\mu}_{21} = 0. \end{aligned} \tag{9}$$

The Newton step is obtained by solving the linear system

$$\begin{bmatrix} I & -\frac{\partial \bar{\mu}_{12}}{\partial \mu_{21}} \\ -\frac{\partial \bar{\mu}_{21}}{\partial \mu_{12}} & I \end{bmatrix} \begin{bmatrix} \Delta \mu_{12} \\ \Delta \mu_{21} \end{bmatrix} = \begin{bmatrix} -C_{12} \\ -C_{21} \end{bmatrix}. \tag{10}$$

Later, we call this the "Newton-spline" method. This approach is closely related to the individual discipline feasible (IDF) method for multidisciplinary design optimization discussed later.

Note that the coefficient matrices in (7) and (10) are fundamentally different. The matrix in (7) involves derivatives of *residuals* with respect to variables appearing in the equations. The matrix in (10) involves the derivatives of *solutions* of equations with respect to inputs to those equations; these latter partial derivatives are far more difficult to obtain. Their calculation is discussed in Section 3.5 on gradient computations.

2.3. *Continuous Design Problem*

The formulation of the continuous design problem boils down to the specification of an objective function, constraints, and design variables. There is by no means any consensus in the literature concerning how to choose these for aeroelastic optimization.

In aerodynamic optimization alone (no structural effects), the typical objective is to come as close as possible to some specified pressure distribution, or to minimize drag. (The latter objective is infrequently used today due to the difficulty of computing drag accurately in transonic flow.) The design variables are shape parameters, say, spline coefficients, that describe either the aerodynamic shape or changes to some base aerodynamic shape. Constraints may be used to prohibit undesirable shapes or flows. For the duct flow model ((1), with $d \equiv 0$), the problem investigated in [6] was to find $A(\xi)$, $A_\xi > 0$, such that $v(\xi)$ satisfies (1) and $\int_0^1 [v(\xi) - \hat{v}(\xi)]^2 \, d\xi$ is minimized. Here, $\hat{v}$ is a specified velocity distribution that we want to come as close to as possible. Some interesting further investigations carried out using this model may be found in [9]. While those investigations suggest that bounds on A_ξ are required for a well-posed optimization problem, we have not explicitly enforced such bounds in this paper.

In structural optimization alone, the objective is typically to minimize weight. The design variables are usually sizes or strengths of structural members, but they can also be shapes. Constraints may specify maximum allowable deflections, maximum stresses, minimum allowable sizes, etc. A suitable optimization problem for the beam equation (2) with the forcing function p specified, would be to find the stiffness distribution $s(\xi)$ such that displacement d was less than some maximum value and weight was minimized. According to [10], a good measure of weight is $\int_0^1 s^{1/2}(\xi) \, d\xi$, where we have assumed that Young's modulus is constant and the moment of inertia is spatially varying.

When the disciplines of flow and structures are combined, there is a large variety of possible combinations of objective, constraints, and design variables, but only a few of the combinations make sense. Ultimately, for real aeroelastic optimization of aircraft, we expect that there will be both aerodynamic and structural design variables and constraints and that the objective function will be related to some measure of overall aircraft performance, like direct operating cost.

For the present purposes of demonstrating the behavior of our model, we choose a particularly simple problem, namely, a slight modification of the one investigated in [6]. Thus, our design problem is:

Find $A(\xi)$, $A_\xi(\xi) > 0$, such that $v(\xi)$ and $d(\xi)$ satisfy (1) and (2) and $\int_0^1 [v(\xi) - \hat{v}(\xi)]^2 \, d\xi$ is minimized.

For this paper, it is assumed that the structural stiffness $s(\xi)$, which is a potential design variable in more complex design problems using this model, is fixed. Thus, in this problem we are essentially doing aerodynamic optimization alone, but the analysis problem includes aeroelastic effects.

2.4. *Discrete Design Problem*

We assume that a desired (or goal) velocity distribution $\hat{V}_m$ is given for each computational cell. Then the discrete design problem is:

> Find b (spline coefficients describing $A(\xi)$), such that (5), (6) is satisfied and $F = \sum_{m=1}^{M} [V_m - \hat{V}_m]^2$ is minimized.

It is this discrete design problem to which the formulations presented in the next section will later be applied. Note that the objective function F should not be confused with the fitting functions F_{12} and F_{21}.

As discussed in [11], it is sometimes desirable to fit the discrete computed and goal velocities with smooth functions (say splines), and to compute the objective function as the integral of the squared difference between the two splines. This tends to make the objective function smoother, which can make the optimization work better in the presence of nonuniform grids and steep solution gradients or shocks. This stratagem will not be required for the present purposes.

3. ALTERNATIVE OPTIMIZATION FORMULATIONS FOR MDO

We now review the three basic approaches presented in [1] for formulating multidisciplinary design problems as optimization problems. This material is repeated here from that reference. We assume that the discretization has been made and we are trying to solve the discrete design problem.

The key issue in the alternative MDO formulations is the kind of feasibility that is maintained at each optimization iteration. The term feasibility here means satisfaction of an equation or a set of coupled equations. In the multidisciplinary feasible (MDF) approach, complete multidisciplinary analysis (MDA) problem feasibility is maintained. Recall that, for our model, this means that the discrete aerodynamics equations are satisfied, the discrete structures equations are satisfied, and the interdisciplinary coupling is correct. Mathematically, Eqs. (5), (6) are simultaneously satisfied.

In the individual discipline feasible (IDF) approach, individual discipline feasibility is maintained at each optimization iteration. For our model, this means that each set of discrete equations (5) are satisfied, but the interdisciplinary coupling (6) is not correct until optimization convergence. In an IDF aeroelastic optimization, at each optimization iteration we have a "correct" aerodynamic analysis and a "correct" structural analysis. However, it is only at optimization convergence that the pressures predicted by the aerodynamic analysis correspond to the loads input to the structures and that the displacements predicted by the structural analysis correspond to the geometry input to the aerodynamics.

In the all-at-once (AAO) approach, all of the analysis variables are optimization variables and all of the analysis discipline equations are optimization constraints. Thus, feasibility in AAO, even for the single discrete equations within a discipline, is guaranteed only at optimization convergence. The optimizer assumes the responsibility for eventually achieving multidisciplinary feasibility.

A drawback of the IDF and AAO approaches is that no useful information may be available if the optimization is stopped short of convergence, since the analysis equations are not necessarily satisfied. By contrast, stopping MDF short of optimization convergence could yield useful design improvement.

We use the convention that all variables controlled by the optimizer are denoted by X with certain subscripts; this convention allows one to immediately identify what is, and what is not, an optimization variable. The original design variables, which are the spline coefficients b describing the duct area $A(\xi)$, are denoted X_D (for design) and are optimization variables in all three formulations. The AAO and IDF formulations induce additional optimization variables as part of their definitions. In the AAO formulation, these additional optimization variables are the optimizer's estimates of the discrete velocities on the flow grid, designated X_{U_1}, and the optimizer's estimates of the discrete displacements on the structures grid, designated X_{U_2}. In the IDF formulation, the additional optimization variables are the optimizer's estimates of the interdisciplinary spline coefficients $X_{\mu_{12}}$ and $X_{\mu_{21}}$. These optimization estimates may be thought of as "surrogates" for the variables appearing as the subscripts on X.

In the following, we explain each of these approaches in a little more detail; a more precise (and more abstract) mathematical presentation, using similar but not identical notation, may be found in [12].

3.1. *All-at-Once (AAO) Approach*

In the all-at-once (AAO) formulation the optimizer "controls" both estimates of the analysis variables X_{U_1}, X_{U_2}, and the design variables X_D; the equations $W_1 = 0$ and $W_2 = 0$ from the analysis disciplines appear as explicit constraints in the optimization. Figure 3 shows the flow of information for the AAO formulation. Note that in this and subsequent figures, the codes that implement the interdisciplinary mappings are not explicitly shown as boxes in the information flow. Also not shown is the flow of design constraint information that may be present in more complicated problems.

In AAO, we do not require analysis problem feasibility in any sense (individual discipline or multidisciplinary) until optimization convergence is reached. In a way, the optimizer does not "waste" effort trying to achieve feasibility when far from

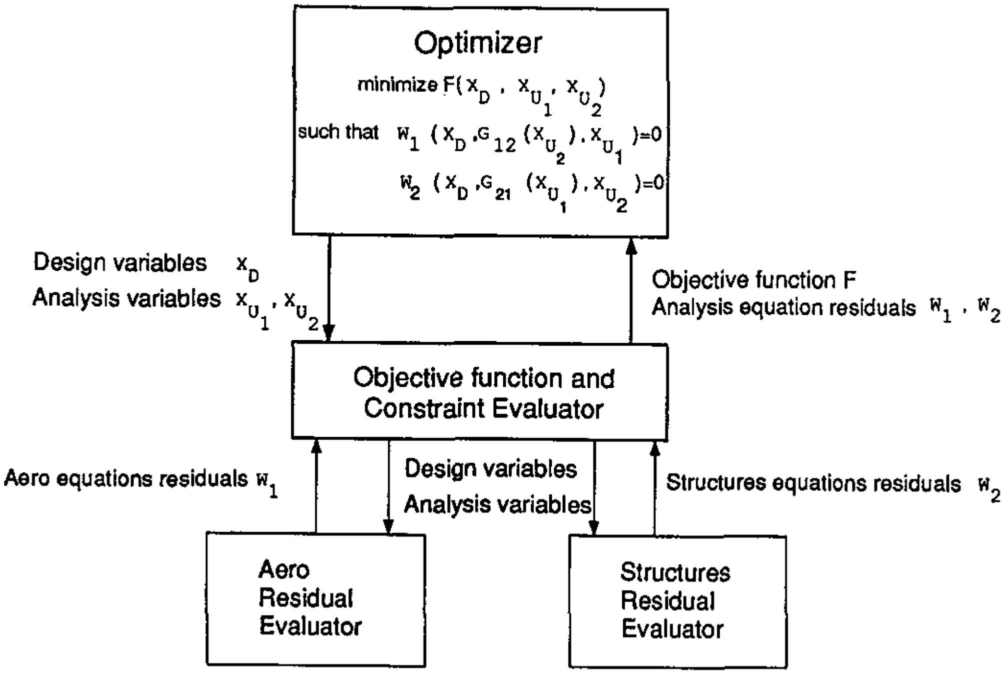

FIG. 3. Flow of information in the all-at-once (AAO) approach.

an optimum; the expectation is that the overall optimization process will thereby be more efficient. In AAO the "analysis code" performs a particularly simple function; it evaluates the *residuals* of the analysis equations, rather than *solving* some set of equations. Ultimately, of course, the optimizer for AAO must solve the analysis discipline equations to attain feasibility. Generally, this means that the optimization method must contain all of the special techniques (especially for the difficult discipline aerodynamics) that a single discipline analysis solver contains. For difficult MDO problems, it is unlikely that "equality constraint satisfaction schemes" (e.g., Newton's method) present in existing, general purpose optimization codes would be equal to this task. For the simple duct flow model studied here such methods are successful; see the results in Section 4.

3.2. *Multidisciplinary Feasible* (*MDF*) *Approach*

As mentioned above, AAO has the disadvantage that the optimization code must assume the difficult task of simultaneously satisfying all the analysis discipline equations. The MDF formulation has the advantage that it uses the specialized software that has been developed for solving the individual discipline equations. Figure 4 shows the flow of information for the multidisciplinary feasible formulation. It is the most common approach to MDO.

MDF is at the opposite end of the spectrum of problem formulations from AAO. In the MDF formulation the optimizer controls only the design variables X_D, and full multidisciplinary analysis problem feasibility is maintained at every optimization iteration. In some sense, MDF is a "black-box" approach, but the black-box solves all of the analysis disciplines and assures that the interdisciplinary coupling is correct.

Note that in Fig. 4 we have assumed that the objective function is expressed in terms of the grid variables U_1 and U_2. In some cases, it might be appropriate to formulate F as a function of the interdisciplinary spline coefficients μ_{12} and μ_{21} to simplify gradient computations. This is especially true if the method used to solve the MDA problem is Newton-spline.

3.3. *Individual Discipline Feasible* (*IDF*) *Approach*

The MDF method has the disadvantage that a full multidisciplinary analysis is required each time the optimization code requires an objective or constraint function evaluation. By adopting the individual discipline feasible (IDF) formulation, *several* methods can be constructed that eliminate the need for multidisciplinary feasibility while taking full advantage of existing analysis codes for individual disciplines. IDF occupies an "in-between" position on a spectrum where the AAO and MDF formulations represent extremes: for AAO, no feasibility is enforced at each optimization iteration, whereas for MDF, complete multidisciplinary feasibility is required. The IDF approach maintains individual discipline feasibility, while allowing the optimizer to drive the individual disciplines toward multidisciplinary feasibility and optimality by controlling the interdisciplinary mappings. Since the optimizer is estimating the interdisciplinary coupling parameters, the analysis disciplines can be solved independently.

While many different IDF formulations are possible [1, 12], Fig. 5 shows the flow of information for a "compressed interdis-

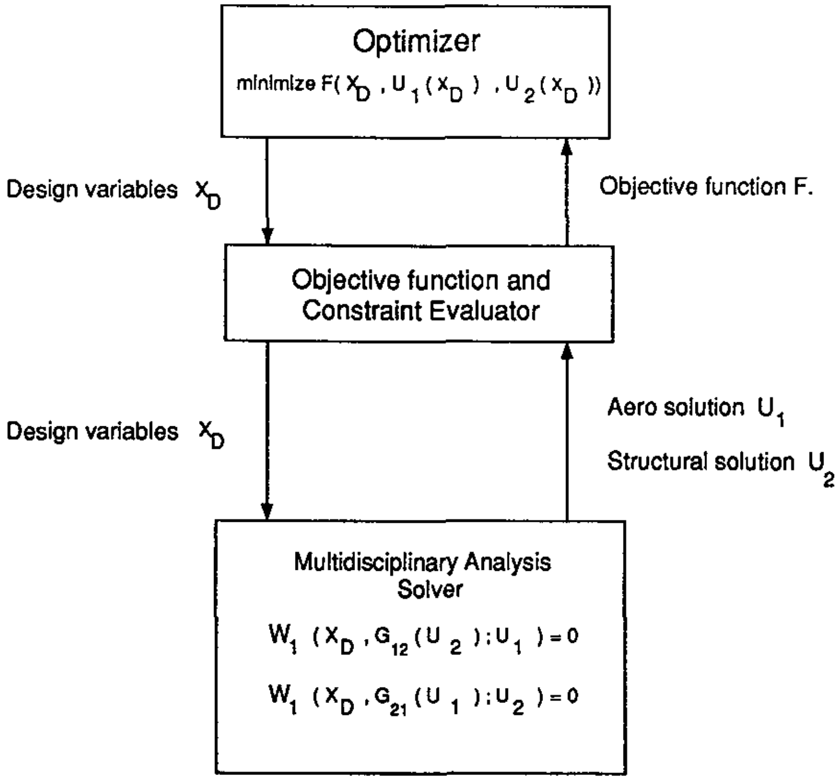

FIG. 4. Flow of information in the multidisciplinary feasible (MDF) approach.

ciplinary bandwidth'' IDF method in which the optimizer controls spline approximations to the interdisciplinary mappings. Note that, in the IDF approach, analysis variables have been ''promoted'' to become optimization variables; they are indistinguishable from design variables from the point of view of an individual analysis discipline solver. In IDF, the specific analysis variables that have been promoted are those that represent communication, or coupling, between analysis disciplines via interdisciplinary mappings.

Note that, as before for MDF, in Fig. 5 the objective function is assumed to be expressed in terms of the grid variables U_1 and U_2. Again, in some cases, it might be appropriate to formulate F as a function of the interdisciplinary spline coefficients μ_{12} and μ_{21}.

We note that the basic idea of promoting some coupling variables to become optimization variables has been mentioned in [13, 14].

3.4. *Role of the Optimizer*

In the preceding, we have tacitly assumed that the optimization method used does *not* enforce feasibility as part of its algorithm; otherwise, there would be little distinction among the MDF, IDF, and AAO approaches. This issue is discussed in more detail in [12].

As an aside, we note that an investigation of the relative merits of several optimization techniques for MDO may be found in [15].

3.5. *Gradient Computations*

Most calculus-based optimization approaches require the gradients of the objective function and constraints with respect to the optimization variables. The computation of gradients is an important issue in MDO, and a factor in choosing among approaches is the number and difficulty of such sensitivities (components of the gradients) that need to be calculated; see [1]. The flow of gradient information is not shown in Figs. 3–5, but is assumed to follow the function and constraint information as required for optimization.

We now make some general comments on gradient computations. Finite differences are popular for small problems, but they will be prohibitively expensive for large ones. Preliminary results using automatic differentiation (AD) suggest that, lacking dramatic improvements in AD technology, AD will be competitive with finite differences for cost. (AD enjoys other advantages, like better accuracy and ease of use.) We assume here that only ''analytic approaches'' such as implicit differentiation, sensitivity equations, or adjoint solutions will be sufficiently cheap for use in large problems. We also assume that these analytic methods will be *roughly* equivalent to each other, so we consider only implicit differentiation. However, the effi-

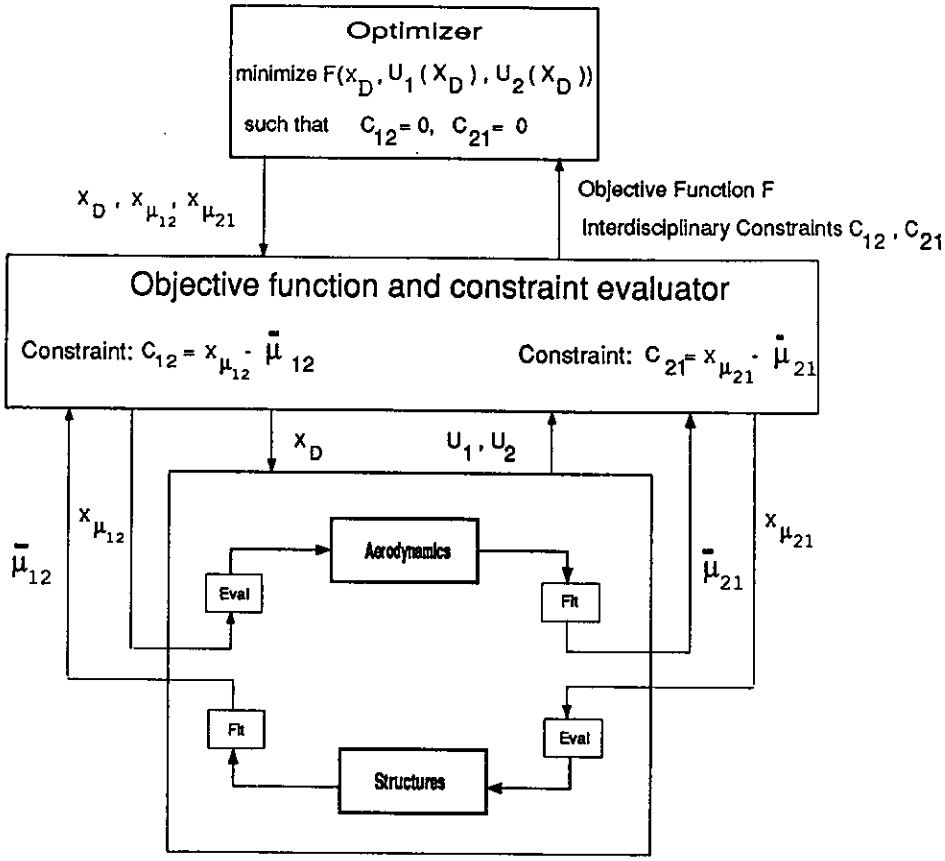

FIG. 5. Flow of information in the individual discipline feasible (IDF) approach. The bottom box is a simplified verison of Fig. 2, where the multidisciplinary analysis loop is broken at the spline coefficients μ_{12} and μ_{21}.

cient calculation of sensitivities is currently an area of intense research interest, and we can hope for breakthroughs that will significantly reduce the cost.

Suppose that we need the gradient of the MDF objective function $F(X_D, U_1(X_D), U_2(X_D))$ with respect to the design variables X_D. That gradient is

$$(\nabla_{X_D} F)^T = \left(\frac{\partial F}{\partial X_D}\right) + \left(\frac{\partial F}{\partial U_1}\right)\left(\frac{\partial U_1}{\partial X_D}\right) + \left(\frac{\partial F}{\partial U_2}\right)\left(\frac{\partial U_2}{\partial X_D}\right). \quad (11)$$

The difficult terms to obtain are the solution sensitivities $(\partial U_1/\partial X_D)$ and $(\partial U_2/\partial X_D)$. These are gotten by differentiating (5), (6), *at a solution of* (5), (6), with respect to X_D, to obtain

$$\begin{bmatrix} \frac{\partial W_1}{\partial U_1} & \frac{\partial W_1}{\partial U_2} \\ \frac{\partial W_2}{\partial U_1} & \frac{\partial W_2}{\partial U_2} \end{bmatrix} \begin{bmatrix} \frac{\partial U_1}{\partial X_D} \\ \frac{\partial U_2}{\partial X_D} \end{bmatrix} = \begin{bmatrix} -\frac{\partial W_1}{\partial X_D} \\ -\frac{\partial W_2}{\partial X_D} \end{bmatrix}. \quad (12)$$

Of course, this linear system has the same coefficient matrix as (7). In small problems one could take advantage of this by reusing a factorization to solve the same system for multiple right-hand sides. In large scale practical applications it would be difficult to exploit this possibility, since the linear algebra in such applications would usually be iterative. We note that (12) is equivalent to Sobieski's GSE1 technique [16]. We call the gradient obtained by using the solution of (12) in (11) an "implicit gradient."

The gradients needed for the IDF objective function are similar to those required in single discipline optimization. This is because these gradients need only be evaluated at single discipline feasibility. They can be obtained by the procedure above, except with the off-diagonal blocks $\partial W_1/\partial U_2$ and $\partial W_2/\partial U_1$ of (12) (the ones representing interdisciplinary coupling) zeroed out. Of course, this makes the computations of the aerodynamics gradients and the structures gradients independent.

We next consider the linearization of the IDF constraints (9). The linear term is

$$\begin{bmatrix} I & -\frac{\partial \bar{\mu}_{12}}{\partial X_{\mu_{21}}} & -\frac{\partial \bar{\mu}_{12}}{\partial X_D} \\ -\frac{\partial \bar{\mu}_{21}}{\partial X_{\mu_{12}}} & I & -\frac{\partial \bar{\mu}_{21}}{\partial X_D} \end{bmatrix} \begin{bmatrix} \Delta X_{\mu_{12}} \\ \Delta X_{\mu_{21}} \\ \Delta X_D \end{bmatrix}. \quad (13)$$

Recall that, e.g., $X_{\mu_{21}}$ is just an optimizer-controlled value of μ_{21}. Consider the computation of a typical block, say $\partial\overline{\mu}_{12}/\partial X_{\mu_{21}}$. From the definition $\overline{\mu}_{12} \equiv F_{12}(U_2(X_D, E_{21}(X_{\mu_{21}})))$, the required derivative is given by

$$\frac{\partial\overline{\mu}_{12}}{\partial X_{\mu_{21}}} = \frac{\partial F_{12}}{\partial U_2}\frac{\partial U_2}{\partial X_{\mu_{21}}}. \tag{14}$$

The first term is the derivative of the fitting process, which is presumed easy to obtain, and the second term is given as described above by the single discipline sensitivity

$$\frac{\partial U_2}{\partial X_{\mu_{21}}} = -\left(\frac{\partial W_2}{\partial U_2}\right)^{-1}\frac{\partial W_2}{\partial X_{\mu_{21}}}. \tag{15}$$

(This formula is obtained by zeroing out the off-diagonal blocks in (12) and replacing X_D with $X_{\mu_{21}}$.) Note that we did not expand $\partial U_2/\partial X_{\mu_{21}} = (\partial U_2/\partial Y_{21})(\partial Y_{21}/\partial X_{\mu_{21}})$, where $Y_{21} = E_{21}(X_{\mu_{21}})$; this would have required more solutions with $\partial W_2/\partial U_2$, since we assume there are fewer coefficients μ_{21} than data Y_{21}.

Note that the computation of (13) requires quite a few single discipline sensitivities. Let n_D be the number of design variables and n_{ij} be the number of interdisciplinary coupling variables from discipline j to discipline i. The computation of a single Jacobian requires $(n_{12} + n_D)$ solutions with $\partial W_1/\partial U_1$ and $(n_{21} + n_D)$ solutions with $\partial W_2/\partial U_2$.

Also note that it is sometimes desirable to take transposes of the above formulas to reduce work; see [1]. However, only analysis codes employing direct linear algebra could easily take advantage of transpose solves. Other codes would need to be retrofitted (say, by automatic differentiation) to compute transpose (adjoint) solutions. In the remainder of the paper we assume transpose solves are not available (even though we use direct linear algebra for our simple model).

Finally, the gradients required by AAO are derivatives of residuals rather than of solutions and are presumed to be easily obtainable.

3.6. *Computational Work Required by the AAO, MDF, and IDF Approaches*

One of the considerations in choosing a formulation for a particular MDO application is the amount of computational work required. However, because the types of work performed by the AAO, MDF, and IDF approaches are different and could be performed in different ways (say by direct or iterative linear algebra), it is difficult to make comparative work estimates. In the following material we make some qualitative observations about computational work.

Obviously, in MDF or IDF the total work will be the work per evaluation of the objective function, constraints, and associated gradients times the number of such evaluations required to obtain convergence, plus the work done in the optimization. We can assume in both cases that the optimization work will be negligible. In IDF, each function evaluation requires the same amount of work from each of aerodynamics and structures that would be required in single discipline optimization, except that there are more optimization variables so more gradients are required. These extra gradients are gradients of the interdisciplinary constraints C_{12} and C_{21}, and the extra optimization variables are the interdisciplinary spline coefficients μ_{12} and μ_{21}. The gradient computation expense grows linearly with the number of such interdisciplinary variables. In the MDF approach, we additionally have to solve the MDA problem at each iteration. The work to solve the MDA problem depends on the method employed, three of which were mentioned previously in Section 2.2. In general, we may expect the MDA solution to be several times as expensive as the sum of a single aerodynamics and a single structures solution. Additionally, while there are fewer gradients required, the gradient computations are more complex, due to the off-diagonal terms in (12).

For AAO, it is even more difficult to discuss computational work. In some sense, in AAO all of the work is "optimization work." Because the optimization problem will be extremely large, and generally sparse, new algorithms and software to solve the problems using iterative linear algebra will probably be developed in a case-specific manner.

4. COMPUTATIONAL RESULTS

4.1. *Discrete Analysis Problem*

In this section we present computational results for the discrete analysis problem described in Section 2.2. Such a multidisciplinary analysis is only required for the MDF optimization approach.

We took $M = 40$ and $N = 25$ for the number of interior grid points on the flow and structures grids, respectively. As in [6] we chose the boundary conditions on V to be $v(0) = 1.299$, $v(1) = 0.506$ which causes the flow problem to contain a shock. The stiffness s was chosen to be a constant sufficiently small (0.2) so that the duct wall was quite flexible. The interdisciplinary mappings used a least squares spline approximation to obtain splines describing the pressure P from the flow problem and the displacement D from the structures problem. The knots were uniformly distributed in $0 \leq \xi \leq 1$. Constraints were applied to assure that the splines exactly matched the boundary values on P and D. Piecewise cubic splines with $n_{12} = 3$ free (non-boundary) spline coefficients were used for D (which is very smooth) and piecewise linear splines with $n_{21} = 5$ free spline coefficients were used for P (which contains a discontinuity—the shock). Experimentation showed that the MDA solutions are quite insensitive to the number of spline coefficients used. This is because the displacement D is so smooth that it is easy to approximate, and the pressure P, while rough, affects the solution of the beam equation as though it had been integrated several times. This can be seen by taking

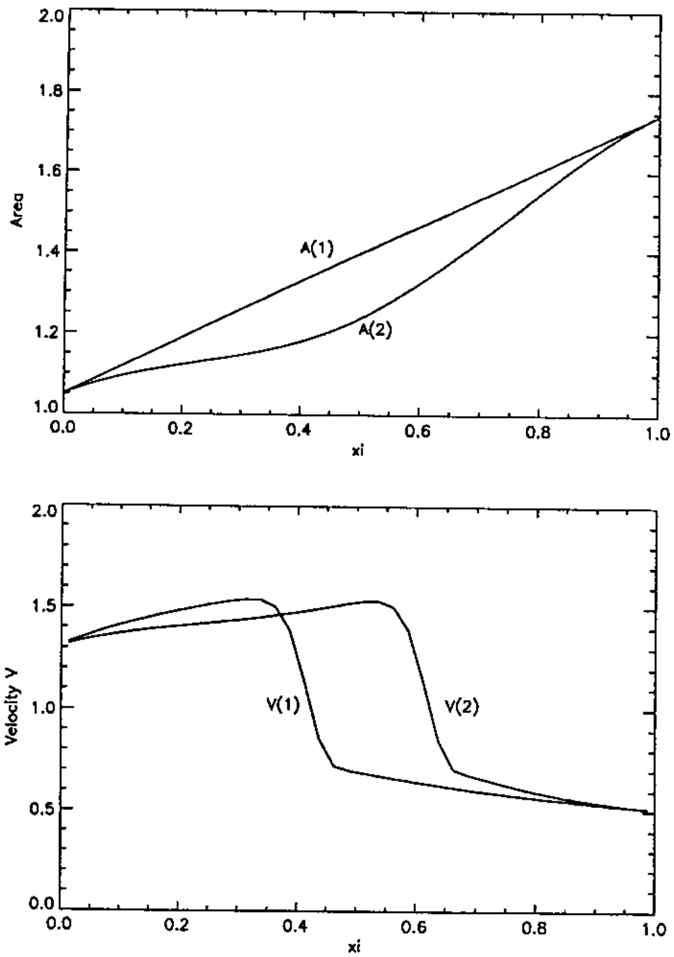

FIG. 6. Solution of discrete analysis problem with no flexibility ($A^{(1)}$, $V^{(1)}$) and with flexibility ($A^{(2)}$, $V^{(2)}$).

s = const in (2), and integrating four times. The values of the remaining constants were $\overline{H} = 1.167$ and $\overline{\gamma} = 0.167$.

Computational results are shown in Fig. 6 for the case of no flexibility ($A^{(1)}$, $V^{(1)}$) and with flexibility ($A^{(2)}$, $V^{(2)}$). As shown, as the wall flexes inward the shock shifts quite a bit to the right (downstream). Essentially the same solution is obtained no matter which of the three MDA solution methods is used. The convergence history is displayed in Fig. 7 for the flexibility cases. In both Newton methods, a crude damping strategy that limited the Newton correction at any iteration to a maximum of 1.0 was exercised on a few of the early iterations. As seen in the figure, quadratic convergence was observed at the end of the Newton iterations. The fixed point iteration convergence, although linear, was remarkably rapid. Information on computational work is displayed in Table I. The fixed point and Newton-grid results are essentially independent of M, N, and the number of interdisciplinary spline coefficients. Of course, the Newton-spline results depend strongly on the number of spline coefficients due to the computation of (10). The entry $47 = 32 + 5 * 3$ indicates, for example, that 32 linear solves of size M were needed to solve the aerodynamics analysis problem five times (to compute C_{21}) and 15 additional solves were required to compute the three columns of the Jacobian block $\partial\overline{\mu}_{21}/\partial\mu_{12}$ five times.

4.2. *Discrete Design Problem*

Here we present results for the discrete design problem described in Section 2.4 obtained by employing the MDF, IDF, and AAO formulation approaches.

In all cases, the optimization code was NPSOL (even though this would only be a good choice for MDF, and possibly IDF, applied to larger problems, since NPSOL does not exploit sparsity). The goal velocity $\hat{V}$ was the solution for the linear duct (with no flexibility) $V^{(1)}$ in Fig. 6. The design variables were coefficients of a cubic spline describing changes to the linear duct $A^{(1)}$. Since the duct wall will flex in, we expect the design for A to bow out (increasing area) to compensate, so that the flexed duct wall $A + D$ becomes linear and the goal velocity is exactly achieved (objective function zero). The remaining discretization parameters were as described for multidisciplinary analysis above.

Before making performance comparisons, we give some results obtained with the "standard" MDF approach. We used the damped Newton-grid method described above to solve the MDA problem and warm-started the Newton iterations with the solution from the previous analysis. Occasionally, this caused Newton's method to fail, in which case we restarted from the default initial guess, which is a linear profile for V and $D = 0$. The gradients of the objective function were obtained by the implicit gradient method (11), (12). The NPSOL optimality and feasibility tolerances were set to 1.e-5. In all cases, NPSOL reported that optimal solutions were found.

Figure 8 displays the results obtained with $n_D = 2$ design variables. The correct solution, a linear duct for $A + D$, is not exactly achieved. NPSOL converged to a final objective function value of 1.2 e-4 in three major iterations and six function evaluations. Figure 9 shows the results obtained with $n_D = 5$ design variables. The resulting duct shape is much closer to linear. In this case, NPSOL converged to a final objective function value of 9.7 e-6 in 13 major iterations and 16 function values.

We now present some comparisons of methods. As noted in Section 3.4, it is difficult to directly compare the amounts of computational work required by the three approaches. However, by making some judicious choices, we can come close to "apples-to-apples" comparisons. In the remaining results we employ five design variables.

Our first comparison involves MDF versus AAO. In this MDF method we solve the multidisciplinary analysis problem with the Newton-grid MDA method. Thus the dominating unit of work is solving linear systems (7) of approximate size $M + N$. Some of these solutions are required to compute implicit gradients of the objective function as described above and thus

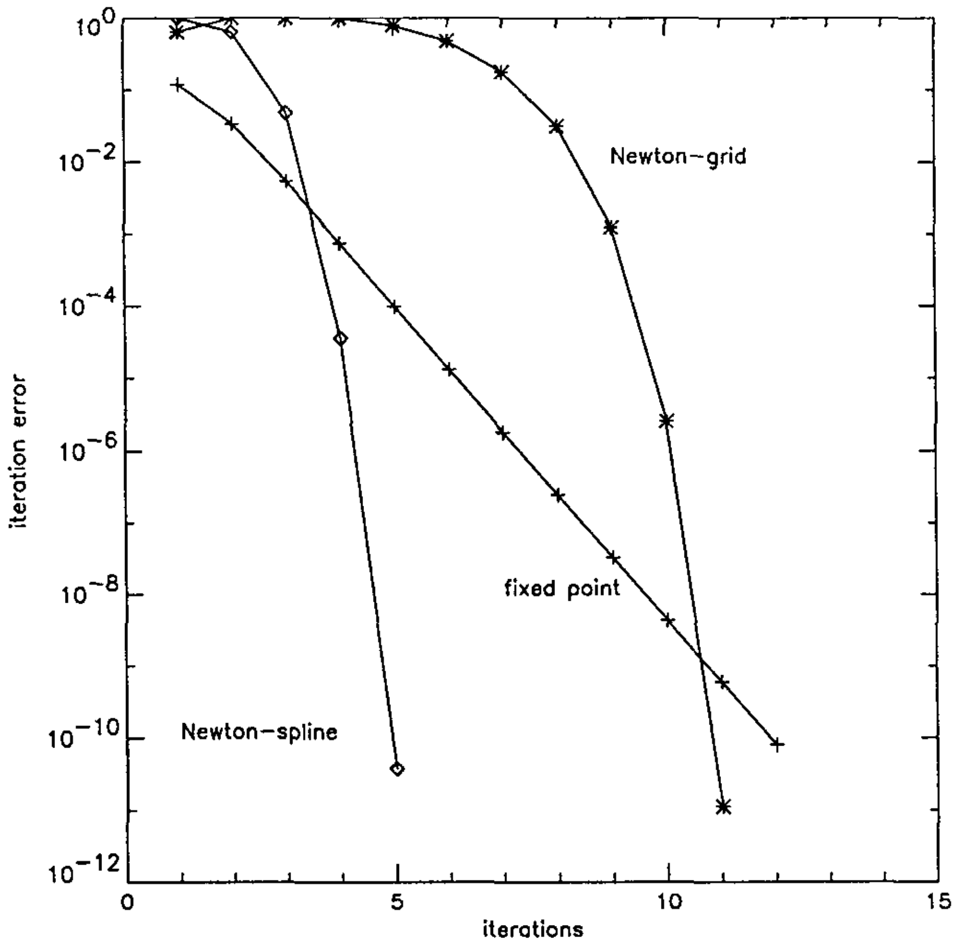

FIG. 7. Convergence history for the discrete multidisciplinary analysis problem.

involve the same coefficient matrix with different right-hand sides. Obviously, if direct linear algebra is used, the work for an additional right-hand side is much lower than for a whole new factorization. However, in larger problems only iterative linear algebra is likely to be employed. Thus we do not report on the savings possible, due to direct solutions with multiple RHSs. In AAO, the optimizer NPSOL has to solve linear systems whose size equals the size of the active set, which is again approximately $M + N$. Thus we can compare this version of

TABLE I

Computational Work for the Discrete Multidisciplinary Analysis Problem

MDA solution method	Linear system solutions of size M	Linear system solutions of size N	Linear system solutions of size $M + N$	Iterations
Fixed point	53	12	—	12
Newton-grid	—	—	11	11
Newton-spline	$47 = 32 + 5 * 3$	$30 = 5 + 5 * 5$	—	5

MDF with AAO by counting the number of such linear solves. The results are presented in Table II. The entry for MDF linear system solutions indicates that 75 of the solutions were due to computing 15 objective function gradients for the five design variables. Note that this could have been accomplished with 15 transpose solves.

As expected, AAO is more efficient than MDF by virtue of postponing feasibility. A similar conclusion was reached in [6] for single discipline optimization. Additionally, experiments showed that for AAO the number of optimization iterations and linear system solutions does not grow with the number of design variables.

Our second comparison involves MDF versus IDF. In this MDF method we solved the multidisciplinary analysis problem with the Newton-spline MDA method. There is a subtlety involved in this MDF–IDF comparison. In solving the MDA problem (for MDF) with the Newton-spline method, we need to solve the linear system (10) which only involves single discipline sensitivities like (14). However, in general the gradient of the MDF objective function, given by (11), requires solutions with the coefficient matrix in (12), the off-diagonal

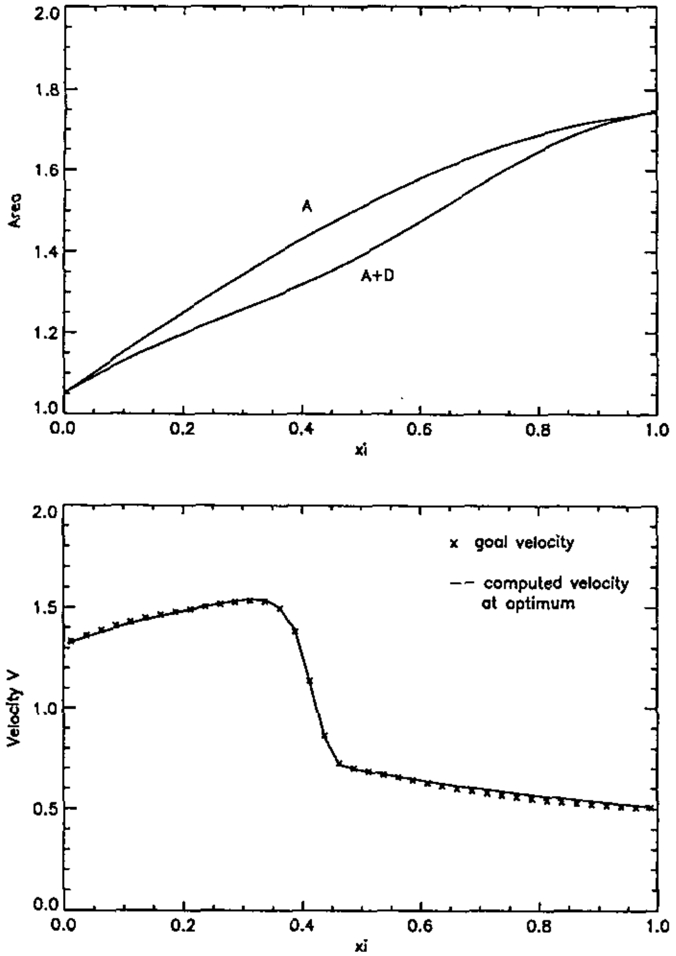

FIG. 8. Solution of discrete design problem using two design variables.

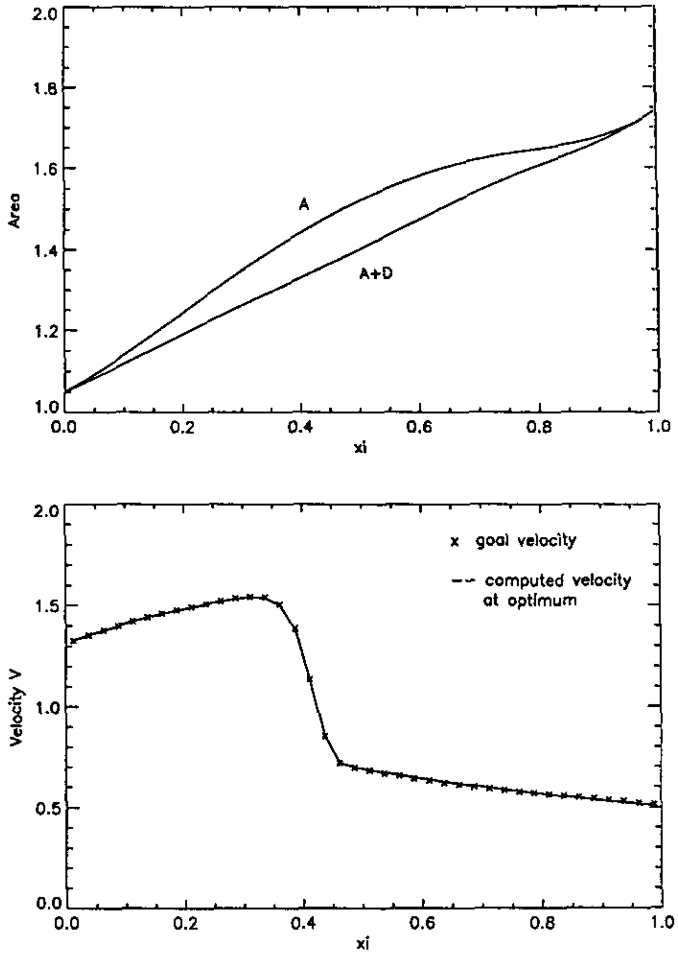

FIG. 9. Solution of discrete design problem using five design variables.

terms of which might not be available in a Newton-spline solution process. To make for a cleaner comparison here, we avoid this issue by simply ignoring the work required to compute the objective function and its gradients. We focus instead on the comparison of the work required in MDF (Newton-spline) to compute the Newton iteration Jacobian (10) versus the work in IDF to compute the constraint Jacobian (13). We further simplify the comparison by counting only the number of Jacobian evaluations, ignoring the different numbers of columns in (10) and (13). Very roughly speaking, this isolates the effects of postponing feasibility in IDF versus maintaining feasibility at each iteration in MDF, from all of the other complexities in making the comparison. The results are displayed in Table III. Once again, there is an advantage to postponing feasibility.

It must be noted that each of the Jacobian calculations reported in Table III requires a significant amount of computation, as detailed in Section 3.5.

5. CONCLUSIONS

We applied the multidisciplinary feasible (MDF), individual discipline feasible (IDF), and all-at-once (AAO) approaches

TABLE II

Comparison of MDF vs AAO

Formulation approach	Linear system solutions of size $M + N$	Optimization iterations
MDF	$176 = 101 + 15 * 5$	13
AAO	26	26

TABLE III

Comparison of MDF vs IDF

Formulation approach	Number of Jacobians calculated	Optimization iterations
MDF (Newton-spline)	33	10
IDF	14	14

for multidisciplinary design optimization to a simple model for duct flow. This model problem exhibits some of the properties of static aeroelasticity but is radically simpler than realistic MDO problems. For this model, the IDF and AAO formulations reduce computational work by about a factor of 2–5 by postponing feasibility of the analysis problem until optimality is achieved. However, that advantage occurs in keeping all other things (roughly) equal, which would probably not be the case in real MDO applications. Additionally, a price is paid in that useful results may not be available in IDF and AAO if the optimization iteration is stopped short of optimality.

Even in the context of this simple model, it is hard to make comparisons between the different formulation approaches because different types of work are required in each. In realistic MDO problems, custom-tailored numerical algorithms will undoubtedly be used to minimize computational expense as much as possible. That will make comparisons even more difficult. However, we expect that many of the key problem features investigated here will influence the choice of a formulation approach in more difficult problems. Among these features are the difficulty and cost of calculating gradients, the "bandwidth" of coupling between disciplines, and the degree to which it is desirable to maintain independence of the disciplines and their software.

In the near term, we expect most MDO problems to be solved by assembling existing software components. Computational cost will be controlled by keeping problem sizes small to moderate. Thus we expect that formulation approaches like MDF or IDF will predominate. Eventually, more analysis disciplines will be employed (including those describing manufacturing processes) and more design variables will be used. We expect the enormous computational expense of such large-scale MDO problems to drive researchers toward more tightly integrated coupling of the analysis disciplines and the optimization as exemplified by the AAO approach.

ACKNOWLEDGMENTS

The author is indebted to Evin J. Cramer and Paul D. Frank for discussions that greatly aided the development of this paper. He is additionally grateful to John Dennis and Michael Lewis, who are coauthors of [1, 12] from which much of the material on alternative formulations is taken or paraphrased.

REFERENCES

1. E. J. Cramer, J. E. Dennis, Jr., P. D. Frank, R. M. Lewis, and G. R. Shubin, "On Alternative Problem Formulations for Multidisciplinary Design Optimization," in *Proceedings, Fourth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, September, 1992*, AIAA 92-4752 (unpublished).
2. *Third Air Force/NASA Symposium on Recent Advances in Multidisciplinary Analysis and Optimization, 1990, San Francisco, CA* (unpublished).
3. *Proceedings, Fourth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, 1992* (American Institute of Aeronautics and Astronautics, Cleveland, OH, 1992).
4. *Proceedings, Fifth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 1994* (American Institute of Aeronautics and Astronautics, Panama City Beach, FL, 1994).
5. G. R. Shubin, Technical Report AMS-TR-189, Boeing Computer Services, Seattle, July 1992 (unpublished).
6. P. D. Frank and G. R. Shubin, *J. Comput. Phys.* **98,** 74 (1992).
7. G. R. Shubin and P. D. Frank, Technical Report AMS-TR-163, Boeing Computer Services, Seattle, April 1991 (unpublished).
8. G. R. Shubin and P. D. Frank, "A Comparison of Two Closely-Related Approaches to Aerodynamic Design Optimization," in *Proceedings, Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-III), October 1991*, edited by G. S. Dulikravich.
9. A. R. Shenoy and E. M. Cliff, Technical Report 93-07-01, Interdisciplinary Center for Applied Mathematics, Virginia Polytechnic Institute and State University, July 1993.
10. R. T. Haftka, Z. Gurdal, and M. P. Kamat, *Elements of Structural Optimization* (Kluwer Academic, Dordrecht/Norwell, MA, 1990).
11. P. D. Frank and G. R. Shubin, "A Comparison of Optimization-based Approaches for Solving the Aerodynamic Design Problem," in *Third Air Force/NASA Symposium on Recent Advances in Multidisciplinary Analysis and Optimization, September 24–26, 1990, San Francisco, CA* (unpublished).
12. E. J. Cramer, J. E. Dennis, Jr., P. D. Frank, R. M. Lewis, and G. R. Shubin, *SIAM J. Optim.* **4,** 754 (1994).
13. R. T. Haftka, J. Sobieszczanski-Sobieski, and S. L. Padula, *Struct. Optim.* **4,** 65 (1992).
14. P. Gage and I. Kroo, "Development of the Quasi-Procedural Method for Use in Aircraft Configuration Optimization," in *Proceedings, the Fourth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, September 1992,* AIAA 92-4693 (unpublished).
15. P. D. Frank, A. J. Booker, T. P. Caudel, and M. J. Healy, Technical Report AMS-TR-172, Boeing Computer Services, Seattle, December 1991 (unpublished).
16. J. Sobieszczanski-Sobieski, *AIAA J.* **28,** 153 (1990).