Proceedings of DETC'05
2005 ASME Design Engineering Technical Conferences
Long Beach, California, USA, September 24-28, 2005

DETC2005-84790

ON THE IMPACT OF COUPLING STRENGTH ON COMPLEX SYSTEM OPTIMIZATION FOR SINGLE-LEVEL FORMULATIONS

James Allison, Michael Kokkolaras, and Panos Papalambros
Optimal Design Laboratory
Department of Mechanical Engineering
University of Michigan, Ann Arbor, Michigan 48109
Email: {optimize,mk,pyp}@umich.edu

ABSTRACT

Design of modern engineering products requires complexity management. Several methodologies for complex system optimization have been developed in response. Single-level strategies centralize decision-making authority, while multi-level strategies distribute the decision-making process. This article studies the impact of coupling strength on single-level Multidisciplinary Design Optimization formulations, particularly the Multidisciplinary Feasible (MDF) and Individual Disciplinary Feasible (IDF) formulations. The Fixed Point Iteration solution strategy is used to motivate the analysis. A new example problem with variable coupling strength is introduced, involving the design of a turbine blade and a fully analytic mathematical model. The example facilitates a clear illustration of MDF and IDF and provides an insightful comparison between these two formulations. Specifically, it is shown that MDF is sensitive to variations in coupling strength, while IDF is not.

KEYWORDS

Complex System Design, Multidisciplinary Design Optimization, Coupling Strength, Multidisciplinary Feasible, Individual Disciplinary Feasible, Fixed Point Iteration

1 INTRODUCTION

This article endeavors to illustrate the implementation of single-level formulations for complex system optimization via an analytical example. Specifically, the effect that subsystem interdependence (coupling) has on the performance of these formulations is explored. Background in complex system optimization is provided, and the mathematical model for the illustrative example is developed and presented. This work aims at improving general understanding of techniques for complex system optimization, and it is part of a greater effort to review single and multi-level MDO formulations [1, 2].

Design of products classified as complex systems poses substantive challenges to both analysis and optimization, necessitating specialized solution techniques. A complex system is defined as an assembly of interacting members that is difficult to understand as a whole. An interaction between members exists if some aspect of one member affects how the system responds to changes in another member. A system is difficult to understand if an individual cannot understand the details of all members and all interactions between members. A system may qualify as complex due to its large scale (large number of members or inputs), or due to strong interactions. These interactions complicate optimization, but provide opportunity to exploit synergy between system members.

Analysis of complex systems as an undivided whole can be inefficient, if not intractable. An alternative is to partition the system into smaller subsystems (or subspaces). Wagner [3] identified four categories of system partitioning methods: by object, by aspect, sequential, or matrix. The aspect (discipline) partitioning paradigm is used in this article, and the term ‘subspace’ is used to refer to system members. System partitioning is also characterized by the structure of its communication pathways. A non-hierarchic system has no restrictions on these pathways

(Figure 1), while a hierarchic system allows communication only between parent and child subspaces, as depicted in Figure 2. The highest level subspace, the master problem, coordinates all subordinate subspaces. Some formulations for complex system optimization transform non-hierarchic systems into hierarchic systems using auxiliary constraints.

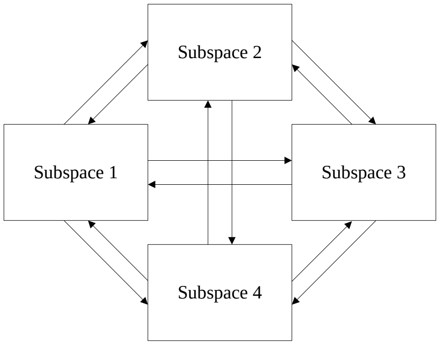

Figure 1. ILLUSTRATION OF A NON-HIERARCHIC SYSTEM

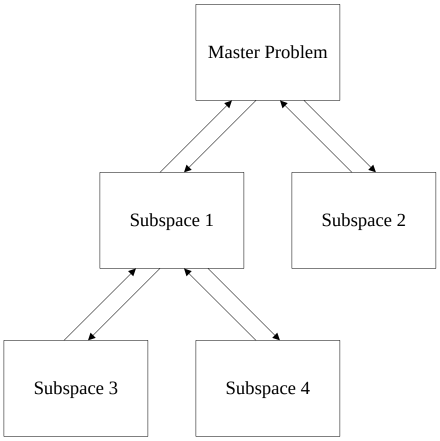

Figure 2. ILLUSTRATION OF A HIERARCHIC SYSTEM

Subspace interaction may be simple and unidirectional, or it may be multidirectional. Classical examples of the latter are aeroelasticity and combustion. In aircraft wing analysis, structural analysis requires the aerodynamic pressure distribution (calculated by the aerodynamic analysis) to find the wing deflections, and the aerodynamic analysis requires wing deflections to predict the pressure distribution. Structural and aerodynamic analyses are coupled—they depend on each other. Combustion requires the analysis of fluid transport, heat transfer, and chemical reactions. Each of the three disciplines depends upon the other two, resulting in a total of six couplings. If a system has N subspaces, then the system has a possible $N(N-1)$ couplings (which may be scalar, vector, or function valued). The combustion example is fully coupled, since all of the possible $3(3-1)=6$ couplings exist. The number and strength of these couplings influences system analysis difficulty. A coupling is strong if the behavior of a subspace is highly sensitive to changes to some aspect of the subspace it is coupled to. Formal methods for quantification of these sensitivities have been developed, such as Global Sensitivity Equations [4] and Coupling Factors [5]. Sosa *et al.* present a scheme to characterize these sensitivities as strong or weak based on design expert knowledge [6, 7].

In systems optimization, each subspace must be a team player, sacrificing its own objectives if required, to enhance the overall system objective. The traditional sequential design process [8] begins by optimizing the first subspace, fixing that aspect of the system design, and then proceeding with subsequent subspace designs. The approach deals with the interactions, but cannot exploit them to synergistically improve system performance. A more holistic approach is required.

A general coupled system is used to introduce basic terminology for complex system analysis and optimization (Fig. 3). All $N(N-1)$ possible subspace interactions are shown (which may not all exist in a real system). A designer, or an optimization algorithm, provides the system analysis with the design variable vector $\mathbf{x}$. When the analysis is complete, the system response functions, such as f, $\mathbf{g}$, and $\mathbf{h}$, are returned. The system design variable vector is separated into two categories of variables: local design variables $\mathbf{x}_{\ell i}$ and shared design variables $\mathbf{x}_{si}$. Local variables $\mathbf{x}_{\ell i}$ are unique to subspace i—no other subspace takes them as inputs. Shared variables $\mathbf{x}_{si}$ are required inputs to subspace i, but are also used as inputs in at least one other subspace. The collection of all local variables is $\mathbf{x}_{\ell}$, and the collection of all shared variables is $\mathbf{x}_s$. The aggregation of local and shared variables for subspace i is $\mathbf{x}_i$.

When interaction is present, quantities referred to as coupling variables are passed between subspaces during analysis. The collection of values generated by subspace j and received by subspace i is $\mathbf{y}_{ij}$. The collection of all coupling variables is $\mathbf{y}$. Coupling variables are artifacts of decomposition; they do not exist in the system's original design problem statement—$\mathbf{x}$ and $\mathbf{y}$ have no common members. In addition to coupling variables, subspaces may compute values such as subspace objective functions and constraint functions: f_i, $\mathbf{g}_i$, and $\mathbf{h}_i$. The system objective function f may simply be one of the subspace responses, or in the general case is a function of several of the subspace responses. The set of all subspace constraints and any system level constraints is $\mathbf{g}$ and $\mathbf{h}$. As with the system objective function, system constraints may be functions of one or more subspace responses.

2 REVIEW OF SINGLE-LEVEL MDO FORMULATIONS

The formulations and strategies for the Multidisciplinary Feasible (MDF), Individual Disciplinary Feasible (IDF), and All-at-Once (AAO) methodologies are presented here. This terminology was introduced in [9]. These are single level formulations—that is, all decision making is centralized and per-

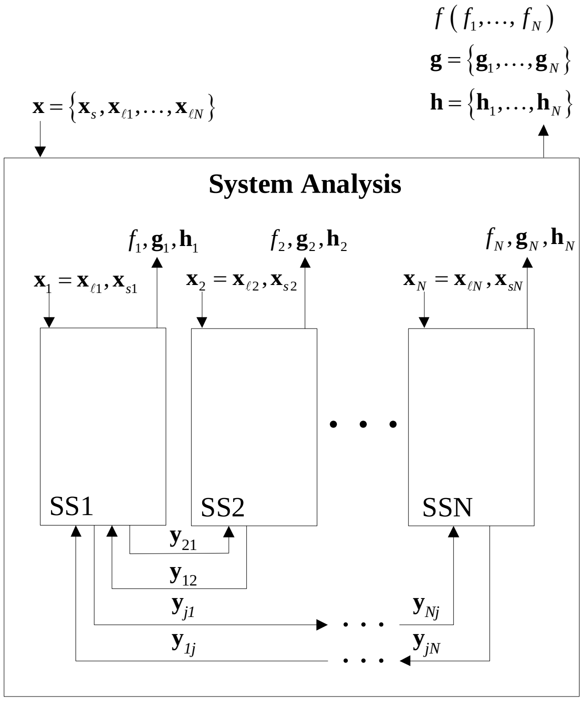
$f(f_1,\ldots,f_N)$
$\mathbf{g}=\{\mathbf{g}_1,\ldots,\mathbf{g}_N\}$
$\mathbf{h}=\{\mathbf{h}_1,\ldots,\mathbf{h}_N\}$
$\mathbf{x}=\{\mathbf{x}_s,\mathbf{x}_{\ell 1},\ldots,\mathbf{x}_{\ell N}\}$
System Analysis
$f_1,\mathbf{g}_1,\mathbf{h}_1$
$f_2,\mathbf{g}_2,\mathbf{h}_2$
$f_N,\mathbf{g}_N,\mathbf{h}_N$
$\mathbf{x}_1=\mathbf{x}_{\ell 1},\mathbf{x}_{s1}$
$\mathbf{x}_2=\mathbf{x}_{\ell 2},\mathbf{x}_{s2}$
$\mathbf{x}_N=\mathbf{x}_{\ell N},\mathbf{x}_{sN}$
SS1
SS2
SSN
$\mathbf{y}_{21}$
$\mathbf{y}_{12}$
$\mathbf{y}_{j1}$
$\mathbf{y}_{Nj}$
$\mathbf{y}_{1j}$
$\mathbf{y}_{jN}$

Figure 3. GENERAL NON-HIERARCHIC COUPLED SYSTEM

formed by a single optimizer. Balling and Sobieski provided an updated review of these formulations in [10], and Balling and Wilkinson implemented these formulations in the solution of an analytical test problem [5]. None of the preceding publications, however, address the variation of formulation performance with respect to changes in coupling strength.

2.1 Multidisciplinary Feasible Formulation

The most basic of MDO formulations is the MDF approach, also known as 'Nested Analysis And Design' (NAND), 'Single-NAND-NAND' (SNN), 'All-in-One' (AIO), and 'One-at-a-Time'. The MDF architecture is depicted in Fig. 4. This formulation is distinct from AAO, presented later in this section. A single system-level optimizer is used, and from the perspective of the optimizer MDF is no different than a 'standard' optimal design problem. A system analyzer coordinates all of the subspace analyzers. The optimizer supplies the system analyzer with a design $\mathbf{x}$, and the system analyzer supplies the optimizer with the appropriate response functions, f, $\mathbf{g}$, and $\mathbf{h}$.

As we will discuss in Section 3, the fixed point iteration algorithm is a popular solution method for MDF, although other analysis options exist. A formulation strategy is classified as MDF if a complete system analysis is performed for every optimization iteration. The analysis is "nested" within the design (hence the name NAND). The optimizer is charged with the responsibility to find the optimal design $\mathbf{x}^*$ (the design solution), while the system analyzer is solely responsible to find the set of consistent coupling variables $\mathbf{y}$.

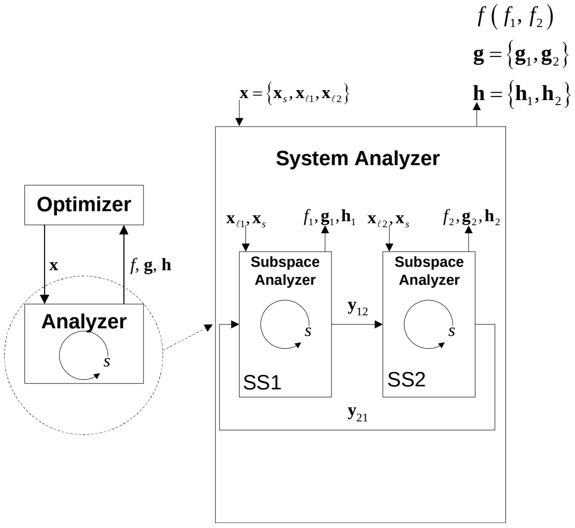
$f(f_1,f_2)$
$\mathbf{g}=\{\mathbf{g}_1,\mathbf{g}_2\}$
$\mathbf{h}=\{\mathbf{h}_1,\mathbf{h}_2\}$
$\mathbf{x}=\{\mathbf{x}_s,\mathbf{x}_{\ell 1},\mathbf{x}_{\ell 2}\}$
System Analyzer
Optimizer
x
f, g, h
Analyzer
s
$\mathbf{x}_{\ell 1},\mathbf{x}_s$
$f_1,\mathbf{g}_1,\mathbf{h}_1$
$\mathbf{x}_{\ell 2},\mathbf{x}_s$
$f_2,\mathbf{g}_2,\mathbf{h}_2$
Subspace Analyzer
Subspace Analyzer
$\mathbf{y}_{12}$
s
s
SS1
SS2
$\mathbf{y}_{21}$

Figure 4. MDF ARCHITECTURE

The MDF problem statement is shown in Eq. (1). MDF is completely non-hierarchic in nature (no communication restrictions). In a purely computational context, this approach is desirable if the subspaces are weakly coupled (fast analysis convergence), and if the subspace analyses are not computationally expensive. In an organizational context, MDF allows the continued use of legacy analysis tools without modification. If the organization already performs a complete analysis before making a design decision, MDF is a natural fit.

$$\begin{aligned} \min_{\mathbf{x}=[\mathbf{x}_\ell,\mathbf{x_s}]} \quad & f(\mathbf{x}) && (1)\\ \text{subject to} \quad & \mathbf{g}(\mathbf{x})=[\mathbf{g_1},\mathbf{g_2},\ldots,\mathbf{g_s}]\leq\mathbf{0} \\ & \mathbf{h}(\mathbf{x})=[\mathbf{h_1},\mathbf{h_2},\ldots,\mathbf{h_s}]=\mathbf{0} \end{aligned}$$

Although the merits of MDF are notable, its shortcomings must be clearly understood. MDF is dependent upon the efficiency and robustness of the analyzer: If the analyzer does not converge for even one design point (or finite difference point), the optimizer may fail. MDF cannot be parallelized, resulting in substantive design cycle time: One analysis may be sitting idle for significant periods of time waiting for required input. Finally, if a gradient-based optimization algorithm is employed, several more complete analyses must be performed for finite differencing. MDF is aptly termed a 'brute force' approach [11]. These limitations motivate the development of other formulations that offer better convergence properties, fit a wider variety of organizational structures, and allow for parallelization of analysis.

2.2 Individual Disciplinary Feasible Formulation

The IDF approach overcomes some of the limitations of MDF. IDF is also known as Simultaneous Analysis And Design (SAND) or Single-SAND-NAND. Like MDF, an analyzer for each subspace is employed, and a single system-level optimizer is used. The key difference is that the optimizer coordinates the interactions between the subspaces. This enables parallelization, improves convergence properties, and drives the design toward better solutions if multiple analysis solutions exist. IDF has notably improved robustness over MDF.

The IDF architecture is illustrated in Fig. 5. The optimizer chooses values for both design and coupling variables **y**. The distinction between analysis and design processes is blurred—they are performed simultaneously (hence the name SAND). The system optimizer provides all inputs required for all subspaces. Since the subspaces no longer must wait for the output of other analyses, they may be evaluated in parallel. As with MDF, design decision making is centralized in the IDF formulation, and analysis is distributed.

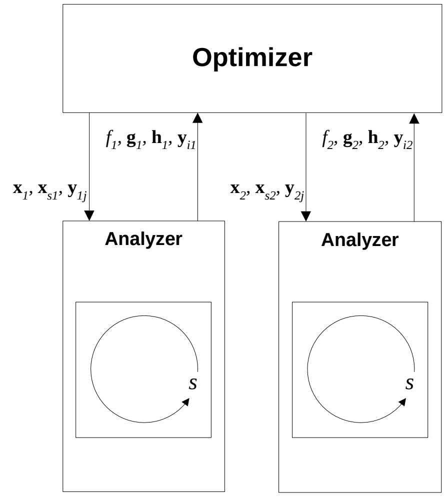
Optimizer
f_1, $\mathbf{g}_1$, $\mathbf{h}_1$, $\mathbf{y}_{i1}$
f_2, $\mathbf{g}_2$, $\mathbf{h}_2$, $\mathbf{y}_{i2}$
$\mathbf{x}_1$, $\mathbf{x}_{s1}$, $\mathbf{y}_{1j}$
$\mathbf{x}_2$, $\mathbf{x}_{s2}$, $\mathbf{y}_{2j}$
Analyzer
Analyzer
s
s

Figure 5. IDF ARCHITECTURE

Auxiliary equality constraints enforce system consistency at convergence, ensuring that the coupling variables computed by the subspaces are equal to the coupling variables supplied by the optimizer. If the process is interrupted, the intermediate design may not be consistent and/or feasible. In contrast, if an MDF process is interrupted prematurely, the design will be consistent but not be feasible.

The auxiliary constraints perform another important function—they break down any non-hierarchical links in the system so that the problem can be solved in a hierarchical manner (Fig. 5). Any arbitrary non-hierarchical system can be reposed as a hierarchical system through the use of auxiliary constraints, allowing the utilization of methods intended for strictly hierarchical systems.

The term Individual Disciplinary Feasible arises since each discipline satisfies its governing equations at each optimization iteration, and the system is only consistent at convergence. IDF is more centralized than MDF, and the dimension of the optimization problem is increased due to the coupling variables becoming decision variables. IDF maps to a design organization with a single project manager, making all of the design decisions and guiding the analysis groups into agreement.

The IDF formulation is given in Eq. (2). It is similar to the MDF formulation, except that the decision variables include the design variables **x** and the coupling variables **y**, and the auxiliary constraints $\mathbf{h_{aux}}(\mathbf{x},\mathbf{y})$ are included to ensure system consistency.

$$\begin{aligned} \min_{\mathbf{x}=[\mathbf{x}_\ell,\mathbf{x_s}],\mathbf{y}} \quad & f(\mathbf{x},\mathbf{y}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x},\mathbf{y}) = [\mathbf{g_1},\mathbf{g_2},\ldots,\mathbf{g_s}] \leq \mathbf{0} \\ & \mathbf{h}(\mathbf{x},\mathbf{y}) = [\mathbf{h_1},\mathbf{h_2},\ldots,\mathbf{h_s}] = \mathbf{0} \\ & \mathbf{h_{aux}}(\mathbf{x},\mathbf{y}) = \mathbf{y}(\mathbf{x},\mathbf{y}) - \mathbf{y} = \mathbf{0} \end{aligned} \tag{2}$$

In both a computational and an organizational context the parallel nature of IDF has an advantage over the sequential MDF approach. If parallel analysis tools (multiple analysis groups or parallel processors) are available, IDF can offer a significant compression of the design process. If a high level of centralization is acceptable, then IDF may be an ideal design strategy.

2.3 All-at-Once Formulation

The last MDO formulation covered here is the All-at-Once strategy (AAO). It is also referred to as Single-SAND-SAND, and sometimes just SAND. Occasionally the term AAO is used to refer to the AIO (i.e., MDF) approach; in the present terminology AAO and AIO are distinct formulations.

AAO is a highly centralized approach. Instead of utilizing analyzers to complete the analysis for each subspace, evaluators are used that compute only the residuals of the governing equations. The system optimizer is now saddled with three sets of decision variables: the original design variables **x**, the coupling variables **y**, and the state variables **s** (such as velocity fields, strain fields, etc.). AAO centralizes both design and analysis, but still distributes evaluation of governing equations. This can result in impressive efficiency, but is difficult to map to organizational structures or simulation tools due to its centralization and specialized structure.

The AAO architecture is illustrated in Fig. 6. The state variable vector is divided into the state variables for each subspace ($\mathbf{s} = [\mathbf{s_1},\mathbf{s_2},\ldots,\mathbf{s_N}]$), and the residuals for each subspace $\mathbf{w_i}$ are reported to the optimizer along with other pertinent values.

The formulation of the AAO approach is given in Eq. 3. It is similar to the IDF formulation, but includes an additional auxiliary constraint to ensure zero residuals at problem convergence,

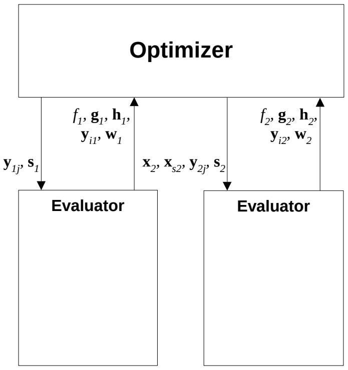
Optimizer
f_1, $\mathbf{g}_1$, $\mathbf{h}_1$, $\mathbf{y}_{i1}$, $\mathbf{w}_1$
$\mathbf{x}_1$, $\mathbf{x}_{s1}$, $\mathbf{y}_{1j}$, $\mathbf{s}_1$
$\mathbf{x}_2$, $\mathbf{x}_{s2}$, $\mathbf{y}_{2j}$, $\mathbf{s}_2$
f_2, $\mathbf{g}_2$, $\mathbf{h}_2$, $\mathbf{y}_{i2}$, $\mathbf{w}_2$
Evaluator
Evaluator

Figure 6. AAO ARCHITECTURE

and the decision variables include **x**, **y**, and **s**. This approach is called All-At-Once, since design, system analysis, and subspace analyses are all performed simultaneously.

$$\begin{aligned}
&\min_{\mathbf{x}=[\mathbf{x}_\ell,\mathbf{x}_\mathbf{s}],\mathbf{y},\mathbf{s}} && f(\mathbf{x},\mathbf{y}) \qquad (3)\\
&\text{subject to} && \mathbf{g}(\mathbf{x},\mathbf{y},\mathbf{s}) = [\mathbf{g_1},\mathbf{g_2},\ldots,\mathbf{g_s}] \leq \mathbf{0}\\
& && \mathbf{h}(\mathbf{x},\mathbf{y},\mathbf{s}) = [\mathbf{h_1},\mathbf{h_2},\ldots,\mathbf{h_s}] = \mathbf{0}.\\
& && \mathbf{h_{aux}}(\mathbf{x},\mathbf{y},\mathbf{s}) = \left\{ \begin{array}{c} \mathbf{y}(\mathbf{x},\mathbf{y},\mathbf{s}) - \mathbf{y} \\ \mathbf{w}(\mathbf{x},\mathbf{y},\mathbf{s}) \end{array} \right\} = \mathbf{0}.
\end{aligned}$$

2.4 Single-Level Summary

Selection of the appropriate formulation is facilitated by an understanding of the strengths and weaknesses of each [9]. For example, a key distinction is the dimension of the optimization problem. Figure 7 illustrates how MDF and AAO may be viewed as opposite extrema with respect to the number of decision variables the system optimizer explicitly controls. IDF is an intermediate occupant of this continuum.

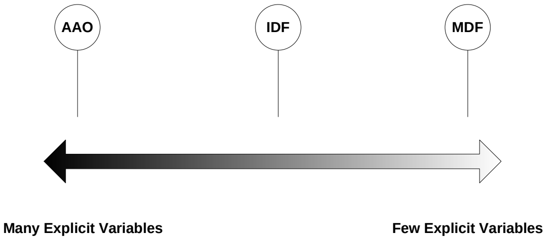

Figure 7. CONTINUUM OF SINGLE-LEVEL MDO FORMULATIONS

The AAO formulation does not allow for the use of legacy simulation tools, while the other two do. Only the MDF approach guarantees system consistency before optimization convergence, yet it lacks robustness due to issues explored in the next section. It was predicted in [9] that IDF would yield faster computation times than MDF. Investigation of this prediction is a focal point of this article.

3 FIXED POINT ITERATION

Coupled systems may be viewed as simultaneous systems of nonlinear equations, which may be solved with iterative methods such as the Newton-Raphson method and Fixed Point Iteration [12]. Fixed Point Iteration (FPI) is regularly employed as the analysis tool for the MDF formulation. Due to its intuitive implementation, MDF is the most frequently utilized MDO strategy [11]. However, it should not be applied without recognition of its shortcomings. This section reviews the nature of FPI, explores some issues with its use in MDF, and presents and proves a new form of convergence proof for FPI.

A simple coupled system with two subspaces is depicted in Fig. 8. The coupling variables y_{ij} are written in an implicit functional form. For example, the expression $y_{21}(y_{12})$ indicates that y_{21} is a dependent variable, and y_{12} is an independent variable. (Variables held constant during analysis are omitted here for clarity.) According to the implicit function theorem, new inverse functions may be found in terms of a new independent variable, i.e., $y_{21}(y_{12})$ may be posed as $y_{12}(y_{21})$.

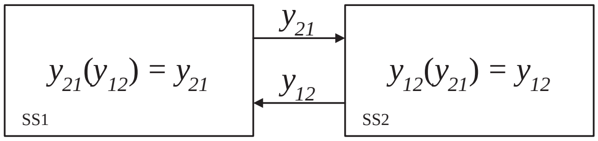

Figure 8. TWO-DIMENSIONAL COUPLED SYSTEM.

This system possesses feedback coupling, since y_{12} depends on y_{21} and *vice versa*. To employ FPI for analysis, an initial guess is made for the input to the subspace executed first. If a guess is made for y_{12}, $SS1$ is evaluated to obtain a value for y_{21}, which is subsequently used in the execution of $SS2$ to update the value of y_{12}. This output will not agree with the initial guess (unless the guess was made at a fixed point). The resulting value of y_{12} can then be used as an updated guess for the input to $SS1$. If the system meets certain criteria, this process will converge to a fixed point. The FPI algorithm for the two-dimensional example problem is [12]:

```
(Step 0) choose initial guess $y_{12}^0$, set $k = 0$
(Step 1) $k = k + 1$
(Step 2) $y_{21}^k = y_{21}(y_{12}^{k-1})$
(Step 3) $y_{12}^k = y_{12}(y_{21}^k)$
(Step 4) if $|y_{12}^k - y_{12}^{k-1}| < \varepsilon$ stop, otherwise go to (Step 1)
```

The superscript k indicates the iteration number, and ε is the maximum inconsistency allowed between subspaces. When the inconsistency is less than ε, the system is said to be consistent—the output of the system coupling variables is equal to the guess for coupling variable values. A point that produces this condition is called a fixed point $\mathbf{y}_p$, since further iterations will not change the location of the point. A fixed point $\mathbf{y}_p$ is defined by the relation $\mathbf{y}_p = \mathbf{y}(\mathbf{y}_p)$.

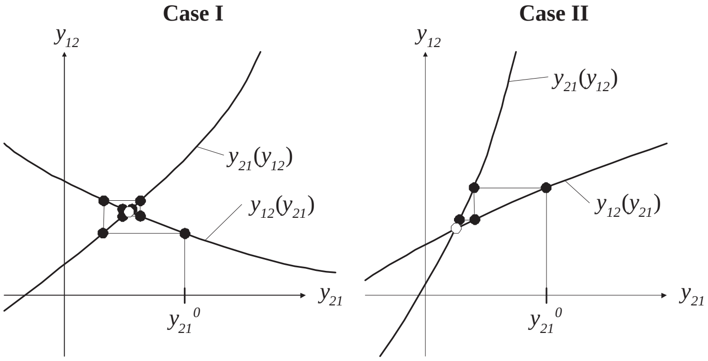

Figure 9. OSCILLATORY AND MONOTONIC FPI CONVERGENCE

Figure 9 illustrates the convergence behavior of FPI. Case I shows oscillatory convergence, and Case II shows monotonic convergence. The initial guess is y_{12}^0. The fixed point is located at the intersection of the functions. In the two convergent examples shown in Figure 9 it is clear that the equivalent relationships expressed in Eq. (4) hold in the neighborhood of a fixed point $\mathbf{y}_p$.

$$\left|\frac{\partial\left(y_{21}(y_{21})\right)}{\partial y_{21}}\right| > \left|\frac{\partial\left(y_{12}(y_{21})\right)}{\partial y_{21}}\right| \Leftrightarrow \left|\frac{\partial\left(y_{12}(y_{12})\right)}{\partial y_{12}}\right| > \left|\frac{\partial\left(y_{21}(y_{12})\right)}{\partial y_{12}}\right| \qquad (4)$$

Figure 10 illustrates a case where Eq. (4) does not hold. A fixed point exists, yet the algorithm diverges from this ‘repelling’ fixed point. If Eq. (4) does hold in the neighborhood of a fixed point $\mathbf{y}_p$, then FPI will converge, and $\mathbf{y}_p$ is termed an attractive fixed point. Some systems have multiple fixed points (Fig. 11), and FPI may not be capable of finding them all. The point FPI converges to depends upon the location of the starting point. Filled circles indicate attractive fixed points; empty circles indicate repelling fixed points.

If FPI is used as an analysis tool for optimization, it will never be known if there was a repelling fixed point that would have led to a better solution. For example, if the objective is to minimize $y_{12} + y_{21}$ from Fig. 11, then the lower left fixed point is the best solution, and cannot be found using FPI.

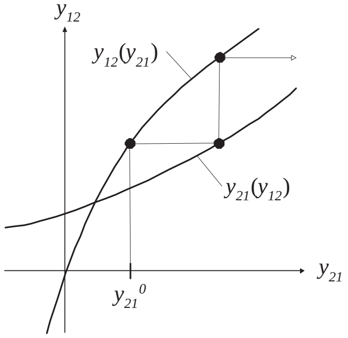

Figure 10. DIVERGENT FPI BEHAVIOR

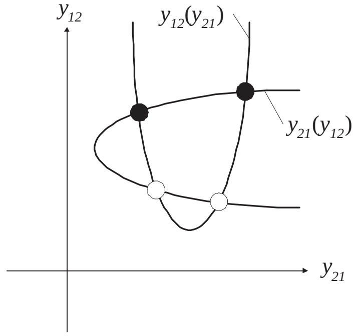

Figure 11. SYSTEM WITH MULTIPLE FIXED POINTS

The coupled system in Fig. 8 may be written as a single composite function of the form $x = g(x)$. A well-known proof of FPI convergence conditions for systems of this form is reviewed [12], and then it is shown here that these conditions are equivalent to Eq. (4). So Eq. (4) provides necessary and sufficient conditions for FPI convergence.

The FPI algorithm reduces to a recursion formula $y_{12}^{i+1} = g(y_{12}^i)$. Assuming a fixed point $y_{12p} = g(y_{12p})$ exists, we can subtract the recursion formula from the fixed point equation to obtain $y_{12p} - y_{12}^{i+1} = g(y_{12p}) - g(y_{12}^i)$. By the derivative mean value theorem, there exists a ξ such that $g'(\xi) = \frac{g(y_{12p})-g(y_{12}^i)}{y_{12p}-y_{12}^i}$. If g is C_1 continuous on the interval $[y_{12p}\, y_{12}^i]$, then $y_{12p} - y_{12}^{i+1} = g(y_{12p}) - g(y_{12}^i) = g'(\xi)(y_{12p} - y_{12}^i)$. The true error $\varepsilon_i = y_{12p} - y_{12}^i$ decreases if and only if $g'(\xi) < 1$, since $\varepsilon_{i+1} = g'(\xi)\varepsilon_i$. Therefore, if a fixed point y_{12p} exists, the condition that $|g'(y_{12})| < 1$ for all points in the neighborhood of y_{12p} is necessary and sufficient for convergence of FPI if the algorithm is given a starting point in the neighborhood of y_{12p}. Using the chain rule, this condition can be put in terms of the original problem.

$$\left|\frac{\partial(g(y_{12}))}{\partial y_{12}}\right| = \left|\frac{\partial(y_{12}(y_{21}(y_{12})))}{\partial y_{12}}\right| = \left|\frac{\partial(y_{12}(y_{21}))}{\partial y_{21}}\frac{\partial(y_{21}(y_{12}))}{\partial y_{12}}\right| < 1 \qquad (5)$$

The composite function may alternatively be written with the reverse evaluation order as $g = y_{21}(y_{12}(y_{21}))$. In this case the convergence condition takes the form of Eq. (6).

$$\left|\frac{\partial(y_{21}(y_{12}))}{\partial y_{12}}\frac{\partial(y_{12}(y_{21}))}{\partial y_{21}}\right| < 1 \tag{6}$$

Equation (4) is as yet unproven, only informally presented. It can be shown that $|g'(y_{21})| < 1$ and Eq. (4) are equivalent. First observe that:

$$|g'(y_{12})| = \left|\frac{\partial(g(y_{12}))}{\partial y_{12}}\right| = \left|\frac{\partial(y_{12}(y_{21}))}{\partial y_{21}}\frac{\partial(y_{21}(y_{12}))}{\partial y_{12}}\right| \tag{7}$$

Then beginning with the first instance of Eq. (4), and using the above expression, the proof is as follows:

$$\begin{aligned} \left|\frac{\partial(y_{21}(y_{21}))}{\partial y_{21}}\right| > \left|\frac{\partial(y_{12}(y_{21}))}{\partial y_{21}}\right| \quad &\Leftrightarrow \\ \left|\frac{\partial(y_{21}(y_{21}))/\partial y_{21}}{\partial(y_{12}(y_{21}))/\partial y_{21}}\right| > 1 \quad &\Leftrightarrow \\ \left|\frac{\partial(y_{12}(y_{21}))}{\partial y_{21}}\left(\frac{\partial(y_{21}(y_{21}))}{\partial y_{21}}\right)^{-1}\right| < 1 \quad & \end{aligned} \tag{8}$$

When the appropriate inverse function is formed, $\left(\frac{\partial y_{21}(y_{21})}{\partial y_{21}}\right)^{-1} = \frac{\partial y_{21}(y_{12})}{\partial y_{12}}$, and so

$$\left|\frac{\partial(y_{21}(y_{12}))}{\partial y_{12}}\frac{\partial(y_{12}(y_{21}))}{\partial y_{21}}\right| = |g'(y_{12})| < 1 \tag{9}$$

Therefore, if a fixed point y_{12p} exists, and if the condition given in Eq. (4) holds for all points in the neighborhood of y_{12p}, necessary and sufficient conditions for convergence of FPI are met if the algorithm is given a starting point in the neighborhood of y_{12p}. The process for proving the second instance of Eq. (4) may be either as above, or simply to show that the two instances are equivalent. The observations from Fig. 9 thus hold generally.

Thus, although FPI is straightforward to implement and enables the use of existing analysis tools, it presents several difficulties. In many cases FPI will not converge to an analysis solution. In addition, if multiple analysis solutions exist, FPI may not find them all. FPI is a sequential process, and does not allow for parallelization of tasks. When FPI is used as the analysis tool for MDF, all of these same issues arise. The optimization problem may be non-convergent, and when it converges the optimal solution may not be found.

4 EXAMPLE PROBLEM: TURBINE BLADE DESIGN

The MDF and IDF formulations are demonstrated on the design of a turbine blade. The turbine blade design problem is fully analytic and allows for variation of coupling strength. This latter feature permits the investigation of how these two formulations respond to increasing levels of coupling strength.

4.1 Design Problem Description

A turbine blade in a gas turbine engine is exposed to high temperature combustion gasses moving at very high velocity, and is subject to high forces due to aerodynamic drag force and centripetal acceleration. Modern alloys, such as Inconel, and advanced cooling systems are employed to ensure the durability of turbine blades. A drawing of a single turbine blade is shown in Fig. 12.

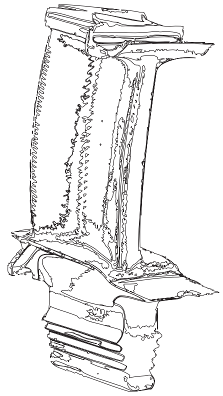

Figure 12. DIAGRAM OF A SINGLE GAS-TURBINE BLADE

Several phenomena were modeled in order to provide sufficient fidelity and to capture the design tradeoffs and coupling behavior:

1. Thermal expansion of the turbine blade in the axial direction.
2. Stress and elongation due to centripetal acceleration.
3. Aerodynamic drag force and the resulting bending stresses.
4. Dependence of thermal conductivity (k), elastic modulus (E), and rupture stress (σ_r) on temperature.

The blade's temperature profile depends upon its dilated length. Elongation due to thermal expansion or centripetal forces exposes more surface area to hot combustion gasses, affecting the heat transfer through the blade and the associated temperature profile. The model also captures the dependence of elastic modulus and thermal conductivity on temperature. Higher temperatures (caused by changes in length) result in lower stiffness, causing greater elongation. In summary, temperature depends on length, and length depends on temperature (Fig. 13).

One possible design objective is to maximize the thermal efficiency. Our simplified model cannot accurately predict this, but it can predict the mass m of the blade, and the heat transfer q through the blade. Both of these metrics should be minimized in order to maximize thermal efficiency.

4.2 Analysis Model

The turbine blade is modeled as a simple rectangular fin (Fig. 14). The design variables are the blade width w and thickness t. The blade has an initial undeformed length of L_0, and is

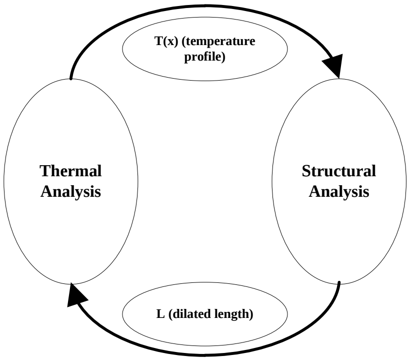

Figure 13. ANALYSIS COUPLING PRESENT IN THE TURBINE BLADE DESIGN PROBLEM

subjected to combustion gas temperature T_g and velocity v_g. The blade is affixed to a rotor with angular velocity ω, resulting in a centripetal acceleration f_{ac}. The axial position x is measured from the attachment point of the blade to the rotor. Four failure modes are considered: melting, interference between the blade and the turbine housing due to elongation, and structural failure due to bending stress σ_b or axial stress σ_a. Several simplifying assumptions were made: constant coefficient of thermal expansion α, no internal blade cooling, constant inertial force f_{ac} over the blade, and no lateral contraction. The dependence of thermal conductivity (k), elastic modulus (E), and rupture stress (σ_r) on temperature is modeled with curve fits based on empirical data.

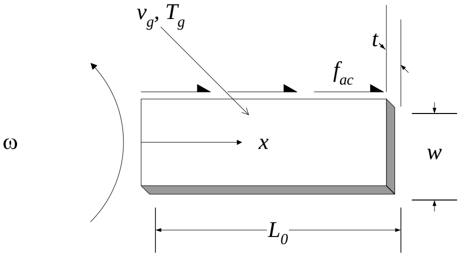

Figure 14. TURBINE BLADE MODEL SCHEMATIC

The complete turbine blade optimization problem is formulated as a multi-objective optimization problem (Eq. 10).

$$\begin{aligned} \min_{\mathbf{x}=[\mathbf{w},\mathbf{t}]} \quad & \{q,m\} \\ \text{subject to} \quad & g_1(\mathbf{x}) = T_{max} - T_{melt} \leq 0 \\ & g_2(\mathbf{x}) = \delta_{total} - \delta_{allow} \leq 0 \\ & g_3(\mathbf{x},x) = \sigma_a(x) - \sigma_r(T(x)) \leq 0 \\ & g_4(\mathbf{x},x) = \sigma_b(x) - \sigma_r(T(x)) \leq 0 \\ & \text{and} \quad 0 \leq x \leq L_0 + \delta_{total} \end{aligned} \tag{10}$$

where T_{max} is the maximum temperature in the blade, T_{melt} is the melting temperature, δ_{total} is the blade elongation, δ_{allow} is the initial clearance between the blade and housing, and $\sigma_a(x)$, $\sigma_b(x)$, and $\sigma_r(T(x))$ are the axial, bending, and rupture stresses along the blade. The analysis from each discipline (structural and thermal) follows.

Structural Analysis The structural objective function is the mass m of the blade (ρ is the blade density).

$$m = wtL_0\rho \tag{11}$$

The total elongation δ_{total} requires computation of both thermal expansion δ_{th}, and elongation due to axial acceleration δ_{ax}. The thermal expansion is dependent upon the change in temperature from the initial temperature T_0.

$$\begin{aligned} d\delta_{th} &= \alpha(T(x) - T_0)dx \\ \delta_{th} &= \int_0^{L_0} T(x)dx - \int_0^{L_0} \alpha T_0 dx \\ \delta_{th} &= \int_0^{L_0} T(x)dx - \alpha T_0 L_0 \end{aligned} \tag{12}$$

The last step is due to the constant thermal expansion α assumption. The temperature profile must be known in order to evaluate δ_{th} (supplied by the thermal analysis). The elongation due to centripetal acceleration is dependent on the rotor speed ω and radius r. First the axial load as a function of axial position is calculated. The portion of the blade outboard of a position x pulls with load $P_a(x)$. The tangential velocity of the blade $v = \omega r$ is assumed to be constant over the blade length, and is valid if $L_0 \ll r$.

$$\begin{aligned} P_a(x) &= \int_x^{L_0+\delta_{total}} \frac{v^2}{r}\rho A_c dx \\ &= \frac{v^2}{r}\rho wt(L_0 + \delta_{total} - x) \\ &= \omega^2 r\rho wt(L_0 + \delta_{total} - x) \end{aligned} \tag{13}$$

The axial deflection δ_{ax} is found using the axial load, and then summed with the thermal expansion δ_{th} to get the total elongation δ_{total}.

$$\delta_{ax} = \int_0^{L_0+\delta_{total}} \frac{P_a(x)dx}{A_c E(T(x))} \qquad (14)$$
$$= \omega^2 r\rho \int_0^{L_0+\delta_{total}} \frac{(L_0+\delta_{total}-x)}{E(T(x))}dx$$

$$\delta_{total} = \int_0^{L_0} T(x)dx - \alpha T_0 L_0 \qquad (15)$$
$$+\omega^2 r\rho \int_0^{L_0+\delta_{total}} \frac{(L_0+\delta_{total}-x)}{E(T(x))}dx$$

Observe that the value of the blade length after elongation $L = L_0 + \delta_{total}$ is required. An iterative solution procedure is required since this is unknown a priori.

The axial stress is a function of axial position, and is calculated with the relation $\sigma_a = P_a/A_c$, where P is the axial load and $A_c = wt$ is the cross sectional area as before.

$$\sigma_a(x) = \omega^2 r\rho(L-x) \qquad (16)$$

The calculation of the bending stress is less trivial. The aerodynamic load is calculated using $P_{aero} = \frac{1}{2}A_f C_D \rho v^2$, where $A_f = wL$ is the frontal area, C_D is the drag coefficient, ρ is the combustion gas density, and v is the combustion gas velocity (assumed to be perpendicular to the blade). The velocity v is constant, as is the combustion gas temperature and density. Therefore, the only variables in the calculation of P_{aero} are w and L. For convenience the constant $K = \frac{1}{2}C_D\rho v^2$ is defined, and the aerodynamic drag force is expressed as $P_{aero} = KwL$. Considering a position x on the blade, the total aerodynamic drag force acting on the blade outboard of that position is $P_{aero}(x) = Kw(L-x)$, and the bending moment at point x is $M(x) = Kw\frac{(L-x)^2}{2}$. Finally, using elementary beam theory, we arrive at the bending stress present in the blade at position x.

$$\sigma_b(x) = \frac{3K(L-x)^2}{4t^2} \qquad (17)$$

Thermal Analysis The thermal model was derived beginning with the steady-state heat equation using constant base temperature and an adiabatic tip boundary condition.

$$\frac{d^2T}{dx^2} + \left(\frac{1}{A_c}\frac{dA_c}{dx}\right) - \left(\frac{1}{A_c}\frac{h}{k}\frac{dA_c}{dx}\right)(T - T_\infty) = 0 \qquad (18)$$

The combustion gas temperature T_∞ is assumed constant, and the temperature dependence of thermal conductivity k on the average blade temperature is modeled with a curve fit. The average convection coefficient $\bar{h}$ was approximated using empirical correlations involving the average Nusselt number $\overline{N_u}$ and the Prandtl number Pr: $\overline{N_u} = \frac{\bar{h}w}{k_g} = CRe_D^m Pr^{1/3}$. The combustion gas conduction coefficient is k_g, $Re_D = vw/\nu$ is the appropriate Reynold's number, m is an empirical exponent of 0.731, and C is the heat capacity of the combustion gas. Solving for $\bar{h}$, and substituting values for the other parameters with SI units (at $T_\infty = 900°C$), we find: $\bar{h}(v,w) = 9.196v^{.731}w^{-.269}$. The temperature profile and the heat transfer through the blade into the rotor at the point of attachment are found through solution of the heat equation with the appropriate boundary conditions.

$$T(x) = \frac{\cosh(m(L-x))}{\cosh(mL)}(T_b - T_\infty) + T_\infty \qquad (19)$$

$$q = wt(T_b - T_\infty)\tanh(mL)\sqrt{2\bar{h}(w+t)wtk} \qquad (20)$$

where $m = \sqrt{2h(t+w)/ktw}$.

Curve Fits Surrogate models based on empirical data [13] were employed in order to capture temperature dependence. The rupture stress σ_r for Inconel X-750 was approximated using an *s-curve* function.

$$\sigma_r(T) = \frac{1300}{1+e^{0.011(T-675)}} \qquad (21)$$

The conductivity of the blade k was modeled using a linear curve fit. The dependence on average temperature $\overline{T}$ was captured from empirical data.

$$k(\overline{T}) = 6.8024 + 0.0172\overline{T} \qquad (22)$$

The final curve fit was a fourth-order polynomial fit to the modulus of elasticity for the blade material.

$$E(T) = 209.8 - 0.0487T - .0002T^2 + 6\cdot 10^{-7}T^3 - 6\cdot 10^{-10}T^4 \qquad (23)$$

4.3 Analysis Summary

Figure 15 illustrates the analysis problem structure, posed as a coupled, two-subspace system. The system has two shared design variables, and no local design variables. The entire design vector is $\mathbf{x} = [w,t]$. The responses of the thermal analysis ($SS1$) are the heat loss q and the temperature distribution $T(x)$, both of which depend on the design vector $\mathbf{x}$ and the input from the structural analysis (coupling variable y_{21}), the dilated length L. The responses of the structural analysis ($SS2$) are the mass m, dilated

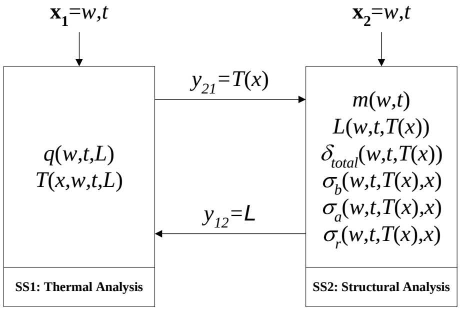

Figure 15. DIAGRAM SUMMARIZING THE TURBINE BLADE ANALYSIS

length L, total deflection δ_{total}, and the bending, axial, and rupture stresses σ_b, σ_a, and σ_r. These responses in general depend upon the design vector $\mathbf{x}$ and the input from the thermal analysis (coupling variable y_{12}), the temperature distribution $T(x)$. Table 1 shows the analysis results (using FPI) for a particular design. Function valued responses such as temperature distribution and stress distribution are displayed in Fig. 16.

Table 1. TURBINE BLADE ANALYSIS RESULTS.

parameters		variables		responses	
ρ	$8510\ kg/m^3$	w	$0.08\ m$	q	$0.2046\ W$
L_0	$0.05\ m$	t	$0.005\ m$	m	$0.1702\ kg$
α	$12.6 \cdot 10^{-6}\ m/K$			L	$0.507\ m$
r_b	$0.5\ m$			δ	$0.007\ m$
ω	$2100\ rad/s$				
δ_{max}	$0.05\ m$				
ρ_g	$3.522\ kg/m^3$				
C_d	2.0				
v	$100\ m/s$				
T_b	$300\ ^\circ C$				
T_g	$900\ ^\circ C$				
ε	$1.0 \cdot 10^{-8}$				

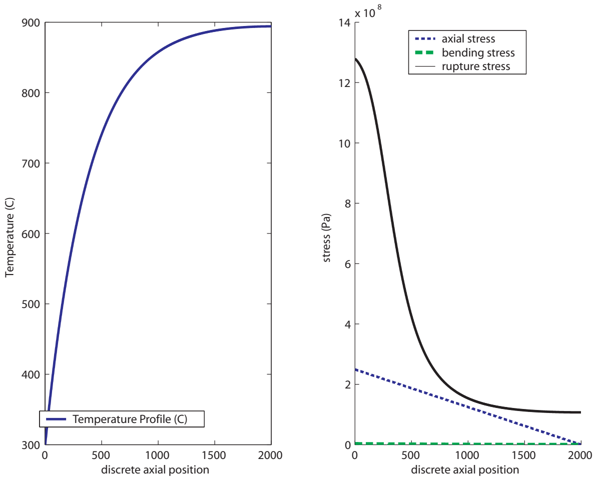

Figure 16. TEMPERATURE AND STRESS RESPONSE OF THE TURBINE BLADE ANALYSIS

5 SINGLE LEVEL COMPARISON: MDF VS. IDF

The turbine blade design was optimized with MDF (using FPI) and the IDF approaches. Since the problem has two objective functions (m and q), one (m) was treated as a constraint, and the other (q) was treated as the objective function. A parametric study on coupling strength was then performed, demonstrating the sensitivity of MDF and IDF to this factor.

5.1 MDF Implementation

The MDF formulation is:

$$\begin{aligned} \min_{\mathbf{x}=[\mathbf{w},\mathbf{t}]} \quad & q \\ \text{subject to} \quad & g_1(\mathbf{x}) = T_{max} - T_{melt} \le 0 \\ & g_2(\mathbf{x}) = \delta_{total} - \delta_{allow} \le 0 \\ & g_3(\mathbf{x},x) = \sigma_a(x) - \sigma_r(T(x)) \le 0 \\ & g_4(\mathbf{x},x) = \sigma_b(x) - \sigma_r(T(x)) \le 0 \\ & g_5(\mathbf{x},x) = m - m_{max} \le 0 \\ & 0 \le x \le L_0 + \delta_{total} \end{aligned}$$

The mass was constrained not to exceed 0.04 kg. The parameter values from Table 1 were used, and the optimal design was found to be $[w_*,\ t_*] = [0.0131,\ 0.0075]$.

5.2 IDF Implementation

The IDF formulation is:

$$\begin{aligned} \min_{\mathbf{x}=[\mathbf{w},\mathbf{t}],T(x),L} \quad & q \\ \text{subject to} \quad & g_1(\mathbf{x}) = T_{max} - T_{melt} \le 0 \\ & g_2(\mathbf{x}) = \delta_{total} - \delta_{allow} \le 0 \\ & g_3(\mathbf{x},x) = \sigma_a(x) - \sigma_r(T(x)) \le 0 \\ & g_4(\mathbf{x},x) = \sigma_b(x) - \sigma_r(T(x)) \le 0 \\ & g_5(\mathbf{x},x) = m - m_{max} \le 0 \\ & g_6(\mathbf{x},x) = T(x) - T(\mathbf{x},x) = 0 \\ & g_7(\mathbf{x},x) = L - L(\mathbf{x}) = 0 \\ & 0 \le x \le L_0 + \delta_{total} \end{aligned}$$

Once again, the parameter values from Table 1 were used. The optimal design found using IDF was almost identical to the MDF results: $[w_*,\ t_*] = [0.0128,\ 0.0074]$.

5.3 Parametric Study on Coupling Strength

In this parametric study the computation time required for both MDF and IDF solutions were recorded over a range of coupling strength levels. The coupling strength of the turbine problem may be varied by adjusting the modulus of elasticity E. A more compliant blade (small E) results in increased blade elongation and exposed surface area, increasing the impact that the structural analysis results have on the thermal analysis. A small value for E results in a strongly coupled system, whereas a perfectly rigid blade (infinite E) would result in completely decoupled structural and thermal analyses. The $E(T)$ curve from Eq. (23) was multiplied by a small scalar to explore the effects of increased coupling strength. Figure 17 illustrates the dependence of MDF and IDF computation time on this modulus multiplier, and hence the dependence on coupling strength. The plot elucidates the robustness of IDF with respect to coupling strength, and the sensitivity of MDF to the same factor. Weakly coupled systems are more efficiently solved with MDF, while strongly coupled systems require excessive iterations for the inner analysis loops of MDF. The computation time required for the IDF approach is virtually constant for all levels of coupling strength investigated here.

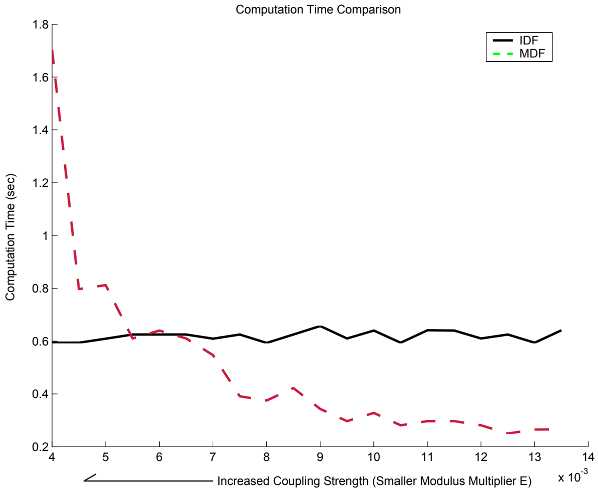

Figure 17. COMPARISON OF MDF AND IDF SOLUTION TIME AS A FUNCTION OF COUPLING STRENGTH.

6 CONCLUSIONS

Fundamental formulations of single-level MDO problems were critically reviewed. MDF and IDF were implemented in the solution of a newly developed, fully analytic, turbine blade problem. The problem illustrated the nature of feedback coupling in an MDO system, and elucidated the nature of MDF and IDF when applied to a coupled problem. IDF was shown to be a more efficient choice for strongly coupled problems. Still, both MDF and IDF are centralized approaches, reducing their desirability as tools for larger design organizations or as model structures for distributed decision making.

ACKNOWLEDGMENTS

This work was partially supported by a National Science Foundation Graduate Research Fellowship, and by the Automotive Research Center, a US Army Center of Excellence at the University of Michigan. Any opinions expressed in this publication are only those of the authors.

REFERENCES

[1] Allison, J. T., 2004. "Complex system optimization: A review of analytical target cascading, collaborative optimization, and other formulations". Master's thesis, Department of Mechanical Engineering, University of Michigan.

[2] Allison, J. T., Kokkalaras, M., and Papalambros, P. Y., 2005. "On the use of analytical target cascading and collaborative optimization for complex system design". In 6th World Conference on Structural and Multidisciplinary Optimization.

[3] Wagner, T. C., 1993. "A general decomposition methodology for optimal system design". PhD thesis, University of Michigan.

[4] Sobieszczanski-Sobieski, J., 1990. "Sensitivity of complex, internally coupled systems". *AIAA Journal,* **28**, pp. 153–158.

[5] Balling, R. J., and Wilkinson, C. A., 1997. "Execution of multidisciplinary design optimization approaches on common test problems". *AIAA Journal,* **35**, pp. 178–186.

[6] Sosa, M. E., Eppinger, S. D., and Rowles, C. M., 2003. "Identifying modular and integrative systems and their impact on design team interactions". *ASME Journal of Mechanical Design,* **125**, pp. 240–252.

[7] Sosa, M. E., Eppinger, S. D., and Rowles, C. M., 2004. "The misalignment of product architecture and organizational structure in complex product development". *Management Science,* **50**, pp. 1674–1689.

[8] Chanron, V., and Lewis, K., 2004. "Convergence and stability in distributed design of large systems". In ASME Design Engineering Technical Conference. DETC2004-57344.

[9] Cramer, E. J., Dennis Jr., J. E., Frank, P. D., Lewis, R. M., and Shubin, G. R., 1994. "Problem formulation for multidisciplinary optimization". *SIAM Journal of Optimization,* **4**, pp. 754–776.

[10] Balling, R. J., and Sobieszczanski-Sobieski, J., 1994. "Optimization of coupled systems: a critical overview of approaches". In 5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA Paper 94-4330.

[11] Braun, R. D., 1996. "Collaborative optimization: An architecture for large-scale distributed design". PhD thesis, Stanford University, April.

[12] Chapra, S. C., and Canale, R. P., 1998. *Numerical Methods for Engineers*, third ed. McGraw-Hill.

[13] Matweb material property data. http://www.matweb.com/. Accessed April 9, 2004.