# A Comparison of Formal and Heuristic Strategies for Iterative Convergence of a Coupled Multidisciplinary Analysis

**K.F. Hulme and C.L. Bloebaum**
Multidisciplinary Optimization and Design Engineering Laboratory (MODEL)
Department of Mechanical and Aerospace Engineering
State University of New York at Buffalo - 1012 Furnas Hall
Buffalo, NY 14260 (USA)

Keywords: Multidisciplinary Design Optimization, CASCADE, coupled analysis, nonlinear equations, simulation, data fusion, neural network

**1. Abstract**
A primary goal of Multidisciplinary Design Optimization (MDO) is decomposition - to break down a large and complex engineering system into a grouping of smaller, more tractable, yet still inter-related (coupled) subsystems. The problem with this procedure is that the resulting decomposition is typically not hierarchical in nature. Because of this lack of hierarchy, the subsystem outputs (behavior variables) must be initialized to some set of values, and iteratively converged thereafter, which can be a timely and computationally costly procedure. This research effort will compare various techniques for attaining a converged solution to a coupled multidisciplinary analysis. These techniques include formal strategies such as Fixed-point Iteration, which is an easily implemented but inefficient means for iterative convergence, and Newton's Method, which is an efficient but costly and often times unstable means for iterative convergence. A new heuristic convergence strategy is under development which hopes to demonstrate characteristics that will serve as a "middle ground" to those of the aforementioned formal strategies. Namely, the new convergence strategy hopes to demonstrate a higher level of efficiency than Fixed-point Iteration, and a higher level of robustness than Newton's Method. The new heuristic strategy will exhibit neural network-like characteristics in its implementation. Further, the new strategy will utilize an approach that is analogous to a systems-based procedure known as data (sensor) fusion. Namely, information from numerous subsystems that require the same behavior variable as input will be intelligently "fused" together to form a single predictive estimate for the true numerical value of the given behavior variable. Due to proprietary limitations, "real world" multidisciplinary problem data is difficult to obtain and due to enormity of size and complexity, difficult to utilize. Hence, the convergence strategies will be compared using two forms of multidisciplinary test problems: CASCADE-generated analysis simulations, and Class I problems from the MDO Test Suite.

**2. Motivation**
The overall design process associated with a non-serial system decomposition is illustrated generically in Figure 1. The design must first be initialized to some logical set of starting values. Perhaps, "current" design data is utilized as the initial design, upon which improvements are desired. Design initialization is typically followed by a system analysis, which is a highly iterative process. Subsequent to the system analysis is a sensitivity analysis, after which the computed sensitivities are then used in a formal optimization procedure. During the optimization, the converged behavior variables (subsystem outputs) computed during the system analysis are held constant, and the design variables, which were held constant during the iterative system analysis, are changeable parameters. After the optimization procedure, the design variables have changed from their starting values. Because the behavior variables are a function of these design variables, they will have changed as well. The design is updated, a convergence check is implemented, and the entire procedure repeats itself until a converged solution is attained.

The design cycle that has just been described has been referred to in literature as the "Multiple-Discipline Feasible", or MDF approach. It has been demonstrated more than any other approach on non-hierarchic multidisciplinary examples [1]. An advantage of MDF is its intuitive implementation; most design managers are familiar with the approach. The primary disadvantage is that it is potentially a very time and cost consuming approach. The large computational cost tends to be expended during both the sensitivity analysis, and especially during the iterative system analysis. A large number of recent research efforts have focused on increasing the efficiency of the iterative system analysis. The present research effort likewise focuses on increasing the efficiency of the iterative system analysis, in the interest of reducing the overall number of iterations expended per design cycle.

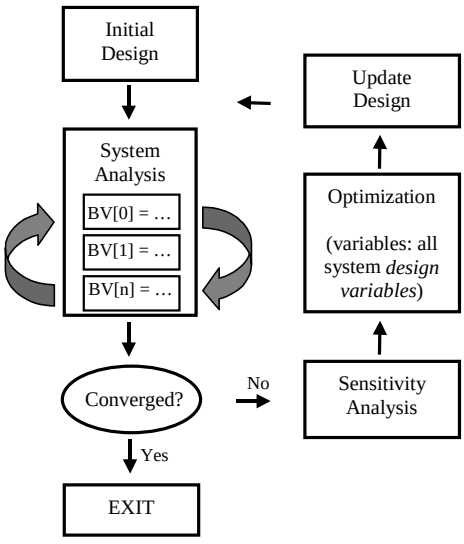


**Figure 1: MDF process cycle**

**3. Heuristic Analysis Convergence**

This section will present a discussion on the specifics of the proposed heuristic convergence technique. This technique serves as an alternative to two widely used formal techniques for iterative convergence, Fixed-point Iteration (FPI) and Newton's Method (NM). (This paper assumes that the reader has a basic understanding of these convergence methods). Numerous optimization and MDO-based concepts were combined to construct the new technique. Each of these background concepts will first be presented, followed by a discussion of how these concepts were culminated to form a useful new algorithm. The background discussion will begin with Neural Networks, the structure of which is fundamental to the present research effort.

Neural Networks

In its most general form, a Neural Network (NN) is a "machine" that is designed to model the way in which the brain performs a particular task or function of interest [2]. NN's typically perform useful computations through a process of learning, and are hence often referred to as "learning computers". To achieve good performance, NN's employ an interconnection of simple computing cells which are commonly referred to as "neurons". A neuron typically has three basic elements: a set of synapses (inputs) and associated weights, an adder to sum the input signals, and an activation function which limits the amplitude of the output signal. A subcomponent of the synapses is a *fixed* input signal, which can be positive or negative in nature.

This model of a neuron can be extended to an MDO context by realizing that each subsystem output (behavior variable) can be simulated as a neuron The fixed inputs are the design variables, which are known quantities, and are held constant during the convergence procedure. The synapses are the behavior variables which are required inputs to compute the given output behavior variable, whose values are unknown. A corresponding weight could be defined for each input behavior variable, the values of which are initialized and to be determined. The activation function in such a proposed model would be equal to one, allowing for the output of the neuron (i.e. the "computed" value of the given behavior variable) to be equal to the full summation of all design and behavior variable terms. Hence, a full MDO model would require n concurrent neurons, where n is the number of behavior variables in the multidisciplinary system.

This NN model will establish a "computed" value for all behavior variables in the system, which will likely be different than the corresponding "initial" value for each. Hence, a sub-optimization problem can be generated which will minimize (correct) the values of the computed errors for each behavior variable to equal zero. This procedure is typically referred to as "error correction learning" in NN theory. The specifics of this sub-optimization problem will soon be discussed. First, another concept must first be presented, which is in fact the primary motivation behind this research effort: Data Fusion.

Data Fusion

The Neural Network-based approach will simultaneously and separately attain solutions for each of the subsystem behavior variables. The problem is that most behavior variables are typically required as input by more than one subsystem. Hence, multiple subsystem neurons may concurrently arrive at different values for the weights corresponding to the same behavior variable input value. At convergence, ALL weights corresponding to the same input quantity must be equal. As a result of this discrepancy, some means of coordination must be devised to "blend" these non-equivalent weight values together to form a single intelligent estimate for the next iteration. This process of "blending" is where data fusion comes into consideration.

Data Fusion techniques combine data from multiple "sensors" to achieve improved accuracy's and more specific inferences than could be achieved by the use of a single sensor alone [3]. The impetus for the incorporation of Data Fusion concepts into MDO stemmed from a preliminary investigation conducted by researcher Bryan J. Moulton during the summer of 1997 [4]. Moulton's efforts focused on increasing the efficiency of the entire MDF solution cycle by eliminating the requirement (and computational overhead) of assigning design variables to individual subsystems. Moulton's proposed fusion approach serves as an alternative to problems presently solved by the Concurrent Subspace Optimization (CSSO) approach [5]. Like the CSSO approach, the present research exploits sensitivity information to guide the convergence process. Unlike Moulton's approach, the present research effort diverts the attention away from the optimization cycle (and design variable allocation altogether) and concentrates solely on increasing the efficiency of the iterative system analysis, through this coordination procedure. With all of the background components described, the specifics of the proposed convergence algorithm can now be discussed.

Proposed heuristic convergence algorithm

The proposed heuristic algorithm begins by modeling each subsystem output (behavior variable) as a separate, concurrently executed neuron. See Figure 2, which is divided into 4 "segments", (a), (b), (c), and (d). Each will now be described.

a. Segment (a)

Each neuron will have fixed input in the form of design variables, and behavior variable input quantities with corresponding variable weights, which together represent the full input behavior variables. There will be one neuron for each "output" behavior variable. Each behavior variable equation has been re-stated in "equality constraint" form, whereby all non-zero terms are moved to one side of the equation. Hence, the representative equation corresponding to each behavior variable need not be separable; this could serve as a tremendous advantage. Because of this, the output behavior variables of each equation are also synapses in the NN equation models. This implies that the desired output of each neuron - the behavior variable equation *residual* - is zero.

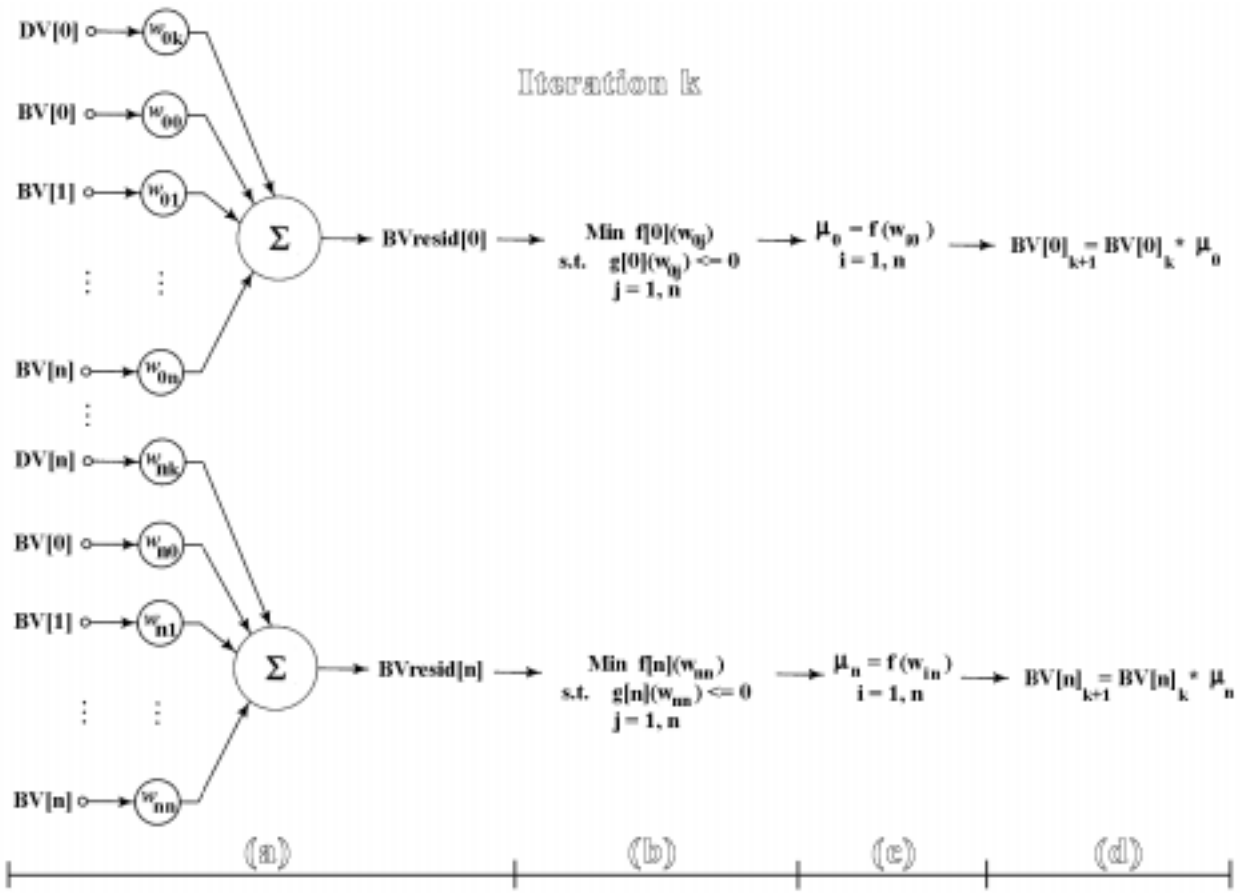


**Figure 2: Proposed convergence algorithm schematic**

b. Segment (b)

To reduce the residuals to zero, a sub-optimization problem is created. Each neuron will have an associated objective function and an associated equality constraint function. The "duty" of the objective function is to keep the values of the weights as close to their initial value as possible. The "duty" of the constraint function is to ensure that at the end of the optimization cycle, the behavior variable equation residual is equal to zero, or very near zero. The forms of the optimization functions for achieving error minimization are given as follows:

$$\begin{aligned} \text{Minimize } F &= \Sigma\, C_o(C_i(x_j - \mu_j))^2 \quad j = 1, n \\ \text{subject to: } g &= \Sigma\, D_o(D_i(BVresid_j))^2 \quad j = 1, n \\ lb &\le x_j \le ub \quad j = 1, n \end{aligned} \tag{1}$$

Here, $C_i$, $C_o$, $D_i$, and $D_o$ are user-defined constants, $x_j$ is the fusion weight corresponding to behavior variable j, and $BVresid_j$ is the equation residual corresponding to behavior variable j. $\mu_j$ is the fusion estimate value corresponding to each fusion weight, which was computed on the previous iteration. The optimization sub-problem has been set up such that the fusion weights will be changed enough to satisfy the constraint equations, while remaining as close to its corresponding fusion estimate value as possible. The fusion weights are assigned to behave within some set of lower and upper bounds, lb and ub, respectively, which are user-defined. The Automated Design Synthesis (ADS) optimizer [6] has been implemented to handle the NN sub-optimizations. For all example problems presented in this study, an exterior penalty function strategy has been used, along with a variable metric optimization search, and a combined Golden Section / Polynomial interpolation 1-D search.

c. Segment (c)

Once all neuron residuals have been reduced to zero, a data fusion procedure is required. This is because the concurrent neuron models will likely arrive at conflicting values for the weights that correspond to the same behavior variable. A converged system analysis occurs when all weight "instances" corresponding to a given behavior variable are equal, for all system behavior variables. The most successful data fusion technique developed thus far is referred to as "derivative-based" fusion. In this scheme, derivatives of each equation residual are taken with respect to each behavior variable weight found in that respective equation. (These derivative quantities are denoted “BVsens” below.) This is analogous to taking the derivative of an equality constraint equation with respect to its design variables. Here, the weights associated with a behavior variable equation whose

sensitivity was found to be the largest will be weighted proportionally larger than those of a behavior variable equation whose sensitivity was found to be smaller:

$$\mu_i = \frac{\sum_{j=1}^{n} (w_{ij} * (BVsens)_j))}{\sum_{k=1}^{n} (BVsens)_k} \qquad j = 1, n \tag{2}$$

Here, "i" represents the fusion variable for which this computation is taking place, "j" and "k" are generic indices, and "n" is the total number of behavior variables (output neurons) in the system.

d. Segment (d)

Once a "fused" weight is calculated for each of the system behavior variables, a new estimate of each behavior variable is calculated for subsequent iterations. This is accomplished by multiplying the input value of a given behavior variable (on a given NN cycle) by its corresponding calculated fused weight, for all system behavior variables:

$$BVout_i = BVin_i * \mu_i \quad i = 1, n \tag{3}$$

These new estimates on analysis cycle "k" are compared to their corresponding values on analysis cycle "k-1". If, for all behavior variables, the difference between "$BV_k$" and "$BV_{k-1}$" is less than some convergence threshold, then the heuristic analysis algorithm has converged. If not, the entire process repeats itself until convergence is achieved, or until some predetermined number of cycles are executed.

e. General characteristics

The proposed convergence technique is not wholly deterministic; some additional parameters contribute to the convergence procedure which contain some degree of uncertainty. These include a [0,1] randomness factor which pre-multiplies the relative finite difference coefficient in the optimizer. In addition, the user can assign a “decimal improvement” parameter, which decides the degree of improvement that must be attained on each iteration to accept the new fusion-generated move. A default value of 1.0 implies that any reduction in the sum of the behavior variable residuals over the corresponding sum (on the previous iteration) will be accepted. A value of 1.25 implies that the residual sum would be allowed to increase by as much as 25%, and the fusion-based move would still be accepted. Similar to heuristic optimization techniques such as Simulated Annealing, there is some probability of accepting a design point with an “inferior” objective function, in hopes of escaping any local minima that reside.

**4. Test Problem Description**

The rigorous testing of iterative convergence strategies requires the use of a wide variety of stable test systems. For this reason, the CASCADE (Complex Application Simulator for the Creation of Analytical Design Equations) simulator has been used for the generation of analytical test systems. A thorough description of CASCADE can be found in past literature [7]. In this research effort, three different fully-coupled test systems have been generated by CASCADE, each having three subsystems, one behavior variable (output) per subsystem, and one design variable term per behavior variable. Hence, all three systems will have the same coupling structure, but different coupling semantics. Figure 3 illustrates the coupling structure of all three test systems, each of which will have differing semantic characteristics.

Though the use of CASCADE-generated test systems is seminal for this research effort, the authors also desired a secondary means on which the proposed heuristic convergence strategy could be tested. These secondary problems should be less "generic" in nature, and should demonstrate characteristics of a "real-world" MDO problem. The obvious choice for the source of such problems is the MDO Test Suite (MDOTS) at NASA Langley, which was initiated in 1996 [8]. The MDOTS provides an on-line means for collecting, distributing, and maintaining standard test problems for MDO research.

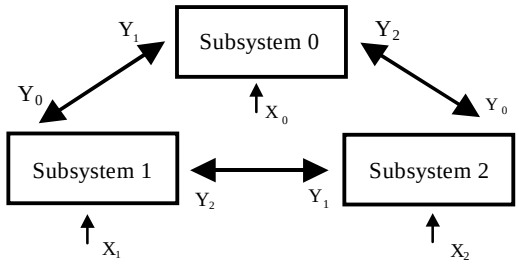


**Figure 3: Structural representation of all 3 CASCADE systems**

Numerous classes of problems are available on the MDOTS, which vary in size, complexity, and scope. For this effort the problems that were chosen are commonly known as the "Heart Dipole (HD)" and the "Combustion of Propane (COP)" problems. Both problems are, in their simplest form, a series of n nonlinear equations and n unknowns, where n is equal to 8 for the HD problem, and 11 for the COP problem. These test problems were modified slightly from their original form, such that they could be used for the research purposes of the present effort. The systems of equations were modified to demonstrate separability in each of the n unknowns, and to demonstrate some degree of convergence stability from numerous starting points, such that FPI and NM could be compared to the proposed heuristic technique.

**5. Results**

The first CASCADE test system required 37 iterations to converge to one decimal place (0.1) via FPI, and diverged via Newton's Method, from the given starting point. The sum of the three residuals at the starting point was 10.27 for the normalized equation set. The proposed heuristic convergence technique performed well for this test system; after 15 iterations, the normalized residual sum was reduced to 0.007. From this point, 15 additional FPI iterations were required to converge the equations to one decimal place. This information, and analogous information for the second and third CASCADE systems, as well as the HD and COP MDOTS problems, is summarized in Table 1.

| | CASCADE 1 | CASCADE 2 | CASCADE 3 | HD | COP |
|---|---|---|---|---|---|
| a. Initial residual sum | 10.27 | 9.51 | 29013.76 | 6.194 | 18.996 |
| b. Final residual sum | 0.007 | 0.012 | 0.111 | 0.031 | 0.084 |
| c. Iterations of proposed algorithm | 15 | 12 | 35 | 25 | 35 |
| d. Subsequent FPI iterations | 15 | 6 | 1 | 12 | 61 |
| e. Total iterations to converge (c+d) | 30 | 18 | 36 | 37 | 96 |
| f. FPI iterations to converge | 37 | 24 | 77 | 60 | 72 |
| g. NM iterations to converge | diverges | 7 | 5 | 7 | diverges |

**Table 1: Result summary**

Finally, Figure 4 is a chart that summarizes the results (formal and heuristic) for all test systems side by side. The convergence techniques are abbreviated FPI, NM, and "PHA", which stands for "proposed heuristic algorithm". Plotted on the y-axis are the number of iterations required to converge the system to the appropriate number of decimal places, as outlined in the previous sub-sections. Note again that NM diverges (denoted by "∞") for the first CASCADE system and for the MTS COP system.

**6. Discussion**

a. CASCADE test systems

The proposed convergence algorithm proved successful for all three CASCADE-based test systems. Here, "success" is measured by the fact that straight FPI iteration was bettered by the proposed technique, in some cases, substantially so. For test system #1, 15 iterations of the proposed technique were implemented, followed by 15 more iterations of straight FPI to converge the system. Hence, this hybrid strategy converged the system in a total of 30 iterations, 7 iterations (18.9%) fewer than straight FPI. Recall that NM diverged from the given starting point. For test system #2, 12 iterations of the proposed strategy were implemented to greatly reduce the initial residuals; this was followed by 6 iterations of straight FPI to achieve full convergence. Hence, this hybrid strategy converged the system in a total of 18 iterations, 6 iterations (25%) fewer than straight FPI.

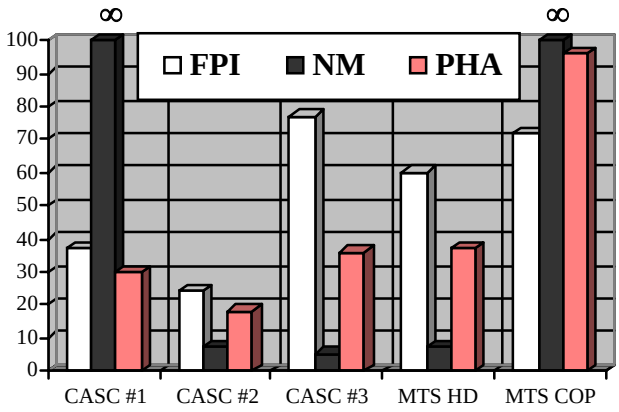


**Figure 4: Overall result summary**

NM converged in a mere 7 iterations for this test system. Finally, for test system #3, 35 iterations of the proposed strategy were implemented to greatly reduce the initial residuals; this was followed by only 1 iteration of straight FPI to achieve full convergence. Hence, this hybrid strategy converged the system in a total of 36 iterations, 41 iterations (53.2%) fewer than straight FPI. NM converged in a mere 5 iterations for this test system.

b. MDO Test Suite modified test systems

Some degree of success was seen for these test systems. For the modified Heart Dipole system, 25 iterations of the proposed strategy were implemented to reduce the initial residuals; this was followed by 12 iterations of straight FPI to achieve full convergence. Hence, this hybrid strategy converged the system in a total of 37 iterations, 23 iterations (38.3%) fewer than straight FPI. NM converged in a mere 7 iterations for this test system. For the Combustion of Propane system, the proposed convergence strategy appeared to work well; the residual sum was reduced to a value of less than 0.1 in a total of 35 iterations. From this point however, a total of 61 iterations were required to converge the system via straight FPI. Hence, this hybrid strategy converged the system in a total of 96 iterations, 24 iterations (33.3%) greater than straight FPI. NM diverged for this test system.

c. General observations

The proposed technique is very good at reducing large initial residuals down to values close to zero, where straight FPI can then be implemented to drive the reduced residuals down to zero. Clearly a large research issue that stems from the current effort is

the ability to identify the threshold point at which the proposed technique is “turned off” and straight FPI is “turned on” to finish the job. For the present effort, the authors discovered that generally, an appropriate number of iterations of the proposed technique was to be k/2, where k is the number of iterations required to converge the system using straight FPI (for every iteration) from the given starting point. The general trend seen in the experimental testing is that for “accurate” starting points that turn out to be “close enough” to the true solution, a formal derivative based scheme such as NM is clearly the convergence strategy of choice. NM will be reliable in cases where the analysis solution space is known to be convex. In realistic MDO applications, this will rarely be the case. Alternatively, when the starting point is known to be a “substantial distance” from the converged solution, or if a relation between the initial and final solutions cannot be identified (proving NM to be an unsafe choice for iterative convergence), the proposed convergence technique has shown to be reliable for reducing large residuals, and can be supplemented with numerous iterations of FPI to achieve full convergence. This hybrid convergence scheme has shown to increase efficiency by a substantial percentage.

The proposed convergence technique involves numerous concepts and procedures, and is clearly more complicated and computationally intensive than is FPI. The ideal application of this technique is envisioned to be a large-scale MDO problem, where iteration time is "large" (i.e. on the order of “days/months” as opposed to “seconds/minutes”). Hence the implementation of the proposed technique will see an increase in time and computational cost per iterative cycle, in the hopes of decreasing the total number of iterations to attain a converged design. NM is clearly more computationally intensive than FPI, as it requires derivative computation and a matrix inversion, which is usually a high cost endeavor. The authors suspect that the computational cost of the proposed algorithm is also larger than that of the NM implementation. While the results of the proposed convergence technique have not yet shown to be as good as those attained by NM, the technique has shown to have the robustness of FPI, which is a large shortcoming of NM.

**7. Conclusions and Future Work**

Overall, the proposed technique was found to be extremely useful for reducing large equation residuals, but not efficient at attaining tight convergence on its own accord. FPI was implemented to supplement the proposed algorithm after a “sufficient” number of iterations had elapsed. This hybrid convergence scheme has shown to increase efficiency by a substantial percentage. Note that due to the comparatively large computational expense of the proposed technique, it will only be useful in “true” multidisciplinary designs, where the overall design time is substantially long. In such situations, an increase in computational overhead each iteration will turn out to be beneficial, as it is hoped that the overall number of iterations will be reduced by a sufficient factor, outweighing the added expense.

Suggestions for future work regarding this research topic are many. The authors feel that the sensitivity-based algorithm can be further enhanced such that its performance can more closely match that of NM from “sufficiently close” starting designs. The authors are quite sure that the failure of COP is not due to its problem size, as it was the largest system tested. This hypothesis should be put to the test by examining both a Class II MTS problem of at least 15-20 behavior variables, and a larger CASCADE system. Clearly, the proposed technique is well suited for parallel computation. The authors plan to perform the sub-optimizations (for a sufficiently large problem) in parallel through the use of the Parallel Virtual Machine (PVM) message passing paradigm. Further, the authors plan to perform these parallel runs on an ultra-fast networking paradigm known as Asynchronous Transfer Mode (ATM), and compare these results to similar results attained across a standard Ethernet platform.

**8. Acknowledgment**

The authors would like to acknowledge the partial funding support of The National Science Foundation from the following Presidential Faculty Fellow (PFF) grant: NSF DMI9553210.

**9. References**

[1] Balling, R.J., and Sobieszczanski-Sobieski, J., "Optimization of Coupled Systems: A Critical Overview of Approaches." AIAA Paper 94-4339, September, 1994.

[2] Haykin, Simon, "Neural Networks - A Comprehensive Foundation." Macmillan College Publishing, New York, 1994.

[3] Hall, David L., and Llinas, James, "An Introduction to Multisensor Data Fusion." Proceedings of the IEEE, Vol. 85, No. 1, January, 1997.

[4] Moulton, B.J., Unpublished research - Preliminary investigation concerning the incorporation of Data Fusion concepts into the MDO solution procedure. SUNY at Buffalo MODEL laboratory, May-August, 1997.

[5] Bloebaum, C.L., Hajela, P., and Sobieszczanski-Sobieski, J., “Non-Hierarchic System Decomposition in Structural Optimization.” Engineering Optimization, Vol. 19, pp. 171-186, 1992.

[6] Vanderplaats, G., "ADS - A FORTRAN Program for Automated Design Synthesis Version 1.10" (user's manual), University of California, Santa Barbara, California, September, 1985.

[7] Hulme, K.F., and Bloebaum, C.L., "Development of a Multidisciplinary Design Optimization Test Simulator." Structural Optimization, Volume 14, Number 2-3, October, 1997.

[8] Padula, S.L., Alexandrov, N., and Green, L.L., "MDO Test Suite at NASA Langley Research Center." 6th Symposium on Multidisciplinary Analysis and Optimization, Bellevue, WA, September 1996.