

OPTIMIZATION ISSUES WITH COMPLEX ROTORCRAFT COMPREHENSIVE ANALYSIS

**Joanne L. Walsh
Katherine C. Young
NASA Langley Research Center
Hampton, VA**

**Frank J. Tarzanin
Joel E. Hirsh
and
Darrell K. Young
The Boeing Company
Philadelphia, PA**

AIAA Paper No. 98-4889

**Presented at
7th AIAA/USAF/NASA/ISSMO Symposium on
Multidisciplinary Analysis and Optimization
St. Louis, Missouri
September 2-4, 1998**

OPTIMIZATION ISSUES WITH COMPLEX ROTORCRAFT COMPREHENSIVE ANALYSIS

Joanne L. Walsh*, Katherine C. Young†
 NASA Langley Research Center, Hampton, VA

Frank J. Tarzanin‡, Joel E. Hirsh§, and Darrell K. Young¶
 The Boeing Company, Philadelphia, PA

Abstract

This paper investigates the use of the general purpose automatic differentiation (AD) tool called Automatic Differentiation of FORTRAN (ADIFOR) as a means of generating sensitivity derivatives for use in Boeing Helicopter's proprietary comprehensive rotor analysis code (V11). ADIFOR transforms an existing computer program into a new program that performs a sensitivity analysis in addition to the original analysis. In this study both the pros (exact derivatives, no step-size problems) and cons (more CPU, more memory) of ADIFOR are discussed. The size (based on the number of lines) of the V11 code after ADIFOR processing increased by 70 percent and resulted in substantial computer memory requirements at execution. The ADIFOR derivatives took about 75 percent longer to compute than the finite-difference derivatives. However, the ADIFOR derivatives are exact and are not functions of step-size. The V11 sensitivity derivatives generated by ADIFOR are compared with finite-difference derivatives. The ADIFOR and finite-difference derivatives are used in three optimization schemes to solve a low vibration rotor design problem.

Introduction

Comprehensive rotorcraft analyses are complex and multidisciplinary in nature. Over the past 15 years, optimization methods have been applied to increasingly sophisticated rotorcraft design problems¹⁻¹³. Most of these authors use a single gradient-based optimizer. An important aspect of any optimization method that uses a gradient-based optimizer is the sensitivity analysis which calculates the derivatives of the objective function

and constraints with respect to the design variables. In most rotorcraft optimization applications¹⁻⁷, finite-difference techniques are used to calculate the derivatives. However, some researchers¹⁰ have derived analytical expressions for the derivatives and added these to the rotorcraft analysis codes.

Finite-difference derivatives are easy to implement but are step-size dependent and can be difficult to use, particularly if the optimal step-size depends on the value of a design variable. In addition, small variations or inaccuracies in the function values lead to large variations in the derivatives. This is particularly true in Boeing's V11 code¹¹⁻¹³, where the calculation is the result of numerous iteration loops that are converged to finite tolerances. Analytical derivatives are not step-size dependent. However, derivation and coding of the analytical expressions can be time-consuming. Furthermore, for comprehensive rotorcraft analysis codes, the required expressions may not be readily available. Symbolic manipulation methods can automate the derivation and coding of analytical derivatives but these methods can lead to large, cumbersome expressions and inefficient codes. An alternate method is the use of automatic differentiation (AD) methods. Progress has been made in developing a general purpose AD tool known as Automatic Differentiation of FORTRAN (ADIFOR).

ADIFOR is a general purpose AD tool that has been applied to many codes¹⁴⁻²¹. ADIFOR development is funded jointly by Rice University, Argonne National Laboratory, NASA Langley Research Center (LaRC), the Department of Energy, and the National Science Foundation. Rice University and Argonne National Laboratory develop the mathematical foundation for the tool as well as the software tool itself. NASA LaRC provides direction for research and development, provides testing and feedback from the user perspective, and is involved in transferring the tools and techniques to industry.

ADIFOR²¹ transforms an existing computer program into a new program that performs both a sensitivity analysis and the original analysis. If the original FORTRAN program calculates a set of dependent

*Engineer, Senior Member AIAA

†Computer Specialist

‡Manager, Dynamics and Loads

§Senior Principal Engineer

¶Senior Technical Specialist

Copyright©1998 by the American Institute of Aeronautics and Astronautics, Inc. No copyright is asserted in the United States under Title 17, U.S. Code. The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein for Governmental purposes. All other rights are reserved by the copyright owner.

(output) variables from a set of independent (input) variables, then the new FORTRAN program calculates the partial derivatives of the dependent variables with respect to the independent variables. The ADIFOR technique is not an automatic implementation of finite differencing that produces approximate derivatives and is dependent upon proper step-size, nor is it related to symbolic manipulation, which requires reprogramming in a special-purpose language and results in convoluted expressions for the derivatives. Rather, ADIFOR is a systematic implementation of the chain rule of differentiation. ADIFOR produces derivatives to machine accuracy at a cost that is comparable with that of finite-differencing methods.

Reference 20 describes the use of ADIFOR to generate sensitivity derivatives of the comprehensive rotor aerodynamic and dynamic analysis CAMRAD/JA²². The ADIFOR derivatives were compared to finite-difference derivatives for four design variables; no optimization was performed. This successful demonstration of ADIFOR on a comprehensive rotorcraft analysis led to a joint effort with Boeing Helicopter Philadelphia to apply ADIFOR to Boeing's proprietary comprehensive rotorcraft analysis code and then to use these derivatives in several gradient-based optimization schemes.

Since 1995 Boeing Helicopter Philadelphia and the Multidisciplinary Optimization Branch (MDOB) at NASA LaRC have been involved in a joint effort examining issues involved in applying gradient-based optimization methods and efficient sensitivity analysis to Boeing Helicopter's proprietary V11 code. The V11 code is Boeing's TECH-01 code coupled with a gradient-based optimizer.

TECH-01²³ is Boeing's interdisciplinary comprehensive rotorcraft code. It includes such features as elastic blades with nonuniform properties, nonuniform downwash with vortex effects, and nonlinear three-dimensional unsteady aerodynamics. The blade dynamics are represented by up to 50 lumped masses interconnected in series by elastic elements. Essentially, the code computes the response due to the blade's airload. As the blade deforms or responds, the effective angle of attack is changed, which therefore changes the perceived airloads. The process is iterated until convergence is obtained.

The optimizer NPSOLN¹¹⁻¹³ is a Boeing-modified version of the NPSOL optimizer²⁴. It is a set of FORTRAN subroutines designed to minimize a smooth function (i.e., at least twice-continuously differentiable) subject to constraints. The constraints can be simple bounds on the design variables, linear constraints, or

smooth nonlinear constraints. The sensitivity analysis uses finite-difference techniques.

Boeing provided the V11 code and a sample optimization problem to MDOB. The sample problem was a low-vibration rotor optimization problem. MDOB applied the ADIFOR tool to the V11 source code to generate the derivatives of the objective function and constraints. This paper describes this effort and compares these ADIFOR derivatives to finite-difference derivatives. Results using the ADIFOR derivatives and finite-difference derivatives are presented for three optimization schemes - a procedure using a sequential linear programming technique, a procedure using a sequential quadratic programming technique, and a procedure using a sequential unconstrained minimum technique.

Rotor Blade Design Problem

The rotor blade design problem¹¹⁻¹² represented a typical three-bladed, tandem rotor helicopter with an advanced rotor having blade tip sweep. The rotor was discretized into a model consisting of 13 bays, of which the 10 outboard bays had airloads applied. Airloads were applied for two flight conditions. This rather simple model was chosen because it captured the main effects of the vibration problem and still had a rather short function evaluation CPU time of about one minute per airspeed on a SunTM UltraTM 2 workstation*. The goal is to reduce the fixed system vibratory hub loads by tailoring the blade properties. There were 56 design variables representing the level of section mass, stiffness (in flap, chord, and torsion), and chordwise center of gravity position for the different stations along the span of the blade. The only constraint limited the nondimensional total rotor weight to be less than or equal to 1.685.

Results for two objective functions are presented in this paper. The first objective function, which will be referred to as the "Vibration Function", minimized the hub forces and moments. It was a linear combination of the weighted fixed system 3/rev and 6/rev three hub forces (F_x , F_y , F_z) and two hub moments (M_x , M_y) for two flight conditions. The 3/rev loads were weighted as being twice as important as the 6/rev loads and the airspeeds were weighted equally. Optimization results using this objective function have been published before¹¹⁻¹².

The second objective function, which will be referred to as the "Inplane Function", minimized just two of the hub forces. It was a linear combination of the weighted fixed system 3/rev and 6/rev two inplane hub forces (F_x ,

*Sun is a trademark of Sun Microsystems, Inc.

F_y) for two flight conditions. The weighting factors were the same as those used for the first objective function. Results for this objective function have not been published before.

Optimization Scheme

To examine the effect of optimizers on the rotor blade design problem described above three different gradient-based optimization schemes were used — a Sequential Linear Programming (SLP) technique, a Sequential Quadratic Programming (SQP) technique, and a Sequential Unconstrained Minimization Technique (SUMT) were used. The reader is referred to references 25 and 26 for more discussion on these optimization techniques.

Sequential Linear Programming (SLP) Technique

The SLP technique consists of the general purpose optimization program CONMIN²⁷ and an approximate analysis which is used to reduce the number of analyses during the iteration process. The approximate analysis is used to extrapolate the objective function and constraints with linear Taylor Series expansions using derivatives of the objective function and constraints with respect to the design variables during a design iteration

$$F = F_0 + \sum_{i=1}^N \left. \frac{\partial F}{\partial X_i} \right|_0 \Delta X_i \quad (1)$$

$$g = g_0 + \sum_{i=1}^N \left. \frac{\partial g}{\partial X_i} \right|_0 \Delta X_i \quad (2)$$

where F is the objective function, g is the constraint vector, X is the design variable vector, ΔX is the design variable increment, N is the number of design variables, $\left. \frac{\partial F}{\partial X} \right|_0$ is the vector of objective function gradients, and $\left. \frac{\partial g}{\partial X} \right|_0$ is the vector of constraint gradients. A design iteration is defined as an analysis (i.e., evaluation of F_0 and g_0), a sensitivity analysis (i.e., evaluation of gradients), and a line search. The assumption of linearity will not introduce a large error into the analysis provided the changes ΔX are small. Errors which may be introduced by use of the approximate analysis are controlled by imposing “move limits” on each design variable during the iteration process. A move limit which is specified as a fractional change of each design variable value is imposed as an upper and lower design variable bound. At the present time the move limits are manually adjusted.

Sequential Quadratic Programming (SQP) Technique

The SQP (NPSOLN) technique is an algorithm in which the search is broken down into a sequence of quadratic subproblems, each of which uses an approximation to the Hessian.

Sequential Unconstrained Minimization Technique (SUMT)

The SUMT used in this work is the general purpose optimizer KSOPT²⁸. KSOPT uses the Kreisselmeier-Steinhauser (KS)²⁹ function to convert a constrained optimization problem into an unconstrained optimization problem. KSOPT uses the Davidon-Fletcher-Powell (DFP) algorithm to find the unconstrained, one-dimensional search direction and a three-point quadratic polynomial approximation to find the unconstrained minimum.

ADIFOR Processing of the V11 Code

The V11 code has been processed through ADIFOR in a “Black Box” fashion — the independent and dependent quantities are identified and the entire code is processed through ADIFOR. For the low vibration rotor blade optimization problem, the independent quantities are the design variables X , and the dependent quantities are the objective function F and the constraints g . The exact form of the objective function F , the number of design variables X , and the number of constraints g are determined from an input file. Thus the V11 code only needs to be processed once through ADIFOR, regardless of the number of design variables, the terms used in the objective function, and the constraints.

Before processing the V11 code through ADIFOR, a few modifications to the V11 code were necessary since ADIFOR required the code to be ANSI standard. Also the call to the NPSOLN optimizer was temporarily removed. After ADIFOR processing the call to NPSOLN was reinstated. The source code of all math library routines was also included in the analysis code. Although ADIFOR will process even if the source code for the math library routines is not included, the resulting derivative may be incorrect because a portion of the derivative may not be calculated.

The V11 code has 139,376 lines. After processing through ADIFOR, the augmented code (original code plus the sensitivity code) grew to 236,531 lines. In the augmented code, if ADIFOR determines there is a dependency of a quantity on the independent variable, then a new array, augmented by another dimension representing the gradients, is added to the code. These new arrays can increase the memory requirements of the augmented code substantially. The size of that dimension is the number of independent variables being

processed. In the worst case, the memory requirement of the augmented code is the original memory times the number of independent variables. When the V11 code was processed through ADIFOR with 56 design variables, it was found that the memory required to execute the augmented code exceeded the memory available. Therefore, the V11 code was processed through ADIFOR using 7 independent variables. The augmented code was then modified to evaluate the ADIFOR derivatives using 7 design variables at a time until all 56 design variables were used. Thus during a design iteration using ADIFOR derivatives, 8 extra function evaluations are performed since the gradients had to be generated in blocks of 7 design variables at a time. In the ADIFOR-generated code, the function evaluation occurs along with the derivative evaluations. If a computer with enough memory were available, these extra function evaluations would not be necessary. When large codes are being used, this expansion in array sizes has an effect on computer resources that must be considered.

Observations on Incorporating ADIFOR Derivatives With Different Optimization Schemes

Processing the V11 code through ADIFOR was fairly simple once the V11 code was changed to satisfy the ANSI standard. Once the code was ANSI standard, it took about two days to pre- and post-process the code through ADIFOR to obtain the augmented code.

Implementing the ADIFOR derivatives in the SLP and SUMT schemes was easier than in the SQP scheme. The optimizers used in the SLP and SUMT schemes (CONMIN and KSOPT, respectively) have a flag indicating when gradients need to be calculated. Therefore the user has control over which version of the V11 code to use — the original code when objective function and constraints are required and the ADIFOR version of the code when gradients are required. The optimizer used in the SQP scheme (NPSOLN) does not have such a flag. The user has no control over which version of the V11 code to use. Thus, NPSOLN used the ADIFOR version of the code even when only the objective function and constraints evaluations were required, resulting in a substantial computational penalty. To use ADIFOR derivatives with the NPSOLN optimizer, Boeing rewrote the NPSOLN subroutine which calculates the gradients so that the original code is used for function evaluations and the ADIFOR code is used for gradient evaluations.

Results

Results for the Vibration Function are presented first, followed by results for the Inplane Function. For each objective function, the ADIFOR derivatives are compared with finite-difference derivatives and then

optimization results are presented. All optimization results start from the initial design provided by Boeing which is a good preliminary rotor blade design. The design variables are normalized so that they are -1.0 at a lower bound and $+1.0$ at an upper bound. The constraint and objective function values are normalized by these initial design values.

Five labels will be used in the following sections to identify the results. CONMIN refers to the SLP scheme in which the optimizer is CONMIN and includes the approximate analysis described by equations (1) and (2). NPSOLN refers to the SQP scheme in which the optimizer is NPSOLN. KSOPT refers to the SUMT scheme in which the optimizer is KSOPT. A suffix of “-AD” will be used to indicate the use of ADIFOR derivatives and a suffix of “-FD” will be used to indicate the use of finite-difference derivatives. Note that all results shown are from the analysis and not from approximations.

Vibration Function Timing Comparison

One result from this study is the rather large difference in how much CPU time was needed to perform one design iteration with ADIFOR derivatives (366 minutes) versus finite-difference derivatives (209 minutes). Recall that a design iteration consists of analysis, derivatives calculation, and line search. Both timings are for the Vibration Function using the NPSOLN optimizer on a Sun™ Ultra™ 2 workstation with 128 megabytes of memory and 877 megabytes of swap space. Note that the finite-difference design iteration does not include timings related to any step-size study and the ADIFOR design iteration includes 8 extra function evaluations as previously described. The timing ratio for a design iteration using the ADIFOR derivatives compared to a design iteration using the finite-difference derivatives in the V11 code is 1.75. This ratio is similar to timing ratios from other ADIFOR applications^{15,20}. For example in reference 15, the timing ratio for the ADIFOR derivatives compared to finite-difference derivatives ranges from 1 to 2.

Vibration Function Derivative Comparison

The ADIFOR and finite-difference derivatives are compared at two different design points. The first comparison is at the initial design and the second comparison is at the final design generated by CONMIN-AD. The finite-difference derivatives are generated with four step-sizes (0.001, 0.0001, 0.00001, 0.000001).

Table 1 compares the objective function gradients for the initial design at selected design variables which are representative of all the gradients. The table shows excellent agreement between objective function derivatives generated by ADIFOR and finite-difference

derivatives generated using Boeing's default finite-difference step-size (0.00001), except when the derivative values are small (e.g., design variables 3 and 5). The discrepancy in the derivatives can be attributed to finite-difference step-size choice. For example for design variables 26, 53, and 56, the finite-difference derivative is dependent on step-size choice. The table also shows that, for finite-difference derivatives, a step-size that is good for one design variable might not be the best for another design variable. For example, a step-size of 0.001 is not good for design variables 21 and 26 but is good for most of the other design variables.

Table 2 compares the objective function gradients at a different point in design space — the CONMIN-AD optimized design. For this design, the finite-difference derivatives are very step-size dependent compared to the initial design (Table 1), indicating that the step-size must be adjusted during the course of optimization if finite-difference derivatives were used. For example, for design variables 2, 4, 7, and 35, a step-size of 0.001 is a better choice than the default step-size of 0.00001 for the finite-difference derivatives. This is not a concern with the ADIFOR derivatives since they are not dependent on step-size.

Vibration Function Optimization Results

The ADIFOR derivatives are used in three optimization procedures and the finite-difference derivatives are used in two optimization procedures. All procedures start from the same initial design. Figure 1 shows the Vibration Function optimization history. The CONMIN-AD, KSOPT-AD, and NPSOLN-AD results are shown in Figure 1a. The KSOPT-FD and NPSOLN-FD results are shown in Figure 1b. Table 3 compares the optimization results for all the optimization schemes. The optimization procedures found different minima and required different numbers of design iterations to converge. As indicated in Figure 1a, the CONMIN-AD result had the lowest objective function value but it required the most design iterations to converge. The nonlinearity of the objective function required the use of small move limits on the design variables during a design iteration. These move limits were reduced manually during the optimization when the analysis and the linear approximation differed significantly at the end of the design iteration. For example, in Figure 1a at design iterations 10 and 21, the increase in the objective function is due to a poor match between analysis and linear approximation.

KSOPT-AD found a lower minimum and required fewer design iterations than KSOPT-FD. However, the NPSOLN results were reversed. NPSOLN-FD found a lower minimum and required fewer design iterations than NPSOLN-AD. This result is consistent with the

fact that differences in the gradient values lead to different search paths which in turn can lead to different local optima. The NPSOLN-FD code failed on a restart because the approximate Hessian was not restored and the algorithm pushed the design into an infeasible region, where the TECH-01 iterations failed to converge.

A closer examination of the designs in Table 3 shows that many of the design variable values are the same for the NPSOLN-AD, KSOPT-FD, and KSOPT-AD designs. The NPSOLN-FD and CONMIN-AD designs are different than the other three and are also different from each other. The NPSOLN-FD optimization design is the only design which has the first design variable reaching its upper bound.

Vibration Function Alpha Test

Figure 1 indicates that various minima were found. A test denoted the "alpha test" was performed to determine whether the optimizers correctly identified several local minima or whether the optimizer stopped prematurely. For example, why did the KSOPT-FD optimizer not find the lower result found by the KSOPT-AD method? (Note that this alpha test was originally suggested by R. T. Haftka and has been used in various studies³⁰⁻³¹). With the alpha test, the objective function is evaluated at regular intervals between two solution points (X^1 and X^2). Each intermediate point X is determined by varying a parameter called α from 0 to 1 in the following expression:

$$X = \alpha X^1 + (1 - \alpha) X^2 \quad (3)$$

The objective function is plotted as a function of α . If a smooth and monotonically decreasing curve results, then one must look at the analysis to determine why the optimizer stopped at X^1 and not X^2 . If the curve is irregular, it usually indicates that the objective function has many local minima.

As shown in Figure 1, KSOPT-AD found a lower minimum than the KSOPT-FD. Figure 2 shows the alpha test results between the KSOPT-FD result X^1 ($F=0.70461$, $\alpha=0.0$) and the KSOPT-AD result X^2 ($F=0.64087$, $\alpha=1.0$). The objective function is very nonlinear between these two designs. There appears to be a local minimum around the finite-difference result X^1 so that the optimizer stopped there instead of finding the lower value found using ADIFOR derivatives.

As shown in Figure 1, CONMIN-AD ($F=0.56481$) found a lower minimum than KSOPT-AD ($F=0.64087$). Another alpha test was performed between these 2 designs. Figure 3 suggests that the objective function

is highly nonlinear and does experience various local minima.

Inplane Function Derivative Comparison

The ADIFOR and finite-difference derivatives are compared at the initial design for the Inplane Function. The finite-difference derivatives are generated with four step-sizes (0.001, 0.0001, 0.00001, 0.000001).

Table 4 compares the objective function gradients for the initial design at selected design variables. The table shows excellent agreement between the objective function derivatives generated by ADIFOR and finite-difference derivatives generated by Boeing's default finite-difference step-size (0.00001) except for design variables 5 and 10. A step-size of 0.0001 appears good for all design variables. A step-size of 0.001 is not good for design variables 21 and 26.

Inplane Function Optimization Results

The ADIFOR derivatives are used in three optimization procedures and finite-difference derivatives are used in two optimization procedures. All procedures start from the same initial design. Figure 4 shows the optimization history for all the schemes as a function of design iteration. The CONMIN-AD, KSOPT-AD, and NPSOLN-AD results are shown in Figure 4a. The KSOPT-FD and NPSOLN-FD results are shown in Figure 4b. For the Inplane Function, neither NPSOLN-FD nor NPSOLN-AD was able to reduce the objective function as well as KSOPT-AD, KSOPT-FD, and CONMIN-AD. Table 5 compares the optimization results for all the optimization schemes. NPSOLN-FD and NPSOLN-AD could only reduce the objective function by about 5 percent. KSOPT-AD, KSOPT-FD, and CONMIN-AD were able to reduce the objective function by about 60 percent. A closer examination of the designs in Table 5 shows that many of the design variable values are the same for the NPSOLN-AD, KSOPT-FD, and KSOPT-AD designs. We are still trying to understand why the NPSOLN optimizer stopped so soon.

Concluding Remarks

Since 1995 Boeing Helicopter Philadelphia and the Multidisciplinary Optimization Branch at NASA Langley Research Center have been involved in a joint effort examining issues involved in applying gradient-based optimization methods and efficient sensitivity analysis to Boeing Helicopter's proprietary V11 code. The general purpose automatic differentiation tool called Automatic Differentiation of FORTRAN (ADIFOR) was used to obtain sensitivity derivatives. Three optimization schemes using general-purpose optimizers (CONMIN, NPSOLN, and KSOPT) were used to solve

a low vibration rotor design problem with 56 design variables representing the level of section mass, stiffness (in flap, chord, and torsion), and chordwise center of gravity position for the different stations along the span of the blade. Two objective functions were used. The first objective function (denoted Vibration Function) was a linear combination of hub loads and forces. The second objective function (denoted Inplane Function) was a linear combination of inplane hub loads. The only constraint was an upper and lower bound on blade weight.

It was fairly simple to process the V11 code through ADIFOR once the code was changed to satisfy the ANSI standard. The ADIFOR derivative code required a large amount of memory to execute. A design iteration (analysis, sensitivity analysis, and line search) using the ADIFOR derivatives took about 75 percent longer to compute than a design iteration using the finite-difference derivatives which is consistent with previous ADIFOR applications. The finite-difference derivative time used for comparison was for a given step-size. The ADIFOR derivatives were compared with finite-difference derivatives at several step-sizes for several designs. It was shown that the finite-difference derivatives were very step-size dependent while the ADIFOR derivatives were exact. The cost of determining the proper step-size should be considered when choosing between ADIFOR and finite-difference derivatives.

Both ADIFOR derivatives and finite-difference derivatives were used in several optimization schemes for both objective functions. For the Vibration Function, the different optimization schemes identified several different local minima. For the Inplane Function, it was found that choice of optimizer was very important. NPSOLN using either ADIFOR derivatives or finite-difference derivatives was not able to reduce the objective function by more than 5 percent. Both CONMIN and KSOPT using either ADIFOR derivatives or finite-difference derivatives were able to reduce the objective function by about 60 percent.

References

1. Bennett, R. L., "Application of Optimization Methods to Rotor Design Problems," *Vertica*, Vol. 7, No. 3, 1983, pp. 201-208.
2. Walsh, J. L., Bingham, G. J., and Riley, M. F., "Optimization Methods Applied to the Aerodynamic Design of Helicopter Rotor Blades," *Journal of the American Helicopter Society*, Vol. 32(4), Oct. 1987, pp. 39-44.
3. Callahan, C. B., and Straub, F. K., "Design Optimization of Rotor Blades for Improved Performance

- and Vibrations,” *Proceedings of the 47th Annual Forum of the American Helicopter Society*, Phoenix, AZ, May, 1991, pp.869–882.
4. Chattopadhyay, A., Walsh J. L., and Riley, M. F., “Integrated Aerodynamic Load/Dynamic Optimization of Helicopter Rotor Blades,” *Journal of Aircraft*, Vol. 28(1), Jan. 1991, pp. 58–65.
 5. Walsh, J. L., LaMarsh II, W. J., and Adelman, H. M., “Fully Integrated Aerodynamic/Dynamic Optimization of Helicopter Rotor Blade,” *Mathematical and Computer Modelling*, Vol. 18(3/4), Aug. 1993, pp. 37–52.
 6. Leconte, P., and Geoffroy, P., “Dynamics Optimization of a Rotor Blade,” *Presented at AHS Aeromechanics Specialists Conference*, San Francisco, CA., Jan. 1994, pp. 5.2-1–5.2-24.
 7. Yuan, K., and Friedmann, P. P., “Structural Optimization of Composite Helicopter Rotor Blades with Composite Helicopter Rotor Blades with Swept Tips for Vibration Reduction in Forward Flight,” *5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA Paper 94-4282, Panama City, FL, Sept. 1994, pp. 281–302.
 8. McCarthy, T. R., Chattopadhyay, A., Zhang, S. “A Coupled Rotor/Wing Optimization Procedure for High Speed Tilt-Rotor Aircraft,” *Proceedings of the 51th Annual Forum of the American Helicopter Society*, Fort Worth, TX, May 1995, pp. 924–936.
 9. Booker, A. J., Frank, P. D., Conn, A. R., Dennis, J. E., Serafini, D., Torczon, V., and Trosset, M., “Multi_Level Design Optimization A Boeing/IBM/Rice Collaborative Project 1996 Final Report,” Technical Report ISSTECH-96-031, Boeing Information & Support Services, December 1996.
 10. Lim, J., and Chopra, I., “Design Sensitivity Analysis for an Aeroelastic Optimization of a Helicopter Blade,” *AIAA Dynamics Specialist Conference*, AIAA Paper 87-0923-CP. Monterey, CA, Apr. 1987, pp. 1093–1102.
 11. Hirsh, J. E., and Young, D. K., “Evolutionary Programming Strategies with Self-Adaptation Applied to the Design of Rotorcraft using Parallel Processing,” *Proceedings of the Seventh Annual Conference on Evolutionary Programming*, Springer-Verlag, San Diego, CA, Mar. 1998.
 12. Tarzanin, F. J., and Young, D. K., “Boeing Rotorcraft Experience with Rotor Design and Optimization,” *Proceedings of the Seventh AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA Paper 98-4733, St. Louis, MO, Sept. 1998.
 13. Young, D. K., and Tarzanin, F. J., “Structural Optimization and Mach Scale Test Validation of a Low Vibration Rotor,” *Journal of American Helicopter Society*, Vol. 38(3), July 1993.
 14. Barthelemy, J. F., and Hall, L. E., “Automatic Differentiation as A Tool in Engineering Design,” *Structural Optimization*, Vol. 9, 1995, pp. 76–82.
 15. Carle, A., Green, L. L., Bischof, C. H., and Newman, P. A., “Application of Automatic Differentiation in CFD,” *25th AIAA Fluid Dynamics Conference*, AIAA Paper 94-2197, Colorado Springs, CO, June 1994.
 16. Bischof, C. H., Green, L. L., Haigler, K. J., and Knauff, Jr., T. L., “Parallel Calculation of Sensitivity Derivatives for Aircraft Design Using Automatic Differentiation,” *5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA Paper 94-4282, Panama City, FL, Sept. 1994, pp. 73–86.
 17. Bischof, C. H., Pusch, G. D., and Knoesel, R., “Sensitivity Analysis of the MM5 Weather Model Using Automatic Differentiation,” Preprint MCS-P532-0895, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, Dec. 1995,
 18. Moen, C. D., Spence, P. A., Mesa, J. C., and Plantenga, T. D., “Automatic Differentiation for Gradient-Based Optimization of Radiatively Heated Microelectronics Manufacturing Equipment,” *6th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA Paper 96-4118, Bellevue, WA, 1996, pp. 1167–1175.
 19. Unger, E. R., and Hall, L. E., “The Use of Automatic Differentiation in an Aircraft Design Problem,” *5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA Paper 94-4260, Panama City, FL, 1994, pp. 64–72.
 20. Walsh, J. L., and Young, K. C., “Automatic Differentiation Evaluated as a Tool for Rotorcraft Design and Optimization,” *Proceedings of the AHS National Technical Specialist Meeting Rotorcraft Structures: Design Challenges and Innovative Solutions*, Williamsburg, VA, Oct. 30-Nov. 2, 1995.
 21. Bischof, C. H., and Griewank, A., “ADIFOR, A Fortran System for Portable Automatic Differentiation,” *Fourth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization*, AIAA-94-4282-CP, Cleveland, OH, Sept. 1992, pp. 433–441.
 22. Johnson, W., *CAMRAD/JA - A Comprehensive Analytical Model of Rotorcraft Aerodynamics and Dynamics — Johnson Aeronautics Version*, Vol. I: Theory Manual and Vol. II: User’s Manual, Johnson Aeronautics, 1988.

23. Shultz, L. A., Panda, B., Tarzanin, F. J., Derham, R. C., Oh, B. K., and Dadone, L., "Interdisciplinary Analysis for Advanced Rotors - Approach, Capabilities and Status," *Presented at AHS Aeromechanics Specialists Conference*, San Francisco, CA., Jan. 1994, pp. PS.4-1-PS.4-15.
24. Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H., "User's Guide for NPSOL (Version 4.0): A FORTRAN Package for Nonlinear Programming," SOL 86-2, Jan. 1986.
25. Vanderplaats, G. N., *Numerical Optimization Techniques for Engineering Design: With Applications*, McGraw-Hill, Inc., New York, 1984.
26. Haftka, R. T., Gurdal, A., Kamat, M. P., *Elements of Structural Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1990.
27. Vanderplaats, G. N., "CONMIN — A FORTRAN Program for Constrained Function Minimization, User's Manual," NASA TMX-62282, Aug. 1973.
28. Wrenn, G. A., "An Indirect Method for Numerical Optimization Using the Kreisselmeier-Steinhauser Function," NASA CR-4220, Mar. 1989.
29. Kreisselmeier, G., and Steinhauser, R., "Systematic Control Design by Optimizing a Vector Performance Index," *International Federation of Active Controls Symposium on Computer-Aided Design of Control Systems*, Zurich, Switzerland, Aug. 1979.
30. Henderson, J. L., Walsh, J. L., and Young, K. C., "Application of Response Surface Techniques to a Helicopter Rotor Blade Optimization Procedure," *Proceedings of the AHS National Technical Specialist Meeting Rotorcraft Structures: Design Challenges and Innovative Solutions*, Williamsburg, VA, Oct. 30-Nov. 2, 1995.
31. Balabanov, V. O., "Development of Approximations for HSCT Wing Bending Material Weight Using Response Surface Methodology," Ph.D. Dissertation, Virginia Polytechnic Institute and State University, Blacksburg, VA, Sept. 1997.

Table 1. Comparison of objective function gradients for initial design — Vibration Function

DV No.	Scaled X	Scaled Objective Function Derivatives				
		Finite difference (step-size)				ADIFOR
		(0.001)	(0.0001)	(0.00001)*	(0.000001)	
1	-1.000000	0.267894	0.264466	0.264996	0.265380	0.265042
2	-0.299511	0.009128	0.009133	0.009194	0.009337	0.009129
3	-0.547333	0.000054	0.000064	0.000061	0.000862	0.000054
4	-1.000000	-0.000302	-0.000301	-0.000260	-0.000176	-0.000302
5	-0.842483	0.000001	0.000006	0.000023	0.000078	0.000001
7	0.017657	0.012525	0.012408	0.011635	-0.007819	0.012547
10	-0.709089	0.000012	0.000012	0.000023	0.000701	0.000012
21	-0.826086	0.030332	0.111996	0.114020	0.114669	0.114294
26	-1.000000	-0.066428	0.004358	0.006627	0.007338	0.006920
35	-0.860322	0.028158	0.028137	0.028161	0.028359	0.028134
53	-1.000000	-0.024887	-0.047656	-0.048948	-0.047871	-0.047964
56	-0.322034	-0.000459	-0.000455	-0.000374	0.000396	-0.000459

* Step-size used in finite-difference optimization

Table 2. Comparison of objective function gradients for CONMIN-AD optimized design - Vibration Function

DV No.	Scaled X	Scaled Objective Function Derivatives				
		Finite difference (step-size)				ADIFOR
		(0.001)	(0.0001)	(0.00001)*	(0.000001)	
1	-0.988316	0.020598	0.022351	0.022270	0.018428	0.022890
2	-0.876564	0.001590	0.001559	0.000922	-0.002849	0.001596
3	-0.703650	0.000083	0.000058	-0.000408	-0.006652	0.000093
4	-0.255481	0.000024	-0.000186	-0.000862	-0.012195	0.000033
5	-0.963706	-0.000002	-0.000015	-0.000638	-0.002012	0.000000
7	0.005231	0.002265	0.011723	0.127495	0.123511	0.002292
10	-1.000000	-0.000003	-0.000046	-0.000101	0.000015	0.000002
21	-0.399443	0.035501	0.036102	0.035369	0.035574	0.036175
26	-0.999300	-0.065723	-0.056733	-0.056674	-0.057738	-0.056652
35	-0.959707	0.006113	0.006115	0.005736	0.003701	0.006195
53	-1.000000	0.214883	0.201452	0.202390	0.196713	0.203081
56	-0.931666	-0.001570	-0.001625	-0.001861	-0.003850	-0.001557

* Step-size used in finite-difference optimization

Table 3. Comparison of optimization results — Vibration Function.

	NPSOLN (SQP)		CONMIN and Approx. Analysis (SLP)	KSOPT (SUMT)	
	FD	ADIFOR	ADIFOR	FD	ADIFOR
Scaled F	0.65361	0.70566	0.56481	0.70461	0.64087
Scaled g	1.685	1.223	1.324	1.198	1.291
No. of design iterations	7	11	68	20	11
DV No.	Scaled X				
1	1.00000	-1.00000	-0.98832	-1.00000	-0.99819
2	-0.56035	-0.31320	-0.87656	-0.31344	-0.31634
3	-0.74979	-0.54748	-0.70365	-0.54746	-0.54780
4	-1.00000	-1.00000	-0.25548	-0.99954	-0.99988
5	-1.00000	-0.84248	-0.96371	-0.84249	-0.84248
7	0.89379	-0.00062	0.00523	0.00022	-0.00475
10	-0.75603	-0.70909	-1.00000	-0.70910	-0.70909
21	-1.00000	-0.40687	-0.39944	-0.58638	-0.29573
26	0.33150	-1.00000	-0.99930	-1.00000	-0.99636
35	-0.89695	-0.90080	-0.95971	-0.90521	-0.89245
53	-0.59309	-1.00000	-1.00000	-1.00000	-1.00000
54	-1.00000	-0.21116	-0.41148	-0.34910	-0.34987
56	-0.33765	-0.34196	-0.93167	-0.33939	-0.36348

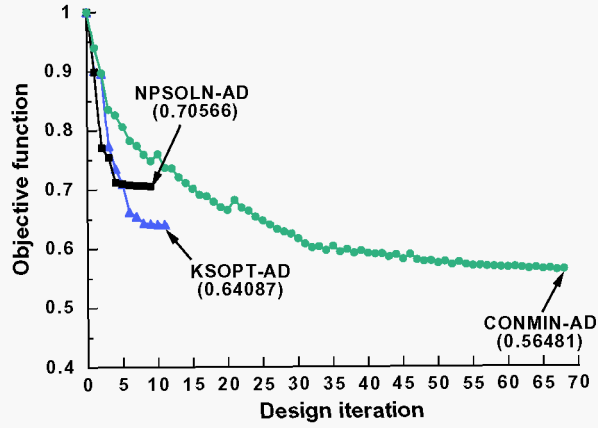
Table 4. Comparison of objective function gradients for initial design — Inplane Function

DV No.	Scaled X	Scaled Objective Function Derivatives				
		Finite difference (step-size)				ADIFOR
		(0.001)	(0.0001)	(0.00001)*	(0.000001)	
1	-1.000000	-0.367694	-0.367301	-0.367549	-0.367673	-0.367539
2	-0.299511	-0.013693	-0.013700	-0.013758	-0.013960	-0.013693
3	-0.547333	-0.000460	-0.000462	-0.000480	-0.000873	-0.000460
4	-1.000000	0.000184	0.000182	0.000171	0.000030	0.000184
5	-0.842483	0.000006	0.000004	-0.000026	-0.000303	0.000007
7	0.017657	-0.019038	-0.018888	-0.017493	-0.007448	-0.019045
10	-0.709089	0.000063	0.000060	0.000036	-0.000397	0.000064
21	-0.826086	0.334576	0.553610	0.552819	0.552591	0.552730
26	-1.000000	-0.543378	-0.350494	-0.351098	-0.351282	-0.351208
35	-0.860322	0.064729	0.064677	0.064644	0.064482	0.064672
53	-1.000000	-0.602101	-0.570575	-0.569799	-0.570418	-0.570271
54	-0.322502	-0.041768	-0.042607	-0.042702	-0.043283	-0.042644
56	-0.322034	-0.006766	-0.006770	-0.006830	-0.007600	-0.006765

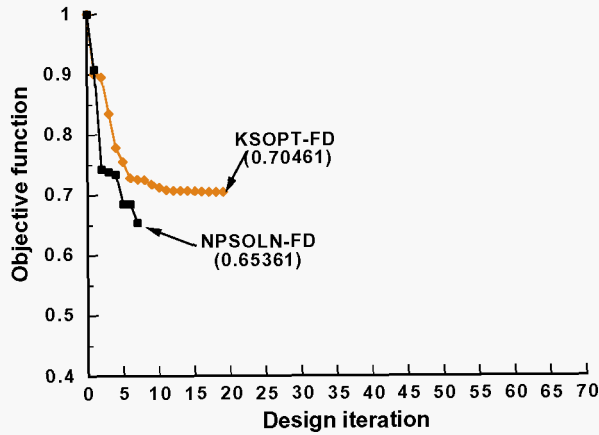
* Step-size used in finite-difference optimization

Table 5. Comparison of optimization results — Inplane Function.

	NPSOLN (SQP)		CONMIN and Approx. Analysis (SLP)	KSOPT (SUMT)	
	FD	ADIFOR	ADIFOR	FD	ADIFOR
Scaled F	0.945997	0.945997	0.411626	0.392064	0.40469
Scaled g	1.685	1.468	1.156	1.087	1.124
No. of design iterations	1	2	25	13	14
DV No.	Scaled X				
1	-0.65358	-1.00000	-0.93437	-0.92768	-0.93627
2	-0.28583	-0.28582	-0.27985	-0.29650	-0.29744
3	-0.54688	-0.54687	-0.48491	-0.54726	-0.54727
4	-1.00000	-1.00000	-1.00000	-1.00000	-1.00000
5	-0.84249	-0.84249	-0.87262	-0.84248	-0.84248
7	0.03671	0.03670	0.01912	0.02160	0.02049
10	-0.70915	-0.70915	-0.78193	-0.70910	-0.70910
21	-1.00000	-1.00000	-0.92234	-0.93769	-0.88858
26	-0.85393	-1.00000	-0.99408	-0.99790	-0.99991
35	-0.92500	-0.92499	-0.90643	-0.86399	-0.86650
53	-1.00000	-1.00000	-0.89057	-0.97352	-0.96392
56	-0.31528	-0.31527	-0.28337	-0.31929	-0.32081



a) ADIFOR derivatives.



b) Finite-difference derivatives.

Figure 1. Vibration Function optimization history.

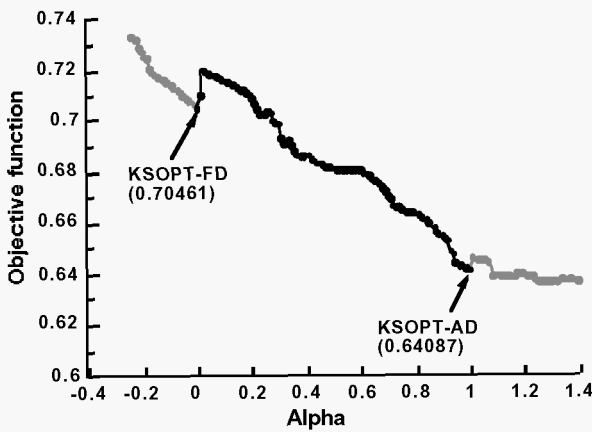


Figure 2. Alpha test for KSOPT design — Vibration Function.

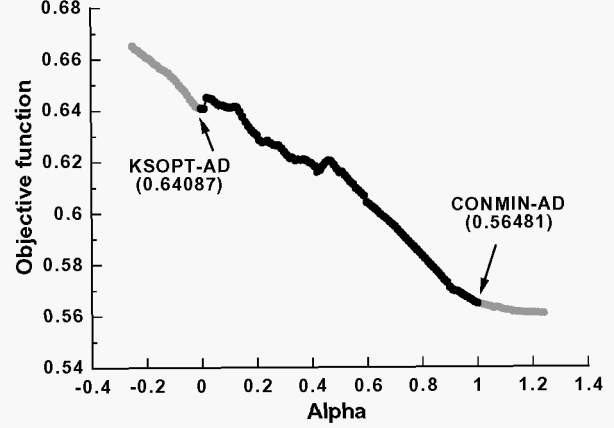
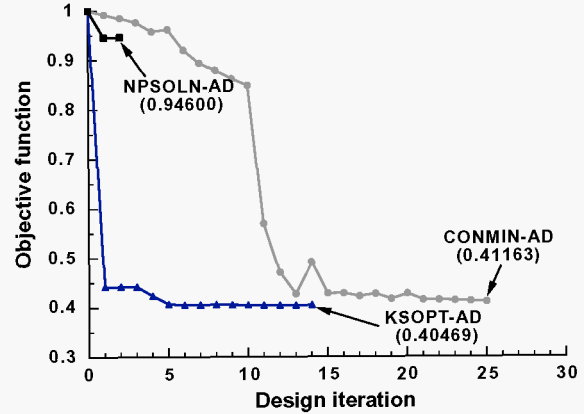
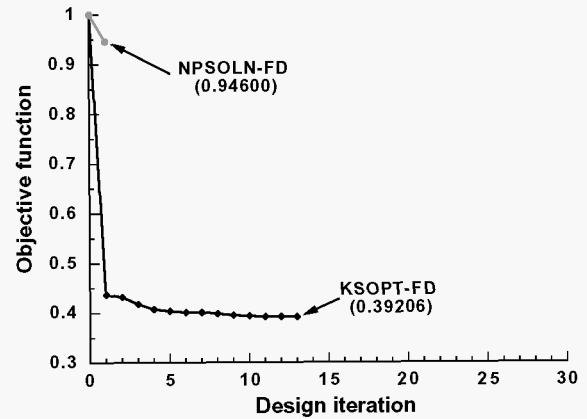


Figure 3. Alpha test for KSOPT-AD and CONMIN-AD design — Vibration Function.



a) ADIFOR derivatives.



b) Finite-difference derivatives.

Figure 4. Inplane Function optimization history.