

**PARALLEL THERMOELASTIC OPTIMIZATION OF
3-D SERPENTINE COOLING PASSAGES IN TURBINE BLADES**

Brian H. Dennis¹

Frontier Simulation Software for
Industrial Science Collaborative Research
Center
Institute of Industrial Science
University of Tokyo
4-6-1 Komaba, Muguro-ku, Tokyo 153-8505,
JAPAN

dennis@garlic.q.t.u-tokyo.ac.jp

Igor N. Egorov²

IOSO Technology Center
Milashenkova 10-201
Moscow 127322
RUSSIA
optim@orc.ru

Helmut Sobieczky³

Deutsche Luft-und-Raumfahrt
Bunsenstrasse 10
D-37073 Goettingen
GERMANY
Helmut.Sobieczky@dlr.de

George S. Dulikravich⁵

Dept. of Mechanical and Aerospace Eng.
Multidisciplinary Analysis, Inverse Design &
Optimization institute, UTA Box 19018
The University of Texas at Arlington, Arlington,
Texas 76019, U.S.A.
dulikra@mae.uta.edu

Shinobu Yoshimura⁶

Institute of Environmental Studies
Graduate School of Frontier Sciences
University of Tokyo
7-3-1 Hongo, Bunkyo-ku,
Tokyo 113-8656, JAPAN
yoshi@q.t.u-tokyo.ac.jp

ABSTRACT

An automatic design algorithm for parametric shape optimization of three-dimensional cooling passages inside axial gas turbine blades has been developed. Serpentine passage configurations are considered. The geometry of the blade and the internal serpentine cooling passages were parameterized using Sobieczky's surface patch analytic formulation, which provides very high degree of flexibility, second order smoothness and a minimum number of parameters. The design variable set defines the geometry of the turbine blade coolant passage including blade wall thickness distribution and blade internal strut configurations. A parallel three-dimensional thermoelasticity finite element analysis (FEA) code from the ADVENTURE project at the University of Tokyo was used to perform automatic thermal and stress analysis of different blade configurations. The same code can also analyze nonlinear (large/plastic deformation) thermoelasticity problems for complex 3-D configurations. Convective boundary conditions were used for the heat conduction analysis to approximate the presence of internal and external fluid flow. The objective of the optimization was to make stresses throughout the blade as uniform as possible. Constraints were that the maximum temperature and stress at any point in the blade were less than the maximum allowable values. A robust semi-stochastic constrained optimizer and a parallel genetic algorithm were used to solve this problem while running on an inexpensive distributed memory parallel computer.

INTRODUCTION

With the continuing growth of computing resources available, the attention of design engineers has been rapidly shifting from the use of repetitive computational analysis, personal experience, and intuition towards a reliable and economical mathematically based optimization algorithms. Such algorithms have the potential to produce improved designs over a shorter period of time. In this paper, the application of optimization to the design of passages for internally cooled 3-D realistic turbine blades is presented.

Internal cooling schemes of modern turbojet and turbofan engines bleed air from the compressor and pass this air into the serpentine coolant flow passages within the turbine blades. The maximum temperature within a turbine blade must be kept below a certain value in order to maintain blade life limited by creep, oxidation, corrosion, and fatigue. To achieve turbine blade durability requirements, section-averaged centrifugal stress limitations should be satisfied, concentrations of thermal stress should be limited in the cold areas to reduce low cycle fatigue, principal strains should be held below a given level in hot areas to reduce thermo-mechanical fatigue, and the maximum temperature in the blade metal and coating material must be below specified limits because of oxidation, corrosion and coating spallation concerns. These objectives can be obtained by the constrained optimization of the coolant passage shapes inside the turbine blade at a fixed level of coolant flow rate.

There is a strong interaction among a number of engineering disciplines when studying the design internally cooled gas turbine blades [1,2,3]. The temperature and the associating stresses within the blade material is considered in detail. However, the effects of the hot gas flow and coolant flow will be treated in very approximate way. In the design process explained in this paper, these individual disciplines will not be solved simultaneously in detail for 3-D designs, because this approach would take an unacceptably long time, even on a cluster of workstations running in parallel. For these pragmatic reasons a more approximate yet computationally affordable design approach was adopted.

The design method is based on a combination of several algorithmic components. These components include optimization, finite element analysis, and shape design parameterization. The optimization and finite element analysis modules can be considered “black boxes” and can be applied directly to any specific passage design situation. Shape design parameterization on the other hand is considered problem-specific and different codes need to be developed for different design problems.

In order to complete the design process in a reasonable amount of time, a parallel computer should be employed. Both the finite element analysis and the optimization codes used here were written to make full use of parallel computing resources.

OPTIMIZATION METHOD

The core of the passage design system is the optimization code. The optimizer directs the design process by generating new designs based on the performance of previous designs, in an iterative manner. In general, one wishes to use optimization methods that are robust and efficient. For optimization on a parallel computer, the optimizer should find a good design in the minimum possible number of iterations. Such algorithms should also be capable of making full use of large-scale parallel computers. Since each design analysis is a full 3-D simulation, the total computation time can be from weeks to months if an efficient and sufficiently parallel algorithm is not used.

Robust optimization algorithm are also desired. The optimization process should not terminate in a local minimum and it should not terminate if the analysis cannot be completed occasionally due to, for example, failure to generate a proper grid for a candidate design. For the particular problem considered in this paper, the method should not require gradients of the objective or constraints so that discontinuous objective functions and discrete design variables can be used (for example, total number of cooling passages which could vary during the optimization).

In our experience, parallel genetic algorithm (PGA) variations [9,10,11,12,] and response surface methods based on Indirect Optimization based on Self Organization (IOSO) [13,14] work well for 3-D turbine coolant passage design optimization.

The core of the design method is the optimization module. The optimizer directs the design process by generating new designs based on the performance of previous designs, in an iterative manner. In general, optimization methods that are robust and efficient are the most desirable. For optimization on a parallel computer, the optimizer should find a good design in the minimum possible number of iterations. Such algorithms should also be capable of making full use of large-scale parallel computers. Since each design analysis is a full 3-D simulation, the total computation time can be from weeks to months if an efficient and sufficiently parallel algorithm is not used. An optimization method for turbine design should also be robust. The optimizer should not become trapped in a local minimum due to a noise generated by the analysis code. It should also not terminate if the analysis cannot be completed due to, for example, failure to generate a proper grid for a candidate design. In our experience, genetic algorithm (GA) variations [3,4,5,6,7] and response surface methods based on Indirect Optimization based on Self Organization (IOSO) [8,9] work well for 3-D turbine coolant passage design optimization.

IOSO Method

The IOSO method is a constrained optimization algorithm based on response surface methods and evolutionary simulation principles. Each iteration of IOSO consists of two steps. The first step is creation of an approximation of the objective function(s). Each iteration in this step represents a decomposition of an initial approximation function into a set of simple approximation functions. The final response function is a multilevel graph such as the one shown in Figure 1. The second step is the optimization of this

approximation function. This approach allows for self-corrections of the structure and the parameters of the response surface approximation. The distinctive feature of this approach is an extremely low number of trial points to initialize the algorithm (30-50 points for the optimization problems with nearly 100 design variables).

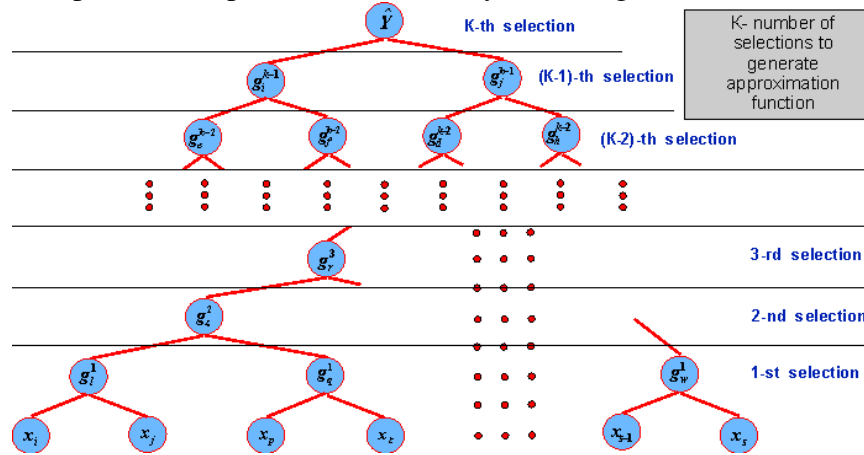


Figure 1: Multilevel approximation function

The obtained response functions are used in the procedures of multilevel optimization with the adaptive changing of the simulation level within the frameworks of both single and multiple disciplines of the object analysis. During each iteration of the IOSO, the optimization of the response function is carried out within the current search area. This step is followed by the direct call to the mathematical model for the obtained point. The information concerning the behavior of the objective function nearby the extremum is stored, and the response function is made more accurate just for this search area. For a basic parallel IOSO algorithm, the following steps are carried out:

1. Generate a group of designs based on a design of experiments (DOE) method;
2. Evaluate the designs in parallel with the analysis code;
3. Build initial approximation based on the group of evaluated designs;
4. Use stochastic optimization method to find the minimum of the approximation;
5. Do adaptive selection of current extremum search area;
6. Generate a new set of designs in current extremum search area using DOE;
7. Evaluate the new set of designs in parallel with the analysis code;
8. Update the approximation with newly obtained result;
9. Goto 4. unless termination criteria is met;

Thus, during each iteration, a series of approximation functions is built for a particular optimization criterion. These functions differ from each other according to both structure and definition range. The subsequent optimization of the given approximation functions allows us to determine a set of vectors of optimized variables, which are used to develop further optimization criteria on a parallel computer.

Multilevel Parallelism in Optimization

The usual approach to parallel optimization is to run a single analysis on each processor per optimization iteration. However, a mesh for a geometrically complex design may be large; sometimes the finite element analysis requires more memory than is available on a single processor. For this reason, the finite element analysis must be distributed among

several processors. If a large number of processors are available, the optimizer can use all of them by running several simultaneous parallel analyses to evaluate several candidate design configurations. For this research an optimization communication module was developed using the MPI library [10] that utilizes this multilevel hierarchy of parallelism. This module can be used with any parallel optimization method including GA and IOSO algorithms. A graphical depiction of the hierarchy of parallelism is shown in Figure 2.

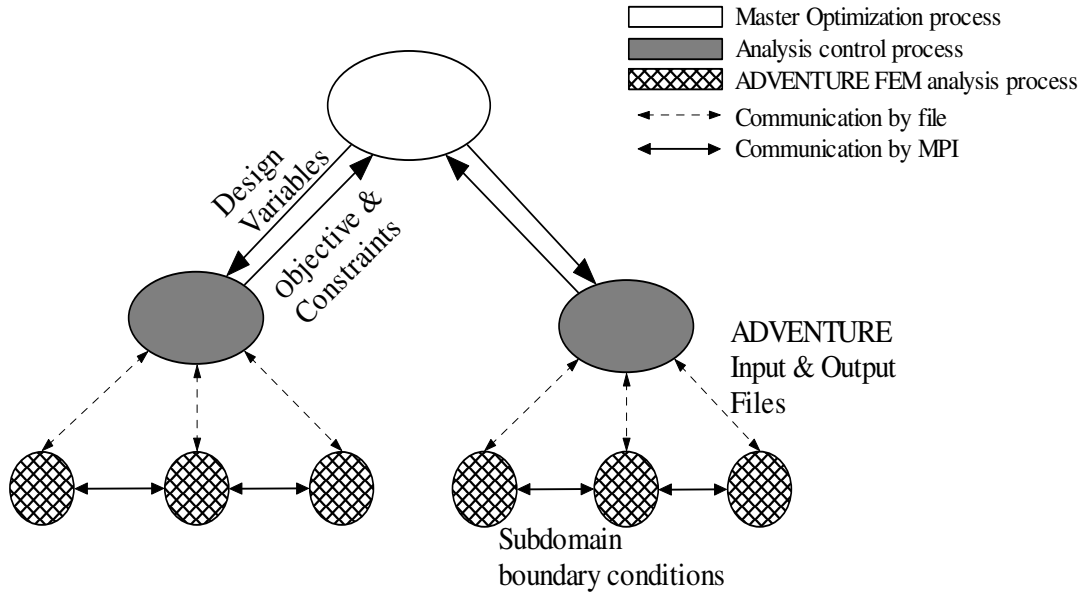


Figure 2: Multilevel parallelism in optimization

DESIGN ANALYSIS

The thermal and thermoelastic analysis is performed by parallel finite element analysis. The finite element analysis codes and tools for mesh generation, mesh partitioning, and others are freely available as a part of the ADVENTURE project [11] lead by the University of Tokyo. The finite element solvers are geared towards large-scale parallel analysis and are well suited to the efficient analysis of complicated geometries.

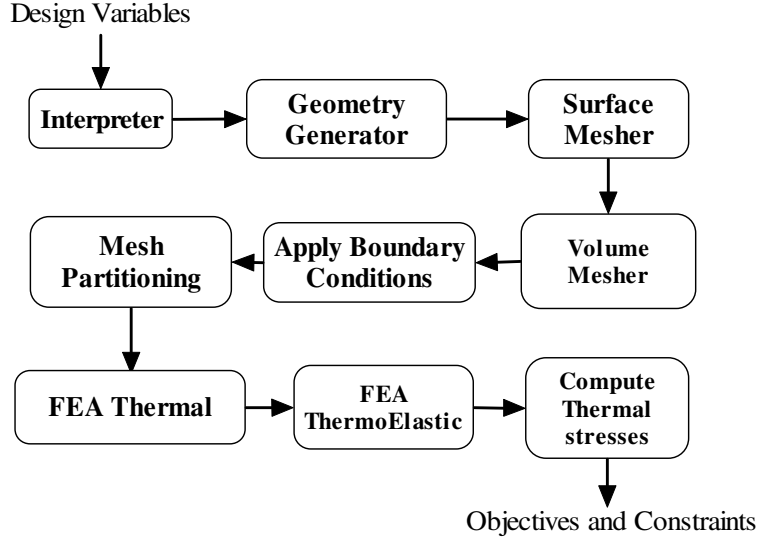


Figure 3: Modules used for automatic parallel FEA

For each design, a series of modules is required to turn a given set of design variables into optimization objective and constraint function values. The flow of data between these modules is depicted graphically in Figure 3. The analysis process may need to be performed hundreds or thousands of times for a single optimization run so it is critical that each module be automatic, robust, and computationally efficient.

OBJECTIVE AND CONSTRAINTS

In this section the design objective and constraint functions are detailed. The objective of the design optimization is to minimize the variation in stress distribution within the blade material. The objective function is computed using the maximum principal stress at each node within the blade. Mathematically, the normalized objective function is expressed as

$$F = \sum_{i=1}^n \frac{\sigma_i^2}{n \sigma_{yield}^2} \quad (1)$$

where σ_i is the maximum principal stress at node i , n is the number of nodes within the blade, and σ_{yield} is the yield stress of the blade material. Only nodes within the blade itself are considered for the objective and constraint functions.

By minimizing this objective function, a smoothing effect on the principal stress field is achieved. In addition, this objective also drives the stresses to lower values, which is also desirable for the durability of the blade.

In addition to minimizing the objective function, the optimizer must find a design that simultaneously satisfies the design constraints. For the design of a turbine rotor blade, the maximum temperature should be less than an allowable temperature, T_{allow} . Similarly, the maximum principal stress should be less than the yield stress, σ_{yield} . These two inequality constraints are expressed mathematically as

$$G_1 = \sum_{i=1}^n \frac{1}{n} \left[100 .0 \frac{(\bar{T}_i - T_{allow})}{T_{allow}} \right]^2 \quad (2)$$

$$G_2 = \sum_{i=1}^n \frac{1}{n} \left[100 .0 \frac{(\bar{\sigma}_i - \sigma_{yield})}{\sigma_{yield}} \right]^2 \quad (3)$$

where the constraints are satisfied if $G_1 \leq 0.0$ and $G_2 \leq 0.0$, while

$$\bar{T}_i = \begin{cases} T_i & \text{if } T_i > T_{allow} \\ T_{allow} & \text{if } T_i \leq T_{allow} \end{cases} \quad (4)$$

$$\bar{\sigma}_i = \begin{cases} \sigma_i & \text{if } \sigma_i > \sigma_{yield} \\ \sigma_{yield} & \text{if } \sigma_i \leq \sigma_{yield} \end{cases} \quad (5)$$

The above constraints on maximum temperature and maximum stress could have been written more simply as

$$G_1 = T_{max} - T_{allow} \quad (6)$$

$$G_2 = \sigma_{max} - \sigma_{yield} \quad (7)$$

where T_{max} and σ_{max} are the maximum nodal temperature and principal stress, respectively. However, constraints (2)-(3) have the effect of penalizing designs with many nodes with infeasible temperature or stress, where as constraints (6)-(7) only consider the worst values at a single node. In our experience we found that the constraints (6)-(7) worked well only when an initial feasible design was given at the start of the optimization. In cases where no initial feasible design was known, the constraints (2)-(3) produced superior results in fewer iterations for both GA and IOSO algorithms.

DESIGN PARAMETERIZATION

The outer blade shape was considered to be fixed and to be provided by the user at the beginning of the design optimization. The shapes of the internal coolant passages were parameterized using analytical shape functions [12,13]. The turbine blades considered in this research had a total of four straight passages connected by U-turn passages. The result is a single serpentine passage with a single inlet and outlet. The spanwise cross-sectional shape of each straight passage is described by four parameters as shown in Figure 3. These parameters include the degree of filleting in the passage, r , the blade wall thickness, d , and the passage chordwise starting and finishing point, x_1 and x_2 respectively. The passage cross-section shapes are determined at the root and the tip by user provided parameter values. The parameters for the middle sections are found by linear interpolation along the blade span.

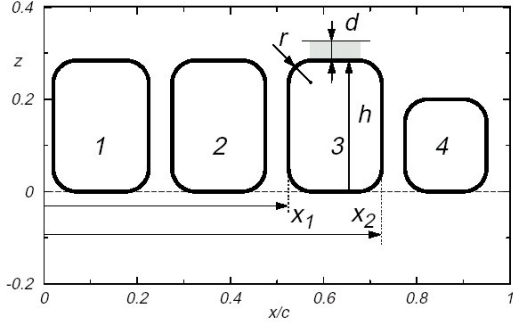


Figure 3: Parameters for passage cross-section shape in x - y plane

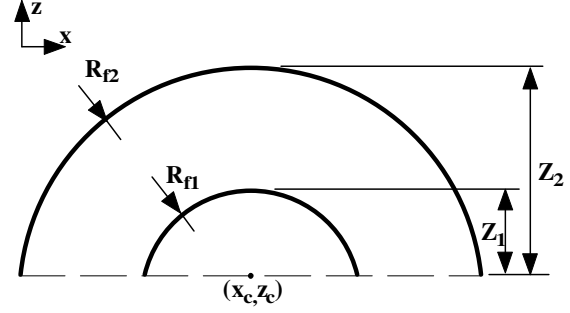


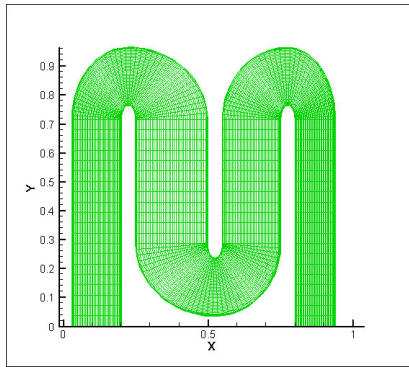
Figure 4: Parameters for U-turn shape in the x - z plane

Three U-turn shapes are used to connect the ends of the coolant passages. The wall shape of the U-turn passage is determined by using analytic functions. For wall n , the half shape can be found by using the following equations

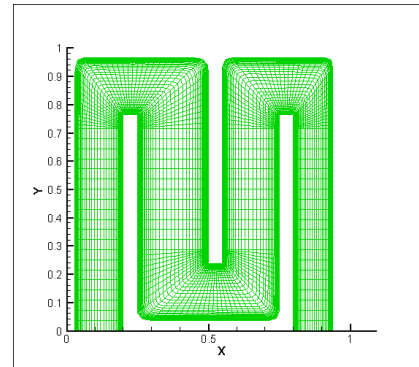
$$\begin{aligned} x_n &= (x_{\max_n} - x_c) |\cos(\theta)|^{R_{fn}} + x_c \\ z_n &= Z_n |\sin(\theta)|^{R_{fn}} + z_c \end{aligned} \quad (8)$$

where x_{\max} is the x position of the end of the straight passage wall and x_c , z_c are the x and z coordinates of the strut center. Four parameters are needed to define each U-turn shape in the x - z plane as shown in Figure 5. The parameters Z_1 and Z_2 control the position of the passage walls in the z -direction. The parameters R_{f1} and R_{f2} control the roundedness of the u-turn shape. More details on construction of the turbine blade passages and outer shape are discussed in the references [12,13].

The straight passage parameterization is somewhat limited because it cannot create designs with angled struts in the x - y plane. Also in the current approach, the number of straight passages cannot be changed easily and is fixed at four. These limitations should be addressed to further increase the usefulness of this approach for creating passage shapes.



$R_{f1}=0.95, R_{f2}=0.95$



$R_{f1}=0.15, R_{f2}=0.15$

Figure 5: Example passage shapes for variation parameters R_{f1} and R_{f2}

The following additional design parameters were also used: the coolant passage bulk temperature, T_c , and blade angle with the disk, θ_b . All together a total of 42 continuous design variables were used to uniquely describe a design.

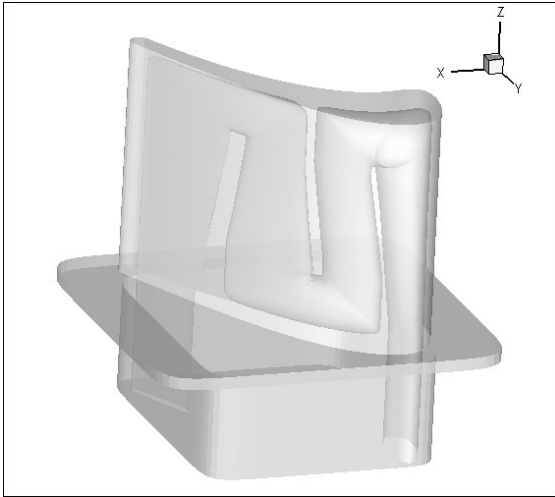


Figure 6: Internally cooled blade example

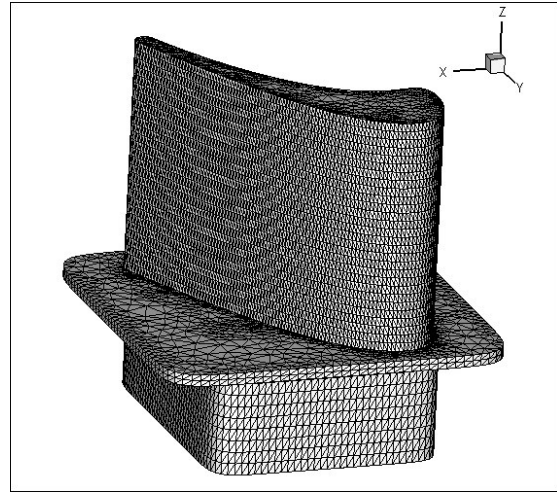


Figure 7: Triangular surface mesh for blade example

The shape parameterization code generates a block-structured grid that describes the shape of the blade. An inner shroud and blade root geometry are generated separately and added to the base of the blade section. The block-structured grids for blade, shroud, and root are then used as the base geometry for generating a triangular surface grid [14]. Sample geometry and the generated surface mesh are shown in Figures 6-7. The triangular surface mesh is then used as input to a tetrahedral mesh generation program [15].

DESIGN OPTIMIZATION EXAMPLE

The design system described in this paper was used to perform an example of design optimization of an internally cooled turbine blade. The outer blade geometry was created by generating a series of 2-D turbine airfoils [6] and stacking the sections along the z -axis. Though the generated geometry is not an actual rotor blade, we tried to make a simplified outer surface that maintains the characteristic shape of a typical turbine rotor. However, if a real outer shape is available from the user, it should be possible to use it directly with the design system with minor modifications.

In this example, the blade material was assumed to be a titanium-aluminum alloy with the properties listed in Table 1.

Table 1: Physical parameters for Titanium-Aluminum alloy

Modulus of Elasticity	118.0 GPa
Poisson Ratio	0.3
Tensile Yield Stress, σ_{yield}	1050.0 MPa

Coefficient of Thermal Expansion	7.7 $\mu\text{m}/\text{m}\cdot^\circ\text{C}$
Density	4507.0 kg/m^3
Thermal Conductivity	7.0 $\text{W}/\text{m}\cdot^\circ\text{C}$
Melting Point	1705.0 $^\circ\text{C}$

For each design mesh, the boundary conditions were applied automatically. The root section of the geometry was set to zero displacement while the blade and inner shroud were left free to deform. In this simplified problem the aerodynamic loads are not included. As for thermal boundary conditions, the outer surface of the blade and top surface of inner shroud were set to convection boundary conditions which require the specification of the convection coefficient, h_B , and the hot gas bulk temperature, T_B . Convection boundary conditions were also applied to the coolant passage surface inside the blade using h_C and T_C . All other surfaces were assumed thermally insulated. Both centrifugal and thermal body forces were applied automatically to each design mesh. Actual values used for this design example are shown in Table 2.

Table 2: Parameters for rotor design problem

Coolant convection coefficient, h_C	500.0 $\text{W}/\text{m}^2\cdot^\circ\text{C}$
Coolant bulk temperature, T_C	150.0-600.0 $^\circ\text{C}$
Hot gas convection coefficient, h_B	150.0 $\text{W}/\text{m}^2\cdot^\circ\text{C}$
Hot gas bulk temperature, T_B	1500.0 $^\circ\text{C}$
Maximum allowable temperature, T_{allow}	900.0 $^\circ\text{C}$
Angular velocity about x -axis	5000.0 RPM
Inner shroud distance from x -axis	0.25 m
Blade span	0.10 m
Blade chord	0.10 m

The optimization run was performed on a commodity component based PC cluster with 54 Pentium II 400 MHz processors. Both PGA and IOSO optimization methods were tested with this problem. A total of 12 analyses were performed per iteration for IOSO method. For PGA, 36 designs were evaluated per generation. For both cases each parallel thermoelastic FEM analysis used 4 processors. A typical analysis mesh contained over 150,000 degrees of freedom and required 4 minutes to complete a full thermoelastic analysis.

A converged result was found by the IOSO optimizer in 70 iterations after consuming approximately 12 hours of total computer time. For PGA, the total computer time was more than 30 hours. The PGA run was terminated before a converged result was found. The convergence history for the objective function for both PGA and IOSO is shown in Figure 8. For all designs the stress constraint was satisfied. However, the initial design violated the temperature constraint so the optimizer had to first determine a feasible design. The convergence history for the temperature constraint function is shown in Figure 9. This figure shows that feasible region was found at iteration 12 for IOSO and iteration 68 for PGA. For the IOSO method, after iteration 12 the best design becomes the feasible design, which is why a spike in Figure 8 occurs at iteration 12. Prior to that only infeasible designs were found and the optimizer clearly tried to improved the

objective function while searching for the feasible region. These convergence results clearly show the computational efficiency of the IOSO approach over the PGA method for this design problem.

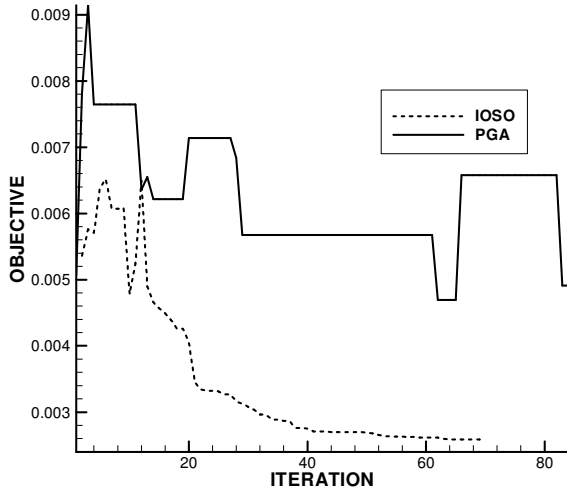


Figure 8: Objective function convergence history

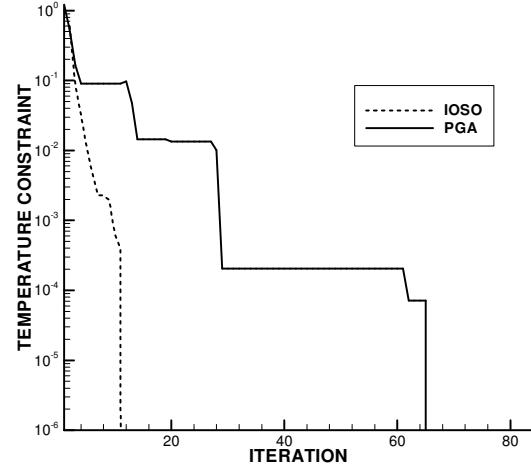


Figure 9: Temperature constraint function convergence history

The principal stress and temperature distributions for the initial design are shown in Figures 11-13. The temperature patterns on the surface of the blade follow the shapes of the passage inside the blade. This shows that the passage shape will have a strong impact on the temperature distribution and hence the thermally induced stresses. Stress in the root of the blade is high due to the centrifugal loading and temperature gradients. The IOSO optimized passage shape is shown in Figure 14. The wall near the tip corners has become much thinner obviously in an effort to keep the temperature in those regions below the maximum allowable value. The principal stress distribution for the optimized design is smoother than in the initial design, as shown in Figure 15. Similarly, the temperature distribution on the surface of the blade is considerably smoother compared with that of the initial design as shown in Figures 16-17.

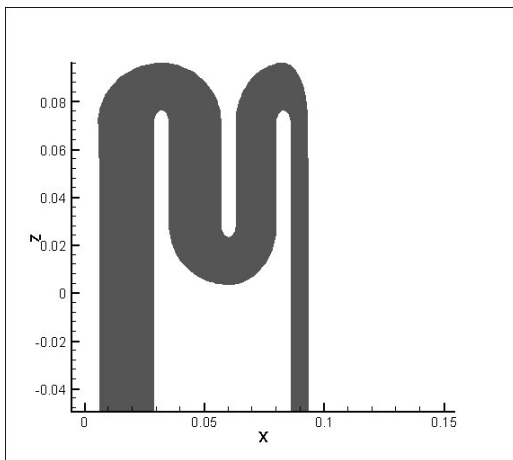


Figure 10: Passage shape in x - z plane for

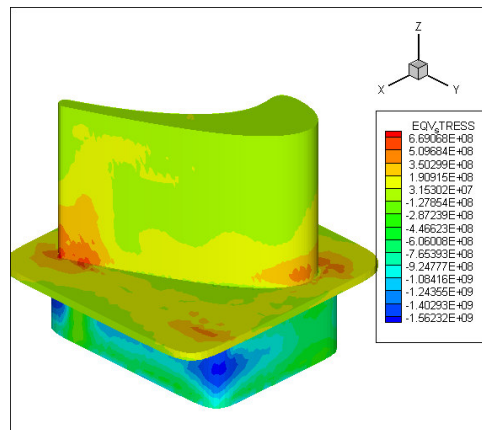


Figure 11: Principal stress contours for

initial design

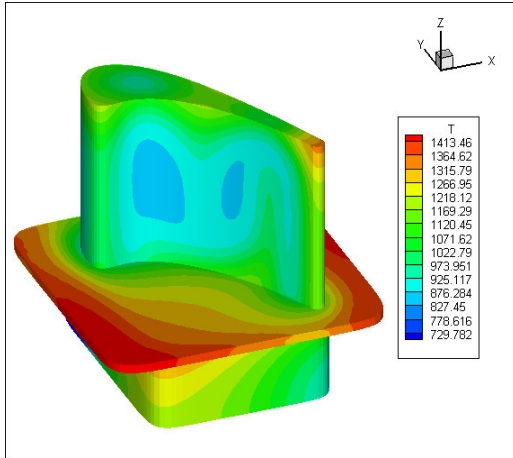


Figure 12: temperature contours for initial design on pressure side

initial design

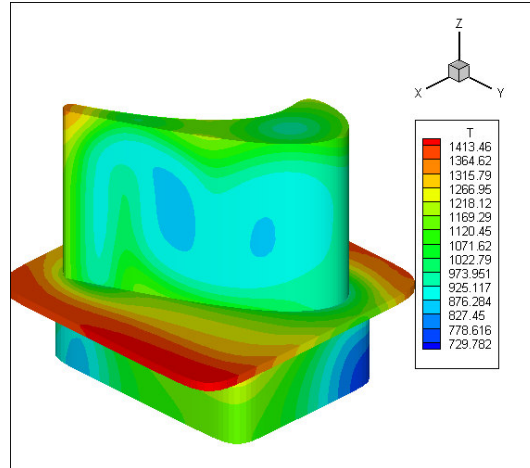


Figure 13: temperature contours for initial design on suction side

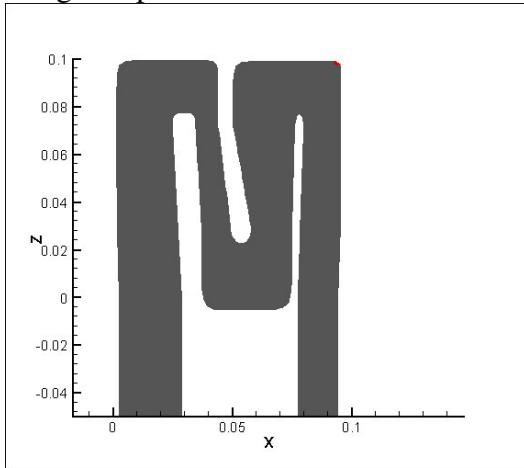


Figure 14: Passage shape in x - z plane for IOSO optimized design

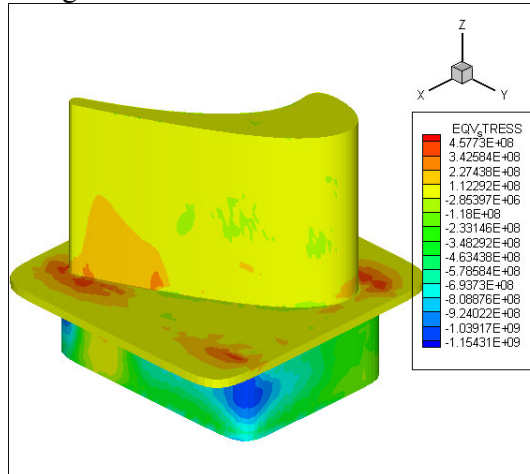


Figure 15: Principal stress contours for IOSO optimized design

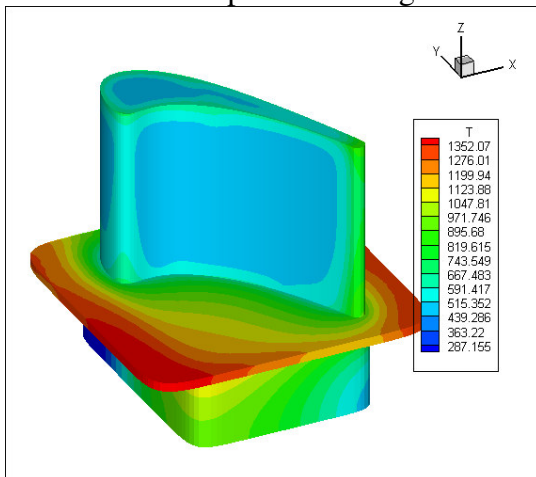


Figure 16: Temperature contours on pressure side for IOSO optimized design

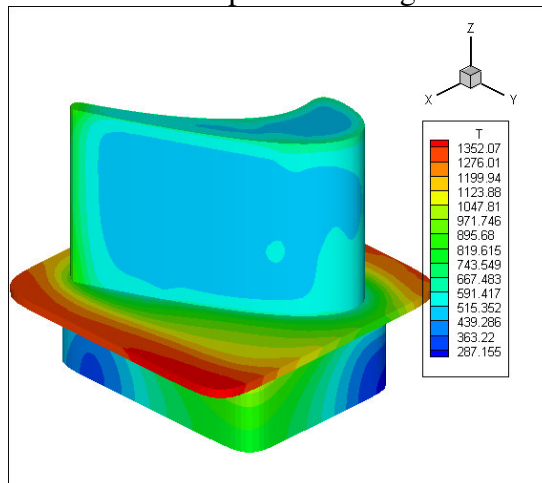


Figure 17: Temperature contours on suction for IOSO optimized design

The IOSO optimized design also includes thick walls near the root of the blade that progressively thin towards the tip of the blade as shown in Figure 14. This is an expected result since more material is needed at the root to carry the centrifugal loads. Table 3 gives a quantitative comparison between the initial and optimized passage design. The initial design exceeds the maximum allowable temperature while satisfying the stress constraint. However, the optimized blade is clearly feasible with respect to the temperature and stress constraints.

Table 3: Comparison of initial and optimized cooling passage design

Quantity	Initial	Optimized
Maximum Temperature, T_{max}	1333.8 °C	894.6 °C
Volume	$9.64 \times 10^{-4} \text{ m}^3$	$8.46 \times 10^{-4} \text{ m}^3$
Maximum Principal Stress, σ_{max}	668.9 MPa	425.1 MPa
Coolant bulk temperature, T_C	600.0 °C	158.0 °C
Objective function value, F	$6.80 \times 10^{-3} \text{ Pa}$	$2.59 \times 10^{-3} \text{ Pa}$

The volume of the optimized blade is slightly smaller than the initial design, most likely due to the thinning of the walls in the tip region of the blade. The optimized design's maximum principal stress was reduced by 36% and its objective function reduced by 62%. The optimizer reduced the coolant temperature design variable, T_C , from 600.0 °C to 158.0 °C. The reduction in coolant temperature was necessary for the satisfaction of the maximum temperature constraint. Although the temperature difference between the coolant and outer hot gas increased, the thermal stresses actually decreased. The most significant thermal stresses appear to be the result of temperature distribution along the span and chord direction. Therefore the optimizer determined the wall thickness distribution such that the variation of temperature in the chord and span direction was reduced.

CONCLUSIONS

A software system for the design of turbine blade coolant passages has been developed using power optimization algorithms and efficient parallel finite element analysis codes. The automatic parametric shape design of an internal serpentine coolant passage was demonstrated. The entire design was completed within 12 hours of computer processing time on a slow small cluster of processors. A performance comparison was made between PGA and IOSO optimization methods. For this design problem, the IOSO approach was found to be more efficient by finding better designs with fewer function evaluations. This example represents a simplified case as the effect of the inner and outer fluid mechanics is very approximate. The next step towards a complete automatic design system should be to add 3-D fluid mechanics analysis codes. However, this would then increase computational requirements by a factor of ten or more. With the recent availability of low cost parallel supercomputing based commodity component, a complete multidisciplinary design system may be proven to be computationally and financially feasible in the very near future.

ACKNOWLEDGEMENTS

This work was done partly in "Frontier Simulation Software for Industrial Science (FSIS)" project supported by IT program of Japan Ministry of Education, Culture, Sports, Science and Technology (MEXT) and partly in the ADVENTURE project of Research for the Future (RFTF) program supported by the Japan Society for Promotion of Science (JSPS).

REFERENCES

1. Martin, T. J. and Dulikravich, G. S., 2001, "Aero-Thermo-Elastic Concurrent Design Optimization of Internally Cooled Turbine Blades", Chapter 5 in *Coupled Field Problems, Series on Advances in Boundary Elements* (eds: Kassab, A. J. and Aliabadi, M. H.), WIT Press, Boston, MA, pp. 137-184.
2. Talya et. al., 2002, "Multidisciplinary Design Optimization Procedure for Improved Design of a Cooled Gas Turbine Blade," *Engineering Optimization*, Vol. 32, No. 2.
3. Muller et. al., 2001, "Evolution Strategies for Film Cooling Optimization," *AIAA Journal*, Vol. 39, No. 3.
4. Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley.
5. Carroll, D. L., 1996, "Chemical Laser Modeling with Genetic Algorithms," *AIAA Journal*, Vol. 34, 2, pp. 338-346.
6. Dennis, B.H., Han, Z.-X. and Dulikravich, G.S., 2001, "Optimization of Turbomachinery Airfoils with a Genetic/Sequential Quadratic Programming Algorithm," *AIAA Journal of Propulsion and Power*, Vol. 17, No. 5, Sept-Oct 2001, pp. 1123-1128
7. Dulikravich, G. S., Martin, T. J., Dennis, B. H. and Foster, N. F., 1999, "Multidisciplinary Hybrid Constrained GA Optimization", Invited lecture, Chapter 12 in EUROGEN'99 - *Evolutionary Algorithms in Engineering and Computer Science: Recent Advances and Industrial Applications*, (editors: K. Miettinen, M. M. Makela, P. Neittaanmaki and J. Periaux), John Wiley & Sons, Ltd., Jyvaskyla, Finland, May 30 - June 3, 1999, pp. 231-260.
8. Egorov, I. N., 1998, "Indirect Optimization Method on the Basis of Self-Organization", Curtin University of Technology, Perth, Australia., *Optimization Techniques and Applications (ICOTA'98)*, Vol.2, pp. 683-691.
9. Egorov, I. N., Kretinin, G. V., Leshchenko, I. A. and Kostiuk, S. S., 1999, "The Methodology of Stochastic Optimization of Parameters and Control Laws for the Aircraft Gas-Turbine Engines Flow Passage Components", ASME paper 99-GT-227.
10. Gropp, W., Lusk, E., and Skjellum, A., 1994, *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, MIT Press.
11. ADVENTURE Project Homepage, <http://adventure.q.t.u-tokyo.ac.jp>.
12. Sobieczky, H. and Dulikravich, G. S., 1993, "Parameterized Aerospace Vehicles for Aerothermodynamic Optimization," DLR-Technical Note H95F-12.93, Goettingen, Germany.
13. Sobieczky, H., Dulikravich, G. S. and Dennis, B. H., 2002, "Parameterized Geometry Formulation for Inverse Design and Optimization," 4th International Conference on Inverse Problems in Engineering, Rio de Janeiro, Brazil.

14. Shewchuk, J.R., 1996, "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator," First Workshop on Applied Computational Geometry, Philadelphia, PA.
15. Marcum, D. L. and Weatherhill, N. P., 1995, "Unstructured Grid Generation Using Iterative Point Insertion and Local Reconnection", *AIAA Journal*, Vol. 33, No. 9, pp. 1619-1625.