**Proceedings of XIII International Symposium on Airbreathing Engines - XIII ISABE (editor: F. Billig)**
Chattanooga, TN, September 8-12, 1997.

# Thermo-elastic Analysis and Optimization Environment for Internally Cooled Turbine Airfoils

Brian H. Dennis * and George S. Dulikravich †
*Department of Aerospace Engineering*
*The Pennsylvania State University*
*University Park, PA 16802*

**A software system is presented for the analysis and optimization of coolant passsages for internally cooled turbine airfoils. The system consists of several layers including grid generation, finite element solver, and optimizer. Analysis results are presented for 2-D airfoil and 3-D turbine blade models. A description of an interactive interface for the development of 3-D turbine blade shapes is also given.**

## Nomenclature

| | |
|---|---|
| $\sigma$ | Normal stress |
| $\tau$ | Shear stress |
| $X$ | Body force in $x$ direction |
| $Y$ | Body force in $y$ direction |
| $Z$ | Body force in $z$ direction |
| $T$ | Temperature |
| $k$ | Fourier coefficient of heat conduction |
| $Q$ | Heat source |
| $q$ | Heat flux |
| $E$ | Elastic modulus of elasticity |
| $\alpha$ | Coefficient of thermal expansion |
| $\Delta T$ | Change in temperature |
| $B$ | Body force due to change in temperature $\Delta T$ |
| $x, y, z$ | Cartesian body axes |

## Introduction

A new software system is being developed for the thermoelastic analysis and optimization of internally cooled turbine airfoils. The system consists of three main parts: grid generation, analysis code, and optimizer. The grid generation is required to produce unstructured grids for the analysis code for a given design during the optimization sequence. The analysis code is based on the finite element method and can perform analysis of heat transfer, elasticity, or thermo-elasticity in two or three dimensions. The analysis code is unique in that it is object-oriented and employs advanced iterative techniques and efficient sparse matrix storage techniques. An interactive tool for the generation of 3-D turbine blade shapes has also been developed.

## Finite Element Solver

The solver in this system is based on the finite element method. The solver can perform analysis in heat transfer, elasticity, thermoelasticity in two or three dimensions. The governing differential equations that describe three-dimensional static elasticity[1] are:

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} + X = 0 \tag{1}$$

$$\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{yz}}{\partial z} + Y = 0 \tag{2}$$

$$\frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \sigma_z}{\partial z} + Z = 0 \tag{3}$$

These equations are solved via the finite element method for problems where elasticity is important. Heat conduction is governed by Poisson's equation:

$$-(\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} + \frac{\partial q_z}{\partial z}) = Q \tag{4}$$

where

$$q_x = -(k_{11}\frac{\partial T}{\partial x} + k_{12}\frac{\partial T}{\partial y} + k_{13}\frac{\partial T}{\partial z}) \tag{5}$$

$$q_y = -(k_{21}\frac{\partial T}{\partial x} + k_{22}\frac{\partial T}{\partial y} + k_{23}\frac{\partial T}{\partial z}) \tag{6}$$

$$q_z = -(k_{31}\frac{\partial T}{\partial x} + k_{32}\frac{\partial T}{\partial y} + k_{33}\frac{\partial T}{\partial z}) \tag{7}$$

These equations are solved with the finite element method for problems involving heat transfer. For thermoelasticity, equation (4) is solved for the temperature field, then equations (1)–(3) are solved including the body forces induced by the temperature field given by equation (8).

$$B = E\alpha\Delta T \tag{8}$$

*Graduate Research Assistant
†Associate Professor. AIAA Associate Fellow

#### Discretization

The governing differential equations are discretized using finite elements, more specifically, a Galerkin approach.[2] Both quadratic and linear Lagrangian shape functions were used to formulate the stiffness matrix and force vectors for three and nine node triangles and four and ten node tetrahedrons. Stiffness and force vectors for 2-D and 3-D elements were integrated analytically using Mathematica and a symbolic integration program developed specifically for integration over triangles and tetrahedrons. Using analytical integration allows one to avoid errors that occur in numerical integration over poorly shaped elements.[3]

#### Object-oriented Program Design

The solver was written in C++ and constructed using object-oriented programming to allow for a code which is easy to extend and improve.[4] In the future, we would like to extend this code into other disciplines such as fluid mechanics and electromagnetics. The base classes will provide the basic functions and data structures which are common to all finite element programs. Figure 1 shows the simplified object-oriented design of the finite element solver.

#### Iterative Solvers

The finite element analysis code uses a variety of iterative methods to solve the sparse linear system of equations that result from finite element discretizations. These methods are:

- Krylov Subspace Methods
  - Conjugate Gradient
  - Conjugate Gradient Squared
  - BiConjugate Gradient Stabilized
  - QMR
  - GMRES
- Classical Methods
  - Jacobi
  - Gauss-Seidel
- Multilevel Methods
  - Multigrid
  - Overlapping Domain Decomposition
- Preconditioners
  - Diagonal
  - Incomplete Lower-Upper
  - Multigrid

One main advantage of iterative methods is that they require less storage than more traditional direct methods.[5] The lower memory requirements gives our finite element solver the ability to run on modest PC's and low-end workstations.

#### Accuracy

The analysis code was tested for several problems with known analytic solutions. For elasticity, 3-D models of cantilever beams were constructed and solved for cases with discrete and distributed transverse loads, discrete and distributed axial loads, and applied torsion moments at the tip. Results were compared to the exact 1-D solutions. Maximum error in the deflections in each case was less than 1%. The thermal portion was tested against the exact solution for heat conduction through a 1-D bar and 2-D plate. The boundary conditions used for the test include adiabatic, specified temperatures, and convective heat transfer. In all cases, error was less than 1%. Thermo-elastic cases tested include a uniformly heated bar constrained at one end and a uniformly heated bar constrained at both ends. Again, the finite element solution was less than 1% in error compared to the exact 1-D solution.

## Grid Generation

Unstructured grids are required due to the complexity of the turbine blade and the coolant passage geometry. Also, since the geometries will be changing during the optimization process, automatic schemes are necessary to provide remeshing when the geometry is changed. In our system, automatic unstructured grid generation is available in two and three dimensions. A Delauney-based triangulation with point insertion[6] is used for triangulation in two dimensions. Three-dimensional grid generation is based on an advancing-front technique which uses a structured background grid.[7] Figure 2 shows a three-dimensional turbine blade with coolant passages meshed with the advancing-front technique. We plan to implement a more robust grid generation scheme based on a combination of advancing-front and Delauney algorithm[8,9]

## Interactive Generation of 3-D Blade Shapes

We have developed an interactive program for the visualization and construction of three-dimensional turbine blade shapes. This utility was written in Visual C++ and runs on Windows95 and WindowsNT platforms. The program has been useful for the generation of blade geometries for testing grid generators and for providing initial guesses for the optimizer. This program allows the user to manipulate ten two-dimensional blade sections as shown in Fig. 3. The two-dimensional blade geometry model is based on methods described in Ref. 10. The internal coolant passages are described with B-splines. The ten blade sections are stacked on the center of gravity of the root section to produce a three-dimensional blade as shown in Fig. 4. The blade geometry can be exported to a file which can be used as input for the grid generator.

The user can adjust the blade shape in real time by using the sliders shown in Fig. 3. Parameters such as the airfoil radius, axial chord, leading edge radius, exit blade angle, tangential chord, and several other parameters can be modified by the user by moving the respective slider control. The user can also manipulate the coolant passages my dragging and dropping the B-spline control points or by changing the spline tension.

## Analysis Results

### Stress Analysis of 3-D Solid Turbine Blade

A static elastic analysis was performed on a rotating solid steel turbine blade shown in Fig. 5. Results are shown in Figs. 6, 7, 8.

### Thermo-elastic Analysis of a 2-D Turbine Blade

A thermo-elastic analysis was performed on a 2-D turbine airfoil shown in Fig. 9. Solution results are shown in Figs. 10, 11, 12, 13.

## Optimizer

Our objective is to use optimization to design the coolant passages of the turbine airfoil while minimizing both the blade weight and the amount of coolant required. The optimizer attempts to achieve these objectives by shaping the coolant passages inside the blade while subject to several constraints. The passage design constraints include local stresses not exceeding material failure limits, wall thickness must be greater than a user specified value, and maximum blade deflections must be lower than a user specified value.

The optimizer that we developed and built into the system is based on a constrained genetic algorithm and written in C++. The genetic algorithm is useful for this problem due to its ability to avoid local minima.[11] A foundation for future work in this area is given in reference 12.

## Acknowledgments

The authors would like to express their appreciation for the grants received from the National Science Foundation DMII Program monitored by Dr. George A. Hazelrigg, NASA Lewis Research Center monitored by Dr. John K. Lytle, and ALCOA Foundation facilitated by Dr. Yimin Ruan. We also are grateful for the help received from Dr. Pirzadeh and Dr. Frink of NASA LaRC working with non-structured grid generation.

## References

[1]Love A.E.H. (1972) *A Treatise on the Mathematical Theory of Elasticity* Dover, New York, fourth edition.

[2]Dhatt G. and Touzot G. (1984) *The Finite Element Method Displayed* John Wiley and Sons, New York.

[3]Mottram J.T. and Shaw C.T. (1996) *Using Finite Elements in Mechanical Design* McGraw-Hill, London.

[4]Buzzi-Ferraris G. (1993) *Scientific C++* Addison-Wesley, Wokingham, England.

[5]Golub G.H. and Loan C F. Van (1996) *Matrix Computations* Johns Hopkins, Baltimore, MD, third edition.

[6]Shewchuk J. R. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator First Workshop on Applied Computational Geometry, May 1996.

[7]Pirzadeh S. (1993) Structured Background Grids for Generation of Unstructured Grids by the Advancing-Front Method. *AIAA Journal.* **31**(2), 257–265.

[8]Marcum D. L. and Weatherill N. P. (1995) Unstructured Grid Generation Using Iterative Point Insertion and Local Reconnection. *AIAA Journal.* **33**(9), 1619–1625.

[9]Mavriplis D. J. (1992) An Advancing Front Delauney Triangulation Algorithm Designed for Robustness Technical Report 92-49, ICASE, Hampton, VA.

[10]Pritchard LJ. An Eleven Parameter Axial Turbine Airfoil Goemetry Model (1985) In *Gas Turbine Conference and Exhibit*, Houston, TX. ASME 85-GT-219.

[11]Foster N.F. and Dulikravich G.S. (1997) Three-Dimensional Aerodynamic Shape Optimization Using Genetic and Gradient Search Algorithms. *AIAA Journal of Spacecraft and Rockets.* **34**(1), 36–42.

[12]Martin T.J. and Dulikravich G.S. Aero-thermal Analysis and Optimization of Internally Cooled Turbine Airfoils (1997) In *13th International Symposium on Airbreathing Engines[XIII ISABE]*, Chattanooga, TN.

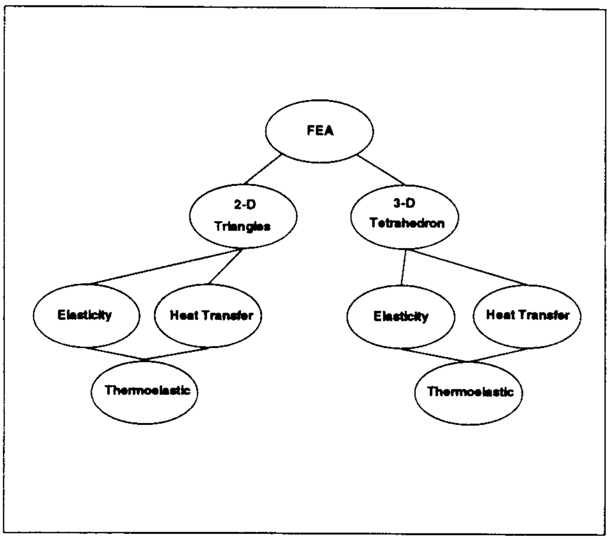
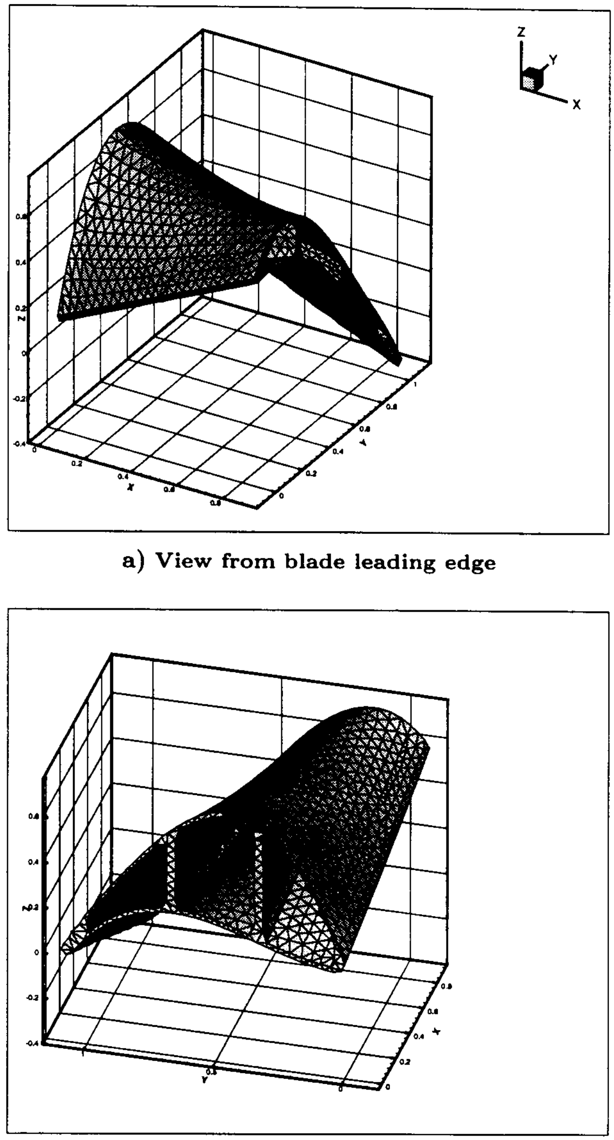


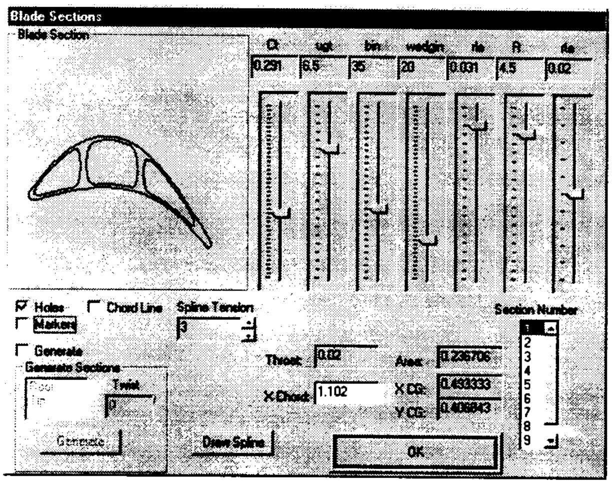

**Fig. 1 Class hiearchy of the finite element solver.**



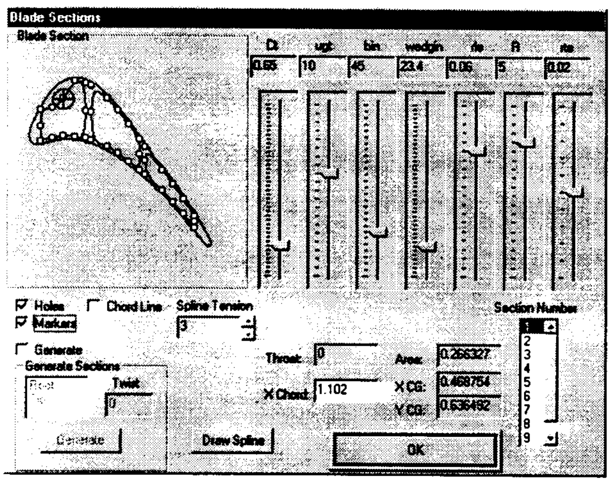

a) View from blade leading edge

b) View from blade root

**Fig. 2 Tetrahedron mesh of a turbine blade with coolant passages**

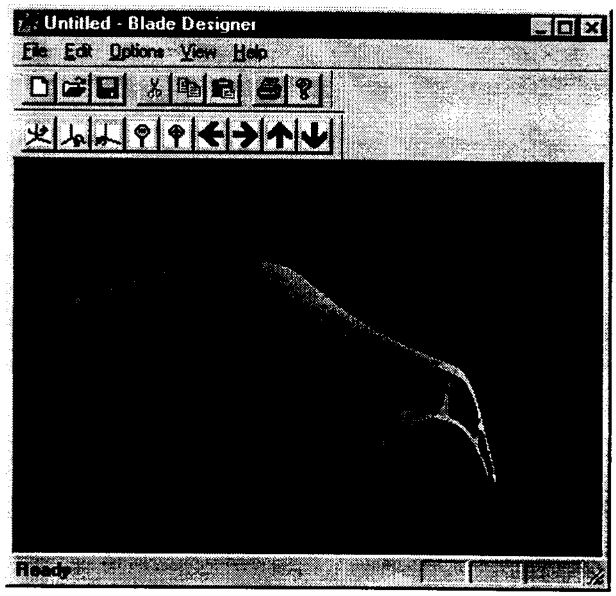



a) Initial 2-D blade

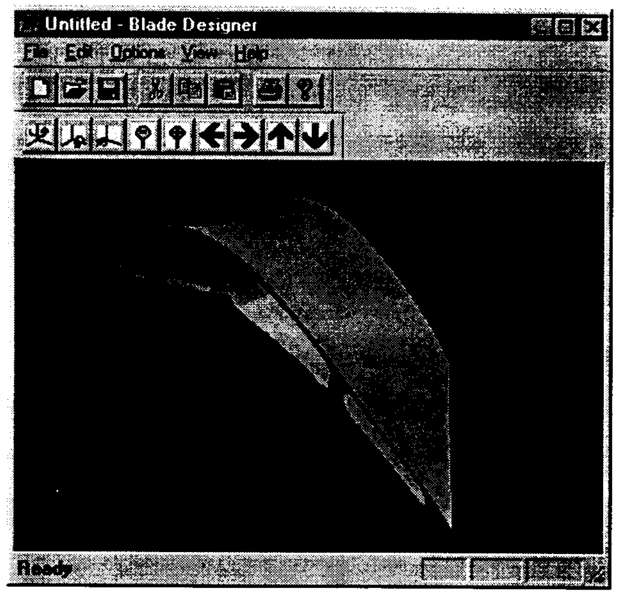



b) User can interactively manipulate the blade

**Fig. 3 Design of blade sections**




a)




b)

**Fig. 4 Resulting 3D blades**

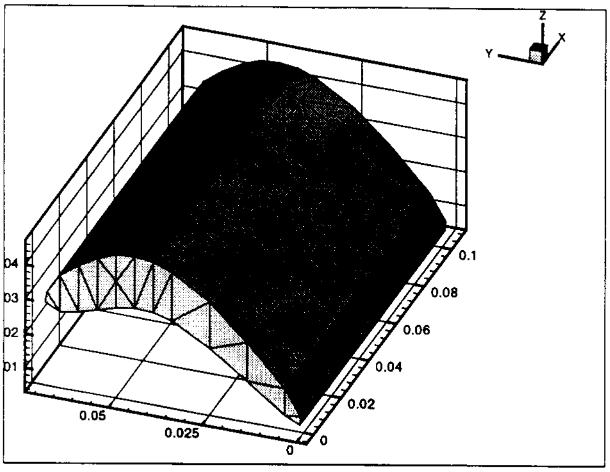



**Fig. 5** Tetrahedral mesh of a solid turbine blade

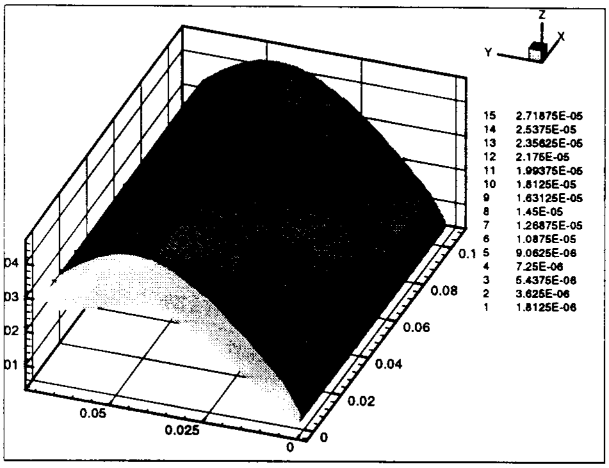



**Fig. 6** Radial displacements of a rotating solid turbine blade

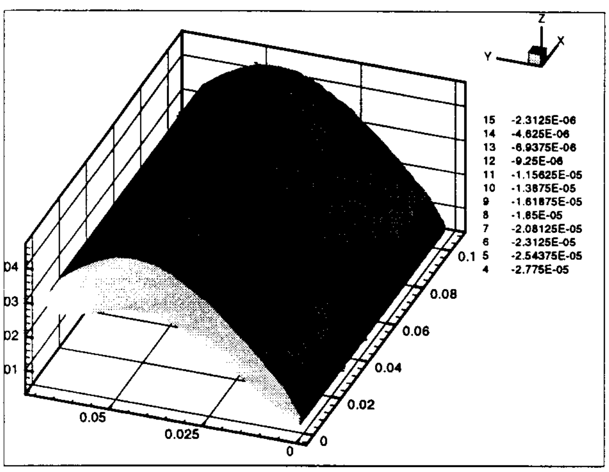



**Fig. 7** Transverse displacements of a rotating solid turbine blade

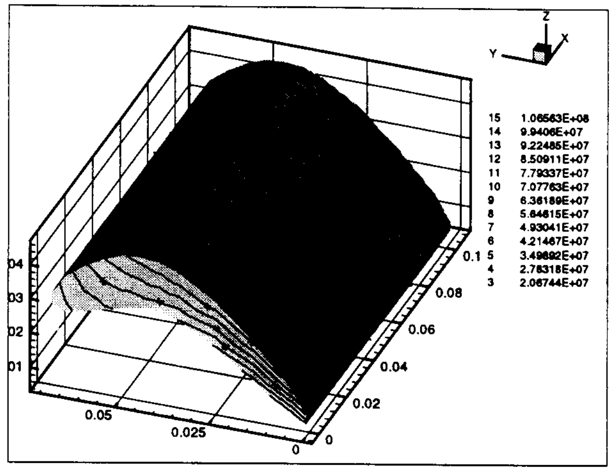



**Fig. 8** Radial stresses of a rotating solid turbine blade

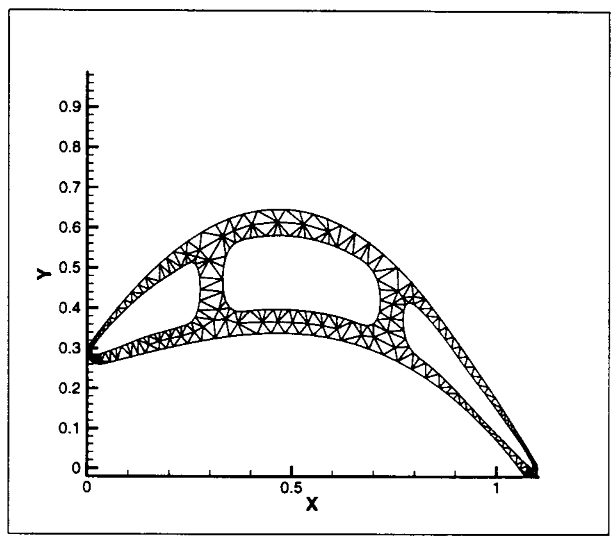



**Fig. 9** Triangular mesh of a turbine blade with coolant passages

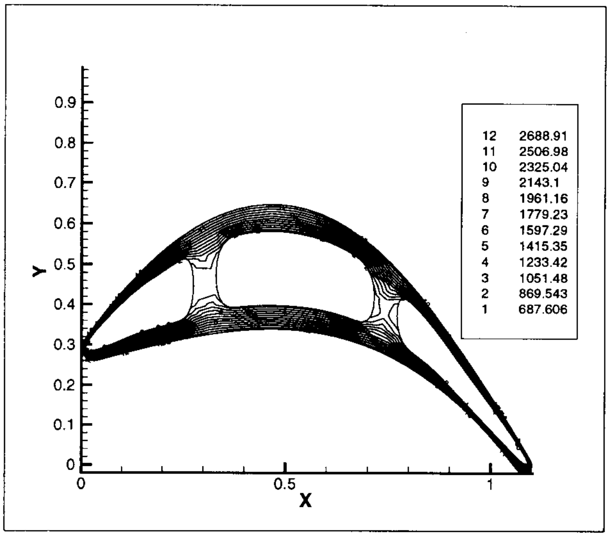



**Fig. 10 Temperature distribution for blade with convection boundary conditions**

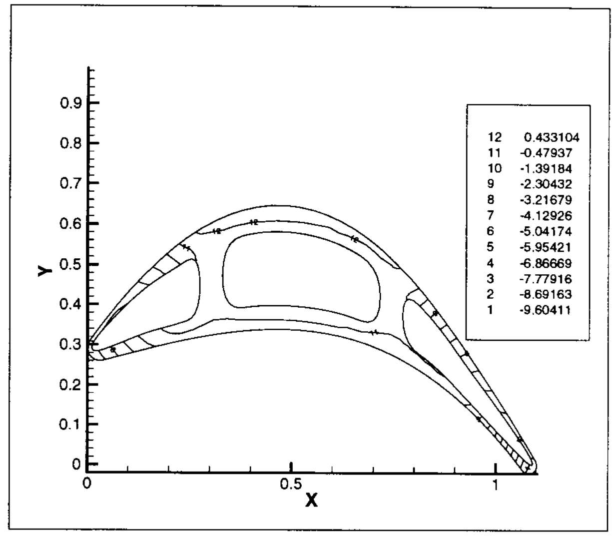



**Fig. 12 Deflections in $y$ direction**

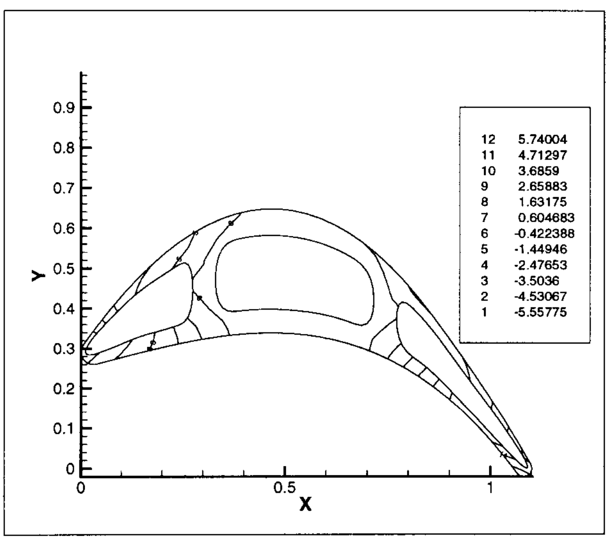



**Fig. 11 Deflections in $x$ direction**

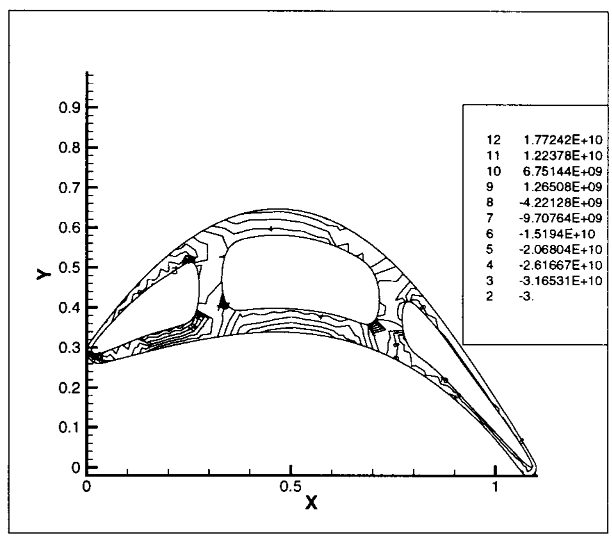



**Fig. 13 Principle stress distribution**