Paper ID: ETC2017-336

Proceedings of 12th European Conference on Turbomachinery Fluid dynamics & Thermodynamics ETC12, April 3-7, 2017; Stockholm, Sweden

IMPLEMENTATION OF AN ADJOINT THERMAL SOLVER FOR INVERSE PROBLEMS

P. Jaksch

Siemens Industrial Turbomachinery AB, Finspong, Sweden, peter.jaksch@siemens.com

ABSTRACT

This paper contains a derivation of the adjoint equations for transient heat conduction and describes how to include the adjoint solution in a nonlinear conjugate gradient method. The adjoint equations are useful for optimization problems where a scalar valued target function is optimized with respect to a large number of design variables. In this paper the target function is a least squares fit between calculated and measured transient metal temperatures. The design variables consist of heat transfer coefficients on a set of mesh facets for a finite element mesh. Contrary to intuition the sensitivity of the target function with respect to the different design variables can be evaluated by just two transient solutions; the direct problem and the adjoint problem, with roughly similar cost. If a brute force approach is used the transient problem has to be solved for each individual design variable. Since a typical mesh contains thousands of facets the speedup of the adjoint approach is enormous.

INTRODUCTION

Adjoint methods for sensitivity calculations are well established within academic research and have been used for applications such as CFD, structural mechanics, and heat transfer (see e.g. Giannakoglou and Papadimitriou (2008), Martins et al. (2005), Neto and Neto (2013)). They originate from the use of Lagrange multipliers in optimization problems (see e.g. Nash and Sofer (1995)) and have been used in optimal control for a long time (Lions (1971)). Depending on the application, the methods can be used in a discrete or a continuous setting. Discrete adjoint methods are useful for discretized problems on matrix form, such as linear elastic structural mechanics. One advantage with this method is that it is relatively easy to verify implementations against finite differences. For time dependent problems the situation may be a bit more difficult since the solution of these problems depend on time step size, method of integration etc. One option is to use algorithmic differentiation (Corliss et al. (2002)) which is used in both Open Source (SU2, OpenFOAM) and commercial CFD packages (Star-CCM+, ANSYS). This approach, however, requires relatively large program code changes and may impact numerical performance if not performed carefully. Another option is to use the *continuous* adjoint method. In this case the governing equations themselves are re-derived in the adjoint setting. Disadvantages of this method include that the mathematical derivation may be cumbersome and that the implementation can be difficult to verify against finite differences because of mesh resolution and time step effects. On the other hand, the big advantage of the continuous adjoint method is that it often requires only minor changes to existing program code.

The purpose of this paper is to provide a method of transient thermal calibration that is compatible with existing in-house program code and gas turbine design processes, thus enabling a high user productivity. The continuous adjoint method fulfills these requirements well and is selected as the method of choice. For the sake of clarity the adjoint equations are re-derived in this setting.

DERIVATION OF THE ADJOINT PROBLEM

The thermal modeling for transient heat transfer with convective boundary conditions is described by the equations

$$
\begin{aligned}
\rho c \frac{\partial T}{\partial t} &= \nabla \cdot (k \nabla T) \quad in\, \Omega, \quad 0 \le t \le t_f &(1)\\
k \frac{\partial T}{\partial \mathbf{n}} &= h(T - T_g) \quad on\, \Gamma, \quad 0 \le t \le t_f &(2)\\
T(\mathbf{x}, 0) &= T_0(\mathbf{x}) \quad in\, \Omega &(3)
\end{aligned}
$$

where $k(\mathbf{x}, T)$ is the conductivity of the material, $\rho(\mathbf{x}, T)$ is the density, $c(\mathbf{x}, T)$ is the heat capacity, $h(\mathbf{x}, t)$ is the heat transfer coefficient, $T_g(\mathbf{x}, t)$ is the gas temperature, and $\mathbf{n}$ is the outward normal vector. Equations (1) to (3) are referred to as the direct problem in this paper.

The goal is to modify the heat transfer coefficient in a subset Γ_1 of the boundary in order to minimize the target function [1]

$$J = \frac{1}{2} \int_0^{t_f} \int_\Gamma (T(\mathbf{x}, t) - T_m(\mathbf{x}, t))^2 \, d\Gamma dt, \tag{4}$$

where $T(\mathbf{x}, t)$ satisfies equations (1) - (3) and $T_m(\mathbf{x}, t)$ is the measured temperature at location $\mathbf{x}$ for time t. According to the Lagrangian formulation minimization problems of this type with equality constraints (equations (1) to (3) are considered as equality constraints) can be solved by finding stationary points of the corresponding Lagrangian function

$$L = J + \int_0^{t_f} \int_\Omega \lambda(\mathbf{x}, t) \left(\rho c \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) \right) d\Omega dt \tag{5}$$

Here, λ is the Lagrange multiplier. Stationary points to the Lagrangian function can be found by enforcing that the variations δL_λ and δL_h are identical to 0. The meaning of a variation δL_h is that the function h is changed into $h + \varepsilon\eta$, where η is a small number and ε is a continuously differentiable function. An expression for δL_h can be derived by calculating the difference $L(h + \delta h) - L(h)$. The variation δh in the heat transfer coefficient is assumed to cause a variation δT in the temperature.

$$
\begin{aligned}
\delta L_h \;=\;& \int_0^{t_f} \int_\Omega \lambda \left(\rho c \frac{\partial (T + \delta T)}{\partial t} - \nabla \cdot (k \nabla (T + \delta T)) \right) d\Omega dt \\
+\;& \frac{1}{2} \int_0^{t_f} \int_\Omega (T + \delta T - T_m)^2 \, d\Omega dt \\
-\;& \int_0^{t_f} \int_\Omega \lambda \left(\rho c \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) \right) d\Omega dt \\
-\;& \frac{1}{2} \int_0^{t_f} \int_\Gamma (T - T_m)^2 \, d\Gamma dt
\end{aligned} \tag{6}
$$

[1] In literature the target function is often written as a summation. It can be turned into an integral by introducing Dirac delta functions. This is however unattractive from a numerical point of view since it introduces high gradients in both space and time.

Due to linearity of the operators $\partial/\partial t$ and ∇ and ignoring second order variations it is possible to simplify equation (6) to

$$\begin{aligned}\delta L_h &= \int_0^{t_f}\int_\Omega \lambda\left(\rho c\frac{\partial \delta T}{\partial t} - \nabla\cdot(k\nabla\delta T)\right) d\Omega dt \\ &+ \int_0^{t_f}\int_\Gamma \delta T\,(T-T_m)\,d\Gamma dt \end{aligned} \tag{7}$$

Using integration by parts it is possible to rewrite the first term

$$\begin{aligned}&\int_0^{t_f}\int_\Omega \lambda\rho c\frac{\partial\delta T}{\partial t} d\Omega dt = \int_\Omega \left[\lambda\rho c\delta T\right]_0^{t_f} d\Omega \\ &-\int_0^{t_f}\int_\Omega \rho c\frac{\partial\lambda}{\partial t} d\Omega\delta T\end{aligned} \tag{8}$$

In a similar fashion the second term in (7) can be rewritten using Green's second identity.

$$\begin{aligned}&\int_0^{t_f}\int_\Omega -\lambda\nabla\cdot(k\nabla\delta T) d\Omega dt = \\ &\int_0^{t_f}\int_\Omega -\delta T\nabla\cdot(k\nabla\lambda) d\Omega dt + \\ &\int_0^{t_f}\int_\Gamma k\left(\delta T\frac{\partial\lambda}{\partial\mathbf{n}} - \lambda\frac{\partial\delta T}{\partial\mathbf{n}}\right) d\Gamma dt\end{aligned} \tag{9}$$

Inserting (8) and (9) into (7) and noting that $\delta T(t=0)$ is identical to 0 because of the initial condition for T.

$$\begin{aligned}\delta L_h &= \int_0^{t_f}\int_\Omega \delta T\left(-\rho c\frac{\partial\lambda}{\partial t} - \nabla\cdot(k\nabla\lambda)\right) d\Omega dt \\ &+ \int_0^{t_f}\int_\Gamma \delta T\,(T-T_m)\,d\Gamma dt + \int_\Omega \lambda\rho c\delta T(t=t_f) d\Omega \\ &+ \int_0^{t_f}\int_\Gamma k\left(\delta T\frac{\partial\lambda}{\partial\mathbf{n}} - \lambda\frac{\partial\delta T}{\partial\mathbf{n}}\right) d\Gamma dt\end{aligned} \tag{10}$$

The last term in (10) can be rewritten by taking the variation of the boundary condition (2).

$$\begin{aligned}&k\frac{\partial(T+\delta T)}{\partial\mathbf{n}} - k\frac{\partial T}{\partial\mathbf{n}} \\ &= (h+\delta h)(T+\delta T-T_g) - h(T-T_g) \\ &\to k\frac{\partial\delta T}{\partial\mathbf{n}} = \delta h(T-T_g) + h\delta T\end{aligned} \tag{11}$$

Equation (10) has to be identically equal to 0 at a stationary point of the Lagrangian (5). Since the variation δT is an arbitrary function, λ is chosen to satisfy the following equations.

$$\begin{aligned}-\rho c\frac{\partial\lambda}{\partial t} &= \nabla\cdot(k\nabla\lambda) \quad in\,\Omega, \quad 0\le t\le t_f \\ k\frac{\partial\lambda}{\partial\mathbf{n}} &= h\lambda - (T-T_m) \quad on\,\Gamma, \quad 0\le t\le t_f \\ \lambda(\mathbf{x},t_f) &= 0 \quad in\,\Omega\end{aligned} \tag{12}$$

With this choice of λ the variation δL_h is reduced to

$$\delta L_h = -\int_0^{t_f} \int_\Gamma \delta h \lambda (T - T_g) d\Gamma dt \tag{13}$$

A remarkable similarity between the equations (1) to (3) for the temperature and the equations (12) for the Lagrange multiplier λ can be noted. By the variable transformation $\tau = t_f - t$ the equations become even more similar.

$$\begin{aligned} \rho c \frac{\partial \lambda}{\partial \tau} &= \nabla \cdot (k \nabla \lambda) \quad in\, \Omega, \quad 0 \leq \tau \leq t_f \\ k \frac{\partial \lambda}{\partial \mathbf{n}} &= h\lambda - (T - T_m) \quad on\, \Gamma, \quad 0 \leq \tau \leq t_f \\ \lambda(\mathbf{x}, 0) &= 0 \quad in\, \Omega \end{aligned} \tag{14}$$

The above equations (14) are referred to as the adjoint problem. Note that the adjoint equations are effectively integrated backwards in time because of the transformation $\tau = t_f - t$. This requires that the computed values of T are saved at all time points. There are a number of practical problems associated with this procedure since data volumes can become very large. For the numerical calculations in this paper a checkpointing approach has been used where values at regular time intervals are stored on disc. To calculate $T(\tau)$ linear interpolation between checkpoint values has been used.

By repeating the exercise for the variation δL_λ one quickly finds that it just leads back to the original direct problem. This derivation will not be done here.

CALCULATION OF SENSITIVITY

It remains to establish the connection between the variation (13) and the sensitivity of the target function (4) with respect to the values of h. The sensitivity is defined as the derivative of the target function with respect to the value of h at a certain location $\mathbf{x}$ and a certain time t. The shorthand notation J_h' will be adopted for the sensitivity. From the definition of the functional derivative

$$\delta L_h = \int_0^{t_f} \int_\Gamma L_h' \delta h d\Gamma dt \tag{15}$$

Since δh is arbitrary a comparison between (15) and (13) yields that

$$L_h' = -\lambda (T - T_g) \tag{16}$$

Since the temperature T satisfies the direct problem equation (5) directly implies that $J_h' \equiv L_h'$. Now a solution method can be outlined for calculating target function sensitivities

1. Solve equations (1) to (3) with a suitable numerical method to obtain the Temperature $T(\mathbf{x}, t)$.

2. Insert the calculated temperature in equations (14) and solve them with a similar method to obtain $\lambda(\mathbf{x}, t)$.

3. Insert the calculated values of λ and T into equation (16) in order to calculate the sensitivity $J_h'(\mathbf{x}, t)$. The sensitivity will then be used by a gradient based optimization method to minimize equation (4).

Due to the similarity of the direct problem and the adjoint problem the entire solver infrastructure can be reused. There is only a slight modification of the boundary conditions.

For the problem considered in this report it is assumed that the transient is defined by a set of time points $0 < t_1 < t_2 < \cdots < t_f$ for which h and T_g are defined. For intermediate time points linear interpolation is used. Also, the problem will be recast into correction factors c_h of the given values of the heat transfer coefficients. The only effect of this reformulation is that equation (16) changes to

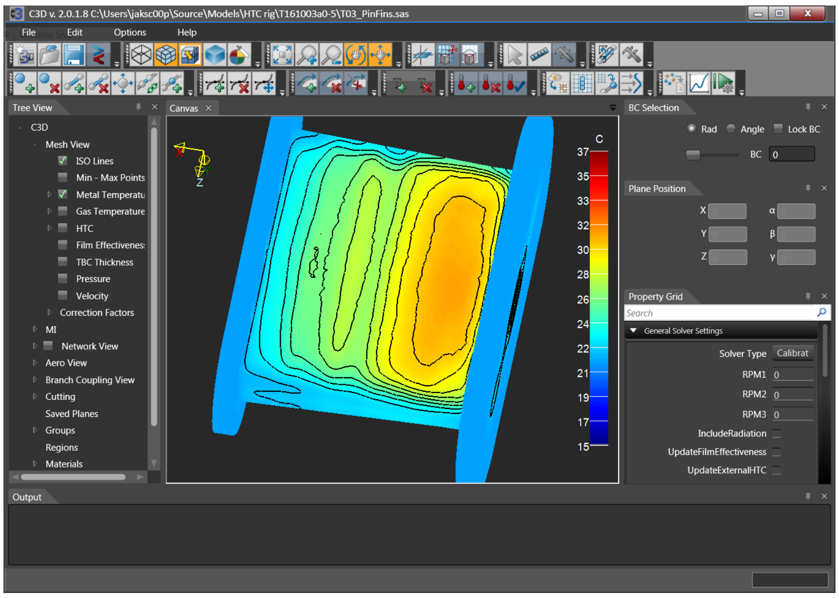
$$J'_{c_h} = -h\lambda(T - T_g) \tag{17}$$

With this definition the sensitivity problem can be rephrased as: "If the value of c_h is changed by an amount dc_h for a given mesh face at $\mathbf{x}_f$ at time point t_i, how much does this affect the target function (4)"? Changing c_h at time $t = t_i$ only affects the target function in the time intervals $[t_{i-1}, t_i]$ and $[t_i, t_{i+1}]$ because of the choice of linear interpolation. Writing the interpolated terms explicitly

$$\begin{aligned}
dJ_{c_h}(\mathbf{x}_f, t_i) = & \int_{t_{i-1}}^{t_i} -\left(h_{f,i-1} + (t - t_{i-1})\frac{h_{f,i} - h_{f,i-1}}{t_i - t_{i-1}}\right) \\
& \lambda\left(T - \left(T_{g,f,i-1} + (t - t_{i-1})\frac{T_{g,f,i} - T_{g,f,i-1}}{t_i - t_{i-1}}\right)\right) dc_h dt \\
& - \int_{t_i}^{t_{i+1}} \left(h_{f,i} + (t - t_i)\frac{h_{f,i+1} - h_{f,i}}{t_{i+1} - t_i}\right) \\
& \lambda\left(T - \left(T_{g,f,i} + (t - t_i)\frac{T_{g,f,i+1} - T_{g,f,i}}{t_{i+1} - t_i}\right)\right) dc_h dt
\end{aligned} \tag{18}$$

VALIDATION OF SENSITIVITY

The method, outlined in the previous section has been implemented in C3D, the in-house tool used at Siemens Industrial Turbomachinery AB for conjugate heat transfer analysis (Figure 1).



Solver Type Calibrat
RPM1 0
RPM2 0
RPM3 0
IncludeRadiation
UpdateFilmEffectiveness
UpdateExternalHTC

Figure 1: **C3D user interface.**

Both the direct and adjoint thermal problems are solved with FEM methods described in Ottosen and Petersson (1992) and Nikishkov (2010) The accuracy of the method is compared to a manually estimated sensitivity using first order finite differences for the object shown in Figure 2. The object consists of a single block of tetrahedral elements with dimensions 200x60x70 mm, for which the temperature is calculated between 0 and 10000 seconds. At $t = 0$ the temperature is uniformly 15 C in the block; the rightmost surface of the block has boundary conditions $h = 100\ W/(K * m^2)$ (uncorrected), a heat transfer correction factor of 1.5, and a gas temperature of 50 C. At $t = 10000$ the rightmost surface of the block has boundary conditions $h = 500\ W/(K * m^2)$ (uncorrected), a heat transfer correction factor of 2.0, and a gas temperature of 150 C. All other surfaces are adiabatic ($h = 0$). Temperature reference points are located on the leftmost surface in the model. The reference value is identical for all points, with a linear increase from 15 C at $t = 0$ to 70 C at $t = 10000$. The total area of the calibration region is 4116 mm^2.

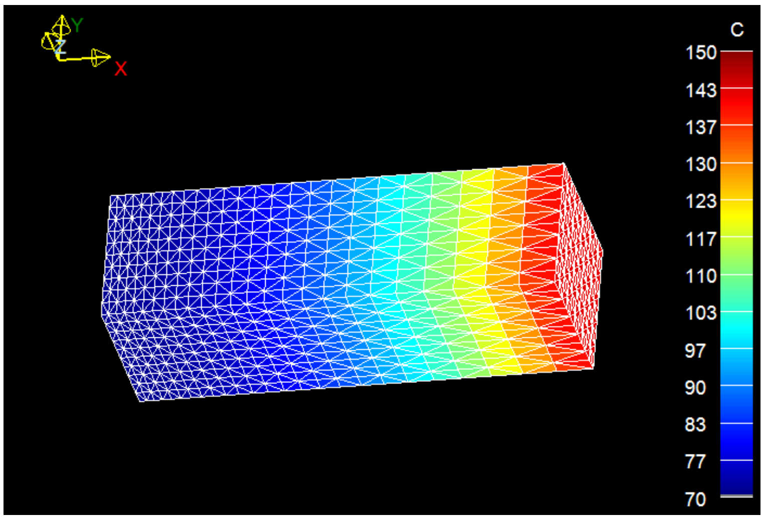


Figure 2: **Test object for sensitivity calculations.**

First, the sensitivity of the target function with respect to a change in heat transfer correction factor at $t = 0$ is evaluated manually. The transient temperature is calculated with a correction factor of 1.5 and exported to a spreadsheet. The squared difference between the calculated temperature and the linearly interpolated reference temperature is summed and weighted by the time increment and area. Next, The correction factor is increased to 1.51 and the procedure is repeated. The value of the sensitivity calculated in this way is $(1547.321 - 1550.987)/0.01 = -366.6$. The corresponding sensitivity, exported from C3D, is -377.11. The two values agree to 2.8 percent. The reason why the two values do not agree exactly is the finite resolution of the mesh. The temperature and sensitivity have different orders of magnitude and they propagate in different directions. Repeating the exercise for $t = 10000$ by increasing the correction factor value to 2.01 gives a sensitivity of $(1547.682 - 1550.987)/0.01 = -330.5$. The corresponding calculated sensitivity, exported from C3D, is -346.46. The two values agree to 4.6 percent. In conclusion, the method for calculating sensitivities with respect to the heat transfer correction factors is likely to be correctly implemented.

HEAT TRANSFER CALIBRATION

With a reliable method for calculating sensitivities the next step is to utilize this information in order to modify the heat transfer correction factors in such a way that the target function (4) is minimized with respect to the correction factors for all mesh faces and all (discrete) time points. Let c_h^i be the i:th iteration of the correction factors. The vector c_h is organized as $c_h = (c_{h,f_1,t_0}, \dots, c_{h,f_n,t_0}, \dots, c_{h,f_1,t_f}, \dots, c_{h,f_n,t_f})^t$. First, the correction factors for all mesh faces f_i are listed at time $t = t_0$. Next, the correction factors are listed for all mesh faces at time $t = t_1$, and so on. The simplest way to implement the minimization is the *steepest descent method*. The new values of the correction factors are then chosen as the old values plus a step length in the negative direction of the sensitivities

$$c_h^{i+1} = c_h^i + \alpha \nabla J \tag{19}$$

The gradient ∇J of the target function can be evaluated by repeated use of equation(18) for all calibration mesh faces and time steps. The step length α has to be chosen so that a large enough reduction of J is achieved. However, the problem with the steepest descent algorithm is that it tends to be very slow in practical applications. A more efficient method is the non-linear *conjugate gradient method* (see e.g. Nash and Sofer (1995)).

1. Calculate the gradient ∇J_i.
2. Calculate $\beta_i = \frac{\nabla J_i^t(\nabla J_i - \nabla J_{i-1})}{\nabla J_{i-1}^t \nabla J_{i-1}}$ with the Polak-Ribiere method.
3. Update the conjugate direction $p_i = \nabla J_i + \beta_i p_{i-1}$.
4. Search along the conjugate direction for a step length α_i that provides a sufficient reduction of the target function value $\alpha_i = \min_\alpha J(c_h^i + \alpha p_i)$.
5. Update the correction factors $c_h^{i+1} = c_h^i + \alpha_i p_i$.

In this paper a common variation is used where β is replaced by $\max(0, \beta)$ that has been empirically shown to perform better Shewchuk (1994). The line search method is implemented by starting with a small step length and successively increase it by a factor of 1.5 until the gradient of the target function starts to increase $\nabla J \cdot p > 0$. The conjugate gradient iteration can be terminated after a fixed number of iterations, or when the target function or its gradient has reached a certain threshold.

A very important observation here is that in order to evaluate ∇J only two transient solutions are needed; the direct problem and the adjoint problem. With a naive approach one solution of the direct problem would be needed for each design variable. A realistic problem may contain thousands of design variables.

STEADY STATE

The derivation of the adjoint equations for steady state problems follow the same approach as in the previous sections, dropping all time dependent terms. The direct steady state problem is formulated as

$$\nabla \cdot (k \nabla T) = 0 \qquad in \quad \Omega, \tag{20}$$

$$k \frac{\partial T}{\partial \mathbf{n}} = h(T - T_g) \qquad on \quad \Gamma \tag{21}$$

The target function is

$$J = \frac{1}{2}\int_{\Gamma} \left(T(\mathbf{x}) - T_m(\mathbf{x})\right)^2 d\Gamma \tag{22}$$

By repeating the derivation of the adjoint equations (equations(5) to (14))

$$\begin{aligned} \nabla \cdot (k\nabla\lambda) &= 0 \qquad in \quad \Omega \\ k\frac{\partial\lambda}{\partial\mathbf{n}} &= h\lambda - (T - T_m) \qquad on \quad \Gamma \end{aligned} \tag{23}$$

The sensitivity (17) stays the same.

VALIDATION OF CALIBRATION

In this section the previously outlined calibration method is validated on a real test piece. The transient thermal step response of the model in Figure 3 has been measured with an IR camera for a period of 7 s. The test object contains a set of pin fin banks. Hot air with temperature 106 degrees C is supplied from the right hand side in the figure. Figure 3 shows the difference between measured and calculated temperatures at $t = 7s$. On average, the temperature difference is between 10 and 15 degrees C.

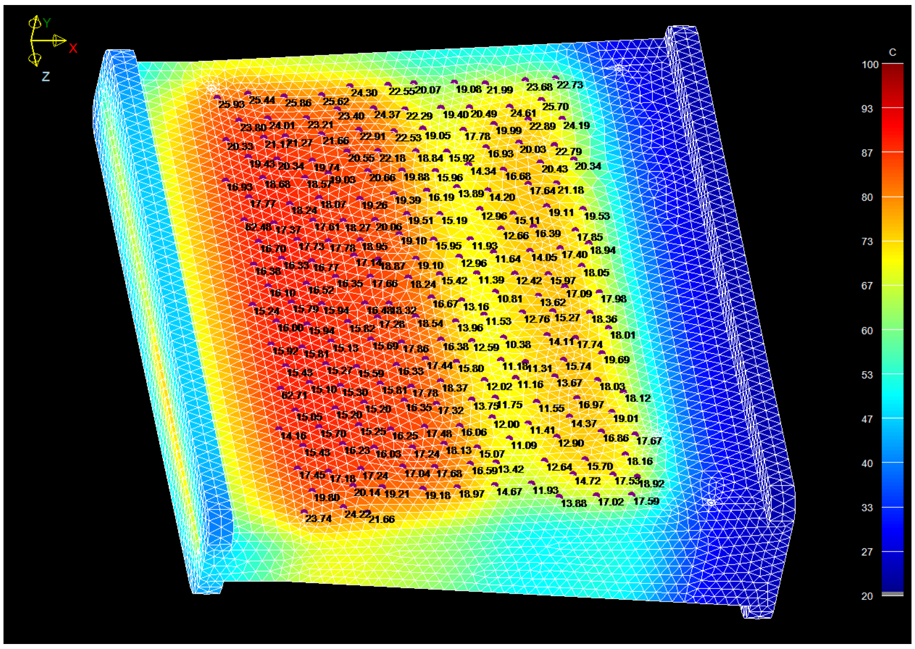


Figure 3: **Test object for calibration validation.**

In Figure 4 a number of passes have been performed with the calibration solver, as implemented in C3D. The temperature difference is now in the order of 1 - 2 degrees C. Similar tests have been performed for real engine components with very good results.

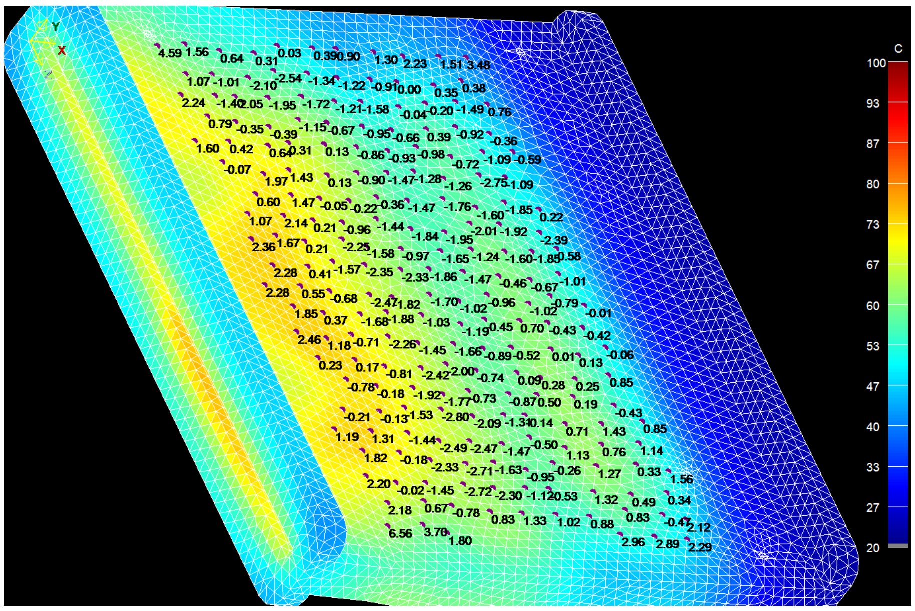


Figure 4: **Test object with calibrated temperatures.**

CONCLUSIONS

The adjoint method for sensitivity calculations described in this paper has been shown to agree well with finite difference calculations. The adjoint method and the method of transient thermal calibration, also described in this paper, have been implemented in C3D, the in-house tool used at Siemens Industrial Turbomachinery AB for conjugate heat transfer analysis. This implementation has been shown to perform transient thermal calibration efficiently on a real test piece for which thermal data has been obtained from IR camera measurements. User experience indicate a speedup of at least an order of magnitude compared to previous methods of thermal calibration used at Siemens Industrial Turbomachinery AB. Complex components such as turbine blades and vanes have been calibrated within a single working day.

References

Corliss, G. et al.
2002. *Automatic Differentiation of Algorithms*. New York: Springer Verlag.

Giannakoglou, K. C. and D. I. Papadimitriou
2008. *Adjoint Methods for Shape Optimization*. Berlin, Heidelberg: Springer Verlag.

Lions, J. L.
1971. *Optimal Control of Systems Governed by Partial Differential Equations*. Berlin: Springer Verlag.

Martins, J. A. et al.
2005. *A Coupled-Adjoint Method for High-Fidelity Aero-Structural Optimization*, volume 6. Springer Verlag.

Nash, S. G. and A. Sofer
1995. *Linear and Nonlinear Programming*. McGraw-Hill.

Neto, F. D. and A. J. Neto
2013. *An Introduction to Inverse Problems With Applications*. Berlin, Heidelberg: Springer Verlag.

Nikishkov, G. P.
2010. *Programming finite elements in Java*. London: Springer Verlag.

Ottosen, N. S. and H. Petersson
1992. *Introduction to the Finite Element Method*. Prentice Hall.

Shewchuk, J. R.
1994. An introduction to the conjugate gradient method without the agonizing pain.