INVERSE DESIGN METHODS FOR HIGH-TEMPERATURE SYSTEMS

Kyle Daun, Hakan Erturk, and John R. Howell*

Department of Mechanical Engineering
The University of Texas at Austin
Austin, Texas, USA

Address for Correspondence:
Department of Mechanical Engineering
The University of Texas at Austin
Austin, TX 78712-1063 USA
e-mail: jhowell@mail.utexas.edu

The Arabian Journal for Science and Engineering, Volume 27, Number 2C.
ABSTRACT

Inverse problems are those in which measured or specified information at a given location is used to infer the conditions at other locations that cause the measured or specified information. This paper discusses design of high temperature systems as an inverse problem, because the designer normally specifies the thermal conditions (temperature, heat flux) necessary to achieve a design goal, and wishes to find the necessary conditions at other locations to achieve that goal (heater locations, heater or burner energy input, best system geometry, etc.). Because the formulation of inverse problems often results in sets of equations that are mathematically ill-conditioned, special techniques are necessary to achieve useful solutions. For design problems, there are questions of solution existence, smoothness, and multiplicity that must be addressed, and these are discussed in the paper. Three classes of solution technique are discussed, and some examples of solutions are presented.

Keywords: Inverse, radiation heat transfer, design.
INVERSE DESIGN METHODS FOR HIGH-TEMPERATURE SYSTEMS

1. INTRODUCTION

High-temperature furnaces and heaters are used in practical applications from petroleum cracking through semiconductor wafer processing, food processing, curing and drying of coatings, materials processing, and many others. In all of these cases, the design goal is known. From the thermal viewpoint, this goal is usually to achieve a desired temperature distribution on an object (the design surface) along with the necessary energy required to maintain that temperature distribution. This energy is input from a heater surface. Designing new systems or redesigning existing systems based on changing conditions is usually based on experience and extrapolation of existing successful designs.

Computer models of these high-temperature systems are available. These models are designed to solve the sets of energy and radiative transfer equations that describe the system, and all require that the user provide some basic information to the model. This information usually includes the following:

- A complete description of the geometry of the system being modeled. This includes not only specification of the system boundaries, but also the placement of the heaters or burners on the heater surface within the enclosure.
- Specification of one thermal condition (prescribed temperature or heat flux) on every boundary, including the heater surface and the design surface.
- Prescription of the thermal and transport properties of all materials that are engaged in the heat transfer process.

In the initial design of a furnace or heater, it is usually one or more of the factors in this list that the designer wishes to find. For example, the designer may want to know the best geometry for the furnace that will achieve its goals, or the best placement of heaters or burners, or the necessary energy input to the heaters that will provide a given energy and temperature on the design surface, or the properties of materials that will attain the design goals. For such problems, the designer must resort to trial-and-error use of the commercial package. That is, the designer first uses experience to make a best-estimate of the unknown furnace or heater characteristics, and then uses the computer package to predict the system behavior. If the initial guess doesn’t satisfy the design requirements, then the assumed design is adjusted and the simulation of system behavior is again calculated. This procedure is continued until a satisfactory result is finally obtained.

An alternative approach is to specify the thermal conditions needed to achieve the desired product of the furnace or heater. This usually results in specifying both the temperature and heat flux on the design surface necessary to process the material passing through the furnace or heater (semiconductor wafer, petroleum feed, rate of foodstuff being processed, etc.) as shown for the bottom surface in Figure 1. For these cases, a desired temperature profile \( T(x_2) \) is generally known, and when the feed rate of material to be processed is included, the necessary energy rate per unit area \( q(x_2) \) to achieve this temperature profile is also known. The designer then wishes to find the necessary energy distribution to the heater surface, \( q(x_1) \) or the geometry of heaters with known energies. Mathematically, this type of specification requires two boundary conditions on the design surface (material being processed), while there is no specified boundary condition on the heater surfaces. This type of “inverse” problem specification is encountered in most design problems: the design goals are specified, and the designer seeks to find a design that will meet these goals. Unfortunately, this type of problem often results in a set of equations (the energy equation plus the radiative transfer equations) that is ill-conditioned. That is, standard solution techniques will fail, and predicted solutions may have non-physical characteristics (negative absolute temperatures, negative energy inputs to heaters or burners), very large oscillations, and other undesirable features that render the solution unusable. These characteristics are the reason that commercial codes for modeling heat transfer do not allow specification of inverse problems.

There are many similar problems in engineering and science, particularly in measurement and instrumentation, that are related to the inverse design problem. In this class of problems, measurements at a boundary are used to infer energy input or temperature at a remote boundary. In many inverse problems of this kind, conditions at a boundary are specified (design) or measured (instrumentation), and conditions at another boundary are to be inferred.
In this paper, we concentrate on the design problem in high-temperature systems where radiative transfer is important, and discuss methods that have proven successful in solution of inverse design problems and in application of inverse techniques to design and control of transient processes using distributed energy sources.

Design problems for thermal systems fall into three broad classes:

- **Boundary condition estimation problems**, where the heater surface energy and temperature must be found that satisfy the design surface conditions.
- **Inverse transient problems**, where imposed time-dependent conditions on the design surface must be satisfied by finding the necessary time- and space-dependent conditions on the heater surface.
- **Unknown geometry problems**, where the prescribed conditions on the design surface are satisfied by finding an enclosure geometry that will satisfy them.

There are three general methods for treating inverse problems, and they can be outlined as follows:

- **Trial-and-error techniques**, in which the designer uses experience and knowledge to refine initial provisional designs until a design is reached that satisfactorily produces the required conditions on the design surface.
- **Direct inversion techniques**, in which the equations that describe the energy transfer from the heater surface into the design surface are directly solved using available tools for treating ill-conditioned problems.
- **Optimization techniques**, which provide a systematic method for reaching an optimized solution by successively solving for conditions on the design surface using a series of improved designs that are found by application of one of many methods for approaching an optimum solution.

Each of the three solution methods can in principle be applied to each of the design classes.

### 1.1. Literature Survey

#### General Inverse Problems in Heat Transfer

Much literature exists that discusses inverse solutions of heat transfer problems. This literature developed chiefly from the need to understand and solve problems in measurement, and is especially devoted to conduction problems. For
example, if the transient surface temperature of a nosecone on a re-entry vehicle is to be measured, sensors must be placed on an interior surface, because the actual surface temperature to be measured is too high for the sensors to survive if placed there. Much research on this important class of problems is available, and is summarized in comprehensive texts [1–5].

More recently, boundary measurements have been used to infer flow conditions, and flow conditions have been measured to infer boundary conditions for systems with convective transfer [6–11].

As in conduction and convection, inverse analysis of radiative heat transfer has been implemented in support of property measurements [12–21] and in determining temperature and source distributions from boundary information [22–34].

Optimization techniques have been widely used to design many different thermal systems. They were first used to solve heat conduction problems; more recent examples include investment casting [35], heat treatment applications [36], and inverse measurement problems [37]. They have also been extended to treat convection problems, for example the design of heat exchangers [9, 10], electronic cooling applications [38], and turbine blades [11].

Optimization techniques have only recently been applied to solve radiant enclosure design problems. Federov et al. [39] and Daun et al. [40] used gradient-based methods to determine the ideal heater settings to achieve uniform temperature and heat flux distributions over the design surface. Daun et al. [41] also applied gradient-based methods to determine the optimal enclosure geometry to achieve a specified radiosity distribution over the design surface. Optimization techniques have also been used to design the geometry of solar collectors [42] and illumination applications [43].

1.2. Design of Systems with Thermal Radiation

The use of inverse analysis presents different challenges from the use of inverse methods in measurement applications. In the latter, a physically real solution must exist for the properties or source distributions that give the measured values at a boundary. Otherwise, experimentally measured values at the boundary would not exist. In design, however, the conditions at the boundary are stipulated, not measured. There is then no guarantee that any real and practical solution exists (heater energy distribution, heater locations or enclosure geometry) that will satisfy the stipulated conditions. On the other hand, it is entirely possible that multiple solutions exist that will satisfy the stipulated conditions, particularly if the designer is willing to relax the degree of precision required in meeting those conditions. This presents the designer with interesting possibilities.

Design of radiating systems has been approached in a number of papers [39–41, 44–72]. The literature through about 2000 is reviewed in considerable detail in Franca et al. [67], so the remainder of this paper will concentrate on recent research on design of high-temperature systems using inverse techniques.

2. SOLUTION BY INVERSE TECHNIQUES

The initial remarks about the concept of well or ill-posedness are attributed to Hadamard, who stated that a problem is ill-posed if it does not have a unique solution or the solution is not a continuous function of the data. This means that the solution is not stable or any small perturbation in the data results in significant changes in the solution. He thought that ill-posed problems are artificial and did not describe real physical systems. On the contrary, many physical systems are fundamentally ill-posed.

In fields of science and engineering, inverse problems arise whenever the input required to achieve a desired output in a physical system is to be estimated or a characteristic of the system is to be predicted from an observed or measured behavior. The former problem resembles the design discussed in this work and the latter is the measurement problem, which is quite similar to the design problem in most senses. Hansen [73] defines a general linear inverse problem as:

\[
\int_{\Psi} \text{Input} \times \text{System} \, d\Psi = \text{Output},
\]  

(1)
where the output is the available information and the input is the unknown that is to be estimated for a given system. Such equations representing the problem are known as the Fredholm equation of the first kind that can be written as a mathematical expression as:

\[ \int f(t)K(t,s)dt = g(s). \] (2)

Intuitively, one would expect that the problem could have numerous solutions as the kernel of the system, \( K(t,s) \) serves as a smoothing operator over the input function \( f(t) \) to achieve the output \( g(s) \). This makes it either difficult or impossible to estimate directly the appropriate input from the output.

Fredholm equations of the first kind are usually discretized to produce a numerical solution, giving an ill-conditioned linear system of equations, presented as \( A\cdot x = b \), results. The coefficient matrix, \( A \), is the discretized kernel that defines the system, where the right hand side vector \( b \) represents the available information (output) and \( x \) is the unknown (input).

The ill-conditioned nature of the system can be explained as follows: A system of linear equations can be interpreted as the linear mapping of vector space \( x \) to vector space \( b \) defined by the matrix \( A \). If \( A \) is singular, there exists a subspace of \( x \) called the null-space, which is mapped to zero rather than \( b \). If the vector \( b \) is in the range of such an \( A \) (the sub-space of \( b \) that can be mapped by \( A \)), the system will have an infinite number of solutions of linear combinations of the null-space.

In order to solve problems of this kind, the system must be regularized. Regularization is simply re-organizing the system so that the ill-conditioned part can be separated and ignored to achieve a stable, accurate, and physically reasonable solution. Before discussing regularization in more detail, it is necessary to define the system residual. For an analytical system, the residual can be defined as:

\[ r(s) = g(s) - \int f(t)K(t,s)dt, \] (3)

and the equivalent relation for the discretized equation is:

\[ r = b - A\cdot x. \] (4)

Regularization is usually performed following one of four different schemes.

1. The norm of the residual is minimized subject to the constraint that the solution belongs to a specified subset.
2. The norm of the residual is minimized subject to the constraint that a measure of the “size” of the solution is less than some specified upper bound.
3. The norm of the residual is minimized so that it is smaller than some specified convergence criteria.
4. A linear combination of the norm of the residual and the measure of the “size” of the solution is minimized.

All of these schemes require the use of a minimization technique such as the least squares minimization, Newton’s method or conjugate gradient method. The details about some of these minimization techniques are presented in the upcoming sections. For problems with complexities such as absorbing, emitting, and anisotropically scattering medium, or problems with non-linearities due to transient effects or multi-mode heat transfer, the calculation of sensitivity coefficients that are required by minimization techniques might not be feasible. For these problems, it is common to use techniques that iteratively find the solution by minimizing the residual or through some other decomposition technique. These techniques can be used to produce a set of solutions at different level of regularization. The designer may then select the “optimal” solution from that set that has acceptable accuracy and smoothness.

The so-called “L-curve” generally helps in this decision. The L-curve is constructed by plotting the norm of the residuals versus the norm of the solutions for every solution in the set. The magnitude of the residual norm represents the solution accuracy while the magnitude of the solution norm represents smoothness. Generally, as the solution norm decreases, the solution becomes smoother at the expense of solution accuracy, which in turn corresponds to an increase in the residual norm.
The typical trend of the $L$-curve for a highly ill-conditioned system is as follows: As the regularization level is increased a slight rise in the norm of the residual with a sharp decrease in the solution norm is observed. This trend slows down quite suddenly after a certain regularization level is reached. The $L$-curve has a corner at this point and later on further regularization greatly increases the norm of the residual with a slight decrease in the solution norm. The optimal solution usually lies near the corner of the $L$-curve where the solution norm is reduced with a small decrease in the norm of the residual.

Some of the most common regularized solution techniques are the truncated singular value decomposition (TSVD), the conjugate gradient method (CGM), and classical Tikhonov regularization. Of these, Tikhonov regularization follows the fourth scheme while TSVD is a decomposition technique that minimizes the residual. The conjugate gradient method is more flexible and can be used coupled with either one of these schemes.

2.1. Truncated Singular Value Decomposition (TSVD)

Truncated singular value decomposition is a solution technique based on singular value decomposition (SVD), a well-known method of decomposing a matrix into the product of three other matrices. If $A$ is an arbitrary $M \times N$ matrix, the singular value decomposition of $A$ is given by:

$$A = U \cdot S \cdot V^T,$$

where $U$ is an $M \times N$ orthogonal matrix, $S$ is a diagonal matrix of $N$ positive or zero singular values sorted in descending order so that $S_{1,1} \geq S_{2,2} \geq \ldots \geq 0$, and $V$ is an $M \times N$ orthogonal matrix. Besides providing the basis for a regularized solution method, SVD is also a very useful tool to analyze and describe the source of the ill-conditioned nature in the system.

Once SVD is performed, the inverse of $A$ can be defined as:

$$A^{-1} = V \cdot S' \cdot U^T,$$

where $S'$ is a diagonal matrix of $N$ elements of which the elements are multiplicative reciprocals of the singular values ($S'_{ii} = S_{ii}^{-1}$). A characteristic of a discrete ill-posed system is that the singular values decay gradually to zero. If the condition number (defined as the ratio of the largest and smallest singular values) is very large, the system can be termed to be ill-conditioned. For that reason, the direct inversion of such a matrix yields a solution that is dominated by high amplitude fluctuations caused by the very small singular values. This usually results in a nonphysical solution in a problem like ours. Therefore, the part of the system that causes this amplification should be removed from the system.

This is carried out by setting those elements in $S'$ that corresponding to the zero valued elements in $S$ equal to zero. Moreover, in practice, the singular values corresponding to other small singular values must also be truncated from the system to eliminate round-off errors in a numerical solution. Then using the truncated $S'$ and the resulting pseudo-inverse of matrix $A$, an approximate solution is found by:

$$x_n = \sum_{k=1}^{R} V_{n,k} \frac{b_m U_{m,k}}{S_{k,k}}, \quad n = 1, 2, \ldots, N,$$

where $R$ is the rank of $A$. Retaining different numbers of singular values yields different solutions to the system. This may present many alternative solutions, which is advantageous from the designer’s point of view. The most suitable of these solutions can be selected with the help of an $L$-curve as explained before. It is shown by Hansen [73] and Press et al. [74] that the solution given in Equation (7) is equivalent to a least squares solution that minimizes the residual of the system.

2.2. The Conjugate Gradient Method (CGM)

The conjugate gradient method is a general iterative minimization technique. The solution is defined as a linear combination of search directions, which are selected as conjugate vectors, $p$’s, that provide the maximum available
information about the function topography. Each step introduces the generation and addition of a new conjugate vector
to the solution of the previous step, providing a monotonic convergence of the objective function to a minimum.

In order to be able to solve a linear system of equation using the CGM, the algorithm presented in Beckman [75] can
be used. The algorithm provides the exact solution \( x_e \) to a linear system, \( Ax = b \), in \( N \) steps, where \( N \) is the number of
unknowns. For a system defined by a symmetric and positive-definite coefficient matrix, \( A \), the objective is to minimize
the functional, \( F(x) = [A(x_e - x)](x_e - x) \), which reaches zero at its minimum when \( x = x_e \). In order to minimize
the functional, the gradient (equivalent to the negative of twice the residual) should reach zero \( (\nabla F(x) = -2r = 0) \). The CGM
algorithm, presented here for symmetric and positive definite systems, can be generalized for any arbitrary system \( M \times N \)
by multiplying both sides of the equation by \( A^T \).

In the CGM presented by Beckman, the Gram–Schmidt orthogonalization technique is used for calculating the
conjugate vectors at each step. (The derivation of the method is presented in [75] in detail and is not repeated here.) The
generalized algorithm is presented in Figure 2.

As an \( N \) step solution technique for inverse problems, the solutions achieved at the end of every step can be considered
as unique, alternative solutions with different accuracy and smoothness characteristics. The \( N \) different solutions can
then be used to construct an \( L \)-curve in order to select the optimal solution.

This is a very simple and elegant method that uses only simple matrix vector multiplications. The small memory
required for storing intermediate results and parameters and computational economy makes it the method of choice for
large systems. It has robust convergence characteristics, each approximation being superior compared with previous
ones. In addition, the original matrix is stored and used without modification.

It should be noted that the CGM and TSVD algorithms are very similar methods that work based on the same
objective; differences in the solutions are due to the different methodologies and the numerical applications they follow.
Erturk et al. [72] present comparisons of solutions produced by both methods for a steady radiating enclosure containing
an absorbing, emitting, and anisotropically scattering medium.

---

**Figure 2. The conjugate gradient algorithm.**

1. \( x^1 = 0 \)
2. \( r^1 = b - Ax^1 = b \)
3. \( p^1 = A^T r^1 \)
4. \( k = 1 \ldots N \)
   i. \( \alpha^k = \frac{|A^T r^k|^2}{|Ap^k|^2} \)
   ii. \( x^{k+1} = x^k + \alpha^k p^k \)
   iii. \( r^{k+1} = r^k - \alpha^k Ap^k \)
   iv. \( \beta^k = \frac{|A^T r^{k+1}|^2}{|A^T r^k|^2} \)
   v. \( p^{k+1} = A^T r^{k+1} - \beta^k p^k \)
2.3. Tikhonov Regularization

Tikhonov’s regularization scheme follows the fourth approach listed earlier, which is minimizing a linear combination of the norm of the residual and the measure of the “size” of the solution. The objective function to be minimized is:

\[ F(x) = \left[ \|Ax - b\| \right]^2 + \sum_{i=1}^{p} \alpha_i^2 \left[ \|L_i x\| \right]^2, \]  

for a \( p \)th order scheme, where \( L_i \) approximates the \( i \)th derivative operator and \( \alpha_i \) is the \( i \)th order regularization parameter. Using a small regularization parameter will result in an accurate solution since the emphasis is placed on minimizing the norm of the residual, while using a large regularization parameter will result in a solution with improved smoothness characteristics at the expense of accuracy. In order to minimize the \( F(x) \), the function must satisfy the condition:

\[ \frac{dF}{dx} = 0 = 2A^T (Ax - b) + 2 \sum_{i=1}^{p} \alpha_i^2 L_i^T L_i x. \]  

For the standard (or zeroth) order Tikhonov regularization scheme, \( p = 0 \) and \( L \), becomes the identity matrix \( I \), leading to the modified set of linear equations:

\[ (A^T A + \alpha_0^2 I) x = A^T b. \]  

Similarly, for a second-order regularization method for which \( p = 2 \) and \( \alpha_0 = \alpha_1 = 0 \), the linear system is equal to:

\[ (A^T A + \alpha_2^2 L_1^T L_2) x = A^T b. \]  

As in the cases of other regularization techniques using the correct regularization parameter is very important to achieve an optimal solution that is smooth and accurate enough for the designer’s needs. A number of solutions with different regularization levels can be used to construct an \( L \)-curve, and the required regularization parameter can be selected accordingly.

3. TREATMENT BY OPTIMIZATION TECHNIQUES

Optimization methods work by solving the well-posed “forward” design problem through successive iteration. Unlike the forward “trial-and-error” design methodology, which relies solely on the designer’s experience and intuition, the optimization methodology uses sophisticated numerical algorithms to adjust the design configuration at each iteration until the optimum design is found. In this way, the number of iterations and consequently the time required to design the enclosure is reduced, and the final solution quality is usually much better than that obtained by the forward design methodology.

The first step of the optimization process is to define an objective function, \( F(\Phi) \), which quantifies the “goodness” of a particular design configuration, in such a way that the minimum of \( F(\Phi) \) corresponds to the optimal design outcome. The objective function is dependent on a set of variables contained in \( \Phi \), called design parameters, which completely specify the design configuration. The goal, then, is to identify the set of design parameters that minimize \( F(\Phi) \):

\[ F(\Phi^*) = \min F(\Phi), \quad \Phi \in \mathbb{R}^n. \]  

Often, it is also necessary to impose design constraints on \( \Phi \) of the form:

\[ c_i(\Phi) = 0, \quad i = 1, \ldots, m' \]
\[ c_i(\Phi) \geq 0, \quad i = m' + 1, \ldots, m, \]
which define the domain of $\Phi$ in $n$-space, called the feasible region. For example, consider the enclosure shown in Figure 3. The objective of this design problem is to find the heater settings and enclosure geometry that produces a specified radiative heat flux distribution, $q_s^{\text{target}}(u)$, over the design surface. A suitable objective function for this class of problem is the variance of the heat flux distribution calculated using a particular set of design parameters and the desired heat flux distribution, evaluated at discrete points over the design surface,

$$F(\Phi) = \frac{1}{b-a} \sum_{j=a}^{b} \left[ q_{sj}(\Phi) - q_{sj}^{\text{target}} \right]^2,$$

with the design parameters in $\Phi$ specifying the heater settings and enclosure geometry. The heat flux distribution over the design surface that best matches the desired distribution is produced by the design configuration corresponding to $\Phi^*$, which in turn is found by minimizing the objective function defined in Equation (14). Design constraints could also be imposed to limit the size of the enclosure, ensure the enclosure remains unobstructed (so all pairs of points on the enclosure surface can “see” each other), and to prevent the heat flux distribution over the heater surface from assuming negative values.

3.1. Objective Function Minimization

A large number of techniques have been developed to minimize objective functions. These techniques are chosen based on the type of optimization problem, the number of design parameters, the types of constraints, and the objective function topography. Minimization techniques can be broadly separated into two groups. The first group consists of techniques that rely on random sampling of the objective function over the entire feasible region; they are well suited for solving combinatorial “traveling-salesman” type problems, as well as for identifying the global minimum of continuous objective functions that have many local minima over the feasible region. Examples of this group include simulated annealing and genetic algorithms.
The second group consists of techniques that work by minimizing the objective function based on the local topography of $F(\Phi)$. These methods, called gradient-based methods, are most often used to minimize continuously-differentiable objective functions having few local minima. Since the objective functions associated with radiant enclosure optimization tend to fall into this category, gradient-based methods have been almost exclusively used to design radiant enclosures. Accordingly, further discussion of optimization techniques in this paper is limited to the gradient-based methods that have been applied to solve this type of problem.

Gradient-based methods all work according to the same iterative scheme; at the $k^{th}$ iteration, a search direction, $p_k$, is chosen based on the local curvature of $F(\Phi_k)$. Next, a step size $\alpha_k$ is chosen, usually by performing a univariate “line” minimization of $F(\Phi_k + \alpha_k p_k)$ using Newton–Raphson, bisection, or golden section techniques. Another possibility is to use a diminishing step size, $\alpha_k = \alpha_0/k^n$, which is done when the number of objective function evaluations is restricted by computational expense. Once the step size and search direction are chosen, a “step” is taken in the $p_k$ direction,

$$\Phi_{k+1} = \Phi_k + p_k \alpha_k .$$

This process is repeated until the set of design parameters, $\Phi^*$, that correspond to a local minimum of the objective function, $F(\Phi^*)$, is identified.

Gradient-based methods differ on how $p_k$ is chosen. (In fact, this is how they are named.) Almost all methods use the first-order curvature information contained in the gradient vector,

$$g(\Phi) = \left[ \frac{\partial F(\Phi)}{\partial \Phi_1}, \frac{\partial F(\Phi)}{\partial \Phi_2}, \ldots, \frac{\partial F(\Phi)}{\partial \Phi_n} \right].$$

Some methods also use the second-order curvature information contained within the Hessian matrix,

$$H(\Phi) = \begin{bmatrix} \frac{\partial^2 F}{\partial \Phi_1^2} & \frac{\partial^2 F}{\partial \Phi_1 \partial \Phi_2} & \cdots & \frac{\partial^2 F}{\partial \Phi_1 \partial \Phi_n} \\ \frac{\partial^2 F}{\partial \Phi_2 \partial \Phi_1} & \frac{\partial^2 F}{\partial \Phi_2^2} & \cdots & \frac{\partial^2 F}{\partial \Phi_2 \partial \Phi_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 F}{\partial \Phi_n \partial \Phi_1} & \frac{\partial^2 F}{\partial \Phi_n \partial \Phi_2} & \cdots & \frac{\partial^2 F}{\partial \Phi_n^2} \end{bmatrix} .$$

Steepest-descent is the simplest and most intuitive of the gradient-based methods. In this method, the search direction in Equation (15) is set equal to:

$$p_k = -\frac{g(\Phi_k)}{\left\| g(\Phi_k) \right\|} ,$$

which is the direction of steepest-descent. Although this method is very simple, it has poor convergence characteristics, since the search direction is chosen based only on first-order curvature information. It can be shown that in the vicinity of $\Phi^*$, this method has a convergence of $O[1]$.

In Newton’s method, the rate of convergence to $\Phi^*$ is greatly improved by using both first- and second-order curvature information to calculate $p_k$. The method is based on the second-order Taylor-series expansion of the objective function. First, let an arbitrary set of design parameters $\Phi_k$ be separated from the local minimum at $\Phi^*$ by a vector $c_k$, so that $\Phi^* = \Phi_k + c_k$. The value of the objective function at the local minimum is then estimated by:
The Arabian Journal for Science and Engineering, Volume 27, Number 2C. December 2002

\[ F(\Phi^*) = F(\Phi_k + c_k) \approx F(\Phi_k) + c_k \cdot g(\Phi_k) + \frac{1}{2} c_k \cdot H(\Phi_k) \cdot c_k. \]  

(19)

If the objective function is a quadratic (which is usually a reasonable assumption if \( \Phi_k \) is near \( \Phi^* \)), then the Hessian matrix can be assumed independent of \( \Phi \). The gradient vector at \( \Phi^* \) can be estimated by taking the derivative of Equation (19) with respect to \( \Phi \),

\[ g(\Phi^*) = g(\Phi_k + c_k) \approx g(\Phi_k) + c_k \cdot H(\Phi_k). \]

(20)

Since all the elements in \( g(\Phi^*) \) are equal to zero, Equation (20) simplifies to:

\[ H(\Phi_k) c_k = -g(\Phi_k). \]

(21)

In Newton’s method, \( p_k \) is set equal to the direction of \( c_k \), which is called Newton’s direction. This method has a much greater order of convergence than the steepest descent model (\( O[2] \) compared with \( O[1] \)), and thus requires fewer iterations to identify \( \Phi^* \). It is important to note, however, that since extra CPU time is required at each iteration to calculate \( H(\Phi) \), Newton’s method can sometimes require more computational effort to calculate \( \Phi^* \) than steepest descent. This is especially true if analytical derivatives of \( F(\Phi) \) are unavailable; in this case, the first- and second-order derivatives are usually approximated by a forward finite difference method. Using this method, the gradient vector is calculated using at least \( n + 1 \) objective function evaluations, while the Hessian matrix requires at least \( n(2n-1) \) evaluations. Clearly, Newton’s method is unsuitable unless the second-order sensitivities can be found in an efficient way.

The quasi-Newton method avoids the computational effort required to generate \( H(\Phi) \) by approximating the second-order sensitivities using the first-order curvature information “built-up” from previous iterations. The search direction \( p_k \) is set equal to the direction of \( c_k \), which in turn is found by solving the equation:

\[ B_k c_k = -g(\Phi_k), \]

(22)

where \( B_k \) is an approximation of \( H(\Phi_k) \) at the \( k \)th iteration. The initial Hessian approximation \( B_0 \) is initially set equal to the identity matrix, so \( c_0 \) is the direction of steepest-descent. At each subsequent iteration, however, the approximation of the Hessian matrix is improved by adding an update matrix, \( U_k \),

\[ B_{k+1} = B_k + U_k, \]

(23)

where \( U_k \) is found from the objective functions and gradient vectors of previous iterations. There are many different variations of the quasi-Newton’s method; the most common is the Broyden–Fletcher–Goldfarb–Shanno (BFGS) scheme, which uses an update matrix defined by:

\[ U_k = \frac{s_k s_k^T}{s_k y_k^T} - \frac{B_k y_k y_k^T B_k}{y_k B_k y_k^T}, \]

(24)

where \( s_k = \Phi_{k+1} - \Phi_k \) and \( y_k = g(\Phi_{k+1}) - g(\Phi_k) \).

Since \( B_k \) becomes an accurate estimate of \( H(\Phi_k) \) only after several iterations, the quasi-Newton method requires more iterations than the Newton’s method to find \( \Phi^* \). Nevertheless, the quasi-Newton method requires less computational effort to perform the minimization less since the calculation of second-order sensitivities is avoided.

These three methods are demonstrated by applying them to minimize Rosenbrock’s function,

\[ F(\Phi) = 100(\Phi_2 - \Phi_1^2)^2 + (1 - \Phi_1)^2, \]

(25)
which has a global minimum $F(\Phi^*) = 0$ at $\Phi^* = \{1, 1\}^T$. Rosenbrock’s function is commonly used as a test case because the long, narrow valley leading up to $\Phi^*$ is a challenge for most gradient-based methods. In this demonstration, each of the three methods is started at $\Phi_0 = \{-1, 1\}^T$ and it is allowed to proceed towards $\Phi^*$ for a maximum of 1000 iterations. The solution paths for the three methods are shown in Figure 4. The Newton and quasi-Newton schemes required 9 and 21 iterations, respectively, to converge to a value of $F(\Phi^*) < 10^{-6}$. The steepest descent became “stuck” along the valley leading up to the global minimum and was unable to identify $\Phi^*$ in fewer than 1000 iterations.

The order of convergence of the different methods is demonstrated by plotting the value of the objective function at a given iteration versus the iteration number on a log–log scale, as shown in Figure 5. Newton’s method converges to the local minimum at a much greater rate than the steepest descent method, because Newton’s direction is a much better search direction than the direction of steepest descent. The quasi-Newton’s method starts out with the same convergence rate as the steepest-descent, since $p_k$ is initially set equal to the direction of steepest-descent; the rate of convergence quickly approaches that of Newton’s method, however, as the accuracy of the Hessian approximation improves as the search direction approaches Newton’s direction.

![Figure 4. Solution paths for the minimization of Rosenbrock’s Function, Equation (25). (a) Steepest-descent, (b) Quasi-Newton, and (c) Newton methods.](image)

![Figure 5. Plot of $F(\Phi_k)$ versus $k$ for steepest-descent, quasi-Newton, and Newton’s method.](image)
4. APPLICATIONS TO ENGINEERING DESIGN

4.1. Transient Problems

The objective of a process furnace is to provide energy to an engineering process such as annealing, curing of paint or surface coating, industrial baking, and manufacturing through chemical deposition applications. The process takes place as long as the design environment satisfies specific thermal conditions and in many applications the quality of the product depends on how accurately the thermal conditions are satisfied. Often, the necessary design conditions are prescribed as a desired temperature distribution over the design surface. For a transient system where design must be considered over a specific time interval, the net heat flux on the design surface provides an additional constraint due to the thermal capacities. Two conditions are thus specified for the design surface while no condition is prescribed for the heater surfaces.

4.2. Inverse Design of Transient Heater Input

The transient heating problem in a radiating enclosure is very challenging because transient, combined mode, heat transfer is highly non-linear. The governing relations include the radiative transfer equation, an integral equation that has absolute temperature to the fourth power, together with the transient energy equation, which is a partial differential equation in terms of absolute temperature.

The main difference between transient and steady-state problems is that in the steady-state problem the objective is to satisfy a certain temperature and heat flux distribution over the design environment, while in the transient problem the goal is to enforce a specified temperature schedule. Due to the change in heat capacity based on the temperature history, the objective at any given time is not much different than it is in the steady problem, because the goal is still to achieve a prescribed heat flux distribution combined with a prescribed temperature distribution. It is then possible to formulate the transient problem as a series of steady-state problems solved sequentially that are dependent on each other.

Consider the discretized energy equation on a design surface element:

$$\sum_{j=1}^{N_h} E_{n}^{j} A_{j} \mathcal{Z}_{ji} = \rho_{t} c_{p} \delta_{t} A_{i} \frac{T_{i}^{n+1} - T_{i}^{n}}{\Delta t} - Q_{cd,i}^{n} - Q_{cv,i}^{n} - \sum_{k=1}^{N_i} E_{n}^{k} A_{k} \mathcal{Z}_{ki} + E_{n}^{i} A_{i}, \quad (26)$$

where $\mathcal{Z}_{ji}$ is the exchange factor between the sub-element $j$ and sub-element $i$ that denotes the rate of energy emitted by element $j$ that is absorbed by element $i$. The $Q_{cd}^{n}$ and $Q_{cv}^{n}$ terms represent the conductive and convective rate of heat transferred to element $i$ at the $n^{th}$ time step. The summation over $N_h$ denotes the elements with unknown temperatures for the $n^{th}$ time step and the summation over $N_i$ denotes the elements with known temperatures in Equation (26). The form of energy equation presented here is discretized in time using an explicit Euler scheme; in Equation (26) all the terms on the right hand side are the known terms, while the terms on the left hand side are the unknowns. The $T_{i}^{n+1}$ term of the design surface is the value defined by the objective function or the temperature history that is followed at $(n+1)^{th}$ time step.

With known exchange factors, this equation is a discretized Fredholm equation of the first kind. When written for all design surface elements, they constitute an ill-conditioned set of equations and in order to achieve a reasonable solution, a regularized system must be considered.

By using the explicit Euler discretization scheme, the resulting system of equations becomes linear, keeping all the conductive, convective, and heat capacitance terms that generate the non-linearity in the right-hand side vector of the equation. If another discretization scheme is preferred such as an implicit Euler scheme or an Adams–Bashforth scheme, the resulting system of equations will have unknown terms both in absolute temperatures and absolute temperature to power four.

Once the temperature distribution at the $n^{th}$ time step is known over all surfaces but the heaters, Equation (27) can be solved for the required temperature distribution in the heaters at the $n^{th}$ time step. This is followed by the calculation of the required power input for the heaters:
\[
Q_j^{n-1} = \rho_j c_p \delta_j A_j \frac{T_j^n - T_j^{n-1}}{\Delta t} - Q_{cd,j}^{n-1} - Q_{cv,j}^{n-1} - \sum_{i=1}^{N} E_i^{n-1} A_i \mathcal{A}_{ij} + E_j^{n-1} A_j .
\] (27)

The convective and diffusive terms in Equation (27) are all in the \((n-1)\)st time step, for which the temperatures are known.

It should be noted that based on the desired temperature distribution, chosen temperature history, and the accuracy of the solutions from previous time steps, it is quite possible for the system to demand removal of heat from the system. This can only be accomplished by having negative power to some of the heaters. Such conditions are not acceptable, as adding and removing heat at the same time is not desired.

In such a situation, the simplest solution is to turn off heaters that demand removal of energy, or in other words set power input equal to zero \((Q_j = 0)\). This produces errors in the temperature distribution over the design surface. The temperature distribution achieved now will be:

\[
T_j^{n} = T_j^{n-1} + \frac{\Delta t}{\rho_j c_p \delta_j A_j} \left( Q_{cd,j}^{n-1} + Q_{cv,j}^{n-1} + \sum_{i=1}^{N} E_i^{n-1} A_i \mathcal{A}_{ij} - E_j^{n-1} A_j \right) .
\] (28)

Once the temperature distribution of the heaters is known for the \(n\)th time step, the net heat flux over all surfaces other than heaters and therefore the resulting temperature distribution at the \((n+1)\)st step can be calculated using:

\[
T_j^{n+1} = T_j^{n} + \frac{\Delta t}{\rho_j c_p \delta_j A_j} \left( Q_j^{n} + Q_{cv,j}^{n} + \sum_{i=1}^{N} E_i^{n} A_i \mathcal{A}_{ij} - E_j^{n} A_j \right) .
\] (29)

The procedure described is applied from the initial time step, where the design, reflector, and the heater surfaces are all at a prescribed initial condition to the final temperature as the surface is heated following the desired temperature history. An outline of the procedure is presented in terms of a flowchart in Figure 6.

![Flowchart](Figure 6. The flowchart of the solution algorithm.)
The proposed design methodology was applied to a sample design problem in [62]. It is a transient thermal boundary condition estimation problem, in which the designer aims at controlling the thermal conditions in the design environment by setting the necessary thermal conditions on the heater surfaces. The design surface is to be heated from an initial state to a final steady state, while it is kept at a spatially uniform temperature, following a specified heating profile.

A two-dimensional evacuated furnace with eight straight surfaces, which are all gray, with heater surfaces 5, 6, and 7, design surface 2 and reflector surfaces 1, 3, 4, and 8 is shown in Figure 7. The heater surfaces are oxidized-cast iron with a rough surface leading to diffusely reflecting surfaces. The reflector surfaces are polished aluminum, and are assumed to be specularly reflecting with a larger reflectivity than the heater and design surfaces. The design surface is silicon carbide, which is diffusely reflecting. The heater surfaces are made up of 30 strip heaters each of uniform temperature that are slightly separated from each other. The assumptions and further detailed information is in [68].

The design surface is to be heated according to the history presented in Figure 8 and the geometric data and required thermal properties of the surfaces are presented in Table 1.

Figure 9 displays the error of the achieved temperature distribution along the design surface throughout the process based on the proposed algorithm. The error is defined as:

\[
\%E_T = 100 \frac{|T_i - T_d(t)|}{T_d(t)}.
\]  

(30)

<table>
<thead>
<tr>
<th>Surface</th>
<th>( \delta ) (m)</th>
<th>( \rho ) (kg/m(^3))</th>
<th>( c_p ) (J/kgK)</th>
<th>( L ) (m)</th>
<th>( k ) (W/mK)</th>
<th>( \varepsilon )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.001</td>
<td>2702</td>
<td>903</td>
<td>0.8</td>
<td>227</td>
<td>0.05</td>
</tr>
<tr>
<td>2</td>
<td>0.001</td>
<td>3160</td>
<td>675</td>
<td>1</td>
<td>454</td>
<td>0.9</td>
</tr>
<tr>
<td>3</td>
<td>0.001</td>
<td>2702</td>
<td>903</td>
<td>0.8</td>
<td>227</td>
<td>0.05</td>
</tr>
<tr>
<td>4</td>
<td>0.001</td>
<td>2702</td>
<td>903</td>
<td>0.4</td>
<td>227</td>
<td>0.05</td>
</tr>
<tr>
<td>5</td>
<td>0.0001</td>
<td>7870</td>
<td>447</td>
<td>1</td>
<td></td>
<td>0.95</td>
</tr>
<tr>
<td>6</td>
<td>0.0001</td>
<td>7870</td>
<td>447</td>
<td>1</td>
<td></td>
<td>0.95</td>
</tr>
<tr>
<td>7</td>
<td>0.0001</td>
<td>7870</td>
<td>447</td>
<td>1</td>
<td></td>
<td>0.95</td>
</tr>
<tr>
<td>8</td>
<td>0.001</td>
<td>2702</td>
<td>903</td>
<td>0.4</td>
<td>227</td>
<td>0.05</td>
</tr>
</tbody>
</table>
Figure 8. The objective function that defines the desired design surface temperature history for the transient heating in a two-dimensional enclosure problem.

Figure 9. The errors based on design surface temperature and design specification.
From Figure 9, the accumulated errors affect the system the most at 27 seconds \( t/60 \approx 0.45 \), reaching a maximum error of 0.65 percent, where the average error is 0.1 percent. The maximum and average errors decrease to 0.16 and 0.07 percent, respectively, as the system reaches the steady state. The solution accuracy is still limited by the assumptions considered for the formulation, property values, and the radiation solver used. The locus of the maximum error is usually at the tips of the design surface.

The conditions on the design surface are achieved when the temperature in the heater strip surfaces satisfies the values displayed in Figure 8. The temperature of the heater strips is slightly above the temperature of the design surface at all times so that they can radiatively heat the design surface to follow the design objective as specified by \( T_d(t) \). The temperature of the heater strips increases until it peaks around 39 seconds at a value near 1100 K and then decreases to the steady state temperature of the design surface.

The 30 heater strips exchange radiative energy with other surfaces and each other, and receive energy as an input. In order to control the temperature distribution of the heater strips that in turn controls the temperature distribution on the design surface, the input power distribution is varied. The necessary power input to maintain the temperatures of the strip heaters (shown in Figure 10) is displayed in Figure 11. In order to obtain the distribution presented in Figure 11 an additional constraint is used for the initial time step. For the problem under consideration, the required heat flux on the design surface is zero with all initial design and reflector surface temperatures being 300 K. It is apparent that a uniform distribution of 300 K, identical to the initial condition, along 30 heater strips will satisfy the design constraints exactly. Due to regularization error, however, it is not possible to achieve a predicted uniform profile over the heaters. Consequently, some heaters require a high amount of power at the initial time step especially when very small time step sizes are used. Therefore, a second constraint is applied for the initial time step to remove the “artificial” peak in the power input that has nothing to do with the physical needs of the system. This constraint is simply not turning the power on for all heater strip elements for the initial time step leading the heater strips not to change their temperature.

All the results presented in Figures 9–11 are obtained using a time step size of \( \Delta t = 0.023 \) seconds and 64 surface elements with a \( 26 \times 10 \) grid resolution. The exchange factors are calculated by Monte Carlo method using 400 million sample photon bundles. The result from the first CG-step was used throughout the transient in order to obtain physical solutions with smooth spatial and time profiles. It was observed that identical solutions result when the grid resolution is doubled to 128 surface elements with a \( 52 \times 20 \) grid. Although the accuracy of the inverse solutions could have been
improved by using the results of further CG-steps, this increases the solution norm as displayed in the $L$-curve. The $L$-curve represents the variation of the solution residual norm with the solution norm for a single time step. A sample $L$-curve from the solution presented in Figures 9–11 is displayed for the 665$^{th}$ time step in Figure 12. Every point in Figure 12 represents a possible solution resulting at the end of each CG-step for the 665$^{th}$ time step. While the decrease in the residual norm represents the increase in solution accuracy, the corresponding increase in the solution norm represents the decrease in the smoothness of the solution. Therefore, the solutions produced by performing further CG-steps are more accurate but are more likely to be non-physical than the solutions generated by earlier CG-steps.

![Figure 12. The $L$-curve, variation of the norm of the solution with norm of the residual for time step 665.](image)

To investigate the effect of the CG-step used in the solution of the sample design problem, the maximum percentage error of solutions produced using different CG-steps and time step sizes are compared in Figure 13. The time step size is an important parameter that affects the level of accuracy in the solution. The discretization error in Equations (26–29) dominates the solution when a large time step is used. Therefore, for time steps larger than 3.76 seconds the system at some point overheats the design surface due to discretization errors in such an amount that satisfying the design goal becomes numerically impossible using positive emissive power on the heater elements. As a result, no physical result is available. On the other hand, when the time step is decreased, the total number of time steps used for the solution increases. This results in increasing the cumulative effect of regularization errors, leading the system to a state where it is also impossible to maintain a physical solution. Therefore, the algorithm cannot produce any solution when the time step is reduced below some value.

The range of time steps that produce a physically meaningful solution differs for solutions that use the results of a different number of CG-steps. This range is from 3.76 to 0.023, 3.76 to 0.3, and 3.76 to 0.019 seconds when the results of the first, second, and third CG-step are used, respectively. The solutions that use the results of the first and the third CG-steps have a larger allowable time step size range than the solution that uses the result of the second CG-step. The solution evaluated using the result of the first CG-step is the smoothest; thus, even though it is affected most by the cumulative effect of the regularization error as the number of time steps used increases, it still can produce physically
reasonable solutions up to some point. On the other hand, the solution evaluated using the result of the third CG-step is subject to the least regularization error among the three solutions, and the cumulative effect of the regularization error starts affecting the system later than the other two although it produces the least smooth solutions. The solution using the results of the second CG-step satisfies neither of the conditions, and as a result has the smallest range of useful time increment size. Thus, as the overall solution accuracy increases with decreasing norm of the residual, the smoothness decreases with the use of more CG-steps. This is a basic trade-off in most inverse problems, and the choice of the CGM step is dependent on the process and the problem considered.

Solutions for modified cases with different geometric configurations or properties are presented in [68]. A solution for a similar problem where the object that is to be heated is moving along a conveyor belt is available in [66].

![Figure 13. The change in solution accuracy with the result of CG-step used and time step size.](image)

4.3. Inverse Control

When a real system designed as in the sample problem of the previous section is built, some discrepancies will result between the design from the numerical model and the real case no matter how accurate the model. The design and control engineer must overcome the discrepancies at some level. A control algorithm is developed in [65] that makes use of the information provided by the inverse design to train an artificial neural network (ANN), which then may be used to control the real system.

In this review, the goal is not to give detailed information about improvements or developments in the state-of-art of the ANN's. Nevertheless, it may be helpful to introduce ANN's to the reader without giving details available in the literature [76] before explaining the control algorithm developed.

Artificial neural networks are a relatively new information processing technique that has its basis in simulating living nerve systems. They are basically parallel processing architectures where the knowledge is represented in terms of weights between input and output layers. A simple neural net structure is shown in Figure 14, which has $N$ layers, $R$ inputs, and $R$ outputs. As indicated in Figure 15, the relation between the inputs $I$ and outputs $O$ is represented in terms of the biases, $b$, and the transfer functions, $f$, of the layers and the weights, $W$, between the nodes of the adjacent
layers. The most commonly used transfer functions are the pure linear function \( f = x \) and the sigmoid functions; the so-called “logsig” \( f = 1/(1+e^{-x}) \) and “tansig” \( f = 1-1/(1+x) \) and \( x < 0 \), \( f = 1-1/(1-x) \) functions. The main significance of the ANN’s when compared with traditional programming is the ability of the network to learn through a training process. By training, the ANN is taught to adjust the weights and biases to follow or repeat what it has learned. The weights and biases are modified so that the average squared error between the target outputs and the network outputs is minimized. In order to perform this minimization, optimization methods are employed.

![Diagram of an N-layered artificial neural network architecture.](image)

**Figure 14.** An N-layered artificial neural network architecture.

![Diagram of Step 2: Training of an ANN.](image)

**Figure 15.** Step 2: Training of an ANN.

When the ANN faces an input for the first time it simply estimates the output for the input based on the experience it has, or the training. If the input lies within the input data range it has trained with, it is probable that the estimation will be relatively successful. On the other hand, the accuracy and the reliability for extrapolated inputs are always in question.

The developed algorithm consisted of three steps:

1. The numerical solution of the design problem;
2. Training of the ANN;
3. The use of ANN as the control device for the “physical system”.

The first step in developing the control algorithm is the design step, which is explained in the previous section. Once the design is completed, the designed system is used to gather required data and train an ANN. This is accomplished by using the designed system as a model of the physical system behavior, because the design was developed for measuring the system response to the defined system. However, due to the approximations inherent to the inverse model and the
uncertainties in the property data used, there will be a certain level of discrepancy between the estimations of the numerical model and the actual conditions achieved on the design surface of the physical system. A complete set of training data consists of the temperature distribution on the design surface for the current and next time step, $T_d'(x, t)$ and $T_d'(x, t+\Delta t)$, and the required power input for the current time step, $Q(x, t)$, along the entire transient, as shown in Figure 15. At the end of the training, the ANN can be used to simulate the physical system, as long as an adequate architecture and training method are used. The accuracy of the simulation with ANN is primarily limited by the interpolation ability of the trained ANN.

Once the ANN is trained, it can be used to control the physical system as shown in Figure 16. As the inputs will include the objective function for the temperature at the next time step, the ANN estimates the required heater input, $Q'(x, t)$, based on the data obtained from the response of the physical system. Therefore, with the use of an adequately trained ANN, the resulting temperature distribution achieved over the design surface, $T_d''(x, t)$, will be more accurate than the distribution, $T_d'(x, t)$ found using only the inputs estimated by the numerical design. Further details about the developed algorithm are presented in [71].

The algorithm was applied to the sample problem of the previous section. Instead of using a real physical system and a numerical approximation of the system to develop the control algorithm, a numerical model is used with two different sets of data to illustrate the method of developing a control algorithm. The numerical model based on the data set explained in the previous section represented the physical system and a numerical model based on a slightly “erroneous” dataset represents the mathematical model. Moreover, as a statistical solution technique (the Monte Carlo method) is used to model the thermal radiation for both cases, a statistical variation is introduced to the solutions. This situation is similar to using a real physical system for the application, as a consistent discrepancy representing the errors due to properties used or approximations considered simulates the statistical variations inherent in physical measurements.

In the less exact mathematical model, the specularly reflecting surfaces (surfaces 1, 3, 4, and 8 in Figure 7) are treated as diffuse and a surface emissivity of 0.075 is used rather than 0.05. The required heater input is calculated with the inverse design approach with an accuracy of 1.2 percent error, where error is calculated using Equation (30). When the calculated input is used together with the physical system (the model that uses specularly reflecting walls with an emissivity of 0.05) the temperature distribution over the design surface differs from the objective function up to 6.9 percent.

In order to decrease this error, the proposed ANN algorithm is applied. The training data consist of one set of input and target data for every time step in the inverse design. Each input set consists of emissive power values at a time $t$ gathered at three points (two at the edges and one in the middle) on the design surface of the system with the accurate numerical model together with the achieved emissive power on the design surface for the same points at time $t + \Delta t$. Therefore, the 159 input vectors, one for each time step, of 6 elements are matched with the 30 power inputs applied to the system at time $t$, leading to 159 target vectors of 30 elements.

The algorithm was used iteratively rather than at a single time. After the first iteration is performed, the training data were expanded with additional data sets gathered during step 3. Then the combined set can be used for training the neural net and the step 3 can be performed. Each iteration step expands the training data sets by the number of time steps the simulation uses in step 3. A two-layered ANN is used with the transfer functions for the first and second layers
being “tansig” and pure linear, respectively. The first layer has 6 nodes while the second has 30 nodes. Then the maximum error was reduced to 1.18 percent at the end of six iterations as displayed in Figure 17, where the power input displayed in Figure 18 is required. As can be observed from Figure 19 the locus of the maximum error has moved to the middle of the design surface from the edges. The further details about the algorithm and solution are available in [71].

![Graph showing convergence of the proposed control algorithm.](image)

**Figure 17.** The convergence of the proposed control algorithm.

![3D graphs showing required radiant heater power input and error in temperature on design surface at iteration 7.](image)

**Figure 18.** Required radiant heater power input. **Figure 19.** Error in temperature on design surface at iteration 7.

### 5. OPTIMIZATION OF RADIANT ENCLOSURES BY NON-LINEAR PROGRAMMING

As discussed previously, a common objective in the design of radiant enclosures is to identify the heater settings and enclosure geometry that produces the desired heat flux distribution over the design surface. This is done by minimizing an objective function of the form:
where \( q_{sj}(\Phi) \) and \( q_{sj}^{\text{target}} \) are the realized and desired heat flux distributions evaluated at discrete locations over the design surface, and \( \Phi \) contains the design parameters that specify the enclosure geometry and heat flux distributions over the heater surface.

In order to evaluate Equation (31), it is necessary to calculate the heat flux distribution over the design surface. It is also desirable to determine the sensitivity of the heat flux distribution to each design parameter: this information is needed to generate the gradient vector and Hessian matrix, which are used to choose search directions during the minimization process. For diffuse-walled enclosures, the heat flux distribution and sensitivities are calculated using the infinitesimal-area method presented by Daun and Hollands [77], while the heat flux distribution in enclosures containing specular surfaces is solved through a Monte Carlo analysis. The analysis methods for both types of enclosures are discussed in the following sections, and are then demonstrated by applying them to design the geometry and heater settings of 2-D radiant enclosures.

5.1. Diffuse-Walled Enclosures

5.1.1. Solution of the Primal Problem

The heat flux distribution over the design surface is found by first solving the radiosity distribution, using the infinitesimal-area method [77]. Once the radiosity distribution is known, the heat flux distribution is quickly found by post-processing.

The first step of this method is to represent the enclosure geometry parametrically,

\[
\mathbf{r} = C(u, \Phi) = \left[ P(u, \Phi), Q(u, \Phi) \right], \quad 0 \leq u \leq 1. \quad (32)
\]

By allowing \( u \) to range over its entire domain, the position vector \( \mathbf{r} \) carves out the shape and extent of the enclosure surface, as shown in Figure 20. The surface emissivity, \( \varepsilon(u) \), is also represented parametrically, and either the temperature, \( T(u, \Phi) \), or the heat flux, \( q_s(u, \Phi) \) is specified at every point on the enclosure surface. Note that the enclosure geometry, heat flux, and temperature are functions of \( \Phi \), while the emissivity is not. This greatly simplifies the sensitivity calculations, as will be demonstrated later.

Once the enclosure is represented parametrically, it can be shown that the radiosity at any point on the enclosure surface is governed by a Fredholm integral equation of the second kind,

\[
q_u(u, \Phi) = b(u, \Phi) + \int_0^1 q_s(u', \Phi) K(u, u', \Phi) \, du', \quad (33)
\]

Figure 20. Parametric representation of the radiant enclosure.
where, if \( u \) lies on a surface where \( E_b(u, \Phi) = \sigma T^4(u, \Phi) \) is known,
\[
b(u, \Phi) = \varepsilon(u) E_b(u, \Phi), \quad g(u) = 1 - \varepsilon(u),
\]
(34)
or, if \( u \) lies on a surface where \( q_s(u, \Phi) \) is known,
\[
b(u, \Phi) = q_s(u, \Phi), \quad g(u) = 1.
\]
(35)

The kernel of Equation (33), \( K(u, u', \Phi) \), represents the differential view factor from \( u \) to an infinitesimal strip element at \( u' \) divided by \( du' \), and is given by:
\[
K(u, u', \Phi) = \beta(u, u', \Phi) \frac{n(u, \Phi) \cdot e(u, u', \Phi)}{s(u, u', \Phi)} J(u, \Phi),
\]
(36)

where \( n(u, \Phi) \) is the unit normal vector at \( u \), \( e(u, u', \Phi) \) is a unit vector at \( u \) in the direction of \( u' \), \( s(u, u', \Phi) \) is a vector connecting \( u \) and \( u' \), and the surface Jacobean, \( J(u, \Phi) \) scales an infinitesimal element in parametric space to an infinitesimal area element on the enclosure surface,
\[
dA_{wp}(u, \Phi) = J(u, \Phi) \ du'.
\]
(37)

The blockage factor, \( \beta(u, u', \Phi) \), is a binary term that is equal to zero if the path between \( u \) and \( u' \) is obstructed by an intervening surface, and is otherwise equal to unity. (This term can be excluded from the kernel if the enclosure remains unobstructed throughout the optimization process.)

The parametric representation of the enclosure cross-section and the geometric quantities used to form \( K(u, u', \Phi) \) are shown in Figure 21. All of the terms in Equation (36) can be found directly from the parametric representation in Equation (32) except for the blockage factor, which must be found either by analytical geometry or through ray-tracing.

Integral equations rarely have tractable analytical solutions, so Equation (33) must be solved analytically. First, the parametric domain of \( u \) is discretized into \( N \) elements, the \( i^{th} \) element centered at \( u_i \) having a width of \( \Delta u_i \). Each segment

![Figure 21. Geometric quantities used to form \( k(u, u', \Phi) \).](image-url)
in parametric space corresponds to a strip element on the enclosure surface of infinite length and finite width, as shown in Figure 22. If the radiosity is assumed uniform over each of these strips, Equation (33) can be approximated by:

$$q_{oi}(\Phi) = b_i(\Phi) + g_i \sum_{j=1}^{N} \beta_{ij}(\Phi) q_{oj}(\Phi) \ dF_{i \text{-strip}}(\Phi),$$  \hspace{1cm} (38)

where $q_{oi}(\Phi) = q_o(u_i, \Phi)$, $b_i(\Phi) = b(u_i, \Phi)$, $g_i = g(u_i)$, $\beta_{ij}(\Phi) = \beta(u_i, u_j, \Phi)$, and $dF_{i \text{-strip}}(\Phi)$ is the view factor from $u_i$ to the strip element centered at $u_j$, given by:

$$dF_{i \text{-strip}} = \frac{1}{2} \left[ \frac{s(u_i, u_j - \Delta u_j/2, \Phi) \times n(u_i, \Phi)}{|s(u_i, u_j - \Delta u_j/2, \Phi)|} - \frac{s(u_i, u_j + \Delta u_j/2, \Phi) \times n(u_i, \Phi)}{|s(u_i, u_j + \Delta u_j/2, \Phi)|} \right].$$  \hspace{1cm} (39)

The radiosity distribution is then found by performing a matrix inversion, $b = A^{-1} x$, or some equivalent operation. Finally, the heat flux distribution is solved by post-processing the radiosity solution:

$$q_{oi}(\Phi) = q_{oi}(\Phi) - \sum_{j=1}^{N} q_{oj}(\Phi) \beta_{ij}(\Phi) \ dF_{i \text{-strip}}(\Phi).$$  \hspace{1cm} (42)
The heat flux sensitivities are calculated from the radiosity sensitivities. The first-order radiosity sensitivities are calculated by differentiating Equation (33) with respect to the relevant design parameter,

$$\frac{\partial q_o(u, \Phi)}{\partial \Phi_p} = \frac{\partial b(u, \Phi)}{\partial \Phi_p} + g(u) \int_0^1 \frac{\partial q_o(u', \Phi)}{\partial \Phi_p} K(u, u', \Phi) du'. \quad (43)$$

Since the boundaries of the parametric domain are constant, the derivative in Equation (43) can be brought into the integral. Assuming that the radiosity distribution has already been solved for, the result is another Fredholm integral equation of the second kind,

$$\frac{\partial q_o(u, \Phi)}{\partial \Phi_p} = \frac{\partial b(u, \Phi)}{\partial \Phi_p} + g(u) \int_0^1 \frac{\partial q_o(u', \Phi)}{\partial \Phi_p} K(u, u', \Phi) du' + \frac{1}{\partial \Phi_p} \int_0^1 q_o(u', \Phi) \frac{\partial K(u, u', \Phi)}{\partial \Phi_p} du'. \quad (44)$$

where $\frac{\partial q_o(u, \Phi)}{\partial \Phi_p}$ is the only unknown.

Again, this integral equation does not have a tractable analytical solution, so $\frac{\partial q_o(u, \Phi)}{\partial \Phi_p}$ must be solved for using numerical techniques. First, the parametric domain of $u$ is discretized into $N$ elements using the same quadrature as before. Again, by assuming that $q_o(u, \Phi)$ and $\frac{\partial q_o(u, \Phi)}{\partial \Phi_p}$ are uniform over each element, Equation (44) can be approximated by:

$$\frac{\partial q_o(\Phi)}{\partial \Phi_p} = b'_1(\Phi) + g_i \sum_{j=1}^{N} \beta_{ij}(\Phi) \frac{\partial q_o(\Phi)}{\partial \Phi_p} dF_{1-\text{strip}}(\Phi), \quad (45)$$

where

$$b'_1(\Phi) = \frac{\partial b(u, \Phi)}{\partial \Phi_p} + g_i \sum_{j=1}^{N} q_{oj}(\Phi) \frac{\partial \beta_{ij}(\Phi)}{\partial \Phi_p} dF_{1-\text{strip}}(\Phi). \quad (46)$$

(If the enclosure geometry is being optimized, it is necessary to force $\beta_{ij}(\Phi) dF_{1-\text{strip}}(\Phi)$ to be continuous by ensuring that the enclosure remains unobstructed.)

Writing Equation (45) for all elements results in a system of $N$ simultaneous equations containing $N$ unknowns, which is rewritten as a matrix equation:

$$A(\Phi) \textbf{x}'(\Phi) = b'(\Phi) \quad (47)$$

where $\textbf{x}' = \left\{ \frac{\partial q_{o1}}{\partial \Phi_p}, \frac{\partial q_{o2}}{\partial \Phi_p}, ..., \frac{\partial q_{oN}}{\partial \Phi_p} \right\}^T$, $b' = \left\{ b'_1, b'_2, ..., b'_N \right\}^T$, and $A$ is defined in Equation (41). If $A$ has already been inverted to solve the radiosity distribution, then the solution of the first-order sensitivities requires little computational effort. Finally, the first-order heat flux sensitivities are found by taking the derivative of Equation (42),

$$\frac{\partial q_o(\Phi)}{\partial \Phi_p} = \frac{\partial q_o(\Phi)}{\partial \Phi_p} - \sum_{j=1}^{N} \left\{ \frac{\partial q_{oj}(\Phi)}{\partial \Phi_p} \beta_{ij}(\Phi) dF_{1-\text{strip}}(\Phi) + q_{oj}(\Phi) \frac{\partial \beta_{ij}(\Phi)}{\partial \Phi_p} dF_{1-\text{strip}}(\Phi) \right\}. \quad (48)$$

The second-order sensitivities are found in a similar way. Differentiating the radiosity equation twice with respect to $\Phi_p$ and $\Phi_q$ results in:
which again is a Fredholm integral equation of the 2nd kind, assuming that the radiosity distribution and first-order sensitivities have already been solved for. By discretizing the parametric domain using the same quadrature as before, and assuming that the radiosity, first-order sensitivities, and second-order sensitivities are constant over each surface element, Equation (49) can be approximated by:

\[
\frac{\partial^2 q_{ij}(\Phi)}{\partial \Phi_p \partial \Phi_q} = \frac{\partial^2 b(u, \Phi)}{\partial \Phi_p \partial \Phi_q} + g(u) \left[ \frac{\partial^2 q_o(u', \Phi)}{\partial \Phi_p \partial \Phi_q} K(u, u', \Phi) + \frac{\partial q_o(u', \Phi)}{\partial \Phi_p} \frac{\partial K(u, u', \Phi)}{\partial \Phi_q} \right] du',
\]

Equation (49) can be approximated by:

\[
\frac{\partial^2 q_{ij}(\Phi)}{\partial \Phi_p \partial \Phi_q} = b^*(\Phi) + g_i \sum_{j=1}^{N} \beta_{ij}(\Phi) \frac{\partial^2 q_{ij}(\Phi)}{\partial \Phi_p \partial \Phi_q} dF_{i\text{-}strip}(\Phi),
\]

where

\[
b^*_i(\Phi) = \frac{\partial^2 b_i(\Phi)}{\partial \Phi_p \partial \Phi_q} + g_i \sum_{j=1}^{N} q_{ij}(\Phi) \beta_{ij}(\Phi) \frac{\partial^2 [dF_{i\text{-}strip}(\Phi)]}{\partial \Phi_p \partial \Phi_q} + \frac{\partial q_{ij}(\Phi)}{\partial \Phi_p} \beta_{ij}(\Phi) \frac{\partial [dF_{i\text{-}strip}(\Phi)]}{\partial \Phi_q} + \frac{\partial q_{ij}(\Phi)}{\partial \Phi_q} \beta_{ij}(\Phi) \frac{\partial [dF_{i\text{-}strip}(\Phi)]}{\partial \Phi_p}.
\]

Writing Equation (50) for every element results in a system of \(N\) linear equations containing \(N\) unknowns, which is rearranged into the matrix equation:

\[
A(\Phi)x^*(\Phi) = b^*(\Phi),
\]

where \(A\) is defined by Equation (41), \(b^*=[b^*_1, b^*_2, \ldots, b^*_N]^T\), and \(x^*=[\partial^2 q_{11}/\partial \Phi_p \partial \Phi_q, \partial^2 q_{12}/\partial \Phi_p \partial \Phi_q, \ldots, \partial^2 q_{NN}/\partial \Phi_p \partial \Phi_q]^T\).

Again, if \(A\) has already been inverted to solve for the radiosity distribution, the second-order radiosity sensitivities are found with little additional computational effort. Finally, the second-order heat flux sensitivities are found by differentiating Equation (42) with respect to the two design parameters of interest,

\[
\frac{\partial^2 q_{ij}(\Phi)}{\partial \Phi_p \partial \Phi_q} = \frac{\partial^2 q_{ij}(\Phi)}{\partial \Phi_p \partial \Phi_q} - \sum_{j=1}^{N} \left[ \frac{\partial^2 q_{ij}(\Phi)}{\partial \Phi_p \partial \Phi_q} \beta_{ij}(\Phi) \frac{dF_{i\text{-}strip}(\Phi)}{\partial \Phi_p} + \frac{\partial q_{ij}(\Phi)}{\partial \Phi_p} \beta_{ij}(\Phi) \frac{dF_{i\text{-}strip}(\Phi)}{\partial \Phi_q} \right]
\]

\[
+ \frac{\partial q_{ij}(\Phi)}{\partial \Phi_p} \beta_{ij}(\Phi) \frac{dF_{i\text{-}strip}(\Phi)}{\partial \Phi_q} + q_{ij}(\Phi) \beta_{ij}(\Phi) \frac{dF_{i\text{-}strip}(\Phi)}{\partial \Phi_p} \beta_{ij}(\Phi) \frac{dF_{i\text{-}strip}(\Phi)}{\partial \Phi_q}.
\]

5.1.2. Optimization Procedure

In order to apply gradient-based optimization techniques, it is necessary to calculate the first- and second-order derivatives of the objective function, which in turn are derived from the heat flux sensitivities. In most problems, this can only be done by performing a finite-difference approximation, which can be both inaccurate and computationally demanding. In this class of problem, however, the analytical solutions of the gradient vector and Hessian matrix are found efficiently via the first- and second-order heat flux sensitivities.
The elements of the gradient vector are calculated by substituting \( q_{sj} \) and \( \partial q_{sj} / \partial \Phi_p \) into the derivative of the objective function defined in Equation (31) with respect to \( \Phi_p \),

\[
g_p(\Phi) = \frac{\partial F(\Phi)}{\partial \Phi_p} = \frac{2}{b-a} \sum_{j=a}^{b} \left[ q_{sj}(\Phi) - q_{sj}^{\text{target}} \right] \frac{\partial q_{sj}(\Phi)}{\partial \Phi_p}. \tag{54}
\]

Likewise, the elements in the Hessian matrix are calculated by substituting \( q_{sj} \), \( \partial q_{sj} / \partial \Phi_p \), and \( \partial^2 q_{sj} / \partial \Phi_p \partial \Phi_q \) into the derivative of the objective function with respect to \( \Phi_p \) and \( \Phi_q \),

\[
H_{pq}(\Phi) = \frac{\partial^2 F(\Phi)}{\partial \Phi_p \partial \Phi_q} = \frac{2}{b-a} \sum_{j=a}^{b} \left[ q_{sj}(\Phi) - q_{sj}^{\text{target}} \right] \frac{\partial^2 q_{sj}(\Phi)}{\partial \Phi_p \partial \Phi_q} - \frac{\partial q_{sj}(\Phi)}{\partial \Phi_p} \frac{\partial q_{sj}(\Phi)}{\partial \Phi_q}. \tag{55}
\]

5.1.3. Demonstration of Method

The technique described in the previous sections is applied to solve two different types of design problems involving radiant enclosures; in both cases, the objective is to produce the desired boundary conditions over the design surface. In the first problem, the desired radiosity distribution is realized by changing the enclosure geometry, while the goal of the second problem is to produce the desired heat flux distribution over the design surface by changing the heat flux distribution over the heater surfaces.

The first problem is shown in Figure 23. The enclosure consists of four sides: one heater surface, two adiabatic surfaces, and one temperature-specified surface. The heater surface has a heat flux of \( q_{s1} = 0.5 \text{ W/m}^2 \), while the temperature-specified surface has an emissive power of \( E_{b4} = 1 \text{ W/m}^2 \). The adiabatic surfaces function to deflect thermal radiation emitted by the heater surface onto the temperature-specified surface. The surface emissivities are set equal to unity except on the temperature-specified surface, where \( \varepsilon_4 = 0.5 \). The top adiabatic surface is represented by a quadratic B-spline curve and the design parameters are the \( y \)-coordinates of the left and center control points, denoted by \( \Phi_1 \) and \( \Phi_2 \), respectively. The height of the left adiabatic surface is also controlled by \( \Phi_1 \).

![Figure 23. Diffuse-walled enclosure design problem: geometric optimization.](image-url)
The design objective is to obtain a linear radiosity distribution over the center of the design surface, given by:

\[ q_{o_1}^{\text{target}}(u) = 1.32 + 0.32u, \ 0.8 \leq u \leq 0.95. \]  

(56)

This is done by minimizing an objective function similar to Equation (31),

\[ F(\Phi) = \frac{1}{b-a} \sum_{j=a}^{b} \left[ q_{o_1}(\Phi) - q_{o_1}^{\text{target}}(\Phi) \right]^2, \]  

(57)

subject to the constraints:

\[ 0.5 \leq \Phi_1 \leq 8 \]
\[ (\Phi_1 + 1)/2 \leq \Phi_2 \leq 8. \]  

(58)

The constraints limit the enclosure size and also ensure that it remains unobstructed throughout the minimization process, so that \( \beta(u,u',\Phi) \) is always equal to unity and can be excluded from the sensitivity calculations. These constraints are enforced using a gradient-projection method. At each iteration, \( \Phi_k \) is checked to see if it lies within the feasible region; if it doesn’t, than a new set of design parameters, \( \Phi_k' \), is found by projecting \( \Phi_k \) onto the boundary of the feasible region so that \( (\Phi_k - \Phi_k')^2 \) is minimized.

Figure 24 shows the topography of the objective function over the feasible region. Two local minima exist; one is near \( \Phi^* = [4.5, 6.5]^T \), while the other is close to the boundary, \( \Phi^* = [0.5, 0.75]^T \). The objective function defined in Equation (57) was minimized using steepest-descent, quasi-Newton, and Newton methods; the solution paths of these three methods are shown in Figure 25. All three methods started from an initial position of \( \Phi_0 = [0.5, 8]^T \) and converged to the local minimum at \( \Phi^* = [4.704, 6.559]^T \). The enclosure geometries that correspond to these design parameters are shown in Figure 26. The analysis was performed in a personal computer with a Pentium III™ 600 MHz processor and 233 MB of RAM.

The performance of the three methods is compared in Table 2. The steepest descent required 23 steps to identify the local minimum, while the Newton and quasi-Newton required 7 and 10 steps, respectively. Because the first- and second-order sensitivities are found efficiently by post-processing the radiosity solution, the calculation of the Hessian matrix requires very little additional effort. Accordingly, the total CPU time required to minimize the objective function is largely dependent on the number of steps. The steepest-descent requires the most steps, and the most CPU time. The quasi-Newton method requires three more steps than the Newton method, and slightly more CPU time.

Figure 24. Topography of \( F(\Phi) \), defined in Equation (57).
Figure 25. Solution paths used by the steepest-descent, quasi-Newton, and Newton methods to minimize $F(\Phi)$, defined in Equation (57).

Figure 26. Enclosure geometries corresponding to the initial set of design parameters, $\Phi_0 = \{0.5, 8\}^T$, and the optimum set of design parameters, $\Phi^* = \{4.704, 6.559\}^T$.

Table 2. Performance of Gradient-Based Techniques Used to Minimize $F(\Phi)$, Defined in Equation (57).

<table>
<thead>
<tr>
<th>Method</th>
<th>Steps</th>
<th>CPU Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steepest-Descent</td>
<td>23</td>
<td>41</td>
</tr>
<tr>
<td>Quasi-Newton</td>
<td>10</td>
<td>16</td>
</tr>
<tr>
<td>Newton</td>
<td>7</td>
<td>15</td>
</tr>
</tbody>
</table>

The second design problem is shown in Figure 27. The goal of this problem is to produce a uniform heat flux distribution of $q_{target}(u) = -2$ W/m$^2$ over the design surface using 24 heaters located on three heater surfaces. Since the heat flux is uniform over each heater, and because the problem is symmetric, only twelve design parameters are required to fully specify the heat flux distribution over the heater surface. Also, because the enclosure geometry is constant throughout the minimization process, the derivatives of $\beta_{ij}$ and $dF_{i,straj}$ with respect to $\Phi$ are equal to zero and can be excluded from the sensitivity calculations.
The heater settings that produce the desired heat flux distribution are found by minimizing the objective function, defined by:

$$F(\Phi) = \frac{1}{b-a} \sum_{j=a}^{b} \left[ q_{ij}(\Phi) - q_{ij}^{\text{target}} \right]^2.$$  \hspace{1cm} (59)

The minimization of $F(\Phi)$ is carried out using an unconstrained Newton method, starting from $\Phi_{0,i} = 1\text{W/m}^2$, $i = 1\ldots12$. Three iterations were required to identify a local minimum of the objective function, $F(\Phi^*) = 2.92 \times 10^{-8}$, corresponding to the solution illustrated in Figure 28 and the design parameters in Table 3. The heat flux distribution realized over the design surface closely matches the desired distribution over the design surface, as shown in Figure 30. Nevertheless, the solution is non-physical because $E_b(u)$ assumes a negative value over some regions of the heater surface. This is a consequence of a violation of the 2nd law of thermodynamics, which is not enforced by the governing radiosity equation.

A more useful solution is found by adding a term to the original objective function equal to the $L_2$ norm of the design parameters, defining a new “penalized” objective function,

$$F_p(\Phi) = \gamma F(\Phi) + (1-\gamma) \sum_{j=1}^{12} \Phi_j^2.$$  \hspace{1cm} (60)

This new objective function was derived based on the observation that, when solving a design problem using inverse analysis, non-physical solutions usually arise from large heat flux variations over the heater surfaces. Accordingly, the penalty term in Equation (60) acts to “regularize” the solution in a way similar to the regularization methods used in inverse analysis. The value of $\gamma$ was set equal to 0.9999, and Newton’s method was again used to minimize the objective function starting from the same initial set of design parameters. Three iterations were required to identify a local minimum of $F_p(\Phi^*_p) = 3.41 \times 10^{-4}$, which corresponds to $F(\Phi^*) = 3.25 \times 10^{-4}$. The heat flux and emissive power distributions corresponding to this solution are illustrated in Figure 29, and the design parameters are included in Table 3. The heat flux distribution realized over the design surface is shown in Figure 30; the maximum difference between $q_s(u, \Phi^*_p)$ and $q_{ij}^{\text{target}}(u)$ is 5.13 % on the edge of the design surface, which is within the tolerances demanded by most engineering applications.
Table 3: Optimal Heater Settings for the Penalized and Non-Penalized Objective Functions.

<table>
<thead>
<tr>
<th>( \Phi^* )</th>
<th>( \Phi_{p}^* )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Phi_1 )</td>
<td>1.33</td>
</tr>
<tr>
<td>( \Phi_2 )</td>
<td>2.11</td>
</tr>
<tr>
<td>( \Phi_3 )</td>
<td>2.29</td>
</tr>
<tr>
<td>( \Phi_4 )</td>
<td>-3.56</td>
</tr>
<tr>
<td>( \Phi_5 )</td>
<td>2.50</td>
</tr>
<tr>
<td>( \Phi_6 )</td>
<td>-10.18</td>
</tr>
<tr>
<td>( \Phi_7 )</td>
<td>-71.25</td>
</tr>
<tr>
<td>( \Phi_8 )</td>
<td>122.8</td>
</tr>
<tr>
<td>( \Phi_9 )</td>
<td>69.94</td>
</tr>
<tr>
<td>( \Phi_{10} )</td>
<td>-100.0</td>
</tr>
<tr>
<td>( \Phi_{11} )</td>
<td>0.10</td>
</tr>
<tr>
<td>( \Phi_{12} )</td>
<td>-3.89</td>
</tr>
</tbody>
</table>

Figure 29. Heater settings found by minimizing \( F_p(\Phi) \) defined in Equation (60).

Figure 30. Distributions of \( q_s(u, \Phi^*) \) and \( q_s(u, \Phi_{p}^*) \) over the design surface, found by minimizing the penalized and non-penalized objective functions.

5.2. Enclosures Containing Specular Surfaces

5.2.1. Solution of the Primal Problem

In this problem, the heat flux distribution is solved using exchange factors calculated by the Monte Carlo method. The Monte Carlo method has been used extensively to solve many different radiant enclosure problems, and is especially well suited for analyzing enclosures containing specularly-reflecting surfaces. Although the Monte Carlo method has been described extensively in the literature [78, 79], it is presented again here to better demonstrate challenges inherent in optimizing a stochastic system.

The first step of this method is to discretize the enclosure surface into \( N \) elements, with the \( i \)th element having an area of \( \Delta A_i \). As in the previous case, the emissivity \( \varepsilon_i \) is known over each element, and either the heat flux, \( q_{si} \), or the temperature, \( T_i \), is specified. Assuming a uniform temperature or heat flux distribution over the element, performing an energy balance over the \( i \)th element results in the equation:

\[
q_{si}(\Phi) + \sum_{j=1}^{N} \varepsilon_j E_{bj}(\Phi) \mathcal{Z}_{ji}(\Phi) \frac{\Delta A_j(\Phi)}{\Delta A_i(\Phi)} = \varepsilon_i E_{hi}(\Phi),
\]

(61)

where \( E_{bj}(\Phi) = \sigma T_j^4(\Phi) \) and \( \mathcal{Z}_{ji}(\Phi) \) is the exchange factor from the \( j \)th to the \( i \)th element, equal to the fraction of the radiant energy emitted by the \( j \)th element that is intercepted by the \( i \)th element. Although the analytical solution of the exchange factor is quite tractable for diffuse-walled enclosure problems, this is generally not the case for enclosures that

Although it was not imposed in obtaining the solutions, symmetry exists in both the geometry and thus the solutions found for this problem. Careful use of symmetry in setting up the equations could reduce the matrix size needed; however, it is not as easy to do this as it may appear, because radiation incident upon a particular element still originates from all surfaces in the enclosure, and the exchange factors must be found for all pairs of elements in the entire enclosure.
contain specular surfaces. The exchange factor, \( \mathbf{\bar{J}}_j(\Phi) \), can be expressed as the expectation of a random variable 

\[ \mathbf{\bar{J}}_j(\Phi, \xi) \]

where \( \xi \) contains three random variables that specify the emission direction and location of a random bundle leaving the \( j \)th surface element. Theoretically, then, \( \mathbf{\bar{J}}_j(\Phi) \) could be found by integrating over the probability distribution of each \( \xi_i \), i.e.:

\[
F_{ji}(\Phi) = E[\mathbf{\bar{J}}_{ji}(\Phi, \xi)] = \int_0^1 \int_0^1 \int_0^1 F_{ji}(\Phi, \xi) \, dp_1(\xi_1) \, dp_2(\xi_2) \, dp_3(\xi_3).
\] (62)

(This is, in fact, how view factors are calculated between diffuse surfaces.) Instead, we use Monte Carlo to estimate \( \mathbf{\bar{J}}_j(\Phi) \) in Equation (62):

\[
E[\mathbf{\bar{J}}_{ji}(\Phi, \xi)] \approx \mathbf{\bar{J}}_{ji}(\Phi) = \frac{1}{N_{bj}} \sum_{k=1}^{N_b} \mathbf{\bar{J}}_{ji}(\Phi, \xi^k) = \frac{N_{bji}}{N_{bj}},
\] (63)

where \( N_{bji} \) is the total number of bundles emitted by the \( j \)th element, and \( N_{bji} \) is the number of those elements that are absorbed by the \( i \)th element. Due to the law of large numbers, the Monte Carlo approximation of \( \mathbf{\bar{J}}_{ji}(\Phi, \xi) \) becomes exact as \( N_{bj} \to \infty \). Since we are restricted to using a finite number of bundles, however, \( \mathbf{\bar{J}}_{ji}(\Phi) \) contains a random error that propagates throughout the solution.

Assume the elements are renumbered so that \( T_i \) is specified for \( i=1\ldots m \) and \( q_{si} \) is specified for \( i=m+1\ldots N \). Equation (61) can then be rewritten as:

\[
q_{ji}(\Phi) + \sum_{j=m+1}^N \varepsilon_j E_{ji}(\Phi) \mathbf{\bar{J}}_{ji}(\Phi) \frac{\Delta A_j(\Phi)}{\Delta A_i(\Phi)} = \varepsilon_i E_{bi}(\Phi) - \sum_{j=1}^m \varepsilon_j E_{bj}(\Phi) \mathbf{\bar{J}}_{ji}(\Phi) \frac{\Delta A_j(\Phi)}{\Delta A_i(\Phi)},
\] (64)

for elements with specified \( T_i \), and:

\[
\varepsilon_i E_{bi}(\Phi) - \sum_{j=m+1}^N \varepsilon_j E_{ji}(\Phi) \mathbf{\bar{J}}_{ji}(\Phi) \frac{\Delta A_j(\Phi)}{\Delta A_i(\Phi)} = q_{si}(\Phi) + \sum_{j=m+1}^N \varepsilon_j E_{bij}(\Phi) \mathbf{\bar{J}}_{ji}(\Phi) \frac{\Delta A_j(\Phi)}{\Delta A_i(\Phi)},
\] (65)

for elements where \( q_{si} \) is specified. Equations (64) and (65) are arranged so the \( q_{si} \) and \( E_{bi} \) terms on the right-hand sides are known, while those on the left-hand side remain unknown. Writing Equations (64) and (65) for all the elements results in a system of \( N \) equations containing \( N \) unknowns, which are rearranged into a matrix equation:

\[
\mathbf{\mathbf{A}} \mathbf{x} = \mathbf{b},
\] (66)

where \( \mathbf{x} = [q_{s1}, \ldots, q_{sm}, E_{b1}, \ldots, E_{bN}]^T \). The dimension of \( \mathbf{\mathbf{A}} \) can be reduced by excluding equations that correspond to elements where either \( q_s \) or \( E_b \) is equal to zero. Since only one boundary condition has been imposed over each surface, Equation (66) is usually well-conditioned and can be solved by standard matrix inversion methods to yield the heat flux distribution over the design surface.

This heat flux is subject to a zero-mean random error induced by the uncertainty in the exchange factors, as well as a “bias” error caused by assuming uniform heat flux and temperature distribution over each surface element. The former error is reduced by increasing the number of bundles emitted by each element, while the latter diminishes with a higher level of grid refinement. Nevertheless, both errors result in a grid-dependent objective function containing a statistical uncertainty, which renders it difficult to minimize.

The magnitude of the random error is estimated through stratified sampling [80]. Suppose a total number of bundles, \( N_{\text{bundles}} \), is used to calculate the exchange factors throughout the process. First, the range of integration over Equation (62) is broken up into \( p \) pieces, each using \( N_{\text{bundles}} / p \) bundles. Each of the resulting \( p \) sets of exchange factors is then used to find an estimate of the heat flux distribution over the design surface, through Equation (66). Performing this procedure for each set of exchange factors results in \( p \) independent solutions, \( \mathbf{x}^k, \ k=1\ldots p \).
The heat flux at each discrete point over the design surface is then approximated by the average of the \( p \) independent solutions,

\[
\tilde{q}_d(\Phi) = \frac{1}{p} \sum_{k=1}^{p} \tilde{q}_d^k(\Phi),
\]

while the corresponding random error associated with each \( \tilde{q}_d(\Phi) \) is estimated from the sample standard deviation,

\[
\sigma_{\text{q},\text{ave}}(\Phi) = \frac{\sigma_{\text{q},\text{ave}}(\Phi)}{\sqrt{p}},
\]

where \( \sigma^2_{\text{q},\text{ave}}(\Phi) \) is the sample variance of the \( p \) independent solutions,

\[
\sigma^2_{\text{q},\text{ave}}(\Phi) = \frac{1}{p-1} \sum_{k=1}^{p} \left[ \tilde{q}_d^k(\Phi) - \tilde{q}_d(\Phi) \right]^2.
\]

### 5.2.2. Optimization Procedure

The goal of the optimization process is to minimize the objective function defined in Equation (31). In this application, however, the heat flux at each discrete point over the design surface, \( q_{sj}(\Phi) \), is estimated by \( \tilde{q}_{sj}(\Phi) \), which is in turn obtained from the Monte Carlo technique presented in the previous section. Accordingly, the objective function is approximated by:

\[
F(\Phi) \approx \tilde{F}(\Phi) = \frac{1}{b-a} \sum_{j=a}^{b} \left[ q_{sj}(\Phi) - q_{sj}^{\text{target}} \right]^2.
\]

This approximation contains a sampling error, \( \delta_1(\Phi) \), which is induced by the statistical uncertainty in \( \tilde{q}_{sj}(\Phi) \). The magnitude of \( \delta_1(\Phi) \) is estimated using the Kline–McClintock method [81],

\[
\delta_1(\Phi) = \frac{2}{b-a} \left\{ \sum_{j=a}^{b} \left[ \tilde{q}_{sj}(\Phi) - q_{sj}^{\text{target}} \right]^2 \cdot \sigma_{\text{q},\text{ave}}^2(\Phi) \right\}^{1/2},
\]

where \( \sigma_{\text{q},\text{ave}}(\Phi) \) are the uncertainty estimations corresponding to each \( \tilde{q}_{sj}(\Phi) \), obtained from the stratified sampling technique discussed in the previous section. (The values of \( \tilde{q}_{sj}(\Phi) \) are assumed to be either weakly dependent or independent, and therefore any covariance is neglected. This assumption is validated later in the paper.)

This uncertainty inherent in the evaluation of \( F(\Phi) \) makes the optimization of stochastic systems somewhat more complicated than that of deterministic systems. Nevertheless, many of the methods used to optimize stochastic systems are based on those used to optimize deterministic systems. The Kiefer–Wolfowitz method [82, 83] is a gradient-based technique that is often used when the analytical evaluation of the gradient of \( \tilde{F}(\Phi) \) is not possible. This method is based on the steepest-descent algorithm; at the \( k^{th} \) iteration, the step size is set equal to a power series based on iteration number,

\[
\alpha_k = \alpha_0 \frac{a}{k^{\alpha}}, \quad 0 \leq a \leq 1,
\]

and the search direction is set equal to:

\[
p_k = \frac{-\tilde{g}(\Phi_k)}{\| \tilde{g}(\Phi_k) \|},
\]
where $\mathbf{g}(\Phi_k)$ is the central-difference approximation of the gradient vector. The $p^{th}$ term of $\mathbf{g}(\Phi_k)$ is given by:

$$
\tilde{g}_p(\Phi_k) = \frac{F(\Phi_k + e_p \cdot h_k) - F(\Phi_k - e_p \cdot h_k)}{2h_k},
$$

(74)

where $e_p$ is the unit vector in the $p^{th}$ direction, and $h_k$ is the interval used in the finite-difference approximation at the $k^{th}$ iteration.

This estimate contains two sources of error: a bias error, $\delta_{2,p,k}(\Phi_k, h_k)$, due to the truncation of higher-order terms in the finite-difference approximation, and a random error, $\delta_{3,p,k}(\Phi_k, h_k)$, induced by the sampling error in $F(\Phi_k)$. (This random error tends to dominate finite difference approximation of higher-order derivatives, which is why steepest-descent is used instead of the Newton and quasi-Newton methods.) The bias error is given by [84]:

$$
\delta_{2,p,k}(\Phi_k, h_k) = g_p(\Phi_k) - \frac{F(\Phi_k + e_p \cdot h_k) - F(\Phi_k - e_p \cdot h_k)}{2h_k},
$$

(75)

and decreases as $h_k$ becomes small. (Note that the “exact” values of the objective function and gradient vector are used in Equation (75)). Assuming independent estimates of $F(\Phi_k + e_p \cdot h_k)$ and $F(\Phi_k - e_p \cdot h_k)$, the random error is found from:

$$
\delta_{3,p,k}(\Phi_k, h_k) = g_p(\Phi_k) - \frac{\sqrt{\delta_{1,p,k}^2(\Phi_k + e_p \cdot h_k, h_k) + \delta_{1,p,k}^2(\Phi_k - e_p \cdot h_k, h_k)}}{2h_k},
$$

(76)

and tends to increase as $h_k$ becomes small, since decreasing $h_k$ does not necessarily decrease the magnitude of the numerator in Equation (76). It is therefore important to select an intermediate step size that ensures both $\delta_{2,p,k}(\Phi_k, h_k)$ and $\delta_{3,p,k}(\Phi_k, h_k)$ are sufficiently small. One choice is to reduce $h_k$ with each successive iteration according to a power series similar to Equation (72),

$$
h_k = \frac{h_0}{b^k}, \quad 0 \leq b \leq 1.
$$

(77)

Pflug [85] recommends a value of $a = 1$ and $b = 1/3$ for Equations (72) and (77) respectively.

Another way of reducing $\delta_{3,p,k}(\Phi_k, h_k)$ is to use the same sequence of random numbers (common random numbers, [86]) to generate both $F(\Phi_k + e_p \cdot h_k)$ and $F(\Phi_k - e_p \cdot h_k)$. This results in similar random errors for both of the objective function estimates, which tend to cancel out in the numerator of Equation (74).

### 5.2.3. Demonstration of Method

The method described in the previous section is applied to solve the design problem shown in Figure 31. The enclosure has four-sides; the bottom (design) and right-hand (heater) surfaces and diffuse and have an emissivity of unity, while the top and left-hand surfaces are perfect specular reflectors. The design surface has a uniform emissive power of $E_{ba} = 0$, while the heater surface has a uniform heat flux of $q_{s1} = 1$ W/m².

The goal of the design problem is to find the enclosure geometry that produces a uniform heat flux of $q_{s,\text{target}} = -1$ W/m² over the design surface. This is done by minimizing the objective function defined by:

$$
\tilde{F}(\Phi) = \frac{1}{b-a} \sum_{j=a}^b [\tilde{q}_{s_j}(\Phi) + 1]^2,
$$

(78)

where $\Phi = \{\Phi_1, \Phi_2\}^T$ control the $x$- and $y$-coordinates of the upper left vertex of the enclosure, as shown in Figure 31. In order to limit the size of the enclosure, the design parameters are constrained to lie within the feasible region defined
by \(-0.5 \leq \Phi_1 \leq 0.5\) and \(0.5 \leq \Phi_2 \leq 1.5\) using the gradient projection method discussed previously. The minimum of Equation (78) is estimated using the Kiefer–Wolfowitz method, starting from \(\Phi = \{-0.5, 0.5\}^T\). The design constraints are enforced using the gradient projection method discussed in Section 5.1.3.

![Figure 31. Geometric optimization of an enclosure containing specular surfaces.](image)

As previously mentioned, assuming a uniform emissive power and heat flux distribution over each surface element produces discretization errors in the values of \(\bar{q}_s(\Phi)\), and accordingly in \(F(\Phi)\), that diminish with increasing levels of grid refinement. Therefore, it is necessary to perform a grid refinement study prior to the optimization process in order to estimate the discretization error in \(F(\Phi)\), and to ensure that a sufficient number of surface elements are used to perform the analysis. (This also helps the designer select a suitable convergence criterion, since seeking a value of \(F(\Phi)\) smaller than the discretization error is computationally expensive and does not necessarily provide further improvement in solution quality.)

It has been shown in previous studies ([77], for example,) that the discretization error can also be measured by calculating the energy imbalance over the enclosure surface, defined by:

\[
\%_{EI} = \frac{\sum_{j=1}^{N} q_{st} \Delta A_j}{\sum_{j=1}^{N} q_{st} \Delta A_j} 
\]

A grid refinement study performed at \(\Phi_0 = \{-0.5, 0.5\}^T\) is shown in Figure 32. A constant ration of bundles, \(N_{\text{bundles}}/N = 2.5 \times 10^5\), was used to obtain approximately the same sampling error at every level of refinement. Richardson extrapolation at the last three levels of refinement, \(\tilde{F}_{1024}(\Phi)\), \(\tilde{F}_{2048}(\Phi)\), and \(\tilde{F}_{4096}(\Phi)\), provides an estimate of the grid-independent objective function value, \(\tilde{F}_{1024}(\Phi) = 0.1694\). The corresponding discretization error at \(N = 4096\) is then \(-0.00068\), or \(-0.4\%)\). Since the discretization error is large for configurations with large heat flux and emissive power variations over the design surface, and because the heat flux distribution over the design surface is near uniform as \(\Phi\) approaches \(\Phi^*\), the discretization error in \(F(\Phi^*)\) is assumed to be no larger than that observed in \(F(\Phi_0)\).
Next, the effect of the number of bundles on random error, $\delta_1(\Phi)$, is demonstrated by systematically increasing the number of bundles used to calculate $\tilde{F}(\Phi_0)$ for an enclosure with $N = 1024$ surface elements. As shown in Figure 33, a power-law relationship exists between the number of bundles and the random error,

$$\delta_1(\Phi_0) = 1.0032 N^{-0.50174},$$

which is consistent with the $1/\sqrt{N_{\text{bundles}}}$ trend predicted by the central limit theorem and validates the assumption that covariance of the $\tilde{\phi}_j(\Phi)$ values can be neglected.

Figure 32. Grid refinement study at $\Phi_0 = \{-0.5, 0.5\}^T$.

Figure 33. Effect of the number of bundles on random error, $\delta_1(\Phi)$. 
Based on the above results, the Kiefer–Wolfowitz minimization was carried out using \( N = 1024 \) elements. In order to decrease \( \delta_k(\Phi_k) \) as \( \Phi_k \) approaches \( \Phi^* \), the number of bundles used in the simulation was increased with each successive iteration according to:

\[
N_{\text{bundles},k} = A \log_{10} k + B,
\]

with \( A = 160 \times 10^6 \) and \( B = 20 \times 10^6 \), values chosen based on the grid and bundle refinement studies. The minimization procedure is stopped when either:

\[
|\tilde{F}(\Phi_k) - \tilde{F}(\Phi_{k-1})| < 5 \times 10^{-4},
\]

or:

\[
|\tilde{F}(\Phi_k)| < 1 \times 10^{-3}.
\]

The resulting solution path is shown in Figure 34. Five steps were required to identify a local minimum at \( \Phi^* = \{0.0139, 0.8342\}^T \), with \( \tilde{F}(\Phi^*) = 8.01 \times 10^{-4} \). The heat flux distributions over the design surface corresponding to \( \Phi_0 \) and \( \Phi^* \) are shown in Figure 35, while the enclosure geometries are shown in Figure 36. The heat flux is expressed as a function of parameter \( u \), which specifies a location over the design surface.

\[\text{Figure 34. Kiefer–Wolfowitz solution path for minimizing } \tilde{F}(\Phi), \text{ defined in Equation (76).}\]
6. CONCLUDING REMARKS

Design of thermal systems by inverse and optimization methods holds promise for providing improved designs with greater efficiency. They provide systematic methods for achieving optimal designs rather than relying solely on intuition and experience. While these human factors are still helpful in recognizing good designs provided by the tools noted here, they may not provide the range of solutions that can be unearthed by these methods. Optimization and inverse methods may give designs that are unlikely to be found by conventional methods of trial and error, and should be considered as a powerful alternative to standard design methods.

Often, an inverse solution provides insight into the reasons for the ill-conditioned nature of the design problem, and this can be helpful in modifying the constraints on the design. For example, design constraints can be imposed for which there is no reasonable physical solution, and the inverse solution may provide the best solution that comes within some possibly very wide tolerance of the imposed requirements.
Optimization methods may fail when no reasonable solution exists, and may provide a solution that is a local minimum of the objective function and not the global minimum unless some sophisticated methods are imposed. On the other hand, these methods are usually fast and accurate, and most design problems are not faced with multiple local minima in the objective function.

The choice of direct inverse solution or optimization thus depends on the particular design problem being considered, and each has particular advantages and disadvantages.

ACKNOWLEDGMENTS

The authors wish to acknowledge the US National Science Foundation for support of this work under Grant CTS 0070545.

NOMENCLATURE

\(A\)  area
\(A\)  coefficient matrix
\(b\)  element of vector \(b\); function defined by Equations (34), (35)
\(b\)  vector of known values
\(B\)  approximation to \(H\), Equations (22), (23)
\(c\)  design constraint
\(c_p\)  specific heat
\(c\)  Newton’s direction, Equation (22)
\(f\)  input function, Equation (2)
\(e\)  unit vector relative to \(n\)
\(E\)  radiative emissive power, \(E_b\); error in inverse solution; expected value
\(E_b\)  blackbody radiative emissive power
\(f\)  transfer function in neural network; radiation configuration factor
\(F\)  functional in conjugate gradient method; objective function in optimization, Equation (12)
\(g\)  output function, Equation (2); function defined by Equations (34), (35)
\(g\)  gradient vector, Equation (16)
\(h\)  optimization step size
\(H\)  Hessian matrix, Equation (17)
\(I\)  input vector to neural network
\(J\)  Jacobian
\(k\)  thermal conductivity
\(K\)  kernel of integral equation; Equation (2)
\(L\)  length of surface
\(L\)  derivative operator in Tikhonov method
\(n\)  unit normal vector
\(N\)  number of unknowns in conjugate gradient method; number of elements in system
\(O\)  output vector from neural network
\(p\)  order of regularization in Tikhonov method
\(p\)  conjugate vectors; search direction in optimization
The Arabian Journal for Science and Engineering, Volume 27, Number 2C. December 2002

$q$ heat flux; energy per unit area per unit time
$q_o$ radiosity
$Q$ energy per unit time; power
$r$ residual of analytical system, Equation (3)
r residual vector, Equation (4); position vector for parameterized surface, Equation (32)
$R$ rank of coefficient matrix $A$
s vector connecting parametric surface elements
$S$ diagonal matrix of singular values
$S$ singular value
t time
$T$ absolute temperature
$u$ location in parameter space
$U$ element of matrix $U$
$U$ orthogonal vector, Equation (5); update vector, Equation (24)
$V$ orthogonal vector, Equation (5)
$W$ matrix of weights between nodes of neural network
$x$ coordinate position
$x$ solution vector
$\alpha$ regularization parameter in Tikhonov method, Equation (8)
$\beta$ blockage factor
$\gamma$ weight for penalty function, Equation (60)
$\delta$ wall thickness
$\varepsilon$ surface radiative emissivity
$\Phi$ design parameter for optimization
$\rho$ density
$\mathcal{I}_{ji}$ radiative exchange factor

Subscripts
$cd$ conduction
$cv$ convection
$e$ exact
$s$ design surface
$T$ known temperature
$u$ unknown

Superscripts
* value obtained by minimization of objective function
$target$ prescribed value on design surface
REFERENCES


K. Daun, H. Erturk, and J.R. Howell


*Invited Paper: Received 6 April 2002.*