Available online at www.sciencedirect.com
ScienceDirect
Journal of Power Sources 166 (2007) 376–385
JOURNAL OF POWER SOURCES
www.elsevier.com/locate/jpowsour

Adjoint method for solid-oxide fuel cell simulations

S. Kapadia *, W.K. Anderson, L. Elliott, C. Burdyshaw

University of Tennessee SimCenter at Chattanooga, 701 East M.L. King Boulevard, Chattanooga, TN 37403, USA

Received 14 December 2006; received in revised form 25 January 2007; accepted 26 January 2007
Available online 6 February 2007

Abstract

Adjoint methods suitable for obtaining sensitivity derivatives for numerical simulations of solid-oxide fuel cells are presented. The adjoint method is derived, and the implementation is discussed, including a methodology for accurately obtaining all the linearizations necessary for correct implementation. Results are included for a one-dimensional anode model that includes diffusion, permeation, and relevant chemical reactions. Using this model, the accuracy of the sensitivity derivatives is demonstrated for design variables describing geometric and material properties of the anode. Finally, the adjoint method is demonstrated for a three-dimensional fuel cell geometry where sensitivity derivatives are obtained for approximately 180,000 design variables. The results are used to modify the upper and lower walls of the plenum to obtain significantly improved distribution of fluid amongst the channels.

Keywords: SOFC; Fuel cell; Design; Adjoint; Sensitivity analysis

1. Introduction

Numerical simulations of solid-oxide fuel cells (SOFC) have been used to gain understanding of important physical phenomena and to supply guidance in the continuing development of improved fuel cells [1–8]. To date, the simulations have been primarily focused on analysis of fuel cells or fuel cell components, without strong emphasis on utilizing the simulations in a design optimization environment. Because of the emphasis on analysis instead of design, sensitivity information to determine the effects of variations in design parameters on performance has been primarily implemented by simply changing the parameter of interest, re-running the simulation, and comparing the results with those from the original simulation [1,5,6,8]. While this approach can be used to determine the effects of parameter variations on fuel cell performance, a more rigorous approach toward optimization would likely lead to better designs, and can also provide improved insight into the parameters affecting the performance of the fuel cell. For SOFC problems, example cost functions that can be used for improving performance include minimizing temperature variations, obtaining equal distribution of fuel in each of the channels, or maximizing power. Although not included in this study, time-dependent formulations are also possible that may be useful for minimizing start-up and short-term transient times. Design variables may be related to the shape/size of the fuel channels, electrodes, electrolyte, and interconnect, but may also be coupled to the stoichiometric composition of fuel or material properties such as the porosity or tortuosity of the electrodes.

In refs. [9,10], optimization algorithms have been used to improve the performance of a polymer-electrolyte-membrane fuel cell (PEM) using four design variables, where the sensitivity derivatives used for the optimization algorithm have been obtained using a finite-difference approach. While finite differences are often a viable means for computing sensitivity derivatives, this method can be computationally restrictive when a sufficiently large number of design variables are present. In addition, accurate derivatives can sometimes be difficult to obtain using finite differences because of subtractive cancellation errors [11,12], which occur when the function evaluations in the numerator become computationally indistinguishable when very small perturbations are used. To date, there has not been extensive research targeted at providing accurate sensitivity derivatives that may be used in conjunction with optimization

* Corresponding author. Tel.: +1 4234255513; fax: +1 4234255517.
E-mail addresses: Sagar-Kapadia@utc.edu (S. Kapadia), Kyle-Anderson@utc.edu (W.K. Anderson), Louie-Elliott@utc.edu (L. Elliott), Chad-Burdyshaw@utc.edu (C. Burdyshaw).

0378-7753/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.jpowsour.2007.01.085

Nomenclature

B	effective permeability coefficient ($m^2\ s^{-1}$)
c	molar concentration vector ($kmol\ m^{-3}$)
D_{ij}	binary diffusion coefficient ($m^2\ s^{-1}$)
D_{ij}^{m}	effective diffusion coefficient ($m^2\ s^{-1}$)
D_i^{kn}	Knudsen diffusion coefficient ($m^2\ s^{-1}$)
f	cost function (cost function dependent)
h	mesh size (m)
I	current density ($A\ m^{-2}$)
J	mass flux vector ($kg\ m^{-2}\ s^{-1}$)
Kn	Knudsen number
kb	backward reaction rate (reaction type dependent)
kf	forward reaction rate (reaction type dependent)
L	augmented cost function (cost function dependent)
M	molecular weight ($kg\ kmol^{-1}$)
N	molar flux vector ($kmol\ m^{-2}\ s^{-1}$)
p	pressure ($N\ m^{-2}$)
Q	solution vector (solution variable dependent)
$\langle r \rangle$	mean pore radii (m)
$\langle r^2 \rangle$	mean of squared pore radii (m^2)
R	discrete residual ($kg\ m^{-3}\ s^{-1}$)
Rate	reaction rate ($kmol\ m^{-3}\ s^{-1}$)
S	source term vector ($kg\ m^{-3}\ s^{-1}$)
t_a	anode thickness (m)
T	temperature (K)
V	control volume (m^3)
X_i	mole fraction of ith species

Greek symbols

Λ	costate variable vector (cost function dependent)
Ω_d	collision integral
β	design variable vector (design variable dependent)
χ	grid vector (m)
ε	Lenard–Jones parameter ($m^2\ kg\ s^{-2}$)
η	molecular viscosity ($kg\ m\ s^{-1}$)
ν	square root of relative molecular weight
ρ	mass concentration ($kg\ m^{-3}$)
σ	collision diameter (Å)
ψ	porosity/totruosity

Constants

F	Faraday constant 96484.56 ($A\ s\ mol^{-1}$)
K	Boltzmann constant $1.3806503 \times 10^{-23}$ ($J\ K^{-1}$)
R_u	universal gas constant 8314.4 ($J\ kmol^{-1}\ K^{-1}$)

Indices

d	diffusion
i, j	chemical species
kn	Knudsen
p	permeation
r	reforming reaction
s	shift reaction

algorithms to systematically improve existing fuel cell designs, particularly when there are many design variables. By using an adjoint method, sensitivity derivatives may be obtained for use in a design optimization environment. A particular strength of adjoint methods is that sensitivity information can be obtained with a computational cost that is only weakly dependent on the number of design variables, and is therefore enabling technology for design studies where many design variables are required.

In recent years, adjoint methods have been developed and utilized for numerical simulations in the aerodynamic community for sensitivity analysis, error estimation, and adaptive meshing [13–24]. For sensitivity analysis, adjoint methods are extremely valuable for determining the derivatives of engineering cost functions that depend on many design variables. Although sensitivity information could theoretically be obtained by systematically varying each design parameter independently, the cost of repeating the simulation with each variation of a design variable renders this approach unusable when a large number of parameters are present. The adjoint method is particularly suited to this class of problems because the sensitivity derivatives can be obtained for all design variables with the computational cost of a single solution of the nonlinear system used for analysis, a single solution of the linear adjoint system, and a matrix-vector multiply.

The primary goal of this study is to formulate and develop adjoint methodology for practical applications to fuel cells, with particular emphasis on solid-oxide fuel cells. The adjoint method is formulated and developed for SOFC applications to provide sensitivity derivatives of engineering cost functions. The adjoint method is then demonstrated using the one-dimensional model of transport phenomena inside an SOFC proposed by Lehnert in ref. [5]. Here, sensitivity derivatives with respect to geometric and material properties of the anode are obtained for the molar concentration of hydrogen at the interface between the anode and the electrolyte. Finally, a three-dimensional application is included where the cost function is based on the requirement of equally distributing fluid through the channels. As a demonstration, sensitivity derivatives are obtained for all the mesh points defining the surface of the geometry, totaling more than 180,000 design variables. Grouping some of these design variables to describe the shape of the upper and lower walls, the resulting sensitivity derivatives are then used to reshape the walls to achieve improved distribution of fluid amongst the channels.

2. Discrete adjoint method

The goal of an adjoint method is to determine sensitivity derivatives that can be used in a formal optimization procedure for minimizing a specified cost function, which is indicative of the performance of the system. In a design setting, a vector of design variables is chosen that will be allowed to change during the optimization procedure. These design variables may be geometric in nature or may be indicative of physical parameters such as porosity and tortuosity of the anode and cathode or specific concentrations of chemical species in the inlet. The sensitivity derivatives reflect the changes in the defined cost function with respect to each of the design variables.

A general optimization procedure begins by first defining a meaningful cost function and a desired set of design variables. A numerical analysis of the baseline system is then performed. The results of the analysis include the solution variables Q of the discretized partial differential equations, which are subsequently used to determine the initial cost. Because the numerical analysis involves discretization of the partial differential equations on a computational mesh, it should be noted that Q represents the vector of solution variables where each element of the vector is representative of one or more physical variables located at each mesh point, χ. Before reaching a local minimum during the optimization procedure, the solution of the partial differential equations and the resulting cost function will change in response to changes in the design variables. Note that if any of the design variables are representative of the shape of the geometry, then with each change in these design variables, the mesh must also deform to represent the new shape. Ultimately, the cost function may have an explicit dependence on the vector of design variables, β, but will also have an implicit dependence because Q and χ may also depend on the design variables. Therefore, the cost function is typically written to indicate the implicit and explicit dependence on the design variables as,

$$f = f(Q(\beta), \chi(\beta), \beta) \tag{1}$$

After the analysis problem has been solved and the cost function has been computed, the next step in the optimization procedure is to determine sensitivity derivatives. There are many options available for determining the sensitivity derivatives including finite differences, forward differentiation [25,26], and adjoint methods [13–23]. In this work, adjoint methods are used because of their ability to efficiently determine sensitivity derivatives for a large number of design variables. Once the sensitivity derivatives are obtained, they are used as input to an optimization code, such as a quasi-Newton method, to determine suitable changes to the design variables to obtain a decrease in the cost function. The design variables are updated and the process repeats by analyzing the new configuration that reflects the updated design variables. After each analysis, the cost function and gradients are typically monitored as an indication of how well the design process is proceeding.

2.1. Sensitivity analysis

As mentioned in the previous section, the adjoint method is used to determine sensitivity derivatives for the cost function, f, described in Eq. (1). If R represents the vector of discrete residuals at each mesh point, an augmented cost function L can be defined in the terms of the original cost function and the vector of discrete residuals as following.

$$L(Q(\beta), \chi(\beta), \beta, \Lambda) = f(Q(\beta), \chi(\beta), \beta) + \Lambda^{\mathrm{T}} R(Q(\beta), \chi(\beta), \beta) \tag{2}$$

In Eq. (2), Λ is the vector of Lagrange multipliers (also known as costate variables). Note that the augmented cost function, L, is a scalar quantity that is identical to the original cost function f, when $R(Q)$ is zero, indicating that the steady-state solution is obtained. The mesh coordinates and design variables are denoted as χ and β, respectively. Differentiating the augmented cost function with respect to each of the design variables yields the following set of equations for $\partial L/\partial\beta$, which is a column vector where each element represents the derivative of the augmented cost function with respect to a particular design variable.

$$\frac{\partial L}{\partial \beta} = \left\{ \frac{\partial f}{\partial \beta} + \left[\frac{\partial \chi}{\partial \beta}\right]^{\mathrm{T}} \frac{\partial f}{\partial \chi} \right\} + \left[\frac{\partial Q}{\partial \beta}\right]^{\mathrm{T}} \left\{ \frac{\partial f}{\partial Q} + \left[\frac{\partial R}{\partial Q}\right]^{\mathrm{T}} \Lambda \right\} + \left\{ \left[\frac{\partial R}{\partial \beta}\right]^{\mathrm{T}} + \left[\frac{\partial \chi}{\partial \beta}\right]^{\mathrm{T}} \left[\frac{\partial R}{\partial \chi}\right]^{\mathrm{T}} \right\} \Lambda \tag{3}$$

Because the elements of Λ are arbitrary, the term involving the derivatives of the dependent variables with respect to the design variables can be eliminated by solving a linear system of equations for the costate variables, also known as the adjoint equation.

$$\left[\frac{\partial R}{\partial Q}\right]^{\mathrm{T}} \Lambda = -\left\{ \frac{\partial f}{\partial Q} \right\} \tag{4}$$

Once the costate variables are obtained, the derivatives of the cost function with respect to all the design variables are obtained using a matrix-vector multiplication.

$$\frac{\mathrm{d}L}{\mathrm{d}\beta} = \left\{ \frac{\partial f}{\partial \beta} + \left[\frac{\partial \chi}{\partial \beta}\right]^{\mathrm{T}} \frac{\partial f}{\partial \chi} \right\} + \left\{ \left[\frac{\partial R}{\partial \beta}\right]^{\mathrm{T}} + \left[\frac{\partial \chi}{\partial \beta}\right]^{\mathrm{T}} \left[\frac{\partial R}{\partial \chi}\right]^{\mathrm{T}} \right\} \Lambda \tag{5}$$

In numerical simulations, the largest computational cost of computing sensitivity derivatives using the adjoint equations is due to the solution of the analysis equations and the adjoint equation, both of which are independent of the number of design variables. The only dependency on the number of design variables is in the evaluation of Eq. (5), which is generally much cheaper to compute than either the analysis or adjoint solutions. The result is a methodology that can be used for computing sensitivity derivatives for a large number of design variables that is only weakly dependent on the number of parameters used in the design.

2.2. Linearization

Note that the terms in Eqs. (2)–(5) involve differentiation of the discrete residual R, the cost function f, and the computational mesh χ with respect to the dependent variables Q, the design variables β, and the location of the mesh points χ. Correct implementation of this procedure can be extremely tedious to accomplish by hand and the resulting code can be difficult to maintain. To overcome the difficulties associated with hand differentiation, the complex-variable technique of Burdyshaw and Anderson [15] and Nielsen and Kleb [23] has been used for evaluating all the terms in the matrices required for solving the adjoint equations and for evaluating Eq. (5) once the costate variables have been obtained.

The complex-variable technique is derived by expanding functions in terms of a complex-variable Taylor series as shown

in Eq. (6). This particular example computes the nth element of the right hand side vector $[\partial f/\partial Q]$ of Eq. (4) by adding a complex perturbation to the nth element of the Q.

$$f(Q^n + \mathrm{i}h) = f(Q^n) + \mathrm{i}hf'(Q^n) + \frac{(ih)^2 f''(Q^n)}{2!} + \frac{(ih)^3 f'''(Q^n)}{3!} + \frac{(ih)^4 f''''(Q^n)}{4!} + \mathrm{O}(h^5)$$
$$= \left[f(Q^n) - \frac{h^2 f''(Q^n)}{2!} + \frac{h^4 f''''(Q^n)}{4!} \right] + \mathrm{i}\left[hf'(Q^n) - \frac{h^3 f'''(Q^n)}{3!} \right] + \mathrm{O}(h^5) \quad (6)$$

where, "i" denotes the imaginary part and "O" is the order of accuracy of the next term in the series. The imaginary and real parts of the above expression can be written separately as shown below.

$$\mathrm{Real}\,[f(Q^n + \mathrm{i}h)] \approx \left[f(Q^n) - \frac{h^2 f''(Q^n)}{2!} + \frac{h^4 f''''(Q^n)}{4!} \right] \quad (7)$$

$$\frac{\mathrm{Imag}\,[f(Q^n + \mathrm{i}h)]}{h} \approx \left[f'(Q^n) - \frac{h^2 f'''(Q^n)}{3!} \right] \quad (8)$$

Eqs. (7) and (8) can further be reduced to the following form to obtain both the function and the derivative with respect to Q^n with second-order accuracy.

$$f(Q^n) = \mathrm{Real}\,(f(Q^n + \mathrm{i}h)) + \mathrm{O}(h^2) \quad (9)$$

$$f'(Q^n) = \frac{\mathrm{Imag}\,(f(Q^n + \mathrm{i}h))}{h} + \mathrm{O}(h^2) \quad (10)$$

This technique has been used for evaluating complicated algebraic expressions in refs. [27,28] and has been extended for cost functions obtained through numerical simulations of partial differential equations in refs. [11,12]. The significance of the equation for the derivative is that by adding a complex perturbation to the variable of interest and re-evaluating the function, sensitivity derivatives can be obtained with relatively modest modifications to an existing code. To develop a "forward mode" sensitivity analysis code, the primary changes to an analysis code involve converting real-valued variables to complex and adding a small complex perturbation to the design variable of interest. After executing the resulting code, the sensitivity derivatives are easily harvested from the complex part of the solution.

Although the truncation error for this expression has the same order of accuracy as a real-valued central finite-difference expression, it is important to recognize that the above technique does not suffer from accuracy problems due to subtractive cancellation errors, which can often require making adjustments on a case-by-case basis when using finite differences. As a result, derivatives accurate to numerical precision can be obtained using the complex-variable technique by choosing a very small step size. The only requirement on the step size is that it should be distinguishable from zero as represented by the computer. A simple example of using the complex-variable technique is shown in Fig. 1, where the derivatives are obtained for a trigonometric function using a real-valued central finite-difference technique and the complex-variable technique. Fig. 1 shows the error in the computed derivatives obtained using both the central-difference and complex-variable methods plotted against the negative of the logarithm of the step size. Initially, both methods exhibit second-order accuracy as demonstrated by the fact that a one order of magnitude reduction in the step size reduces the error by two orders of magnitude. As the step size is continually reduced, the order of accuracy of the finite-difference method eventually degrades and the error begins to increase with subsequent reduction in the step size. In contrast, the derivatives computed using the complex-variable technique continue to exhibit true second-order accuracy regardless of the step size. In practice, the step size is chosen to be below the square root of the machine zero, guaranteeing derivatives accurate to machine precision.

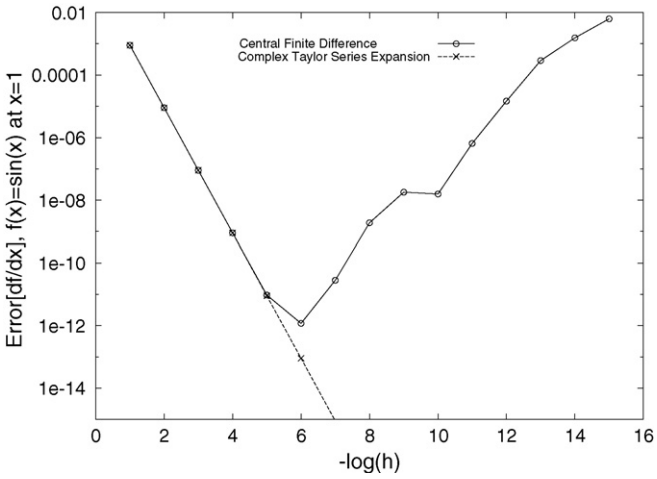

Fig. 1. Derivatives obtained using finite-difference and complex-variables.

In the context of the present work, the complex-variable technique is used to determine the elements of the matrices and vectors in Eqs. (2)–(5). In each case, a complex perturbation is added to the appropriate variables with respect to which differentiation is required. For example, to compute the elements of the matrix representing the linearization of the discrete residual with respect to the flow variables $\partial R/\partial Q$, a complex perturbation can be added to a particular flow variable and the complex-valued residual may be computed at each mesh point so that each perturbation effectively yields a column of the matrix. However, naive implementation of this technique would be very restrictive for three-dimensional problems due to the large number of flow variables. To overcome this difficulty, the method described in refs. [15,16] is used to exploit the fact that the computation of each residual only depends on mesh points within a finite stencil. In this method, the mesh is "colored" into a series of non-overlapping stencils to allow multiple columns to be computed simultaneously. By following this procedure, the computational burden is significantly reduced, requiring only the number of residual computations as there are colors in the mesh. Further details of this technique are given in refs. [15,16]. A similar technique has been described in ref. [23].

A few comments regarding the complex-variable approach in comparison to automatic differentiation (AD) techniques such as ADIFOR [29], ADIC [30] and TAMC [31] are in order. Because the complex-variable method can use arbitrarily small step sizes without suffering from subtractive cancellation errors, both methods can be used to determine sensitivity derivatives to numerical accuracy. In this regard, each method may be viewed as an alternative method for achieving the same goals. However, each method also has some advantages and disadvantages over the other in terms of developing and maintaining software that is continually evolving.

First, each method typically requires some modifications to the analysis code before being able to extract derivative information. In the complex-variable approach, the modifications are required to change the declaration of floating-point numbers to be of complex data type, to modify comparisons to account for mixing of complex and real-valued constants, and to sometimes supply intrinsic functions that do not automatically accept and return complex numbers. In the examples given in this paper, no new intrinsic functions had to be supplied. For automatic differentiation, the analysis code is often modified so that the automatic differentiation software can successfully process the analysis source code without errors. In this regard, it should be noted that AD software accepts source code as input and generates new source code as output that includes additional lines of code that provide the capability to compute sensitivity derivatives. The types of source-level changes required depend somewhat on the computer language and the specific AD software being used. However, the changes often include removing non-standard language extensions, removing arrays of unspecified length, removing common blocks from FORTRAN code that are used to pass non-static data to subroutines, adding compiler directives to assist in the generation of efficient code or preprocessing directives to deal with parallelization, eliminating system calls, changing function calls to eliminate the use of a single variable for both input and output, removing comparisons between inconsistent data types, and removing discontinuous flow control [32].

A performance comparison of the complex-variable approach with automatic differentiation is difficult to state categorically because the performance of each approach is dependent on the structure of the analysis code, the computer language, compiler options and source-level directives passed to the AD software, and whether or not further human generated modifications are done to the source code generated by the AD software. Experience has shown that for the complex-variable approach, software written in FORTRAN runs about twice as slow when complex arithmetic is used in comparison to when real arithmetic is used. For C++, there is a greater variation in run times depending on the compiler and the complex classes. The overhead associated with using complex arithmetic ranges from about 2–5. Relying on the results presented in ref. [32], the range of overhead for automatic differentiation is between 2–8, although most of the results are within a factor of about 2–4. Based on these results, the complex-variable and automatic differentiation approach exhibit roughly comparative efficiency.

One feature of the complex-variable approach is that in the process of converting the initial analysis code to accommodate complex arithmetic, the code can be made very easy to maintain. In particular, as new capability is added or errors are corrected, the software can be "self maintaining". Code generated using automatic differentiation software can also be easily maintained provided extensive modifications are not required to the AD generated source code to obtain efficiency. In this regard, the complex-variable approach has the advantage that the source code used to obtain sensitivity derivatives is identical to the analysis code so it is easy for developers to maintain. Code that has been generated from automatic differentiation software includes many new lines of code and can be difficult to follow.

Ultimately, both the complex-variable approach and the use of automatic differentiation software are viable solutions for obtaining sensitivity derivatives. In this work, the complex-variable approach is used because the code is easy to use and the code is easy to maintain.

3. Results

3.1. One-dimensional SOFC model

Initial use of the adjoint methodology is demonstrated for a numerical simulation similar to that described in ref. [5] and illustrated in Fig. 2. Here, the concentration of hydrogen, methane, carbon monoxide, carbon dioxide, and water vapor are computed through the thickness of an anode with concentrations and pressure prescribed at the boundary of the anode with the fuel channel, and the current density is specified at the interface with the electrolyte. The simulation assumes a constant temperature of 1123 K and includes molecular and Knudsen diffusion, permeation and reaction kinetics. The anode is assumed to be made of standard cermet as described in ref. [5], for which properties are given in Table 1. Although details of the governing equations can be found in ref. [5], a summary is provided

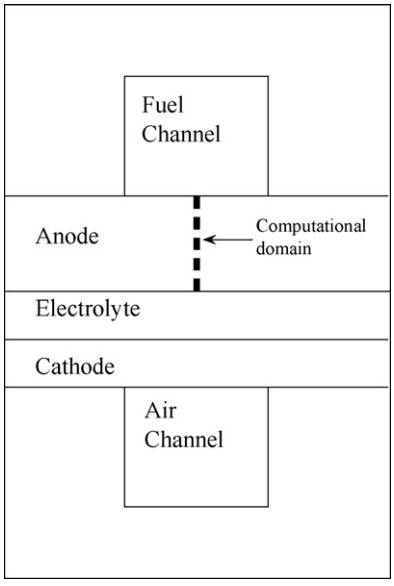

Fig. 2. Computational domain for one-dimensional SOFC anode simulation.

Table 1
Material properties and reaction rate constants from ref. [5]

kf_r	8.0×10^{-11} kmol m^{-3} Pa^{-2} s^{-1}
kb_r	1.5×10^{-23} kmol m^{-3} Pa^{-4} s^{-1}
kf_s	3.2×10^{-11} kmol m^{-3} Pa^{-2} s^{-1}
kb_s	3.5×10^{-11} kmol m^{-3} Pa^{-2} s^{-1}
ψ	1.56×10^{-1}
$\langle r \rangle$	1.07×10^{-6} m
$\langle r^2 \rangle$	3.8×10^{-11} m

below to describe the level of physical modeling included in the simulations.

The governing equations used in the simulations are based on steady-state mass conservation for each species.

$$\nabla \cdot J_i(Q) - S_i(Q) = 0 \tag{11}$$

In Eq. (11), Q represents the vector of solution variables. For this particular problem, $Q = \left[\rho_{H_2O}, \rho_{H_2}, \rho_{CO}, \rho_{CO_2}, \rho_{CH_4}\right]^T$, where ρ_i denotes the mass concentration of the ith species.

$$J_i = M_i N_i \tag{12}$$

Here, a subscript, "i", is used to denote each of the species included in the simulation and Eq. (12) indicates the relationship between mass and molar fluxes. The source term "S_i" indicates the production and consumption of each species due to chemical/electrochemical reactions.

The molar flux can be subdivided into (1) diffusion flux and (2) permeation flux as shown in Eq. (13).

$$N_i = N_i^{\mathrm{d}} + N_i^{\mathrm{p}} \tag{13}$$

The Mean Transport Pore Model (MTPM) [33] is used to compute the diffusion flux. According to MTPM,

$$\frac{N_i^{\mathrm{d}}}{D_i^{\mathrm{kn}}} + \sum_{\substack{j=1 \\ j \neq i}}^{\mathrm{NS}} \frac{X_j N_i^{\mathrm{d}} - X_i N_j^{\mathrm{d}}}{D_{ij}^m} = -c_t \nabla X_i \tag{14}$$

where, superscripts "kn" and "m" stand for the Knudsen and effective binary diffusion coefficients, respectively. Eqs. (15)–(18) illustrate further details involved in computing the aforementioned diffusion fluxes [34,35].

$$D_{ij}^m = \psi D_{ij} \tag{15}$$

$$D_{ij} = 1.8823 \times 10^{-2} \frac{\sqrt{T^3 \left(\frac{1}{M_i} + \frac{1}{M_j}\right)}}{p \sigma_{ij}^2 (\Omega_{\mathrm{d}})_{ij}} \tag{16}$$

$$\sigma_{ij} = \frac{\sigma_i + \sigma_j}{2} \tag{17.1}$$

$$(\Omega_{\mathrm{d}})_{ij} = \frac{1.06036}{\hat{T}^{0.1561}} + \frac{0.193}{\exp(0.47635\hat{T})} + \frac{1.03587}{\exp(1.52996\hat{T})} + \frac{1.76474}{\exp(3.89411\hat{T})} \tag{17.2}$$

$$\hat{T} = \frac{KT}{\varepsilon_{ij}} \tag{17.3}$$

$$\varepsilon_{ij} = \sqrt{\varepsilon_i \varepsilon_j} \tag{17.4}$$

$$D_i^{\mathrm{kn}} = \psi \langle r \rangle \frac{2}{3} \left(\frac{8 R_{\mathrm{u}} T}{\pi M_i}\right)^{1/2} \tag{18}$$

The permeation flux is given by Eq. (19), and is derived from Darcy's law [33].

$$N_i^{\mathrm{p}} = -X_i B_i \nabla c_t \tag{19}$$

Various terms appearing in this equation can be computed using Eqs. (20) and (21). Here, B_i is the effective permeability coefficient of species i and is described by Eq. (20). Though several different values are suggested for ω in the literature [5,33], the present simulation uses the value of $\pi/4$. Also, the Knudsen number of each component is indicated by Kn_i.

$$B_i = D_i^{\mathrm{kn}} \frac{\omega \nu_i + Kn_i}{1 + Kn_i} + \frac{\langle r^2 \rangle \psi p}{8\eta} \tag{20}$$

$$\nu_i = \sqrt{\frac{M_i}{M_{\mathrm{mixture}}}} \tag{21.1}$$

$$M_{\mathrm{mixture}} = \sum_{i=1}^{\mathrm{NS}} X_i M_i \tag{21.2}$$

The simulation presented in this paper considers two chemical reactions, namely, methane reforming and water gas shift reaction as presented by Eqs. (22.1) and (22.2), respectively.

$$CH_4 + H_2O \underset{kb_r}{\overset{kf_r}{\rightleftarrows}} CO + 3H_2 \tag{22.1}$$

$$CO + H_2O \underset{kb_s}{\overset{kf_s}{\rightleftarrows}} CO_2 + H_2 \tag{22.2}$$

Reaction rates are computed using the global reaction model. According to this model,

$$\mathrm{Rate_r} = kf_r p_{CH_4} p_{H_2O} - kb_r p_{CO} p_{H_2}^3 \tag{23.1}$$

$$\mathrm{Rate_s} = kf_s p_{CO} p_{H_2O} - kb_s p_{CO_2} p_{H_2} \tag{23.2}$$

Subscripts "r" and "s" stand for reforming and shift reactions, respectively. Reaction rate constants are taken from [5] and are also tabulated in Table 1. The source terms for each species are computed using the above reactions rates.

Two boundary conditions are required to solve Eq. (11) in a single spatial direction. Dirichlet boundary conditions are applied on the boundary between the anode and the fuel channel, i.e. fixed values of mole fractions and pressure are specified. The pressure is assumed to be one atmosphere and the mole fractions are given in Table 2. At the anode–electrolyte interface the mass flux of each species is specified to reflect a prescribed current density of 3000 A m^{-2}, which is a Neumann boundary condition. The following two electrochemical reactions are accounted

Table 2
Prescribed mole fractions at interface between anode and fuel channel

Species	H_2	H_2O	CO	CO_2	CH_4
Mole fraction	0.263	0.493	0.029	0.044	0.171

for at the anode–electrolyte interface.

$$H_2 + \frac{1}{2}O_2 \rightarrow H_2O \tag{24.1}$$

$$CO + \frac{1}{2}O_2 \rightarrow CO_2 \tag{24.2}$$

The current density is related to the species mass flux according to Faraday's law. Thus, conversion rates of the species involved in above reactions can be written as the following.

$$\text{Rate}_{H_2} = (-\text{Rate}_{H_2O}) = \frac{I}{2F} \tag{24.3}$$

$$\text{Rate}_{CO} = (-\text{Rate}_{CO_2}) = \frac{I}{2F} \tag{24.4}$$

To solve the governing equations, the computational domain is divided into a finite number of control volumes, and the partial differential equations are discretized using a finite-volume technique. Newton's method is used to solve the resulting system of nonlinear algebraic equations. Other methods could be used with no effect on the final solution. At each iteration, the complex-variable technique is used to fill the matrix representing the linearization of the discrete residuals with respect to the field variables. It should be noted that this matrix is the transpose of the matrix required in the development of the adjoint method. Although the linearization can be accomplished by hand differentiation, the complexity of the fluxes shown in Eqs. (13)–(21) requires significant effort to obtain an error-free implementation. In contrast, linearizing these fluxes is easily accomplished using the complex-variable approach.

For the first simulation, the computational domain is divided into 20 equally spaced intervals. Fig. 3 compares the molar formation rates of all five species from the current simulation with those presented by Lehnert et al. [5]. The agreement is satisfactory but indicates that there are some modeling differences. Specifically, the binary diffusion coefficients and Knudson numbers used in ref. [5] are not given in the reference, and are likely different than those employed here.

To demonstrate the use of the adjoint method for sensitivity analysis, the cost function is chosen as the molar concentration of hydrogen at the anode–electrolyte interface and the design variables include the ratio of porosity to tortuosity, the mean pore radii, and the anode thickness. Although only three design variables are used, many more could be added without significantly impacting the computational cost. Fig. 4 shows the plot of costate variables through the anode. Note that the costate variables do not typically follow trends exhibited by the species formation rates or other field variables such as concentrations, but are indicative of the sensitivity of the cost function to changes in the residual. This can be seen from Eq. (5), where it is clear that in computing the sensitivity derivatives, the costate variables weight the contributions due to changes in the residuals with respect to the design variables.

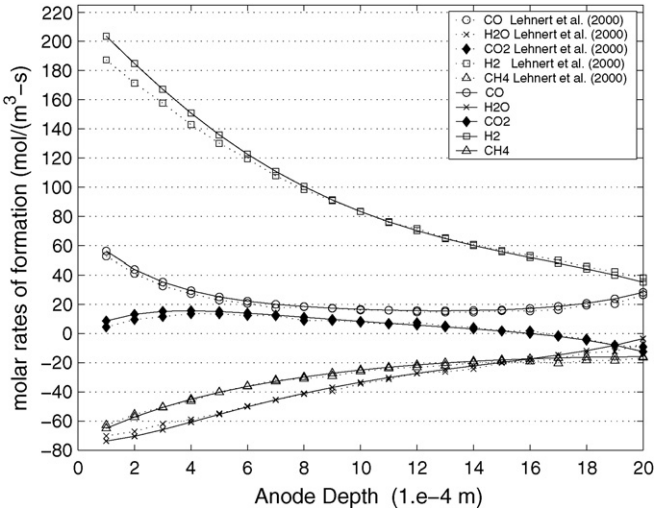

Fig. 3. Molar rates of formation through the thickness of the anode.

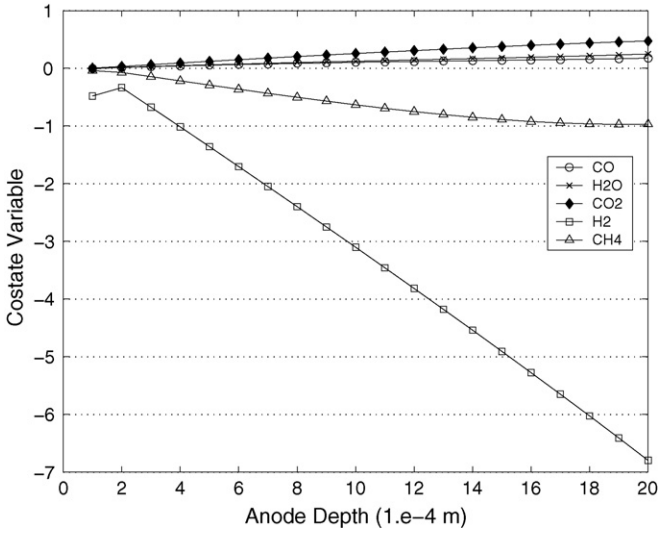

Fig. 4. Distribution of costate variables through thickness of the anode.

Table 3
Comparison between sensitivity derivatives obtained using the adjoint method and finite-difference method

Cost function – molar concentration of hydrogen at anode–electrolyte interface			
	$\partial L/\mathrm{d}t_a$	$\partial L/\mathrm{d}\psi$	$\partial L/\mathrm{d}\langle r\rangle$
Finite differences	3.0103686566e-01	−1.0500656390e-03	−7.2445001109e+01
Adjoint	3.0103704637e-01	−1.0500656734e-03	−7.244506333e+01

Table 3 shows a comparison of sensitivity derivatives obtained using the adjoint formulation to those obtained using a centered finite-difference technique. For the adjoint results shown in Fig. 4, a complex-valued step size of 1×10^{-8} is used when computing the linearizations. It should be noted that smaller step sizes have also been used without changing the results, indicating that the derivatives are consistent to machine accuracy. The step size for the finite-difference calculations is 1×10^{-5} and has been determined experimentally based on a procedure similar to that shown in Fig. 1. As seen, the comparisons of these results are excellent. However, it should be noted that accurate sensitivity derivatives were obtained with the finite-difference approach by a trial-and-error procedure to determine the optimal step size. Secondly, sensitivity derivatives obtained using finite differences require a solution of the governing equations for each design variable. Because this is a one-dimensional simulation and there are few design variables, this cost is not prohibitive. However, for high-fidelity three-dimensional simulations, repeated computations to obtain the sensitivity derivatives would be too costly if there are more than just a few design variables.

3.2. Three-dimensional flow problem

To demonstrate the use of adjoint methodology for a case with a large number of design variables, sensitivity derivatives are obtained for a three-dimensional fluid flow problem that is representative of fuel flowing between channels in a fuel cell. The geometry, illustrated in Fig. 5, is similar to that described in ref. [36]. For this problem, the fluid enters the cell on the lower left and exits on the upper right after flowing through a series of manifolds.

The numerical solution of this problem is obtained by assuming incompressible, non-reacting flow and using the three-dimensional, implicit, unstructured Navier–Stokes flow solver described in ref. [37]. This particular geometry is selected as it resembles a shape of the planar type SOFC. Note that this simulation does not include a high-fidelity physical model required for a complete fuel cell simulation. However, the model is useful for studying general flow distribution between the channels. Furthermore, it provides an excellent example for demonstrating the use of the adjoint method for determining sensitivity derivatives for a large number of design variables.

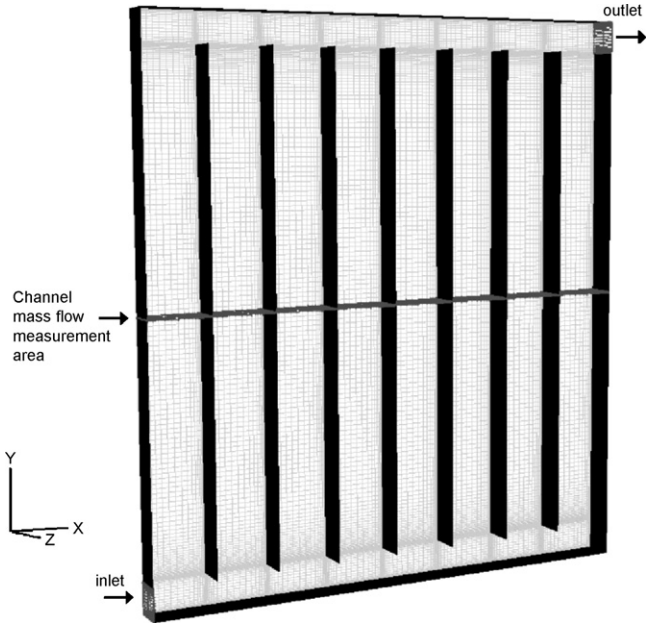

Fig. 5. Three-dimensional fuel cell geometry.

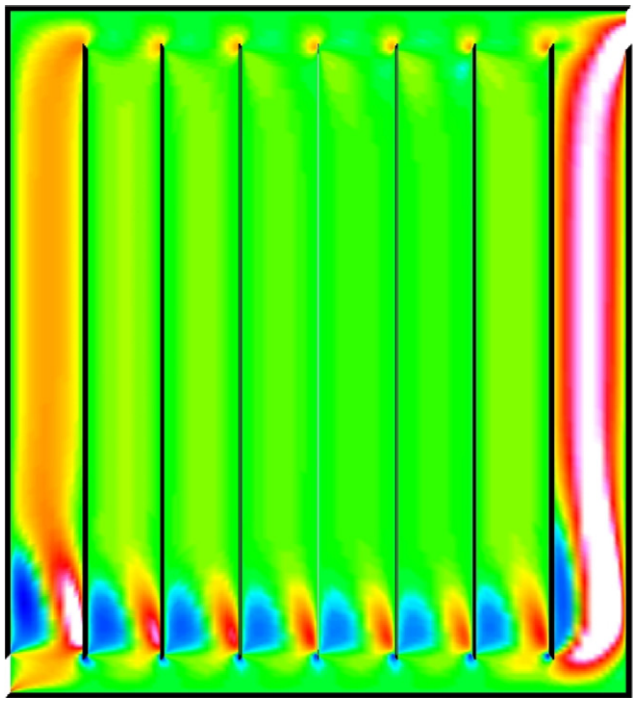

Fig. 6. Contours of vertical velocity for initial flow solution.

The mesh includes approximately 55,000 hexahedral elements and about 86,000 nodes. Of these nodes, approximately 60,000 are distributed along the surfaces of the geometry. Contours of the computed vertical velocity components are depicted in Fig. 6. As seen in the figure, a disparate portion of the mass flow traverses through the channels closest to the inlet and exit opening.

To obtain better distribution of fuel amongst the channels, a cost function is evaluated along the plane seen in Fig. 5. The cost function is the sum of the deviation of the mass flow in each channel from the average mass flow through the overall cell [16]. In a design optimization, minimizing this cost function would improve distribution of the mass flow through each channel. The adjoint method has been used to obtain the sensitivity derivatives of this cost function with respect to the coordinates of each node on the surface of the geometry, yielding a total number of design variables in excess of 180,000. Fig. 7 depicts the gradient vectors overlain on top of the vertical velocity contours indicating the sensitivity of the objective function to changes in the surface geometry. Although sensitivity derivatives have been obtained for all the surface points in the mesh to demonstrate the utility of the adjoint method, an examination of the gradient vectors in Fig. 7 suggests that the sensitivity derivatives are largest along

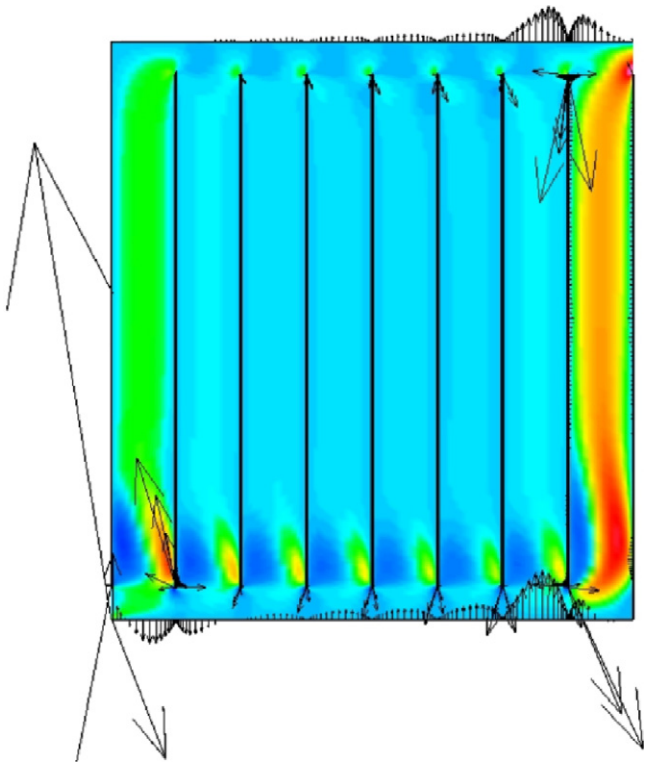

Fig. 7. Gradients of design variables.

the upper and lower walls. New design variables are therefore obtained by grouping the original design variables on the upper and lower walls to allow ramping of the upper and lower plenum surfaces. Similar grouping of design variables in this manner is often used to prevent oscillations from appearing in the surface and to simplify the design space. Results of the flow distribution after one design iteration using steepest descent are shown in Fig. 8. As seen in the figure, modifying the shape of the upper and lower walls has dramatically improved the distribution of fuel through the channels. In addition, the cost function is reduced by a factor of four.

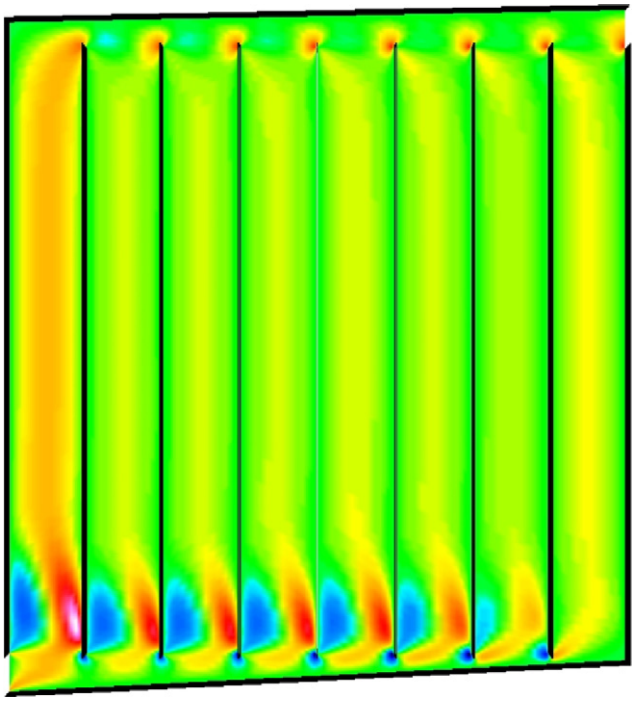

Fig. 8. Vertical velocity contours after modifying upper and lower walls.

4. Conclusions and future work

An adjoint based sensitivity analysis technique has been developed and demonstrated for applications to solid-oxide fuel cells. Sensitivity analysis is demonstrated for a one-dimensional transport problem through the anode of a solid-oxide fuel cell, where the physical model includes multicomponent diffusion, permeation, and appropriate chemical and electrochemical reactions. The accuracy of sensitivity derivatives is established for design variables describing geometric and material properties of the anode.

Because one of the strengths of adjoint methods is in determining sensitivity derivatives for large numbers of design variables, the adjoint method is further demonstrated on a three-dimensional design problem where the objective is to achieve equal distribution of fluid between multiple channels in a manifold geometry. Sensitivity derivatives are obtained for approximately 180,000 design variables describing the geometry of the cell. This problem emphasizes the usefulness of adjoint methods in obtaining sensitivity derivatives for a design problem with many parameters, which may otherwise be prohibitively expensive to compute by individually changing each parameter and re-running the entire simulation. The results of the sensitivity analysis are used to improve the distribution of fluid amongst the channels by grouping the design variables to describe the rotational positions of the upper and lower walls. Adjusting the walls accordingly results in much improved distribution of fluid in the channels.

Future work is targeted at further developing the adjoint method for industrial applications to fuel cell designs. This work includes merging more realistic physical models into the three-dimensional simulation and adjoint codes as well as coupling the sensitivity analysis with formal optimization procedures.

Acknowledgements

This work was supported by the Tennessee Higher Education Commission (THEC) Center of Excellence in Applied Computational Science and Engineering, and by the Department of Energy Chattanooga Fuel Cell Demonstration Project, Award number DE-FC36-04GO14261. This support is greatly appreciated.

References

[1] J.R. Ferguson, J.M. Fiard, R. Herbin, J. Power Sources 58 (1996) 109–122.
[2] J.M. Fiard, R. Herbin, Comp. Methods Appl. Mech. Eng. 115 (1994) 315–338.
[3] B.A. Haberman, J.B. Young, Int. J. Heat Mass Transfer 47 (2004) 3617–3629.
[4] M.M. Hussain, X. Li, I. Dincer, Int. J. Energy Res. 29 (2005) 1085–1101.

[5] W. Lehnert, J. Meusinger, F. Thom, J. Power Sources 87 (2000) 57–63.
[6] P.W. Li, M.K. Chyu, J. Power Sources 124 (2003) 487–498.
[7] S.H. Chan, K.A. Khor, Z.T. Xia, J. Power Sources 93 (2001) 130–140.
[8] S. Campanari, P. Iora, J. Power Sources 132 (2004) 113–126.
[9] M. Grujicic, K.M. Chittajallu, Appl. Surf. Sci. 227 (2004) 56–72.
[10] M. Grujicic, K.M. Chittajallu, Chem. Eng. Sci. 59 (2004) 5883–5895.
[11] J.C. Newman, W.K. Anderson, D.L. Whitfield, Multidisciplinary Sensitivity Derivatives Using Complex Variables, MSSU-COE-ERC-98-08, Mississippi State University, July 1998.
[12] W.K. Anderson, J.C. Newman, D.L. Whitfield, E.J. Eric Nielsen, AIAA J. 39 (1) (2001) 56–63.
[13] W.K. Anderson, V. Venkatakrishnan, Comput. Fluids 28 (4–5) (1999) 443–480.
[14] W.K. Anderson, D.L. Bonhaus, AIAA J. 37 (2) (1999) 185–191.
[15] C.E. Burdyshaw, W.K. Anderson, AIAA J. Aerospace Comput. Inform. Commun. 29 (10) (2006) 401–413.
[16] C.E. Burdyshaw, Achieving Automatic Concurrency Between Computational Field Solvers and Adjoint Sensitivity Codes, Ph.D. Thesis, University of Tennessee, Chattanooga, May 2006.
[17] A. Jameson, J. Sci. Comput. 3 (1998) 233–260.
[18] A. Jameson, J.J. Alonso, J. Reuther, L. Martinelli, J.C. Vassberg, Aerodynamic Shape Optimization Techniques Based on Control Theory, AIAA Paper No. 98-2538, 1998.
[19] B. Mohammadi, Optimal Shape Design, Reverse Mode of Automatic Differentiation and Turbulence, AIAA Paper No. 97-0099, 1997.
[20] E.J. Nielsen, W.K. Anderson, AIAA J. 40 (6) (2002) 1155–1163.
[21] E.J. Nielsen, W.K. Anderson, AIAA J. 37 (11) (1999) 1411–1419.
[22] E.J. Nielsen, Aerodynamic Design Sensitivities on an Unstructured Mesh Using the Navier–Stokes Equations and a Discrete Adjoint Formulation, Ph.D. Dissertation, Virginia Polytechnic Institute and State University, 1998.
[23] E.J. Nielsen, W.L. Kleb, AIAA J. 44 (4) (2005) 827–836.
[24] M. Park, Adjoint-Based, Three-Dimensional Error Prediction and Grid Adaptation, AIAA Paper No. 2002-3286, 2002.
[25] G.J.-W. Hou, A.C. Taylor III, V.M. Korivi, Int. J. Num. Methods Eng. 37 (1994) 2251–2266.
[26] G.W. Burgreen, Three-Dimensional Aerodynamics Shape Optimization Using Discrete Sensitivity Analysis, Ph.D. Dissertation, Old Dominion University, 1994.
[27] J.N. Lyness, C.B. Moler, SIAM J. Num. Anal. 4 (1967) 202–210.
[28] W. Squire, G. Trapp, SIAM Rev. 10 (1) (1998) 110–112.
[29] C. Bischof, A. Carle, G. Corliss, A. Griewank, P. Hovland, Scientific Programming (1992) 1–29, no. 1.
[30] C. Bischof, L. Roh, A. Mauer-Oats, Software: Pract. Exp. 27 (12) (1999) 1427–1456.
[31] M.S. Gockenbach, "Understanding code generated by TAMC", IAAA Paper TR00-29, Department of Computational and Applied Mathematics, Rice University, Texas, USA, 2000.
[32] P.A. Cusdin, "Automatic Sensitivity Code for Computational Fluid Dynamics", Ph.D. Thesis, School of Aeronautical Engineering, Queen's University, Belfast, U.K., 2005.
[33] D. Arnost, P. Schneider, Chem. Eng. J. 57 (1995) 91–99.
[34] R. Bird, W. Stewart, E. Lightfoot, Transport Phenomena, second ed., John Wiley & Sons, Inc., 2002, 863–866.
[35] J. Anderson Jr., Hypersonic and High Temperature Gas Dynamics, first ed., McGraw-Hill Book Company, 1989, 596–600.
[36] S. Kim, E. Choi, Y. Cho, Int. Comm. Heat Mass Transfer 22 (3) (1995) 329–341.
[37] K. Sreenivas, D.G. Hyams, D.S. Nichols, B. Mitchell, L.K. Taylor, W.R. Briley, D.L. Whitfield, "Development of an Unstructured Parallel Flow Solver for Arbitrary Mach Numbers", AIAA Paper No. 2005-0325, 2005.