46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit
25 - 28 July 2010, Nashville, TN

AIAA 2010-6845

Design Optimization of Rocket Nozzles

A. A. Ezertas[1], M. Yumusak[2] and S. Eyi.[3]

In this study it is aimed to develop a computational design tool, which can be used in design of rocket components in chemically reacting flows. The flow analysis is based on axisymmetric Euler and the finite rate reaction equations. These coupled equations are solved by utilization of Newton's method. The evaluation of Jacobian matrices that are needed by Newton's method is performed both numerically and analytically. UMFPACK direct sparse matrix solver is use to solve the resultant linear system of equations. Adjoint method is applied to evaluate the sensitivities required in the design optimization. The performance of the gradient-based optimization method is demonstrated by rocket motor nozzle design where the produced thrust is maximized.

I. Introduction

Computational Fluid Dynamics (CFD) is extensively used in rocket motor design. In traditional design methods, engineers modify initial prototypes with their experience, and analyze the modified configurations with CFD solvers. In general, these modifications are based on the trial and error approach and several configurations are analyzed before the final decision is reached. In traditional methods, it is difficult to know that if the final design is the best one, and computational time is very long. In order to reduce the computational time, sometimes simpler models are used for flow analyses. However, these models may reduce the reliability of the design results. In recent years, new advancements are achieved to increase the reliability and the efficiency of design methods. Automatic design methods are developed by combining the CFD and optimization codes. Automatic design methods have several advantages. The experience and information needed for design can be reduced, and innovative designs can be made at the conditions that have never been tried before. The design cost can also be decreased by reducing the engineers' workload and design time. Another advantage of the automatic methods is that the design can be obtained as a solution of optimization problem. In this way, if there is no solution with the specified design conditions, the design can be calculated as a best available solution.

The performance of a rocket can be improved with the aerodynamics and aero-thermodynamics design optimization. Minimizing the pressure and friction losses of internal and external flows may increase the range or payload capacity of rockets. The infrared detection can be minimized by optimizing the composition of exhaust gases. The maneuverability of rockets can be improved with the optimum design of control surfaces. Nozzle is one of the important components of rockets because its efficiency significantly affects the performance of rocket. By minimizing the losses in nozzle, the thrust of a rocket can be increased. The studies related to design of rocket's nozzle are not new. Optimization techniques used to design rocket nozzle contours have been utilized since 1950. Rao[1] developed a method which optimizes a rocket nozzle contour for a given length or expansion ratio such that maximum thrust is achieved. Rao's method was based on the assumption of inviscid isentropic flow. This method has been used in many studies of different classes of rocket engines with a variety of results. In a study by Farley,[2] three large Rao optimized bell nozzles were compared to a 15 degree conical nozzle. The thrust produced by the optimized nozzles was greater than that obtained with the conical nozzle. In fact, bell contour nozzles have been used routinely for many years in large liquid rocket engines.[3] Conical nozzles are typically used only when fabrication and design costs outweighed performance. In recent years, more accurate design methods are developed by using Navier-Stokes equations[4,5].

Main objective of this study is to develop a reliable and efficient aerothermodynamic design optimization tool that can be used for rocket's nozzle. The reliability of design methods depends on the accuracy of flow models used. In order to capture the chemically reacting and rotational flow physics, the finite-rate reaction and axisymmetric Euler equations are solved simultaneously. In design optimization, usage of implicit methods for flow analysis is advantageous since efficient evaluation of sensitivities is enabled. In order to improve the efficiency of design method and to reduce the numerical stiffness that occurs in the solution of reaction equation, exact Newton's method is used. Newton's method necessitates the evaluation of Jacobian matrix whose elements

[1] Graduate student, Aerospace Engineering, Middle East Technical University Ankara, Turkey
[2] Chief Engineer, ROKETSAN Missiles Industries Inc., Ankara, Turkey
[3] Assoc. Professor, Aerospace Engineering Department, Middle East Technical University Ankara, Turkey

are the derivative of residual vector with respect flow variable vector. Analytical or numerical methods can be used in the calculation of Jacobian matrices[6,7]. The analytical method is more accurate and faster in Jacobian evaluation compared to the numerical method. However, the implementation of numerical method is much easier; the analytical method requires code development. In gradient base design optimization, the derivatives of objective function with respect to design variables are needed. In literature, these derivatives are called sensitivities. The accurate and efficient calculation of sensitivities is important for the performance of design method. Direct differentiation or adjoint methods can be used to evaluate the sensitivities. Adjoint method has more advantageous because the Jacobian matrix is solved only once to evaluate the sensitivities of all design variables. In order to avoid solving large Jacobian matrix for each design variable the adjoint method is used in this study. The performance of the developed design tool is presented by the maximization of the thrust generated by rocket motor nozzle that is performed with the gradient-based optimization technique.

II. Flow Model

In order to model chemically reacting flows in combustion chamber and nozzle the finite reaction rate and axisymmetric Euler equations are solved simultaneously. The steady state axisymmetric Euler and chemical reaction equations in generalized coordinates can be written in non-dimensional form as below:

$$\frac{\partial \hat{F}(\hat{W})}{\partial \xi} + \frac{\partial \hat{G}(\hat{W})}{\partial \eta} + \hat{H} - \hat{S} = 0 \tag{1}$$

$$\hat{W} = \frac{1}{J}\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e_t \\ \rho \\ \vdots \\ \rho_{K-1} \end{bmatrix}, \quad \hat{F} = \frac{1}{J}\begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U \\ (\rho e_t + p) U \\ \rho_1 U \\ \vdots \\ \rho_{K-1} U \end{bmatrix} \quad \hat{G} = \frac{1}{J}\begin{bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ (\rho e_t + p) V \\ \rho_1 V \\ \vdots \\ \rho_{K-1} V \end{bmatrix} \quad \hat{H} = \frac{1}{yJ}\begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 \\ (\rho e_t + p) v \\ \rho_1 v \\ \vdots \\ \rho_{K-1} v \end{bmatrix} \quad \hat{S} = \frac{1}{J}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \dot{\omega}_1 \\ \vdots \\ \dot{\omega}_{K-1} \end{bmatrix} \tag{2}$$

where ρ is the density, u and v are the components of the velocity vector, p is the pressure, e_t is the total energy per unit volume, U and V are contravariant velocity components. In Equation (2), J is the coordinate transformation Jacobian, ξ and η are the curvilinear coordinates, and ξ_x, ξ_y, η_x, η_y are the transformation metrics.

The discretized form of the steady axisymetric Euler and chemical reaction equations given in Equation 1 can be written as:

$$\frac{\delta_\xi \hat{F}}{\Delta \xi} + \frac{\delta_\eta \hat{G}}{\Delta \eta} + \hat{H} - \hat{S} = 0 \tag{3}$$

For a cell centered finite volume method Equation 3 can then be written as:

$$(\hat{F}_{i+1/2,j} - \hat{F}_{i-1/2,j}) + (\hat{G}_{i,j+1/2} - \hat{G}_{i,j-1/2}) + \hat{H}_{i,j} - \hat{S}_{i,j} = 0 \tag{4}$$

where the i±1/2 and j±1/2 denote a cell interfaces. The inviscid fluxes of the Euler equations represent the convective phenomena. Van Leer's upwind flux splitting scheme[8] is used for the spatial discretization of the flux vectors. The fluxes are calculated at the cell interfaces by using the flow variables interpolated from the cell center. Second order accuracy in space is obtained by using MUSCL (Monotonic Upstream-Centered Scheme Conservation Law) [9] interpolation technique. MUSCL scheme requires the limiter function, in order to prevent oscillations and spurious solutions in regions of high gradients hence Van Albada's [10] limiter is used in the developed flow solver.

The chemical reaction includes 8 species. They are CO_2, CO, OH, H, O_2, H_2, H_2O, O. The reactions considered are given in Table 1.

Table 1 Chemistry models for simulation

$OH+CO \Leftrightarrow H+CO_2$	$H_2+O \Leftrightarrow H+OH$
$CO+O_2 \Leftrightarrow CO_2+O$	$H_2O+O \Leftrightarrow 2OH$
$O_2+H \Leftrightarrow O+OH$	$H_2+OH \Leftrightarrow H_2O+H$

A. Thermodynamic Model

The present solver presumes that the standard-state thermodynamic properties are thermally perfect, in that they are only functions of temperature and are given in terms of polynomial fits to the molar heat capacities at constant pressure.

$$\frac{C_{pk}^0}{R} = \sum_{n=1}^{N} a_{nk} T^{(n-1)} \tag{5}$$

The superscript 0 refers to the standard-state of 1 atmosphere. For perfect gases, however the heat capacities are independent of temperature, and the standard state values are actual values. Other thermodynamic properties are given in terms of integrals of the molar heat capacities. First, the standard-state molar enthalpy is given by heat capacities. The standard-state molar enthalpy is given by

$$H_k^0 = \int_{T_{ref}}^{T} C_{pk}^0 dT + H_k^0\left(T_{ref}\right) \tag{6}$$

The standard-state molar entropy is given as

$$S_k^0 = \int_{T_{ref}}^{T} \frac{C_{pk}^0}{T} dT + S_k^0\left(T_{ref}\right) \tag{7}$$

In above equations, the reference temperature, T_{ref}, is chosen as 298 K. One also often needs the mixture-averaged thermodynamic properties. As with pure-species properties, the present solver thermodynamic subroutines return in either mass or molar units. The mixture-averaged specific heats are:

$$\overline{C}_p = \frac{\sum_{k=1}^{K} C_{pk}^0 X_k}{\overline{W}} \tag{8}$$

$$\overline{C}_v = \frac{\sum_{k=1}^{K} C_{vk}^0 X_k}{\overline{W}} \tag{9}$$

Then, energy equation gives:

$$E = \sum_{k=1}^{K} X_k \left(\int_{T_{ref}}^{T} C_{pk} dT + h_{fk}^0 \right) - \frac{p}{\rho} + \frac{1}{2}\left(u^2 + v^2\right) \tag{10}$$

B. Chemical Model

Elementary reversible (or irreversible) reactions involving *K* chemical species can be represented in the general form:

$$\sum_{k=1}^{K} \nu'_{ki} X_k \Leftrightarrow \sum_{k=1}^{K} \nu''_{ki} X_k \qquad (i= 1,.....I) \tag{11}$$

The production rate $\dot{\omega}_k$ of the kth species can be written as a summation of the rate-of-progress variables for all reactions involving the kth species:

$$\dot{\omega}_k = \sum_{i=1}^{I} \nu_{ki} q_i \qquad (k=1,.....K) \tag{12}$$

The rate progress variable q_i for the *i*th reaction is given by the difference of the forward and reverse rates as

$$q_i = k_{fi} \prod_{k=1}^{K} \left[X_k \right]^{\nu'_{ki}} - k_{ri} \prod_{k=1}^{K} \left[X_k \right]^{\nu''_{ki}} \qquad (k=1,.....K) \tag{13}$$

The forward rate constants for the *i*th reactions are generally assumed to have the following Arrhenius temperature dependence:

$$k_{fi} = A_i T^{\beta_i} \exp\left(\frac{-E_i}{R_c T} \right) \tag{14}$$

The reverse rate constant k_{ri} are related to the forward rate constant through the equilibrium constant (k_{ci}) by:

$$k_{ri} = \frac{k_{fi}}{k_{ci}} \tag{15}$$

The equilibrium constant k_{pi} are obtained with the relationship

$$k_{pi} = \exp\left(\frac{\Delta S_i^0}{R} - \frac{\Delta H_i^0}{RT} \right) \tag{16}$$

III. Solution Method

The system of non-linear discretized governing equations can be written in the form:

$$\hat{R}(\hat{W}) = 0 \tag{17}$$

where $\hat{R}$ is the residual vector and is defined as

$$\hat{R}(\hat{W}) = \frac{\partial \hat{F}(\hat{W})}{\partial \xi} + \frac{\partial \hat{G}(\hat{W})}{\partial \eta} + \hat{H} - \hat{S} \tag{18}$$

Expanding $R(\hat{W})$ in a Taylor series about (n)[th] iteration and discarding high order (or nonlinear) terms yields:

$$\hat{R}^{n+1}(\hat{W}) = \hat{R}^n(\hat{W}) + \left(\frac{\partial \hat{R}}{\partial \hat{W}} \right)^n \Delta \hat{W}^n \tag{19}$$

where $\frac{\partial \hat{R}}{\partial \hat{W}}$ is the Jacobian matrix. Solving above equation for $\hat{R}^{n+1}(\hat{W}) = 0$ formulates Newton's Method as:

$$\left(\frac{\partial \hat{R}}{\partial \hat{W}}\right)^n \Delta \hat{W}^n = -R(\hat{W}^n) \tag{20}$$

The new values of flow variable vector $\hat{W}$ at the (n+1)[th] iteration can be calculated as:

$$\hat{W}^{n+1} = \hat{W}^n + \Delta \hat{W}^n \tag{21}$$

In the solution of equations with Newton's method, the evaluation of the flux Jacobian matrix is needed. The entries of Jacobian matrix are the derivatives of the residual vector with respect to the flow variables vector. In the calculation of these derivatives a finite difference method or analytical derivation method can be used, and the resulting matrices are called numerical or analytical Jacobians, respectively.

A. Numerical Jacobian Calculation

Using a small finite-difference perturbation magnitude ε, the numerical Jacobian can be calculated by the forward-difference method as follows[11]

$$\frac{\partial \hat{R}_m}{\partial \hat{W}_n} = \frac{\Delta \hat{R}_m(\hat{W})}{\Delta \hat{W}_n} = \frac{\hat{R}_m(\hat{W} + e_n \varepsilon) - \hat{R}_m(\hat{W})}{\varepsilon}$$

$$where \quad m = 1, mmax \quad and \quad n = 1, (mmax + nbound) \tag{21}$$

where, $\hat{R}_m$ is the m^{th} component of the residual vector and the $\hat{W}_n$ is the n^{th} component of the flow variable vector, e_n is the n^{th} unit vector. The value of the n^{th} component of the unit vector e_n is one, and all other components are zero. The size of the residual vector is defined by *mmax*. The size of the flow variable vector is larger as much as the number of boundary cells, *nbound*. In the numerical method, Jacobian evaluation does not require the large coding effort as needed in the analytical method. The same residual discretization is used for both the original and perturbed flow variables. This reuse of the same code is one of the important advantages of the numerical approach. Moreover, for cases in which the analytical derivation is difficult, numerical Jacobian can be obtained without any difficulty.

The inaccuracy and long computation time are the two main disadvantages of the numerical Jacobian evaluation. The error in numerical Jacobian is function of the finite-difference perturbation magnitude. The accuracy of the numerical Jacobians can be improved with the usage of an optimum perturbation magnitude that minimizes the total error in the finite difference evaluation.

B. Accuracy of Numerical Jacobians

In numerical Jacobian calculation, mainly two types of errors occur. These are truncation and condition errors[12]. While truncation error is due to neglected terms in the Taylor's series expansion, condition error is caused by loss of computer precision. The error in numerical Jacobian is highly dependent on perturbation magnitude, ε. For the small values of perturbation magnitude the condition error grows up and dominates the total error. On the other hand as the magnitude of perturbation gets larger, the truncation error becomes dominantly larger. Hence, there should be an optimal value for the perturbation magnitude that minimizes the total error in numerical Jacobian and it can be estimated by following relation[12].

$$\varepsilon_{OPT} = 2\sqrt{\varepsilon_M} \tag{22}$$

where ε_M is the machine epsilon, which depends on the computer processor and compiler A reasonable estimate of ε_M can be given as follows:

$$\varepsilon_M = \frac{1}{2^m} \quad \text{such that} \quad 1 + \varepsilon_M > 1 \tag{23}$$

where m is the number of possible highest bits in the binary representation of the mantissa. . The machine epsilon ε_M values of the compiler-computer configuration are found according to Equation 23 as $\varepsilon_M \cong 1.19 \times 10^{-7}$ for single precision , and $\Sigma_M \cong 2.2 \times 10^{-16}$ for double precision computations.

To examine the validity of the estimation formula given in Equation 22, the optimum perturbation magnitude is also evaluated by a trial error process. Using the Euler equations the numerical flux Jacobian matrices are evaluated with wide range of perturbation magnitudes and compared with the analytically derived ones. The difference between the numerical and the analytical matrices is defined as error. The change of the error with variation of perturbation magnitude is presented in Figure-1. Analyzing the variation of error given in Figure-1 it is observed that, the most accurate numerical Jacobian evaluation can be achieved by the usage of perturbation magnitude approximately equal to 3×10^{-8}. The optimum value obtained from the Equation 22 is compared with the results of the trial error procedure in Table-1.

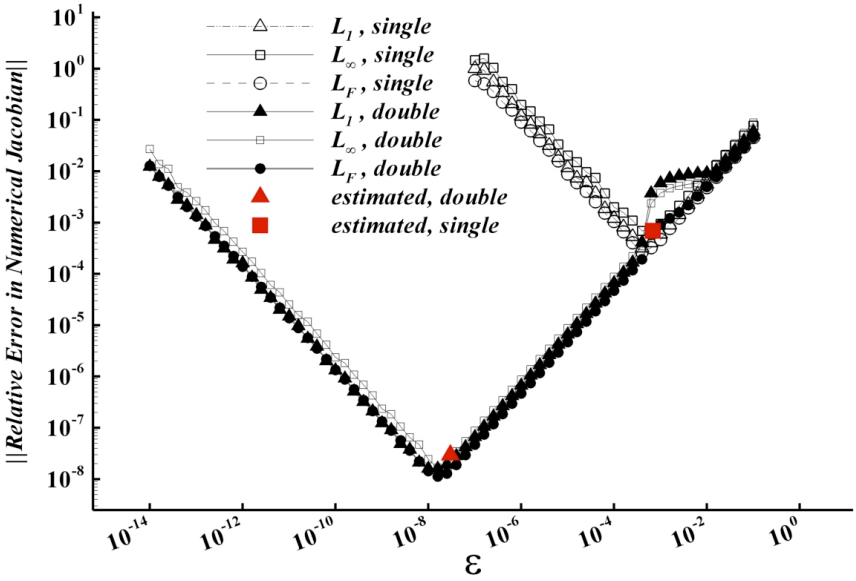

Figure 1 Variation of relative error in numerical Jacobian with finite difference perturbation magnitude

Table 2 Estimated and evaluated optimum perturbation magnitudes

	ε_{OPT} (Equation 22)	ε_{OPT} (Plotted)	minimum relative error (Plotted)
L_1	2.97×10^{-8}	2.51×10^{-8}	2.10×10^{-8}
L_∞	2.97×10^{-8}	2.51×10^{-8}	2.37×10^{-8}
L_F	2.97×10^{-8}	2.51×10^{-8}	1.47×10^{-8}

C. Structure of Jacobian Matrix and Solution Strategies

The most of the Jacobian matrix entries equal to zero since the discretized residual equations only depend on state variables in the neighboring cells. In this study, the UMFPACK (Unsymmetric-pattern MultiFrontal PACKage) sparse matrix solver package[13] is used in order to solve the linear system of equations. In this method, the full matrix is converted into sparse storage mode and then factorized using a sequence of small dense frontal matrices by LU factorization.

Solution by Newton's method requires a good initial guess for convergence. This is a considerable drawback. In this study, flow variables are initialized with their free-stream values, although it may be a poor guess. In order to improve the stability a time-like term is added to the diagonal of the Jacobian matrix and it is made more diagonally dominant. Although this modification is useful in early iterations the convergence rate is no longer quadratic. As the solution becomes more accurate, the diagonal term addition may not be needed. The

withdrawal of the diagonal term from the matrix at the proper convergence level significantly reduces the number of iterations and CPU time required to reach final solution. Initial and the withdrawal values of diagonal term that gives the best convergence performance are chosen by trial-error.

With the addition of a time-like term, the modified Newton's method becomes:

$$\left(\frac{1}{\Delta t}[I]+\frac{\partial \hat{R}}{\partial \hat{W}}\right)^{n} \Delta \hat{W}^{n} = -R(\hat{W}^{n}) \tag{24}$$

The original Newton's method can be constructed as $\Delta t \rightarrow \infty$. In the modified Newton's method, a small initial value Δt_0 is chosen and a new value of Δt can be obtained using L_2-norm of the residuals as

$$\Delta t^{n} = \Delta t^{0} \frac{\left\| R(\hat{W}^{0}) \right\|_2}{\left\| R(\hat{W}^{n}) \right\|_2} \tag{25}$$

Convergence performance of the Newton's method solver developed in this study is presented by the residual history graph given in Figure-2

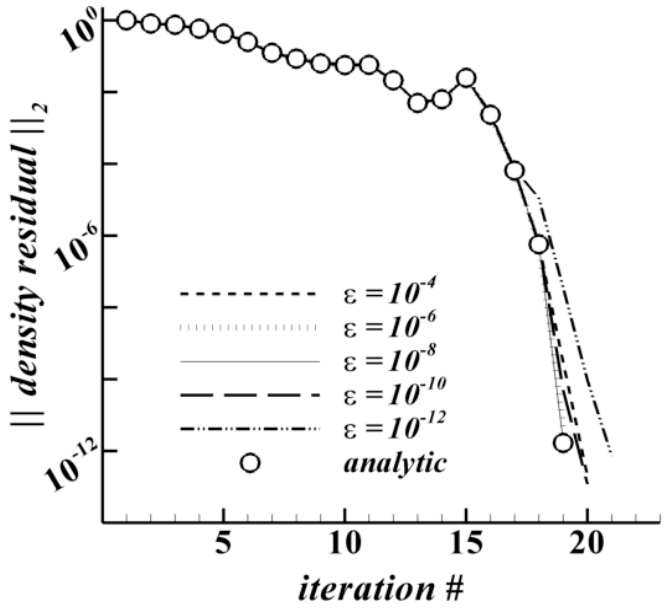

Figure 2 Residual history of the solver using Van Leer scheme

IV. Design optimization

In general, an optimization problem can be defined in the following form:

Objective function:

$$F\left[\hat{W}\left(D_i\right), X\left(D_i\right), D_i\right] \quad (\text{i} = 1, \ldots \text{I}) \tag{26}$$

Constraint functions:

$$G_j\left[\hat{W}\left(D_i\right), X\left(D_i\right), D_i\right] \quad (\text{j} = 1, \ldots \text{J}) \tag{27}$$

In above functions, $\hat{W}$, *X, D, I,* and *J*, state or flow variables, grid coordinates, design variables, numbers of design variables and constraint functions, respectively. In gradient base design optimization sensitivities are needed. Sensitivities can be defined as the derivatives of objective functions with respect to design variables. The accurate and efficient calculation of sensitivities improves the performance of the design optimization. The convergence of design optimization depends on the accuracy of sensitivities. In design optimization most of the CPU time is spend in calculation of sensitivities. The efficient calculation of sensitivities reduces the CPU time spend for design.

Sensitivities can be calculated by finite-difference or analytical methods. The implementation of the finite-difference method is simple any finite-difference stencil can be used to calculate the sensitivities. However, the finite-difference method has some disadvantages. The accuracy of finite-difference method depends on the size of the finite-difference perturbation magnitude. The finite difference method requires the solution of flow equations for each design variable perturbation and this increases the CPU time significantly. Compare to finite-difference methods, analytical methods are more accurate and efficient. In analytical method, a sensitivity code is developed by taking the derivatives of flow equations with respect to design variables. The main disadvantage of analytical method is the effort and time spent for code development. In this study, analytical method is used for sensitivity calculations. In analytical method, the sensitivities can be calculated with two different methods. These methods are direct differentiation and adjoint methods[14].

A. Direct Differentiation Method

In direct differentiation method, the objective function is differentiated with respect to design variables in the following form:

$$\frac{dF}{dD_i} = \frac{\partial F}{\partial \hat{W}} \cdot \left(\frac{d\hat{W}}{dD_i} \right) + \frac{\partial F}{\partial X} \cdot \left(\frac{dX}{dD_i} \right) + \frac{\partial F}{\partial D_i} \tag{28}$$

In above equation, the derivatives of grid coordinated with respect to design variables, dX / dD_i, are easy to evaluate. However, the derivatives of flow variables with respect to design variables, $d\hat{W} / dD_i$, can only be calculated by using a sensitivity code. In direct differentiation method, the sensitivity code is developed by taking the derivatives of discretized flow equations with respect to the design variables. The discretized flow equations can be written in the following form:

$$\hat{R}(\hat{W}(D_i), X(D_i), D_i) = 0 \tag{29}$$

Analytically differentiating above equation with respect to design variables gives:

$$\left[\frac{\partial \hat{R}}{\partial \hat{W}} \right] \left(\frac{d\hat{W}}{dD_i} \right) + \left[\frac{\partial \hat{R}}{\partial X} \right] \left(\frac{dX}{dD_i} \right) + \frac{\partial \hat{R}}{\partial D_i} = 0 \tag{30}$$

In order to solve the flow variable sensitivities, $d\hat{W} / dD_i$, above equation can be rearranged in the following form:

$$\left[\frac{\partial \hat{R}}{\partial \hat{W}} \right] \left(\frac{d\hat{W}}{dD_i} \right) = - \left[\frac{\partial \hat{R}}{\partial X} \right] \left(\frac{dX}{dD_i} \right) - \frac{\partial \hat{R}}{\partial D_i} \tag{31}$$

The advantage of this method is that the Jacobian matrix is already calculated and factorized in the solution of flow equations, and the flow variable sensitivities can be calculated very efficiently by solving the same matrix with different right hand sides. the sensitivities of objective function can be evaluated by substituting the flow variable sensitivities into Equation 28.

B. Adjoint Method

In adjoint method, the discretized flow equations can be considered as a constraint function to be satisfied. By multiplying Equation 30 by Lagrange multipliers, Λ, and adding to Equation 28, the sensitivities of objective function can be calculated with the following relation:

$$\frac{dF}{dD_i} = \frac{\partial F}{\partial \hat{W}}^T \cdot \left(\frac{d\hat{W}}{dD_i} \right) + \frac{\partial F}{\partial X} \cdot \left(\frac{dX}{dD_i} \right) + \frac{\partial F}{\partial D_i} + \Lambda^T \left\{ \underbrace{\left[\frac{\partial \hat{R}}{\partial \hat{W}} \right]^T \left(\frac{d\hat{W}}{dD_i} \right) + \left[\frac{\partial \hat{R}}{\partial X} \right] \left(\frac{dX}{dD_i} \right) + \frac{\partial \hat{R}}{\partial D_i}}_{= 0 \text{ from Eq.(30)}} \right\} \tag{32}$$

In above equation, the term multiplied by Lagrange multipliers is already zero from Equation 30. Hence, the sensitivities of objective function calculated with Equations 28 and 32 have the same value. Above equation can be rearranged in the following form:

$$\frac{dF}{dD_i} = \frac{\partial F}{\partial X} \cdot \left(\frac{dX}{dD_i} \right) + \frac{\partial F}{\partial D_i} + \Lambda^T \left\{ \left[\frac{\partial \hat{R}}{\partial X} \right] \left(\frac{dX}{dD_i} \right) + \frac{\partial \hat{R}}{\partial D_i} \right\} + \left\{ \underbrace{\left[\frac{\partial \hat{R}}{\partial \hat{W}} \right]^T \Lambda + \frac{\partial F}{\partial \hat{W}}^T}_{\text{choose as zero}} \right\} \left(\frac{d\hat{W}}{dD_i} \right) \tag{33}$$

In above equation, by choosing the summation of terms in the last parenthesis as zero, the sensitivities of objective function can be written as:

$$\frac{dF}{dD_i} = \frac{\partial F}{\partial X} \cdot \left(\frac{dX}{dD_i} \right) + \frac{\partial F}{\partial D_i} + \Lambda^T \left\{ \left[\frac{\partial \hat{R}}{\partial X} \right] \left(\frac{dX}{dD_i} \right) + \frac{\partial \hat{R}}{\partial D_i} \right\} \tag{34}$$

The advantage sensitivity calculation with adjoint method is that the Jacobian matrix is not solved for each design variables. The Jacobian matrix is solved only for the calculation of Lagrange multipliers. Once the Lagrange multipliers are evaluated, Equation 34 is used for the calculation of sensitivities for all design variables. In the calculation of Lagrange multipliers the following system of equations is solved

$$\left[\frac{\partial \hat{R}}{\partial \hat{W}} \right]^T \Lambda = - \frac{\partial F}{\partial \hat{W}}^T \tag{35}$$

C. Accuracy of the Sensitivity Evaluation

Although the implementations of direct differentiation and adjoint methods are different, the common property of these methods is the need for the calculation of Jacobian matrix. Hence the accurate and efficient calculation of Jacobian matrices is important in sensitivity analysis as in flow analysis. The difference between the sensitivities that are evaluated from numerical and analytical Jacobian is defined as the sensitivity error. To be able to present the total error in the sensitivity vector, vector norm definitions are used.

$$\left(Error_{\text{sensitivity}} \right)_i = \left(\frac{\partial W}{\partial \beta_k}_{\ numeric\ jacobian} \right)_i - \left(\frac{\partial W}{\partial \beta_k}_{\ analytic\ jacobian} \right)_i \tag{36}$$

The relative error in the numerical sensitivity vector is defined as:

$$\left(Error_{\text{sensitivity}} \right)_{relative} = \frac{\left\| Error_{\text{sensitivity}} \right\|}{\left\| \frac{\partial W}{\partial \beta_k}_{\ by\ analytic\ jacobian} \right\|} \tag{37}$$

The variation of the relative sensitivity error by the finite differencing perturbation magnitude used in Jacobian evaluation is given in Figure-3. Comparison of the Figures 3 with the Figure 1 shows that; same perturbation magnitude minimizes both of the error in the numerical Jacobian and sensitivity evaluations. The optimum perturbation magnitudes presented by those plots are in good agreement with the one approximated by Equation 22.

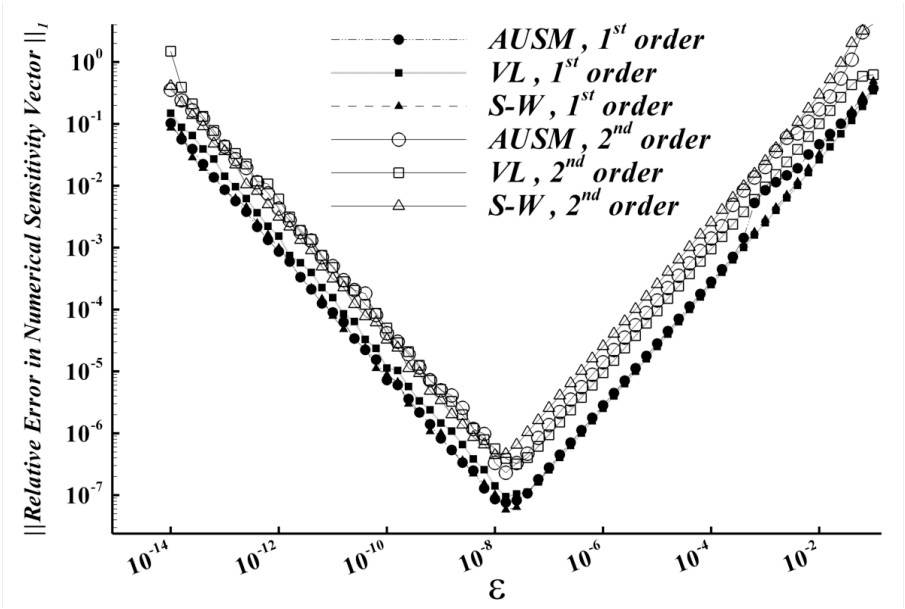

Figure 3 Variation of sensitivity error with respect to finite difference perturbation magnitude used in Jacobian evaluation

V. Results

A. Base Geometry Results

The initial baseline nozzle has an axisymmetric contour with an area ratio of 6.1 whose expansion allows having a description of the fundamental behavior of the chemical kinetics in the nozzle[15]. The inlet flow condition is given in Table 3. At the inlet section, the mass fraction of species is evaluated by assuming equilibrium condition in the combustion chamber with specified pressure and temperature. Aim of this test case is to evaluate the effects of subsonic (convergent part) and supersonic (divergent part) flow regimes on the chemical kinetics in converging diverging nozzle. Throat of the nozzle occurs at x/L = 0.2568, where L is the nozzle length.

Table 3 Nozzle inlet flow conditions

$Y_{H_2O} = 0.390$	$Y_{CO_2} = 0.211$	$Y_{CO} = 0.289$	$Y_{H_2} = 0.012$
$Y_{OH} = 0.054$	$Y_{O_2} = 0.029$	$Y_{O} = 0.01$	$Y_{H} = 0.00021$
$T_t = 3000$ K		$P_t = 17$ MPa	

A 81x21 grid is used for the computations. In Figure 4 and 5 iso-contours are reported for Mach number, temperature and mass fractions of H_2, H_2O, CO and CO_2 are. As expected they respectively increase and decrease regularly with axial distance.

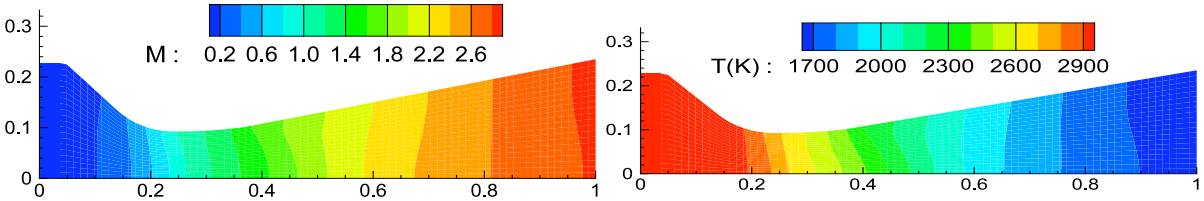

Figure 4 Distributions of Mach number and temperature in the baseline nozzle configuration

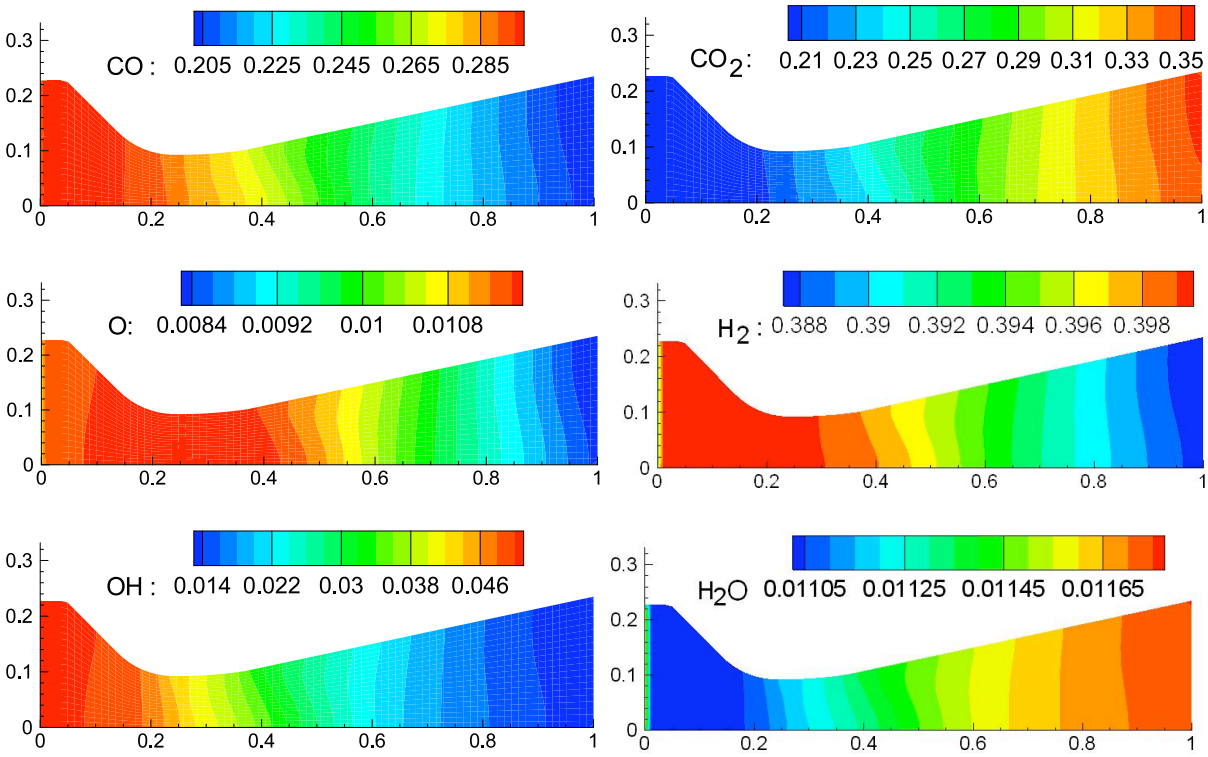

Figure 5 Distributions of mass fractions of CO, CO_2, O, H_2, OH and H_2O in the baseline nozzle configuration

Above figures show that Mach number increases and temperature decreases in axial direction. As the fluid accelerates towards the throat area the averaged species concentration starts to change. Flow upstream of $x/L = 0.18$ shows very small variation of properties. As an overall result H_2 and CO_2 are produced, while other concentrations decay with axial distance towards the throat, area averaged species concentration start to change. In convergent, where T is still high, the first reaction destroys H rapidly and forms OH.

B. Design Results

The goal of the present design is to generate a nozzle geometry that produces the maximum thrust at a specified combustion chamber condition. Hence, the objective function is defined as:

$$F = \int_{Aexit} \left(\rho u^2 + p\right) dA \tag{38}$$

The sensitivity derivatives are evaluated by adjoint method in the developed code. The numerical optimization of the present study employs a commercial optimization package[16]. A total of ten Hicks-Henne functions are used to change nozzle geometry starting from throat to exit [17]. The throat and exit areas are fixed. The design is completed in three iterations and thirteen function calls. Last, the designed nozzle is analyzed by using Euler and reaction equations.

Table 3 compares the thrust of baseline and optimized nozzles. Approximately 2.6% of thrust increase is achieved in the optimized nozzle. Figure 5 shows the distribution of Mach number and temperature in optimized nozzle. As shown in figures, the design optimization produced a bell shape nozzle. Figure 6 shows the distribution of mass fraction of species. As in the baseline nozzle the mass fractions of H_2O and CO_2 increase, and the mass fractions of other species decreases in axial direction.

Table 4 The results of two nozzles

	Baseline	Optimized
Expansion ratio	6.45	6.45
Nozzle throat diameter , D_t (m)	0.021	0.021
Vacuum thrust, F (kN)	87.08	89.31

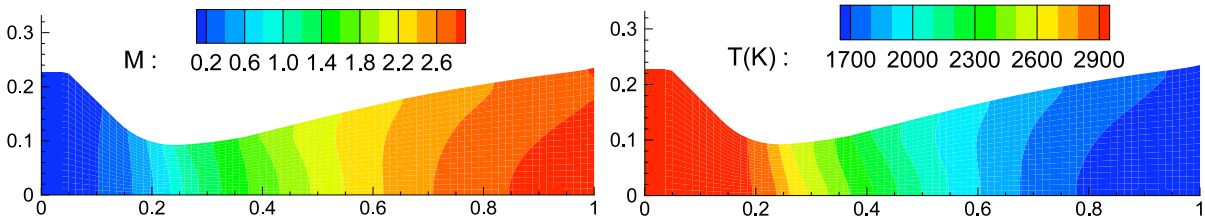

Figure 6 Distributions of Mach number and temperature in the optimized nozzle configuration

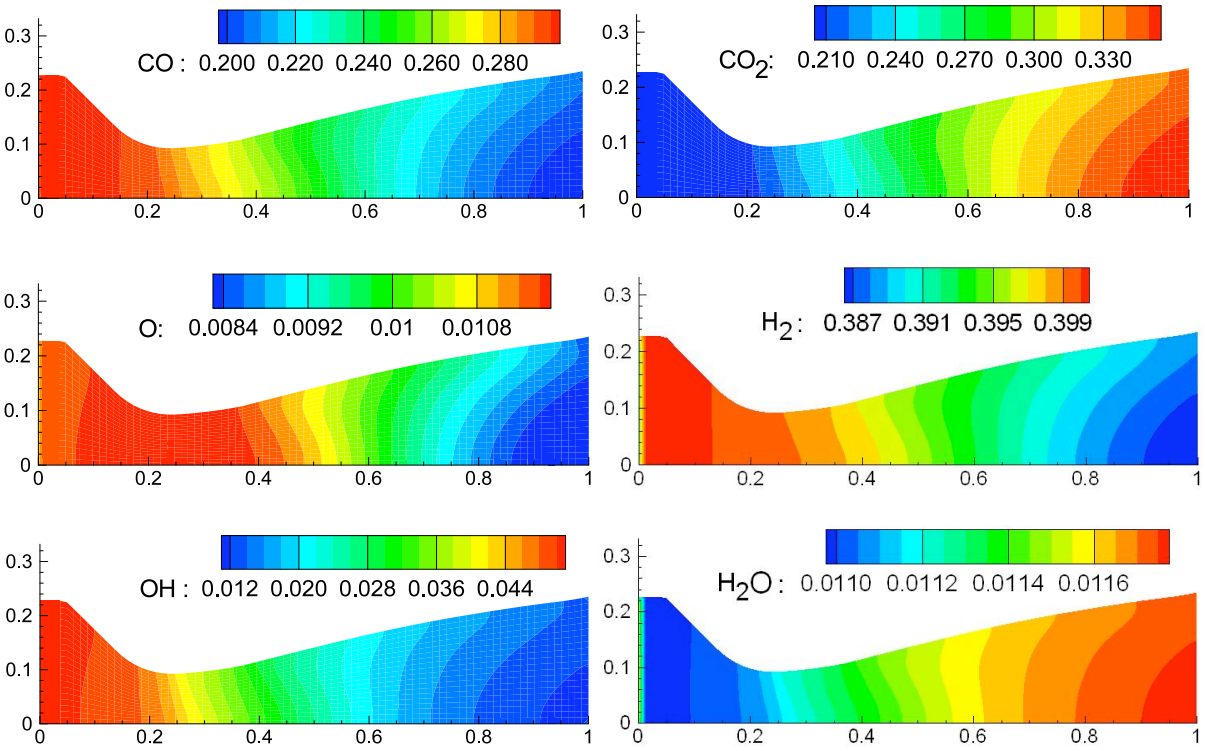

Figure 7 Distributions of mass fractions of CO, CO_2, O, H_2, OH and H_2O in the optimized nozzle configuration

VI. Conclusion

Design optimization application aiming the performance improvement of the rocket nozzle in chemically reacting flow is demonstrated. Exact Newton's method is implemented successfully for the solution of the large linear system of equations of flow and sensitivity analysis. Accurate and efficient sensitivity evaluations are performed both by the adjoint and direct differentiation methods with the utilization of direct sparse matrix solver. Numerical Jacobians, which are evaluated with minimum error, are employed for the Newton's method solution. During the design optimization Euler equations are analyzed and the final increment in thrust is calculated by solving Euler and chemical reaction equations for the baseline and the optimized geometries. A small improvement is achieved in thrust produced by the nozzle. A larger increase in thrust can be achieved by solving the Euler and reaction equations simultaneously during design optimization. The performance of the solver may also be improved by the usage of analytical Jacobian in the solution of the flow and chemical reaction equations.

Acknowledgement

This research was supported by Turkish Scientific and Technological Research Council through the project 'TUBITAK/107M103'. The authors gratefully acknowledge the support given.

References

[1] Rao, G.V.R., 'Exhaust Nozzle Contour for Optimum Thrust,' Jet Propulsion, Vol. 28, No. 6, 1958.

[2] Farley, J.M. and Campbell, C.E., 'Performance of Several Method of Characteristics Exhaust Nozzles,' NASA TN D293, October 1960.

[3] 'Liquid Rocket Engine Nozzles,' NASA SP-8120, July 1976.

[4] Xing, X. Q. and Damodaran, M. 'Design of Three-dimensional Nozzle Shapes Using Hybrid Optimization Techniques, AIAA Paper-2004-26, 42nd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, Jan. 5-8, 2004

[5] Guobiao C., Jie F., Xu X. and Minghao L. , 'Performance prediction and optimization for liquid rocket engine nozzle,' Science and Technology Vol. 11 pp. 155–162, 2007

[6] Onur Ö, Eyi, S., 'Effects of the Jacobian Evaluation on Newton's Solution of the Euler Equations,' International Journal for Numerical Methods in Fluids, Vol. 49, pp. 211-231, 2005.

[7] Ezertaş. A, Eyi S. 'Performances of Numerical and Analytical Jacobians in Flow and Sensitivity Analysis' AIAA Paper 2009-4140, 19th AIAA Computational Fluid Dynamics Conference, San Antonio TX, June 22 -25, 2009.

[8] Van Leer, B., "Flux Vector Splitting for the Euler Equations", ICASE Report 82-30, September 1982.

[9] Van Leer, B., "Towards the Ultimate Conservative Difference Scheme, V. A Second Order Sequel to Godunov's Method", Journal of Computational Physics, Vol.32.1979, pp. 101–136

[10] Van Albada, G.D., Van Leer, B., Roberts, W.W., "A Comparative Study of Computational Methods in Cosmic Gas Dynamics", *Astronomy and Astrophysics*, Vol 108, 1982, pp. 76-84.

[11] Dennis J.E., Schnabel R.B., Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice Hall, New Jersey, 1983.

[12] Gill, P.E., Murray W., and Wright M.H., Practical Optimization, Academic Press, London, 1992

[13] Davis, T. A., UMFPACK Version 4.1 User Manual, University of Florida, Florida, 2003

[14] adjoint

[15] Yumuşak M., Vuillot F., Tınaztepe T "Viscous Flows in Solid Propellant Rocket Motors", 42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Sacramento, USA, 2006

[16] Vanderplaats, G. N. and Hansen, S. R., 'DOT User Manual,' VMA Engineering , Goleta, CA, 1989.

[17] Eyi S., Hager, J. O., and Lee K. D., "Airfoil Design Optimization Using the Navier-Stokes Equations", Journal of Optimization Theory and Applications, Vol. 83, No. 3, pp. 447-461, December 1994.