AIAA JOURNAL
Vol. 44, No. 7, July 2006

Neural Network Modeling of Deterministic Unsteadiness Source Terms

Bojan Lukovic,[*] Paul D. Orkwis,[†] and Mark G. Turner[‡]
University of Cincinnati, Cincinnati, Ohio 45221-0070
and
Balu Sekar[§]
U.S. Air Force Research Laboratory, Wright–Patterson Air Force Base, Ohio 45433-7251

A neural-network-based lumped deterministic source term technique is presented that results in the prediction of an approximate time-average solution when used to modify a steady-state solver. Three different neural networks are developed for simple cavity flows using Mach number, cavity length-to-depth ratio, and aft wall translation as parameters. The results indicate that axial force data can be reproduced with less than 15% error as compared to the time average of a fully unsteady calculation. Computation times for the resultant neural network lumped deterministic source term approach were up to two orders of magnitude less than the comparable unsteady solution and were essentially identical to that of a steady-state calculation, although it must be noted that a database of unsteady calculations is required to develop the technique. The lumped deterministic source terms did not appear to affect the robustness of the steady-state solver adversely.

Nomenclature

A	=	control surface area
$\boldsymbol{a}$, a_i^n	=	neural network output neuron
b_i^l	=	neural network bias
$\boldsymbol{c}$, c_i	=	neural network input neuron
D	=	cavity length
E	=	error vector function
$\boldsymbol{e}$	=	neural network error vector
e_t	=	total energy
$\boldsymbol{F}$	=	force vector
f	=	function
k	=	turbulent kinetic energy
L	=	cavity length
L/D	=	cavity length-to-depth ratio
M	=	Mach number
$\boldsymbol{n}$	=	normal direction vector
n_i^j	=	neural network neuron
$\boldsymbol{P}$	=	primitive variable vector
p	=	pressure
$\boldsymbol{Q}$	=	conserved variable vector
R_i	=	ith residual operator
R^j	=	real space of dimension j
T	=	temperature
t	=	time
u	=	x-direction velocity
v	=	y-direction velocity
$[\boldsymbol{W}]$	=	weight factor matrix
w	=	z-direction velocity
(x, y, z)	=	Cartesian coordinate directions
y^+	=	nondimensional turbulent near-wall distance
γ	=	ratio of specific heats
ε	=	turbulent dissipation
μ	=	molecular viscosity
μ_t	=	turbulent viscosity
ρ	=	density
υ	=	kinematic viscosity

Subscripts

wall	=	wall value
0	=	stagnation conditions

Superscripts

-	=	time average
=	=	derived quantity time average
$\sim$	=	deterministic fluctuation
$\prime$	=	stochastic fluctuation

Introduction

UNSTEADINESS is a natural condition of fluid flow and must be considered to simulate accurately most flowfields. However, it is often the case that detailed unsteadiness is not of paramount interest to the simulator but rather its effect on the time-average solution. This idea is utilized in efforts that model the random unsteadiness associated with turbulence but can be extended to model so-called deterministic unsteadiness, that is, nonrandom unsteadiness that can be simulated, if one is interested in the time-averaged effect of unsteadiness rather than its details.

Simulation of self-sustaining unsteady flows such as those about cavities can take as many as 70 characteristic flowfield times before convergence of the deterministic unsteadiness, that is, tens of thousands of iterations. One might be tempted to use a steady-state procedure to obtain the time-averaged solution; however, the two satisfy different governing equations, and, hence, one cannot be obtained from the other. This difference can be overcome by augmenting the steady-state equations with source terms that represent the effect of unsteadiness in the time-average solution. These source terms are typically called lumped deterministic source terms (LDST) because they combine all of the relevant deterministic unsteadiness

Received 30 July 2003; revision received 14 December 2005; accepted for publication 16 December 2005. Copyright © 2006 by the authors. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0001-1452/06 $10.00 in correspondence with the CCC.

[*] Graduate Research Assistant, ML70, Department of Aerospace Engineering and Engineering Mechanics; Bojan.Lukovic@UC.Edu. Student Member AIAA.

[†] Professor and Director of Graduate Studies, ML 70, Department of Aerospace Engineering and Engineering Mechanics; Paul.Orkwis@UC.Edu. Associate Fellow AIAA.

[‡] Professor, ML 70, Department of Aerospace Engineering and Engineering Mechanics; Mark.Turner@UC.Edu. Associate Fellow AIAA.

[§] Aerospace Engineer, Propulsion Directorate, Turbine Engine Division, Room D026, Building 18, 1950 Fifth Street; Balu.Sekar@wpafb.af.mi. Associate Fellow AIAA.

terms into one value for each equation. They can be applied in a manner similar to turbulence closure schemes for stochastic unsteadiness. These source terms can be formed directly from the time average of an unsteady solution; however, like turbulence modeling, the approach has merit if a simpler model of the LDSTs can be found. It should be obvious that the LDSTs cannot provide a general means to include unsteady effects in all flows but rather hold the possibility that classes of flows with similar behavior might be modeled by a single function. The technique becomes attractive if one considers small-scale phenomenon/features with respect to a main flow, for example, wall cooling jets, surface gaps and bumps, etc., items that are sometimes excluded from simulation but that exist in actual hardware. The technique is also useful if the source terms can be calculated via a lower-fidelity technique, as applied by others.

The LDST technique has been used in this latter fashion by Sondak et al.[1] and Busby et al.[2] to model rotor–stator interaction unsteadiness in turbomachinery. They performed unsteady inviscid simulations to extract the lumped deterministic stress field and then applied it as a source term in steady viscous calculations to model unsteady flow effects. This technique was successful in reducing the overall time required to obtain the approximate time-average solution but still required the use of a time-consuming unsteady inviscid calculation.

Adamczyk et al.[3] developed an approach based on the average passage equations to model the time-averaged flowfield through the blade passages of multistage turbomachinery. In that case, the source terms represented a circumferential average solution of an adjacent blade row, thereby including blade–row interaction effects. However, this method solves a steady solution in a coordinate system relative to the blade and does not model true unsteadiness.

Meneveau and Katz[4] also employed the average passage formulation to include cyclic deterministic contributions in a rotor–stator interaction problem. Their model was compared to experimental values of the deterministic stresses.

Ning and He[5] calculated the vortex-shedding unsteady stresses from unsteady solutions and then solved the time-averaged Navier–Stokes equations together with the unsteady stresses. They showed that the time-average of vortex shedding could be obtained with a time-independent solver by this technique. However, none of the aforemention researchers studied the source terms further with the idea of modeling them so that additional simulations could be avoided.

The current work presents a simple approach to modeling the LDSTs of deterministic unsteadiness for a specific class of flows. In this case, a neural network is trained from detailed unsteady simulations to define the LDSTs over a range of parameters, so that, given local conditions, the neural network returns approximate source terms for each equation and grid cell. The technique is used to modify a steady Navier–Stokes solver so that it models the time average of an unsteady Navier–Stokes solution.

The test case was simple cavity flow, which, although geometrically very simple, exhibits a self-sustained unsteadiness. The goal of this work was to model the time-average effect of cavity unsteadiness on the adjacent flow. Hence, the modeled time-average solution does not simulate the cavity geometry. Several neural network LDST models are developed. The first is for an $L/D = 1$ cavity over a range of subsonic Mach numbers, the second extends the first to cavities with a range of L/D values, and the third extends the first to cavities with translating aft walls. Note that this test case was chosen to highlight the ability of the technique to handle a flow in which unsteadiness is present throughout the entire solution domain. The technique may be more useful for cases where the unsteadiness occurs in an important but small part of the flowfield, for example, flow control devices. However, the ability to model the former demonstrates the ability to model the latter, hence, the more stringent test case is the better example problem.

This paper presents the approach by first describing the mathematical basis for the LDST idea. The details of the neural network approach used to model the source terms are then presented. The important details of CFD++, the Navier–Stokes solver used in this work, are presented next, followed by a short discussion of a technique used to judge the accuracy of the final modeled solution. Finally, the results obtained with this technique are given and compared to results obtained from unsteady computations to demonstrate its feasibility and ascertain its accuracy.

LDST Approach

The following paragraphs illustrate the LDST concept, including how they are obtained and used to create quasi-time-average solutions. The technique is developed from the inviscid form of the two-dimensional governing equations; viscous terms, turbulence equations, and the third dimension have been dropped for clarity, but are included in the actual equation set used in this work.

The LDST methodology is obtained by first considering the unsteady governing equations:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) = 0$$

$$\frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(\rho u^2 + p) + \frac{\partial}{\partial y}(\rho u v) = 0$$

$$\frac{\partial}{\partial t}(\rho v) + \frac{\partial}{\partial x}(\rho u v) + \frac{\partial}{\partial y}(\rho v^2 + p) = 0$$

$$\frac{\partial(\rho e_t)}{\partial t} + \frac{\partial}{\partial x}[(\rho e_t + p)u] + \frac{\partial}{\partial y}[(\rho e_t + p)v] = 0 \tag{1}$$

When the time derivatives are zero, the spatial derivatives represent the steady-state governing equations. For numerical simulations, the discrete form of these terms are often called the solution residual $R_k(\boldsymbol{Q})$, $k = 1, \ldots, 4$, as illustrated hereafter,

$$R_1(\boldsymbol{Q}) = \frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) = 0$$

$$R_2(\boldsymbol{Q}) = \frac{\partial}{\partial x}(\rho u^2 + p) + \frac{\partial}{\partial y}(\rho u v) = 0$$

$$R_3(\boldsymbol{Q}) = \frac{\partial}{\partial x}(\rho u v) + \frac{\partial}{\partial y}(\rho v^2 + p) = 0$$

$$R_4(\boldsymbol{Q}) = \frac{\partial}{\partial x}[(\rho e_t + p)u] + \frac{\partial}{\partial y}[(\rho e_t + p)v] = 0$$

$$\boldsymbol{Q} = [\rho,\ \rho u,\ \rho v,\ \rho e_t]^T \tag{2}$$

Note that the conserved variable vector $\boldsymbol{Q}$, which satisfies Eqs. (2), is not the same as the time average of the unsteady solution $\bar{\boldsymbol{Q}}$. The difference between them can be observed, and the LDSTs defined by considering the vector residual operator $R_k(\boldsymbol{Q})$.

To understand this better, start by considering the conservative-variable vector $\boldsymbol{Q}$ and represent it as

$$\boldsymbol{Q} = \bar{\boldsymbol{Q}} + \tilde{\boldsymbol{Q}} + \boldsymbol{Q}' \tag{3}$$

where $\bar{\boldsymbol{Q}}$ is the time-averaged value over a large time ΔT, $\tilde{\boldsymbol{Q}}$ is the deterministic fluctuation, and $\boldsymbol{Q}'$ is a stochastic fluctuation. The quasi-time-averaged primitive variable vector is then obtained from these time-averaged conservative variables using

$$\boldsymbol{P} = \begin{bmatrix} \bar{\bar{p}} \\ \bar{\bar{T}} \\ \bar{\bar{u}} \\ \bar{\bar{v}} \end{bmatrix} = \begin{bmatrix} (\gamma - 1)\left\{\overline{\rho e_t} - \dfrac{1}{2\bar{\rho}}[(\overline{\rho u})^2 + (\overline{\rho v})^2 + (\overline{\rho w})^2]\right\} \\ \dfrac{(\gamma - 1)}{\bar{\rho} R}\left\{\overline{\rho e_t} - \dfrac{1}{2\bar{\rho}}[(\overline{\rho u})^2 + (\overline{\rho v})^2 + (\overline{\rho w})^2]\right\} \\ \overline{\rho u}/\bar{\rho} \\ \overline{\rho v}/\bar{\rho} \end{bmatrix} \tag{4}$$

The decomposed conservative variables are then substituted into the unsteady governing equations and the stochastic terms ignored for the sake of clarity in this discussion. The stochastic terms are included in the usual Reynolds-averaged Navier–Stokes (RANS) manner via modeling of the Reynolds stress terms. Because these equations are nonlinear, the resulting equation set consists of three distinct sets of terms: terms containing only $\bar{\boldsymbol{Q}}$ products, terms containing only $\tilde{\boldsymbol{Q}}$ products, and terms containing a mixture of $\bar{\boldsymbol{Q}}$ and $\tilde{\boldsymbol{Q}}$ variables. Similar to the Favre decomposition, all of the linear $\tilde{\boldsymbol{Q}}$ terms and the unsteady terms vanish on time averaging, so that one is left with the steady-state residual operator acting on the time-averaged solution plus the time-average of all of the higher-order perturbation terms. It is then the higher-order nonlinear perturbation terms that define the effect of unsteadiness on the steady-state solution. The LDSTs can be obtained from the time-average solution using the simple equation

$$R_k(\bar{\boldsymbol{Q}}) + \mathrm{LDST}_k = 0 \tag{5}$$

Again, note that the first term in Eq. (5) is the usual residual operator R_k, whereas the second term represents the source terms that must be added to the steady-state equations to include the effect of unsteadiness, that is, the LDST. The two solutions, one obtained from the steady-state equation, $R_k(\boldsymbol{Q}) = 0$, and the other the solution of Eq. (5), are different because the second represents the time average of the unsteady solution and includes the effect of unsteadiness on the flow, whereas the first does not.

Interestingly, the continuity equation, because it is linear in the conservative variables, reduces on variable decomposition and time averaging such that the mass source terms are identically zero. This implies that the number of required source terms can be reduced by one.

For completeness, note that for turbulent viscous flows the source terms are again equal to the negative of the steady-state residual and are found in an identical manner by including the turbulence equations with the additional variables ρk and $\rho\varepsilon$.

This approach can be employed in several ways as described by previous authors.[1–5] In the current application, it is applied to remove the cavity from the simulation domain. That is, if one considers Fig. 1, the resulting time-averaged solution is obtained for the flow only above the cavity. The technique replaces the time-averaged effect of the cavity on the main flow with LDSTs. In so doing, an unsteady calculation is avoided.

To summarize, the LDST are found by forming the time mean of the unsteady conservative variable vector, applying the residual operator to it, and utilizing Eq. (5). This technique then allows the time-average of unsteady effects to be included in steady-state simulations. However, a means to predict the source terms is needed to avoid direct computation of the time-average solution and make the approach feasible. Many modeling approaches are possible, ranging from physics-based techniques, such as those used for turbulence modeling, to simple linear interpolation. The results indicated that the source terms are somewhat complicated functions in space that vary with important flow parameters like Mach and Reynolds numbers and the physical geometry itself. Under these circumstances, simple linear interpolation would require a much more significant database than more advanced interpolation techniques like neural networks. The authors chose neural networks because of their ability to model complicated functions while being trained with limited data. In this way, the model can be developed with a minimal number of unsteady calculations.

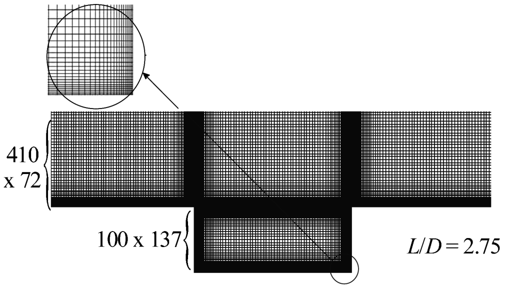


Fig. 1 Meshes for *L/D* = 1 and 2.75.

Neural Network LDST Modeling

This section describes the details of the neural-network-based LDST (NN-LDST) model. A neural network is a system that, due to its topological structure, can adaptively learn nonlinear mappings from input to output space when the network has a large database of prior examples from which to draw. It simulates human functions such as learning from experience, generalizing from previous to new data, and abstracting essential characteristics from inputs containing irrelevant data. It is an ideal way to develop LDSTs without the need for significant physical modeling or insight because the source terms are highly nonlinear functions of the input parameters. Hence, linear interpolation is not an appropriate approach to their modeling, unless each parameter of the data set is divided into an enormous number of small increments.

The basic architecture of a neural network includes layers of interconnected processing units called neurons (comparable to the dendrites in the biological neuron) that transform an input vector $[c_1, c_2, \ldots, c_M]^T$ into an output vector $[a^n_1, a^n_2, \ldots, a^n_S]^T$. Neurons without predecessors are called input neurons and constitute the input layer. All other neurons are called computational units because they are developed from the input layer. A nonempty subset of the computational units is specified as the output units. All computational units that are not output neurons are called hidden neurons. Each interconnection between two neurons $n^{l-1}_i \to n^l_j$ has an associated weight factor w^l_{ij} and bias b^l_i that can be adjusted by using an appropriate learning algorithm such as the Levenberg–Marquardt method (see Refs. 6 and 7). As such, it is a technique similar to least-squares curve fitting. The output of each neuron is

$$f^l\left(\sum_{j=1}^{S^{l-1}} w^l_{ji} a^{l-1}_j + b^l_i\right) \tag{6}$$

where

$$a^{l-1}_j =$$

$$\begin{cases} c_j & \text{if } l = 1 \ (j \text{ is an input unit}) \\ f^{l-1}\left(\displaystyle\sum_{j=1}^{n} w^{l-1}_{ji} a^{l-2}_j + b^{l-1}_i\right) & \text{otherwise} \end{cases} \tag{7}$$

and f^{l-1} is the transfer function.

The universal approximation theorem states that a neural network with one hidden layer utilizing a sigmoid transfer function is able to approximate any continuous function $f\colon R^M \to R^{S_2}$ (where M and S_2 are dimensions of the function domain and range, respectively) in any domain, with a given accuracy based, in part, on the amount of training data. Features of the input data are extracted in the hidden layer with a hyperbolic tangent transfer function and in the output layer with a purely linear transfer function. Based on the theorem and thanks to the topological structure of the neural network, one can generate complex data dependencies without performing time-consuming computations.

However, any neural network application depends on the training algorithm. The learning algorithm is the repeated process of adjusting weights to minimize the network errors. These errors are defined by $\boldsymbol{e} = \boldsymbol{t} - \boldsymbol{a}$, where $\boldsymbol{t}$ is the desired network output vector and $\boldsymbol{a} = \boldsymbol{a}(\boldsymbol{c}, [\boldsymbol{W}])$ is the actual network output vector, a function of the input data and network weights. This weight adjustment is repeated for many training samples and is stopped when the errors reach a sufficiently low level.

The majority of neural network applications are based on the backpropagation algorithm. The term backpropagation refers to the process by which derivatives of the network error, with respect to network weights and biases, are calculated, from the last layer of the network to the first. The Levenberg–Marquardt backpropagation scheme was employed for this purpose.

In this work, the neural networks were trained to generate point source terms for each governing equation based on input parameters x, y, M, L/D, and w_{wall}. Every case computed with unsteady computations produced six source terms for each grid cell corresponding to the three momentum equation components, the energy equation, and the k and ε turbulence model equations. (Recall that the mass source term is zero.) In other words, the six source terms are functions of Mach number, cavity length-to-depth ratio, translating wall velocity, and the local x and y coordinates. This was true even for the three-dimensional cases because the solutions were identical on each constant z plane, resulting in no z momentum source term or dependence of the other source terms on z. Tests were performed in which the turbulence model source terms were eliminated but prediction accuracy suffered, indicating that they are needed.

The neural network training/validation data consist of each individual source term at all of the (x, y) coordinates for every test case and are stored in a series of vectors, one for each equation. In general, no more than five test cases were needed to obtain reasonable accuracy. These vectors were then randomly reordered to achieve efficient training and then divided into training data (75%) and validation data (25%). Note that this division is done on a pointwise basis, not a casewise basis. That is, approximately three-quarters of each case (M, L/D, and w) were used for training and one-quarter for validation. The latter were used to determine the error when the network generates data it has never seen before. This error was used to set a stopping criterion so that the network would not overfit data by continued training after the validation error reached its minimum. The training process was then stopped and the weights corresponding to the minimum validation error used to generate the source terms and the interface boundary conditions. The networks used in this work had 25 neurons in the hidden layer. This was found to be the optimum number for source term modeling based on the backpropagation algorithm for the first neural network developed in this study. Note that this value was used for all of the neural networks developed in this work, although it may not be optimal for all.

An additional data set was employed after the neural network was trained to test the accuracy of the resultant model. These data were used to compute cavity force values as a way to provide a quantitative measure of model accuracy.

The final source terms are then inserted in a steady-state solver, producing a quasi-time-average solution while avoiding a fully unsteady calculation.

Note that whereas the completed technique allows the user to avoid performing an unsteady calculation, the development of the neural network requires a database of unsteady solutions. The relationship between the number of test cases used and the accuracy of the neural network was not explored exhaustively but should be in follow-on work.

Numerical Technique

Note that the foregoing discussion does not depend on the solver employed other than that a residual must be calculated and a source term included in the governing equations. The technique can be applied to any solver that permits this, although the resultant source terms will be strictly valid only for that solver. With that in mind, the commercial code CFD++, developed by Metacomp Technology, was used to obtain unsteady and LDST modeled time-average solutions. A multidimensional total variation diminishing interpolation was used to avoid spurious numerical oscillations in the computed flowfields.[8] These polynomials are exact fits of multidimensional linear data. An approximate Riemann solver was used to define upwind fluxes and preconditioning used for low-speed flows. The code as employed in this study has second-order spatial accuracy, fourth-order accuracy in time, and a finite volume framework. A wall-distance-free cubic k–ε turbulence model[9] was adopted. This model is tensorially invariant and frame indifferent. It accounts for normal-stress anisotropy, swirl, and streamline curvature effects.

Boundary Conditions

The inflow boundary conditions should maintain the mean aerodynamics of the flowfield and minimize numerical reflections that inhibit convergence for both the unsteady and quasi-time-average calculations. Therefore, a subsonic reservoir boundary was used in which the total temperature, total pressure, and turbulence variables are imposed. The inflow velocity is directly extrapolated from the interior while the inflow total pressure and total temperature are held fixed across the inlet.

Turbulence kinetic energy and its dissipation were taken such that the turbulence intensity was 5%. The value of ε was then obtained by adjusting the length scale until a turbulent to molecular viscosity ratio of 5/20 was achieved.

A no-slip boundary condition was applied for solid walls. The wall was also assumed adiabatic. Pressure was imposed at the outflow boundary, whereas all other quantities were extrapolated from the interior.

A symmetry boundary condition was used at the upper plane, in which all scalar and vector quantities are reflected in the plane defined by the boundary. The flow velocity experiences a free slip at the boundary, becoming tangential to the surface on the boundary itself.

For the quasi-time-average computations, the boundary conditions for the inlet, outlet, upper plane, and walls were the same as for the unsteady simulations. The primitive variables at the cavity interface were extracted from the time-averaged unsteady simulations at the cavity interface because these calculations were performed without a cavity. These variables were then frozen at the interface and used as inflow/outflow conditions. A separate neural network was trained to provide these interface conditions. This ensures that the correct velocities are obtained at the cavity interface even when the cavity is not physically present in the simulation.

Freestream variables were set everywhere in the domain as the initial condition for the first unsteady flow. For subsequent cases, the preceding solution was taken as the initial condition whenever possible, that is, the solution for $M = 0.3$ was taken as the initial condition for $M = 0.4$, and so on. This resulted in much faster cyclic convergence and savings in computation time.

The initial conditions for the quasi-time-average computations were freestream variables everywhere for all cases.

Grids

As shown in Fig. 1, a series of fine two-dimensional meshes were generated for this study with points clustered along the walls, such that the dimensionless distance from the wall to the first wall-adjacent control volume y^+ was approximately 1. A two-dimensional mesh was then extruded in the third dimension for cases that implemented translating wall boundary conditions; however, the minimum number of z-direction cells was used because the resultant flow was identical on all constant z planes.

Several different grids were required for the different L/D cavities studied in this work. All were built from the $L/D = 1$ grid by adding cells of identical size to the center of the cavity. For comparison purposes, there were $100 \times 100 \times 5$ grid cells in the passage and $100 \times 100 \times 5$ grid points in the cavity for the $L/D = 1$ grid. The basic dimensions of this cavity were depth $D = 0.02$ m and length $L = 0.02$ m.

Quantitative Analysis

The accuracy of the unsteady calculations was assessed previously[10] from sound pressure level (SPL) results determined at several points within the cavity and compared to the experiment by Disimile and Orkwis.[11] Excellent agreement was obtained.

The primary analysis tool used in this study was the comparison of forces acting on a control volume sans the cavity, as defined from the control volume relation

$$\boldsymbol{F} = \int_A \boldsymbol{V} \rho \boldsymbol{V} \cdot \boldsymbol{n} \, \mathrm{d}A + \int_A p \boldsymbol{n} \, \mathrm{d}A \tag{8}$$

The integrals are replaced with appropriate summations for the discrete data. The force represents the net exchange of momentum due to unsteadiness and viscous shear. This value is used to compare the time average of the unsteady solution to the NN-LDST augmented steady-state solution to provide a quantitative comparison of accuracy. The specific control volume used in the analysis corresponds to the computational domain. Other reduced extent control volumes were tested and found to have similar results.

The NN-LDST approach is demonstrated in the next section by application to simple subsonic cavity flow.

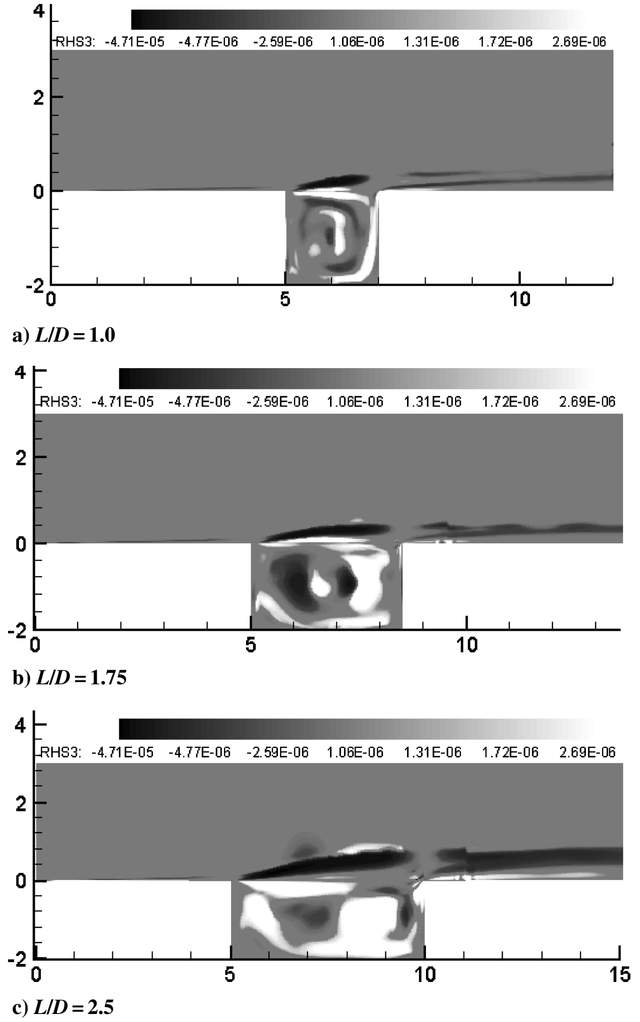
Results

This section presents the results obtained by applying the NN-LDST approach to three sets of unsteady cavity data. It begins with a description of the test cases employed in this study. Examples of the LDSTs are then presented to demonstrate their complexity, discuss their meaning in the context of the time-average solution, and show how they vary with cavity length-to-depth ratio. LDSTs developed from a time-average solution are then applied as source terms to a steady-state solver and the solution variable results compared to verify that LDSTs are the time-average effect of unsteadiness. The feasibility of the NN-LDST approach is then ascertained both qualitatively, by comparisons of solution variable contours, and quantitatively, by comparison of axial and transverse force results.

Three distinct neural networks were developed in this study. The first test case studied was a $L/D = 1$ cavity with Mach number ranging from 0.3 to 1.3. Reynolds number based on momentum thickness varied directly with Mach number and was not studied as a separate parameter; its value was 4.8163×10^4 for $M = 0.3$ and increased to just over 20×10^5 for $M = 1.3$. The second neural network was trained to provide LDST results for cavities with Mach numbers from 0.3 to 0.8 and length-to-depth ratios from 1.0 to 2.75. The third neural network modeled cavities with Mach numbers from 0.3 to 0.8, $L/D = 1$ and aft wall translation velocities from 0 to 287.6 m/s. The goal in all cases was to recreate the time-average solution of the main flow sans cavity. The primary quantitative measure of success was the axial and transverse force prediction.

It was discussed earlier that the mass LDST should be zero, which was found in practice to be true. The primary source terms of significance to this work were the x and y momentum and turbulence quantity source terms. Figures 2 and 3 show the momentum source terms for $M = 0.3$ and L/D values of 1.0, 1.75, and 2.5. Evident in Figs. 2 and 3 are significant source terms in the cavity itself, which are, in general, of no consequence to the modeling of the external flow. Concentrating attention on the external flow, one notes that a significant drag force is evident just above the cavity as expected and illustrated by a large region of negative LDST. The negative values extend significantly downstream and also slightly upstream, evidence of slight upstream influence through the boundary layer. Also evident, particularly in the higher L/D cases is a region of positive LDST toward the aft end of the cavity and above the just-mentioned large negative LDST. This region is associated with acceleration in the boundary layer caused by the displacement of streamlines by the cavity. It is also clear in Figs. 2 and 3 that the effect of increased L/D is to increase the cavity drag on the main flow. Interestingly, the LDSTs internal to the cavity demonstrate that the external flow is adding momentum to the flow in the cavity and thereby speeding it up.

Figure 3 demonstrates similar effects for the y direction, in which the cavity can be expected to significantly increase the boundary-layer thickness by its presence. A strong region of positive LDST is apparent over most of the cavity, although a small region of negative LDST is present just aft of the cavity leading edge, perhaps evidence of shear-layer spreading. Again the LDSTs are apparent

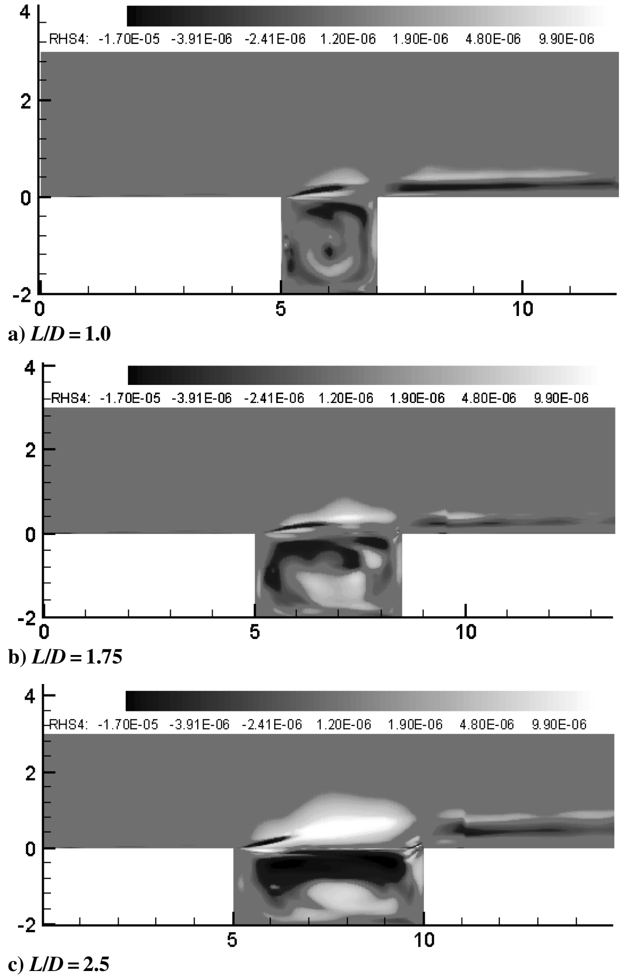


a) *L/D* = 1.0

b) *L/D* = 1.75

c) *L/D* = 2.5

Fig. 2 *X*-momentum source terms for $M = 0.3$ and various length-to-depth ratios.



a) *L/D* = 1.0

b) *L/D* = 1.75

c) *L/D* = 2.5

Fig. 3 *Y*-momentum source terms for $M = 0.3$ and various length-to-depth ratios.

downstream of the cavity but not especially apparent upstream. Figure 3 also shows that the effect of the cavity increases with L/D.

Note that it is the functional form of these LDSTs that are to be modeled by the neural network. The spatial distribution of LDST for any given case is clearly a nonlinear function, but the general form of the LDST is similar in appearance across the cases. It is also clear that its effect is confined to the near-cavity region and downstream, which might be used at a later time to reduce the extent of the modeled LDSTs. The LDSTs are sufficiently nonlinear that a linear interpolation procedure would be unlikely to produce high-quality results unless a huge sample space of the parameters is mapped, increasing the complexity, computation requirements, and memory needed for the approach.

Source terms like those shown in Figs. 2 and 3 are meaningless unless they truly represent the time-average effect of unsteadiness. This idea is confirmed in Fig. 4 where solution variables for the time average of an unsteady solution are compared with a steady-state solution that has been augmented by LDSTs developed from the time-average solution. The latter cases were run with the LDSTs applied only in the region external to the cavity whereas the wall boundary condition along the interface region was replaced with frozen values from the time-average solution. Figure 4 shows Mach number; simulated schlieren, that is, magnitude of density gradient; and z-component vorticity contours. Careful comparison shows that the solutions are identical, verifying the claim that the LDSTs represent the time-averaged effect of unsteadiness. Noteworthy is the match in solution features very near the cavity/main flow interface. All three variables have very rich details in this region that are reproduced identically by the exact LDSTs. This reproduction was evident in all of the cases tested and should be expected based on the earlier mathematical development. The question next is how well the neural network can reproduce the LDSTs, especially for cases that are not used for training.

The first neural network was developed over a Mach number range from 0.3 to 1.3 using unsteady solutions at 0.1 Mach number increments. Figure 5 shows Mach number and vorticity contours for a case not included in the training set, $M = 0.75$. On the left is the time-average obtained from the unsteady solution, and on the right

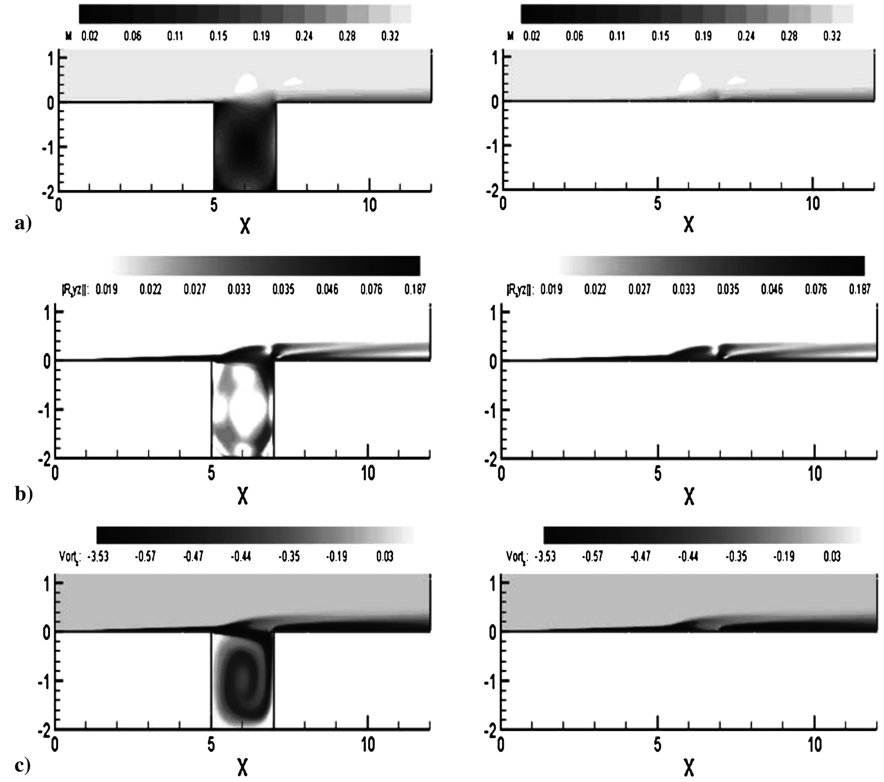


Fig. 4 Time-average solutions obtained from time mean calculation (left) and exact LDST augmented steady-state calculation (right) for $M = 0.3$, and $Re = 48{,}163$: a) Mach number, b) numerical schlieren contours, and c) Z-direction vorticity.

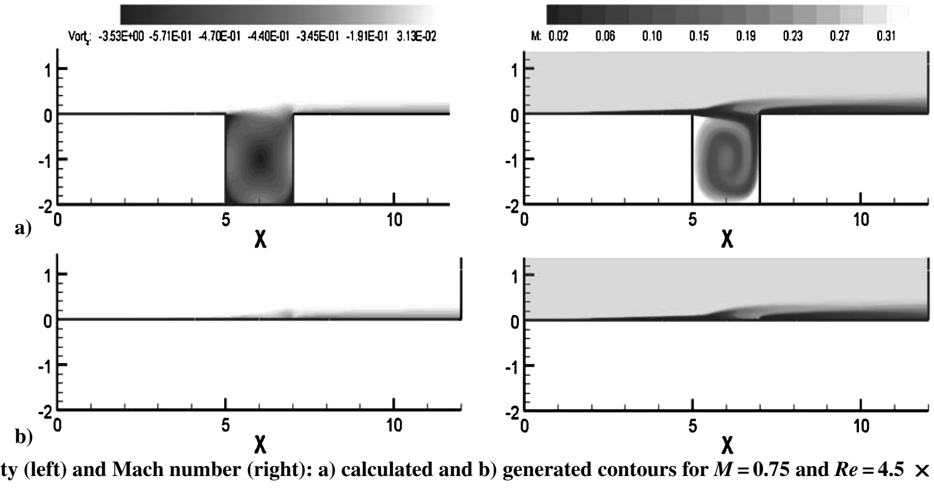


Fig. 5 Vorticity (left) and Mach number (right): a) calculated and b) generated contours for $M = 0.75$ and $Re = 4.5 \times 10^6$, gas path only.

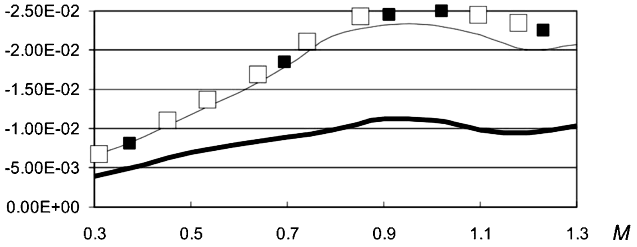


Fig. 6 Axial drag force vs Mach number: ——, time-average of unsteady solution; ▬▬, steady-state without source terms; ■, NN-LDST steady state, and □, NN-LDST steady state, data not included in training set.

is the NN-LDST modified steady-state result. Note that the unsteady solution required approximately 60,000 iterations to achieve cyclic convergence, whereas the NN-LDST modified steady state only required approximately 1000 iterations. The qualitative agreement between the two solutions is remarkable because they are impossible to distinguish with the naked eye. For this reason, the solutions were compared in a more quantitative fashion via the cavity drag force.

Figure 6 shows the cavity drag force vs Mach number as modeled by the first neural network. The thin solid line connects the time-average of the unsteady solutions used to develop the NN-LDSTs. This line should be directly compared to the thick solid line that represents the results of computing without the cavity and applying only no-slip adiabatic wall conditions. Clearly, the two results are significantly different, as expected. Note that if the cavity is neglected in the computation and no source terms are used, an error of between 40 and 55% is produced.

Also shown in Fig. 6 are the cavity drag force data obtained with the NN-LDST modified steady-state solver. The chart distinguishes between data used to train the neural network and that ignored in the training. The error associated with the NN-LDST approach is between 4 and 14%. Interestingly, the error grows considerably as the Mach number increases. It was determined that this is related in part to the added complexity of modeling supersonic LDSTs, which are substantial in the main flow away from the cavity because of oscillating external shock and expansion waves. Subsequent testing showed that reducing the range of modeled data to below the transonic range greatly increased the accuracy of the approach. For this reason, the second neural network was confined to purely subsonic Mach numbers. However, note that the error as compared to ignoring the cavity was reduced in this case by as much as an order of magnitude with the NN-LDST approach.

The second neural network was trained for a Mach number range between 0.3 and 0.7 based on the experience of the first neural network. This time, the cavity length-to-depth ratio was also included as a parameter and ranged from 1.0 to 2.75. Training data at 0.1 Mach and 0.25 L/D increments were employed. Figure 7 shows the variation of cavity drag force with both Mach number and L/D. Once again, lines represent the time-average of the unsteady solution whereas symbols represent NN-LDST augmented steady-state solutions. In this case, $L/D = 0$ corresponds to the case of no cavity, that is, a flat plate. Figure 7 shows that, in general, the axial force increases with Mach number increase, which agrees with the results from the first neural network. The axial force also increases as the L/D increases, as expected. Note that the error associated with ignoring the cavity is considerable and depends greatly on L/D.

Figure 7 also shows four select locations (solid squares) in (L/D, M) space that compare results obtained with the NN-LDST approach for cases not used to train the network. These data were analyzed in a quantitative fashion and were shown to produce errors on the order of 4–14%, with the error greater for larger Mach number and L/D. This should be compared to errors as large as 400% obtained by simply ignoring the cavity. Once again, the results show that the NN-LDST approach reduces error by at least an order of magnitude as compared to ignoring the cavity.

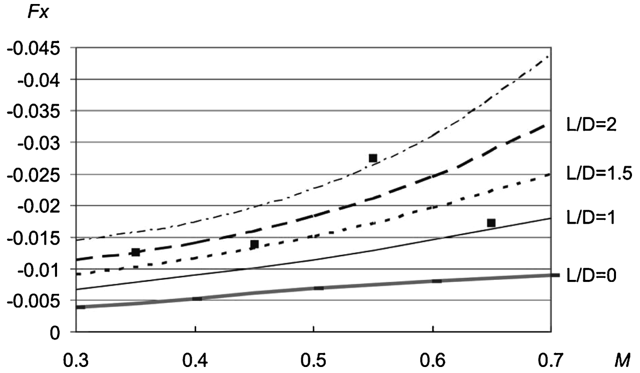


Fig. 7 Axial drag force vs Mach number and cavity length-to-depth ratio: lines, time average of unsteady solution and ■, NN-LDST steady state.

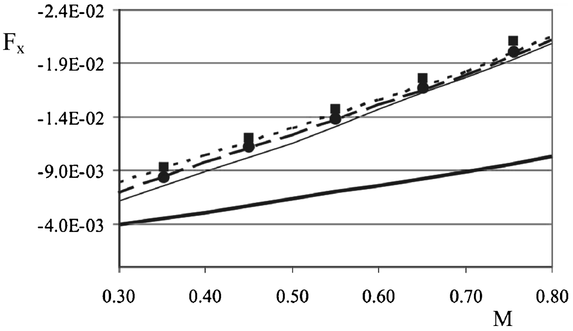


Fig. 8 Axial drag force vs Mach number and cavity aft wall velocity: ▬▬, no cavity; ——, $w_{\text{wall}} = 0$; – – –, $w_{\text{wall}} = 143.8$ m/s; - - - -, $w_{\text{wall}} = 287.6$ m/s; ■, NN-LDST $w_{\text{wall}} = 143.8$ m/s; and ●, NN-LDST $w_{\text{wall}} = 287.6$ m/s.

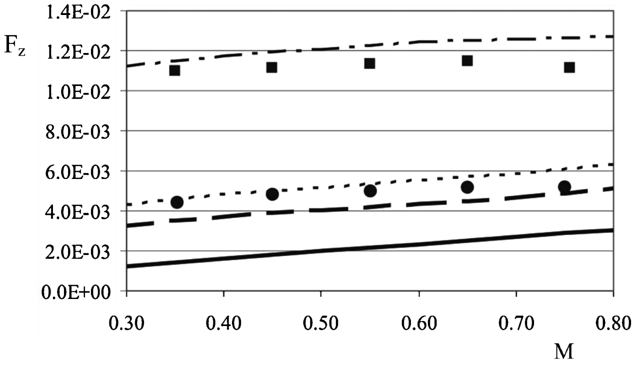


Fig. 9 Transverse drag force vs Mach number and cavity aft wall velocity: ▬▬, no cavity; – – –, $w_{\text{wall}} = 0$; - - - -, $w_{\text{wall}} = 143.8$ m/s; – - –, $w_{\text{wall}} = 287.6$ m/s; ■, NN-LDST $w_{\text{wall}} = 143.8$ m/s; and ●, NN-LDST $w_{\text{wall}} = 287.6$ m/s.

Similar results were obtained with the third neural network, which modeled an $L/D = 1$ cavity in flows with Mach number from 0.3 to 0.8 and aft wall translation varying from 0 to 287.6 m/s. Once again Mach increments of 0.1 were used, this time with three wall velocities: 0, 143.8, and 287.6 m/s. The axial and transverse force results are shown in Figs. 8 and 9, respectively. As with the earlier cases the axial force increases with Mach number, although the transverse force is nearly constant with Mach number. In addition, the transverse force increases with wall translation velocity, as expected. Quantitative analysis of the error shows that the maximum axial force error was approximately 5%, whereas the maximum transverse force error was about 13%. Once again the NN-LDST approach provides a considerably more accurate solution as compared to ignoring the cavity.

In summary, cavity drag force errors of less than 15% were computed with all three neural networks. The results demonstrate that improved results are obtained when the functional space is limited to LDSTs with similar functional forms. In particular, the LDSTs performed better for subsonic cases as compared to supersonic cases because of the increased amount of field source terms in the latter. Clearly, the NN-LDST is not exact but does offer a considerable benefit as compared to ignoring the cavity geometry. This work demonstrated the feasibility of applying an NN-LDST technique for modeling the time-average of an unsteady calculation with a modified steady-state solver. It is expected that this technique will find application in large-scale flowfield simulations in which small-scale cavities are typically ignored. The results demonstrate that small-scale unsteady effects can be included with minimal additional computation effort.

Conclusions

An NN-LDST technique was presented that permits the prediction of a time-average solution with a modified steady-state solver. The technique was applied to three different sets of simple cavity data with Mach number, cavity length-to-depth ratio, and aft wall translation as parameters. The results indicate that axial force data can be reproduced with less than 15% error. The results also indicate that the approach works best for source terms that are locally dominant near the cavity (subsonic) but can be made to work for cases with source terms disbursed throughout the entire field (transonic/supersonic). Computation times for the NN-LDST approach were approximately two orders of magnitude less than the comparable unsteady solution and were essentially identical to that of a steady-state calculation. However, note that a database of unsteady solutions is required to train the neural network; hence, the technique is more suited to cases where multiple simulations will be performed. The LDST source terms did not appear to affect the robustness of the steady-state solver.

Acknowledgments

The authors thank the Dayton Area Graduate Studies Institute and the U.S. Air Force Research Laboratory for their support of this work through Grant PR-AFIT-99-07 and the University of Cincinnati for matching funds. The authors also thank Metacomp Technologies for their help, advice, and consultation.

References

[1] Sondak, D. L., Dorney, D. J., and Davis, R. L., "Modeling Turbomachinery Unsteadiness with Lumped Deterministic Stresses," AIAA Paper 96-2570, July 1996.

[2] Busby, J., Sondak, D., Staubach, B., and Davis, R., "Deterministic Stress Modeling of Hot Gas Segregation in a Turbine," *Journal of Turbomachinery*, Vol. 122, No. 1, 2000, pp. 62–67.

[3] Adamczyk, J. J., Mulac, R. A., and Celestina, M. L., "A Model for Closing the Inviscid Form of the Average-Passage Equation System," *Journal of Turbomachinery*, Vol. 108, No. 1, 1986, pp. 180–186.

[4] Meneveau, C., and Katz, J., "A Deterministic Stress Model for Rotor–Stator Interactions in Simulations of Average-Passage Flow," *Journal of Turbomachinery*, Vol. 124, No. 4, 2002, pp. 550–554.

[5] Ning, W., and He, L., "Some Modeling Issues on Trailing Edge Vortex Shedding," *AIAA Journal*, Vol. 39, No. 5, 2001, pp. 787–793.

[6] Levenberg, K., "A Method for the Solution of Certain Non-Linear Problems in Least Squares," *Quarterly Applied Mathematics*, Vol. 2, No. 1, 1944, pp. 164–168.

[7] Marquardt, D. W., "An Algorithm for Least Squares Estimation of Nonlinear Parameters," *Journal of Society of Industrial Applied Mathematics*, Vol. 11, No. 2, 1963, pp. 431–441.

[8] Chakravarthy, S., "A Unified-Grid Finite Volume Formulation for Computational Fluid Dynamics," *International Journal of Numerical Methods in Fluids*, Vol. 31, No. 1, 1999, pp. 309–323.

[9] Goldberg, U., Peroomian, O., and Chakravarthy, S., "A Wall-Distance-Free k–ε Model with Enhanced Near-Wall Treatment," *Journal of Fluids Engineering*, Vol. 120, No. 3, 1998, pp. 457–462.

[10] Orkwis, P. D., Sekar, B., Chakravarthy, S., and Peroomian, O., "Comparison of Three Navier–Stokes Equation Solvers for Supersonic Open Cavity Simulations," *AIAA Journal*, Vol. 36, No. 5, 1998, pp. 865, 866.

[11] Disimile, P. J., and Orkwis, P. D., "Sound-Pressure-Level Variations in a Supersonic Rectangular Cavity at Yaw," *Journal of Propulsion and Power*, Vol. 14, No. 3, 1998, pp. 392–398.

S. Aggarwal
Associate Editor