JOURNAL OF PROPULSION AND POWER
Vol. 22, No. 6, November–December 2006

Multiscale Modeling of Solid Rocket Motors: Computational Aerodynamics Methods for Quasi-Steady Burning

D. Scott Stewart,* K. C. Tang,† Sunhee Yoo,* Quinn Brewster,‡ and Igor R. Kuznetsov†
University of Illinois, Urbana, Illinois 61801

DOI: 10.2514/1.16111

We consider the problem of efficiently computing the stable burnback of a solid rocket motor when the motor is in the quasi-steady burning regime of operation. When the motor is modeled as a cavity, filled with a compressible fluid with injection from the solid-propellant surface, the problem is a standard one in steady computational aerodynamics. For large rockets, the problem of quasi-steady solid rocket motor core flow is analogous to the steady aerodynamic flow past a large aircraft flying at speeds such that compressible flow effects must be included. One can adopt advanced time-integration strategies that have been applied to commercial or military aircraft to solid rocket motor grain burning to compute a series of steady flows as the grain burns back to near completion. The slow regression of the burning solid-propellant surface is analogous to the motion of control surfaces on the aircraft. With straightforward two-timing, multiscale asymptotic analysis we develop a reduced quasi-steady description of the quasi-steady burning that is extensible to three dimensions. We show how to integrate reduced equations for stable quasi-steady motor burning and use the multigrid method as a representative method to obtain the steady flows in the frozen configurations.

Nomenclature

A	=	cross section area, $0.09 - 2.25$ (m^2)
c	=	sound speed $\sim$1000 (m/s)
h_f	=	enthalpy of the combustion products from the propellant $\sim 4.0 \times 10^6$ (J/kg)
p	=	gas chamber gas pressure $\sim 0 - 8.0 \times 10^6$ (Pa)
$\dot{r}$	=	regression rate of solid propellant, 9.03×10^{-3} m/s at $p = 4.309 \times 10^6$ (Pa)
S	=	perimeter of the solid propellant, $1.88 - 9.5$ (m)
u	=	core gas velocity $\sim 300 - 1000$ (m/s)
u_f	=	injection velocity of the combustion products from propellant $\sim$5 (m/s)
γ	=	ratio of specific heats, 1.138
ρ	=	core gas density $\sim$1.5 (kg/m^3)
ρ_p	=	solid propellant density, 1750.0 (kg/m^3)

I. Introduction

THE industry standard model for solid rocket motor (SRM) combustion uses the compressible Euler equations for the gasses with mass, momentum, and energy injected normal to the solid-propellant grain surface into the core cavity. A representative derivation of this model used by industry has been given by Salita [1,2]. Successful motor operation has a long-lived stable burning regime. After the ignition phase, the motor comes up to a nominally constant operating pressure and vents the combustion products through the nozzle until the final burnout phase. The stable burn is the longest operational phase of the motor where most of the acceleration from the motor thrust is achieved. For the space shuttle booster, the time from ignition to burnout is on the order of about 100 s. During this phase, the propellant surface regresses at approximately 10^{-2} m/s, whereas the maximum particle speed (and sound speed) in the core through the nozzle is on the order of 10^3 m/s. The ratio of these two characteristic velocities, the regression velocity over the characteristic acoustic signaling velocity, is a small parameter about $O(10^{-5})$. Thus, on the time scale that would resolve acoustic events in the motor, the geometry of the core cavity is frozen. Clearly, if the motor is operating in the stable quasi-steady mode, one can compute a sequence of essentially steady flows with side wall injection to simulate the stable quasi-steady burnout event.

The Center for Simulation of Advanced Rockets (CSAR) at the University of Illinois is a Department of Energy program, funded by the Accelerated Strategic Computing Initiative, the focus of which is to develop simulation capability for large solid rocket motors on modern, large-scale parallel computing platforms. At the heart of the project is the central task to simulate a complete burn of a large motor. To accomplish this task (and indeed the same task for any large motor or spatially well-resolved small motor), it is essential to take full advantage of the properties of the stable quasi-steady burning phase to effect a realizable and efficient computation.

In the case of the steady aerodynamic flow past a large commercial aircraft such as a Boeing 757 or a military aircraft such as an F16, flight speeds are such that compressible flow must be fully accounted for. Designers that simulate aircraft face exactly the same challenge, and have almost an identical mathematical problem to solve. For aircraft in steady flight at speeds that require compressible flow modeling, the ratio of the speed of the movement of control surfaces and sound speed in the fluid flowing over the aircraft is comparable to that cited above, i.e., $O(10^{-5})$. One can wholly adopt the highly advanced time-integration strategies developed for application to commercial or military aircraft to the solid rocket motor grain burning to compute a series of realizations of steady flow as the grain burns back to near completion. The slow regression of the burning solid-propellant surface is analogous to the motion of control surfaces on the aircraft as it moves through a controlled maneuver.

Through the use of straightforward, two-timing, multiscale asymptotic analysis, we develop the reduced quasi-steady description of the quasi-steady burning regime for a model problem that is extensible to a full three-dimensional (3-D) rocket. We illustrate the basic strategies for a reduced one-dimensional model, but specifically at the time of this writing, a 3-D axisymmetric version of

Presented as Paper 357 at the 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno Nevada, 10–13 January 2005; received 15 February 2005; revision received 8 February 2006; accepted for publication 8 February 2006.

*Mechanical and Industrial Engineering.
†Center for Simulation of Advanced Rockets.
‡Mechanical and Industrial Engineering.

this strategy is being implemented by us and will be reported on in the future. However, both the two-dimensional (2-D) and 3-D acceleration strategies for computing internal rocket flows require a steady-state Euler solver that implements techniques that are in use for external computational aerodynamics flows, mainly developed for aircraft and are well-documented in references such as Erbele et al. [3] and Jameson [4,5].

We show that if the quasi-steady burn is stable then it can be solved by time-integrating a set of steady core flows that are parametrically dependent on time, combined with the surface regression that evolves on the slow regression time scale. We use a multigrid technique, as proposed by Jameson [4,5] to determine the frozen, quasi-steady flow states required by the slow time-integration steps, as a concrete demonstration of these strategies.

Even at modest spatial resolutions, a motor simulation of the complete burnout that marches at time increments defined by the CFL condition is not practical or, at the current time, realizable. It is a simple matter to convince oneself of this fact by the following estimates. Consider a rocket core with length L, average cross-sectional area A, final simulation time t_f, average spatial resolution $\triangle x$, number of spatial nodes in the main stream direction $N_x = L/\triangle x$, and total of time steps for the simulation $N_t = t_f/\triangle t$. The relation between the time step and spatial increment defined by the CFL condition is given by $\triangle t = \alpha \triangle x/|u + c|$, where $|u + c|$ is the characteristic signaling velocity and $0 < \alpha < 1$ is a constant that defines the CFL condition. The total operational count required to simulate the total burnout can be written as $N_{\text{ops}} = \beta N_t \times N_x^d$ where β is a geometry, machine, and algorithm dependent constant and d is the dimension of the model ($d = 1$, 2, or 3). Combining these formulas obtains the formula $N_{\text{ops}} = (\beta/\alpha)(t_f|u + c|L^d)/[(\triangle x)^{d+1}]$. As an example for the space shuttle solid rocket booster, for a 30 m rocket with 3 cm resolution in the stream direction (with $N_x = 1000$), with $|u + c| \sim 1000$ m/s, with $t_f = 100$ s, for a three-dimensional simulation with $d + 1 = 4$, leads to the estimate $N_{\text{ops}} \sim (\beta/3\alpha) \times 10^{16}$. Even with advanced algorithms that could lead to small values of β, simple CFL-based time-marching simulation is likely to be incomputable, and the design parameter space unsearchable.

Thus, we draw the simple conclusion that for the regimes of stable operation of a motor at pressures on the order of hundreds of atmospheres, marching at the CFL-prescribed time step is not viable for practical computation; another strategy is required. Implementing a different time-marching strategy on the regression time scale can provide a significant reduction of the total number of time steps required for a simulation. The computational strategy is that associated with progressing through a series of quasi-steady states as defined by the intrinsic multiscale asymptotic formulation for quasi-steady burning.

The following sections describe our simplified one-dimensional model, which can be used to study longitudinal dynamics of the rockets and has all the elements in common with the three-dimensional Euler model. Then, we derive reduced equations for quasi-steady burning for the model based on rational asymptotics. (The formulation of the reduced equation are essentially the same for a three-dimensional rocket.) For the one-dimensional model, the steady equations of the frozen flow can be solved exactly by a simple algebraic procedure, which we explain. Then, those steady states can be used to define the surface regression velocity that can in turn be used to advance the surface to burn out the solid-propellant grain, and provide a baseline for comparison with the direct solver. We briefly describe the direct solver, composed of the combined Euler solver coupled to the surface evolution. Then, we review some of the time-marching strategies that are likely to be the most efficient for large-scale rockets that employ parallel architecture drawn from established methods of computational aerodynamics which can be directly implemented to compute the long-lived quasi-steady burn. We present a clear demonstration that acceleration of the time-integration process can be obtained by solving the reduced equations, with the multigrid method chosen as the representative method to compute the quasi-steady flow states.

II. Multi-Scale Modeling of Solid Rocket Motors

A. Simplified Model of an SRM

The following model posed by Salita [1] allows for time evolution of the shape of the solid-propellant grain (i.e., the domain geometry). Let the cross section area of the ballistic, gas-filled core of the rocket be given by $A(x, t)$, where x is the distance along the axis of the rocket and t is time. Then, the governing equations for a rocket with a slowly varying cross section are

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial x} = \boldsymbol{s}(\boldsymbol{u}) \tag{1}$$

with

$$\boldsymbol{u} = \begin{bmatrix} \rho A \\ \rho u A \\ \rho e_T A \end{bmatrix}, \qquad \boldsymbol{f}(\boldsymbol{u}) = \begin{bmatrix} \rho u A \\ (\rho u^2 + p)A \\ u(\rho e_T + p)A \end{bmatrix}$$
$$\boldsymbol{s}(\boldsymbol{u}) = \begin{bmatrix} \rho_p \dot{r} S \\ p \frac{\partial A}{\partial x} \\ \rho_p \dot{r} S \left(h_f + \frac{1}{2} u_f^2 \right) \end{bmatrix} \tag{2}$$

where $e_T = p/[\rho(\gamma - 1)] + u^2/2$ is the total energy, $S = 2\pi r$ is the perimeter, r is the effective cross section radius, with $A = \pi r^2$. The propellant density is ρ_p, the enthalpy of the combustion products injected from the propellant surface is h_f, and the injection velocity of the combustion products is $u_f = (\rho_p/\rho)\dot{r}$. Note that $r(x, t)$ is a variable that is determined through the addition of the surface regression evolution law:

$$\dot{r} \equiv \frac{\partial r(x, t)}{\partial t} = kF(p, \rho, u) \tag{3}$$

that is prescribed by specifying the combustion processes at the grain surface, where k is a characteristic regression velocity that defines the rate. The preceding equations hold in the region of the grain. For the examples discussed in this paper, we use the pressure dependent regression law $\dot{r} = ap^b$. The model configuration is completed by the addition of a nozzle. In such a region, there is no side wall injection and the source terms in $\boldsymbol{s}$ are modified by setting $\dot{r} = 0$ so that

$$\boldsymbol{s}(\boldsymbol{u}) = \begin{bmatrix} 0 \\ p \frac{\partial A}{\partial x} \\ 0 \end{bmatrix} \tag{4}$$

B. Scaling

Here, we consider scaling, such that the problem is posed on the scale of the longitudinal acoustics of the motor. Let a c subscript be a dimensional, characteristic scale. Choose the length scale to be L (the nominal length of the motor), let p_c be characteristic chamber pressure of steady-state operation of the motor (i.e., on the order of 100 atmospheres), let ρ_c be a characteristic chamber density, choose the characteristic velocity be defined by the relation $V_c^2 = p_c/\rho_c$, so that the functional form of the governing equations is unchanged under scaling, and let the time scale be given by $t_c = L/V_c$.

Under the change of variables to dimensionless form, the scaled version of the equations are the same as the preceding equations, except that $\boldsymbol{s}(\boldsymbol{u})$ is replaced by

$$\boldsymbol{s}(\boldsymbol{u}) = \begin{bmatrix} \epsilon F S \\ p \frac{\partial A}{\partial x} \\ \epsilon F S \left[Q + \frac{1}{2} \epsilon^2 f^2/\rho^2 \right] \end{bmatrix} \tag{5}$$

and the regression rate law becomes simply

$$\frac{\partial r}{\partial t} = \delta F \tag{6}$$

where we identify the dimensionless parameter ϵ and δ as the dimensional ratios

$$\epsilon = \frac{\rho_p}{\rho_c} \times \frac{k}{V_c}, \qquad \delta = \frac{k}{V_c} \tag{7}$$

Based on characteristics of typical solid-propellant rocket motors, $\rho_p/\rho_c \sim 1750/1.5 \sim O(1000)$, V_c is about the sound speed which is about 1000 m/s, whereas the regression velocity is about 10^{-2} m/s, so that $k/V_c \sim O(10^{-5})$, with $\epsilon \sim 10^{-2}$ and $\delta \sim 10^{-5}$. Also, we should note that $Q = h_f/V_c^2$ is $O(1)$.

Measured on the acoustic time scale, the burning surface evolves with $O(1)$ changes to its configuration on an $O(1/\delta) \sim 10^5$ time scale, a very long time indeed. The time required to observe changes in the shape of the grain is so long that resolved time evolution on the acoustic time scale is essentially, in practical terms, incomputable for a large rocket. The only way to compute the configuration changes in a straightforward way is to compute a series of essentially (quasi-) steady states, and then make configuration changes on the regression rate time scale.

What follows next is a simple but rigorous derivation of the reduced equations for quasi-steady burning in SRMs, explained in the context of this simplified model.

III. Quasi-Steady Burn Approximation

Here, we consider the following limit. Let ϵ be $O(1)$ on the scale of δ and simply consider the limit as $\delta \to 0$. Further, let $t = t_0$ (we will drop the zero subscript subsequently) be the "fast" acoustic time scale, and let $\tau = \delta t$ be the "slow" regression (shape change) time scale. For a multiple-time scale expansion of the form

$$\boldsymbol{u} = \boldsymbol{u}^{(0)}(x, t_0, \tau) + \delta \boldsymbol{u}^{(1)}(x, t_0, \tau) + \cdots \tag{8}$$

one calculates

$$\frac{\partial \boldsymbol{u}}{\partial t} = \frac{\partial \boldsymbol{u}^{(0)}}{\partial t_0} + \delta \frac{\partial \boldsymbol{u}^{(0)}}{\partial \tau} + \delta \frac{\partial \boldsymbol{u}^{(1)}}{\partial t_0} + \cdots \tag{9}$$

$$\begin{aligned} \frac{\partial \boldsymbol{f}}{\partial x} &= \frac{\partial \boldsymbol{f}[\boldsymbol{u}^{(0)}(x, t_0, \tau) + \delta \boldsymbol{u}^{(1)}(x, t_0, \tau) + \cdots]}{\partial x} \\ &= \frac{\partial \boldsymbol{f}(\boldsymbol{u}^{(0)})}{\partial x} + \delta \frac{\partial}{\partial x}\left[\frac{\partial \boldsymbol{f}(\boldsymbol{u}^{(0)})}{\partial \boldsymbol{u}} \boldsymbol{u}^{(1)}\right] + \cdots \end{aligned} \tag{10}$$

and

$$\boldsymbol{s}(\boldsymbol{u}) = \boldsymbol{s}(\boldsymbol{u}^{(0)}) + \cdots \tag{11}$$

The regression rate law expansion is expressed as

$$\frac{\partial r}{\partial t}|_t = \frac{\partial r^{(0)}}{\partial t_0} + \delta\left[\frac{\partial r^{(1)}}{\partial t_0} + \frac{\partial r^{(0)}}{\partial \tau}\right] + \cdots = \delta[F(p^{(0)}, \rho^{(0)}, u^{(0)}) + \cdots] \tag{12}$$

The leading order $O(1)$ formulation in the propellant grain domain is then expressed as

$$\frac{\partial \boldsymbol{u}^{(0)}}{\partial t_0} + \frac{\partial \boldsymbol{f}(\boldsymbol{u}^{(0)})}{\partial x} = \boldsymbol{s}(\boldsymbol{u}^{(0)}) \tag{13}$$

$$\boldsymbol{s}(\boldsymbol{u}^{(0)}) = \begin{bmatrix} \epsilon F^{(0)} S^{(0)} \\ p^{(0)} \frac{\partial A^{(0)}}{\partial x} \\ \epsilon f^{(0)} S^{(0)} \left[Q + \frac{1}{2}\epsilon^2 \frac{(F^{(0)})^2}{(\rho^{(0)})^2}\right] \end{bmatrix} \tag{14}$$

with the leading order regression, effectively motionless, as

$$\frac{\partial r^{(0)}}{\partial t_0} = 0 \tag{15}$$

In the nozzle region, Eq. (13) holds with $\boldsymbol{s}(\boldsymbol{u})$ identified by Eq. (4). This leading order system has fast-time dynamics, with injection from the side walls for a fixed grain geometry.

A. Summary of the Reduced Equations Appropriate for Quasi-Steady Motor Burning

If we assume that the flow is steady on the fast-time scale, but allow for changes on the slow-time scale, then steady flow equations

$$\frac{\partial \boldsymbol{f}(\boldsymbol{u}^{(0)})}{\partial x} = \boldsymbol{s}(\boldsymbol{u}^{(0)}) \tag{16}$$

hold in the core. The regression rate law expansion at $O(\delta)$ given by Eq. (12) is expressed uniformly as

$$\frac{\partial r}{\partial t} = \delta F(\rho^{(0)}, p^{(0)}, u^{(0)}) \tag{17}$$

These reduced equations govern leading order, quasi-steady rocket motor burn. The balance Eqs. (16) depend parametrically on the slow time δt. These equations are not the fully unsteady equations. A separate time-integration strategy must be devised to achieve their solution. If the quasi-steady flow is unstable on the fast, acoustic time scale, then the solutions achieved by the integration of this system are not the physically appropriate ones. But if the quasi-steady flow is stable on the acoustic time scale, then a time-integration method which is appropriate for the computation of steady compressible, transonic flows can be used. The methods developed principally for aircraft are appropriate ones to consider. These will be discussed next.

We make a passing comment about stability of these quasi-steady flows. Since the quasi-steady burn evolves on the slow time scale, it is steady on the fast, acoustic time scale. On the fast-time scale, the geometry of the rocket, here represented by $A(x, t)$, is frozen in time. And so, there is a possibility that the rocket can slowly burn into a geometry where the quasi-steady solution is no longer stable and a transition to instability occurs. The detailed analysis of acoustic instabilities is not within the scope of our present discussion and will be addressed in a sequel to this work.

B. Quasi-Steady Burning

Quasi-steady motor burning is the special case when the leading order solution is steady on the fast, acoustic time scale. The special steady burning solution that has $t_0 = t$ and $\partial/\partial t_0 = 0$, in which case, one must solve the ordinary differential Eqs. (16). To these equations we add the boundary condition at the head end $u = 0$ at $x = 0$. Specification of the head end pressure provides enough information to integrate Eqs. (16) forward to determine profiles in the grain region. These values are determined by insisting that at the entrance to the inert nozzle the state variables, i.e., ρ, u, and p, and hence the mass flux are continuous, and that the nozzle flow is an isentropic expansion with a sonic point at the throat. These calculations follow.

For steady flow in a rocket with a slowly varying propellant and nozzle shape, the calculation of the steady flow with steady side wall injection reduces to the solution of a nonlinear eigenvalue problem. The steady equations in the grain and nozzle section are (dropping the zero superscript)

$$\frac{d\boldsymbol{f}(\boldsymbol{u})}{dx} = \boldsymbol{s}(\boldsymbol{u}), \qquad \boldsymbol{f}(\boldsymbol{u}) = \begin{bmatrix} \rho u A \\ (\rho u^2 + p)A \\ u(\rho e_T + p)A \end{bmatrix} \equiv \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \tag{18}$$

where $\boldsymbol{s}(\boldsymbol{u})$ is given in Eq. (2) or Eq. (4) in the grain or nozzle segment, respectively.

To integrate these equations from the head end of the rocket $x = 0$, we prescribe the values of f_i at $x = 0$. The particle velocity u is zero at the head end, which sets $f_1(0) = f_3(0) = 0$ and $f_2(0) = p(0)$ $A(0)$. The value of area function $A(x)$ is assumed to be known. To start the integration, one makes a guess for the pressure at the head end $p(0)$ and integrates to the inlet of the nozzle $x = x_{\text{inlet}}$. The primitive states can be calculated from the algebraic formulas

$$p = \frac{f_2 + \sqrt{f_2^2\gamma^2 + 2f_1f_3(1-\gamma^2)}}{A(1+\gamma)}, \qquad u = \frac{f_2}{f_1} - \frac{Ap}{f_1} \tag{19}$$
$$\rho = \frac{f_1}{Au}$$

The Mach number at the inlet is given by

$$M_{\text{inlet}} = u_{\text{inlet}}\sqrt{\frac{\rho_{\text{inlet}}}{\gamma p_{\text{inlet}}}}$$

For the same point, we calculate the inlet Mach number from the area Mach number relation [6] for the nozzle, here designated by a plain M as

$$\frac{A_{\text{inlet}}}{A_{\text{throat}}} = \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}} \frac{1}{M}\left(1 + \frac{\gamma-1}{2}M^2\right)^{\frac{\gamma+1}{2(\gamma-1)}} \tag{20}$$

where the nozzle contraction ratio $A_{\text{throat}}/A_{\text{inlet}}$ is prescribed. Iterations on $p(0)$ are repeated until the residual RES $\equiv (M_{\text{inlet}} - M) <$ ERRABS, or until the difference between two successive guesses for the absolute error pressure (ERRABS) is less than the relative error (ERRREL).

Once the iterations converge, the steady solution is determined throughout the domain from the head end to the nozzle inlet. To obtain the solution in the nozzle part of the domain, we employ algebraic nozzle relations [6] as follows. Using the inlet values as a starting point, at the neighboring grid point inside the nozzle, area ratio is calculated. Based on that ratio, and inlet values, the values of primitive variables at the location of the current grid point are obtained.

For our representative baseline simulation we choose a booster that is 14 m long and 3 m in diameter. (The dimensions, thermochemical, and burn rate parameters are not unlike those of the NASA space shuttle SRM, which is the main object of CSAR's studies. Similar parameters are listed in Salita's original study [1,2]). Reduction of the motor length allows us to reduce the cost of the simulation. Because the goal of this test is to estimate the speedup, and not to build a precise model of SRM, such a modification of geometry was deemed reasonable. The baseline parameters for our test simulation are shown in Table 1.

Another restriction that we applied to our test geometry was smoothness of the area function $A(x)$; specifically, we required that the function $A(x)$, along with its derivative $A'(x)$ was continuous. In the one-dimensional formulation, the area function is introduced to modify the governing equations to account for the variation of two-dimensional geometry of the rocket motor. In some circumstances, a quasi-one-dimensional mathematical model can be quite accurate, even predictive for basic design, and satisfy all the modeling assumptions. But discretization of $A(x)$ can introduce large numerically generated errors into the source terms near points where $A(x)$ is not smooth or has sharp transitions. To maintain the desired accuracy of the direct solver, it was found sufficient to require that

Table 1 Parameters for the baseline simulation

Symbol	Value	Dimension	Name
C_p	2.478×10^3	J/kg K	specific heat
γ	1.138	——	ratio of specific heat
ρ_p	1.756×10^3	kg/m^3	propellant density
T_f	3.422×10^3	K	injecting gas temperature
u_f	$(\rho_p/\rho)\dot{r}$	m/s	injecting gas velocity
h_f	C_pT_f	J/kg	injecting gas enthalpy
$\dot{r}$	ap^b	m/s	surface regression rate
a	2.44×10^{-3}	m/(s atm^b)	power law premultiplier
b	0.35	——	power law exponent
S	$2\pi r$	m^2	cross section perimeter
A	πr^2	m^3	cross section area
e_T	$p/[\rho(\gamma-1) + u^2/2]$	J/kg	total energy
p_c	10^6	Pa	pressure scale
u_c	10^3	m/s	velocity scale
t_c	10^{-3}	s	time scale

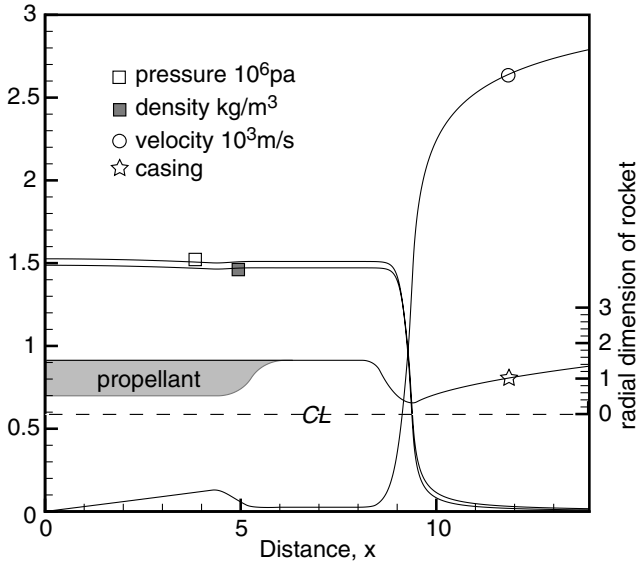

Fig. 1 Primitive variable profiles in the rocket booster obtained by ODE solver STEADY.

$A(x)$ be C^1 continuous, which eliminated the spurious numerical difficulties that came from the idealizations of the one-dimensional model.

Figure 1 shows an example of solid rocket motor geometry and steady-state primitive variable profiles computed by the ODE solver "STEADY". The same steady states were computed by allowing a direct numerical simulation to relax to a steady state defined by steady injection, and the solution profiles thus obtained are indistinguishably close to the profiles shown in Fig. 1. [A brief description of the direct numerical simulation (DNS) code dubbed "WT/AXS" follows later.] Both the steady and the DNS simulations were computed with 800 nodes ($dx = 14/800 = 0.0175$ m). For this paper, the steady-state solver that solves the steady ODEs and carries out the nonlinear iteration, is dubbed STEADY.

The root finding iterations for steady-state solver STEADY were performed with IMSL routine DZREAL, EPS $= 1 \times 10^{-15}$, ERRABS $= 1 \times 10^{-15}$, ERRREL $= 1 \times 10^{-15}$, and ETA$= 1 \times 10^{-12}$. The root of Eq. (20) is found with DZBREN, ERRABS $= 1 \times 10^{-15}$, and ERRREL $= 1 \times 10^{-15}$. Subroutine DIVPRK was implemented to solve the ODEs of Eq. (18) from the head end to the nozzle inlet, with tolerance parameter set to TOLR $= 5 \times 10^{-5}$. Accuracy of the second order solver was verified by substituting the solution back into the governing equations and evaluating the residuals and demonstrating that the error was reduced by the appropriate power law. The order of self-convergence was found to be slightly less than two.

C. Direct Simulation and Comparison with Reduced Equations Using the "Exact" Steady Solver

1. *Representation of the Solid-Propellant Grain Shape*

The direct numerical simulation of solid-propellant grain burnout requires the solution of a moving boundary problem, which for the present discussion is represented by Eq. (1) for evolution of the flow states and Eq. (3) for the evolution of the solid grain shape. To carry out the solution, one must choose an algorithm to represent the grain surface at any instant and to compute its motion. We note that even for this model, although the hydrodynamics are simplified and quasi-one-dimensional, the grain shape is fully two-dimensional, and so a two-dimensional representation of the grain shape is required. For all the calculations discussed in this paper, we used a recently developed hybrid level set method developed by Yoo and Stewart [7] to represent and evolve the grain shape. We dub the code that uses this method for the grain shape "WT" for wave tracker. The WT code has two reciprocal procedures. The first stores a parametric representation of the surface using nodes (or particles) that correspond to the intersection of the shape with an underlying regular

mesh. The second procedure constructs a level set field on a band of grid points that embeds the surface curve and uses an orthogonality condition to compute the exact normal distance function to the stored curve. Time-integration of the level set partial differential equation is used to evolve the interface according to an assignment of the normal velocity of the surface, determined by the prescription of the grain regression rate. The level set partial differential equation was solved with a fifth order weighted essentially non-oscillatory (WENO) upwinding spatial discretization and a third order Runga–Kutta scheme method of lines.

D. Direct Numerical Simulation Code WT/AXS

The direct numerical simulation of the model was carried out with a combined solver that computes the flow states and the grain motion. The flow states are computed with a now standard, high resolution fifth order WENO scheme that uses the method of lines and a third order Runga–Kutta scheme [8] with the CFL number set to 0.9. We dub this flow solver AXS. The direct numerical solution of our model was accomplished by using a hybrid level set algorithm WT to evolve the grain geometry combined with the flow solver AXS. The direct simulation code, which combines WT and AXS, is dubbed WT/AXS. Tests to determine the steady-state profiles were carried out with initial profiles for the state variables computed with STEADY that were perturbed by 5%. Computations were carried out until a steady flow was achieved for a frozen (fixed) geometry. When plotted, the steady solution obtained with WT/AXS is indistinguishable from the steady solution shown in Fig. 1.

E. Simulation of the Reduced Equations Using the Exact Steady Solver

Numerical simulation of the reduced equations were carried out using the exact steady solver STEADY, in combination with WT, to evolve the grain surface. Since the flow states are solved very quickly with few iterations, almost all of the computation comes from the surface evolution. The use of an exact steady solver in some sense defines the ideal case and provides an estimate of the absolute minimum computational requirement needed to compute the complete solid grain burnout. We note that steady exact solutions are probably only available for this one-dimensional model, albeit asymptotic solutions maybe available for quasi-steady axisymmetric, time-averaged steady, two-dimensional flows. The combination of the STEADY flow state solver with the interface tracker WT is dubbed WT/STEADY.

F. Comparisons of WT/AXS and WT/STEADY: Estimates of Ideal Speedup for the Model

The WT/STEADY solver was used to perform complete motor burnout tests to estimate the maximum speedup of the total simulation. For each grain configuration, STEADY is used to compute a steady-state solution. The pressure distribution is then used to evaluate the burn rate using $\dot{r} = a(\hat{p})^b$, where $\hat{p}$ is pressure measured in atmospheres, $a = 2.44 \times 10^{-3}$ m/(s atm^b) with $b = 0.35$. Based on the calculated burn rate, WT advances the propellant surface to a new configuration at the next time step. Table 2 presents burnout test results for WT/STEADY. Here N_{pts} is a number of grid points, N_{iter} is a number of time steps (geometry updates) taken by WT to reach final burnout time of 140 s, $\triangle t$ is the time step WT-CFL condition which is based on the surface regression velocity, and T_{sim} is total simulation run time (in seconds) on a single AMD 2000+ 1.67 GHz processor.

For $N_{\text{pts}} = 200$ points, the Wave Tracker time step $\triangle t = 0.0275$ s requires approximately 5×10^3 time steps for a complete burnout,

Table 2 Complete (140 s) grain burnout test results for WT/STEADY

N_{pts}	N_{iter}	$\triangle t$	T_{sim}(CPU − s)
200	5095	0.0275	522
400	10,444	0.0138	2432
800	20,241	0.00069	11,011

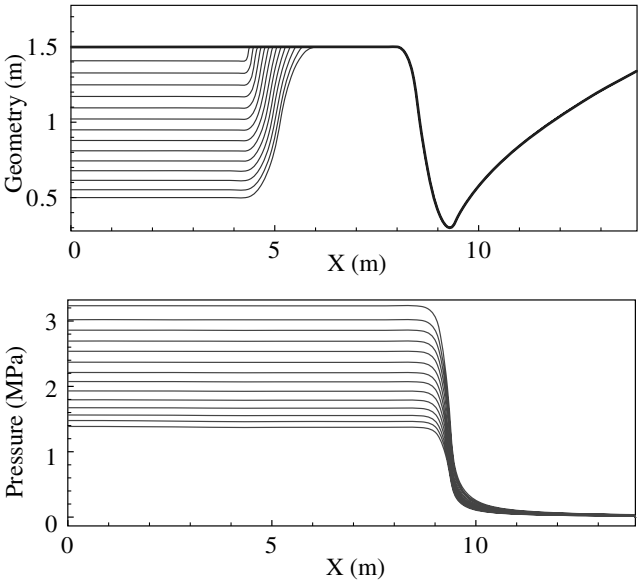

Fig. 2 Burnout test of 140 s with grain shape shown at 10 s intervals.

whereas the CFL restriction for a direct numerical simulation as carried out with WT/AXS at the same spatial resolution ($N_{\text{pts}} = 200$) would be $\triangle t_e = 7 \times 10^{-6}$ s, requiring 2×10^7 iterations. This provides us with an excellent estimate of the upper bound for a speedup of a complete burnout simulation performed by solving the reduced equations, marching through a series of quasi-steady-state flow states that are used to advance the surface regression, as compared with the direct numerical simulation of the Euler equations for the flow states coupled with the surface regression. That simple estimate, expressed as a ratio of the time step required for solving the reduced equations over the time step required for the DNS, is $27.5 \times 10^3/7 \times 10^{-6}$ and for $N_{\text{pts}} = 200$ is $\approx 4 \times 10^3$.

Figure 2 shows the time evolution of grain geometry and pressure, as computed with WT/STEADY, with 800 spatial points ($dx = 0.0175$ m). The parameters used to simulate the complete grain burnout are listed in Table 1. The receding grain shapes are shown at 14 equal time intervals of 10 s. As the propellant grain burns out, the internal radius of the cavity increases, leading to an increase in burning area. At the same time, the length of the propellant grain decreases, leading to a decrease of the burning area. The choice of the initial geometry of the grain defines the balance between the two processes. In our test case, gradual burning surface area increase leads to an increase in chamber pressure.

IV. Discussion of Basic Strategies from Computational Aerodynamics: Example Using the Multigrid Method

Here, we present a brief summary of some conclusions and strategies from computational aerodynamics which can be applied to integration of the reduced quasi-steady equations for large-scale motors. Recently, at a U.S. Air Force Office of Scientific Research workshop on "Advances and Challenges in Time-Integration of PDEs" held at Brown University in August 2003, A. Jameson, Stanford University, presented a summary of modern methods and strategies from computational aerodynamics developed specifically for steady compressible Euler flows to be carried out on parallel supercomputers. Here, we recount some of his conclusions [5] which we believe identify some of the best strategies that can be applied to large-scale motor simulation. One is referred to a recent monograph [3] and Jameson's original article on a multigrid method [4].

If one needs only to compute a succession of quasi-steady flows (in our case as levels in the temporal discretization of the slow regression), a fast steady-state solver is always needed, even for a fully implicit scheme. The simplest method is to compute the transient solutions to the unsteady Euler equations until a new steady state is achieved, but this is not efficient as there is a long relaxation time required for pressure equilibration, often requiring at least 1000

pressure wave traverses through the entire (large) structure. Alternatively, one can identify a preconditioner that multiplies the steady-state homogenous operator and adds to the homogenous spatial operator a pseudotime derivative of the dependent variables. Then the steady state is achieved by relaxation to the steady state in pseudotime. One can consider fully implicit schemes. These schemes have the property that a fully nonlinear set of algebraic equations result, and even the linear solve of these equations requires the inversion of a full matrix. The implicit schemes are based on backward time differencing that is linearized by an approximate factorization which leads to an alternating direction implicit scheme and are not easily made parallel, as the entire grid must be updated sequentially on a sweep.

To complete our analysis, we choose to implement a standard steady solver to demonstrate that in a more complex (full two- and three-dimensional) model simulation time acceleration could be achieved. We choose the multigrid (MG) method, with a modified Runge–Kutta time-integration which is one of the preferred and widely used schemes for large aircraft, is easy to implement, has good accelerated convergence properties, and can be easily made parallel, as an example of a standard method. The idea is to take large time steps on a sequence of coarse grids and interpolate between the finer and coarser meshes with a sequence of time steps which do not violate the stability restriction. Next, we show that a multigrid scheme can produce accelerated time-integration.

A. Multigrid with Runge–Kutta Integration: Solving the Reduced Equations with WT/MG

In a quasi-one-dimensional formulation, the fastest and the most accurate way to solve for the steady-state solution is to solve the steady equations directly, which we have shown can be done by integrating the steady ODEs followed by an iteration on the head end pressure. In two and three dimensions, there is no such simple and straightforward way to solve the system of steady governing equations, hence we need to use another method, such as multigrid [4,9] to obtain the steady-state solution. For our one-dimensional model, the multigrid method is used to produce the frozen steady flow states, and thus replaces the solver STEADY which is used in combination with the interface solver WT to advance the grain interface. We dub the solver that uses the combination of multigrid for the steady flow states and WT for the interface WT/MG.

1. *Brief Description of the Multigrid Method*

The multigrid method effectively accelerates convergence to the steady state by solving unsteady governing equations on successively coarser meshes which allow larger time steps. We used the V-cycle with fourth order modified Runge–Kutta time-integration [4,5] on each grid level. For our test geometry, the minimum number of grid points that allow us to capture all significant geometry details, such as nozzle throat width, was considered to be 200. For these tests, we assumed that the maximum desired resolution is around 1 cm, thus we considered three levels of refinement with a four-grid level system of 200, 400, 800, and 1600 grid points.

After we implemented the basic algorithm, we observed that numerical error generated at locations where function $A(x)$ was not C^2 smooth was magnified in the multigrid simulation, which resulted in slow, or no, convergence to the steady-state solution. To remedy this difficulty, we reformulated the governing equations so that all area-related terms were only present as source terms and did not appear in the flux calculation. In this way, we excluded possible sources of singularity due to geometry from the flux calculation.

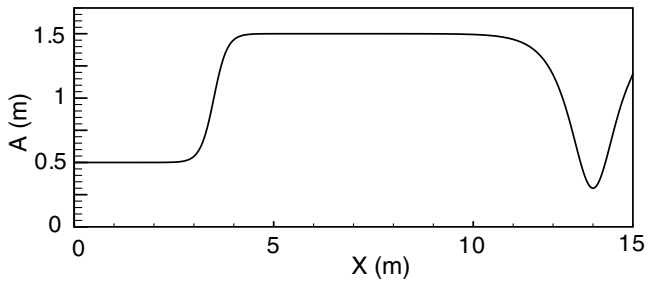

Fig. 3 Area function $A(x)$ used for multigrid calculations.

$$\boldsymbol{u} = \begin{bmatrix} \rho \\ \rho u \\ \rho e_T \end{bmatrix}, \qquad \boldsymbol{f}(\boldsymbol{u}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(\rho e_T + p) \end{bmatrix}$$
$$\boldsymbol{s}(\boldsymbol{u}) = \frac{1}{A}\begin{bmatrix} \rho_p \dot{r} S - \rho u \frac{dA}{dx} \\ -\rho u^2 \frac{dA}{dx} \\ \rho_p \dot{r} S\left(h_f + \frac{1}{2}u_f^2\right) - \frac{dA}{dx}[u(\rho e_T + p)] \end{bmatrix} \tag{21}$$

Secondly, we changed geometry such that resulting function $A(x)$ is C^2 smooth by using hyperbolic functions to represent the shape of the propellant grain and nozzle (Fig. 3). The grain area function $A(x)$ is defined with a $\tanh(x)$ function, and the nozzle part is defined using a $\mathrm{sech}(x)$ function. We emphasize that these remedies are only needed in one-dimensional formulation, where function $A(x)$ is present and source terms are modified to account for chamber area change. In higher dimensions, area change is introduced through the boundary conditions of the problem and does not enter into the governing equations. Therefore, in two and three dimensions, the current multigrid method can be applied to arbitrary solid-propellant grain and nozzle shapes without smoothness restrictions.

An "ideal" multigrid algorithm figure of merit can be estimated as follows. Following Jameson [5] for the V-cycle, the computational cost of one multigrid time step (cycle) is evaluated to be

$$1 + \frac{1}{2} + \frac{1}{4} + \cdots < 2 \text{ in one dimension,}$$
$$1 + \frac{1}{4} + \frac{1}{16} + \cdots < \frac{4}{3} \text{ in two dimensions,}$$
$$1 + \frac{1}{8} + \frac{1}{64} + \cdots < \frac{8}{7} \text{ in three dimensions}$$

The effective time step of one multigrid cycle is

$$\triangle T = (1 + 2 + 4 + 8 + \cdots)\triangle t$$

where $\triangle t$ is the time step determined by the CFL condition for the finest grid. We can estimate the figure of merit, or speedup for calculating the steady state, as compared with that of relaxation to steady state achieved by a direct solution of the unsteady Euler equations, for an ideal multigrid algorithm with linear convergence, in one dimension as

$$\sum^{n} 2^i \Big/ \sum^{n} 2^{-i}$$

For four grid levels ($n = 3$) the ideal speedup is eight times.

In reality, due to additional costs associated with transferring information between grid levels and with possible nonlinear convergence, the figure of merit is smaller. With higher number of grid levels, the performance (speedup) of the multigrid method increases as compared with relaxation of an unsteady Euler solver, and that figure of merit increases with increasing dimension. The one-dimensional problem represents the worst-case scenario and better speedup results are expected in higher dimensions.

2. *Multigrid Results*

Figures 4 and 5 represent the comparison of relaxation to steady state by an MG solver with one grid level and with several grid levels. The initial data for this test was taken to be the output from STEADY solver perturbed by 1% and the number of iterations required to relax to the steady-state solution within prescribed accuracy was measured. The L_2 norm used to measure the convergence to the steady state was calculated as

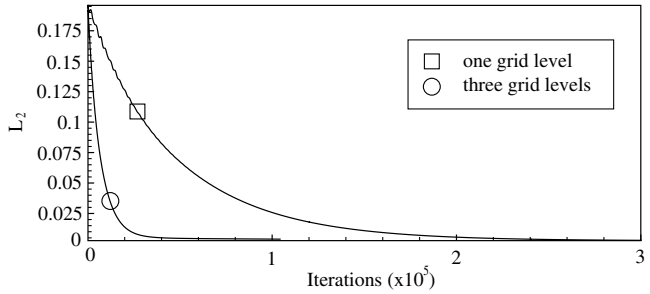

Fig. 4 Error L_2 vs iteration; one grid 800 nodes, three grid levels (200, 400, 800).

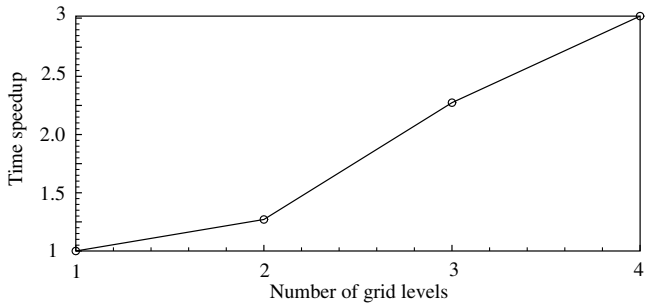

Fig. 5 Time speedup of convergence to steady state as function of number of grid levels.

$$L_2 = \sqrt{\sum^{n} (\bar{p}_i - p_i)^2}$$

where p_i is pressure at grid point i, and $\bar{p}_i$ is the pressure at grid point i calculated by STEADY.

Figure 4 shows eight that to converge to a steady solution, a calculation on one grid with 800 nodes marching at the time step determined by the unsteady Euler equations, required approximately four times more iterations than the multigrid relaxation calculation that used three grid levels (200, 400, and 800). We should mention that the number of iterations, or multigrid cycles, is not a very accurate measure of the simulation speedup because for larger number of grid levels the simulation algorithm requires more calls to the flux calculation subroutine per effective time step (multigrid cycle). To give a better idea of the actual acceleration of steady-state calculation, we present the measurement of the time speedup next.

The time speedup of MG simulation on up to four grid levels is presented in Fig. 5. For this test, the convergence criterion was imposed as L_2 norm of the change between two successive multigrid time steps being less than preset tolerance (10^{-8}). The time speedup on n grids is defined as the ratio of simulation time taken by MG on one grid (i.e., an unsteady Euler relaxation calculation on the finest grid) and simulation time taken by MG on n grids. We see that actual time speedup of the simulation on four grids is equal to three, less than expected for the best-case scenario. Nevertheless, it is a good result which is predicted to improve as we go from one dimension to two and three dimensions and as the number of grid levels is increased.

V. Conclusions and Future Work

The principal conclusion is that starting from a standard compressible Euler equation model of solid rocket motor combustion, one can use rational multiscale asymptotics to pose a problem for the quasi-steady phase of operation of the motor that can be solved to simulate a complete burnout of the grain. Further, we believe that significant to dramatic decreases in required simulation time can be achieved by solving the reduced system on the time scale of propellant regression, instead of the full system whose simulation time step is restricted by the CFL condition on the much smaller acoustic time scale. The one-dimensional model provides an excellent framework to formulate and demonstrate these concepts. The one-dimensional model has a special, steady, frozen solution that can be obtained very simply. The strategy of integrating the reduced equations on the regression time scale relies on finding fast ways to compute steady solutions to the Euler equations (i.e., solutions for the frozen configuration). Determination of the steady Euler flows is the limiting computational step. The methods of computational aerodynamics available in the literature and widely used in air vehicle applications, like multigrid, are directly applicable. We showed that in this model one obtains speedups (relative to direct simulation) that are comparable to those expected from the literature (albeit, well below the specially obtained speedup afforded by using the exact solution available for the one-dimensional model). We do not claim that our implementation is optimal, rather that this is a demonstration in the context of SRM simulation, which shows that standard methods from computational aerodynamics can in fact produce significant simulation speedups. Improvements in the algorithms and implementation of methods that are used to compute steady, compressible Euler flows in and through complex aerodynamic shapes will lead directly to reduced simulation time for the stable burn phase of solid rocket motors.

Future work, some ongoing now at University of Illinois, includes extension to two-dimensional axisymmetric flows and fully three-dimensional flow. These extensions require consideration of time averaging of flows that have weak instabilities and recirculation regions, not unlike those which are required for aircraft flows. Methods other than mulitgrid are being examined for computation of the frozen, quasi-steady Euler flows.

Acknowledgments

The University of Illinois Center for Simulation of Advanced Rockets research program is supported by the U.S. Department of Energy through the University of California under subcontract B341494. D. S. Stewart and S. Yoo have also been supported by the U.S. Air Force Research Laboratory, Munitions Directorate under subcontract F08630-00-1-0002, and by U.S. Air Force Office of Scientific Research, Physical and Applied Mathematics under subcontract F49620-03-1-0048.

References

[1] Salita, M., "Verification of Spatial and Temporal Pressure Distribution in Segmented Solid Rocket Motors," AIAA Paper 89-0298, 1989.

[2] Salita, M., "Closed-Form Analytical Solutions for Fluid Mechanical, Thermomechanical, and Thermal Processes in Solid Rocket Motors," AIAA Paper 98-3965, 1998.

[3] Erberle, A., Rizzi, A., and Hirschel, E. H., "Numerical Solutions of the Euler Equations for Steady Flow Problems," *Notes on Numerical Fluid Mechanics*, Vol. 34, Vieweg, Brunswick, Germany, 1992.

[4] Jameson, A., "Solution of the Euler Equations for Two-Dimensional Transonic Flow by a Multigrid Method," *Applied Mathematics and Computation*, Vol. 13, No. 3, 1983, pp. 327–355.

[5] Jameson, A., "Time-Integration Methods in Computational Aerodynamics," *U. S. Air Force Office of Scientific Research Workshop on Advances and Challenges in Time-Integration of PDEs*, Providence, RI, Aug. 2003.

[6] Sutton, G. P., and Biblarz, O., *Rocket Propulsion Elements*, Interscience, New York, 7 ed., 15 Dec. 2000, p. 45.

[7] Yoo, S., and Stewart, D. S., "A Hybrid Level-Set Method for Modeling Detonation and Combustion Problems in Complex Geometries," *Combustion Theory and Modeling*, Vol. 9, No. 2, 2005, pp. 219–254.

[8] Xu, S., Aslam, T., and Stewart, D. S., "High Resolution Numerical Simulation of Ideal and Non-Ideal Compressible Reacting Flows with Embedded Internal Boundaries," *Combustion Theory and Modeling*, Vol. 1, No. 1, 1997, pp. 113–142.

[9] Fedorenko, R. P., "A Relaxation Method for Solving Elliptic Difference Equations," *USSR Computational Mathematics and Mathematical Physics* , Vol. 1, 1961, pp. 1092–1096.

J. Powers
Associate Editor