ARL-TR-10471 ● Oct 2026

Progress Toward a Robust and Computationally Efficient Burner-Stabilized Premixed Flame Model for Propellant Burning Rate Predictions

by Christopher P. Stone

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-10471 ● OCT 2026





Progress Toward a Robust and Computationally Efficient Burner-Stabilized Premixed Flame Model for Propellant Burning Rate Predictions

Christopher P. Stone
DEVCOM Army Research Laboratory

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
October 2026	Technical Report	06/01/2026	08/31/2026

4. TITLE AND SUBTITLE
Progress Toward a Robust and Computationally Efficient Burner-Stabilized Premixed Flame Model for Propellant Burning Rate Predictions

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Christopher P. Stone

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
DEVCOM Army Research Laboratory ATTN: FCDD-RLA-WC Aberdeen Proving Ground, MD 21005	ARL-TR-10471

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
ORCID: Christopher P. Stone, 0000-0002-9621-5334

14. ABSTRACT
To improve the robustness, computational efficiency, and maintainability of propellant burning rate prediction models, the suitability of the burner-stabilized flame solver, part of the open-source Cantera modeling framework, was studied and compared with the legacy Chemkin-III PREMIX flame model currently used in ARL's CYCLOPS burning rate model. Cantera's internal flame solver was shown to have lower single-core run times (~20%–33%) compared with PREMIX when applied to predicting the burning rate of nitroglycerin, a common benchmark, using both a semi-empirical and detailed chemical kinetics model. Only with a specialized version of PREMIX that supports multicore parallelism is the legacy solver able to overcome Cantera's improved algorithm efficiency. While not currently supported, it is expected that Cantera would be substantially faster with parallelism. To investigate the possibility of a Cantera-based parallel solver, a new flame solver was implemented using Cantera's internal kinetics, transport, and thermodynamic modeling capabilities but with support for parallelism. Promising parallel efficiencies were achieved (i.e., >75%) with the residual function, which is the backbone of the nonlinear flame solver. A semi-analytical Jacobian assembly method was also demonstrated, which reduced the matrix assembly cost by nearly 10 times over Cantera's finite-difference method. To address concerns with Cantera's (and PREMIX's) nonlinear convergence criteria, a new nonlinear flame solver was demonstrated in which the residual error was directly used as the convergence criteria. This reduced predicted residual errors through the peak heat release zones of the propellant flames.

15. SUBJECT TERMS
Weapons Sciences, premixed flame model, propellant burning rate, numerical methods

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UU	34

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Christopher P. Stone	(520) 691-5244

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures iv

Acknowledgments v

Declaration of AI-Assisted Technologies v

1. Introduction 1

2. Methodology 4

3. Results and Discussion 11

3.1 Cantera-based CYCLOPS 11

3.2 Residual-based Convergence 15

3.3 Performance Analysis of the New Solver 17

4. Summary and Conclusions 21

5. References 24

List of Symbols, Abbreviations, and Acronyms 26

Distribution List 27

List of Figures

Figure 1. Conceptualization of the CYCLOPS two-phase combustion model showing the solid propellant and gas-phase combustion regimes. 2

Figure 2. Predictions of neat NG linear burning rates using PREMIX- (solid lines) and Cantera-based (dashed lines) CYCLOPS frameworks. Both NG and DZ (small molecule) chemical kinetics models are used. Results are compared with measurements of Andreev. Note that the NG and DZ predictions with PREMIX and Cantera overlap. 12

Figure 3. Predicted temperature profile and relative difference (%) at the converged burning rate (5,500 atm) using DZ kinetic model with PREMIX- and Cantera-based CYCLOPS. 13

Figure 4. Computational costs of CYCLOPS with Cantera and Chemkin-III PREMIX using the small molecule (a) DZ and larger (b) NG kinetics model to solve the burning rate of neat NG at P = 5,500 atm. Benchmarks conducted on Blueback at the Navy DSRC. Parallel computations with Chemkin-III/PREMIX (i.e., par) used 48 cores with OpenMP. Speed-up values are relative to the serial PREMIX (center) run times, indicating the serial improvement with Cantera and parallel speed-up provided by the OpenMP-based PREMIX solver. 14

Figure 5. Temperature and Y_{NO} residual errors with the new model (top) and Cantera (middle) P = 23.8 MPa. Also showing T and Y_{NO} predicted differences (normalized by inlet boundary values) between Cantera and the new model (bottom) for clarity. 16

Figure 6. Computational cost of the residual function for the (a) DZ and (b) NG chemistry models using Cantera and the proposed flame solver. 17

Figure 7. Jacobian assembly cost for the (a) DZ and (b) NG kinetic models using the brute-force and functional splitting (split) finite-difference methods compared with the semi-analytical approach within the new flame model with and without the frozen transport approximation. Also showing the parallel run-time (and speed-up with symbols) using 16 cores with the frozen transport approximation. For reference, Cantera's assembly times are shown by the horizontal dashed line. 19

Acknowledgments

Funding for this effort was provided by ARL. Computational resources were provided by the DoD High-Performance Computing Modernization Program. Computations were conducted on the Blueback system at the U.S. Navy DoW Supercomputing Resource Center.

Declaration of AI-Assisted Technologies

Claude Opus 5 was used for software model development supporting this research. All research has been personally verified by the author.

1. Introduction

The constant pressure, burner-stabilized, premixed flame model[1] is the primary gas-phase component of the multiphase, physics-based propellant burning rate prediction model, CYCLOPS.[2,3] The CYCLOPS propellant burning rate model, based on the work of Miller and Anderson,[4] predicts burning rates by modeling a gas-phase flame anchored above a pyrolyzing propellant. This method has been demonstrated on both neat energetic materials (e.g., nitroglycerin [NG][5,6]) and single-, double-, and triple-based solid propellants for gun and rocket propulsion applications.[2,7]

The linear burning rate, r, is predicted by iteratively solving the interface energy and mass flux balance equations across the burning propellant and the reacting gases immediately above the surface. This is conceptually rendered in Figure 1. Neglecting condensed-phase reactions and diffusion, the interface energy flux conservation equation is

$$\left[\lambda \frac{\mathrm{dT}}{\mathrm{dz}}\right]^{+0} = \dot{m}\left[\sum_{k}^{K} {Y_k}^{-0}\ {h_k}^{+0}\ - {h_c}^{-\infty}\right] \tag{1}$$

where λ is the gas-phase mixture thermal conductivity, ${h_k}^{+0}$ and Y_k^{-0} are the gas-phase enthalpies and mass fractions for the K chemical species, ${h_c}^{-\infty}$ is the condensed-phase bulk enthalpy far from the interface, $\dot{m}$ is the mass flow rate, T is the temperature profile, and z is the special coordinate. The superscripts $+0$ and -0 indicate properties evaluated (or defined) on the gas- and condensed-phase sides of the interface, respectively.

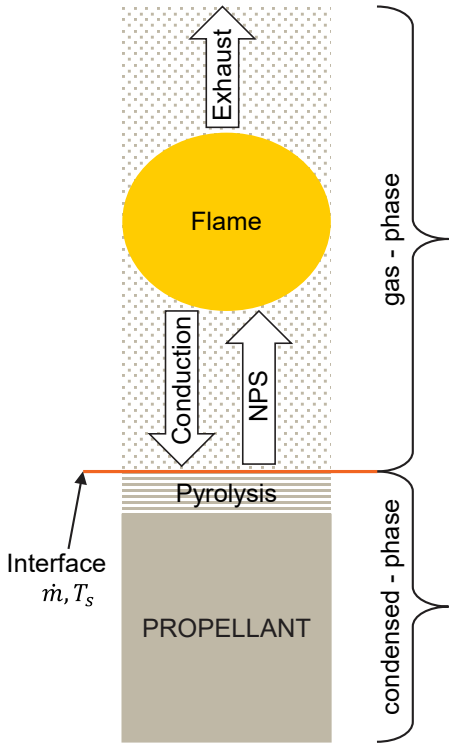


Figure 1. Conceptualization of the CYCLOPS two-phase combustion model showing the solid propellant and gas-phase combustion regimes.

Equation 1 is solved iteratively to determine the eigenvalue $\dot{m}$ at a single pressure. The burning rate, r, is determined then from the relation $\dot{m} = \rho_c r$, where ρ_c is the condense-phase mass density. Note that conservation of mass requires that $\dot{m}$ is constant across the interface. A second equation, defining the relationship $\dot{m} = \dot{m}(T_s, Y_k^{-0})$, is commonly included so that both $\dot{m}$ and T_s, the surface temperature, can be predicted simultaneously.

The gas-phase flame (see Figure 1) is represented as a 1-D, steady-state, burner-stabilized premixed diffusion flame. At each iteration when iteratively solving Equation 1, a new burner flame is solved with the trial interface boundary conditions $\dot{m}$, T_s, and Y_k^{-0}. Here, Y_k^{-0} are the mass fractions of the gases that are assumed to emanate from the propellant during pyrolysis. This nascent product set (NPS) gases are currently assumed to be constant and independent of the condensed-phase temperature and pressure. The burner flame model and overall energy balance assumes constant pressure.

To predict the pressure dependence on the burning rate, a sequence of CYCLOPS predictions is conducted over a range of pressures relevant to the modeling application. These tabulated burning rates are then commonly fitted in a Vielle's law form, $r(P) = \alpha P^{\beta}$, where α and β are determined from a least-squares method.

As summarized in Stone et al.,[3] recent updates to the CYCLOPS solver improved the efficiency and robustness of the original method to solve Equation 1 and show reasonable predictions of single- and double-base propellants (e.g., M10, JA2). The current CYCLOPS model uses the legacy (pre-commercial) Chemkin-III combustion modeling framework[8] for gas-phase chemical kinetics, thermodynamic, and transport property calculations. It also uses the companion PREMIX burner flame solver[9] to solve the gas-phase flame above the propellant. While these legacy libraries and solvers, last released in 1999, have been successfully used in CYCLOPS for over 25 years, there are limitations in terms of performance, robustness, and maintainability.

The Cantera[10] framework is an open-source software framework that provides the necessary gas-phase kinetic rates, thermodynamic and transport properties, and flame solvers needed by CYCLOPS. It is actively maintained, continuously developed, and includes state-of-the-art rate parameterizations not supported by Chemkin-III, such as pressure-interpolation (PLOG) and state-of-the-art mixture rules for composition-dependent rate expressions that have shown to be influential when predicting propellant flames that contain high mole fractions of efficient third-body colliders.[11,12] Cantera also provides multiple canonical premixed and non-premixed flame solvers, which closely follow the original algorithms used in PREMIX to solve the nonlinear, steady-state flame equations (detailed later) and to dynamically adapt the mesh to resolve steep gradients in the flame and at the burner surface.

In-house enhancements at U.S. Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory (ARL) have added the PLOG rate parameterization to Chemkin-III to improve pressure-dependent predictions, add parallelism to the PREMIX flame solver to reduce computational times,[7] and port several recent advances from Cantera's flame solver workflow into PREMIX to improve robustness and efficiency.[3] However, these efforts to maintain the legacy Chemkin-III libraries and solvers are costly and further algorithm enhancements are constrained by the overall architecture and design of Chemkin-III.

While Cantera contains all the necessary modeling capabilities for a CYCLOPS implementation, it lacks parallel computing capabilities (e.g., multicore parallelism), which can limit the throughput for burning rate predictions. Cantera shares (with minor differences) the original Chemkin-III nonlinear solver strategy based on a damped, modified Newton–Raphson method with optional pseudo-time integration for robustness to solve the residual errors in vector form of the steady-state, 1-D flame equations. Unlike most Newton solvers that monitor a norm of the residual vector as the metric for convergence and progress, the Chemkin-III and Cantera flame solvers iterate until the Newton *correction* falls below some relative

threshold. While this has proven to be robust for decades in predicting premixed flame-speed calculations, it is possible for unconstrained residual errors to remain, which may impact the sensitive predictions near the burner surface. This can also impact the convergence of CYCLOPS, which uses an iterative Newton (or Secant) method itself, since small corrections to $\dot{m}$ and T_s may not produce continuous (monotonic) changes to the flame solution near the burner surface. This limits the accuracy to which Equation 1 can be solved, which in turn limits the application of optimization methods to define the NPS of new ingredients.[13]

The goals of this effort were to 1) examine the feasibility of building a CYCLOPS implementation using Cantera's existing burner-stabilized premixed flame solver and its internal kinetic, thermodynamic, and transport libraries; and 2) construct a stand-alone burner flame solver that uses Cantera's internal kinetic, transport, and thermodynamic libraries but with a more rigorous nonlinear solver to directly enforce residual convergence and in a manner that enables parallel computing and other robustness and performance improvements.

2. Methodology

In this section, the burner-stabilized, premixed flame model and nonlinear solver implementation are described. As noted, the goal is to create a new burner flame model that supports parallelism and improved efficiency compared with Cantera's internal method. Further implementation details of CYCLOPS can be found elsewhere[2–4] and are not repeated here.

The steady-state, burner-stabilized premixed diffusion flame governing equations can be derived from the generalized quasi–1-D, constant-pressure, axisymmetric, diffusion flame model equations of Kee et al.[1] by neglecting any radial gradients and assuming the stream tube cross-sectional area is constant. This results in the following set of ordinary differential equations (ODEs) governing the conservation of energy and chemical species mass fractions.

Energy:

$$\frac{\dot{m}}{\rho}\frac{dT}{dz} = \frac{1}{\rho c_p}\left[\frac{d}{dz}\left(\lambda\frac{dT}{dz}\right) - \sum_k j_k \frac{dh_k}{dz} - \sum_k h_k W_k \omega_k\right] \tag{2}$$

Mass fractions (Y_k):

$$\frac{\dot{m}}{\rho}\frac{dY_k}{dz} = \frac{1}{\rho}\left[-\frac{dj_k}{dz} + W_k\omega_k\right] \tag{3}$$

The gases are treated as a thermally perfect mixture so that Equation 2—the energy (enthalpy) equation—can be cast into an equation for gas temperature, T. This assumption may be strained in a gun propellant combustion environment where pressures may approach 10^3 MPa.

In Equations 2 and 3, the mass flow rate ($\dot{m}$) is prescribed (and constant) and the stream tube cross-sectional area is constant throughout the entire domain (z). Therefore, the gas velocity is found from the local thermodynamic state alone, i.e., $u(z) = \dot{m}/\rho(T, Y_k)$. The remaining terms are the *mixture* thermal conductivity (λ) and mass specific heat capacity (c_p), as well as the *species* partial molar enthalpy (h_k), molecular weight (W_k), diffusive mass flux (j_k), and net molar production rate due to chemical reactions (ω_k).

Diffusive transport terms (i.e., thermal conduction and species diffusion) are evaluated using the mixture-averaged approximation. Thermal diffusion effects (i.e., Soret) are ignored. The diffusive mass flux is evaluated as,

$$j_k^0 = -\rho D_{k,m}\frac{dY_k}{dz} \tag{4}$$

where $D_{k,m}$ is the mixture-averaged diffusion coefficient. This is corrected to enforce mass conservation (i.e., $\sum j_k = 0$) since the mixture-average approximation introduces non-conservative errors, i.e.,

$$j_k = j_k^0 - \left(\sum_m j_m^0\right) Y_k. \tag{5}$$

The flow is assumed to be in the $+z$ direction with the inlet (i.e., burner surface) at $z = 0$ and the outflow at $z = L$. The burner flame model prescribes the inlet composition (Y_k^0) and temperature (T^0) along with the constant $\dot{m}$. Since the flow is diffusion-dominated near the burner surface, a Danckwerts boundary condition is applied at the inlet. Equations 6 and 7 are solved at the inlet boundaries:

$$j_k + \dot{m}Y_k = \dot{m}Y_k^0\,|_{z=0} \tag{6}$$

and

$$T = T^0 \mid_{z=0}. \tag{7}$$

Zero-gradient boundary conditions (i.e., extrapolation) at $z = L$ are used to enforce zero diffusion in the far field. As such, the domain length (L) must be long enough so that the zero-gradient and nonreactive conditions are realized. That is, the exit conditions are

$$\frac{dY_k}{dz} = \frac{dT}{dz} = 0 \mid_{z=L}. \tag{8}$$

The governing ODEs and boundary conditions in the flame solver proposed here are discretized using finite-difference on a nonuniform mesh that closely matches Cantera's implementation. By matching the discretization, the solution with the new model can be cross validated easily and attention can be given to the impact on robustness of the nonlinear solver and performance and efficiency of the computational kernels.

Like Chemkin-III and Cantera, the residual form of the steady-state equations is solved using a modified Newton–Raphson method. That is, a solution to the vector residual function $F(x) = 0$ is sought starting from some initial guess (x) of the flame profile at the discretized solution points. The residual equations are highly nonlinear, and the initial guess is often far from a solution. As a result, the Newton solver frequently fails to converge. To overcome this common difficulty, multiple pseudo–time steps are taken to drive the solution closer to a converging well. Since time accuracy is not needed, a simple implicit backward Euler integration method is used resulting in the following modified residual equation:

$$F(x)^{unsteady} = F(x)^{steady} - \frac{x - x^{\text{prev}}}{\Delta t}. \tag{9}$$

Here, Δt is the pseudo–time-step size and x^{prev} is the solution at the previous time step. When Δt is small, the condition number of the system is reduced significantly (i.e., the diagonal terms become relatively large) making the pseudo–time step nonlinear system easier to solve. This relaxation approach is a common strategy for solving nonlinear systems when starting from a poor initial guess. And, since the pseudo–time step equations have the same structure as the steady equations, the numerical method and nonlinear solvers can be reused.

Newton methods solve the nonlinear residual equations iteratively. Starting from an initial guess x^0, the solution is updated as

$$x^{k+1} = x^k + \alpha\, \delta^k \tag{10}$$

Here, α is a (scalar) damping term and δ^k is the correction (or step length) vector. In a Newton method, this is found by solving the following linear system:

$$J(x^m)\,\delta^k = -F(x^k) \tag{11}$$

where J is the system Jacobian (i.e., $J = \partial F / \partial x$). Since the Jacobian is only used to drive the system to convergence (and is not part of the solution itself), the Jacobian is often reused for several iterations to reduce computational cost (i.e., $m \leq k$) as long as the system is converging at an acceptable rate.

The damping term ($0 < \alpha \leq 1$) is found from a *line search* algorithm designed to optimize the reduction in the residual norm. A traditional (undamped) Newton step uses $\alpha = 1$. When a line search fails (i.e., α cannot be found that lowers the residual *sufficiently*) and the Jacobian is out of date (i.e., $m \neq k$), the Jacobian is updated with the latest solution, δ^k is recomputed, and the search is tried again. Otherwise, the Newton solver fails.*

The iterations are continued until the absolute (or relative) residual normal falls below a threshold (i.e., $\|F(x^k)\| \leq F_{tol}$) or the correction step becomes too small to continue (i.e., $\|\delta^k\| \leq s_{tol}$). If the latter condition is reached with an out-of-date Jacobian, it is common to retry the step with an updated Jacobian before accepting this pseudo-convergence.

While the governing equations and numerical discretization follow close to Cantera, the assembly of the system Jacobian matrix (*J*) is implemented differently to improve accuracy and efficiency. In Cantera (and previously in Chemkin-III), the Jacobian is assembled using one-sided finite-differences. This requires $3N$ evaluations of the residual function where N is the number of unknowns at each mesh point (i.e., $N = K + 1$ in the burner flame model). Note that the *3N* evaluations arise from the mixed first and second-order finite-difference discretization of the governing equations in which each mesh point *i* is only dependent upon its immediate neighbor points at *i-1* and *i+1*. The resulting Jacobian matrix is block-tridiagonal with square NxN blocks and *M* block rows (*M* is the number of mesh points, including the boundaries). This structure is independent of the assembly method (i.e., finite-difference or analytical); however, various approximations may alter the sparsity of the diagonal or off-diagonal blocks.

The finite-difference assembly can be done in parallel (i.e., multiple columns concurrently) but it is still a significant cost. See Stone[7] for performance details on the parallel assembly of finite-difference Jacobians in a related diffusion flame model. Assembling *J* analytically has the potential for reducing the *assembly* time and improving the scalability of this phase of the nonlinear solver workflow as the

* If the Newton solver fails to solve the steady-state problem, several pseudo–time steps are taken. If it fails during pseudo–time stepping, Δt is reduced and the step is retried.

number of species increases. However, the high cost of matrix *factorization* is unchanged. The computational cost of direct matrix factorization scales as $O(MN^3)$, which can dominate as the number of chemical species increases.

Here, the Jacobian is assembled using a *semi-analytical (SA)* approach. The Jacobian is split into three contributions: $J = J^{chemistry} + J^{transport} + J^{dt}$. The chemistry and pseudo–time step terms are evaluated analytically while the transport terms still use the one-sided finite-differences method as before. Note that the chemistry and pseudo–time step Jacobian terms only apply to the diagonal blocks and are independent across the computational domain. As such, they can be evaluated concurrently (i.e., in parallel) across the mesh points.

Cantera* provides analytical derivatives for $\partial\omega_k/\partial C_j$ as part of its kinetics library. Here, C_j is the molar concentration of species j. This building block is used to construct the chemistry contributions to $\partial F/\partial Y_i$ (i.e., derivations with respect to mass fractions). Note that any change in Y_k impacts all molar concentrations through ρ via the definition $C_i = \rho Y_i/W_i$, which results in $\partial C_i/\partial Y_i = \rho/W_i$. In general, using a mass fraction formulation with detailed kinetics results in dense Jacobian blocks. Approximations[14] are needed to introduce sparsity.

The full expansion of the chemistry Jacobian terms for mass fraction is

$$\frac{\partial F_{Yk}}{\partial Y_i} = \frac{W_k}{W_i}\left[\frac{\partial \omega_k}{\partial C_i} - v_k + \frac{\omega_k W_{\text{mix}}}{\rho}\right] \tag{12}$$

where W_{mix} is the mixture (mean) molecular weight. The term $v_k = \sum_m \frac{\partial \omega_k}{\partial C_m} X_m$ accounts for changes to all species molar concentrations due to ρ and the last term comes for $\partial\rho/\partial Y_i|_T = -\rho W_{\text{mix}}/W_i$.

Similarly, the chemistry Jacobian for the temperature equation is

$$\frac{\partial F_T}{\partial Y_i} = \frac{1}{c_p W_i}\left[-\sum_k h_k \frac{\partial \omega_k}{\partial C_i} + \sum_k h_k\, v_k + \frac{Q W_{\text{mix}}}{\rho}\right] \tag{13}$$

where $Q = \sum_k \omega_k\, h_k$ is the net heat release.

The single column of the chemistry Jacobian accounting for derivatives with respect to temperature (i.e., $\partial F/\partial T$) is currently evaluated with one-sided finite-differences for simplicity. This only requires a single *additional* residual evaluation (at a perturbed state) of the chemical residual since the analytical terms above already require an evaluation at the current state. Note that it is possible to evaluate

* This is provided by the function `netProductionRates_ddCi`.

this column analytically and this may be evaluated in the future. However, Cantera currently (as of v3.2.0) does not provide a function for $\partial\omega_k/\partial T$.

When pseudo–time stepping is used (i.e., $\Delta t \neq \infty$), the steady Jacobian is first computed and then the diagonal elements are updated with the single analytical term $J^{dt} = -1/\Delta t$ for all rows in the current burner flame model.

The $J^{transport}$ terms are currently approximated using finite-difference. To reduce the cost, the species diffusion and thermal conductivity terms are computed at the current state and then held fixed. That is, terms such as $\partial D_{k,m}/\partial T$ are neglected in the Jacobian. This *frozen transport property* approximation, used in both Cantera and legacy PREMIX implementations, has been shown to be robust and is quite beneficial in terms of computational cost. The impact of changes to T and Y_k on thermodynamic terms in the advection and diffusive fluxes (e.g., $\partial\rho/\partial T$ and $\partial h_k/\partial Y_i$) are still accounted for in this approximation. Future efforts will evaluate the merit of assembling the $J^{transport}$ terms analytically under the frozen property assumption.

Cantera, like the original PREMIX flame solver, stores the block tridiagonal matrix in the BLAS/LAPACK banded matrix storage format (with $2N - 1$ upper and lower bandwidths) and factorizes and solves the linear system (i.e., Equation 11), using LAPACK routines `DGBTRF` and `DGBTRS`. I demonstrated an optimized block tridiagonal format (assuming equal block sizes) and related factorization and solver routines in Stone,[7] which improved the computational performance and reduced memory usage. For simplicity, the standard band storage format with LAPACK factorization methods is used here to, again, focus on the effect of the assembly method.

Lapointe et al.[14] examined the use of sparse approximations for the chemical and transport Jacobians for premixed and non-premixed flames with Jacobian-Free Newton–Krylov (JFNK) methods. This approach can provide significant performance and scalability benefits since the system Jacobian is not explicitly constructed. In JFNK methods, Equation 11 is solved iteratively with a Krylov method (e.g., generalized minimal residual [GMRES] method). The key benefit of this approach is that only the Jacobian-vector product is needed, which can be approximated without the Jacobian itself being assembled explicitly. In Lapointe et al.,[14] an approximate Jacobian was still assembled (and factorized) and used as a *preconditioner* in the GMRES linear solver. Larger errors can be tolerated in preconditioners than direct Newton solver Jacobians, allowing for more approximations. Lapointe et al. approximated analytical or SA chemical and transport Jacobians (e.g., neglecting third-body collision contributions) combined with operator splitting methods to significantly reduce factorization costs in their

preconditioner, demonstrating significant performance benefits but with consequential impacts on solver robustness. Open-source versions[15] of the JFNK-based flame models are maintained by Lawrence Livermore National Laboratory. However, the burner-stabilized flame model is not supported, and significant effort would be required to add the necessary boundary conditions within the existing framework. While this approach shows promise for performance, model development is needed to improve robustness, and the workflows must be tailored to work within CYCLOPS.

As noted earlier, PREMIX and Cantera use an internal Newton solver that differs from common practice. First, their metric for convergence is solely based on the norm of the weighted (or relative) step length δ; the residual error is not monitored nor a tolerance enforced. Second, the simple line-search algorithm searches for a damping term, α, that gives $\|\delta^{k+1}\| < \|\delta^{k}\|$. Again, the residual error is not considered. Overall, both the Newton and line-search algorithms can pass step convergence criteria but result in an uncontrolled growth in the residual error. These flame solvers have been successfully used and accepted by the combustion modeling community for decades. However, a more rigorous approach is sought that directly controls the steady-state error.

Open-source packages such as *Kinsol* (part of the SUNDIALS[16] framework) provide generalized implementations of direct Newton and indirect JNFK solvers. Kinsol includes several efficient line-search algorithms and other advanced numerical methods (e.g., JFNK and direct solvers) to improve robustness and performance. However, Kinsol lacks a method comparable with PREMIX or Cantera to constrain the mass-fractions efficiently and negative Y_k can occur, leading to undefined and unphysical behavior. Further, official Python support for Kinsol was released during this investigation and Cantera (v3.2) does not yet officially support this new release of SUNDIALS. For these reasons, a simple modified, direct Newton solver with non-optimal line-search algorithm was created that directly monitors the residual error and, optionally, the step length. This is paired with a simple time-stepping algorithm. These were sufficient to evaluate the feasibility of directly solving the residual error and for assessing the impact of SA Jacobians. Future work will examine the use of the Kinsol solver framework and its impact on performance and robustness.

The new burner flame model was implemented in a mixture of Python and C++. The computationally intensive residual function and Jacobian assembly functions, which rely heavily on Cantera's C++ library, were implemented in C++. The internal Newton solver, time-stepping methods, and other high-level data management tasks were implemented in Python (v3.14) for rapid prototyping. The C++ functions were exposed to the high-level Python solver through pybind11.

3. Results and Discussion

Before examining the impact of the Jacobian assembly method and feasibility of a solver with direct residual convergence, the effectiveness of using Cantera within a CYCLOPS burning rate prediction framework is evaluated. The CYCLOPS model was reimplemented using Cantera's thermodynamic, kinetic, and transport property methods for perfect gas mixtures to evaluate Equation 1. The CYCLOPS framework outlined in Stone et al.[3] is closely followed to allow direct comparison. Cantera's built-in burner-stabilized flame model (*BurnerFlame*) was used to solve the flame at each iteration of the CYCLOPS algorithm.

3.1 Cantera-based CYCLOPS

Cantera and Chemkin-III use different physical units (i.e., MKS and CGS, respectively). Therefore, care was taken to define comparable convergence metrics and initial guesses when solving Equation 1. To avoid round-off differences caused by the very different energy flux units, convergence was defined entirely by the relative change in $\dot{m}$ (i.e., step convergence tolerance was 10^{-4}). Additionally, mesh refinement parameters and initial mesh point distribution were matched. Despite these efforts, the converged flame solutions and final burner rates are not expected to match exactly due to the highly nonlinear nature of the combustion model and subtle numerical and implementation differences between Cantera's burner-flame and PREMIX's version.

The burning rate of neat nitroglycerin (NG) was used as a test case and computational benchmark. Two chemical kinetics models were used: 1) the small molecular "dark-zone" (DZ) kinetics model[4] suitable for a range of nitrate esters and nitramines; and 2) an NG kinetics model.[6] The DZ model has 59 species and 368 reactions, and the NG model has 227 species and 1,594 reactions. The same initial mass flux, surface temperature, and mesh were used for both kinetics models. The primary difference between the use of the two kinetics models is the treatment of the surface gas composition. The DZ kinetics model only contains small gas molecules (e.g., C_2 and smaller) and the pyrolysis gas products, which may be large fragments of the energetic ingredients, must be represented as a collection of these small molecules. In this scenario, the pyrolysis gas composition is specified on an empirical or semi-empirical basis to match measured burning rate data. With the NG model, the product gas is modeled as pure, gaseous NG (i.e., the condensed phase is assumed to vaporize, not fragment) and semi-empirical tuning is not necessary.

Both CYCLOPS models use a simplified pyrolysis model of Zenin[17] to relate mass flux and temperature at the burner surface. That is, the surface temperature is found

by solving $\dot{m}(T_s) = \rho_c\, A\, \exp(T_s\, /B)$ with $A = 1500\, cm/s$, $B = 5000\, K$, and $\rho_c = 1.59\, g/cm^3$ (A, B, and ρ from Zenin[17]) at each CYCLOPS iteration given the predicted $\dot{m}$. In this approach, only the $\dot{m}$ is solved iteratively and T_s is evaluated directly from this at each iteration.

The predicted linear burning rates using PREMIX- and Cantera-based CYCLOPS implementations are shown in Figure 2. While the two kinetics models give different predictions, the difference between the two CYCLOPS implementations is negligible relative to the uncertainties associated with the overall model itself.

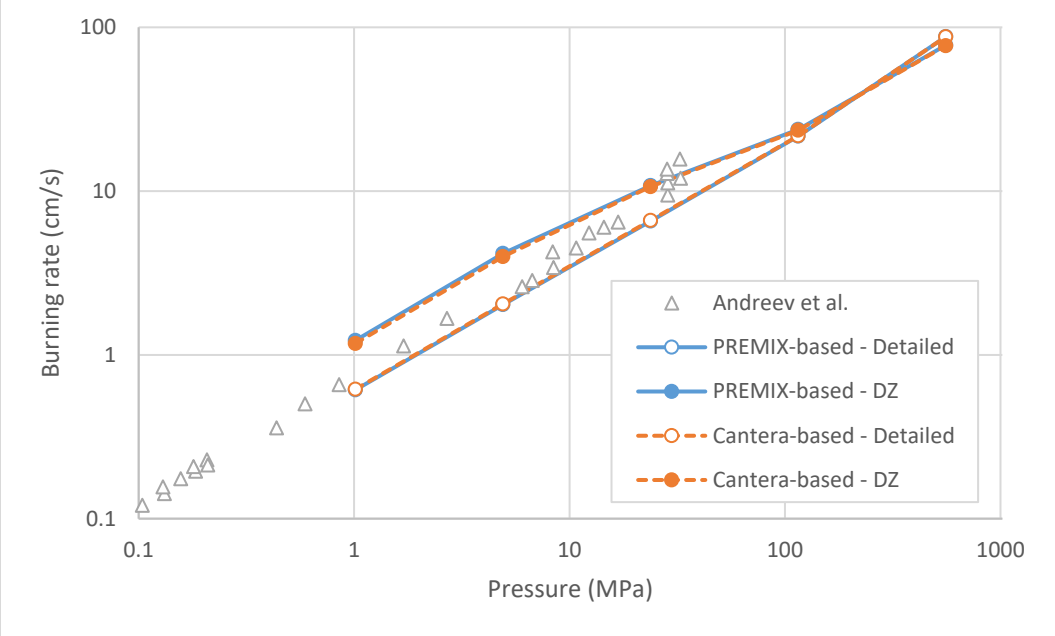


Figure 2. Predictions of neat NG linear burning rates using PREMIX- (solid lines) and Cantera-based (dashed lines) CYCLOPS frameworks. Both NG[6] and DZ (small molecule) chemical kinetics models are used. Results are compared with measurements of Andreev.[4] Note that the NG and DZ predictions with PREMIX and Cantera overlap.

The predicted temperature profile and relative difference at the converged burning rate (at $P = 5500\, atm$) using the DZ kinetics model is shown in Figure 3. Since the converged meshes do not match (node for node), the PREMIX solution is projected onto the Cantera mesh using cubic spline interpolation. As seen, the difference in the predictions is small overall. The largest difference in the temperature profile, though small, is at the burner surface. This is due to the small difference in the predicted burning rate, which directly impacts T_s as described above from the pyrolysis closure model. Note that the temperature gradient is the primary driver in the energy flux balance model, not the temperature itself.

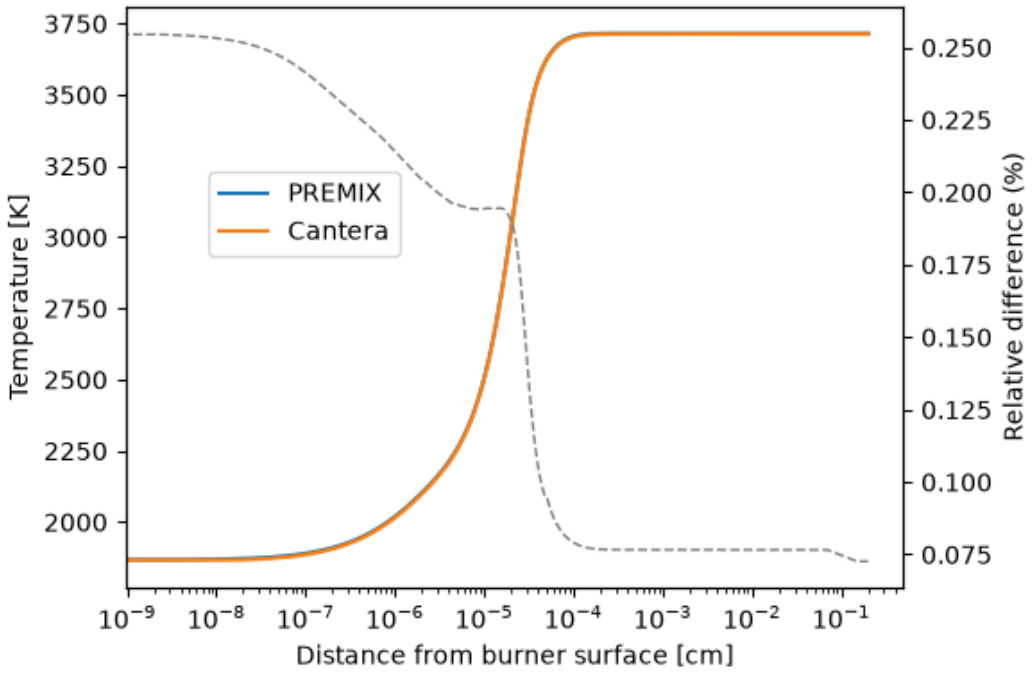
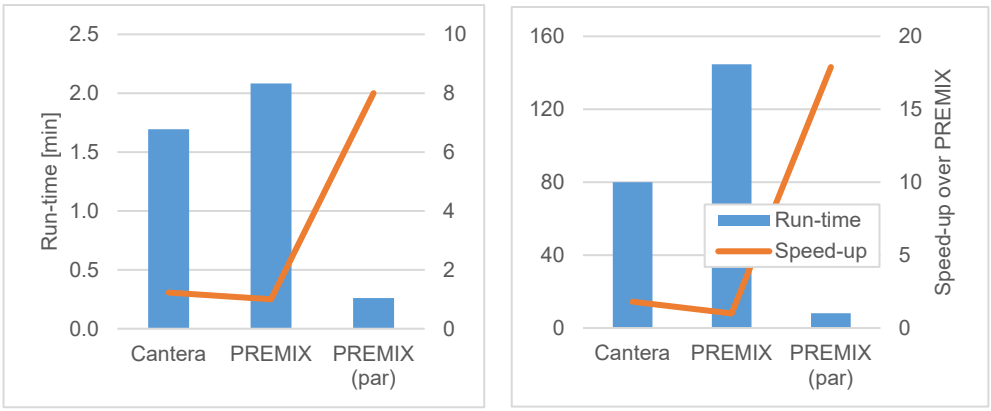


Figure 3. Predicted temperature profile and relative difference (%) at the converged burning rate (5,500 atm) using DZ kinetic model with PREMIX- and Cantera-based CYCLOPS.

Figure 4 shows the computational cost of the Cantera- and Chemkin-III–based CYCLOPS solvers with the small DZ and larger detailed NG kinetics model. The benchmark was conducted on the *Blueback* system at the U.S. Navy DoW Supercomputing Resource Center (DSRC). Blueback's computational nodes have two AMD EPYC 9654 96-core CPUs. Both PREMIX and Cantera were compiled with the Clang AMD Optimizing C/C++/Fortran (AOCC) Compilers v4.1.0. Cantera v3.2.0 was used for all benchmarks. Multicore parallelism was enabled with OpenMP (v3.0) to provide multiple threads of execution.



(a) DZ (b) NG

Figure 4. Computational costs of CYCLOPS with Cantera and Chemkin-III PREMIX using the small molecule (a) DZ and larger (b) NG kinetics model to solve the burning rate of neat NG at P = 5,500 atm. Benchmarks conducted on Blueback at the Navy DSRC. Parallel computations with Chemkin-III/PREMIX (i.e., par) used 48 cores with OpenMP. Speed-up values are relative to the serial PREMIX (center) run times, indicating the serial improvement with Cantera and parallel speed-up provided by the OpenMP-based PREMIX solver.

For this benchmark, CYCLOPS was run at 5,500 atm with the same starting conditions for both solvers. As Table 1 shows, the converged CYCLOPS burning rates and temperature profiles match closely within the uncertainty of the model and tolerances. There were, however, differences in the final mesh size, which can influence the computational cost. For example, with the detailed NG mechanism, Cantera converged with 206 mesh points while Chemkin-III/PREMIX required 292. Note that the residual function and Jacobian assembly and factorize times all scale linearly with the mesh size.

To elucidate the cost differences with Chemkin-III/PREMIX and Cantera, the overall CYCLOPS solution times in Table 1 are not normalized by the mesh size nor the number of iterations required to solve Equation 1. We see that Cantera provides a 19%–33% run-time improvement over PREMIX when using only a single core. The faster run-times with the Cantera solver are due to the smaller overall mesh sizes and (not quantified above) a more efficient (and more aggressive) pseudo–time-stepping algorithm. Combined, these overcome the more efficient block-triagonal matrix solver available in the ARL-specific PREMIX flame solver (see Stone[7]). Note that it is possible to implement this optimized matrix solver in Cantera; however, that would require significant refactoring to align with Cantera's object-oriented design, which abstracts the Jacobian matrix assembly and related linear operations.

Recall that Cantera does not support parallelism natively. When using 48 cores, solutions converged 6.5 and 9.9 times faster with ARL's in-house parallel PREMIX solver compared to the serial Cantera run-times. Speed-up is not universal and depends on the specific flame modeling conditions since Cantera has subtle algorithm improvements over PREMIX (e.g., more aggressive time relaxation) that can accelerate the solution.

3.2 Residual-based Convergence

The new burner-flame model using Cantera's thermodynamic, kinetics, and transport libraries is now examined regarding the feasibility of solving the nonlinear flame equations with direct control over the residual error.

The new burner model was used to solve the flame profile at $P = 23.7\ MPa$ using the same burning rate parameters ($\dot{m}$ and T_s) from the converged CYCLOPS predictions reported above as well as the domain width and initial mesh. Since comparisons will be made with the Cantera flame solutions, the Cantera-based burning rate predictions were used. The residual function tolerance was set uniformly to $ftol = 1$ along with a default step tolerance of $stol = \epsilon^{2/3} \approx 3.67 \times 10^{-11}$.

The residual function errors for the temperature and NO mass fraction equations (i.e., F_T and $F_{Y_{NO}}$) from the final, converged flame solution predicted by the new flame model (top) and Cantera (middle) are shown in Figure 5. NO is an important gas product in the burning of nitrate ester and nitramine-based propellants and is highly reactive. The predicted T and Y_{NO} differences between the two solvers (bottom) are also shown. Note that the differences were normalized by the inlet boundary values for clarity.

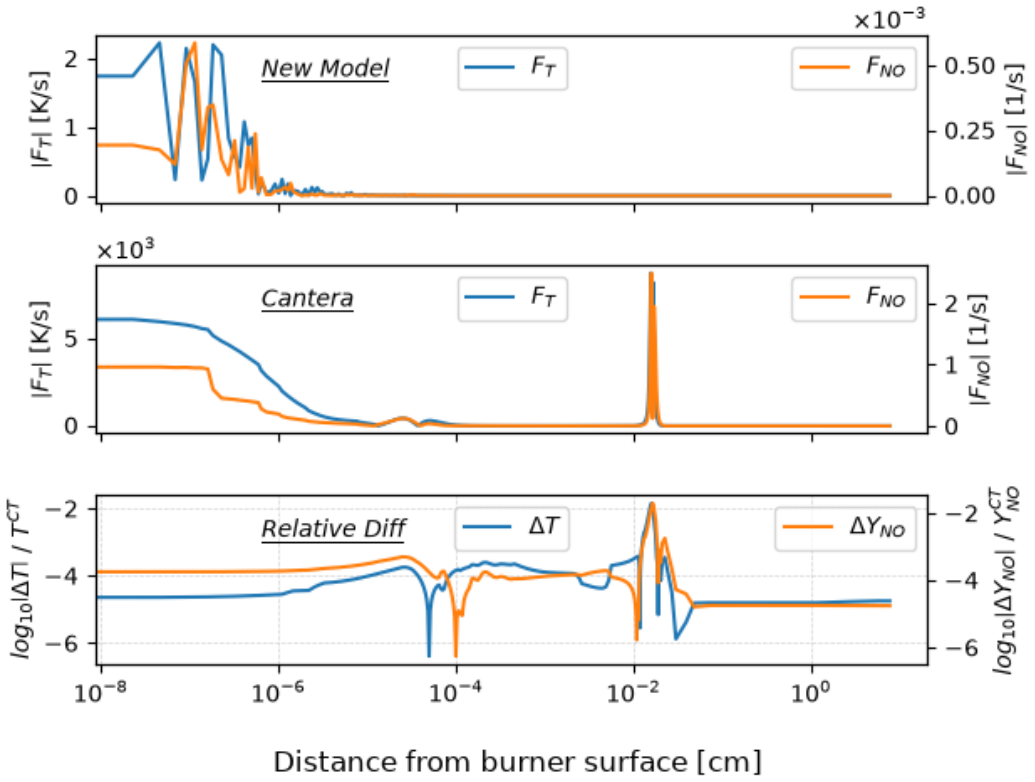


Figure 5. Temperature and Y_{NO} residual errors with the new model (top) and Cantera (middle) $P = 23.8$ MPa. Also showing T and Y_{NO} predicted differences (normalized by inlet boundary values) between Cantera and the new model (bottom) for clarity.

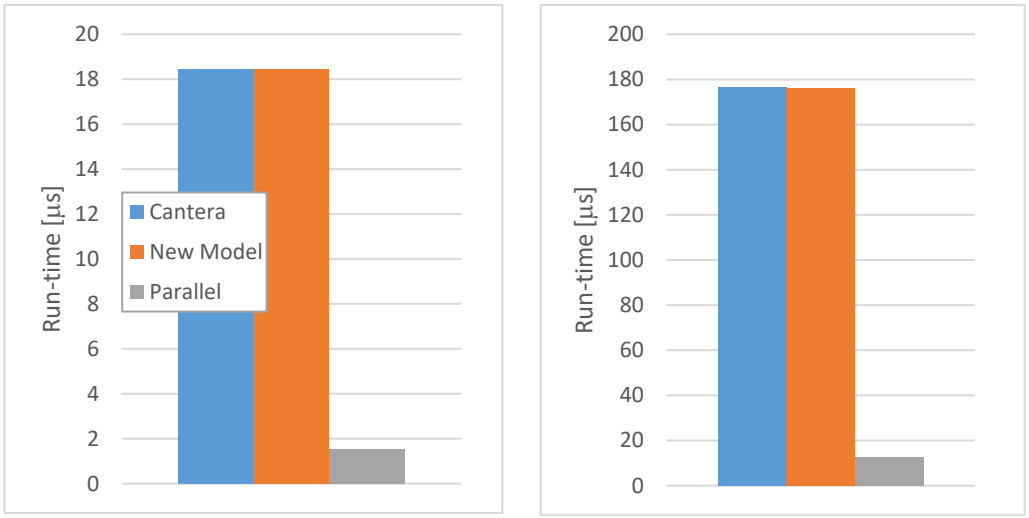
Large residual errors in F_T ($\gg 1000\ K/s$) are observed in the Cantera solution near the burner surface and in the secondary flame corresponding to the peak heat release location (i.e., the second temperature rise in the temperature profile). Likewise, the species mass fraction residual error, $F_{Y_{NO}}$, peaks at the same locations. Since the burning rate prediction model is dependent on the temperature gradient and composition at the surface, these errors, which are unavoidable in an iterative solver, may impact the CYCLOPS predictions. Residual errors at the peak heat release may alter the flame speed calculation for free-flame models. Flame speeds are a common validation metric for chemical kinetics models and improved reliability could have implications for those efforts. However, the secondary flame profile is of limited concern for burning rate predictions.

The energy residual error is greatly reduced compared to Cantera with peak reductions of over 1000 K/s. (The Cantera residuals in Figure 5 are scaled by 10^3 for visualization.) Note that the residual temperature error in the converged solution with the new model is above the specified tolerance since the step length threshold was reached. A comparable reduction was achieved for the mass fraction, too. The most prominent differences to the predicted temperature and composition profiles

are in the secondary flame near the maximum heat release zone. Changes are also seen through the primary flame (i.e., the first temperature plateau). Despite the high residual errors near the burner surface in the Cantera solution, the impact on the predicted near-surface temperature and composition appears minor. The impact on the predicted burning rate will be assessed in future studies.

3.3 Performance Analysis of the New Solver

The performance of the residual function and SA Jacobian assembly method are now examined. Figure 6 shows the cost of the residual function *per point*, evaluated by the prototype burner flame model compared with Cantera's internal implementation for the small DZ and larger NG chemistry model using a single core. The run-time using the new model using 16 cores on the Blueback system is also shown.



(a) DZ (b) NG

Figure 6. Computational cost of the residual function for the (a) DZ and (b) NG chemistry models using Cantera and the proposed flame solver.

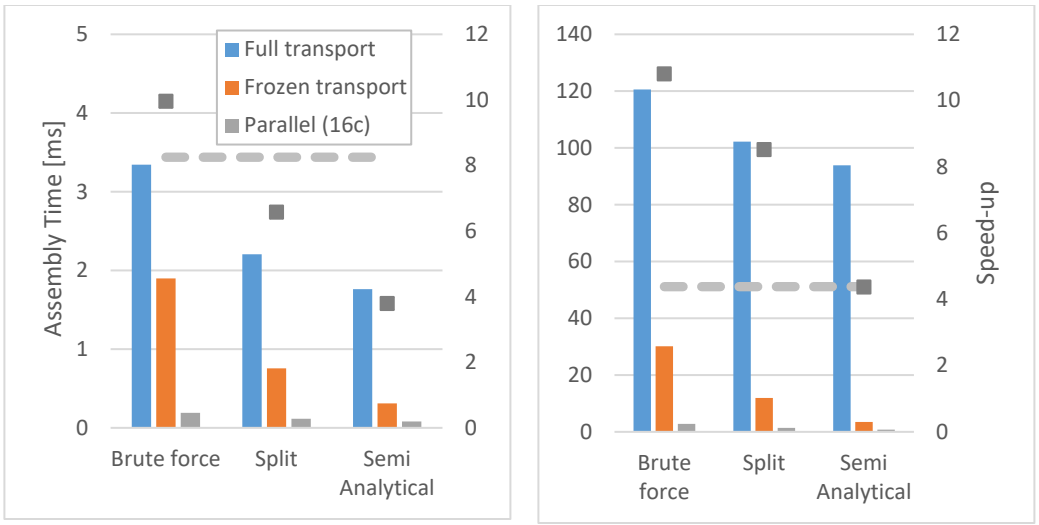
The new residual function was implemented, by design, to closely align with Cantera's implementation while still allowing parallelism. The modifications to support parallelism impose negligible costs as evident in the equal costs of the serial run-times. The parallel efficiencies, deduced from the parallel and serial run-times, were 76% and 89%, indicating the new method has efficient parallelism. Recalling that Cantera does not support parallelism, the parallel speed-up provided by the prototype flame model should translate to substantial performance speed-up of the overall solver. The residual function with the larger NG mechanism, which has approximately 4 times as many species (3.9) and reactions (4.3), costs 9.6 times that of the DZ mechanism. This scaling is expected as there is a combination of

linear scaling costs associated with the number of reactions and quadratic scaling associated with the number of species (which dominate the transport coefficients).

The single-core Jacobian assembly costs,[*] as measured on the Blueback system, are shown in Figure 7. For comparison, the Jacobian is assembled using a brute-force (BF) finite difference approach (i.e., calling the residual function 3N times as done in Cantera) as well as with functional splitting (FS) in which the chemistry and transport Jacobian terms are approximated with finite-difference separately to avoid redundant chemistry evaluations (i.e., 3N transport and N chemistry residual evaluations). Recalling that the chemistry terms only contribute to the diagonal matrix blocks, the BF approach wastefully evaluates the nonexistent influence of neighbor mesh points on the diagonal terms. The SA approach, as detailed earlier, evaluates the chemistry Jacobian terms semi-analytically (i.e., the temperature derivatives are still approximated with finite-difference) and uses the same finite-difference approach for the transport as the FS approach. The significant savings from neglecting the transport coefficient derivatives is also reported in the plots.

Cantera's assembly times are included in Figure 7 for reference. Cantera uses the BF algorithm with frozen transport approximation. However, Cantera stores (or caches) selective internal properties and attempts to reuse previous computations when possible. For example, if temperature is unchanged, previously computed reaction rates or thermodynamic properties that depend on temperature are reused on subsequent evaluations. This approach can reduce computational costs but at the expense of parallelism and performance predictability. For this reason, Cantera's approach theoretically should fall between the BF and FS methods; however, it is difficult to predict Cantera's performance in general since the specific cost depends on the instantaneous thermodynamic state of the flame model.

[*] This excludes the cost of matrix factorization.



(a) DZ (b) NG

Figure 7. Jacobian assembly cost for the (a) DZ and (b) NG kinetic models using the brute-force and functional splitting (split) finite-difference methods compared with the semi-analytical approach within the new flame model with and without the frozen transport approximation. Also showing the parallel run-time (and speed-up with symbols) using 16 cores with the frozen transport approximation. For reference, Cantera's assembly times are shown by the horizontal dashed line.

From the results in Figure 7, the benefit of the frozen transport approximation, in terms of computational cost, is clear with cost savings ranging from 1.8 to 26.7 times. More important for this study is the cost savings from the SA approach. Over the best-case finite-difference approach (i.e., FS), SA reduces the cost by 2.4 and 3.4 times with the smaller DZ and larger NG chemical kinetics models, respectively, with the added benefit of lessened approximation errors.

Compared with Cantera, SA reduces the assembly cost by 11.1 and 14.6 times with the two mechanisms. Cantera's assembly costs were surprisingly high considering the theoretical performance predictions. This may be due to inefficient matrix access pattern (cache efficiency) and superfluous bookkeeping in Cantera's object-oriented implementation.

As reported earlier, the residual function in the new flame model shows promising parallel speed-up. This can be used to reduce the matrix assembly cost with the finite-difference approximations. That is, each of the 3N residual functions evaluations are executed sequentially but each evaluation uses internal parallelism. This is not an optimal approach* and incurs significant parallel overhead costs due

* A more efficient approach is to call the 3N evaluations in parallel with each evaluation being done serially. This incurs only O(1) parallel region invocations and—for even modest-sized chemistry models—exposes ample parallelism to saturate modern high-performance computing systems.

to repeatedly spawning the parallel thread team. However, it can provide modest performance gains.

The parallel run-times using 16 cores with the frozen transport approximation are also shown in Figure 7. The associated speed-up is shown as well. As seen, all three assembly methods attain positive parallel speed-up (i.e., speed-up >1), but the benefits decrease as the assembly method efficiency increases due to the unavoidable parallel overhead (i.e., Amdahl's law). Nonetheless, the efficient SA method is still able to achieve a 40%–75% performance gain over the parallel FS approach with this less-than-ideal parallelism approach.

The experimental flame model was implemented to assess the impact of direct residual convergence on the solution and performance implications of 1) parallelism in the residual and Jacobian evaluations and 2) the Jacobian assembly strategy. The nonlinear solver, line search method, pseudo-time solver (if the prior two fail), and overall workflow have not been optimized. Nevertheless, the new solver was able to solve the $P = 23.8\,MPa$ flame (used in the comparison in Figure 5), starting from the same boundary conditions and initial guess on a coarse mesh as Cantera, within 10% of Cantera's run time (single core) with the larger NG mechanism. With 16 cores, the new solver reduced the run time by 37%. This low parallel efficiency is largely due to the substantial time spent solving the solution on the coarse mesh. Nearly 70% of the overall run time[*] was spent solving the initial 22-point (coarse) initial flame solution. Cantera, PREMIX, and the new flame solver use iterative mesh refinement to resolve steep gradients in the flame. Since the parallelism targets grid loops, poor parallel efficiency is obtained on small grids. Due to the poor initial guess, this phase of the solver relies heavily on the efficacy of the pseudo-time strategy, and Cantera's strategy has been optimized heavily within its convergence scheme. Optimization is needed in the new residual-based flame solver to improve the time-stepping strategy.

[*] It is speculated that this high initial cost can be reduced by tailoring the residual tolerance to be less stringent on a coarse initial mesh and progressively tightening as the mesh is refined toward the final solution.

4. Summary and Conclusions

Two novel studies were presented. First, Cantera's built-in burner stabilized premixed flame model (or solver) was demonstrated as a viable alternative to the legacy (Chemkin-III) PREMIX version when used within the CYCLOPS propellant burning rate prediction model. Second, a new burner flame model was implemented and successfully demonstrated to 1) more rigorously solve the nonlinear system and 2) support multicore parallel computing to reduce computational time. Unlike Chemkin-III and PREMIX, the open-source Cantera library is actively maintained and developed. There is a desire to transition to Cantera to take advantage of continuous improvements for chemical kinetics (e.g., pressure and mixture dependencies) and thermodynamic modeling (e.g., nonideal gases) from the broader combustion modeling community.

For the initial study, Cantera's burner flame model was demonstrated to give the same burning rate predictions, within the convergence tolerances, as the legacy PREMIX solver when used within CYCLOPS to predict the burning rate of neat NG using the dark-zone[4] and NG[5] chemical mechanisms. The predicted temperature profiles using PREMIX and Cantera, at the same operating conditions, were shown to also have negligible differences.

While PREMIX and Cantera follow the same solution strategy, the single-core (serial) run time with Cantera was 23%–81% faster than PREMIX due to a more efficient pseudo-time integration strategy and other minor cost-saving approaches. With parallelism, PREMIX was faster for this benchmark problem. But PREMIX's nonlinear solver is generally less efficient and robust than Cantera's implementation and, as such, is more susceptible to failure than Cantera. This does not imply that Cantera will not fail, only that it is less susceptible.

From this study, we see that Cantera is a suitable replacement and computational costs could be brought in line with or exceed that of PREMIX if parallelism was available. To address this, a new burner flame model using Cantera's internal thermodynamic, transport, and kinetics libraries was implemented in a manner to support multicore parallelism. Two key differences in the numerical method were implemented: 1) the solver rigorously drives the residual error to below the defined tolerance (within step-length limits) and 2) the Jacobian is assembled semi-analytically to reduce the computational cost and rounding errors associated with the standard one-sided finite-difference approach used in PREMIX and Cantera. Care was taken to implement the residual function as close to Cantera's as possible while still enabling parallelism to isolate influences from these two modifications.

In Cantera, residual error is not directly monitored or controlled, which can lead to non-monotonic behavior when coupled with CYCLOPS. The new flame solver's nonlinear Newton solver directly controls the residual error and was shown to provide consistent convergence across the flame with notable changes to the predicted solution near the peak heat release zones. Future studies are needed to assess the impact of this more rigorous convergence on burning rate prediction accuracy, robustness, and cost in the broader CYCLOPS workflow. Additionally, with a residual-based convergence for the flame equations, it is now theoretically possible to solve the coupled two-phase CYCLOPS model in a monolithic fashion (i.e., solve Equations 1–3 together and not as two segregated nonlinear systems).

Multicore parallelism was implemented with OpenMP and efficiencies of over 75% were measured with the residual function. The serial run-time costs were nearly equal to Cantera's, showing that the modifications to support parallelism had only a small—though nonzero—cost.

The SA method was shown to significantly reduce the computational cost of assembling the Jacobian matrix compared with finite-difference methods. Run-time costs savings of 3–4 times were observed over the best-case, finite-difference method without parallelism. Compared with Cantera's internal method, the SA approach reduced assembly costs by more than 10 times with 16 cores, with increased savings scaling with an increased number of chemical species. This latter effect is important for detailed propellant modeling as the number of species can be quite large.

The above performance analysis only considered the cost of individual computational kernels. An initial head-to-head comparison was reported in which the new solver achieved only a 37% improvement with parallelism. Closer analysis suggested that the low efficiency was due to the large portion of the run-time spent on the coarse initial mesh coupled with the inefficient pseudo-time strategy. The pseudo-time strategy is not completely decoupled from the nonlinear convergence or line search schemes, and Cantera's optimized approach cannot be used directly within this new approach. Future work is needed to design an efficient pseudo-time strategy for the residual-based convergence proposed here.

In addition to a new pseudo-time strategy, several other advances are needed in the new flame solver, based on results from this study, and will be reported in subsequent reports. These include 1) a more efficient and robust nonlinear solver, 2) a more efficient parallelism in the Jacobian assembly, 3) an SA Jacobian transport Jacobian (within the frozen transport approximation), and 4) scalable convergence criteria.

Regarding item (1), the simple modified Newton solver used here, which was sufficient for an initial study but should be replaced by an externally maintained framework, such as Kinsol[15] (or other open-source, community-supported frameworks), with optimized line search and backtracking algorithms.

For item (4), the residual equations solved by Cantera and the new solver are in full dimensional form. This results in a wide range of state vector scales between temperatures (300–4,000 K) and mass fractions (0–1). The residual error has a wide imbalance in scales, which impedes the nonlinear solver efficiency by increasing the condition number (stiffer) and makes the convergence tolerances dependent on the input conditions. (Recall that the residual tolerance was not achieved in Figure 5 since the change in the dimensioned temperature equation fell below machine zero.) Normalizing the state vector so that all values are roughly bounded between 0 and 1 is relatively straightforward (e.g., normalized temperature by T_{inlet}), lessens the step-length imbalance, and can improve the condition number modestly. However, it does not alter how the residuals scale with pressure, which is a critical aspect of burning rate predictions. A physics-based normalization strategy for the residual equations must account for the dependence on operating pressures to improve robustness.

5. References

1. Kee RJ, Coltrin ME, Glarborg P, Zhu H. Chemically reacting flow: theory and practice. 2nd ed. John Wiley & Sons; 2017.
2. McQuaid M, Stone C. CYCLOPS-3 burning rate prediction: PREMIX3_WPSB Version. 2024.
3. Stone C, Chen C-C, McBain A, Veals J, McQuaid M. On predicting gun propellant burning rates within a digital engineering framework for performance screening. JANNAF PEDCS; 2024.
4. Miller M, Anderson W. Burning-rate predictor for multi-ingredient propellants: nitrate-ester propellants. J Propuls Power. 2004;20(3):440–454.
5. Chen C-C, McQuaid MJ. A detailed, finite-rate chemical kinetic mechanism for modeling the thermal decomposition and combustion of gaseous nitroglycerin. 2022.
6. Stone C, Chen C-C. Addendum to ARL-TR-9536, a detailed, finite rate chemical kinetic mechanism for modeling the thermal decomposition and combustion of gaseous nitroglycerin. DEVCOM Army Research Laboratory (US); 2024 June. Report No.: ARL-TN-1207.
7. Stone C. Computational advances in modeling opposed-flow diffusion flames with detailed chemical. DEVCOM Army Research Laboratory (US); 2024 Apr. Report No.: ARL-TR-9908.
8. Kee R, Rupley F, Meeks E. CHEMKIN-III: a Fortran chemical kinetics package for the analysis of gas-phase chemical and plasma kinetics. Sandia National Laboratories (US); 1996. Report No.: SAND96-8216.
9. Kee R et al. CHEMKIN Collection, Release 3.6. Reaction Design, Inc.; 2000.
10. Goodwin D, Moffat H, Schoegle I, Speth R, Weber B. Cantera: an object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes. Cantera Developers; 2025. https://www.cantera.org
11. Singal PJ, Lee J, Lei L, Speth RL, Burke MP. Implementation of new mixture rules has a substantial impact on combustion predictions for H2 and NH3. Proc Combust Inst. 2024;40.
12. Singal P, Rambur A, Cornell R, Burke M. New mixture rules affect kinetic predictions of energetic material combustion: a case study for RDX, ammonium perchlorate, and nitroglycerin. Proc Combust Inst. 2026;42.

13. Stone C, Lansford J, McBain A, Chen C-C. Recent advances in prediction gun propellant formulation burning rates with physics-based flame models and chemical kinetics. JANNAF PEDCS; 2026; Pittsburgh, PA.

14. Lapointe S, Whitesides R, McNenly M. Sparse, iterative simulation methods for one-dimensional laminar flames. Combust Flame. 2019;204:23–32.

15. Zero-RK: zero order reaction kinetics. Lawrence Livermore National Laboratory (US); 2019. https://github.com/LLNL/zero-rk

16. Hindmarsh AC et al. SUNDIALS: suite of nonlinear and differential/algebraic equation solvers. ACM Trans Math Softw. 2005;31(3):363–396.

17. Zenin A. HMX and RDX: combustion mechanism and influence on modern double-base propellant combustion. J Propuls Power. 1995;11(4):752–765.

List of Symbols, Abbreviations, and Acronyms

1-D	One-Dimensional
AI	Artificial Intelligence
ARL	Army Research Laboratory
BF	Brute Force
CGS	Centimeter-Gram-Second
CPU	Central Processing Unit
DEVCOM	U.S. Army Combat Capabilities Development Command
DoD	Department of Defense
DoW	Department of War
DSRC	DoW Supercomputing Resource Center
DZ	Dark Zone
FS	Functional Splitting
GMRES	Generalized Minimal Residual
JFNK	Jacobian-Free Newton–Krylov
MKS	Meter-Kilogram-Second
NG	Nitroglycerin
NO	Nitrogen Monoxide
NPS	Nascent Product Set
ODE	Ordinary Differential Equation
SA	Semi-Analytical

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DEVCOM ARL FCDD RLB CB TECH LIB