

Code-Verification Techniques for Hypersonic Reacting Flows in Thermochemical Nonequilibrium

Brian A. Freno^a, Brian R. Carnes^a, V. Gregory Weirs^a

^a*Sandia National Laboratories, PO Box 5800, MS 0828, Albuquerque, NM 87185*

Abstract

The study of hypersonic flows and their underlying aerothermochemical reactions is particularly important in the design and analysis of vehicles exiting and reentering Earth's atmosphere. Computational physics codes can be employed to simulate these phenomena; however, code verification of these codes is necessary to certify their credibility. To date, few approaches have been presented for verifying codes that simulate hypersonic flows, especially flows reacting in thermochemical nonequilibrium. In this paper, we present our code-verification techniques for **verifying the spatial accuracy and thermochemical source term in hypersonic reacting flows in thermochemical nonequilibrium**. We demonstrate the effectiveness of these techniques on the Sandia Parallel Aerodynamics and Reentry Code (**SPARC**).

Keywords: code verification, hypersonic flow, thermochemical nonequilibrium, manufactured solutions

1. Introduction

Hypersonic flows are distinguished by flow velocities and stagnation enthalpies that are high enough to induce chemical reactions and excitation of thermal energy modes. In particular, the time scales of the reactions and thermal excitation are comparable to the characteristic flow time, requiring a fully coupled modeling approach. The study of hypersonic flows and their underlying aerothermochemical phenomena is particularly important in the design and analysis of vehicles exiting and reentering Earth's atmosphere [1, 2]. For much of this flight regime, the gas can be modeled as a continuum. The relevant chemical species are each tracked, such that the interactions within the flow field are accounted for in the evolution of the chemical state. Similarly, considering each internal energy mode allows the different modes to relax towards equilibrium while accounting for the flow and chemical states. As the flow velocity or stagnation enthalpy increases, more complex chemical kinetics and internal energy models must be included, and ionization, radiation, and other phenomena may become important.

Verification and validation of the implementation and suitability of such models are necessary to develop confidence in the credibility of the simulations. Validation assesses how well the models instantiated in the code represent the relevant physical phenomena. It is typically performed by comparing simulation results with experimental results to assess the suitability of the models, the model error, and the practical bounds of validity of the models. On the other hand, verification assesses the accuracy of the numerical solutions the code produces, relative to the assumptions and expectations associated with the numerical methods. Following Roache [3], Salari and Knupp [4], and Oberkampf and Roy [5], verification can be divided into code verification and solution verification. Solution verification focuses on numerical-error estimation for a particular simulation, whereas code verification focuses on the correctness of the numerical-method implementation in the code. A review of code and solution verification is presented by Roy [6].

This paper focuses on code verification. **When numerically solving the underlying equations of the aforementioned models**, the equations must be discretized using, for example, finite differences, finite volumes, or finite elements. Due to the finite nature of the discretization, the equations incur a truncation error and, consequently, their solutions introduce a discretization error. As the discretization is refined, the

Email addresses: bafreno@sandia.gov (Brian A. Freno), bcarnes@sandia.gov (Brian R. Carnes), vgweirs@sandia.gov (V. Gregory Weirs)

discretization error should decrease. More rigorously, a code should achieve an expected order of accuracy: as the mesh is refined by a factor, the error should decrease at a rate that is an expected power of that factor. In practice, since the exact solution is generally unavailable, manufactured solutions are frequently employed [7].

Code verification has been performed on computational physics codes associated with several physics disciplines, including fluid dynamics [8, 9, 10, 11], solid mechanics [12], fluid–structure interaction [13], heat transfer in fluid–solid interaction [14], multiphase flows [15], radiation hydrodynamics [16], electrodynamics [17], and electromagnetism [18]. Though not as common, code-verification techniques for hypersonic flows have been presented by Roy et al. [19] for a single-species perfect gas and by Gollan and Jacobs [20] for a multi-species gas in thermal equilibrium.

In this paper, we discuss the code-verification techniques we have employed for flows without discontinuities. The most noteworthy contribution of this work is our approach to verifying hypersonic reacting flows in thermochemical nonequilibrium. As the scope of this paper is limited to code verification, subsequent instances of ‘verification’ are used to abbreviate ‘code verification’.

To assess the spatial discretization, we employ manufactured and exact solutions on uniform and nonuniform meshes and study the convergence of the spatial error norms. Because we consider smooth flows, we employ the more rigorous L^∞ -norm, of which we demonstrate the greater effectiveness in detecting deviations in the expected accuracy. These tests identified lower-order boundary-condition implementations, which reduced the convergence rate and spatial accuracy before being corrected.

However, while these techniques are effective for assessing the spatial discretization, they do not directly reveal errors in the algebraic thermochemical source term. Therefore, to verify these terms, we compare with an independently developed code for thousands of samples, spanning extreme conditions, with the expectation that the values agree to machine precision. Additionally, we examine the convergence of the statistics with respect to the number of samples to assess their sufficiency. This work exposed disagreement due convergence criteria, as well as errors in model parameters, resulting in corrections to both.

This paper is organized as follows. Section 2 describes the governing conservation, energy-exchange, and chemical-kinetics equations. Section 3 details our approach for verifying the spatial discretization through the use of manufactured and exact solutions. Section 4 demonstrates the effectiveness of the verification techniques for the spatial discretization. Section 5 presents our approach for verifying the thermochemical source term. Section 6 demonstrates the effectiveness of the verification techniques for the thermochemical source term. Section 7 provides conclusions and an outlook for future work.

2. Governing Equations

In this paper, we consider hypersonic reacting flows in thermochemical nonequilibrium. We make the following approximations: (1) electronic energy is negligible, (2) vibrationally excited molecules can be characterized by a single vibrational temperature T_v , and (3) the translational and rotational temperatures are in thermal equilibrium: $T_t = T_r = T$.

2.1. Conservation Equations

With these approximations, we model the conservation of mass, momentum, and energy for a gas in thermochemical nonequilibrium [1, 2]:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}_c(\mathbf{U}) = -\nabla \cdot \mathbf{F}_p(\mathbf{U}) + \nabla \cdot \mathbf{F}_d(\mathbf{U}) + \mathbf{S}(\mathbf{U}), \quad (1)$$

where the conservative-variable state vector \mathbf{U} , convective flux \mathbf{F}_c , pressure flux \mathbf{F}_p , diffusive flux \mathbf{F}_d , and thermochemical source term \mathbf{S} are

$$\mathbf{U} = \begin{Bmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \\ \rho e_v \end{Bmatrix}, \quad \mathbf{F}_c(\mathbf{U}) = \begin{Bmatrix} \rho \mathbf{v}^T \\ \rho \mathbf{v} \mathbf{v}^T \\ \rho E \mathbf{v}^T \\ \rho e_v \mathbf{v}^T \end{Bmatrix}, \quad \mathbf{F}_p(\mathbf{U}) = \begin{Bmatrix} \mathbf{0} \\ p \mathbf{I} \\ p \mathbf{v}^T \\ \mathbf{0}^T \end{Bmatrix}, \quad \mathbf{F}_d(\mathbf{U}) = \begin{Bmatrix} -\mathbf{J} \\ \tau \\ (\tau \mathbf{v} - \mathbf{q} - \mathbf{q}_v - \mathbf{J}^T \mathbf{h})^T \\ (-\mathbf{q}_v - \mathbf{J}^T \mathbf{e}_v)^T \end{Bmatrix},$$

$$\mathbf{S}(\mathbf{U}) = \begin{Bmatrix} \dot{\mathbf{w}} \\ \mathbf{0} \\ 0 \\ Q_{t-v} + \mathbf{e}_v^T \dot{\mathbf{w}} \end{Bmatrix}.$$

$\rho = \{\rho_1, \dots, \rho_{n_s}\}^T$ is the vector of the densities of the n_s species, and $\rho = \sum_{s=1}^{n_s} \rho_s$ is the mixture density. For the particular case of air with $n_s = 5$, which is considered later in this paper, the species are N₂, O₂, NO, N, and O. $\mathbf{v} = \{u, v, w\}^T$ is the velocity vector, $p = \sum_{s=1}^{n_s} \frac{\rho_s}{M_s} \bar{R}T$ is the pressure, M_s is the molecular weight of species s , $\bar{R} = 8314.47 \text{ J/kmol/K}$ is the universal gas constant, and T is the translational-rotational temperature. $\mathbf{J} = [\rho_1 \mathbf{u}_1, \dots, \rho_{n_s} \mathbf{u}_{n_s}]^T$ is the diffusion flux of the species, and \mathbf{u}_s is the diffusion velocity of species s . $\boldsymbol{\tau}$ is the viscous stress tensor, \mathbf{q} is the heat flux, \mathbf{q}_v is the vibrational heat flux, and $\mathbf{h} = \{h_1, \dots, h_{n_s}\}^T$ is the vector of enthalpies per mass of the species. $\dot{\mathbf{w}} = \{\dot{w}_1, \dots, \dot{w}_{n_s}\}^T$ is the vector of mass production rates of the species per volume, and Q_{t-v} is the translational-vibrational energy exchange..

At high temperatures, vibrational energy is internally excited within molecules, which can cause dissociation or ionization within the gas [2]. The mixture vibrational energy per mass is

$$e_v = \sum_{s=1}^{n_s} \frac{\rho_s}{\rho} e_{v_s},$$

where $\mathbf{e}_v = \{e_{v_1}, \dots, e_{v_{n_s}}\}^T$ is the vector of vibrational energies per mass of the species, such that

$$e_{v_s} = \begin{cases} \sum_{m=1}^{n_{v_s}} e_{v_{s,m}}(T_v) & \text{for molecules,} \\ 0 & \text{for atoms.} \end{cases}$$

The vibrational energy per mass of mode m of species s is assumed to have a Boltzmann distribution over the vibrational energy levels [1, 2]:

$$e_{v_{s,m}}(T') = \frac{\bar{R}}{M_s} \frac{\theta_{v_{s,m}}}{\exp(\theta_{v_{s,m}}/T') - 1},$$

where n_{v_s} is the number of vibrational modes of species s ($n_{v_s} = 0$ for atoms), and $\theta_{v_{s,m}}$ is the characteristic vibrational temperature of mode m of species s . $\theta_{v_{s,m}}$ is typically obtained from experiments, and we retrieve the value from a lookup table, such as Table A2.

The total energy per mixture mass is

$$E = \frac{|\mathbf{v}|^2}{2} + \sum_{s=1}^{n_s} \frac{\rho_s}{\rho} (c_{v_s} T + e_{v_s} + h_s^o),$$

where c_{v_s} is the specific heat at constant volume of species s , which is the sum of the translational and rotational contributions. For diatomic molecules, $c_{v_s} = \frac{5}{2} \frac{\bar{R}}{M_s}$ and, for atoms, $c_{v_s} = \frac{3}{2} \frac{\bar{R}}{M_s}$. h_s^o is the formation enthalpy of species s .

2.2. Diffusive-Term Modeling

We use Fick's first law to model the diffusion flux of the species : $\mathbf{J} = -\rho \mathbf{D} \nabla \frac{\rho}{\rho}$, where $\mathbf{D} = \text{diag}\{\mathbf{D}_1, \dots, \mathbf{D}_{n_s}\}$ contains the effective diffusion coefficients of the species.

Assuming the flow is Newtonian and satisfies Stokes' hypothesis, we model the viscous stress tensor as $\boldsymbol{\tau} = \mu (\nabla \mathbf{v} + (\nabla \mathbf{v})^T) - \frac{2}{3} \mu (\nabla \cdot \mathbf{v}) \mathbf{I}$, where μ is the dynamic viscosity.

To model the heat fluxes, we employ Fourier's law: $\mathbf{q} = -\kappa \nabla T$ and $\mathbf{q}_v = -\kappa_v \nabla T_v$, where κ and κ_v are, respectively, the translational-rotational and vibrational heat conductivities of the mixture.

μ and κ for the mixture are computed using Wilke's mixing rule [21].

2.3. Translational–Vibrational Energy Exchange Modeling

The translational–vibrational energy exchange is computed using the Landau–Teller model [22]:

$$Q_{t-v} = \sum_{s=1}^{n_s} \rho_s \sum_{m=1}^{n_{v_s}} \frac{e_{v_{s,m}}(T) - e_{v_{s,m}}(T_v)}{\langle \tau_{s,m} \rangle}, \quad (2)$$

where $\langle \tau_{s,m} \rangle$ is the translational–vibrational energy relaxation time for mode m of species s .

To compute the relaxation time, we use the semi-empirical approach of Millikan and White [23], which is extended to account for higher temperatures [24, 25]:

$$\langle \tau_{s,m} \rangle = \left(\sum_{s'=1}^{n_s} \frac{y_{s'}}{\tau_{s,m,s'}} \right)^{-1} + \left[\left(N_A \sum_{s'=1}^{n_s} \frac{\rho_{s'}}{M_{s'}} \right) \sigma_{v_s} \sqrt{\frac{8 \bar{R} T}{\pi M_s}} \right]^{-1},$$

where the mole fraction of species s is

$$y_s = \frac{\rho_s / M_s}{\sum_{s'=1}^{n_s} \rho_{s'} / M_{s'}},$$

and the translational–vibrational energy relaxation time for mode m of species s when colliding with species s' is modeled by

$$\tau_{s,m,s'} = \frac{\exp [a_{s,m,s'} (T^{-1/3} - b_{s,m,s'}) - 18.42]}{p'}. \quad (3)$$

p' is the pressure expressed in atmospheres. For many gases, $a_{s,m,s'}$ and $b_{s,m,s'}$ can be modeled by

$$\begin{aligned} a_{s,m,s'} &= 1.16 \times 10^{-3} \mu_{s,s'}^{1/2} \theta_{v_{s,m}}^{4/3}, \\ b_{s,m,s'} &= 0.015 \mu_{s,s'}^{1/4}, \end{aligned} \quad (4)$$

where $\mu_{s,s'} = \frac{M_s M_{s'}}{M_s + M_{s'}}$ is the reduced mass of species s and s' . Additionally, the collision-limiting vibrational cross section is modeled by

$$\sigma_{v_s} = \sigma'_{v_s} \left(\frac{50,000 \text{ K}}{T} \right)^2,$$

where $N_A = 6.022140857 \times 10^{26} \text{ kmol}^{-1}$ is the Avogadro constant, and σ'_{v_s} is the collision-limiting vibrational cross section at 50,000 K.

2.4. Chemical-Kinetics Modeling

The mass production rate per volume for species s is modeled by

$$\dot{w}_s = M_s \sum_{r=1}^{n_r} (\beta_{s,r} - \alpha_{s,r}) (R_{f_r} - R_{b_r}), \quad (5)$$

where $\alpha_{s,r}$ and $\beta_{s,r}$ are, respectively, the reactant and product stoichiometric coefficients for species s in reaction r . R_{f_r} and R_{b_r} are the forward and backward reaction rates for reaction r .

The reaction rates are defined by

$$R_{f_r} = \gamma k_{f_r} \prod_{s=1}^{n_s} \left(\frac{1}{\gamma} \frac{\rho_s}{M_s} \right)^{\alpha_{s,r}}, \quad (6)$$

$$R_{b_r} = \gamma k_{b_r} \prod_{s=1}^{n_s} \left(\frac{1}{\gamma} \frac{\rho_s}{M_s} \right)^{\beta_{s,r}}, \quad (7)$$

such that, in (6) and (7), k_{f_r} , k_{b_r} , and ρ_s/M_s are expressed using the centimeter–gram–second system of units (CGS), R_{f_r} and R_{b_r} are expressed using the meter–kilogram–second system of units, and $\gamma = 1000 \frac{\text{cm}^3 \cdot \text{kmol}}{\text{m}^3 \cdot \text{mol}}$ is the conversion factor. The forward and backward reaction-rate coefficients k_{f_r} and k_{b_r} are modeled using the approach of Park [26]:

$$k_{f_r}(T_c) = C_{f_r} T_c^{\eta_r} \exp(-\theta_r/T_c),$$

$$k_{b_r}(T) = \frac{k_{f_r}(T)}{K_{e_r}(T)},$$

where C_{f_r} and η_r are empirical parameters, and θ_r is the activation energy of reaction r , divided by the Boltzmann constant. T_c is the rate-controlling temperature; it is set to $T_c = \sqrt{T T_v}$ for dissociative reactions and $T_c = T$ for exchange reactions. K_{e_r} is the equilibrium constant for reaction r , modeled by

$$K_{e_r}(T) = \exp \left[A_{1_r} \left(\frac{T}{10000} \right) + A_{2_r} + A_{3_r} \ln \left(\frac{10000}{T} \right) + A_{4_r} \frac{10000}{T} + A_{5_r} \left(\frac{10000}{T} \right)^2 \right], \quad (8)$$

where A_{i_r} are empirical curve-fit coefficients.

In this paper, $K_{e_r}(T)$ is limited to $[\exp(-81), \exp(81)]$, and, when computing $\dot{\mathbf{w}}$, T and T_v are increased to 500 K if they are less than 500 K.

2.5. Gas Modeling

We limit the scope of this paper to the five-species air model, which consists of N_2 , O_2 , NO , N , and O . These species can undergo the dissociation and exchange reactions listed in Table 1. Additional properties of the species and their reactions are provided in Appendix A.

r	Reaction	Type of Reaction
1–5	$\text{N}_2 + \mathcal{M} \rightleftharpoons \text{N} + \text{N} + \mathcal{M}, \quad \mathcal{M} = \{\text{N}_2, \text{O}_2, \text{NO}, \text{N}, \text{O}\}$	Dissociation
6–10	$\text{O}_2 + \mathcal{M} \rightleftharpoons \text{O} + \text{O} + \mathcal{M}, \quad \mathcal{M} = \{\text{N}_2, \text{O}_2, \text{NO}, \text{N}, \text{O}\}$	Dissociation
11–15	$\text{NO} + \mathcal{M} \rightleftharpoons \text{N} + \text{O} + \mathcal{M}, \quad \mathcal{M} = \{\text{N}_2, \text{O}_2, \text{NO}, \text{N}, \text{O}\}$	Dissociation
16	$\text{N}_2 + \text{O} \rightleftharpoons \text{N} + \text{NO}$	Exchange
17	$\text{NO} + \text{O} \rightleftharpoons \text{N} + \text{O}_2$	Exchange

Table 1: Five-species air model: Reactions.

We note that the five-species air model is just one of a few options. Gimelshein et al. [27, 28] provide a comparison of air models represented by different numbers of species. Nonetheless, the verification techniques we propose for the thermochemical source term are applicable to any number of species.

3. Verification Techniques for Spatial Discretization

We begin our approach to code verification by computing the spatial accuracy of the numerical discretization. To compute the spatial accuracy, we compare the solution to the discretized equations with the solution to the continuous equations. For each of the flow variables, we compute error norms and compare the rates at which the error norms decrease with respect to the rates at which the mesh size increases.

3.1. Spatial Accuracy

In a steady state, the governing system of partial differential equations (1) can be written generally as

$$\mathbf{r}(\mathbf{U}) = \mathbf{0}, \quad (9)$$

where \mathbf{r} is the residual vector, and $\mathbf{U} = \mathbf{U}(\mathbf{x})$ is the state vector. To solve (9) numerically, it must be discretized:

$$\mathbf{r}_h(\mathbf{U}_h) = \mathbf{0}, \quad (10)$$

where \mathbf{r}_h is the residual of the discretized system of equations, \mathbf{U}_h is the solution to the discretized equations, and h describes the size of the mesh employed by the discretization. To simplify the notation in this section, we assume appropriate mappings of the residuals and solutions onto continuous or discrete space.

The truncation error is

$$\boldsymbol{\tau}_h(\mathbf{V}) = \mathbf{r}_h(\mathbf{V}) - \mathbf{r}(\mathbf{V}). \quad (11)$$

Letting $\mathbf{V} = \mathbf{U}_h$ and adding (9), (11) becomes

$$\boldsymbol{\tau}_h(\mathbf{U}_h) = \mathbf{r}_h(\mathbf{U}_h) - \mathbf{r}(\mathbf{U}_h) + \mathbf{r}(\mathbf{U}) = \mathbf{r}(\mathbf{U}) - \mathbf{r}(\mathbf{U}_h).$$

When \mathbf{r} is linear or linearized with respect to \mathbf{U} , the discretization error $\mathbf{e}_h = \mathbf{U}_h - \mathbf{U}$ is related to the truncation error by $\mathbf{r}(\mathbf{e}_h) = -\boldsymbol{\tau}_h(\mathbf{U}_h)$ [29, 5].

For a p^{th} -order-accurate discretization, the truncation error is

$$\boldsymbol{\tau}_h = \mathbf{C}_r h^p + \mathcal{O}(h^{p+1}),$$

where \mathbf{C}_r is a function of derivatives of the state vector but is independent of h . Once the meshes are fine enough that the approximation is within the asymptotic region, $h^{p+1} \ll h^p$, then $\boldsymbol{\tau}_h \approx \mathbf{C}_r h^p$ and $\mathbf{e}_h \approx \mathbf{C}_U h^p$, where \mathbf{C}_U is also independent of h .

The observed accuracy can be computed using two meshes. For example, let the coarser mesh be characterized by h . The second mesh, if q -times as fine in each dimension as the first, is characterized by h/q . For each scalar field α in \mathbf{U} (e.g., $\alpha = \{\rho_1, \dots, \rho_{n_s}, u, v, w, T, T_v\}$), each mesh has a discretization error: $e_h^\alpha \approx C_\alpha h^p$ and $e_{h/q}^\alpha \approx C_\alpha (h/q)^p$. p can be approximated locally by

$$p \approx \frac{\log |e_h^\alpha / e_{h/q}^\alpha|}{\log q} = \log_q |e_h^\alpha / e_{h/q}^\alpha|. \quad (12)$$

3.2. Solutions

Computing the spatial accuracy p in (12) requires computing errors \mathbf{e}_h , which, in turn, require solutions \mathbf{U} . Therefore, we employ exact and manufactured solutions.

3.2.1. Exact Solutions

For limited cases, there exist exact solutions $\mathbf{U}_{\text{Exact}}$ to (9), which can be directly compared with the computed solutions \mathbf{U}_h from (10) with negligible implementation effort. However, available exact solutions only span a small subset of the application space, so we require additional approaches to thoroughly test the capabilities of the code.

3.2.2. Manufactured Solutions

Manufactured solutions \mathbf{U}_{MS} enable us to develop solutions that exercise the features we intend to test. Unless the manufactured solutions are exact solutions, they will not satisfy (9): $\mathbf{r}(\mathbf{U}_{\text{MS}}) \neq \mathbf{0}$. Therefore, a forcing term is added to (10) to account for the presence of the manufactured solutions:

$$\mathbf{r}_h(\mathbf{U}_h) = \mathbf{r}(\mathbf{U}_{\text{MS}}). \quad (13)$$

$\mathbf{r}(\mathbf{U}_{\text{MS}})$ in (13) is computed analytically since \mathbf{r} and \mathbf{U}_{MS} are known.

Because the equations are differential and the error is a function of derivatives of the state vector, the manufactured solutions should be smooth, continuously differentiable functions with generally nonzero derivatives. Additionally, for the approximation to be in the asymptotic region without requiring especially fine meshes, variations over the domain should not be large.

3.3. Error Norms

The spatial accuracy p can be computed at a single location in the domain, as is done in (12); however, this approach has two shortcomings: (1) for cell-centered schemes, the cell centroids of a coarser mesh only coincide with those arising after mesh refinement in very limited cases, and (2) in regions where the errors vanish, the computed spatial accuracy is meaningless. Therefore, for each scalar field α , we use error norms ε_α to quantify the spatial accuracy p :

$$p = \log_q \left(\varepsilon_{\alpha_h} / \varepsilon_{\alpha_{h/q}} \right). \quad (14)$$

Because the error norms are global quantities, they do not require coincident solution locations across meshes, and they will only be zero for trivial flows.

We consider two norms in particular:

1. L^1 -norm:

$$\varepsilon_\alpha^1 = \|\alpha_h(\mathbf{x}) - \alpha(\mathbf{x})\|_1 = \int_{\Omega} |\alpha_h(\mathbf{x}) - \alpha(\mathbf{x})| d\Omega. \quad (15)$$

The L^1 -norm enables us to compute the spatial accuracy based on the average error throughout the domain without significant contamination from localized deviations. Such localized deviations can arise from discontinuities, such as shocks, as well as from boundary conditions discretized with lower-order spatial accuracy than what is used for the domain interior.

2. L^∞ -norm:

$$\varepsilon_\alpha^\infty = \|\alpha_h(\mathbf{x}) - \alpha(\mathbf{x})\|_\infty = \max_{\mathbf{x} \in \Omega} |\alpha_h(\mathbf{x}) - \alpha(\mathbf{x})|. \quad (16)$$

Unlike the L^1 -norm, the L^∞ -norm catches the aforementioned localized deviations by computing the maximum error throughout the domain. This is particularly useful when such deviations are unexpected.

For the smooth flows considered in this paper, the observed orders of accuracy computed by the L^1 -norm and L^∞ -norm are expected to be the same.

4. Spatial-Discretization Verification Results

We demonstrate the aforementioned and forthcoming code-verification techniques on the Sandia Parallel Aerodynamics and Reentry Code (SPARC) [30, 31, 32, 33] presently being developed at Sandia National Laboratories. SPARC is a compressible computational fluid dynamics code designed to model transonic and hypersonic reacting turbulent flows.

SPARC employs a cell-centered finite-volume discretization, and the simulations presented herein use the Steger–Warming flux-vector splitting scheme. The expectation is that SPARC is second-order accurate ($p = 2$) in space for flows without discontinuities. Therefore, because the midpoint rule has a discretization error of $\mathcal{O}(h^2)$, we integrate $\mathbf{r}(\mathbf{U}_{\text{MS}})$ in (13) using the midpoint rule, without decreasing the order of accuracy. We additionally use approximations to compute the L^1 -norm in (15) and the L^∞ -norm in (16):

$$\varepsilon_\alpha^1 \approx \sum_{i=1}^n (\Omega_i |\alpha_h(\mathbf{x}_i) - \alpha(\mathbf{x}_i)|), \quad (17)$$

$$\varepsilon_\alpha^\infty \approx \max_{1 \leq i \leq n} |\alpha_h(\mathbf{x}_i) - \alpha(\mathbf{x}_i)|, \quad (18)$$

where \mathbf{x}_i and Ω_i are, respectively, the centroid and volume of cell i , and n is the number of cells. For a uniform mesh, (17) is the same as the discrete L^1 -norm, when multiplied by the domain volume Ω . Equation (18) is the discrete L^∞ -norm.

SPARC computes $\mathbf{r}(\mathbf{U}_{\text{MS}})$ in (13) through the automatic differentiation tool Sacado [34, 35]. This avoids the need for using an external package to compute $\mathbf{r}(\mathbf{U}_{\text{MS}})$ through symbolic manipulation. A major drawback to using symbolic manipulation is the large amount of output generated, which must be copied and formatted

into a source-code file. For the work in this paper, a single term for one equation would have required several lines of code. On the other hand, the automatic-differentiation approach we employ only requires coding the terms in the differential equation. By templating the exact solution, we can generate the equivalent source term using only a few lines of code. The source term is never provided in symbolic form, but it can be efficiently evaluated. The Sacado implementation in **SPARC** has been unit tested by comparing with examples computed using symbolic manipulation. The results of those tests agree within machine precision.

For these results, the goal is to ensure second-order accuracy is achievable. Therefore, limiters are disabled and boundary conditions are expected to be second-order accurate. For complex simulations, the accuracy may be deliberately reduced in favor of numerical stability. The implications of which can be assessed through solution verification, which is not the focus of this paper.

Our verification of **SPARC** begins with the simplest tests, and, as these tests are satisfied, we add complexity to the subsequent tests. We begin by verifying single-species supersonic inviscid flow, ultimately increasing the complexity to multi-species hypersonic inviscid flow in thermochemical nonequilibrium.

For many of our two-dimensional manufactured solutions, we use the following solution structures, or subsets thereof:

$$\begin{aligned}
\rho_{N_2}(x, y) &= \bar{\rho}_{N_2} [1 - \epsilon \sin(\frac{5}{4}\pi x) (\sin(\pi y) + \cos(\pi y))], \\
\rho_{O_2}(x, y) &= \bar{\rho}_{O_2} [1 + \epsilon \sin(\frac{3}{4}\pi x) (\sin(\pi y) + \cos(\pi y))], \\
\rho_{NO}(x, y) &= \bar{\rho}_{NO} [1 + \epsilon \sin(\pi x) (\sin(\pi y) + \cos(\pi y))], \\
\rho_N(x, y) &= \bar{\rho}_N [1 + \epsilon \sin(\pi x) (\cos(\frac{1}{4}\pi y) + \sin(\frac{1}{4}\pi y))], \\
\rho_O(x, y) &= \bar{\rho}_O [1 + \epsilon \sin(\pi x) (\sin(\pi y) + \cos(\frac{1}{4}\pi y))], \\
u(x, y) &= \bar{u} [1 + \epsilon \sin(\frac{1}{4}\pi x) (\sin(\pi y) + \cos(\pi y))], \\
v(x, y) &= \bar{v} [-\epsilon \sin(\frac{5}{4}\pi x) (\sin(\pi y) + \cos(\pi y))], \\
T(x, y) &= \bar{T} [1 + \epsilon \sin(\frac{5}{4}\pi x) (\sin(\pi y) + \cos(\pi y))], \\
T_v(x, y) &= \bar{T}_v [1 + \epsilon \sin(\frac{3}{4}\pi x) (\sin(\frac{5}{4}\pi y) + \cos(\frac{3}{4}\pi y))]. \tag{19}
\end{aligned}$$

These solutions are shown in Figure 1 for $(x, y) \in [0, 1] \text{ m} \times [0, 1] \text{ m}$. The inflow and outflow boundaries are located at $x = 0 \text{ m}$ and $x = 1 \text{ m}$. The velocity field ensures the flow is tangential to the slip-wall (tangent-flow) boundaries located at $y = 0 \text{ m}$ and $y = 1 \text{ m}$.

4.1. Single-Species Inviscid Flow in Thermochemical Equilibrium

The first set of tests consists of a single species ($n_s = 1$, $\rho = \rho$) inviscid ($\mathbf{F}_d = 0$) flow in thermochemical equilibrium ($\mathbf{S} = \mathbf{0}$, $T = T_v$). For this type of flow, the reference velocity \bar{u} is determined from a reference Mach number \bar{M} :

$$\bar{u} = \bar{M} \sqrt{\gamma R \bar{T}}, \tag{20}$$

where γ is the ratio of specific heats for air, and R is the specific gas constant of air.

4.1.1. 1D Supersonic Flow using a Manufactured Solution

For this test, we simulate a simple, one-dimensional flow using manufactured solutions:

$$\begin{aligned}
\rho(x) &= \bar{\rho} [1 - \epsilon \sin(\pi x)], \\
u(x) &= \bar{u} [1 - \epsilon \sin(\pi x)], \\
T(x) &= \bar{T} [1 + \epsilon \sin(\pi x)],
\end{aligned}$$

with $x \in [0, 1] \text{ m}$, $\bar{\rho} = 1 \text{ kg/m}^3$, $\bar{M} = 2.5$, $\bar{T} = 300 \text{ K}$, and $\epsilon = 0.05$.

Upon solving (13), the observed order of accuracy p is computed from (14) for $\alpha = \{\rho, u, T\}$. Five 1D meshes are used, consisting of 50, 100, 200, 400, and 800 elements.

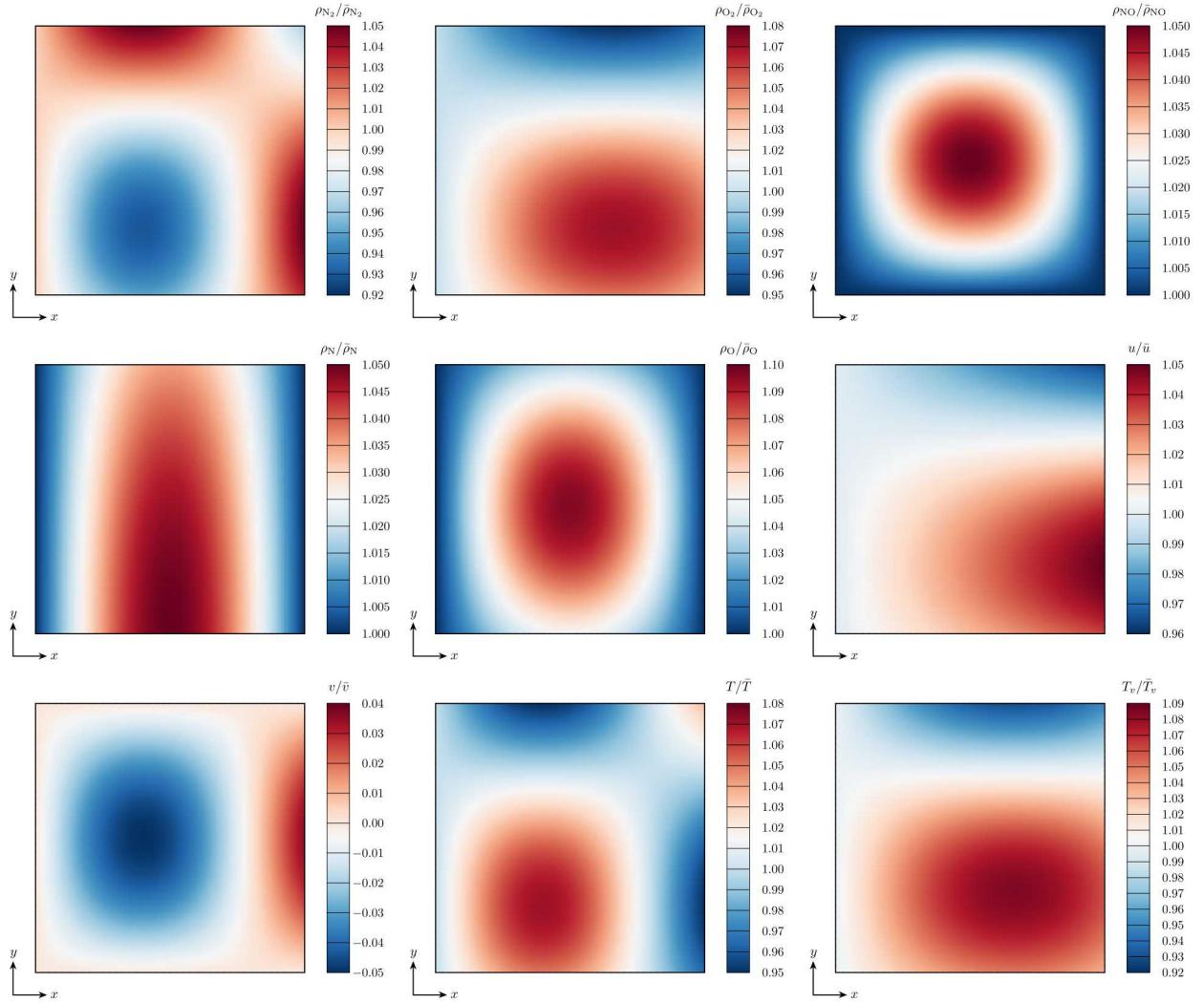


Figure 1: 2D manufactured solutions from Equation (19) with $\epsilon = 0.05$.

Table 2 shows the observed accuracy from the original state of the code using the L^∞ -norm and L^1 -norm of the error. The L^1 -norm indicates second-order accuracy ($p = 2$), whereas the L^∞ -norm indicates first-order accuracy ($p = 1$). This example demonstrates the usefulness of the L^∞ -norm. The L^1 -norm suggests the code, on average, is second-order accurate; however, the L^∞ -norm captures localized deviations in the accuracy. These deviations are due to the supersonic-inflow and supersonic-outflow boundary-condition implementations being only first-order accurate. For this case, the accuracy reduction is limited to the vicinity of the boundaries.

Mesh	L^∞ -norm			L^1 -norm		
	ρ	u	T	ρ	u	T
1–2	1.0008	1.0008	1.0008	1.9955	1.9956	1.9956
2–3	1.0002	1.0002	1.0002	1.9980	1.9981	1.9981
3–4	1.0001	1.0001	1.0000	1.9994	1.9994	1.9995
4–5	1.0000	1.0000	1.0000	1.9998	1.9998	1.9998

Table 2: 1D MMS, $n_s = 1$, $T_v = T$, $\dot{\mathbf{w}} = \mathbf{0}$: Observed accuracy p from original boundary conditions.

Mesh	L^∞ -norm			L^1 -norm		
	ρ	u	T	ρ	u	T
1–2	2.0313	2.0362	2.0351	2.0489	2.0526	2.0521
2–3	2.0157	2.0184	2.0178	2.0252	2.0271	2.0268
3–4	2.0079	2.0093	2.0090	2.0128	2.0138	2.0136
4–5	2.0040	2.0047	2.0045	2.0064	2.0070	2.0069

Table 3: 1D MMS, $n_s = 1$, $T_v = T$, $\dot{\mathbf{w}} = \mathbf{0}$: Observed accuracy p from corrected boundary conditions.

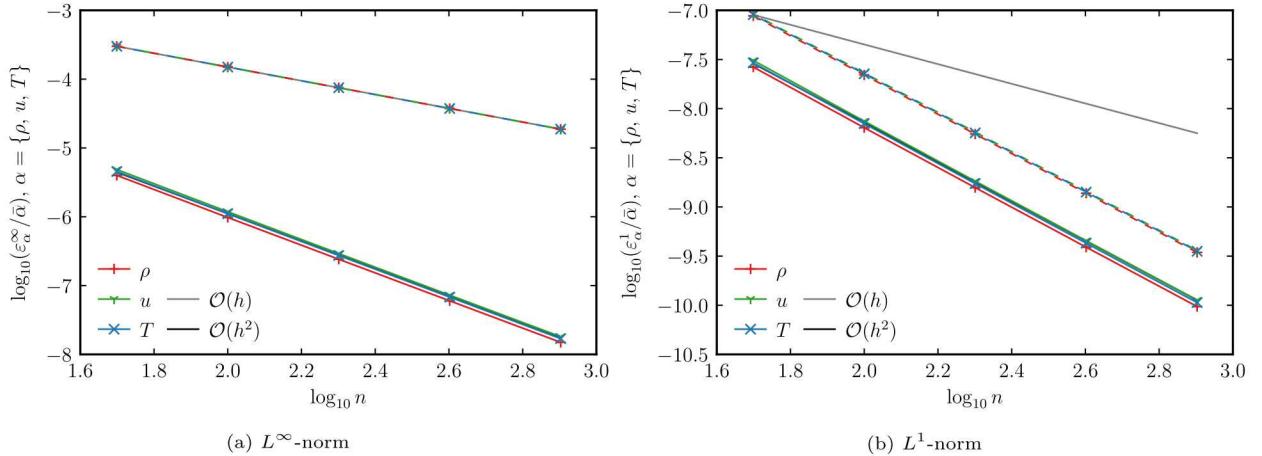


Figure 2: 1D MMS, $n_s = 1$, $T_v = T$, $\dot{\mathbf{w}} = \mathbf{0}$: Norms of the error. Dashed lines denote original boundary conditions; solid lines denote corrected boundary conditions.

We corrected the two boundary-condition implementations to be second-order accurate, which is confirmed in Table 3. Additionally, Figures 2a and 2b show the two error norms for each of the flow variables, before and after correcting the boundary conditions. As shown in Figure 2a, the maximum errors are reduced by orders of magnitude upon correcting the boundary conditions. Furthermore, Figure 2b shows that the correct boundary conditions reduce the average error by a factor of approximately three.

4.1.2. 2D Supersonic Flow using a Manufactured Solution

This test increases the complexity of the first test by including variations along a second dimension. In addition to the supersonic-inflow and supersonic-outflow boundary conditions of the first test, the slip-wall boundary condition is exercised.

The manufactured solutions for this case for $\alpha = \{\rho, u, v, T\}$ are listed in (19) and shown in Figure 1. In (19) and in Figure 1, $\rho = \rho_{N_2}$, $\bar{\rho} = \bar{\rho}_{N_2}$, and $\bar{v} = \bar{u}$ (20). The domain is a square with $(x, y) \in [0, 1] \text{ m} \times [0, 1] \text{ m}$, and $\bar{\rho} = 1 \text{ kg/m}^3$, $\bar{M} = 2.5$, $\bar{T} = 300 \text{ K}$, and $\epsilon = 0.05$.

Upon solving (13), the observed order of accuracy p is computed from (14). Five 2D meshes are used, consisting of 25×25 , 50×50 , 100×100 , 200×200 , and 400×400 elements. These meshes are chosen to test the spatial accuracy of the discretization for nonuniform meshes, and are created using the approach in Appendix B. The 50×50 mesh is shown in Figure 3.

Table 4 shows the observed accuracy using the L^∞ -norm and L^1 -norm of the error. Both norms indicate first-order accuracy ($p = 1$), despite the second-order-accuracy expectation. This inconsistency is due to the supersonic-inflow, supersonic-outflow, and slip-wall boundary-condition implementations being only first-order accurate. Unlike the first case, the implications of the first-order-accurate boundary conditions are global for this case.

The corrected boundary-condition implementations are confirmed to be second-order accurate ($p = 2$) in Table 5. Figures 4a and 4b show the two error norms for each of the flow variables, before and after correcting the boundary conditions. The correct boundary conditions reduce both the maximum and average error by

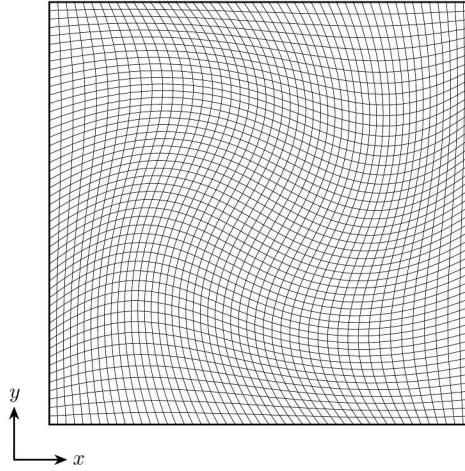


Figure 3: 2D MMS, $n_s = 1$, $T_v = T$, $\dot{\mathbf{w}} = \mathbf{0}$: Mesh with 50×50 elements.

Mesh	L^∞ -norm				L^1 -norm			
	ρ	u	v	T	ρ	u	v	T
1–2	0.9420	0.9409	0.9721	0.9628	1.0083	0.9838	0.9255	0.9861
2–3	0.9850	0.9902	0.9910	0.9874	1.0082	0.9980	0.9686	0.9957
3–4	0.9960	1.0002	0.9924	0.9952	1.0043	1.0008	0.9871	0.9985
4–5	0.9989	1.0009	0.9959	0.9984	1.0022	1.0008	0.9943	0.9995

Table 4: 2D MMS, $n_s = 1$, $T_v = T$, $\dot{\mathbf{w}} = \mathbf{0}$: Observed accuracy p from original boundary conditions.

Mesh	L^∞ -norm				L^1 -norm			
	ρ	u	v	T	ρ	u	v	T
1–2	2.0623	1.9188	1.8174	1.8598	2.2440	2.1789	2.1000	2.1802
2–3	2.1304	1.9450	1.9221	1.9280	2.1701	2.1248	2.0745	2.1038
3–4	2.0902	1.9603	1.9671	1.9586	2.0788	2.0577	2.0436	2.0461
4–5	2.0128	1.9823	1.9860	1.9809	2.0303	2.0246	2.0230	2.0220

Table 5: 2D MMS, $n_s = 1$, $T_v = T$, $\dot{\mathbf{w}} = \mathbf{0}$: Observed accuracy p from corrected boundary conditions.

orders of magnitude. For the subsequent results, we omit the L^1 -norm and consider only the corrected boundary conditions.

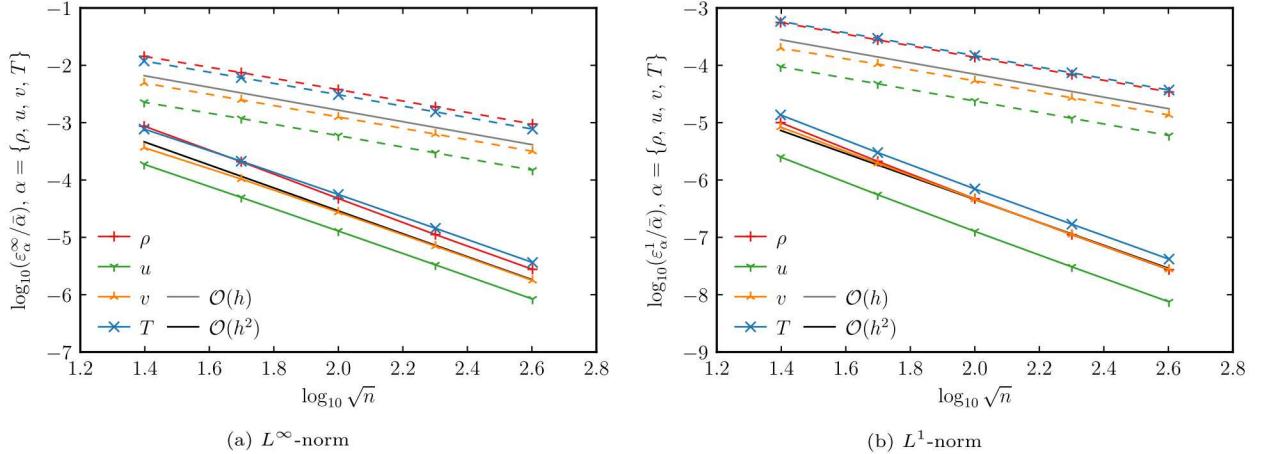


Figure 4: 2D MMS, $n_s = 1$, $T_v = T$, $\dot{\mathbf{w}} = \mathbf{0}$: Norms of the error. Dashed lines denote original boundary conditions; solid lines denote corrected boundary conditions.

4.1.3. 2D Supersonic Flow using an Exact Solution

This test exercises the same boundary conditions exercised in Section 4.1.2, but for an exact solution that does not require an additional source term. The exact solution is a steady, isentropic vortex, which we simulate in a quarter-annulus domain [36, 37, 38].

The exact solutions for this case are

$$\begin{aligned} \rho(r) &= \rho_i \left[1 + \frac{\gamma-1}{2} M_i^2 \left(1 - \left(\frac{r_i}{r} \right)^2 \right) \right]^{\frac{1}{\gamma-1}}, \\ u_r(r) &= 0, \\ u_\theta(r) &= -a_i M_i \frac{r_i}{r}, \\ T(r) &= T_i \left[1 + \frac{\gamma-1}{2} M_i^2 \left(1 - \left(\frac{r_i}{r} \right)^2 \right) \right], \end{aligned}$$

with $\rho_i = 1$, $a_i = 1$, $M_i = 2.25$, and $T_i = 1/(\gamma R)$. r is the distance from the center of the full annulus, which is bounded between $r_i = 1$ and $r_o = 1.384$. The solutions are shown in Figure 5, with $u_i = v_i = a_i M_i$.

Upon solving (10), the observed order of accuracy p is computed from (14) for $\alpha = \{\rho, u, v, T\}$. Six 2D meshes are used, consisting of 32×8 , 64×16 , 128×32 , 256×64 , 512×128 , and 1024×256 elements. The 64×16 mesh is shown in Figure 6.

Table 6 shows the observed accuracy, using the L^∞ -norm of the error, which indicates second-order accuracy ($p = 2$). Figure 7 shows the L^∞ -norm for each of the flow variables.

Mesh	ρ	u	v	T
1-2	1.9896	1.9119	1.9943	1.9699
2-3	1.9735	1.9589	2.0070	1.9979
3-4	1.9954	1.9760	2.0099	2.0076
4-5	1.9972	1.9879	2.0054	2.0044
5-6	1.9986	1.9940	2.0029	2.0025

Table 6: 2D Exact, $n_s = 1$, $T_v = T$, $\dot{\mathbf{w}} = \mathbf{0}$: Observed accuracy p using L^∞ -norm of the error.

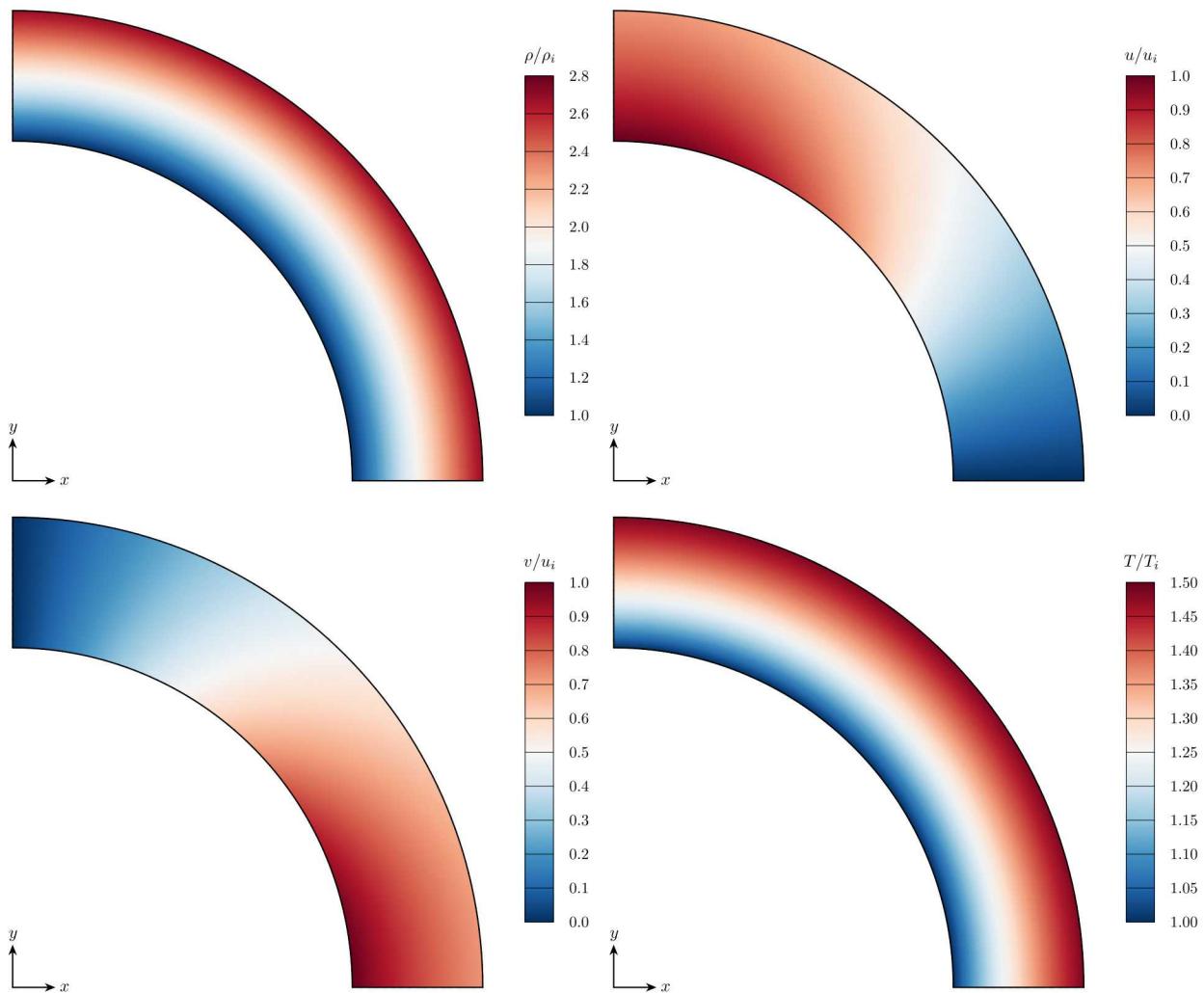


Figure 5: 2D Exact, $n_s = 1$, $T_v = T$, $\dot{\mathbf{w}} = \mathbf{0}$: Exact solutions.

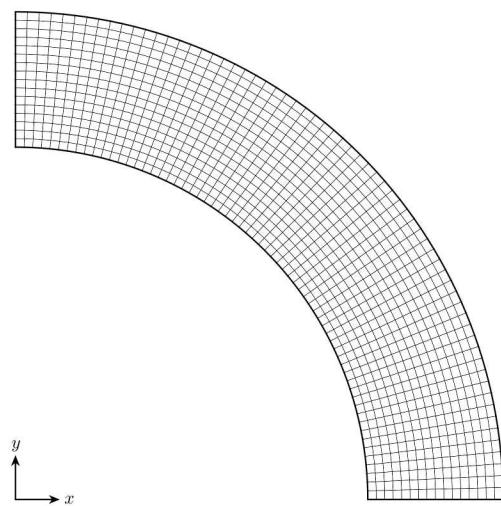


Figure 6: 2D Exact, $n_s = 1$, $T_v = T$, $\dot{\mathbf{w}} = \mathbf{0}$: Mesh with 64×16 elements.

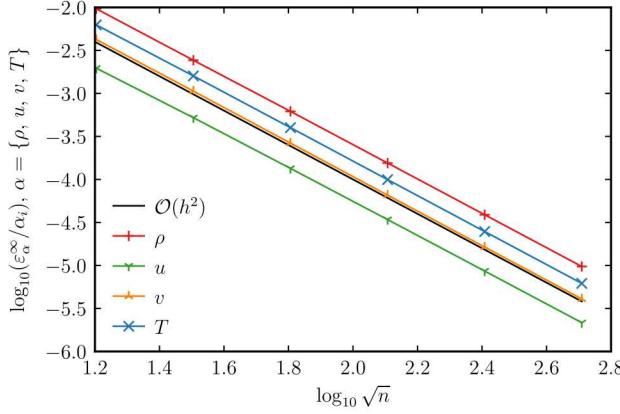


Figure 7: 2D Exact, $n_s = 1$, $T_v = T$, $\dot{\mathbf{w}} = \mathbf{0}$: L^∞ -norm of the error.

4.1.4. 3D Supersonic Flow using a Manufactured Solution

For this test, we consider a three-dimensional flow. The manufactured solutions for this case are

$$\begin{aligned}
 \rho(x, y, z) &= \bar{\rho} \left[1 - \epsilon \sin \left(\frac{5}{4} \pi x \right) (\sin(\pi y) + \cos(\pi y)) (\sin(\pi z) + \cos(\pi z)) \right], \\
 u(x, y, z) &= \bar{u} \left[1 + \epsilon \sin \left(\frac{1}{4} \pi x \right) (\sin(\pi y) + \cos(\pi y)) (\sin(\pi z) + \cos(\pi z)) \right], \\
 v(x, y, z) &= \bar{v} \left[-\epsilon \sin \left(\frac{5}{4} \pi x \right) (\sin(\pi y) + \cos(\pi y)) (\sin(\pi z) + \cos(\pi z)) \right], \\
 w(x, y, z) &= \bar{w} \left[-\epsilon \sin \left(\frac{5}{4} \pi x \right) (\sin(\pi y) + \cos(\pi y)) (\sin(\pi z) + \cos(\pi z)) \right], \\
 T(x, y, z) &= \bar{T} \left[1 + \epsilon \sin \left(\frac{5}{4} \pi x \right) (\sin(\pi y) + \cos(\pi y)) (\sin(\pi z) + \cos(\pi z)) \right],
 \end{aligned}$$

with $(x, y, z) \in [0, 1] \text{ m} \times [0, 1] \text{ m} \times [0, 1] \text{ m}$, and $\bar{\rho} = 1 \text{ kg/m}^3$, $\bar{M} = 2.5$, $\bar{v} = \bar{w} = \bar{u} = 20$, $\bar{T} = 300 \text{ K}$, and $\epsilon = 0.05$.

Upon solving (13), the observed order of accuracy p is computed from (14) for $\alpha = \{\rho, u, v, w, T\}$. Five 3D meshes are used, consisting of $25 \times 25 \times 25$, $50 \times 50 \times 50$, $100 \times 100 \times 100$, $200 \times 200 \times 200$, and $400 \times 400 \times 400$ elements. These meshes are chosen to test the spatial accuracy of the discretization for nonuniform meshes, and are created using the approach in Appendix B. The $50 \times 50 \times 50$ mesh is shown in Figure 8.

Table 7 shows the observed accuracy, using the L^∞ -norm of the error, which indicates second-order accuracy ($p = 2$). Figure 9 shows the L^∞ -norm for each of the flow variables.

Mesh	ρ	u	v	w	T
1–2	2.0849	1.8731	1.9841	1.7039	1.9404
2–3	2.1406	1.9923	1.9295	1.8621	1.9774
3–4	2.0990	2.0115	1.9623	1.9349	1.9922
4–5	2.0585	2.0100	1.9820	1.9571	1.9964

Table 7: 3D MMS, $n_s = 1$, $T_v = T$, $\dot{\mathbf{w}} = \mathbf{0}$: Observed accuracy p using L^∞ -norm of the error.

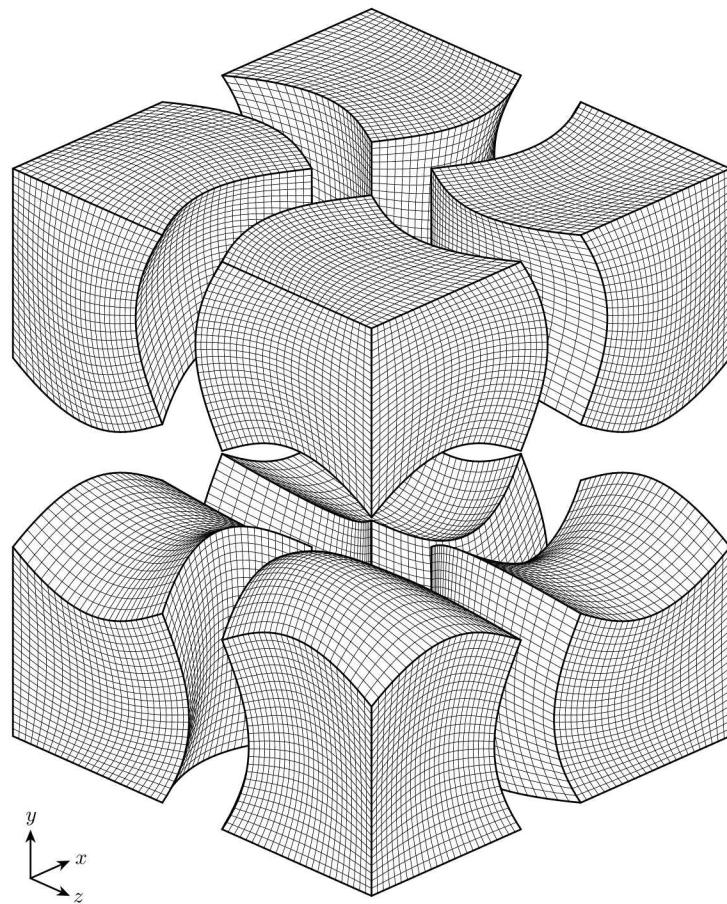


Figure 8: 3D MMS, $n_s = 1$, $T_v = T$, $\dot{\mathbf{w}} = \mathbf{0}$: Mesh with $50 \times 50 \times 50$ elements.

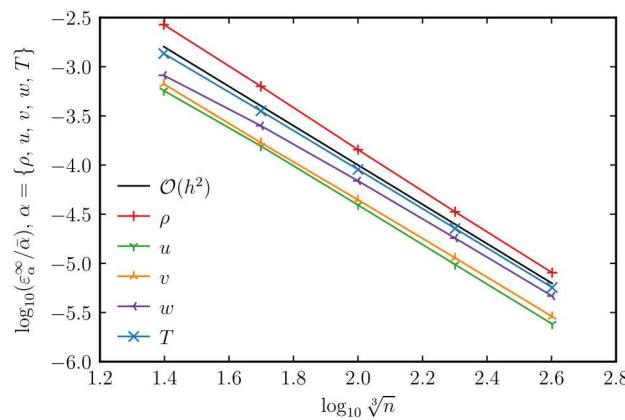


Figure 9: 3D MMS, $n_s = 1$, $T_v = T$, $\dot{\mathbf{w}} = \mathbf{0}$: L^∞ -norm of the error.

4.2. Five-Species Inviscid Flow in Chemical Nonequilibrium

This set of tests uses the five-species air model ($n_s = 5$, $\boldsymbol{\rho} = \{\rho_{N_2}, \rho_{O_2}, \rho_{NO}, \rho_N, \rho_O\}$) mentioned in Section 2.5. The flow is inviscid ($\mathbf{F}_d = 0$) and in chemical nonequilibrium ($\dot{\mathbf{w}} \neq \mathbf{0}$). For this type of flow, the reference velocity \bar{u} is determined from a reference Mach number \bar{M} :

$$\bar{u} = \bar{M} \sqrt{\gamma \left(\sum_{s=1}^{n_s} \frac{\rho_s}{\rho} \frac{\bar{R}}{M_s} \right) \bar{T}}. \quad (21)$$

While these tests do not directly test the thermochemical-source-term implementation (as described in Section 5), they do test the coupling of the source term with the differential terms, as well as the spatial discretizations of multiple species and temperatures.

4.2.1. 2D Supersonic Flow in Thermal Equilibrium using a Manufactured Solution

For this test, the flow is in thermal equilibrium ($T = T_v$). The manufactured solutions for this case for $\alpha = \{\rho_{N_2}, \rho_{O_2}, \rho_{NO}, \rho_N, \rho_O, u, v, T\}$ are listed in (19) and shown in Figure 1.

The domain is a square with $(x, y) \in [0, 1] \text{ m} \times [0, 1] \text{ m}$, and $\bar{\rho}_{N_2} = 0.77 \text{ kg/m}^3$, $\bar{\rho}_{O_2} = 0.20 \text{ kg/m}^3$, $\bar{\rho}_{NO} = 0.01 \text{ kg/m}^3$, $\bar{\rho}_N = 0.01 \text{ kg/m}^3$, $\bar{\rho}_O = 0.01 \text{ kg/m}^3$, $M = 2.5$, $\bar{v} = \bar{u}$ (21), $\bar{T} = 3500 \text{ K}$ and $\epsilon = 0.05$.

Upon solving (13), the observed order of accuracy p is computed from (14) for $\alpha = \{\rho_{N_2}, \rho_{O_2}, \rho_{NO}, \rho_N, \rho_O, u, v, T\}$. Seven 2D meshes are used, consisting of 25×25 , 50×50 , 100×100 , 200×200 , 400×400 , 800×800 , and 1600×1600 elements. These meshes are the same nonuniform meshes as those used in Section 4.1.2.

Table 8 shows the observed accuracy, using the L^∞ -norm of the error, which indicates second-order accuracy ($p = 2$). Figure 10 shows the L^∞ -norm for each of the flow variables.

Mesh	ρ_{N_2}	ρ_{O_2}	ρ_{NO}	ρ_N	ρ_O	u	v	T
1–2	2.0608	2.1382	2.0698	2.0644	2.1885	1.8425	1.8289	1.7351
2–3	2.1161	2.1219	2.1127	2.1072	2.1697	1.8875	1.9220	1.7923
3–4	2.0798	2.0813	1.8555	2.0754	2.0971	1.9200	1.9686	1.8525
4–5	2.0456	2.0458	1.8917	2.0428	2.0806	1.9522	1.9871	1.9079
5–6	2.0243	2.0243	1.9427	2.0228	2.0529	1.9735	1.9939	1.9485
6–7	2.0125	2.0125	1.9790	2.0118	2.0318	1.9865	1.9969	1.9737

Table 8: 2D MMS, $n_s = 5$, $T_v = T$, $\dot{\mathbf{w}} \neq \mathbf{0}$: Observed accuracy p using L^∞ -norm of the error.

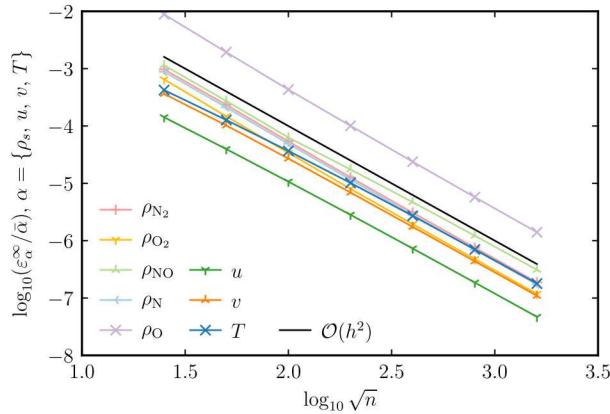


Figure 10: 2D MMS, $n_s = 5$, $T_v = T$, $\dot{\mathbf{w}} \neq \mathbf{0}$: L^∞ -norm of the error.

4.2.2. 2D Hypersonic Flow in Thermal Nonequilibrium using a Manufactured Solution

For this test, the flow is in thermal nonequilibrium ($T \neq T_v$). The manufactured solutions for this case for $\alpha = \{\rho_{N_2}, \rho_{O_2}, \rho_{NO}, \rho_N, \rho_O, u, v, T, T_v\}$ are listed in (19) and shown in Figure 1.

The domain is a square with $(x, y) \in [0, 1] \text{ m} \times [0, 1] \text{ m}$, and $\bar{\rho}_{N_2} = 0.0077 \text{ kg/m}^3$, $\bar{\rho}_{O_2} = 0.0020 \text{ kg/m}^3$, $\bar{\rho}_{NO} = 0.0001 \text{ kg/m}^3$, $\bar{\rho}_N = 0.0001 \text{ kg/m}^3$, $\bar{\rho}_O = 0.0001 \text{ kg/m}^3$, $\bar{M} = 8$, $\bar{v} = \bar{u} (21)$, $\bar{T} = 5000 \text{ K}$, $\bar{T}_v = 1000 \text{ K}$, and $\epsilon = 0.05$.

Upon solving (13), the observed order of accuracy p is computed from (14) for $\alpha = \{\rho_{N_2}, \rho_{O_2}, \rho_{NO}, \rho_N, \rho_O, u, v, T, T_v\}$. Seven 2D meshes are used, consisting of 25×25 , 50×50 , 100×100 , 200×200 , 400×400 , 800×800 , and 1600×1600 elements. These meshes are the same nonuniform meshes as those used in Section 4.1.2.

Table 9 shows the observed accuracy, using the L^∞ -norm of the error, which indicates second-order accuracy ($p = 2$). Figure 11 shows the L^∞ -norm for each of the flow variables.

Mesh	ρ_{N_2}	ρ_{O_2}	ρ_{NO}	ρ_N	ρ_O	u	v	T	T_v
1–2	1.5659	1.6370	1.6555	1.6046	1.5869	1.7742	1.7337	1.7814	1.5545
2–3	1.9067	1.6944	1.6986	1.7598	1.8819	1.8916	1.8701	1.8768	1.9150
3–4	1.9868	2.0475	2.0698	2.0477	2.0110	1.9488	1.9357	1.9349	2.0082
4–5	2.0074	1.9941	2.0138	1.9936	2.0089	1.9752	1.9684	1.9672	2.0168
5–6	2.0062	1.9939	2.0004	1.9935	2.0061	1.9879	1.9843	1.9836	2.0111
6–7	2.0037	1.9965	1.9994	1.9962	1.9955	1.9940	1.9922	1.9918	2.0063

Table 9: 2D MMS, $n_s = 5$, $T_v \neq T$, $\dot{\mathbf{w}} \neq \mathbf{0}$: Observed accuracy p using L^∞ -norm of the error.

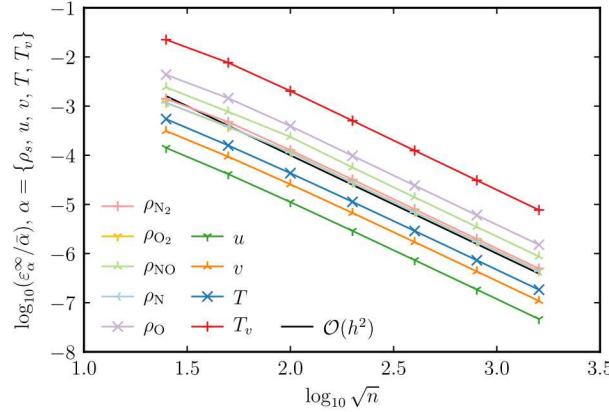


Figure 11: 2D MMS, $n_s = 5$, $T_v \neq T$, $\dot{\mathbf{w}} \neq \mathbf{0}$: L^∞ -norm of the error.

5. Verification Techniques for Thermochemical Source Term

While measuring the spatial accuracy is an effective technique for assessing the discretization, it does not directly reveal errors in the coding of the source term $\mathbf{S}(\mathbf{U})$ in (1). With manufactured solutions, for example, terms containing derivatives in (1) are evaluated numerically when computing $\mathbf{r}_h(\mathbf{U}_h)$ in (13) and analytically when computing $\mathbf{r}(\mathbf{U}_{MS})$. The thermochemical source term is similarly evaluated on both sides of (13); however, because the algebraic source-term evaluation does not depend on the spatial discretization, these evaluations use the same source code. Therefore, there are no differences in the evaluation on each side, and, as a result, errors in the source-term implementation are not detected.

To address this limitation, we independently developed a code to compute the terms within the source term, specifically $Q_{t-v}(\rho, T, T_v)$, $\mathbf{e}_v(\rho, T, T_v)$, and $\dot{\mathbf{w}}(\rho, T, T_v)$ in (1). We compute these terms for many samples of $\{\rho, T, T_v\}$ and compare with those obtained from SPARC for a single-cell mesh when initialized to

those values with no velocity. Additionally, we assess the sufficiency of the number of samples by performing convergence studies on distribution properties of the values and on the differences between the two codes.

Because this approach only computes the thermochemical source term, it can be considered an extensive unit test. Examples of simpler, traditional unit tests include test computations of the forward reaction-rate coefficient k_{fr} and the equilibrium constant K_{er} for each type of reaction, the translational–vibrational energy relaxation time $\tau_{s,m,s'}$ for each collision, and the dependent intermediate computations to obtain Q_{t-v} and $\dot{\mathbf{w}}$.

However, chemical-kinetics models provide individual reaction-rate parameters as a set, with the individual parameters determined in a manner such that a model best matches experimental data in the regime of interest. The nonlinear interactions of the individual reactions can lead to much richer model behavior than the simple structure of the reaction-rate forms suggests. In addition to detecting errors, our sampling approach attempts to confirm that the performance of the chemical-kinetics model is not sensitive to the particular implementation. These remarks apply to the vibrational nonequilibrium model as well, but to a lesser extent, as the models are less complicated and model parameters are more accessible to theoretical determination.

We have opted for sampling as an expedient approach that can be easily applied to other algebraic terms. However, it may be more efficient to analyze a chemical-kinetics model in detail to identify regions or surfaces of high sensitivity, and use an adaptive sampling or optimization approach to choose the locations to compare alternative implementations. Nonetheless, through our approach, we are able to quickly detect errors.

While it may be instinctive to dismiss these techniques as a typically low-rigor code-to-code comparison, we clarify the distinctive and rigorous features.

1. This code is independently developed, using the same models and material properties expected to be employed by SPARC but taken directly from the original references. Alternatively, external software could be used, but, given the variety of published models and material properties, quantifying the agreement and, consequently, assessing the implementation becomes non-trivial.
2. Because the models and material properties are the same, when computing the difference in the source terms, the required tolerance is tightened from what may typically be a few percent to near machine precision.
3. For numerical solutions to partial differential equations, code-to-code comparisons are typically employed for a few canonical cases, through which it is difficult to identify and isolate errors in the numerical-method implementation or attribute differences to specific sources. On the other hand, our code-to-code comparison targets the portion of the code that manufactured solutions do not assess, and heavily queries conditions covering the thermochemical model’s domain of validity.

The effectiveness of these techniques is demonstrated in Section 6.

6. Thermochemical-Source-Term Verification Results

To assess the correctness of the thermochemical-source-term implementation, we generate n_S Latin hypercube samples using the ranges and spacings listed in Table 10, with $n_S = 2^i$, for $i = 0, \dots, 17$. At these samples, we query SPARC and the independent code described in Section 5 to compute $Q_{t-v}(\rho, T, T_v)$, $\mathbf{e}_v(\rho, T, T_v)$, and $\dot{\mathbf{w}}(\rho, T, T_v)$ in (1).

For each n_S , Figures 12a, 13a, and 14a show the minimum, mean, and maximum of the translational–vibrational energy exchange Q_{t-v} , the vibrational energies per mass \mathbf{e}_v , and the mass production rates per volume $\dot{\mathbf{w}}$, as computed from the independent code. For the vector quantities \mathbf{e}_v and $\dot{\mathbf{w}}$, the elements are pooled. As n_S is increased, these values are expected to converge; however, because the samples at each n_S are independently determined, the convergence is not monotonic. Convergence of these values suggests the distribution of the values is sufficiently resolved, and, therefore, the values are sufficiently represented.

For $n_S = 2^{17} = 131,072$, Figures 15–17 show the distributions of $|Q_{t-v}|$, \mathbf{e}_v , and $|\dot{\mathbf{w}}|$, as computed from the independent code. n_q denotes the number of queries within the ranges on the abscissa. Since Q_{t-v} and $\dot{\mathbf{w}}$ can be non-positive, their ranges, as well as those of \mathbf{e}_v are listed in Table 11. All of these values vary drastically in magnitude, and, with the exception of \mathbf{e}_v , in sign.

Variable	Minimum	Maximum	Units	Spacing
ρ_{N_2}	10^{-6}	10^1	kg/m ³	Logarithmic
ρ_{O_2}	10^{-6}	10^1	kg/m ³	Logarithmic
ρ_{NO}	10^{-6}	10^1	kg/m ³	Logarithmic
ρ_N	10^{-6}	10^1	kg/m ³	Logarithmic
ρ_O	10^{-6}	10^1	kg/m ³	Logarithmic
T	100	15,000	K	Linear
T_v	100	15,000	K	Linear

Table 10: $\mathbf{S}(\mathbf{U})$, $n_s = 5$, $T_v \neq T$, $\dot{\mathbf{w}} \neq \mathbf{0}$: Ranges and spacings for Latin hypercube samples of ρ , T , and T_v .

For every sample, we compute a symmetric relative difference, defined by

$$\delta_\beta = 2 \frac{|\beta_{\text{SPARC}} - \beta'|}{|\beta_{\text{SPARC}}| + |\beta'|}, \quad (22)$$

where $\beta = \{Q_{t-v}, e_{v_{N_2}}, e_{v_{O_2}}, e_{v_{NO}}, \dot{w}_{N_2}, \dot{w}_{O_2}, \dot{w}_{NO}, \dot{w}_N, \dot{w}_O\}$, and the prime denotes computation by the independent code.

For each n_s , Figures 12b, 13b, and 14b show the maximum relative difference across the samples. As with Figures 12a, 13a, and 14a, the maximum relative difference is expected to increase and converge, through not monotonically. Additionally, for $n_s = 2^{17}$, Figures 18a, 19a, and 20a show the relative differences in Q_{t-v} , \mathbf{e}_v , and $\dot{\mathbf{w}}$.

As mentioned in Section 5, the relative differences (22) are expected to be near machine precision; however, this is clearly not the case for the red curves in Figures 12b and 13b or the histograms in Figures 18a and 19a. As shown in Figure 18a, for approximately 8.7% of the queries, $\delta_{Q_{t-v}}$ is greater than 10%, and, for 29% of the queries, $\delta_{Q_{t-v}}$ is greater than 1%. Additionally, as shown in Figure 19a, although $\delta_{\mathbf{e}_v}$ is less than 10^{-12} for 99% of the queries, $\delta_{\mathbf{e}_v}$ is greater than 100% for a few of the queries. Even with $n_s = 1$, the red curve in Figure 12b indicates $\delta_{Q_{t-v}} = 3.9\%$. On the other hand, in Figure 13b, $\delta_{\mathbf{e}_v}$ is within machine precision through $n_s = 32$, but increases by orders of magnitude at $n_s = 64$ and $n_s = 128$. These observations demonstrate how thirty-two samples are not enough to represent a seven-dimensional space.

These high relative differences were due to two causes.

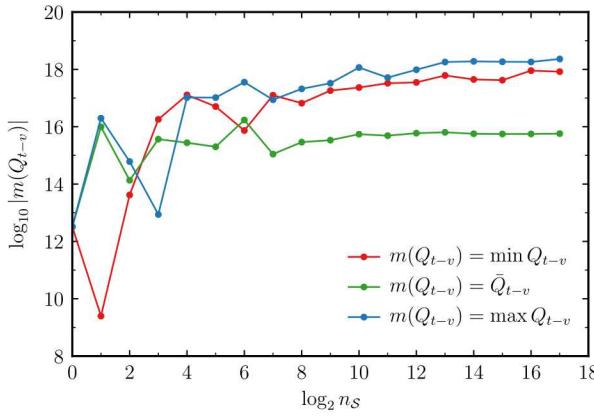
1. The lookup table used by SPARC contained incorrect values for the vibrational constants used in (3) for N_2 and O_2 when the colliding species is NO. These incorrect values introduced an error in Q_{t-v} for all samples.
2. The convergence criteria specified in the implementation of Newton's method used to compute T_v from ρe_v was loose. Though sufficient for most values of T_v , these criteria prove unsuitable for low values. These criteria introduced errors in Q_{t-v} and \mathbf{e}_v for a few samples. T_v also appears in T_c for dissociative reactions, but the actual impact on $\dot{\mathbf{w}}$ was quite small. For a converged, steady problem, however, the original convergence criteria is not expected to affect the final solution.

Upon correcting the lookup-table values and tightening the convergence criteria, we reran the SPARC simulations and recomputed $\delta_{Q_{t-v}}$, $\delta_{\mathbf{e}_v}$, and $\delta_{\dot{\mathbf{w}}}$, which are shown in Figures 18b, 19b, and 20b for $n_s = 2^{17}$. These results are consistent with our expectations, as all $\delta_{Q_{t-v}}$ values are less than 10^{-10} ; all $\delta_{\mathbf{e}_v}$ values are less than 10^{-14} ; and, with the exception of one query, which is slightly greater, all $\delta_{\dot{\mathbf{w}}}$ values are less than 10^{-10} .

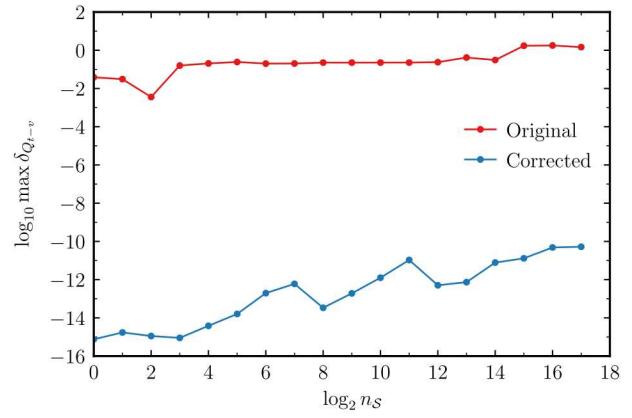
Of the 131,072 $\delta_{Q_{t-v}}$ values, the forty-eight greater than 10^{-12} occur when T and T_v have a relative difference of less than 0.2%. As a result, in the numerator of (2), $e_{v_{s,m}}(T)$ and $e_{v_{s,m}}(T_v)$ share many of the leading digits; therefore, precision is lost when computing their difference.

Of the 655,360 $\delta_{\dot{\mathbf{w}}}$ values computed from the 131,072 samples, 109 are greater than 10^{-12} . These slightly elevated differences are a result of the precision loss that can occur from subtraction in (5).

The blue curves in Figures 12b, 13b, and 14b show how the maximum relative differences vary with respect to n_s after correcting the lookup-table values and tightening the convergence criteria. As we expect, these curves are generally increasing and converging. The change in T_c impacts $\delta_{\dot{\mathbf{w}}}$ only about as much as the precision loss.

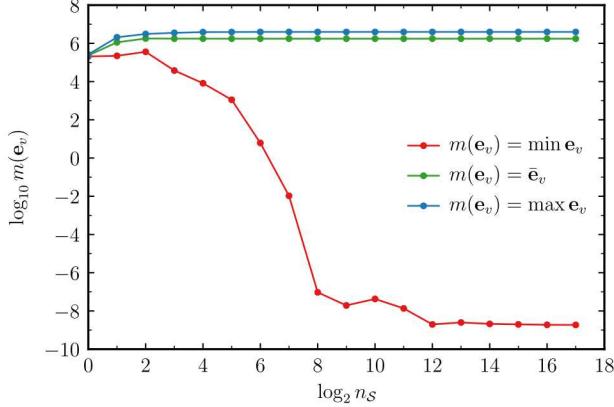


(a) Minimum, mean, and maximum

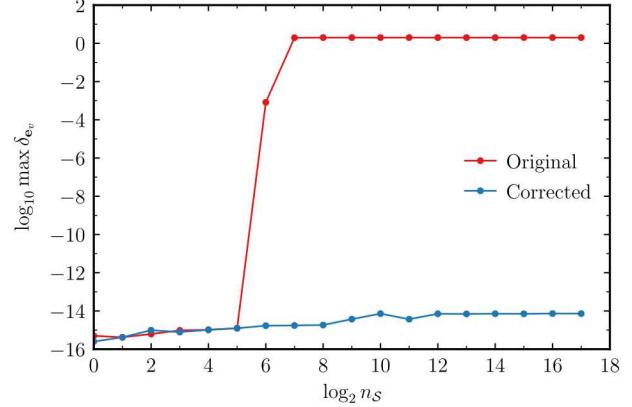


(b) Maximum relative difference

Figure 12: $\mathbf{S}(\mathbf{U})$, $n_s = 5$, $T_v \neq T$, $\dot{\mathbf{w}} \neq \mathbf{0}$: Convergence history of Q_{t-v} sampling.

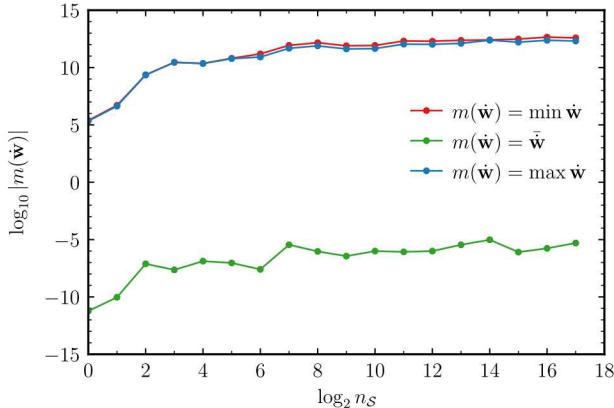


(a) Minimum, mean, and maximum

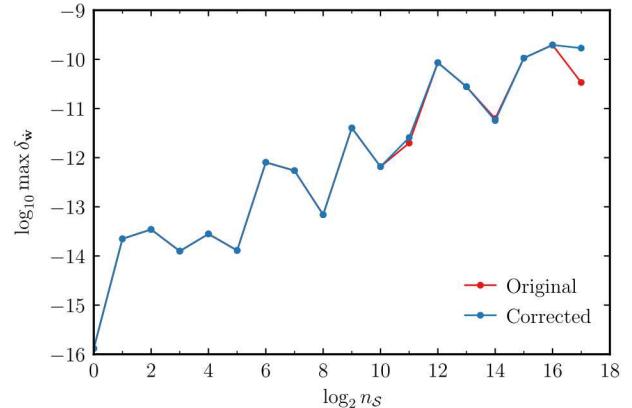


(b) Maximum relative difference

Figure 13: $\mathbf{S}(\mathbf{U})$, $n_s = 5$, $T_v \neq T$, $\dot{\mathbf{w}} \neq \mathbf{0}$: Convergence history of e_v sampling.



(a) Minimum, mean, and maximum



(b) Maximum relative difference

Figure 14: $\mathbf{S}(\mathbf{U})$, $n_s = 5$, $T_v \neq T$, $\dot{\mathbf{w}} \neq \mathbf{0}$: Convergence history of \hat{w} sampling.

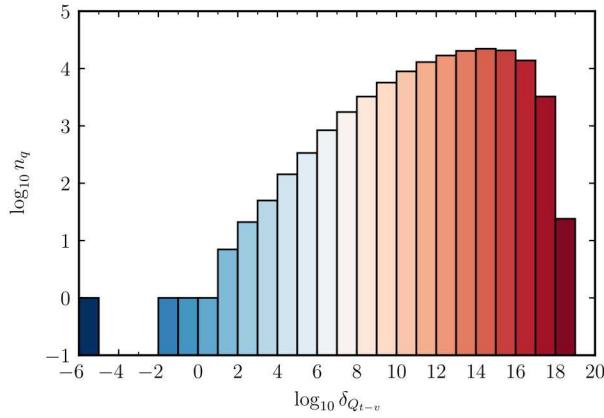


Figure 15: $\mathbf{S}(\mathbf{U})$, $n_s = 5$, $T_v \neq T$, $\dot{\mathbf{w}} \neq \mathbf{0}$: Absolute values of the translational–vibrational energy exchange Q_{t-v} for $n_S = 2^{17}$.

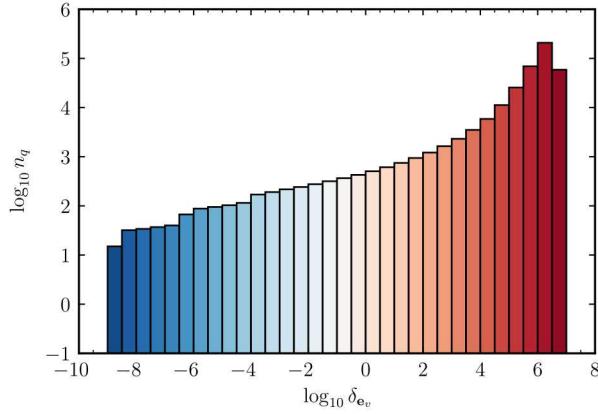


Figure 16: $\mathbf{S}(\mathbf{U})$, $n_s = 5$, $T_v \neq T$, $\dot{\mathbf{w}} \neq \mathbf{0}$: Vibrational energies per mass \mathbf{e}_v for $n_S = 2^{17}$.

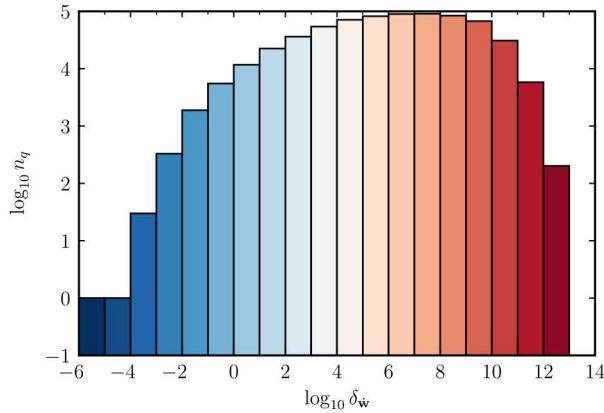


Figure 17: $\mathbf{S}(\mathbf{U})$, $n_s = 5$, $T_v \neq T$, $\dot{\mathbf{w}} \neq \mathbf{0}$: Absolute values of the mass production rates per volume $\dot{\mathbf{w}}$ for $n_S = 2^{17}$.

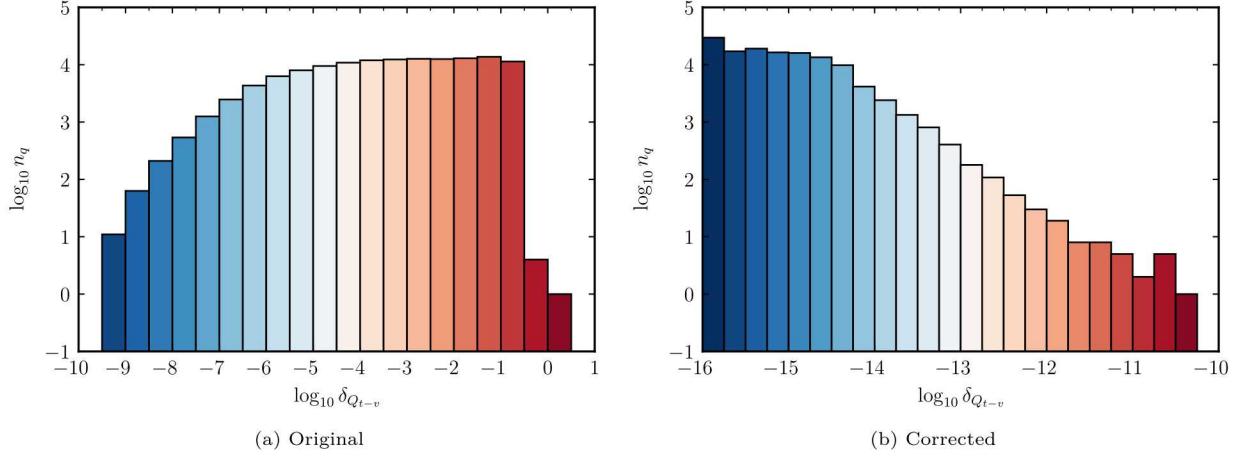


Figure 18: $\mathbf{S}(\mathbf{U})$, $n_s = 5$, $T_v \neq T$, $\dot{\mathbf{w}} \neq \mathbf{0}$: Relative differences in the translational–vibrational energy exchange Q_{t-v} for $n_S = 2^{17}$, using the original lookup table and convergence criteria (a), and the corrected lookup table with tighter convergence criteria (b). Queries with $\delta_{Q_{t-v}} = 0$ are placed in the lowest bin.

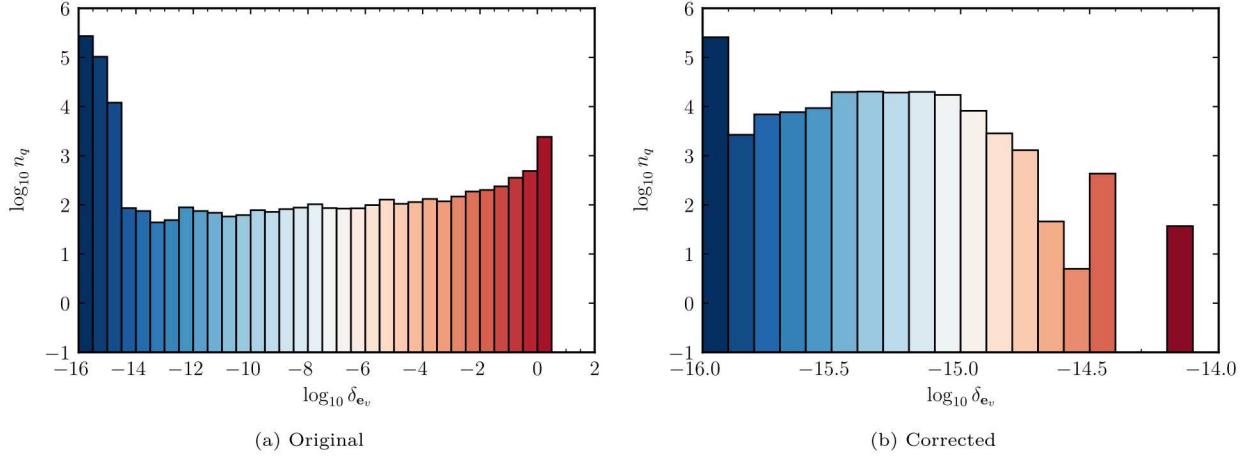


Figure 19: $\mathbf{S}(\mathbf{U})$, $n_s = 5$, $T_v \neq T$, $\dot{\mathbf{w}} \neq \mathbf{0}$: Relative differences in the vibrational energies per mass \mathbf{e}_v for $n_S = 2^{17}$, using the original convergence criteria (a), and the tighter convergence criteria (b). Queries with $\delta_{\mathbf{e}_v} = 0$ are placed in the lowest bin.

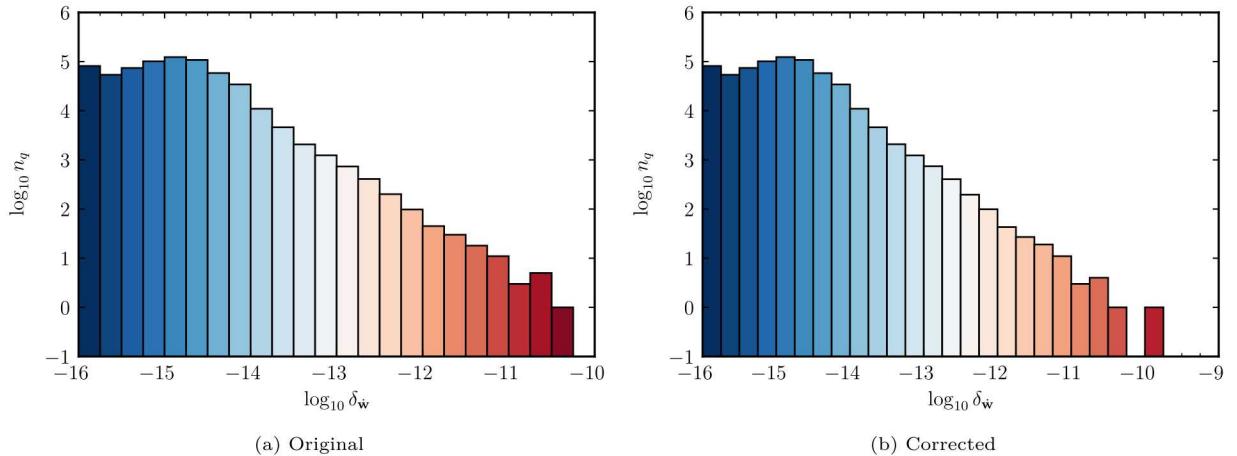


Figure 20: $\mathbf{S}(\mathbf{U})$, $n_s = 5$, $T_v \neq T$, $\dot{\mathbf{w}} \neq \mathbf{0}$: Relative differences in the mass production rates per volume $\dot{\mathbf{w}}$ for $n_S = 2^{17}$, using the original convergence criteria (a), and the tighter convergence criteria (b). Queries with $\delta_{\dot{\mathbf{w}}} = 0$ are placed in the lowest bin.

Variable	Minimum	Maximum	Units
Q_{t-v}	-8.3102×10^{17}	2.3066×10^{18}	$\text{J/m}^3/\text{s}$
$e_{v_{\text{N}_2}}$	1.8510×10^{-9}	3.9668×10^6	J/kg
$e_{v_{\text{O}_2}}$	1.1129×10^{-4}	3.6138×10^6	J/kg
$e_{v_{\text{NO}}}$	4.6272×10^{-7}	3.7781×10^6	J/kg
\dot{w}_{N_2}	-2.1148×10^{10}	4.4677×10^{11}	$\text{kg/m}^3/\text{s}$
\dot{w}_{O_2}	-1.9297×10^{12}	1.1891×10^{11}	$\text{kg/m}^3/\text{s}$
\dot{w}_{NO}	-3.8719×10^{12}	1.8000×10^{12}	$\text{kg/m}^3/\text{s}$
\dot{w}_{N}	-8.4915×10^{11}	1.7631×10^{12}	$\text{kg/m}^3/\text{s}$
\dot{w}_{O}	-3.6139×10^{11}	2.0645×10^{12}	$\text{kg/m}^3/\text{s}$

Table 11: $\mathbf{S}(\mathbf{U})$, $n_s = 5$, $T_v \neq T$, $\dot{\mathbf{w}} \neq \mathbf{0}$: Ranges for Q_{t-v} , \mathbf{e}_v , and $\dot{\mathbf{w}}$ for $n_{\mathcal{S}} = 2^{17}$.

Though the incorrect lookup-table values introduced high relative differences immediately, the convergence criteria required more samples to detect their looseness. The convergence of the distribution properties in Figures 12a, 13a, and 14a and the convergence and bounds of the errors in Figures 12b, 13b, and 14b provide insight into how many samples are enough to assess the correctness of the thermochemical-source-term implementation over the ranges listed in Table 10.

To determine the impact of these modifications, we reran a high-enthalpy (20 MJ/kg), hypersonic, laminar double-cone flow case after correcting the lookup table values and tightening the convergence criteria. Relative to the original values and convergence criteria, we observed up to a 1.4% and 2.7% change in the pressure and heat flux, respectively, on the surface of the body.

7. Conclusions

In this paper, we presented our code-verification techniques for hypersonic reacting flows in thermochemical nonequilibrium.

To assess the spatial accuracy, we employed manufactured and exact solutions with the L^∞ -norm and L^1 -norm. These approaches revealed the impact of the lower-order boundary conditions.

To assess the algebraic thermochemical source term, we queried an independent code, with the expectation that agreement be near machine precision. We queried the independent code many times and studied the convergence properties. This approach revealed the impact of erroneous lookup table entries and insufficiently tight convergence criteria.

While the scope of this paper has been limited to flows in vibrational nonequilibrium with five species that undergo dissociation and exchange reactions, these techniques could be analogously extended to address more complex flows in rotational and electronic nonequilibrium that contain additional species capable of undergoing ionization reactions. However, it is important to perform a convergence study to determine whether the number of samples sufficiently spans the ranges of the dependencies.

Acknowledgments

The authors thank Derek Dinzl, Travis Fisher, Micah Howard, and Ross Wagnild, for their valuable assistance with SPARC and the underlying models and properties. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

Appendix A. Five-Species Air Model Properties

The molecular weights M_s and formation enthalpies h_s^o for the five species are tabulated in Table A1.

s	Species	M_s [g/mol]	h_s^o , 0 K [J/kg]
1	N_2	28.016	0
2	O_2	32.000	0
3	NO	30.008	2.996123×10^6
4	N	14.008	3.362161×10^7
5	O	16.000	1.542000×10^7

Table A1: Five-species air model: Molecular weights M_s and formation enthalpies h_s^o [39].

The characteristic vibrational temperatures $\theta_{v_{s,m}}$ [39] and collision-limiting cross sections at 50,000 K σ'_{v_s} [24, 25] are tabulated in Table A2.

s	Species	n_{v_s}	$\theta_{v_{s,m}}$ [K]	σ'_{v_s} [m 2]
1	N_2	1	3395	3×10^{-21}
2	O_2	1	2239	3×10^{-21}
3	NO	1	2817	3×10^{-21}

Table A2: Five-species air model: Characteristic vibrational temperatures $\theta_{v_{s,m}}$ [39] and collision-limiting cross sections at 50,000 K σ'_{v_s} [24, 25].

The vibrational constants $a_{s,m,s'}$ and $b_{s,m,s'}$ for Equation (3), associated with the reactions listed in Table 1, are listed in Table A3.

s'	N_2		O_2		NO	
	$a_{s,m,s'}$	$b_{s,m,s'}$	$a_{s,m,s'}$	$b_{s,m,s'}$	$a_{s,m,s'}$	$b_{s,m,s'}$
N_2	Equation (4)		Equation (4)		49.5	0.0420
O_2	Equation (4)		Equation (4)		49.5	0.0420
NO	Equation (4)		Equation (4)		49.5	0.0420
N	Equation (4)	72.4	0.0150	49.5	0.0420	
O	72.4	0.0150	47.7	0.0590	49.5	0.0420

Table A3: Five-species air model: Vibrational constants $a_{s,m,s'}$ and $b_{s,m,s'}$ [24].

The stoichiometric coefficients $\alpha_{s,r}$ and $\beta_{s,r}$ associated with the reactions listed in Table 1 are listed in Table A4.

r	$\alpha_{s,r}$					$\beta_{s,r}$				
	N_2	O_2	NO	N	O	N_2	O_2	NO	N	O
1	2					1			2	
2	1	1					1		2	
3	1		1					1	2	
4	1			1					3	
5	1				1			2	1	
6	1	1				1			2	
7		2					1		2	
8		1	1					1	2	
9		1		1					1	2
10		1			1				3	
11	1		1			1			1	1
12		1	1				1		1	1
13			2					1	1	1
14			1	1				2	1	
15			1		1				1	2
16	1				1			1	1	
17			1	1		1			1	

Table A4: Five-species air model: Stoichiometric coefficients $\alpha_{s,r}$ and $\beta_{s,r}$.

The reaction-rate-coefficient dependencies C_{f_r} , η_r , and θ_r associated with the reactions listed in Table 1 are listed in Table A5.

r	C_{f_r} [CGS]	η_r	θ_r [K]
1	7.0×10^{21}	-1.6	113,200
2	7.0×10^{21}	-1.6	113,200
3	7.0×10^{21}	-1.6	113,200
4	3.0×10^{22}	-1.6	113,200
5	3.0×10^{22}	-1.6	113,200
6	2.0×10^{21}	-1.5	59,500
7	2.0×10^{21}	-1.5	59,500
8	2.0×10^{21}	-1.5	59,500
9	1.0×10^{22}	-1.5	59,500
10	1.0×10^{22}	-1.5	59,500
11	5.0×10^{15}	0.0	75,500
12	5.0×10^{15}	0.0	75,500
13	1.1×10^{17}	0.0	75,500
14	1.1×10^{17}	0.0	75,500
15	1.1×10^{17}	0.0	75,500
16	6.4×10^{17}	-1.0	38,400
17	8.4×10^{12}	0.0	19,400

Table A5: Five-species air model: Reaction-rate-coefficient dependencies C_{f_r} , η_r , and θ_r [26].

The equilibrium-constant coefficients, for Equation (8), associated with the reactions listed in Table 1 are listed in Table A6.

r	A_{1_r}	A_{2_r}	A_{3_r}	A_{4_r}	A_{5_r}
1–5	1.606000	1.57320	1.39230	–11.53300	–0.0045430
6–10	0.641830	2.42530	1.90260	–6.62770	0.0351510
11–15	0.638170	0.68189	0.66336	–7.57730	–0.0110250
16	0.967940	0.89131	0.72910	–3.95550	0.0064880
17	–0.003732	–1.74340	–1.23940	–0.94952	–0.0461820

Table A6: Five-species air model: Equilibrium-constant coefficients A_{1_r} , A_{2_r} , A_{3_r} , A_{4_r} , and A_{5_r} [26].

Appendix B. Approach to Creating Nonuniform Meshes

To generate the nonuniform meshes in Figures 3 and 8, we employ the following nonlinear transformation from $(\xi, \eta, \zeta) \in [0, 1] \times [0, 1] \times [0, 1]$ to $(x, y, z) \in [0, L_x] \times [0, L_y] \times [0, L_z]$:

$$\mathbf{x} = \mathbf{L}\boldsymbol{\xi} + \mathbf{A}\hat{\boldsymbol{\xi}},$$

where $\mathbf{x} = \{x, y, z\}^T$, $\mathbf{L} = \text{diag}\{L_x, L_y, L_z\}$, $\boldsymbol{\xi} = \{\xi, \eta, \zeta\}^T$, $\hat{\boldsymbol{\xi}} = \mathbf{L}(\frac{1}{2} - \boldsymbol{\xi})$, and

$$\mathbf{A} = \begin{bmatrix} \alpha(s_{\xi,\eta} + s_{\xi,\zeta}) & -\beta s_{\xi,\eta} & \beta s_{\xi,\zeta} \\ \beta s_{\eta,\xi} & \alpha(s_{\eta,\xi} + s_{\eta,\zeta}) & -\beta s_{\eta,\zeta} \\ -\beta s_{\zeta,\xi} & \beta s_{\zeta,\eta} & \alpha(s_{\zeta,\xi} + s_{\zeta,\eta}) \end{bmatrix},$$

with $\alpha = 1 - \cos \frac{\pi}{6}$, $\beta = \sin \frac{\pi}{6}$, and $s_{\xi',\eta'} = \sin(\pi\xi') \sin(\pi\eta')$. For the mesh in Figure 3, $\zeta = z = 0$.

References

- [1] P. A. Gnoffo, R. N. Gupta, J. L. Shinn, Conservation equations and physical models for hypersonic air flows in thermal and chemical nonequilibrium, Tech. Rep. NASA-TP-2867, NASA Langley Research Center (1989).
- [2] J. D. Anderson, Jr., Hypersonic and High-Temperature Gas Dynamics, 2nd Edition, American Institute of Aeronautics and Astronautics, 2006. [doi:10.2514/4.861956](https://doi.org/10.2514/4.861956).
- [3] P. J. Roache, Verification and Validation in Computational Science and Engineering, Hermosa Publishers, 1998.
- [4] K. Salari, P. Knupp, Code verification by the method of manufactured solutions, Sandia Report SAND2000-1444, Sandia National Laboratories (Jun. 2000). [doi:10.2172/759450](https://doi.org/10.2172/759450).
- [5] W. L. Oberkampf, C. J. Roy, Verification and Validation in Scientific Computing, Cambridge University Press, 2010. [doi:10.1017/cbo9780511760396](https://doi.org/10.1017/cbo9780511760396).
- [6] C. J. Roy, Review of code and solution verification procedures for computational simulation, Journal of Computational Physics 205 (1) (2005) 131–156. [doi:10.1016/j.jcp.2004.10.036](https://doi.org/10.1016/j.jcp.2004.10.036).
- [7] P. J. Roache, Code verification by the method of manufactured solutions, Journal of Fluids Engineering 124 (1) (2001) 4–10. [doi:10.1115/1.1436090](https://doi.org/10.1115/1.1436090).
- [8] C. J. Roy, C. C. Nelson, T. M. Smith, C. C. Ober, Verification of Euler/Navier–Stokes codes using the method of manufactured solutions, International Journal for Numerical Methods in Fluids 44 (6) (2004) 599–620. [doi:10.1002/fld.660](https://doi.org/10.1002/fld.660).

[9] R. B. Bond, C. C. Ober, P. M. Knupp, S. W. Bova, Manufactured solution for computational fluid dynamics boundary condition verification, *AIAA Journal* 45 (9) (2007) 2224–2236. [doi:10.2514/1.28099](https://doi.org/10.2514/1.28099).

[10] S. Veluri, C. Roy, E. Luke, Comprehensive code verification for an unstructured finite volume CFD code, in: 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, American Institute of Aeronautics and Astronautics, 2010. [doi:10.2514/6.2010-127](https://doi.org/10.2514/6.2010-127).

[11] T. Oliver, K. Estacio-Hiroms, N. Malaya, G. Carey, Manufactured solutions for the Favre-averaged Navier–Stokes equations with eddy-viscosity turbulence models, in: 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, American Institute of Aeronautics and Astronautics, 2012. [doi:10.2514/6.2012-80](https://doi.org/10.2514/6.2012-80).

[12] É. Chamberland, A. Fortin, M. Fortin, Comparison of the performance of some finite element discretizations for large deformation elasticity problems, *Computers & Structures* 88 (11) (2010) 664 – 673. [doi:10.1016/j.compstruc.2010.02.007](https://doi.org/10.1016/j.compstruc.2010.02.007).

[13] S. Étienne, A. Garon, D. Pelletier, Some manufactured solutions for verification of fluid–structure interaction codes, *Computers & Structures* 106–107 (2012) 56–67. [doi:10.1016/j.compstruc.2012.04.006](https://doi.org/10.1016/j.compstruc.2012.04.006).

[14] A. Veeraragavan, J. Beri, R. J. Gollan, Use of the method of manufactured solutions for the verification of conjugate heat transfer solvers, *Journal of Computational Physics* 307 (2016) 308–320. [doi:10.1016/j.jcp.2015.12.004](https://doi.org/10.1016/j.jcp.2015.12.004).

[15] P. T. Brady, M. Herrmann, J. M. Lopez, Code verification for finite volume multiphase scalar equations using the method of manufactured solutions, *Journal of Computational Physics* 231 (7) (2012) 2924–2944. [doi:10.1016/j.jcp.2011.12.040](https://doi.org/10.1016/j.jcp.2011.12.040).

[16] R. G. McClarren, R. B. Lowrie, Manufactured solutions for the p_1 radiation-hydrodynamics equations, *Journal of Quantitative Spectroscopy and Radiative Transfer* 109 (15) (2008) 2590–2602. [doi:10.1016/j.jqsrt.2008.06.003](https://doi.org/10.1016/j.jqsrt.2008.06.003).

[17] J. R. Ellis, C. D. Hall, Model development and code verification for simulation of electrodynamic tether system, *Journal of Guidance, Control, and Dynamics* 32 (6) (2009) 1713–1722. [doi:10.2514/1.44638](https://doi.org/10.2514/1.44638).

[18] R. G. Marchand, The method of manufactured solutions for the verification of computational electromagnetic codes, phdthesis, Stellenbosch (Mar. 2013).

[19] C. J. Roy, M. A. McWherter-Payne, W. L. Oberkampf, Verification and validation for laminar hypersonic flowfields, part 1: Verification, *AIAA Journal* 41 (10) (2003) 1934–1943. [doi:10.2514/2.1909](https://doi.org/10.2514/2.1909).

[20] R. Gollan, P. Jacobs, About the formulation, verification and validation of the hypersonic flow solver Eilmer, *International Journal for Numerical Methods in Fluids* 73 (1) (2013) 19–57. [doi:10.1002/fld.3790](https://doi.org/10.1002/fld.3790).

[21] C. R. Wilke, A viscosity equation for gas mixtures, *The Journal of Chemical Physics* 18 (4) (1950) 517–519. [doi:10.1063/1.1747673](https://doi.org/10.1063/1.1747673).

[22] L. Landau, E. Teller, Zur theorie der schalldispersion, *Physikalische Zeitschrift der Sowjetunion* 10 (34) (1936).

[23] R. C. Millikan, D. R. White, Systematics of vibrational relaxation, *Journal of Chemical Physics* 39 (12) (1963) 3209–3213. [doi:10.1063/1.1734182](https://doi.org/10.1063/1.1734182).

[24] C. Park, Review of chemical-kinetic problems of future NASA missions, I: Earth entries, *Journal of Thermophysics and Heat Transfer* 7 (3) (1993) 385–398. [doi:10.2514/3.431](https://doi.org/10.2514/3.431).

[25] C. Park, J. T. Howe, R. L. Jaffe, G. V. Candler, Review of chemical-kinetic problems of future NASA missions, II: Mars entries, *Journal of Thermophysics and Heat Transfer* 8 (1) (1994) 9–23. doi:10.2514/3.496.

[26] C. Park, *Nonequilibrium Hypersonic Aerodynamics*, John Wiley & Sons, Inc., 1990.

[27] S. F. Gimelshein, I. J. Wysong, Impact of the ionization reaction set in nonequilibrium hypersonic air flows, *AIAA Journal* 58 (3) (2019) 1255–1265. doi:10.2514/1.j058895.

[28] S. F. Gimelshein, I. J. Wysong, Applicability of 5, 7, and 11 species air models in nonequilibrium hypersonic reacting flows, *AIAA SciTech Forum*, American Institute of Aeronautics and Astronautics, 2020. doi:10.2514/6.2020-2190.

[29] J. H. Ferziger, M. Perić, *Computational Methods for Fluid Dynamics*, Springer, 2002. doi:10.1007/978-3-642-56026-2.

[30] M. Howard, A. Bradley, S. W. Bova, J. Overfelt, R. Wagnild, D. Dinzl, M. Hoemmen, A. Klinvex, Towards performance portability in a compressible CFD code, in: *23rd AIAA Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, 2017. doi:10.2514/6.2017-4407.

[31] B. Carnes, V. G. Weirs, T. Smith, Code verification and numerical error estimation for use in model validation of laminar, hypersonic double-cone flows, in: *AIAA SciTech 2019 Forum*, American Institute of Aeronautics and Astronautics, 2019. doi:10.2514/6.2019-2175.

[32] S. L. Kieweg, J. Ray, V. G. Weirs, B. Carnes, D. Dinzl, B. Freno, M. Howard, E. Phipps, W. Rider, T. Smith, Validation assessment of hypersonic double-cone flow simulations using uncertainty quantification, sensitivity analysis, and validation metrics, in: *AIAA SciTech 2019 Forum*, American Institute of Aeronautics and Astronautics, 2019. doi:10.2514/6.2019-2278.

[33] J. Ray, S. L. Kieweg, D. Dinzl, B. Carnes, V. G. Weirs, B. Freno, M. Howard, T. Smith, I. Nompelis, G. V. Candler, Estimation of inflow uncertainties in laminar hypersonic double-cone experiments, in: *AIAA SciTech 2019 Forum*, American Institute of Aeronautics and Astronautics, 2019. doi:10.2514/6.2019-2279.

[34] E. Phipps, R. Pawłowski, Efficient expression templates for operator overloading-based automatic differentiation, in: *Recent Advances in Algorithmic Differentiation*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 309–319. doi:10.1007/978-3-642-30023-3_28.

[35] R. A. Bartlett, D. M. Gay, E. T. Phipps, Automatic differentiation of C++ codes for large-scale scientific computing, in: *Computational Science – ICCS 2006*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 525–532. doi:10.1007/11758549_73.

[36] M. Aftosmis, D. Gaitonde, T. S. Tavares, Behavior of linear reconstruction techniques on unstructured meshes, *AIAA Journal* 33 (11) (1995) 2038–2049. doi:10.2514/3.12945.

[37] H. Luo, J. Baum, R. Lohner, An improved finite volume scheme for compressible flows on unstructured grids, in: *33rd AIAA Aerospace Sciences Meeting and Exhibit*, American Institute of Aeronautics and Astronautics, 1995. doi:10.2514/6.1995-348.

[38] L. Krivodonova, M. Berger, High-order accurate implementation of solid wall boundary conditions in curved geometries, *Journal of Computational Physics* 211 (2) (2006) 492–512. doi:10.1016/j.jcp.2005.05.029.

[39] G. V. Candler, The computation of weakly ionized hypersonic flows in thermo-chemical nonequilibrium, Ph.D. thesis, Stanford University (1988).