Chemical Equilibrium Analysis with Adjoint Derivatives for Propulsion Cycle Analysis

Justin Gray[1] and Jeffrey Chin[2] and Tristan Hearn[3] and Eric Hendricks[2] and Thomas Lavelle[2]
NASA Glenn Research Center, Cleveland, OH, 44139

Joaquim R. R. A. Martins [4]
University of Michigan, Ann Arbor, MI, 48109

The design optimization of aircraft engines considering their integration with the airframe has been limited by challenges with existing propulsion modeling tools. Gradient-based optimization with derivatives computed using adjoint methods has been successful in solving aerodynamic and structural shape optimization problems but has not yet been applied to coupled propulsion airframe optimization partly because existing tools lack analytic derivative computation. As a step towards obtaining a full cycle analysis with efficient analytic derivative computation, we develop a chemical equilibrium thermodynamics analysis for propulsion applications. This method provides a continuous formulation that enables computing analytic derivatives using a coupled adjoint approach. We present the method formulation and verify analysis results against the well-established Chemical Equilibrium and Applications code. We also verify the analytic derivatives by comparing them with finite-difference approximations. The performance of the analytic derivative computations is tested using two optimizations maximize the combustion temperature with respect to equivalence ratio and air pressure. The results show a clear speed and numerical stability benefit when comparing the proposed method against finite-difference approximations. It is

---

[1] Aerospace Engineer PSA Branch, 21000 Brookpark Rd., MS 5-11; Doctoral Candidate Department of Aerospace Engineering, University of Michigan; AIAA Member
[2] Aerospace Engineer PSA Branch, 21000 Brookpark Rd., MS 5-11; AIAA Member
[3] Aerospace Engineer PSA Branch, 21000 Brookpark Rd., MS 5-11
[4] Professor, Department of Aerospace Engineering; AIAA Associate Fellow

now possible to use this new method as the foundation for further development of a complete propulsion analysis for integrated propulsion-airframe design optimization.

I. Introduction

New aircraft concepts with a high degree of propulsion-airframe integration have been proposed to dramatically reduce fuel burn, emissions, and noise. These concepts, such as over-wing nacelles [1, 2] and engine installations designed for boundary-layer ingestion (BLI) [3–6], couple the thermodynamic performance of the propulsion system with the aerodynamic performance of the airframe. This requires coupling the aerodynamic and propulsion-analysis tools for aircraft design and motivates the application of multidisciplinary design analysis and optimization (MDAO) to navigate the potentially large design space [7]. Drela [8] underscores the importance of capturing this coupling in the D8 aircraft concept, and Welstead and Felder [9] make the same point for the STARC-ABL BLI concept.

Applying MDAO requires a set of analysis tools well suited to optimization. This means that the tools must be numerically stable, be capable of returning physically meaningful results even when starting with poor designs, and be computationally efficient. In cases where gradient-based optimization is used, it is highly desirable that the analysis tools can efficiently and accurately compute derivatives, and that the formulation of the analysis lead to smooth and differentiable functions. Computing the derivatives of a function of interest is always possible by using finite-difference approximations. However, such an approach is not efficient because the cost of computing the complete gradient is proportional to the number of variables. In addition, this method is subject to errors due to subtractive cancellation [10]. Analytic methods, where the analysis is linearized, are much more accurate and generally more efficient [11]. Two main forms of analytic methods exist: the direct method and the adjoint method. The computational cost of the direct method is proportional to the number of design variables and independent of the number of functions of interest, whereas the cost of the adjoint method is proportional to the number of functions of interest and independent of the number of design variables.

Applying MDAO to propulsion airframe integration problems requires coupling viscous computational fluid dynamics (CFD) to propulsion analysis. When using CFD-based shape optimization, the most common technique is to use gradient-based optimization with the gradients computed via the adjoint method. The reason for this choice is is that there are few functions of interest (usually drag, lift, and moment coefficients), but a large number of design variables are required to parametrize three-dimensional shapes. Many CFD tools that compute adjoint derivatives are already available, such as FUN3D [12], SU2 [13], elsA [14], and ADflow [15–18]. The adjoint method, in conjunction with gradient-based optimization, has been used in many studies of aerodynamic-shape optimization, as demonstrated by the series of benchmarks developed by the AIAA Aerodynamic Design Optimization Discussion Group [14, 17, 19–21]. However, propulsion analysis tools are not as well developed in this regard, and many existing applications still rely on gradient-free methods, which cannot handle large numbers of design variables.

To date, gradient-free methods have been successful because the number of design variables required to formulate the problem have been kept relatively small, but the integration of propulsion analysis with viscous CFD requires a large number of design variables, which makes gradient-based optimization a requirement [22]. Sandhi et al. [23] identified many different options in their survey of propulsion modeling tools, but none compute analytic derivatives. Of the tools identified, Numerical Propulsion System Simulation (NPSS) [24] is the most recently developed and is widely used. Because NPSS does not compute analytic derivatives, the finite-difference method is used to approximate derivatives. The finite-difference derivatives from NPSS could, in theory, be combined with the adjoint derivatives from the CFD tool to form a semi-analytic coupled derivative that would enable gradient-based optimization.

Although the semi-analytic approach would allow integrated propulsion-airframe optimization with existing tools, such a solution is far from ideal. Finite-difference derivative approximations from NPSS result in numerical issues. They offer limited accuracy, which can require the optimization to go through many more iterations and reduce the efficiency of finding a true optimum [25]. Numerical stability issues also exist when finite-differencing NPSS to compute derivatives, which Geiselhart [26] identifies as a major motivation for applying gradient-free methods in his low-boom design optimization of a supersonic business jet. Similarly, Allison et al. [27–29] also noted problems with stability and convergence in their extensive work on integrating NPSS into the conceptual design for military aircraft. These problems require moving away from finite-difference derivative approximations toward a new propulsion analysis tool based on more accurate analytic derivatives.

Future work will first use this propulsion analysis tool to apply gradient-based optimization to propulsion-only models to minimize specific fuel consumption or emissions, and later will use coupled propulsion airframe models to minimize mission fuel burn or total energy consumption. Before a new propulsion analysis tool can be built, we must develop a core thermodynamics model capable of providing analytic derivatives. The thermodynamics model computes all properties of the working fluid (enthalpy, entropy, temperature, pressure, density, etc.) given any two thermodynamic states and is a fundamental building block in a full propulsion analysis framework. This model must provide analytic derivatives of computed gas properties with respect to the prescribed properties (e.g., $\partial\gamma/\partial T$, $\partial S/\partial P$), which are then used to compute the larger derivative for the full propulsion model.

The core thermodynamics model is implemented in the OpenMDAO framework, which uses a Newton solver to converge the nonlinear chemical-equilibrium equations and facilitates the computation of both the derivatives for this Newton solver and the derivatives required for gradient-based optimization via analytic methods (both direct and adjoint) [11, 30–32].

The rest of the paper is organized as follows: Section II reviews the available thermodynamics methods and justifies the use of the chemical-equilibrium-based method. Section III summarizes the Gibbs energy equilibrium equations and the numerical solver used to converge the chemical equilibrium equations. The details of the coupled-derivative implementation are given in Section IV, and Section V verifies the results of the analysis and computations of the derivatives. Finally, Section VI presents the optimization results for maximum combustion temperature; the results demonstrate the speed and improvements in numerical stability achieved by using analytic derivatives.

II. Existing Thermodynamics Tools

A number of different methods exist for computing the thermodynamic properties of air and air-fuel mixtures, and many implementations of these methods are available. NPSS provides a set of thermodynamics libraries (CEA, JANAF, ALLFUEL, GasTbl), which the user can select at runtime [24]. ALLFUEL and GasTbl are computationally efficient but are based on interpolated tabular data and are only accurate for Jet-A fuels or other fuels that are chemically very similar to Jet-A fuel. Tran and Snyder [33] demonstrated that tabular thermodynamics data offer limited accuracy when considering fuels that are chemically dissimilar from those used to generate the data. They recommend either generating new tabular data when switching fuels or integrating a chemical-equilibrium analysis directly into the propulsion code. For unconventional cycle configurations that use multiple fuels (for example, combining Jet-A fuel with liquid natural gas [34]), tabular thermodynamics data are not suitable.

A more advanced technique for building a Kriging surrogate model of the equilibrium gas composition was proposed by Walter and Owen [35] and offers a more flexible option for an interpolation-based method. However, this technique does not provide a means of computing the thermodynamic properties of the composite gas. The CEA and JANAF libraries use a chemical-equilibrium-based method, but the JANAF library only considers a fixed set of species. The CEA [36] library is by far the most general of the options in NPSS, accepting an arbitrary number of chemical species that enables it to consider a wide range of fuels.

There are also a number of chemical-equilibrium libraries developed for fields outside of cycle analysis. MINEQL+ [37] specializes in equilibrium reactions in aqueous solutions. ChemSage [38] supports reactions that include metallic elements and metal oxides. JANAF, CEA, MINEQL+, and ChemSage all solve for chemical equilibrium directly by minimizing the Gibbs energy. These codes differ primarily in the details of how they converge the nonlinear chemical-equilibrium equations and also offer specialized features for their specific applications. Another alternative is the nonstoichiometric equilibrium method [39], which is used by the open-source library Cantera [40]. Chin et al. [41] used the Cantera implementation for an application involving cycle analysis.

For each propulsion analysis, the thermodynamics module is called hundreds of times, so computational efficiency is important. However, flexibility to consider a wide range of fuels is also important to allow alternative fuels to be investigated, such as hydrogen [42], natural gas [34, 43], and bio-fuels. The low flexibility of interpolative methods makes them less desirable, despite their simplicity. Chemical-equilibrium methods provide the needed flexibility. The Cantera library was ruled out due to our prior experience using it in this application where it led to problems regarding performance. We considered the option of differentiating one of the other chemical-equilibrium libraries, but none implement a continuous solution algorithm that is differentiable. The CEA and JANAF libraries use a method for handling trace species—those with negligible concentrations—where a heuristic controls their addition and removal from the equilibrium calculations, causing small discontinuities in the solution. These discontinuities are problematic from an analysis point of view [44] and create points where derivatives are not defined. Therefore, we developed a new algorithm for solving the chemical-equilibrium analysis that is differentiable and thus suitable for gradient-based optimization.

III. Chemical Equilibrium Equations

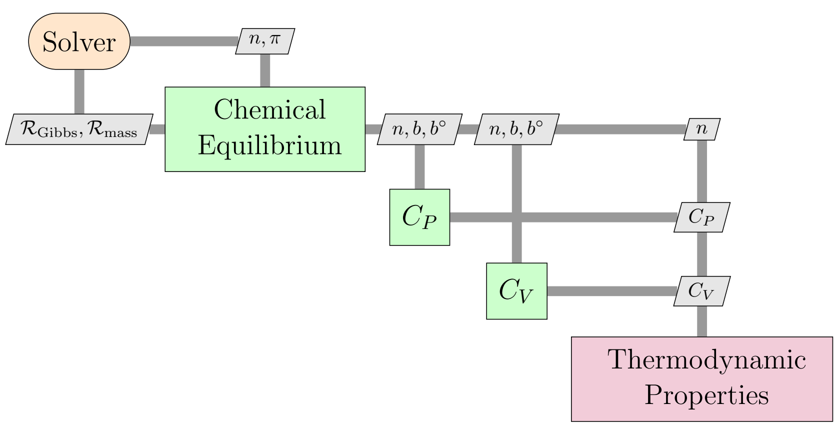

Fig. 1: XDSM diagram of the chemical equilibrium analysis sequence.

A. Thermodynamic Properties Prediction

To find the thermodynamic state of a gas, CEA uses the multistep process illustrated in Figure 1 as an extended design structure matrix (XDSM) diagram [45]. First, it solves a system of nonlinear equations to minimize the Gibbs energy, which gives the equilibrium composition for the gas at the prescribed state. Second, it solves an additional set of equations to compute the thermodynamic state of the gas by using the converged equilibrium composition. These equations are documented in detail by Gordon and McBride [36] in their seminal paper on CEA. However, the form of the equations presented in their work combines the residuals from the Gibbs-energy minimization with a customized procedure for solving for the Newton update. This combination was motivated by the limitations imposed at the time by computer hardware. However, with modern computers and more powerful linear algebra libraries, it is much better to keep the residuals separate from the numerics to simplify the task of differentiating the analysis. The purely physical form of the equations is presented here. We refer the reader to the original CEA publication for details on the second step where the thermodynamic properties are computed from the converged equilibrium solution; it remains unchanged from the original formulation [36].

B. Gibbs Energy Minimization

The full thermodynamic state of a real gas can be defined by any two of the five physical state variables: temperature T, pressure P, density ρ, entropy S, and enthalpy h. Three specific combinations are useful for propulsion analysis: temperature and pressure (TP), enthalpy and pressure (hP), and entropy and pressure (SP). The TP formulation is the most fundamental. The hP and SP formulations augment the TP-formulation equations with a new state variable T and the associated residual to drive the solution to the prescribed value of h or S.

1. Temperature-Pressure Formulation

The Gibbs energy g is defined as

$$g = \sum_{j}^{N_s} (\mu_j n_j) \,, \tag{1}$$

where N_s is the number of chemical species, and n_j and μ_j are the concentration (kg-mol/kg-mixture) and the chemical potential of the j^{th} species, respectively. The chemical potential is a function of temperature, pressure, and concentration and is given by

$$\frac{\mu_j}{RT} = \frac{H_j^\circ(T)}{RT} - \frac{S_j^\circ(T)}{RT} + \ln\left(\frac{P}{P_a}\right) + \ln(n_j) - \ln\left(\sum_k^{N_s} n_k\right), \tag{2}$$

where R is the universal gas constant, and H° and S_j° are given by

$$\frac{H_j^\circ(T)}{RT} = \frac{-c_0}{T^2} + \frac{c_1 \ln(T)}{T} + c_2 + \frac{c_3}{2}T + \frac{c_4}{3}T^2 + \frac{c_5}{4}T^3 + \frac{c_6}{5}T^4 + \frac{c_7}{T}, \tag{3}$$

$$\frac{S_j^\circ(T)}{R} = \frac{-c_0}{2T^2} - \frac{c_1}{T} + c_2 \ln(T) + c_3 T + \frac{c_4}{2}T^2 + \frac{c_5}{3}T^3 + \frac{c_6}{4}T^4 + c_8, \tag{4}$$

where $c_0, \ldots, c_8$ are constants for each species, taken from the NIST-JANAF Thermochemical Tables [46].

The equilibrium composition is defined by the values of the concentration variables n that minimize the Gibbs energy subject to conservation-of-mass constraints. The mass is tracked on an elemental basis and is given by

$$\mathcal{R}_{\text{mass } i} = \sum_{j=1}^{N_s} (a_{ij} n_j) - b_i^\circ = 0 \tag{5}$$

where N_s is the number of species in the mixture, b_i° is the amount of each element in the initial composition, and a_{ij} is the stoichiometric constant for element i of species j. The mass-balance residuals $\mathcal{R}_{\text{mass } i}$ yield N_e constraints—one per element present in the mixture. To solve the mass-constrained minimization of the Gibbs energy, we form a Lagrangian

$$G = \sum_{j=1}^{N_s} (\mu_j n_j) + \sum_{i=1}^{N_e} \lambda_i \left(\sum_{j=1}^{N_s} (a_{ij} n_j) - b_i^\circ\right), \tag{6}$$

where λ_i is the Lagrange multiplier for the i^{th} element. We differentiate with respect to the n and λ variables to build a system of nonlinear equations whose solution minimizes the Lagrangian:

$$\delta G = \sum_{j=1}^{N_s} \left(\mu_j + \sum_{i=1}^{N_e} (\lambda_i a_{ij})\right) \delta n_j + \sum_{i=1}^{N_e} \left(\sum_{j=1}^{N_s} (a_{ij} n_j) - b_i^\circ\right) \delta \lambda_i = 0. \tag{7}$$

Because δG is linear in δn_j and $\delta \lambda_i$, we can split Equation (7) into two sets of equations. We get

N_s equations,

$$\mathcal{R}_{\text{Gibbs}} = \mu_j + \sum_{i=1}^{N_e} (\lambda_i a_{ij}) = 0, \tag{8}$$

as residuals representing the Gibbs energy. For convenience, because of the factors of $1/RT$ in Equations (2) and (4), we define an alternative Lagrange multiplier as follows:

$$\pi_i = -\frac{\lambda_i}{RT}. \tag{9}$$

This yields an alternate form of Equation (8):

$$\mathcal{R}_{\text{Gibbs}} = \frac{\mu_j}{RT} - \sum_{i=1}^{N_e} (\pi_i a_{ij}) = 0. \tag{10}$$

Equations (5) and (10) yield a system with $N_s + N_e$ unknowns and $N_s + N_e$ residual equations, which can then be solved numerically.

2. Enthalpy-Pressure Formulation

The hP formulation retains the state variables (n and π) from the TP solver and the associated residuals from Equations (5) and (10). In addition, it adds a new state variable T and a new residual to drive the computed enthalpy to match the the specified enthalpy value h_0:

$$\mathcal{R}_h = h_0 - \sum_{j=1}^{N_s} \big()n_j H_j^\circ(T)\big) = 0. \tag{11}$$

3. Entropy-Pressure Formulation

Like the hP problem, the SP formulation uses the same setup as the TP problem with one additional state variable and residual. In this case, the new state variable is the prescribed entropy, S_0. The new residual drives the computed entropy to match the prescribed entropy:

$$\mathcal{R}_S = S_0 - R\sum_{j=1}^{N_s} \left\{ n_j \left(\frac{S_j^\circ(T)}{R} - \ln\left(\frac{P}{P_a}\right) - \ln(n_j) + \ln\left(\sum_k^{N_s} n_k\right)\right)\right\} = 0, \tag{12}$$

where the pressure term is nondimensionalized by standard atmospheric pressure ($P_a = 1.01325$ Bar). The reference condition is necessary because entropy is defined as a variation from a reference condition.

C. CEA Modified Newton's Method for Chemical Equilibrium

1. Newton Convergence Scheme

Gordon and McBride [36] applied Newton's method to converge the chemical-equilibrium system. When applied to Gibbs-energy minimization, Newton's method consists of successive solutions of the linear system

$$\frac{\partial \mathcal{R}}{\partial U}\Delta U = -\mathcal{R}(U), \tag{13}$$

where $U = [n, \pi]$ for a TP problem, and $U = [n, \pi, T]$ for hP and SP problems. ΔU is iteratively computed and applied until the residuals (5), (10)–(12) converge to zero within a chosen tolerance.

Note that Equation (10) involves computing μ_j, which, through Equation (2), requires taking the natural logarithm of n. In addition, Equation (4) involves taking the natural logarithm of T. Thus, neither n nor T can be negative during the iterations. The natural logarithms also cause numerical difficulties, because the derivative with respect to n tends to infinity as n tends to zero. In addition, this means that the Newton system becomes ill conditioned as n tends toward zero. In CEA this problem is partially dealt with via a logarithmic transformation, where ΔU is split into two parts: one for $[n, T]$ and another for π. The $[n, T]$ component of the updates are treated as $\Delta \ln(n)/n$ and $\Delta \ln(T)/T$. The Newton update equation is modified to account for this as follows:

$$n_{k+1} = n_k \exp\left(\frac{\Delta \ln(n)}{n}\right), \tag{14}$$

$$T_{k+1} = T_k \exp\left(\frac{\Delta \ln(T)}{T}\right). \tag{15}$$

By using the exponential update form, negative values from the Newton solution are converted into multiplicative updates that are always positive, so, assuming a positive initial guess, the values of n

and T never become negative. The π update variables are treated normally, with a Newton update

$$\pi_{k+1} = \pi_k + \Delta\pi. \tag{16}$$

Although the CEA method deals with the need to keep both n and T positive, it does not solve the problem whereby trace species create poorly conditioned Jacobians. For this problem, species with n lower than a set value (10^{-5} by default) are discarded from the solution. Removal of species, even trace species, introduces a nondifferentiable discontinuity that needs to be avoided. Leal et al. [44] propose a method that retains all species and modifies the computed Newton step for trace species to keep them positive. Because this method retains all species, it is continuous and thus differentiable. We adopt this method for this work and use a step-limiting method similar to that used by the built-in solvers in OpenMDAO.

2. Computing $\partial\mathcal{R}/\partial U$

We compute $\partial\mathcal{R}/\partial U$ analytically, where the nonzero elements for the TP residual partial derivatives are given by

$$\frac{\partial \mathcal{R}_{\text{Gibbs j}}}{\partial n_k} = \begin{cases} \frac{-1}{\sum_l^{N_s} n_l} & \text{if } j \neq k \\ \frac{1}{n_k} - \frac{-1}{\sum_l^{N_s} n_l} & \text{if } j = k, \end{cases} \tag{17}$$

$$\frac{\partial \mathcal{R}_{\text{Gibbs j}}}{\partial \pi_i} = -a_{ij}, \tag{18}$$

$$\frac{\partial \mathcal{R}_{\text{mass } i}}{\partial n_j} = a_{ij}. \tag{19}$$

When solving an hP or SP problem, additional nonzero partial derivatives of Equation (8) with respect to T are given by

$$\frac{\partial \mathcal{R}_{\text{Gibbs j}}}{\partial T} = \frac{\partial H_j^\circ}{\partial T} - \frac{\partial S_j^\circ}{\partial T}. \tag{20}$$

For an hP problem, the residual (11) contributes the following nonzero partial derivatives:

$$\frac{\partial \mathcal{R}_h}{\partial n_j} = -RTH_j^\circ, \tag{21}$$

$$\frac{\partial \mathcal{R}_h}{\partial T} = -RT\sum_{j=1}^{N_s} n_j \left(\frac{\partial H_j^\circ}{\partial T} + H_j^\circ\right). \tag{22}$$

Similarly, for an SP problem the residual (12) contributes the following nonzero partial derivatives:

$$\frac{\partial \mathcal{R}_S}{\partial n_j} = -R\left[\frac{S_j^\circ(T)}{R} - \ln\left(\frac{P}{P_a}\right) - \ln(n_j) + \ln\left(\sum_k^{N_s} n_k\right) - 1\right], \tag{23}$$

$$\frac{\partial \mathcal{R}_S}{\partial T} = -R\sum_{j=1}^{N_s}\left(n_j\frac{\partial H_j^\circ}{\partial T}\right). \tag{24}$$

All these nonzero terms can be assembled into a matrix, which is then inverted by using a direct method because the size of the matrix is at most $(N_s + N_e + 1)(N_s + N_e + 1)$, where the largest term N_s is on the order of hundreds of species.

IV. Multidisciplinary Derivatives

We define *multidisciplinary derivatives* as the total derivative of a function (objective or constraint) with respect to the design variables of a problem where the multidisciplinary system is converged. In other contexts, these could also be called total derivatives or coupled derivatives [11]. This work involves a single engineering discipline: thermodynamics. However, the model is built up of multiple components, as seen in Figure 1, and each component can be thought of as a subdiscipline. In that sense, any derivative of an OpenMDAO model is a multidisciplinary derivative. OpenMDAO automatically computes the multidisciplinary derivatives of an arbitrary model, assuming that each component in the model provides its own partial derivatives [47, 48].

Partial derivatives are computed for a single component (i.e., derivatives of component outputs with respect to its own input variables); they must be provided to the framework for each component. Partial derivatives can be computed analytically or via automatic differentiation [11, 49]. In this work, analytic expressions are derived by hand for all components. The derivations are straightforward because of the fine-grained breakdown of the analysis into multiple components, where each component consists of a limited amount of code. This style of building up an analysis via a combination of small components is encouraged when using the OpenMDAO framework, since it facilitates differentiation.

Computing the required partial derivatives for a given output is done in a manner that is almost identical to the computation of the partial derivatives for the Newton solver. The only difference is that, for the Newton solver, we compute only the partial derivatives of the residual equations with respect to the state variables. To compute the multidisciplinary derivatives, we reuse the derivatives from the Newton solver but now also include partial derivatives with respect to all other inputs of that component. As an example, consider Equation (12). The two partial derivatives needed for the Newton solver are given by Equations (23) and (24). These two derivatives are augmented with the two following additional derivatives:

$$\frac{\partial \mathcal{R}_s}{\partial S_0} = 1, \tag{25}$$

$$\frac{\partial \mathcal{R}_s}{\partial P} = \frac{R}{P}\sum_{j=1}^{N_s} n_j. \tag{26}$$

Once all the partial derivatives are computed, OpenMDAO automatically assembles them into a linear system that, by using either a coupled direct or a coupled adjoint method [50], is solved to compute the multidisciplinary derivatives. In OpenMDAO, this is achieved via the unified derivative equation [11]:

$$\frac{\partial \mathcal{R}}{\partial U}\frac{du}{dr} = \mathcal{I} = \frac{\partial \mathcal{R}}{\partial U}^T \frac{du}{dr}^T, \tag{27}$$

where $\partial\mathcal{R}/\partial U$ is a Jacobian matrix of partial derivatives, $\mathcal{I}$ is the identity matrix, and du/dr is the matrix of total derivatives for which we want to solve. The left-hand side of Equation (27) represents the coupled direct method—solved one time per design variable—whereas the right-hand side represents the coupled adjoint method—solved one time per function of interest. The coupled adjoint capability is the most significant, given the ultimate goal of integration with adjoint CFD codes, but once the partial derivatives are given, both direct and adjoint solvers are available from

OpenMDAO with no additional work. In Equation (27) the Jacobian $\partial\mathcal{R}/\partial U$ is similar to that used in the Newton solver from Equation (13), but it includes the additional partial derivatives with respect to all the inputs—represented by Equations (25) and (26) in our case.

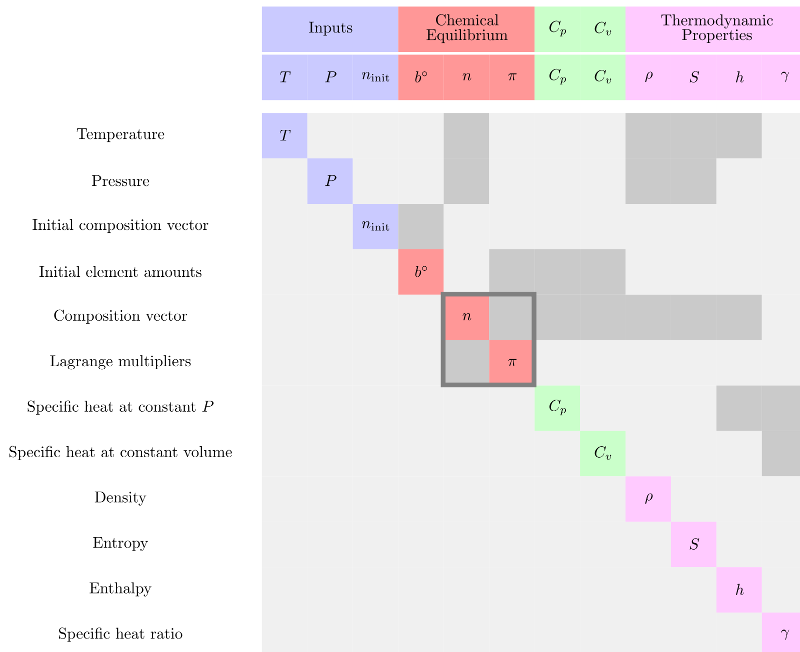

Fig. 2: Structure of partial derivative matrix for adjoint multidisciplinary derivatives of TP solver. The subset of partial derivatives needed for the Newton solver is highlighted by the gray box.

For the chemical-equilibrium model developed herein, the structure of $\partial\mathcal{R}/\partial U^T$ is shown in Figure 2 for a TP solver and in Figure 3 for the hP or SP solver. The diagonal terms are the partial derivatives of the residuals with respect to the associated state variable or an output with respect to itself. The off-diagonal terms are partial derivatives of residuals or outputs with respect to the other variables. In Figure 2, the 2×2 block of partial derivatives, outlined in dark gray, are the same derivatives needed for the Newton solver. These partial derivatives are given by Equations (17)–(19). For the TP solver, temperature is an input to the calculation, but, for the hP and SP solvers, it becomes a state variable. For these slightly more complex solvers, the $\partial\mathcal{R}/\partial U$ matrix includes h or S as inputs and T as an additional state variable. The 3×3 block, outlined in dark gray in Figure 3, contains the Newton derivatives for the corresponding solvers. All other shaded boxes outside the Newton blocks are the additional partial derivatives needed to compute the multidisciplinary derivatives. It is convenient that the partial derivatives necessary for a Newton solver can be reused with the unified derivative equations, but Figures 2 and 3 also show that significantly more partial derivatives may be required in the latter case.

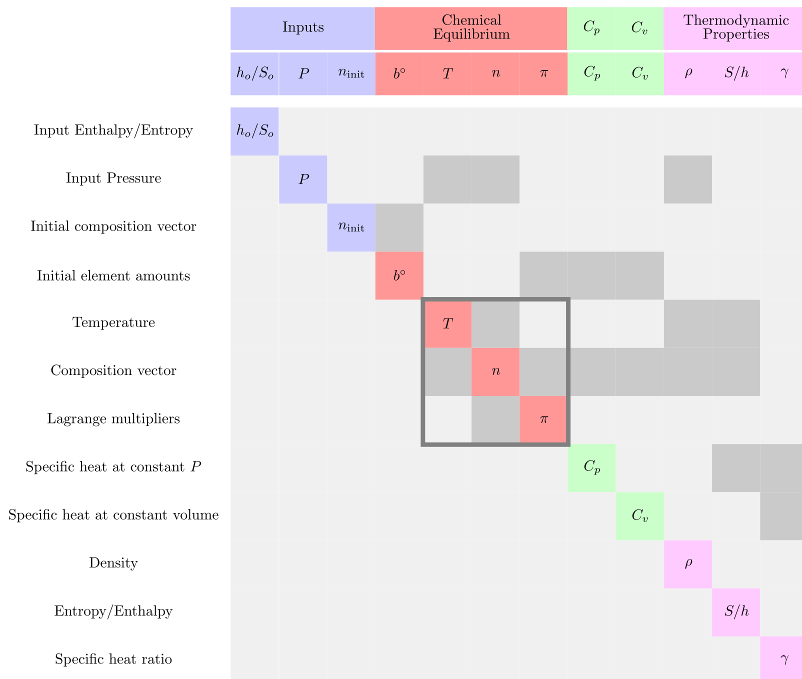

Fig. 3: Structure of partial derivative matrix for adjoint multidisciplinary derivatives of hP and SP solvers. The subset of partial derivatives needed for the Newton solver are highlighted by the gray box.

Although Equation (27) is just a linear system, solving it efficiently for a chemical equilibrium model is not easy. The lower diagonal elements of the matrix prevent the use of Gaussian elimination. A naive implementation could use a direct method to solve the entire linear system monolithically. Although this would work for an isolated chemical-equilibrium analysis, a more complex propulsion model with many such solutions would become prohibitively large for that approach. The modular analysis and unified derivatives (MAUD) architecture for solving the derivative equations suggests a more efficient approach that uses a nested linear solver that follows the hierarchy of the nonlinear model to generate effective preconditioners for use with sparse iterative linear solvers such as GMRES [48].

OpenMDAO implements the MAUD solver architecture by allowing each component to specify its own nonlinear solver and linear solver. Figure 4 illustrates the model hierarchy and shows which linear and nonlinear solvers are used for each component. At the top level, one iteration through each subcomponent is used for the nonlinear solver, because there is no intercomponent coupling in this model. The sequential nature of the nonlinear solver enables a Gaussian elimination at the top level, implemented by one-block linear Gauss–Seidel (LNGS) iteration for the linear solution. The *Inputs* and *Thermodynamic Properties* components are both a collection of explicit equations that can also use a LNGS algorithm with a single iteration for the linear solution. The *Chemical Equilibrium* component uses a Newton solver for the nonlinear solution and a GMRES linear solver. The C_P and C_V components are composed of linear equations that can be solved by using a direct method for both the nonlinear and the linear solvers—although this component is actually only linear. For *Chemical Equilibrium*, C_P, and C_V, a natural synergy exists between the nonlinear and linear solvers. For *Chemical Equilibrium*, the same linear solver needed to solve for the Newton update can be reused for the derivatives. For C_P and C_V, the exact same solver can be used for both because both are linear systems.

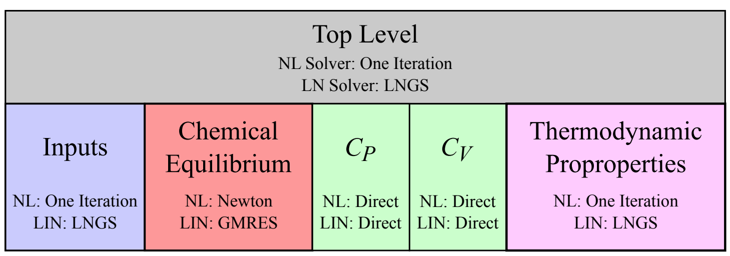

Fig. 4: Hierarchy for chemical-equilibrium solver with both nonlinear (NL) and linear (LIN) solvers indicated for each component.

V. Verification of Analysis and Derivatives

The thermodynamics module is to be the foundation for a new cycle analysis tool, so it is important that it accurately models the thermodynamic properties of air and air-fuel mixtures across a wide range of temperatures and pressures. Given the goal of using this work for optimization, it is also important that the analytic derivatives be correct. This section presents the verification of the analysis and the corresponding derivatives.

A. Approach to Analysis Verification

The new code was verified against CEA predictions. The verification cases correspond to the temperature and pressure combinations listed in Table 1. A total of 3600 different conditions were examined with temperatures ranging from 200 to 4800 degrees Rankine, and pressures from 1 to 1500 psi. This regular grid was run at four equivalence ratios ϕ of 0, 0.015, 0.3, and 0.44 to provide a wide range of combustion conditions. The equivalence ratio (the ratio of the actual fuel-to-air ratio to the stoichiometric value) is a convenient way to express the amount of potential combustion in a manner that is independent of the specific fuel being used. This verification grid includes low temperatures that are not physically meaningful and that actually extend below the valid range of the thermodynamics data provided as input. Such low temperatures were run not to test the physical predictive power of the code under invalid conditions, but rather to compare it numerically with CEA under extreme conditions. Because the new code is to be used for optimization, it must be numerically stable even under nonphysical conditions because optimizations often iterate through physically invalid areas on their way to the optimum.

Table 1: Temperature and pressure conditions used for verification cases.

	Low	High	Step
Temperature (°R)	200	4800	200
Pressure (psi)	1	1500	10

o compute the verification data, the new code was set up with air at the temperature and pressure conditions prescribed in Table 1 and then combusted at the given ϕ with Jet-A, which is a hydrocarbon fuel ($C_{12}H_{23}$) with a stoichiometric fuel-air ratio of 0.06817 corresponding to $\phi = 1$.

The combustion was modeled as a prescribed enthalpy process by computing the overall enthalpy of the air-fuel combination and holding it constant while solving for a chemical-equilibrium composition. Because the total enthalpy calculation is already required for the combustion model in the new code, the corresponding CEA runs are set up with h and P by using the enthalpy output from the corresponding case. For both codes, a reduced set of 19 chemical species is considered in order to reduce the number of trace species present in the converged equilibrium results and to improve computational speed. The following species are included: N, NH_3, N_2, NO, NO_2, NO_3,

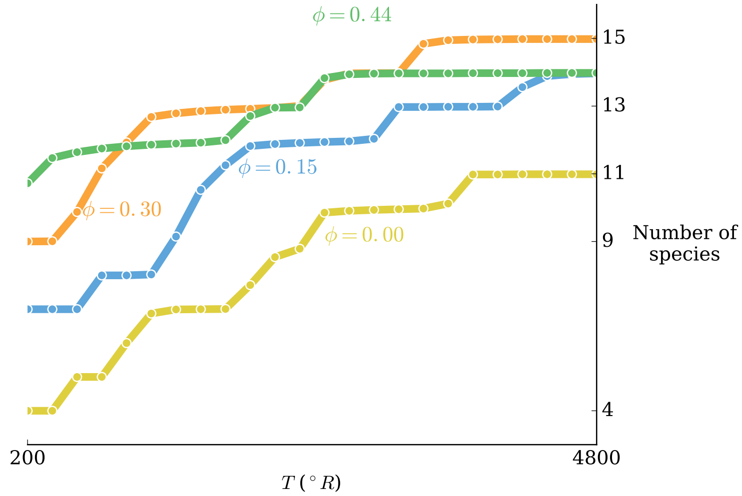

Fig. 5: Variation of total number of active chemical species in CEA solutions with respect to temperature.

CH_4, C_2H_4, CO, CO_2, O, OH, O_2, H, H_2, H_2O, HO_2, H_2O_2, Ar.

Two different types of verification are performed. First, the predicted chemical-equilibrium compositions are compared to ensure that the proper amount of each chemical species is present at each test point. Next, the actual thermodynamic properties (P, T, ρ, h, S, C_P, C_V, γ) are compared. To verify the composition, the average discrepancy between the codes is 5.2×10^{-6} moles and the maximum error is 0.00011 moles. The average discrepancy in the prediction of the thermodynamic properties is 0.03%, and the maximum error is 0.52%. These results demonstrate a strong agreement over a wide range of temperature and pressure.

One unavoidable source of discrepancy between the new code and CEA comes from implementation details related to handling trace species, as discussed in Section III C 1. Figure 5 quantifies this effect more clearly, showing the number of species retained in the final solution from CEA for different values of ϕ over a range of temperatures. The number of active species varies most for $\phi = 0$ (from 4 to 11) because, at lower temperatures, the composition of air stays relatively close to atmospheric, but NO_x starts to form and dissociation starts to occur at the higher temperatures. Note that the data in Figure 5 are pressure-averaged over the entire range of pressures from Table 1. A slight negative correlation exists between pressure and the number of active species in the data, but this is negligible compared with the temperature effect. If a larger set ofspecies were considered, the pressure effect could be more pronounced.

B. Verification of Chemical Equilibrium

The first verification serves to confirm that the new code returns the same chemical composition as does CEA over the verification grid from Table 1. The comparison is done on a per-species basis and is measured with absolute differences. Air, even when combusted with $\phi = 0.44$, is composed of over 70% diatomic nitrogen, which means that any other chemical species (e.g., CO_2, H_2O), although important to the thermodynamic properties, makes up less than 30% of the overall gas. Some species (e.g., NH_3, NO_3) are present only in trace amounts and have almost no impact on the thermodynamics. The accuracy of the amount of trace species predicted by using a chemical-equilibrium method is highly limited but is still of practical significance to ensure that the two codes predict the same trace species. The low-concentration of many species requires an absolute error to measure the discrepancy between codes, and, to be meaningful, absolute error requires knowledge of the actual value for n_j. Figure 6 shows the accuracy of the predicted values for n averaged over the full T and P verification set as a function of equivalence ratio. The mean n_j are shown as solid blue lines and can be compared with the absolute error $||n_{j\ \mathrm{CEA}} - n_j||_2$, which is represented as orange lines. Most of the errors are at least two orders of magnitude smaller than the mean concentrations, indicating strong agreement between the two codes. For NH_3 and NO_3 the discrepancies are the same order of magnitude as the mean value because CEA results includes these species at the lower limit of 10^{-10}. These two species further highlight the subtle difference between the trace-species method for CEA and the new code. At such small concentrations, these species do not have a meaningful impact on the thermodynamic properties. Note that CH_4 and C_2H_4 are excluded from Figure 6 because they are always trace species for CEA, and their compositions are always below 10^{-13} for the new code.

In addition to showing the discrepancy for each species individually, Figure 7 also shows how the norm of the absolute error, $||n_{\mathrm{CEA}} - n||_2$, varies with T and P for the case $\phi = 0.44$. Although small, there is a clear trend toward increasing discrepancy with higher temperature and a much weaker correlation with increasing pressure. Although the discrepancy grows from 3×10^{-5} to 2×10^{-4} upon moving from the lower-left corner to the upper-right corner, it is still small. This trend is consistent with Figure 5, where CEA starts introducing small concentrations of new species as the

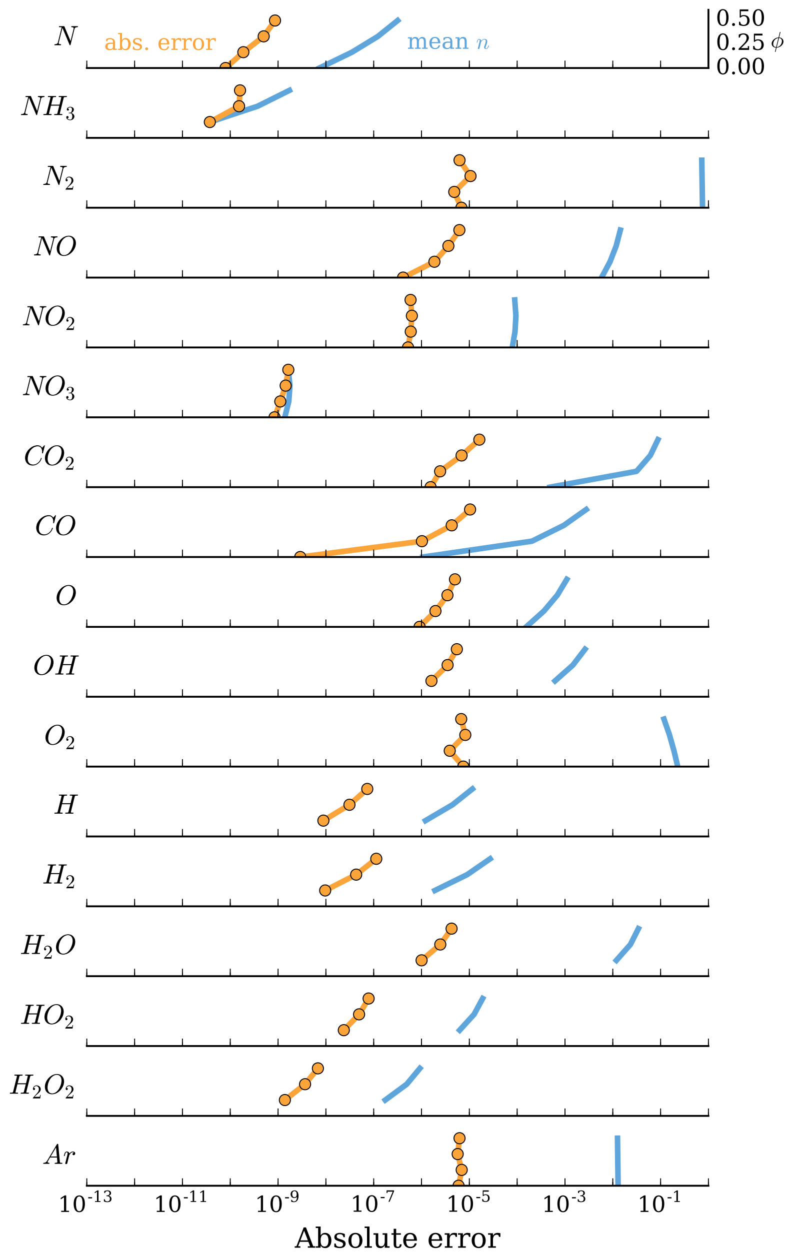

Fig. 6: Mean differences between equilibrium gas compositions predicted by CEA and by proposed code.

temperature rises. Because different methods are used to handle these trace species, we expect additional discrepancy in areas where trace species are prevalent. Figures 6 and 7 demonstrate that both codes give results that are in strong agreement in terms of overall composition across the entire verification grid. The overall conclusion is that the proposed code and CEA both compute the same composition, within solver tolerance.

From a cycle-analysis perspective, the composition vector n is not important in itself; however,

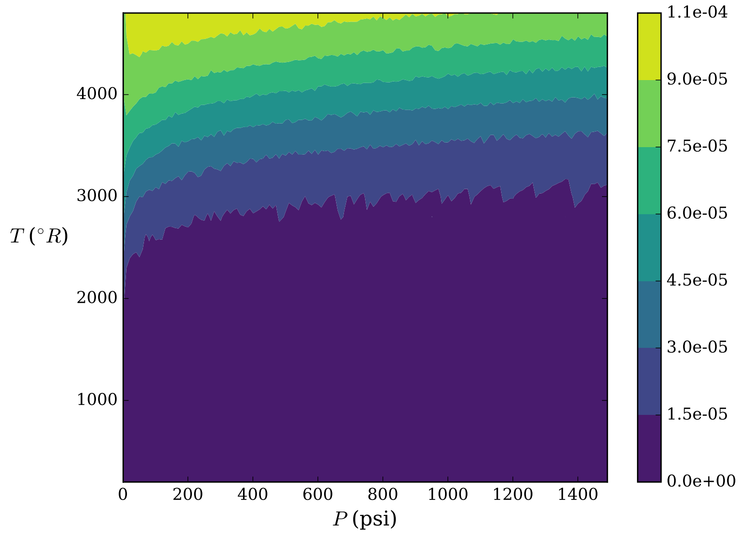

Fig. 7: l^2 norm of difference between concentration vectors, $||n_{\mathrm{CEA}} - n||_2$.

because the thermodynamic properties are computed as a function of n, it is indirectly important. Therefore, the verification of predicted composition serves as a preliminary verification of the thermodynamics itself. The next section presents the verification of the thermodynamic properties, but these data reinforce those results by ensuring that the calculations are fundamentally based on the same chemical compositions.

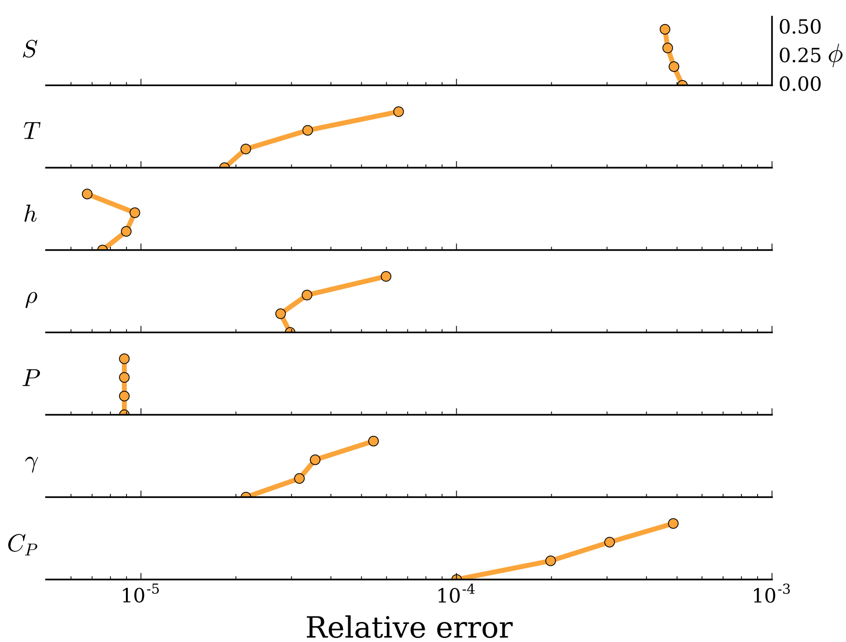

Fig. 8: Mean relative errors between the thermodynamic states predicted by CEA and by proposed code is less than 0.1%.

C. Verification of Thermodynamic Properties

We use relative measurements to compare the thermodynamic state variables predicted by CEA with those predicted by the new code. Figure 8 shows mean discrepancy, measured across the full verification grid, for entropy S, temperature T, enthalpy h, density ρ, pressure P, specific-heat ratio γ, and specific heat at constant pressure C_P. The mean error for all cases, across all properties, is 0.03%, and the maximum error is 0.52%. Note that, because both h and P are set directly for the CEA run, from the output of the corresponding cases with the new code, these properties have the lowest errors in Figure 8. The other errors are larger but are still less than 0.1%. This demonstrates a strong agreement between CEA and the new code and verifies its predictions for cycle-analysis applications.

D. Verification of Multidisciplinary Derivatives

The derivative accuracy is verified by comparing analytic derivatives (direct method in this case) to finite-difference approximations by using forward-, central-, and backward-difference methods. Figures 9 and 10 show the relative error between the three finite-difference derivative approximations and the analytic reference values. This relative error is plotted for varying finite-difference step sizes. The plots follow the structure of the Jacobian, showing the derivatives of h, S, ρ, C_P, and γ (rows) with respect to T and P (columns). Figure 9 corresponds to standard day conditions (288 K, 1 bar), whereas Figure 10 corresponds to a representative cruise condition (1500 K, 10 bar). The analytic and finite-difference derivatives all agree to at least 10^{-5}, although the point of best agreement occurs at different step sizes for different variables.

Whereas the results for standard day conditions in Figure 9 show good agreement for all step sizes, the results for the cruise condition in Figure 10 shows a dramatic increase in accuracy when the relative step-size is less than 10^{-5}. At cruise conditions, the temperature and pressure are higher, and the converged mixture contains many more active species, which makes the solution more sensitive to input values. Thus, keeping the step size sufficiently small to prevent major changes in the solution is the key to obtaining accurate finite-difference approximations. The central-difference approximation converges faster than the forward or backward schemes and offers better overall

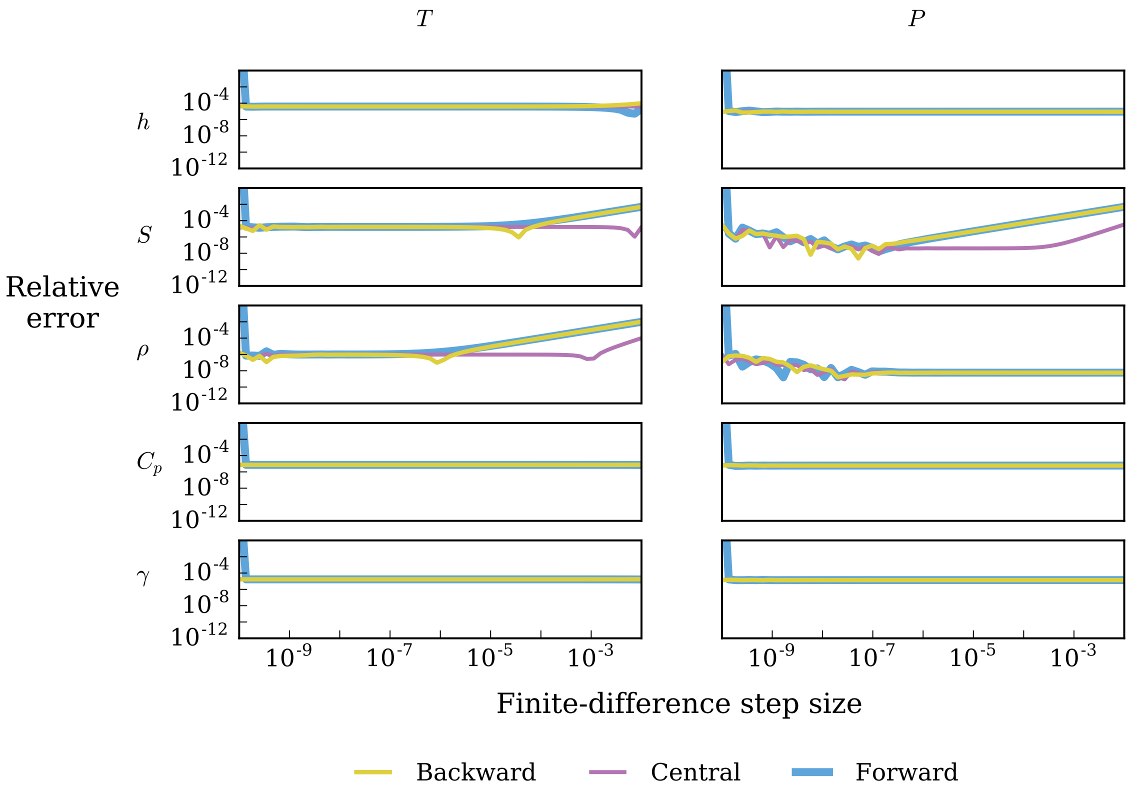

Fig. 9: Relative error of finite-difference approximations versus step size obtained for standard day conditions and by using computed analytic values as reference.

accuracy over a range of step sizes. This is expected because central differencing is second order accurate, whereas the other two schemes are first order. However, this accuracy comes at a cost because central differencing requires two function evaluations for each derivative.

Figure 11 shows the computational times for assembling the full 5×2 Jacobian of the responses with respect to the design variables. All computational times were measured on an Apple MacBook Pro laptop with a 2.6 GHz dual-core Intel Core i5 processor and 16 GB of memory. The computational cost of the central-difference approximation is twice that of the right- and left-difference schemes. Computing the derivatives with the direct method is faster than with the finite-difference approximations because each finite-difference step requires the convergence of the full nonlinear model, as opposed to the solution of a linear system for the direct method.

Of the two analytic methods, the direct method is faster than the adjoint method for this problem. Recall that the direct method corresponds to solving the left-hand side of the unified derivatives equation (27), whereas the adjoint method solves the right-hand side of the same equation. In this

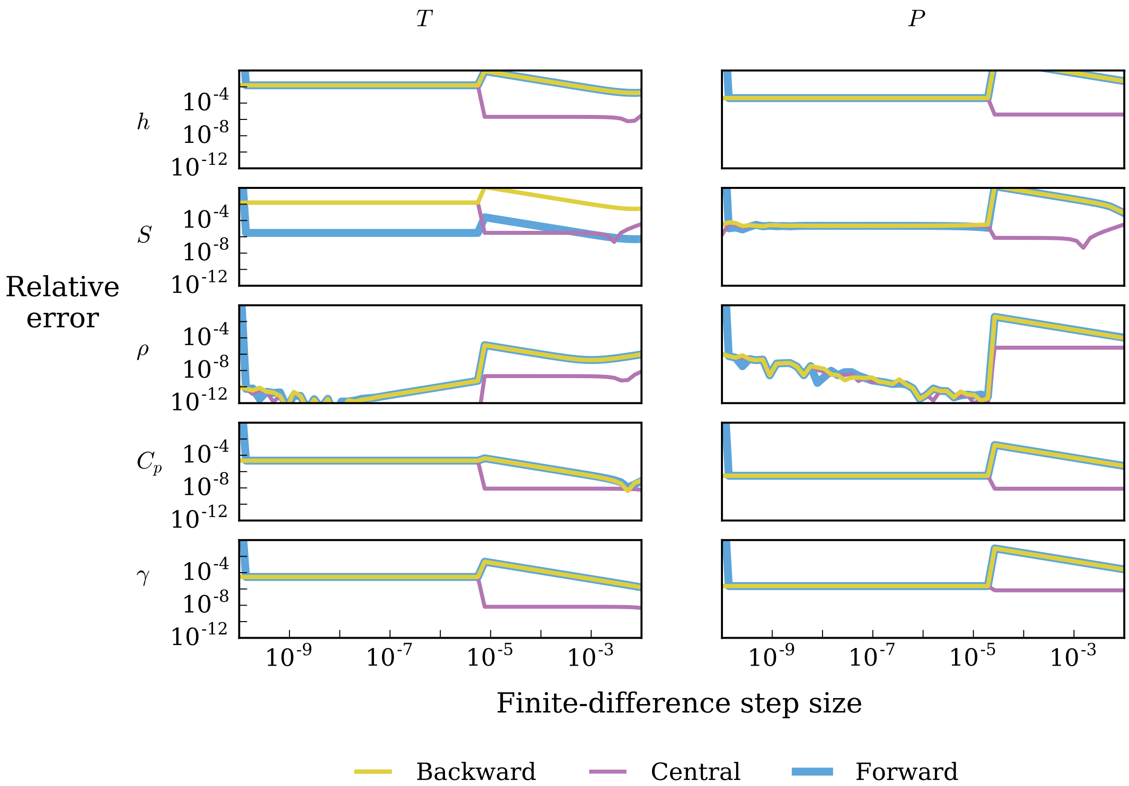

Fig. 10: Relative error of finite-difference approximations versus step size for cruise conditions and with computed analytic values as reference.

case, the direct method is expected to be faster because it involves two independent variables and five responses. Thus, for the direct method, two linear solutions are needed to compute the full Jacobian. For the adjoint method, five linear solutions are needed. For problems with more design variables than constraints, the adjoint method would become advantageous.

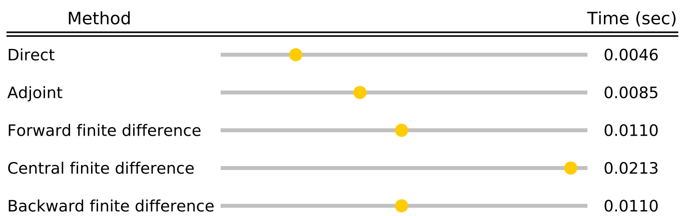

Fig. 11: Wall times required to compute the 5×2 Jacobian by using analytic methods (direct and adjoint) and finite-difference approximations.

VI. Optimization Results

A. Equivalence Ratio Optimization

To demonstrate the use of analytic derivatives in a gradient-based optimization, we ran a series of unconstrained optimizations at a fixed pre-combustion gas temperature of $518\,^{\circ}R$. Each optimization seeks to maximize the combustion temperature by varying ϕ for a given fixed pressure. We solved 200 optimizations for pressures ranging from 15 to 1500 psi. Figure 12 plots the optimal temperatures and the corresponding values of ϕ for the various pressures. The maximum temperatures occur for ϕ between 1.02 and 1.07, which seems counterintuitive. Under the assumption of perfect combustion, the maximum temperature would occur at the stoichiometric value, $\phi = 1$, where every molecule of diatomic oxygen would be converted to water. However, equilibrium calculations take into account disassociation effects, which simultaneously lower the maximum achievable temperature and cause that temperature to occur at a richer ϕ [51]. The effect of dissociation becomes less severe at higher pressures, which tends to favor the creation of slightly larger molecules, and the reaction more closely approximates ideal combustion. This is shown in Figure 12 by both the increasing maximum temperature and the decreasing optimal ϕ as pressure increases. These optimization problems are solved by using gradients computed with both the finite-difference method and the adjoint method. The results are identical in terms of both overall execution time and final objective value because, for this problem with a single design variable, the adjoint method does not significantly improve the speed or accuracy.

B. Equivalence Ratio and Pressure Optimization

These results establish that combustion becomes more efficient as pressure increases, even when accounting for equilibrium-chemistry effects. Therefore, we expect that including pressure as a design variable would yield the same result without requiring the parameter study for pressure. To verify this, a second series of optimizations are solved with both ϕ and P as design variables, again seeking to maximize combustion temperature. For this set of optimizations, the tolerance of the chemical equilibirum numerical solver is varied from 10^{-8} to 10^{-12} to test how sensitive the the optimization is to the accuracy of the solution. Again, we used two methods to compute deriva-

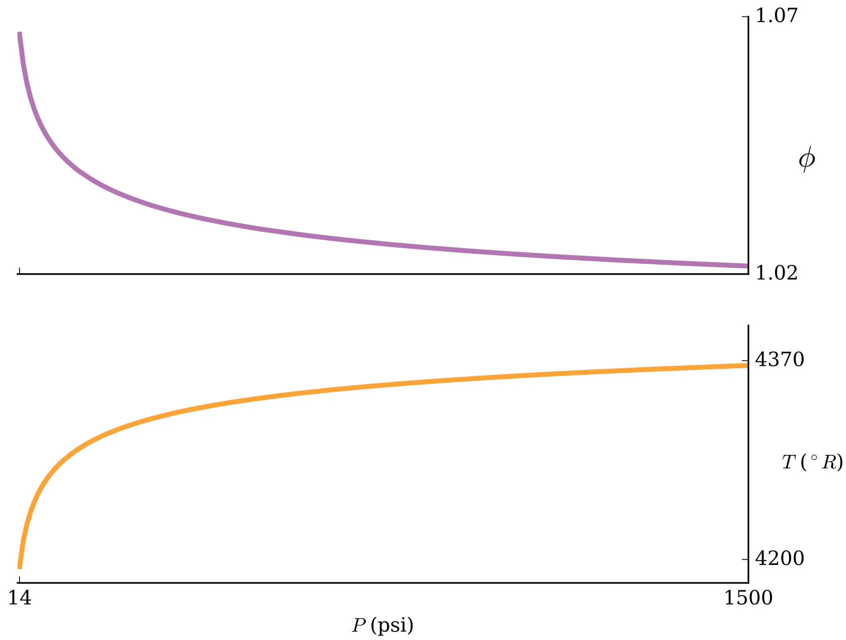

Fig. 12: Air-fuel equivalence ratio ϕ that maximizes the combustion temperature T as a function of pressure P ranging from 15 to 1500 psi.

tives: the adjoint method and the forward finite-difference approximation. Both derivative methods converge to $\phi = 1.021$ and $P = 1500$ psi, which is consistent with the previous optimization results shown in Figure 12. However, unlike the single-design-variable optimization, the adjoint method clearly performs better than the finite-difference method. The finite- difference approximation is at least twice as expensive as the adjoint method, as shown in Figure 13. One of the main challenges with using finite-difference derivatives is the need to have tight tolerances for the solvers, such as the one used to converge chemical equilibrium. The data in Figure 13 quantify this effect by comparing the performance for different tolerances of chemical- equilibrium solvers. The optimizations that use adjoint-computed gradients require a nearly constant computation time. However, the optimizations that use finite-difference derivatives require between 5.5 and 110 s, depending on solver tolerance and finite-difference step size. This wide range in computational time results from inaccurate derivatives, which force the optimizer to iterate more to converge to the required tolerance. The worst performance occurs when using a step size of 10^{-6} with tolerances between 10^{-11} and 10^{-10}, but the times are reduced for step sizes between 10^{-10} and 10^{-9}. This erratic behavior is particularly troubling and shows that finite-difference approximations are not reliable. For both step sizes, with a solver tolerance of 10^{-8}, the computational times started to rise. Beyond that point, numerical noise prevents the optimizer from converging when using finite-difference derivatives. This result highlights the value of the analytic approach. Even for just two design variables, the adjoint derivatives enable both faster and more stable optimization.

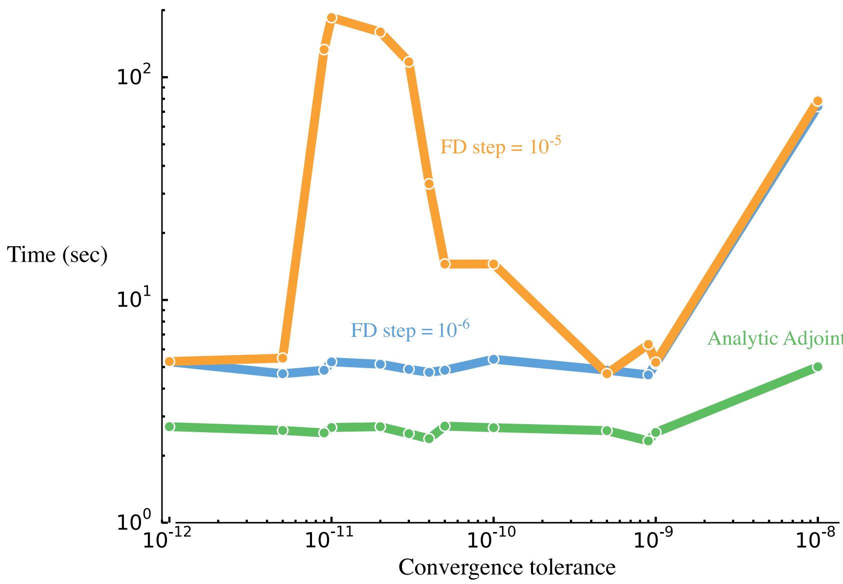

Fig. 13: Optimization wall time versus tolerance of chemical-equilibrium solver obtained by using analytic and finite-difference methods to compute derivatives.

VII. Conclusion

The need to incorporate propulsion cycle modeling into the MDO of aircraft motivates the development of a new propulsion cycle analysis model that can efficiently compute derivatives. As a first step toward this new capability, a new chemical equilibrium thermodynamics method was developed that predicts the gas properties of air and fuel-air mixtures for a wide range of fuels in a computationally efficient manner. The major contribution of the new method is the addition of analytic derivative computation (both direct and adjoint). To compute the coupled derivatives, it was necessary to re-formulate the mechanism for handling trace-species in the chemical equilibrium solver to create a continuous analysis response.

The new method was implemented using the OpenMDAO framework and its automatic coupled-derivatives capability to simplify the development. This implementation was verified by comparing its results against those obtained with the CEA code over a wide range of temperatures, pressures, and equivalence ratios. The verification results closely match, with a maximum discrepancy between the proposed implementation and CEA of less than 0.1%. In addition to validating the analysis, the accuracy of the analytic derivatives was verified by comparing them to finite-difference approximations. The analytic derivatives agree with the finite-difference approximations to within a relative error of $O(10^{-5})$.

The value of the analytic-derivatives approach is further demonstrated via two optimizations whose goal is to maximize the combustion temperature with respect to pressure and equivalence ratio. These optimizations compare the performance of the adjoint derivatives with that of the finite-difference derivatives and show that, even for an optimization with only two design variables, adjoint derivatives can significantly reduce the computational cost and increase the numerical stability. The improvements in speed and accuracy clearly demonstrate the value of the adjoint derivatives for optimization applications and suggest that similar improvements in performance for larger propulsion-cycle analyses are possible. Future work will consist in building a new propulsion-analysis method based on this thermodynamics method. Models for inlets, compressors, combustors, turbines, and nozzles will be developed with adjoint derivatives and then combined into full propulsion models.

VIII. Acknowledgments

We would like to thank Dr. Kenneth Moore and Bret Naylor for their technical support in developing the OpenMDAO framework features needed to enable this work and for helping to debug some of the more challenging analytic derivatives. We are also grateful to Chris Snyder for his advice and insight in helping us to understand the CEA tool. We would also like to thank the NASA Transformational Tools and Technologies Project, under the Aeronautics Research Mission Directorate, for funding this work.

[1] Hooker, J. R., Wick, A., Zeune, C., and Agelastos, A., "Over Wing Nacelle Installations for Improved Energy Efficiency," *31st AIAA Applied Aerodynamics Conference*, AIAA 2013-2920, 2013. doi:10.2514/6.2013-2920.

[2] Fujino, M. and Kawamura, Y., "Wave-drag Characteristics of an Over-The-Wing Nacelle Business-Jet Configuration," *Journal of Aircraft*, Vol. 40, No. 6, 2003, pp. 1177–1184. doi:10.2514/2.7207.

[3] Smith, A. M. O. and Roberts, H. E., "The Jet Airplane Utilizing Boundary Layer Ingestion for Propulsion," *Journal of Aeronautical Sciences*, Vol. 14, No. 2, 1947, pp. 97–109.

[4] Smith, L. H., "Wake Ingestion Propulsion Benefit," *Journal of Propulsion and Power*, Vol. 9, No. 1, February 1993, pp. 74–82. doi:10.2514/6.1991-2007.

[5] Plas, A. P., Sargeant, M. A., Madani, V., Crichton, D., Greitzer, E. M., Hynes, T. P., and Hall, C. A., "Performance of a Boundary Layer Ingesting (BLI) Propulsion System," *45th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2007-450, 2007. doi:10.2514/6.2007-450.

[6] Felder, J. L., Kim, H. D., and Brown, G. V., "Turboelectric Distributed Propulsion Engine Cycle Analysis for Hybrid-Wing-Body Aircraft," *47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, AIAA 2009-1132, 2009. doi:10.2514/6.2009-1132.

[7] Martins, J. R. R. A. and Lambe, A. B., "Multidisciplinary Design Optimization: A Survey of Architectures," *AIAA Journal*, Vol. 51, No. 9, September 2013, pp. 2049–2075. doi:10.2514/1.J051895.

[8] Drela, M., "Simultaneous Optimization of the Airframe, Powerplant, and Operation of Transport Aircraft," *Dept. of Aeronautics and Astronautics, Massachusetts Inst. of Technology*, 2010.

[9] Welstead, J. R. and Felder, J. L., "Conceptual Design of a Single-Aisle Turboelectric Commercial Transport with Fuselage Boundary Layer Ingestion," *54th AIAA Aerospace Sciences Meeting*, AIAA 2016-1027, 2016. doi:10.2514/6.2016-1027.

[10] Martins, J. R. R. A., Sturdza, P., and Alonso, J. J., "The Complex-Step Derivative Approximation," *ACM Transactions on Mathematical Software*, Vol. 29, No. 3, September 2003, pp. 245–262. doi:10.1145/838250.838251.

[11] Martins, J. R. R. A. and Hwang, J. T., "Review and Unification of Methods for Computing Derivatives of Multidisciplinary Computational Models," *AIAA Journal*, Vol. 51, No. 11, 2013, pp. 2582–2599. doi:10.2514/1.J052184.

[12] Nielsen, E. J. and Diskin, B., "Discrete Adjoint-Based Design for Unsteady Turbulent Flows on Dynamic Overset Unstructured Grids," *AIAA Journal*, Vol. 51, No. 6, 2013, pp. 1355–1373. doi:10.2514/1.J051859.

[13] Palacios, F., Economon, T. D., Aranake, A., Copeland, S. R., Lonkar, A. K., Lukaczyk, T. W., Manos-

alvas, D. E., Naik, K. R., Padron, S., Tracey, B., Variyar, A., and Alonso, J. J., "Stanford University Unstructured (SU2): Open-source Analysis and Design Technology For Turbulent Flows," *52nd Aerospace Sciences Meeting*, AIAA 2014-0243, 2014. doi:10.2514/6.2014-0243.

[14] Carrier, G., Destarac, D., Dumont, A., Meheut, M., Din, I. S. E., Peter, J., Khelil, S. B., Brezillon, J., and Pestana, M., "Gradient-Based Aerodynamic Optimization with the elsA Software," *52nd Aerospace Sciences Meeting*, Feb. 2014. doi:10.2514/6.2014-0568.

[15] Lyu, Z., Kenway, G. K., Paige, C., and Martins, J. R. R. A., "Automatic Differentiation Adjoint of the Reynolds-Averaged Navier–Stokes Equations with a Turbulence Model," *21st AIAA Computational Fluid Dynamics Conference*, San Diego, CA, Jul. 2013. doi:10.2514/6.2013-2581.

[16] Lyu, Z. and Martins, J. R. R. A., "Aerodynamic Design Optimization Studies of a Blended-Wing-Body Aircraft," *Journal of Aircraft*, Vol. 51, No. 5, September 2014, pp. 1604–1617. doi:10.2514/1.C032491.

[17] Lyu, Z., Kenway, G. K., and Martins, J. R. R. A., "Aerodynamic Shape Optimization Investigations of the Common Research Model Wing Benchmark," *AIAA Journal*, Vol. 53, No. 4, April 2015, pp. 968–985. doi:10.2514/1.J053318.

[18] Chen, S., Lyu, Z., Kenway, G. K. W., and Martins, J. R. R. A., "Aerodynamic Shape Optimization of the Common Research Model Wing-Body-Tail Configuration," *Journal of Aircraft*, Vol. 53, No. 1, January 2016, pp. 276–293. doi:10.2514/1.C033328.

[19] Vassberg, J. and Jameson, A., "Influence of Shape Parameterization on Aerodynamic Shape Optimization," Tech. rep., Von Karman Institute, Brussels, Belgium, April 2014.

[20] Kenway, G. K. W. and Martins, J. R. R. A., "Multipoint Aerodynamic Shape Optimization Investigations of the Common Research Model Wing," *AIAA Journal*, Vol. 54, No. 1, January 2016, pp. 113–128. doi:10.2514/1.J054154.

[21] Dumont, A. and Meheut, M., "Gradient-Based Optimization of CRM Wing-alone and Wing-body-tail Configurations by RANS Adjoint Technique," *54th AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics (AIAA), January 2016. doi:10.2514/6.2016-1293.

[22] Lyu, Z., Xu, Z., and Martins, J. R. R. A., "Benchmarking Optimization Algorithms for Wing Aerodynamic Design Optimization," *Proceedings of the 8th International Conference on Computational Fluid Dynamics*, Chengdu, Sichuan, China, ICCFD8-2014-0203, July 2014.

[23] Sanghi, V., Lakshmanan, B. K., and Sundararajan, V., "Survey of Advancements in Jet-Engine Thermodynamic Simulation," *Journal of Propulsion and Power*, Vol. 16, No. 5, 2000, pp. 797–807. doi:10.2514/2.5644.

[24] Jones, S., "An Introduction to Thermodynamic Performance Analysis of Aircraft Gas Turbine Engine

Cycles Using the Numerical Propulsion System Simulation Code," Tech. rep., Glenn Research Center, 2007, NASA TM-2007-214690.

[25] Hendricks, E. S., Jones, S. M., and Gray, J. S., "Design Optimization of a Variable-Speed Power-Turbine," *50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, Cleveland, Ohio, AIAA-2014-3445, July 2014. doi:10.2514/6.2014-3445.

[26] Geiselhart, K., Ozoroski, L. P., Fenbert, J. W., Shields, E. W., and Li, W., "Integration of Multifidelity Multidisciplinary Computer Codes for Design and Analysis of Supersonic Aircraft," *49th AIAA Aerospace Sciences Meeting*, AIAA 2011-465, 2011. doi:10.2514/6.2011-465.

[27] Allison, D., Morris, C., Schetz, J., Kapania, R., Sultan, C., Deaton, J., and Grandhi, R., "A Multidisciplinary Design Optimization Framework for Design Studies of an Efficient Supersonic Air Vehicle," *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA 2012-5492, 2012. doi:10.2514/6.2012-5492.

[28] Allison, D. L., Alyanak, E., and Bhagat, N., "High Fidelity, Nonlinear, Integrated Nozzle Installation Effects for Numerical Propulsion System Simulation," *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA 2015-0649, 2015. doi:10.2514/6.2015-0649.

[29] Allison, D. L., Alyanak, E. J., and Shimmin, K., "Aircraft System Affects Including Propulsion and Air Cycle Machine Coupled Interactions," *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA 2016-0671, 2016. doi:10.2514/6.2016-0671.

[30] Kenway, G. K. W., Kennedy, G. J., and Martins, J. R. R. A., "Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Adjoint Derivative Computations," *AIAA Journal*, Vol. 52, No. 5, 2014, pp. 935–951. doi:10.2514/1.J052255.

[31] Hwang, J. T., Lee, D. Y., Cutler, J. W., and Martins, J. R. R. A., "Large-Scale Multidisciplinary Optimization of a Small Satellite's Design and Operation," *Journal of Spacecraft and Rockets*, Vol. 51, No. 5, September 2014, pp. 1648–1663. doi:10.2514/1.A32751.

[32] Hwang, J. T. and Martins, J. R. R. A., "A computational architecture enabling modular coupling and differentiation of heterogeneous numerical models," *ACM Transactions on Mathematical Software*, 2016, (Submitted).

[33] Tran, D. H. and Snyder, C. A., "Effects of Chemical Equilibrium on Turbine Engine Performance for Various Fuels and Combustor Temperatures," Tech. Rep. NASA-TM-105399, Glenn Research Center, 1992.

[34] Nuzum, S. R., Donovan, A., Roberts, R. A., and Wolff, M., "Dynamic Modeling at a Vehicle Level of a

Cryogenic Based Thermal System for a High Powered System," *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA 2016-0673, 2016. doi:10.2514/6.2016-0673.

[35] Walter, M. A. T. and Owen, P. R., "Element Potential-Based Procedure for Metamodeling of Combustion Products," *Journal of Propulsion and Power*, Vol. 30, No. 6, 2014, pp. 1711–1720. doi:10.2514/1.B35189.

[36] Gordon, S. and McBride, B. J., "Computer Program for Calculation of Complex Chemical Equilibrium Compositions, Rocket Performance, Incident and Reflected Shocks, and Chapman-Jouguet Detonations," 1994, NASA RP-1311.

[37] Schecher, W. D. and McAvoy, D. C., "MINEQL+: A Software Environment for Chemical Equilibrium Modeling," *Computers, Environment and Urban Systems*, Vol. 16, No. 1, 1992, pp. 65–76. doi:10.1016/0198-9715(92)90053-T.

[38] Konigsberger, E. and Eriksson, G., "A New Optimization Routine for Chemsage," *Calphad*, Vol. 19, No. 2, 1995, pp. 207–214. doi:10.1016/0364-5916(95)00021-6.

[39] Smith, W. R. and Missen, R. W., *Chemical Reaction Equilibrium Analysis: Theory and Algorithms*, Wiley, 1st ed., 1982.

[40] Goodwin, D. G., Moffat, H. K., and Speth, R. L., "Cantera: An Object-oriented Software Toolkit for Chemical Kinetics, Thermodynamics, and Transport Processes," `http://www.cantera.org`, 2014, Version 2.1.2.

[41] Chin, J., Gray, J., Jones, S., and Berton, J., "Open-Source Conceptual Sizing Models for the Hyperloop Passenger Pod," *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA 2015-1587, 2015. doi:10.2514/6.2015-1587.

[42] Haglind, F. and Singh, R., "Design of Aero Gas Turbines Using Hydrogen," *Journal of Engineering for Gas Turbines and Power*, Vol. 128, No. 4, 2006, pp. 754–764. doi:10.1115/1.2179468.

[43] Fulton, K., "Cryogenic Fueled Turbofans: Kuznetsov Bureau's Pioneer Work on LH2 and LNG Duel-Fuel Engines," *Aircraft Engineering and Aerospace Technology*, Vol. 65, No. 11, 1993, pp. 8–11. doi:10.1108/eb037431.

[44] Leal, A. M., Blunt, M. J., and LaForce, T. C., "A Robust and Efficient Numerical Method for Multiphase Equilibrium Calculations: Application to CO2-Brine-Rock Systems at High Temperatures, Pressures and Salinities," *Advances in Water Resources*, Vol. 62, Part C, 2013, pp. 409–430. doi:10.1016/j.advwatres.2013.02.006.

[45] Lambe, A. B. and Martins, J. R. R. A., "Extensions to the Design Structure Matrix for the Description

of Multidisciplinary Design, Analysis, and Optimization Processes," *Structural and Multidisciplinary Optimization*, Vol. 46, 2012, pp. 273–284. doi:10.1007/s00158-012-0763-y.

[46] Jr., M. W. C., "NIST-JANAF Thermochemical Tables, Forth Edition, Monograph 9," *Journal of Physical and Chemical Reference Data*, 1998.

[47] Gray, J. S., Hearn, T. A., Moore, K. T., Hwang, J., Martins, J. R. R. A., and Ning, A., "Automatic Evaluation of Multidisciplinary Derivatives Using a Graph-Based Problem Formulation in OpenMDAO," *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA 2014-204280, August 2014. doi:doi:10.2514/6.2014-2042.

[48] Hwang, J. T., *A Modular Approach to Large-Scale Design Optimization of Aerospace Systems*, Ph.D. thesis, University of Michigan, 2015.

[49] Griewank, A., *Evaluating Derivatives*, SIAM, Philadelphia, 2000.

[50] Martins, J. R. R. A., Alonso, J. J., and Reuther, J. J., "A Coupled-Adjoint Sensitivity Analysis Method for High-Fidelity Aero-Structural Design," *Optimization and Engineering*, Vol. 6, No. 1, March 2005, pp. 33–62. doi:10.1023/B:OPTE.0000048536.47956.62.

[51] Lefebvre, A. H., *Gas Turbine Combustion*, CRC Press, 3rd ed., 1998.