PERFORM
Prototyping Environment for Reacting Flow Order Reduction Methods

Solver Documentation

Christopher R. Wentland*, Ashish S. Nair, Elnaz Rezaian,
Cheng Huang, Karthik Duraisamy

February 3, 2021

*Email: chriswen [at] umich [dot] edu

Contents

1 Introduction

This code was originally written to be a sort of low-fidelity 1D Python port of the General Mesh and Equations Solver (GEMS), originally written by Guoping Xia at Purdue University and expanded on over the decades by researchers from Purdue University, the University of Michigan, Ann Arbor, and the Air Force Research Laboratory. However, it became apparent that this could be a useful tool for members of the reduced-order model (ROM) community to implement and test new ROM methods for challenging (yet computationally-manageable) reacting flow problems. Hopefully, this code may help lower the barrier to entry for folks who are not familiar with multi-species flows or combustion, or don't want to invest in licensing and learning an enormous research combustion code. Ideally, this code enables users to rapidly prototype new ROM algorithms in an extremely accessible Python environment.

Although this port is still "low-fidelity" in the sense that the code lacks much of the functionality of the original GEMS solver (e.g. some robustness controls, a wide variety of flow physics and reaction models, parallel processing), the problems it is designed to model are still extremely challenging for modern ROM methods. We hope that open access and contribution to this code can help tackle some of the many problems in the field of ROMs for reacting flows.

This document specifically serves as a reference for the theory behind the unsteady reacting flow solver and ROM methods. A separate user guide is supplied in the documentation directory of the code repository. If you notice any errors in either document (or the code), please feel free to email me. If you would like to contribute to the code, please email me and I can give you contributor permissions.

The mathematical notation in this paper is as follows: scalars and scalar-valued functions are denoted by italicized, lowercase letters/symbols (e.g., a), vectors and vector-valued functions are denoted by boldface, lowercase letters/symbols (e.g., $\mathbf{a}$), matrices and matrix-valued functions are denoted by boldface, uppercase letters/symbols (e.g., $\mathbf{A}$), and vector spaces are denoted by calligraphic, uppercase letters/symbols (e.g., $\mathcal{A}$).

2 Physics

2.1 Governing Equations

The governing equations for the 1D Navier-Stokes equations with species transport and reactions are

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} - \frac{\partial \mathbf{f}_v}{\partial x} = \mathbf{s}. \tag{1}$$

The conservative state $\mathbf{q}$, the inviscid flux vector $\mathbf{f}$, the viscous flux vector $\mathbf{f}_v$, and the source term $\mathbf{s}$ are given by

$$\mathbf{q} = \begin{bmatrix} \rho \\ \rho u \\ \rho h^0 - p \\ \rho Y_l \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho h^0 u \\ \rho Y_l \end{bmatrix}, \quad \mathbf{f}_v = \begin{bmatrix} 0 \\ \tau \\ u\tau - q \\ -\rho V_l Y_l \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dot{\omega}_l \end{bmatrix}, \tag{2}$$

where ρ is density, u is velocity, h^0 is stagnation enthalpy, p is static pressure, and Y_l is the mass fraction of the lth chemical species. For a system with N_Y chemical species, only $N_Y - 1$ species transport equations are solved, as the final species mass fraction Y_{N_Y} can be computed from the fact that all mass fractions must sum to unity,

$$Y_{N_Y} = 1 - \sum_{l=1}^{N_Y - 1} Y_l \tag{3}$$

The stagnation enthalpy h^0 is given by

$$h^0 = \sum_{l=1}^{N_Y} h_l Y_l + \frac{1}{2} u^2, \tag{4}$$

where h_l is the enthalpy of the lth species. The shear stress τ is given by

$$\tau = \frac{4}{3} \mu \frac{\partial u}{\partial x}, \tag{5}$$

where μ is the mixture dynamic viscosity. The heat flux q is given by

$$q = -K \frac{\partial T}{\partial x} + \rho \sum_{l}^{N_Y} V_l Y_l h_l, \tag{6}$$

where K is the mixture thermal conductivity. The diffusion velocity term $V_l Y_l$, which also appears in the species transport viscous flux, is approximated as

$$V_l Y_l = -D_{l,M} \frac{\partial Y_l}{\partial x} \tag{7}$$

where $D_{l,M}$ is the mass diffusion coefficient for the lth species diffusing into the mixture.

The calculation of the species enthalpy h_l, the mixture dynamic viscosity μ, the mixture thermal conductivity K, and the mass diffusion coefficient $D_{l,M}$ depends on the gas model being used, and will be discussed in Section 2.2. Additionally, the species production rate $\dot{\omega}_l$ depends on the reaction model being used, and will be discussed in Section 2.3.

2.2 Gas Models

Several gas models are available in `PERFORM` to model the thermodynamic and transport properties of gases, each with varying levels of accuracy in certain circumstances. We begin by defining several variables whose calculations are universal across all models.

The mixture molecular weight W is given by

$$W = \frac{1}{\sum_{l=1}^{N_Y} Y_l W_l} \tag{8}$$

where W_l is the molecular weight of the lth species, given in units (g/mol). The mole fraction of the lth species is given by

$$X_l = W \frac{Y_l}{W_l} \tag{9}$$

The molar concentration, denoted by $[X_l]$, can be computed similarly from

$$[X_l] = \rho \frac{Y_l}{W_l}. \tag{10}$$

The specific gas constant R_l of the the lth species is given by

$$R_l = \frac{R_u}{W_i} \tag{11}$$

where $R_u = 8314.4621$ J/K-kmol is the universal gas constant. For a mixture with molecular weight W, the mixture specific gas constant is similarly given by

$$R = \frac{R_u}{W} \tag{12}$$

The ratio of specific heats γ can be calculated as

$$\gamma = \frac{c_p(T)}{c_v(T)}, \tag{13}$$

where c_p is the mixture specific heat capacity at constant pressure, and c_v is the mixture specific heat capacity at constant volume, both which may be a function of temperature depending on the gas model being used. The mixture specific heat capacity at constant pressure may be simply calculated from

$$c_p(T) = \sum_{l=1}^{N_Y} Y_l c_{p,l}(T) \tag{14}$$

Throughout this section, derivatives of certain quantities with respect to species mass fractions will leverage the fact that the species mass fractions must sum to unity (Eq. 3). Several mixture quantities,

using the dummy variable ξ as an example, can be represented as

$$\xi = \sum_{l=1}^{N_Y} Y_l \xi_l \tag{15}$$

$$= \sum_{l=1}^{N_Y-1} Y_l \xi_l + Y_{N_Y} \xi_{N_Y} \tag{16}$$

$$= \sum_{l=1}^{N_Y-1} Y_l \xi_l + \xi_{N_Y} \left(1 - \sum_{l=1}^{N_Y-1} Y_l \right) \tag{17}$$

$$= \xi_{N_Y} + \sum_{l=1}^{N_Y-1} Y_l (\xi_l - \xi_{N_Y}) \tag{18}$$

The derivatives of some quantity ξ with respect to the first $N_Y - 1$ species mass fractions can often be simplified by using this format.

2.2.1 Calorically-perfect Gas (CPG)

The CPG model is the simplest gas model. As a perfect gas, no inter-molecular forces are accounted for (which may not be valid at extremely high pressures). The CPG model distinguishing itself by the assumption that the heat capacity at constant pressure of each species, $c_{p,l}$, is constant, i.e. $c_{p,l}(T) = c_{p,l}$.

Under this model, the governing equations (Eq. 1) are closed by the ideal gas law, where density is calculated as

$$\rho = \frac{p}{RT}. \tag{19}$$

From the ideal gas law, it follows that the individual species densities can be computed as

$$\rho_l = \rho \frac{W_l}{W}. \tag{20}$$

The absolute species enthalpy for the lth species is computed as

$$h_l = h_{ref,l} + c_{p,l} T \tag{21}$$

where $h_{ref,l}$ is the enthalpy of formation of the lth species at a reference temperature of 0 K. The species entropy for the lth species is given by

$$s_l = c_{p,l} \ln(T) - R_l \ln(p) \tag{22}$$

The mixture density, enthalpy, entropy can be computed very simply from

$$\rho = \left(\sum_{l=1}^{N_Y} \frac{Y_l}{\rho_l} \right)^{-1}, \quad h = \sum_{l=1}^{N_Y} Y_l h_l, \quad s = \sum_{l=1}^{N_Y} Y_l s_l. \tag{23}$$

For various calculations (particularly for the Jacobians of the flux and source terms), calculating derivatives of density, stagnation enthalpy, and entropy with respect to the primitive state are useful. The deriva-

tives of density are given by

$$\rho_p = \frac{\partial \rho}{\partial p} = \frac{1}{RT} = \frac{\rho}{p}, \tag{24}$$

$$\rho_u = \frac{\partial \rho}{\partial u} = 0, \tag{25}$$

$$\rho_T = \frac{\partial \rho}{\partial T} = -\frac{p}{RT^2} = -\frac{\rho}{T}, \tag{26}$$

$$\rho_{Y_l} = \frac{\partial \rho}{\partial Y_l} = \rho W \left(\frac{1}{W_{N_Y}} - \frac{1}{W_l} \right). \tag{27}$$

The derivatives of stagnation enthalpy are given by

$$h^0_p = \frac{\partial h^0}{\partial p} = 0, \tag{28}$$

$$h^0_u = \frac{\partial h^0}{\partial u} = u, \tag{29}$$

$$h^0_T = \frac{\partial h^0}{\partial T} = c_p, \tag{30}$$

$$h^0_{Y_l} = \frac{\partial h^0}{\partial Y_l} = h_l - h_{N_Y}. \tag{31}$$

The derivatives of entropy are given by

$$s_p = \frac{\partial s}{\partial p} = -\frac{R}{p} = -\frac{1}{\rho T}, \tag{32}$$

$$s_u = \frac{\partial s}{\partial u} = 0, \tag{33}$$

$$s_T = \frac{\partial s}{\partial T} = \frac{c_p}{T}, \tag{34}$$

$$s_{Y_l} = \frac{\partial s}{\partial Y_l} = s_l - s_{N_Y}. \tag{35}$$

Here, the derivatives with respect to species mass fraction are valid for the first $N_Y - 1$ species.

The sound speed in the mixture may be calculated from these derivative quantities or the familiar simplified form, respectively given by

$$c = \sqrt{\frac{1}{\rho_p + \rho_T \frac{1-\rho h_p}{\rho h_T}}} \tag{36}$$

$$= \sqrt{\gamma RT} \tag{37}$$

Under the ideal gas assumption, the specific heat capacity at constant volume can be computed from

$$c_v = c_p - R, \tag{38}$$

leading to the simplification of the ratio of specific heats,

$$\gamma = \frac{c_p}{c_p - R}. \tag{39}$$

For the CPG model, the species dynamic viscosities are computed as a function of temperature from Sutherland's law, given by

$$\mu_l(T) = \mu_{ref,l} \left(\frac{T}{T_{ref,l}} \right)^{3/2} \left(\frac{T_{ref,l} + S_l}{T + S_l} \right) \tag{40}$$

where $\mu_{ref,l}$, $T_{ref,l}$, and S_l are the reference dynamic viscosity, reference temperature, and Sutherland temperature (constant, tabulated quantities), respectively, for the lth species. If $T_{ref,l} = 0$ K, then it is assumed $\mu_l = \mu_{ref,l}$. The mixture dynamic viscosity is computed from Wilke's mixing law,

$$\mu = 2\sqrt{2} \sum_{l=1}^{N_Y} \frac{X_l \mu_l}{\phi_l}, \tag{41}$$

where the denominator term is given by

$$\phi_l = \sum_{m=1}^{N_Y} X_m \left(1 + \left(\frac{\mu_l}{\mu_m} \right)^{1/2} \left(\frac{W_m}{W_l} \right)^{1/4} \right)^2 \left(1 + \frac{W_l}{W_m} \right)^{-1/2} \tag{42}$$

Species thermal conductivity values are computed from

$$K_l = \frac{\mu_l c_{p,l}}{\mathrm{Pr}_l} \tag{43}$$

where Pr_l is the Prandtl number of the lth species. The mixture thermal conductivity is computed from the mixing law of Mathur, Tondon, and Saxena,

$$K = \frac{1}{2} \left(\sum_{l=1}^{N_Y} X_l K_l + \left(\sum_{l=1}^{N_Y} \frac{X_l}{K_l} \right)^{-1} \right), \tag{44}$$

Species mass diffusion coefficients for the lth species diffusing into the mixture are computed from

$$D_{l,M} = \frac{\mu_l}{\rho \mathrm{Sc}_l} \tag{45}$$

where Sc_l is the Schmidt number of the lth species.

2.2.2 Thermally-perfect Gas (TPG)

This section will be updated shortly, check back soon!

2.3 Reaction Models

Various models exist for computing the species production rate $\dot{\omega}_l$. Of course, the most accurate method would be elementary reaction mechanisms, but these often involve thousands of species and reactions, pressure corrections and are thus infeasible even for 1D problems. Reduced mechanisms and global reactions offer two comparatively inexpensive alternatives. We describe both in turn.

2.3.1 Reduced Mechanisms

This section describes the modeling of reaction source terms by reduced mechanisms. The same methodology may be applied to elementary reaction mechanisms, but we assume that some empirical, reduced mechanism is used to grossly reduce the number of modeled species and reactions to $\mathcal{O}(10-100)$.

We begin by supposing that a reaction mechanism is made up of N_r reactions governing the production and consumption of N_Y chemical species. The rth reaction of this mechanism can be described by the formula

$$\sum_{l=1}^{N_Y} \nu'_{l,r}\chi_l \underset{k_{f,r}}{\overset{k_{r,r}}{\rightleftharpoons}} \sum_{l=1}^{N_Y} \nu''_{l,r}\chi_l \tag{46}$$

where χ_l is the chemical formula for the lth chemical species, and $\nu'_{l,r}$ and $\nu''_{l,r}$ are the stoichiometric coefficients of the reactants and products, respectively, for the lth species in the rth reaction. The coefficients $k_{f,r}$ and $k_{r,r}$ forward and reverse reaction rates for the rth reaction. For example, the single-step combustion of methane can be described by

$$\mathrm{CH_4} + 2\mathrm{O_2} \underset{k_f}{\overset{k_r}{\rightleftharpoons}} 2\mathrm{H_2O} + \mathrm{CO_2} \tag{47}$$

In this case, $\nu'_{\mathrm{CH_4}} = 1$, $\nu''_{\mathrm{CH_4}} = 0$, $\nu'_{\mathrm{H_2O}} = 0$, $\nu''_{\mathrm{H_2O}} = 2$, and so on. The species production rates can be written in terms of these stoichiometric coefficients and reaction rates,

$$\dot{\omega}_l = W_l \sum_{r=1}^{N_r} (\nu''_{l,r} - \nu'_{l,r}) w_r \tag{48}$$

where the rate-of-progress w_r is given by

$$w_r = k_{f,r} \prod_{l=1}^{N_Y} [X_l]^{\nu'_{l,r}} - k_{r,r} \prod_{l=1}^{N_Y} [X_l]^{\nu''_{l,r}} \tag{49}$$

where we recall that $[X_l]$ is the molar concentration of the lth species. The forward reaction rate is modeled by the Arrhenius rate equation

$$k_{f,r} = A_r T^{b_r} \exp\left(\frac{-E_{a,r}}{R_u T}\right) \tag{50}$$

where the coefficients A_r, b_r, and $E_{a,r}$ are tabulated constants given by the reaction mechanism. The reverse reaction rate is computed from

$$k_{b,r} = \frac{k_{f,r}}{k_{c,r}} \tag{51}$$

where the equilibrium constant $k_{c,r}$ is formulated in terms of the change in Gibbs free energy. When reversible reactions are added to `PERFORM`, this section will be updated to include this calculation.

With all of these components in hand, the species production rate can be computed from Eq. 48. We emphasize that the constant coefficients $\nu'_{l,r}$, $\nu''_{l,r}$, A_r, b_r, and $E_{a,r}$ are all provided by a given reaction mechanism. On the other hand, only the temperature T, mixture gas constant R, and molar concentrations $[X_l]$ will change throughout the simulation.

2.3.2 Global Mechanism

Global reaction mechanisms vastly simplify the above reversible reaction model by assuming that all reactions only proceed in the forwards direction, i.e. $k_{r,r} = 0$ for all N_r reactions. As an example, the single-step combustion of methane (Eq. 47) would thus be written as

$$\mathrm{CH_4} + 2\mathrm{O_2} \xrightarrow{k} 2\mathrm{H_2O} + \mathrm{CO_2} \tag{52}$$

These global reaction mechanisms must be fitted to empirical data to best model the reaction without including reversible reactions. As such, the rate-of-progress is modified to have the form

$$w_r = k_r \prod_{l=1}^{N_Y} [X_l]^{\alpha_{l,r}} \tag{53}$$

where the coefficients $\alpha_{l,r}$ can be any real-valued number. The reaction rate k_r is still computed from the Arrhenius rate defined in Eq. 50. Usually this rate-of-progress is only defined for a single species involved in the reaction, and production rates for other species in the reaction are determined from stoichiometric ratios. For example, the single-step Westbrook-Dryer methane combustion global reaction mechanism (defined by Eq. 52) provides

$$w = 2.12 \times 10^{12} \exp\left(\frac{-24,358}{T}\right) [X_{\mathrm{CH_4}}]^{0.2} [X_{\mathrm{O_2}}]^{1.3} \tag{54}$$

Note that here, the Arrhenius temperature exponent coefficient $b = 0$, and the factor E_a/R_u have been combined.

3 Numerics

In this section, we describe the various methods used to discretize the governing PDE in both time and space. Details of new methods will be added when they are implemented in `PERFORM`.

3.1 Spatial Discretization

A cell-centered finite volume formulation is used to discretize the one-dimensional spatial domain into N_c discrete control volumes (or "cells"). The spatial coordinate of the center of the ith cell is denoted by x_i, and the coordinate of the cell's left "face" and right "face" are given by $x_{i-1/2}$ and $x_{i+1/2}$, respectively. The general form of the finite volume formulation is given by

$$\frac{\mathrm{d}}{\mathrm{dt}} \int_{\Omega} \mathbf{q} \, \mathrm{dV} + \int_{\partial\Omega} \nabla \cdot (\mathbf{f} - \mathbf{f}_v) \cdot \vec{n} \, \mathrm{ds} = \int_{\Omega} \mathbf{s} \, \mathrm{dV} \tag{55}$$

where Ω is the interior domain of the finite volume cell, and $\partial\Omega$ is the surface of the cell. Simplifying this formulation to one spatial dimension results in

$$\frac{\mathrm{d}}{\mathrm{dt}} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{q} \, \mathrm{d}x + \Big((\mathbf{f} - \mathbf{f}_v)_{x_{i+1/2}} - (\mathbf{f} - \mathbf{f}_v)_{x_{i-1/2}} \Big) = \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{s} \, \mathrm{d}x \tag{56}$$

where, for example, the term $\mathbf{f}_{x_{i-1/2}}$ is the inviscid flux through the ith cell's left face. The calculation of these fluxes is discussed in Section 3.3.

Evaluating the volume integrals,

$$\Delta x_i \frac{\mathrm{d}\mathbf{q}_i}{\mathrm{dt}} + \Big((\mathbf{f} - \mathbf{f}_v)_{x_{i+1/2}} - (\mathbf{f} - \mathbf{f}_v)_{x_{i-1/2}} \Big) = \Delta x_i \, \mathbf{s}_i \tag{57}$$

where the cell length is $\Delta x_i = x_{i+1/2} - x_{i-1/2}$. We can rearrange terms to arrive at

$$\frac{\mathrm{d}\mathbf{q}_i}{\mathrm{dt}} = \frac{1}{\Delta x_i} \Big((\mathbf{f} - \mathbf{f}_v)_{x_{i-1/2}} - (\mathbf{f} - \mathbf{f}_v)_{x_{i+1/2}} \Big) + \mathbf{s}_i \tag{58}$$

With this definition, we can extend the definition of the conservative solution, flux, and source vectors to include those at every cell in the spatial domain. For example, the conservative state would be defined as

$$\mathbf{q} := [\rho_1, \ldots, \rho_{N_c}, (\rho u)_1, \ldots, (\rho u)_{N_c}, (\rho h^0 - p)_1, \ldots, (\rho h^0 - p)_{N_c}, (\rho Y_l)_1, \ldots, (\rho Y_l)_{N_c}]^\top \tag{59}$$

and the flux/source terms are defined accordingly. Thus, the ODE governing the entire discrete system state is given by

$$\frac{\mathrm{d}\mathbf{q}}{\mathrm{dt}} = \frac{1}{\Delta \mathbf{x}} \left((\mathbf{f} - \mathbf{f}_v)_L - (\mathbf{f} - \mathbf{f}_v)_R \right) + \mathbf{s} \tag{60}$$

where we have denoted the left and right face flux vectors with the subscripts L and R, respectively. The vector of cell lengths is denoted by $\Delta\mathbf{x}$. We can lump the terms on the right-hand side as the "right-hand side" (RHS) term

$$\mathbf{g}(\mathbf{q}) := \frac{1}{\Delta \mathbf{x}} \left((\mathbf{f} - \mathbf{f}_v)_L - (\mathbf{f} - \mathbf{f}_v)_R \right) + \mathbf{s} \tag{61}$$

This definition will be useful for describing ROM formulations compactly in Section 4.

3.2 Time Discretization

Here we describe various methods for discretizing the time derivative in Eq. 61, i.e. solving for the solution state at a discrete time instance t^n, given by $\mathbf{q}(t^n) = \mathbf{q}^n$. Here, n indicates the number of the next time step to be solved for. All schemes use a fixed time step size Δt; adaptive time steps (for robustness controls) have not been implemented, and are unlikely to be implemented.

All time discretizations can be described by a residual function for which we seek the solution to

$$\mathbf{r}(\mathbf{q}^n) := \mathbf{r}^n = \mathbf{0} \tag{62}$$

As additional time integration methods are implemented in `PERFORM`, they will be detailed here.

3.2.1 Runge-Kutta Schemes

The residual function for a general s-stage Runge-Kutta scheme is given by

$$\mathbf{r}^n = \mathbf{q}^n - \mathbf{q}^{n-1} - \Delta\mathrm{t}\sum_{i=1}^{s} b_i \mathbf{h}_i, \tag{63}$$

where

$$\mathbf{h}_1 = \mathbf{g}\left(\mathbf{q}^{n-1}, \mathrm{t}^{n-1}\right) \tag{64}$$

$$\mathbf{h}_i = \mathbf{g}\left(\mathbf{q}^{n-1} + \Delta\mathrm{t}\sum_{j=1}^{i-1} a_{ij}\mathbf{h}_j, \mathrm{t}^{n-1} + c_i\Delta\mathrm{t}\right) \quad \text{for } 1 < i \leq s, \tag{65}$$

and the scalar constants a_{ij}, b_i, and $c_i \in \mathbb{R}$ are specific to each Runge–Kutta scheme. The scheme is explicit (only dependent on the state at past stages) if $a_{ij} = 0\ \forall j \geq i$, and implicit (dependent on the future state to be solved for) otherwise. These coefficients are generally presented in the form of a "Butcher tableau,"

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \dots & a_{1s} \\ c_2 & a_{11} & a_{22} & \dots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \dots & a_{ss} \\ \hline & b_1 & b_2 & \dots & b_s \end{array}$$

A Butcher tableau is provided for each Runge-Kutta method included in `PERFORM`.

- Classic RK4 ($s = 4$, explicit, fourth-order)

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

- Strong stability preserving RK3 ($s = 3$, explicit, third-order)

0	0	0	0
1	1	0	0
1/2	1/4	1/4	0
	1/6	1/6	2/3

3.2.2 Linear Multi-step Schemes

The residual function for a general s-stage linear multi-step scheme is given by

$$\mathbf{r}^n = a_0\mathbf{q}^n + \sum_{i=1}^{s} a_i\mathbf{q}^{n-i} - \Delta t b_0 \mathbf{g}\left(\mathbf{q}^n,\ t^n\right) - \Delta t \sum_{i=1}^{s} b_i \mathbf{g}\left(\mathbf{q}^{n-i},\ t^{n-i}\right), \tag{66}$$

where the scalar constants a_i and b_i are specific to each linear multi-step scheme. The scheme is explicit if $b_0 = 0$, and implicit otherwise. Listed below are coefficients for the linear multi-step schemes implemented in `PERFORM`.

- Backwards differentiation formulae (BDF, implicit): for all schemes, $b_1 = b_2 = \ldots = b_s = 0$, and $b_0 = 1$.
  - BDF1/implicit Euler (first-order): $a_0 = 1$, $a_1 = -1$
  - BDF2 (second-order): $a_0 = 3/2$, $a_1 = -2$, $a_2 = 1/2$
  - BDF3 (third-order): $a_0 = 11/16$, $a_1 = -3$, $a_2 = 3/2$, $a_3 = -1/3$
  - BDF4 (fourth-order): $a_0 = 25/12$, $a_1 = -4$, $a_2 = 3$, $a_3 = -4/3$, $a_4 = 1/4$

3.2.3 Solving Implicit Schemes via Newton's Method

Implicit time integration schemes naturally lead to a residual which takes the form of a non-linear system of equations, which requires an iterative solution. In `PERFORM`, we use Newton's method for solving this system. First, Newton's method replaces the solution at the next time step $\mathbf{q}^n$ with an intermediate solution $\mathbf{q}_k^n$, where k is the Newton iterate number. For example, the BDF1/implicit Euler residual would take the form

$$\mathbf{r}\left(\mathbf{q}_k^n\right) = \mathbf{q}_k^n - \mathbf{q}^{n-1} - \Delta t \mathbf{g}\left(\mathbf{q}_k^n\right) \tag{67}$$

Then, as a general step for any implicit scheme, the iterative update to the intermediate solution is given by

$$\frac{\partial \mathbf{r}}{\partial \mathbf{q}}(\mathbf{q}_{k-1}^n)\left(\mathbf{q}_k^n - \mathbf{q}_{k-1}^n\right) = -\mathbf{r}\left(\mathbf{q}_{k-1}^n\right) \tag{68}$$

with increased iteration, the residual converges to zero and the solution at the next physical time step is set to the last iterative solution, i.e. $\mathbf{q}^n \leftarrow \mathbf{q}_k^n$. The initial guess for the initial iterative solution is taken as $\mathbf{q}_0^n \leftarrow \mathbf{q}^{n-1}$.

3.2.4 Solving Implicit Schemes via Dual Time-stepping

Dual time-stepping is an alternative method for solving stiff implicit systems. Among other benefits, it allows us to solve directly for the primitive state

$$\mathbf{q}_p = [p \quad u \quad T \quad Y_l]^\top \tag{69}$$

This fact is crucial for efficiently calculating thermodynamic and transport properties of TPGs and real gases. Dual time stepping begins by adding a pseudo-time derivative to the governing ODE

$$\mathbf{\Gamma}\frac{\partial \mathbf{q}_p}{\partial \tau} + \frac{\mathrm{d}\mathbf{q}}{\mathrm{dt}} = \mathbf{g}\left(\mathbf{q}_p\right) \tag{70}$$

where the Jacobian $\mathbf{\Gamma} := \partial\mathbf{q}/\partial\mathbf{q}_p$ can be computed analytically (given in Section 5). We abuse notation in denoting $\mathbf{g}\left(\mathbf{q}_p\right) = \mathbf{g}\left(\mathbf{q}\right)$; in reality, a combination of both the primitive and conservative variables are used to compute the RHS terms. Next, the time derivatives are discretized in terms of intermediate iterative conservative and primitive solutions $\mathbf{q}^n_{p,k}$ and $\mathbf{q}^n_k$, respectively. For example, discretizing the pseudo-time derivative with a first-order finite difference scheme and the physical time derivative with BDF2 results in

$$\mathbf{\Gamma}^n_{k-1}\frac{\mathbf{q}^n_{p,k} - \mathbf{q}^n_{p,k-1}}{\Delta\tau} + \frac{3\mathbf{q}^n_k - 4\mathbf{q}^{n-1} + \mathbf{q}^{n-2}}{2\Delta\mathrm{t}} = \mathbf{g}\left(\mathbf{q}^n_{p,k}\right) \tag{71}$$

The terms $\mathbf{q}^n_k$ and $\mathbf{g}\left(\mathbf{q}^n_{p,k}\right)$ are linearized about $\mathbf{q}^n_{p,k-1}$

$$\mathbf{q}^n_k \approx \mathbf{q}^n_{k-1} + \left(\frac{\partial \mathbf{q}}{\partial \mathbf{q}_p}\right)^n_{k-1}(\mathbf{q}^n_{p,k} - \mathbf{q}^n_{p,k-1}) = \mathbf{q}^n_{k-1} + \mathbf{\Gamma}^n_{k-1}(\mathbf{q}^n_{p,k} - \mathbf{q}^n_{p,k-1}), \tag{72}$$

$$\mathbf{g}\left(\mathbf{q}^n_{p,k}\right) \approx \mathbf{g}\left(\mathbf{q}^n_{p,k-1}\right) + \mathbf{J}^n_{p,k-1}\mathbf{\Gamma}^n_{k-1}(\mathbf{q}^n_{p,k} - \mathbf{q}^n_{p,k-1}), \tag{73}$$

where the Jacobian $\mathbf{J}^n_{p,k} = (\partial\mathbf{g}/\partial\mathbf{q}_p)^n_k$ can be computed (approximately) analytically. The Jacobians of the various flux and source terms are given in Section 5.

Inserting these approximations into Eq. 71 and rearranging terms arrives at

$$\left(\left(\frac{1}{\Delta\tau} + \frac{1}{\Delta\mathrm{t}}\right)\mathbf{I} - \frac{3}{2\Delta t}\mathbf{J}^n_{p,k-1}\right)\mathbf{\Gamma}^n_{k-1}\left(\mathbf{q}^n_{p,k} - \mathbf{q}^n_{p,k-1}\right) = -\frac{3\mathbf{q}^n_{k-1} - 4\mathbf{q}^{n-1} + \mathbf{q}^{n-2}}{2\Delta\mathrm{t}} + \mathbf{g}\left(\mathbf{q}^n_{p,k-1}\right) \tag{74}$$

Similar to Newton's method, this linear system of equations is solved for the change in the *primitive* state. Of course, this method comes with some additional computational costs (particularly the calculation of the Jacobian $\mathbf{\Gamma}$ and the associated matrix-matrix multiplications), but allows us improved access to the primitive state as it evolves. This permits simple applications of filters and limiters to relevant engineering quantities, and eases the subsequent calculation of thermodynamic and transport quantities.

3.3 Flux Schemes

Numerical inviscid and viscous fluxes available in PERFORM are described here; this section will be updated as new schemes are implemented.

3.3.1 Roe Flux

The Roe flux differencing scheme solves for the numerical inviscid flux at the $(i+1/2)$th face given by

$$\widetilde{\mathbf{f}}_{x_{i+1/2}} = \frac{1}{2}\left[\mathbf{f}_i + \mathbf{f}_{i+1}\right] - \frac{1}{2}\left[\left|\frac{\partial \mathbf{f}}{\partial \mathbf{q}}\right| \delta\mathbf{q}\right]_{x_{i+1/2}} \tag{75}$$

We hereafter denote the Jacobian of the flux with respect to the conservative state as $\mathbf{J}_{x_{i+1/2}} := (\partial \mathbf{f}/\partial \mathbf{q})_{x_{i+1/2}}$. The last term in Eq. 75 can be rewritten in terms of the primitive state

$$\left[|\mathbf{J}|\,\delta\mathbf{q}\right]_{x_{i+1/2}} = \left[\left|\frac{\partial \mathbf{f}}{\partial \mathbf{q}_p}\frac{\partial \mathbf{q}_p}{\partial \mathbf{q}}\right| \frac{\partial \mathbf{q}}{\partial \mathbf{q}_p}\delta\mathbf{q}_p\right]_{x_{i+1/2}} \tag{76}$$

$$= \left[\left|\mathbf{J}_p\boldsymbol{\Gamma}^{-1}\right| \boldsymbol{\Gamma}\delta\mathbf{q}_p\right]_{x_{i+1/2}} \tag{77}$$

$$= \left[\boldsymbol{\Gamma}\left|\boldsymbol{\Gamma}^{-1}\mathbf{J}_p\right| \delta\mathbf{q}_p\right]_{x_{i+1/2}} \tag{78}$$

This term is evaluated with respect to the "Roe average" state, defined based on the density, velocity, and stagnation enthalpy

$$\rho_{x_{i+1/2}} = \sqrt{\rho_{R,i}\rho_{L,i+1}} \tag{79}$$

$$u_{x_{i+1/2}} = \frac{u_{R,i}\sqrt{\rho_{R,i}} + u_{L,i+1}\sqrt{\rho_{L,i+1}}}{\sqrt{\rho_{R,i}} + \sqrt{\rho_{L,i+1}}} \tag{80}$$

$$h^0_{x_{i+1/2}} = \frac{h^0_{R,i}\sqrt{\rho_{R,i}} + h^0_{L,i+1}\sqrt{\rho_{L,i+1}}}{\sqrt{\rho_{R,i}} + \sqrt{\rho_{L,i+1}}} \tag{81}$$

Here, the subscripts L and R for the ith cell denote the reconstruction of the state at the ith cell's left and right faces, respectively. The topic of face reconstructions is discussed in Section 3.4.

As the equivalent Roe average procedure applied to static pressure and temperature results in a state which does not match the Roe average density and enthalpy, PERFORM utilizes an iterative procedure which incrementally adjusts the pressure and temperature using the derivatives of density and stagnation enthalpy with respect to temperature and pressure. Upon convergence, a complete primitive state is achieved which agrees with the fixed Roe average density and stagnation enthalpy. This Roe average state at the face is also used later for computing the viscous fluxes in Section 3.3.2.

The analytical form of the matrix $\boldsymbol{\Gamma}|\boldsymbol{\Gamma}^{-1}\mathbf{J}_p|$ is given in Section 5. With this term computed, the inviscid flux vectors $\mathbf{f}_i$ and $\mathbf{f}_{i+1}$ can be easily computed from the conservative and primitive state at each cell, and the numerical flux $\widetilde{\mathbf{f}}_{x_{i+1/2}}$ computed from Eq. 75.

3.3.2 Viscous Fluxes

The viscous fluxes are fairly simple, only requiring a means of computing gradients at the cell faces. For the uniform meshes used in PERFORM, this is accomplished by a simple second-order finite difference

scheme (which is just the slope between adjacent cell-centered values). For a given quantity α, this is given by

$$\nabla \alpha_{x_{i+1/2}} = \frac{\alpha_{i+1} - \alpha_i}{x_{i+1} - x_i} \tag{82}$$

The gradients of velocity, temperature, and species can thus be computed to calculate the stress tensor, heat flux, and diffusion velocity. Some measure of the average state at the face must also be used to complete the calculation of the viscous flux tensor $\mathbf{f}_v$; for example, the arithmetic mean of the left/right face reconstructions discussed in Section 3.4, or the Roe average when using the Roe scheme to compute the inviscid fluxes.

3.4 Face Reconstruction

For a first-order accurate flux scheme, the state at the left and right face of a cell is equal to the cell-centered state, i.e. the state is assumed to be uniform throughout the cell volume. In order to achieve higher-order accuracy in computing face fluxes, gradients must be computed at the cell centers and used to calculate a more accurate reconstruction of the state at the left and right cell faces.

Again, thanks to the uniform meshes used in `PERFORM`, these gradients are just computed from central finite difference schemes. For example, the fourth-order gradient of the quantity α at the ith cell is computed by

$$\nabla\alpha_i = \frac{\alpha_{i-2} - 8\alpha_{i-1} + 8\alpha_{i+1} - \alpha_{i+2}}{12\Delta x} \tag{83}$$

The reconstructed primitive state at the ith cell's left and right face thus takes the form

$$\mathbf{q}_{p,L,i} = \mathbf{q}_{p,i} - \mathbf{\Phi}_i \nabla\mathbf{q}_{p,i}\left(\frac{\Delta x}{2}\right) \tag{84}$$

$$\mathbf{q}_{p,R,i} = \mathbf{q}_{p,i} + \mathbf{\Phi}_i \nabla\mathbf{q}_{p,i}\left(\frac{\Delta x}{2}\right) \tag{85}$$

where the matrix $\mathbf{\Phi}_i = diag(\phi_p, \phi_u, \phi_T, \phi_{Y_l})_i$ is the *gradient limiter term*, which is discussed in Section 3.5. The numerical flux at each face may thus be computed using these face reconstructions.

3.5 Gradient Limiters

Although higher-order schemes can help improve the resolution of strong gradients, they often produce non-monotonic face reconstructions which can lead to highly oscillatory or unstable solutions. As such, the gradient limiting term $\mathbf{\Phi}_i$ introduced in Eq. 83 is crucial for improving the behavior of these reconstructions. Those methods implemented in `PERFORM` are detailed below, and will be updated as new methods are implemented.

3.5.1 Barth and Jespersen's Limiter

The limiter by Barth and Jespersen aims to limit the face reconstruction such that no new minima or maxima are created, preserving the monotonicity of the solution between two cell-centered averages. In order to do that, the method first determines the maximuma and minimum cell averages between each cell and its neighbors,

$$\mathbf{q}_{p,i}^{min} = \min\left(\min\left(\mathbf{q}_{p,i}, \mathbf{q}_{p,i-1}\right), \mathbf{q}_{p,i+1}\right) \tag{86}$$

$$\mathbf{q}_{p,i}^{max} = \max\left(\max\left(\mathbf{q}_{p,i}, \mathbf{q}_{p,i-1}\right), \mathbf{q}_{p,i+1}\right) \tag{87}$$

Next, the unconstrained face reconstructions are computed

$$\mathbf{q}'_{p,L,i} = \mathbf{q}_{p,i} - \nabla\mathbf{q}_{p,i}\left(\frac{\Delta x}{2}\right) \tag{88}$$

$$\mathbf{q}'_{p,R,i} = \mathbf{q}_{p,i} + \nabla\mathbf{q}_{p,i}\left(\frac{\Delta x}{2}\right) \tag{89}$$

With these quantities, the limiter for both faces is computed, for example at the cell's left face

$$\mathbf{\Phi}_{L,i} = \begin{cases} \min\left(1, \frac{\mathbf{q}_{p,i}^{max} - \mathbf{q}_{p,i}}{\mathbf{q}'_{p,L,i} - \mathbf{q}_{p,i}}\right) & \mathbf{q}'_{p,L,i} - \mathbf{q}_{p,i} > 0 \\ \min\left(1, \frac{\mathbf{q}_{p,i}^{min} - \mathbf{q}_{p,i}}{\mathbf{q}'_{p,L,i} - \mathbf{q}_{p,i}}\right) & \mathbf{q}'_{p,L,i} - \mathbf{q}_{p,i} < 0, \\ 1 & \mathbf{q}'_{p,L,i} - \mathbf{q}_{p,i} = 0, \end{cases} \tag{90}$$

The same calculations may be computed at the right face, replacing each L with an R in the above formulation. The gradient limiter for the entire cell is thus chosen to be the most restrictive between those computed at the left and right face,

$$\mathbf{\Phi}_i = min(\mathbf{\Phi}_{L,i}, \mathbf{\Phi}_{R,i}), \tag{91}$$

Most notably, the $\min(1, y)$ function in the Barth and Jespersen limiter results in a non-differentiable limiting function. This can degrade the convergence of iterative solvers.

3.5.2 Venkatakrishnan's Limiter

Venkatakrishnan's limiter attempts to eliminate the convergence issues in Barth and Jespersen's limiter by replacing the $\min(1, y)$ with a differentiable function, namely

$$\psi(x) = \frac{x^2 + 2x}{x^2 + x + 2} \tag{92}$$

Other than this replacement, the limiter calculations are identical to those in Barth and Jespersen's limiter. The limiting function is thus differentiable, and exhibits improved convergence. The main drawback of this limiter is the fact that it may apply limiting (i.e. $\phi_{\alpha,i} < 1.0$) in uniform regions of the field. Some modifications have been developed to counteract this issue, but they have not yet been implemented in `PERFORM`.

3.6 Boundary Conditions

Boundary conditions are enforced by an explicit ghost cell formulation. Below, we detail the available boundary conditions in PERFORM. Throughout this section, we denote the quantities at the inlet ghost cell with the zero subscript (e.g. α_0), and at the outlet ghost cell with the $N_c + 1$ subscript (e.g. α_{N_c+1}).

As incoming/outgoing characteristics and their associated Riemann invariants are important for several boundary conditions, we state their definitions here. To begin, the 1D Euler equations provide the following relations for the characteristic variables

$$d\rho - \frac{dp}{c^2} = 0 \quad for \quad \frac{dx}{dt} = u \tag{93}$$

$$du + \frac{dp}{\rho c} = 0 \quad for \quad \frac{dx}{dt} = u + c \tag{94}$$

$$du - \frac{dp}{\rho c} = 0 \quad for \quad \frac{dx}{dt} = u - c \tag{95}$$

$$dY_i = 0 \quad for \quad \frac{dx}{dt} = u \tag{96}$$

An alternative formulation for the first characteristic (the entropy wave), which is useful when working with temperature as a primitive variable, is given by $dT - dp/(\rho c_p) = 0$.

Under the assumption of isentropic flow, these relations can be integrated to provide the associated Riemann invariants

$$\begin{bmatrix} J \\ J^+ \\ J^- \\ J^{Y_i} \end{bmatrix} = \begin{bmatrix} \frac{p}{\rho^\gamma} \\ u + \frac{2c}{\gamma - 1} \\ u - \frac{2c}{\gamma - 1} \\ Y_i \end{bmatrix} \tag{97}$$

which are constant along their associated characteristics.

For non-reflective boundaries, the linearization of the characteristic equations assuming small local perturbations at the boundaries allows us to compute the characteristic variables,

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} T - \frac{p}{\rho c_p} \\ u + \frac{p}{\rho c} \\ u - \frac{p}{\rho c} \\ Y_i \end{bmatrix} \tag{98}$$

Here, the quantities $\overline{\rho c}$ and $\overline{\rho c_p}$ represent mean quantities from which it is assumed the state only experiences small perturbations.

3.6.1 Inlet Boundary Conditions

3.6.1.1 Specified Inlet Stagnation Pressure and Stagnation Temperature This boundary condition specifies the stagnation pressure and stagnation temperature at the inlet.

$$\frac{T^0}{T} = 1 + \frac{\gamma - 1}{2} M^2 \tag{99}$$

$$\frac{p^0}{p} = \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{\gamma}{\gamma-1}} \tag{100}$$

$$= \left(\frac{T^0}{T}\right)^{\frac{\gamma}{\gamma-1}} \tag{101}$$

The local sound speed c, given a specific gas constant R for the mixture, can be written in terms of the stagnation temperature and Mach number,

$$c^2 = \gamma R T \tag{102}$$

$$= \frac{\gamma R T^0}{1 + \frac{\gamma-1}{2} M^2} \tag{103}$$

The Riemann invariant J^- of the outgoing $u - c$ characteristic is given by,

$$J^- = u - \frac{2c}{\gamma - 1} \tag{104}$$

$$= Mc - \frac{2c}{\gamma - 1} \tag{105}$$

Squaring both sides of this relationship results in,

$$(J^-)^2 = c^2 M^2 - \frac{4c^2}{\gamma - 1} M + \frac{4c^2}{(\gamma - 1)^2} \tag{106}$$

Substituting in Eq. 102 and rearranging,

$$(J^-)^2 \left(1 + \frac{\gamma - 1}{2} M^2\right) = \gamma R T^0 M^2 - \frac{4\gamma R T^0}{\gamma - 1} M + \frac{4\gamma R T^0}{(\gamma - 1)^2} \tag{107}$$

$$0 = \left(\gamma R T^0 - \frac{\gamma - 1}{2} (J^-)^2\right) M^2 - \frac{4\gamma R T^0}{\gamma - 1} M + \frac{4\gamma R T^0}{(\gamma - 1)^2} - (J^-)^2 \tag{108}$$

This quadratic equation can be solved for the Mach number using the following relationships,

$$a = \gamma R T^0 - \frac{\gamma - 1}{2} (J^-)^2 \tag{109}$$

$$b = -\frac{4\gamma R T^0}{\gamma - 1} \tag{110}$$

$$c = \frac{4\gamma R T^0}{(\gamma - 1)^2} - (J^-)^2 \tag{111}$$

$$M = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{112}$$

producing two solutions. Obviously, if there is one positive solution then that is the physically-relevant solution. If there are two positive solutions, the smaller one is chosen. The static pressure p and static temperature T can be solved for from Eqs. 99 and 100.

For computing these quantities at an inlet, only the stagnation temperature and stagnation pressure are known, leaving the Riemann invariant J^- of the outgoing characteristic as an unknown. Assuming a uniform grid, this can be extrapolated from the interior points 1 and 2 to the ghost cell by linear extrapolation,

$$J_0^- = 2J_1^- - J_2^- \tag{113}$$

Substituting this into Eq. 107, solving for the Mach number at the inlet exterior, and computing the primitive and conservative state at the boundary then allows for the calculation of boundary fluxes.

This inlet boundary condition results in reflections of acoustic waves and should not be used with unsteady calculations with significant system acoustics.

3.6.1.2 Specified Inlet Full State This boundary condition specifies the full primitive state (static pressure, velocity, temperature, and species mass fractions) at the inlet ghost cell. Of course, this method is only truly appropriate at a supersonic inlet, but can be useful for testing the response of outlet boundary conditions to upstream perturbations originating from the inlet.

3.6.1.3 Non-reflective Inlet The non-reflective inlet operates on the assumption of a fixed "mean" upstream state, about which the instantaneous inlet state is simply a perturbation. This can be thought of as the state infinitely far upstream from the inlet. Keeping this upstream state fixed essentially fixes the incoming characteristics and allows the outgoing characteristic to exit the domain without reflections. As such, in the absence of any acoustic sources or outlet perturbations the flow at the inlet will always return to the mean upstream state.

As there are three incoming characteristics at the inlet, we specify three exterior flow variables. In this case, we specify the upstream static pressure p_{up}, the upstream chemical composition $Y_{l,up}$, and the upstream temperature T_{up}. These upstream quantities are *not* the quantities at the inlet ghost cell, but can be thought of as the quantities at negative infinity. We determine the mean quantities $(\overline{\rho c})_{up}$ and $(\overline{\rho c_p})_{up}$ from the mean solution at the inlet.

There is only one outgoing characteristic at a subsonic inlet (w_3), which is extrapolated from the interior. We apply a mean flow approximation for the incoming $u + c$ characteristic in terms of the inlet ghost cell velocity u_0 and pressure p_0,

$$u_0 + \frac{p}{(\overline{\rho c})_{up}} = \frac{p_{up}}{(\overline{\rho c})_{up}} \tag{114}$$

$$u_0 = \frac{p_{up} - p_0}{(\overline{\rho c})_{up}} \tag{115}$$

This relation can be substituted into the formulation for the extrapolated $u - c$ characteristic variable,

$$w_3 = \frac{p_{up} - 2p_0}{(\overline{\rho c})_{up}} \tag{116}$$

$$p_0 = \frac{u_0 - w_3(\overline{\rho c})_{up}}{2} \tag{117}$$

We also provide a mean flow approximation for the incoming entropy wave,

$$T_0 - \frac{p_0}{(\overline{\rho c_p})_{up}} = T_{up} - \frac{p_{up}}{(\overline{\rho c_p})_{up}} \tag{118}$$

$$T_0 = T_{up} + \frac{p_0 - p_{up}}{(\overline{\rho c_p})_{up}} \tag{119}$$

After calculating the inlet pressure from Eq. 116, the inlet temperature can be computed from Eq. 118, and the inlet velocity can be computed from Eq. 114. The inlet species mass fraction is completely specified, so this completes the inlet primitive state.

For computing the user inputs, the density, sound speed, and mixture specific heat capacity at constant pressure can all be ripped directly from the last interior cell of the mean solution. We compute the upstream pressure by,

$$p_{up} = p_0 + u_0(\overline{\rho c})_{up} \tag{120}$$

and then compute the upstream temperature by,

$$T_{up} = T_0 + \frac{p_{up} - p_0}{(\overline{\rho c_p})_{up}} \tag{121}$$

where the pressure p_0, velocity u_0, and temperature T_0 are taken from the first cell of the mean solution.

3.6.2 Outlet Boundary Conditions

3.6.2.1 Specified Outlet Static Pressure In the case of a subsonic outflow, one may specify a fixed outlet ghost cell pressure p_{N_c+1} and compute the remainder of the exterior state from the outgoing entropy wave (associated with the u characteristic) and the Riemann invariant associated with the $u + c$ characteristic. Recall that the outgoing entropy wave is calculated as,

$$J = \frac{p}{\rho^\gamma} \tag{122}$$

and the Riemann invariant J^+ of the outgoing $u + c$ characteristic is calculated as,

$$J^+ = u + \frac{2c}{\gamma - 1} \tag{123}$$

These can be extrapolated from the interior to the exterior using the first-order linear extrapolation

$$J_{N_c+1} = 2J_{N_c} - J_{N_c-1} \tag{124}$$
$$J^+_{N_c+1} = 2J^+_{N_c} - J^+_{N_c-1}, \tag{125}$$

From the entropy wave, and knowing the fixed back pressure, the outlet ghost cell density can be calculated,

$$\rho_{N_c+1} = \left(\frac{p_{N_c+1}}{J_{N_c}}\right)^{1/\gamma} \tag{126}$$

With this, the outlet ghost cell sound speed can be calculated as

$$c_{N_c+1} = \sqrt{\frac{\gamma p_{N_c+1}}{\rho_{N_c+1}}} \tag{127}$$

and the outlet ghost cell velocity can be computed from the extrapolated Riemann invariant of the $u + c$ characteristic,

$$u_{N_c+1} = J^+_{N_c+1} - \frac{2c_{N_c+1}}{\gamma - 1} \tag{128}$$

The remaining primitive and conservative variables can be computed from gas relations, completing the outlet ghost cell state.

This outlet boundary condition results in reflections of acoustic waves and should not be used with unsteady calculations with significant system acoustics.

3.6.2.2 Non-reflective Outlet The non-reflective outlet follows a similar framework to that of the non-reflective inlet, modified with respect to the incoming and outgoing characteristics at the outlet. In this case, as there is only one incoming characteristic for subsonic flow, only a back pressure p_{back} is specified along with the mean quantities $(\overline{\rho c_p})_{back}$ and $(\overline{\rho c})_{back}$. Again, recall that this back pressure is *not* the static pressure at the outlet ghost cell, but can be thought of as the static pressure at infinity.

The outgoing characteristics (w_1, w_2, and w_4) are directly extrapolated from the interior cells to the outlet ghost cell. The final necessary equation is supplied by the mean flow relation

$$u_{N_c+1} - \frac{p_{N_c+1}}{(\overline{\rho c})_{back}} = -\frac{p_{back}}{(\overline{\rho c})_{back}} \tag{129}$$

With these relationships, we can solve for the primitive state, beginning with

$$u_{N_c+1} = \frac{p_{N_c+1} - p_{back}}{(\overline{\rho c})_{back}} \tag{130}$$

and substituting this into w_2,

$$w_2 = \frac{2p_{N_c+1} - p_{back}}{(\overline{\rho c})_{back}} \tag{131}$$

$$p_{N_c+1} = \frac{w_2(\overline{\rho c})_{back} + p_{back}}{2} \tag{132}$$

This can be then substituted into Eq. 130 to compute the outlet ghost cell velocity, and into the equation for w_1 to compute the outlet ghost cell temperature. This completes the outlet state.

For determining the user inputs, the density, sound speed, and mixture specific heat capacity at constant pressure can all be taken directly from the last interior cell of the mean solution. The back pressure should be calculated from Eq. 129, where the velocity u and static pressure p are taken from the last interior cell. Thus, we calculate the back pressure as,

$$p_{back} = p_{N_c+1} - u_{N_c+1}(\overline{\rho c})_{back} \tag{133}$$

4 Reduced-Order Modeling

This section will be updated shortly, check back soon!

5 Jacobians

This section will be updated shortly, check back soon!