Notes Towards PhD

Kyle B. Thompson

October 24, 2016

1 Adjoint Derivation

The FUN3D adjoint derivation is given in Eric Nielson's PhD Thesis. Starting with forming the Lagrangian as

$$L(\mathbf{D}, \mathbf{Q}, \mathbf{X}, \mathbf{\Lambda}) = f(\mathbf{D}, \mathbf{Q}, \mathbf{X}) + \mathbf{\Lambda}^T \mathbf{R}(\mathbf{D}, \mathbf{Q}, \mathbf{X}) \tag{1}$$

Where $\mathbf{R}$ is the residual of the flow equations. Differentiating with respect to the design variables $\mathbf{D}$ yields

$$\frac{\partial L}{\partial \mathbf{D}} = \left\{ \frac{\partial f}{\partial \mathbf{D}} + \left[\frac{\partial \mathbf{X}}{\partial \mathbf{D}}\right]^T \frac{\partial f}{\partial \mathbf{X}} \right\} + \left[\frac{\partial \mathbf{Q}}{\partial \mathbf{D}}\right]^T \left\{ \frac{\partial f}{\partial \mathbf{Q}} + \left[\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\right]^T \mathbf{\Lambda} \right\} + \left\{ \left[\frac{\partial \mathbf{R}}{\partial \mathbf{D}}\right]^T + \left[\frac{\partial \mathbf{X}}{\partial \mathbf{D}}\right]^T \left[\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right]^T \right\} \mathbf{\Lambda} \tag{2}$$

To eliminate the dependence of conserved variables $\mathbf{Q}$ on the design variables, we solve the adjoint equation

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\right]^T \mathbf{\Lambda} = -\frac{\partial f}{\partial \mathbf{Q}} \tag{3}$$

Where the Lagrange multipliers (also known as costate variables), $\mathbf{\Lambda}$ are the cost function dependence on the residual

$$\mathbf{\Lambda} = -\frac{\partial f}{\partial \mathbf{R}} \tag{4}$$

This can ultimately be used to error estimation and sensitivity analysis for design optimization. With the second term in eq. (2) eliminated, the derivative of the Lagrangian becomes

$$\frac{\partial L}{\partial \mathbf{D}} = \left\{ \frac{\partial f}{\partial \mathbf{D}} + \left[\frac{\partial \mathbf{X}}{\partial \mathbf{D}}\right]^T \frac{\partial f}{\partial \mathbf{X}} \right\} + \left\{ \left[\frac{\partial \mathbf{R}}{\partial \mathbf{D}}\right]^T + \left[\frac{\partial \mathbf{X}}{\partial \mathbf{D}}\right]^T \left[\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right]^T \right\} \mathbf{\Lambda} \tag{5}$$

By solving the adjoint equation in Eq. (3)) to obtain the costate variable vector, $\mathbf{\Lambda}$, we can now use a non-linear optimizer to determine the optimum set of design variables, $\mathbf{D}^*$. This can be done using **PORT** or **KSOPT** in FUN3D, as well as a host of other non-linear optimizers.

2 Decoupled Adjoint

The purpose of this is to show the relationship between the costate variable for total density λ_ρ to the costate variables for species densities λ_{ρ_s}. Beginning with the definition of the Adjoint Equations:

$$\left(\frac{\partial R}{\partial Q}\right)^T \lambda = \frac{\partial F}{\partial Q} \tag{6}$$

Where the R is the residual of the governing equations, Q is the vector of conserved variables, and F is the cost function (i.e. lift, drag, etc.). Note that the first term is simply the transpose of the jacobian multiplied by the costate variable vector λ, and can be written as:

$$\left(\frac{\partial R}{\partial Q}\right)_i^T \lambda = \sum_{j=1}^{N_{eq}} \left(\frac{\partial R_j}{\partial Q_i}\right) \lambda_j \tag{7}$$

Suppose we define the system of equations in two different ways. The first system, which we'll call the "meanflow system", consists of 5 equations:

$$R = \begin{pmatrix} R_\rho \\ R_{\rho u} \\ R_{\rho v} \\ R_{\rho w} \\ R_{\rho E} \end{pmatrix}, \quad \lambda = \begin{pmatrix} \lambda_\rho \\ \lambda_{\rho u} \\ \lambda_{\rho v} \\ \lambda_{\rho w} \\ \lambda_{\rho E} \end{pmatrix} \tag{8}$$

The second system consists of the full system of equations:

$$R = \begin{pmatrix} R_{\rho_1} \\ \vdots \\ R_{\rho_s} \\ R_{\rho u} \\ R_{\rho v} \\ R_{\rho w} \\ R_{\rho E} \end{pmatrix}, \quad \lambda = \begin{pmatrix} \lambda_{\rho_1} \\ \vdots \\ \lambda_{\rho_s} \\ \lambda_{\rho u} \\ \lambda_{\rho v} \\ \lambda_{\rho w} \\ \lambda_{\rho E} \end{pmatrix} \tag{9}$$

By making the approximation that the mass fraction c_s is constant, we can show that the full system of equations reduces to the meanflow system. By this approximation the derivatives with respect to species density become:

$$\frac{\partial R}{\partial \rho} = \frac{\partial R}{\partial \rho_s}\frac{\partial \rho_s}{\partial \rho} = c_s \left(\frac{\partial R}{\partial \rho_s}\right) \tag{10}$$

Thus, for a single row of the full system:

$$\left(\frac{\partial R}{\partial Q}\right)_{\rho_s}^T \lambda = \sum_{j=1}^{N_{full}} \left(\frac{\partial R_j}{\partial \rho}\right) \frac{\lambda_j}{c_s} = \frac{1}{c_s}\left(\frac{\partial F}{\partial \rho}\right) \tag{11}$$

After cancelling the mass fractions, this allows the first row of the full system to be equated to the first row of the meanflow system:

$$\sum_{j=1}^{N_{full}} \left(\frac{\partial R_j}{\partial \rho}\right) \lambda_j = \sum_{j=1}^{N_{meanflow}} \left(\frac{\partial R_j}{\partial \rho}\right) \lambda_j \tag{12}$$

Expanding this out, it becomes clear many terms cancel:

$$\frac{\partial R_{\rho_1}}{\partial \rho}\lambda_{\rho_1} + \cdots + \frac{\partial R_{\rho_s}}{\partial \rho}\lambda_{\rho_s} + \frac{\partial R_{\rho u}}{\partial \rho}\lambda_{\rho u} + \frac{\partial R_{\rho v}}{\partial \rho}\lambda_{\rho v} + \frac{\partial R_{\rho w}}{\partial \rho}\lambda_{\rho w} + \frac{\partial R_{\rho E}}{\partial \rho}\lambda_{\rho E} =$$
$$\frac{\partial R_{\rho}}{\partial \rho}\lambda_{\rho} + \frac{\partial R_{\rho u}}{\partial \rho}\lambda_{\rho u} + \frac{\partial R_{\rho v}}{\partial \rho}\lambda_{\rho v} + \frac{\partial R_{\rho w}}{\partial \rho}\lambda_{\rho w} + \frac{\partial R_{\rho E}}{\partial \rho}\lambda_{\rho E} \quad (13)$$

$$\frac{\partial R_{\rho_1}}{\partial \rho}\lambda_{\rho_1} + \cdots + \frac{\partial R_{\rho_s}}{\partial \rho}\lambda_{\rho_s} = \frac{\partial R_{\rho}}{\partial \rho}\lambda_{\rho} \quad (14)$$

Finally, because the individual species mass fluxes must sum to the total mass flux:

$$\sum_{s=1}^{N_{species}} R_{\rho_s} = R_{\rho} \quad (15)$$

Eqn (14) can be rewritten as:

$$\frac{\partial R_{\rho_1}}{\partial \rho}\lambda_{\rho_1} + \cdots + \frac{\partial R_{\rho_s}}{\partial \rho}\lambda_{\rho_s} = \frac{\partial R_{\rho_1}}{\partial \rho}\lambda_{\rho} + \cdots + \frac{\partial R_{\rho_s}}{\partial \rho}\lambda_{\rho} \quad (16)$$

Which implies that the species mass costate variables are all equal to the total mass costate variable, yielding:

$$\lambda_{\rho} = \lambda_{\rho_s} \quad (17)$$
$$d\lambda_{\rho} = d\lambda_{\rho_s} \quad (18)$$

A. Decoupled Flux Derivation

For the Roe Flux Difference Splitting scheme, the species mass fluxes are given by:

$$F_{\rho_s} = \frac{\rho_s^L \overline{U}^L + \rho_s^R \overline{U}^R}{2} - \frac{\tilde{c}_s(\lambda_1 dv_1 + \lambda_2 dv_2) + \lambda_3 dv_{3_s})}{2} \tag{19}$$

$$dv_1 = \frac{p^R - p^L + \tilde{\rho}\tilde{a}(\overline{U}^R - \overline{U}^L)}{\tilde{a}^2} \tag{20}$$

$$dv_2 = \frac{p^R - p^L - \tilde{\rho}\tilde{a}(\overline{U}^R - \overline{U}^L)}{\tilde{a}^2} \tag{21}$$

$$dv_{3_s} = \frac{\tilde{a}^2(\rho_s^R - \rho_s^L) - \tilde{c}_s(p^R - p^L)}{\tilde{a}^2} \tag{22}$$

$$\lambda_1 = \mid \overline{\mathbf{U}} + \tilde{a} \mid, \quad \lambda_2 = \mid \overline{\mathbf{U}} - \tilde{a} \mid, \quad \lambda_3 = \mid \overline{\mathbf{U}} \mid \tag{23}$$

where the ˜ notation signifies a roe-averaged quantity, given by:

$$\tilde{\mathbf{U}} = w\tilde{\mathbf{U}}^L + (1-w)\tilde{\mathbf{U}}^R \tag{24}$$

$$w = \frac{\tilde{\rho}}{\tilde{\rho} + \rho^R} \tag{25}$$

The species mass fluxes must sum to the total mass flux; thus, the total mixture mass flux is given as:

$$F_\rho = \sum_s F_{\rho_s} = \frac{\rho^L \overline{U}^L + \rho^R \overline{U}^R}{2} - \frac{\tilde{c}_s(\lambda_1 dv_1 + \lambda_2 dv_2) + \lambda_3 dv_3)}{2} \tag{26}$$

$$dv_3 = \frac{\tilde{a}^2(\rho^R - \rho^L) - (p^R - p^L)}{\tilde{a}^2} \tag{27}$$

Multiplying eq. (26) by the roe-averaged mass fraction and substituting into eq. (19) results in:

$$F_{\rho_s} = \tilde{c}_s F_\rho + \frac{(c_s^L - \tilde{c}_s)\rho^L(\overline{U}^L + \mid \tilde{U} \mid)}{2} + \frac{(c_s^R - \tilde{c}_s)\rho^R(\overline{U}^R - \mid \tilde{U} \mid)}{2} \tag{28}$$

The notation can be further simplified by defining the normal velocities as follows:

$$\lambda^+ = \frac{\overline{U}^L + \mid \tilde{U} \mid}{2}, \quad \lambda^- = \frac{\overline{U}^R - \mid \tilde{U} \mid}{2} \tag{29}$$

Finally, substituting Eq. (29) into Eq. (28) yields the final result for calculating the species flux in the decoupled system:

$$F_{\rho_s} = \tilde{c}_s F_\rho + (c_s^L - \tilde{c}_s)\rho^L \lambda^+ + (c_s^R - \tilde{c}_s)\rho^R \lambda^- \tag{30}$$

Forming the convective contributions to the Jacobians is straightforward. Because the $\mathbf{U}'$ level variables are constant, only the left, right, and roe-averaged state mass fractions vary. Differentiating Eq. (30) with respect to the mass fraction, c_s, the left and right state contributions are:

$$\frac{\partial F_{\rho_s}}{\partial c_s^L} = wF_\rho + (1-w)\rho^L\lambda^+ - w\rho^R\lambda^- \tag{31}$$

$$\frac{\partial F_{\rho_s}}{\partial c_s^R} = (1-w)F_\rho + (w-1)\rho^L\lambda^+ + w\rho^R\lambda^- \tag{32}$$

Because there is no dependence between species in decoupled convective formulation, the Jacobian block elements are purely diagonal for the convective contributions, of the form:

$$\begin{pmatrix} \frac{\partial F_{\rho_1}}{\partial c_1} & & 0 \\ & \ddots & \\ 0 & & \frac{\partial F_{\rho_{ns}}}{\partial c_{ns}} \end{pmatrix} \tag{33}$$

3 Thrust Optimization

A simple optimization example for an hypersonic testcase is to determine a level of thrust for a Supersonic Retro-Propulsion (SRP) blunt body geometry. The grid used for this test is a structured grid, show in figure 1 The nozzle parameters has a number of degrees of freedom with regard to

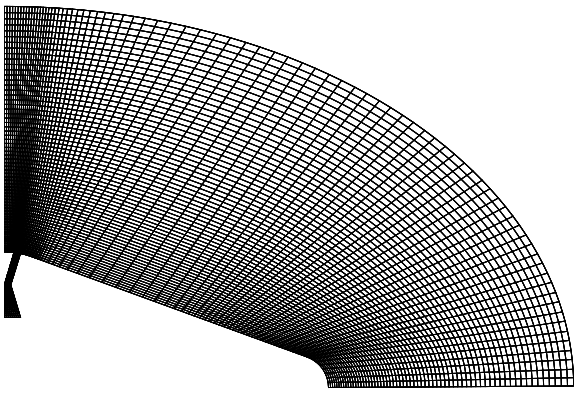

Figure 1: SRP Configuration Grid

user input. Combustion products can be specified separately from the freestream flow, and will be mixed in as the solution progresses in time. Additionally, the total pressure and temperature in the nozzle plenum can be specified. Through testing, this particular configuration has been determined to be steady and is therefore a good candidate for testing with the adjoint. In terms of optimization, this case can be optimized for maximum drag, using a number of design parameters. The simplist design parameter is total pressure and/or temperature in the nozzle plenum. This results in a significant simplification of the objective function to be minimized, because there is no mesh dependence on the design variables; thus, eq. (5) can be simplified to

$$\frac{\partial L}{\partial \mathbf{D}} = \frac{\partial f}{\partial \mathbf{D}} + \frac{\partial \mathbf{R}}{\partial \mathbf{D}}^T \mathbf{\Lambda} \tag{34}$$

Now the cost function and residual dependence upon the plenum initial conditions must be derived. For the drag cost function, this is achieved by looping over all of the surface faces and linearizing the drag cost function w.r.t. the pressure/temperature (CHECK THIS!). For the residual, the

4 Specific Heat Derivatives

4.1 Linear Interpolation

NOTE: THIS IS NOT USED (we're quadratically interpolating)

In order to check Specific heat (C_p) derivatives from the curve fits, C_p and species enthalpy (h_{ij}) must have continuous derivatives. This is not inherently the case from the Gordon-McBride curve fits employed; therefore blending is employed to ensure that both C_p and h_{ij} are C_1 and C_0 continuous. The blending is implemented as

$$C_p = f_n C_{p_n} + f_m C_{p_m} \tag{35}$$

$$h_{ij} = \int C_p dT = \int (f_n C_{p_n} + f_m C_{p_m}) dT \tag{36}$$

The curve fits defined by Gordon-McBride are a 7th-degree polynomial of the form

$$C_p = \sum_{i=1}^{7} C_i T^{i-3} \tag{37}$$

And the blending parameters, f_n and f_m are

$$f_n = \begin{cases} \frac{1}{2dT_h}(T_u - T) & (T_u - dT_h) < T \leq T_u \\ \frac{1}{2dT_l}(T - T_l) & T_l < T \leq (T_l + dT_h) \end{cases} \tag{38}$$

$$dT_h = \frac{T_u}{100} \quad dT_l = \frac{T_l}{100} \tag{39}$$

$$f_m = 1 - f_n \tag{40}$$

Where T_u and T_l are the upper and lower valid temperature bounds on a range. Thus, substituting into eqn. (36) yields

$$h_{ij} = \int \left(f_n \sum_{i=1}^{7} C_{n_i} T^{i-3} + (1 - f_n) \sum_{i=1}^{7} C_{m_i} T^{i-3} \right) dT \tag{41}$$

If $(T_u - dT_h) < T \leq T_u$, integrating eqn. (41) yields

$$h_{ij} = 50h_n - \frac{50}{T_u} \left(C_{n_1} \ln(T) + \sum_{i=2}^{7} \frac{C_{n_i} T^{i-1}}{i-1} \right) - 49h_m + \frac{50}{T_u} \left(C_{m_1} \ln(T) + \sum_{i=2}^{7} \frac{C_{m_i} T^{i-1}}{i-1} \right) + C \tag{42}$$

Where

$$h_n = \int C_{p_n} dT, \quad h_m = \int C_{p_m} dT \tag{43}$$

If $T_l < T \leq (T_l + dT_h)$, integrating eqn. (41) yields

$$h_{ij} = -50h_n + \frac{50}{T_l} \left(C_{n_1} \ln(T) + \sum_{i=2}^{7} \frac{C_{n_i} T^{i-1}}{i-1} \right) + 49h_m - \frac{50}{T_l} \left(C_{m_1} \ln(T) + \sum_{i=2}^{7} \frac{C_{m_i} T^{i-1}}{i-1} \right) + C \tag{44}$$

It quickly becomes apparent that eqn.s (42) and (44) can be combined into

$$h_{ij} = \lambda \left[50h_n - \frac{2}{dT_s} \left(C_{n_1} \ln(T) + \sum_{i=2}^{7} \frac{C_{n_i} T^{i-1}}{i-1} \right) - 49h_m + \frac{2}{dT_s} \left(C_{m_1} \ln(T) + \sum_{i=2}^{7} \frac{C_{m_i} T^{i-1}}{i-1} \right) \right] + C \tag{45}$$

with

$$\lambda,\ dT_s = \begin{cases} 1,\ dT_u & (T_u - dT_h) < T \leq T_u \\ -1,\ dT_l & T_l < T \leq (T_l + dT_h) \end{cases} \tag{46}$$

To detemine the constant of integration, C, we use the condition

$$h_{ij}(T_{co}) = \frac{1}{2}(h_n(T_{co}) + h_m(T_{co})), \quad T_{co} = T_u = T_l \tag{47}$$

which specifies that the enthalpy at the intersection of temperature ranges should the average of the enthalpy in the respective ranges. Evaluating eqn. (45) with condition provided my eqn. (47) yields

$$\begin{aligned} C = &(\frac{1}{2} - 50\lambda)h_n \mid_{T=T_{co}}) + (\frac{1}{2} + 49\lambda)h_m \mid_{T=T_{co}}) \\ &+ \frac{2\lambda}{dT_s}(C_{n_1} \ln(T_{co}) + \sum_{i=2}^{7} \frac{C_{n_i} T_{co}^{i-1}}{i-1}) \\ &- \frac{2\lambda}{dT_s}(C_{m_1} \ln(T_{co}) + \sum_{i=2}^{7} \frac{C_{m_i} T_{co}^{i-1}}{i-1}) \end{aligned} \tag{48}$$

Inserting into eq. (48), the final form of the blended enthalpy function is

$$\begin{aligned} h_{ij} = &\lambda \left[50h_n - \frac{2}{dT_s} \left(C_{n_1} \ln(T) + \sum_{i=2}^{7} \frac{C_{n_i} T^{i-1}}{i-1} \right) - 49h_m + \frac{2}{dT_s} \left(C_{m_1} \ln(T) + \sum_{i=2}^{7} \frac{C_{m_i} T^{i-1}}{i-1} \right) \right] \\ &+ (\frac{1}{2} - 50\lambda)h_n \mid_{T=T_{co}}) + (\frac{1}{2} + 49\lambda)h_m \mid_{T=T_{co}}) \\ &+ \frac{2\lambda}{dT_s} \left(C_{n_1} \ln(T_{co}) + \sum_{i=2}^{7} \frac{C_{n_i} T_{co}^{i-1}}{i-1} \right) \\ &- \frac{2\lambda}{dT_s} \left(C_{m_1} \ln(T_{co}) + \sum_{i=2}^{7} \frac{C_{m_i} T_{co}^{i-1}}{i-1} \right) \end{aligned} \tag{49}$$

4.2 Quadratic Interpolation Between Thermodynamic Curve Fits

We seek to blend the two thermodynamic curve fits in such a way that we maintain c_0 continuity in both specific heat (C_p) and enthalpy (h). To accomplish this, a quadratic function must be used, of the form

$$aT^2 + bT + c = C_p \tag{50}$$

The coefficients a, b, and c are determined by solving the system that results from the boundary value problem

$$\begin{cases} a{T_1}^2 + bT_1 + c = C_{p_1} \\ a{T_2}^2 + bT_2 + c = C_{p_2} \\ a\frac{\left({T_2}^3 - {T_1}^3\right)}{3} + b\frac{\left({T_2}^2 - {T_1}^2\right)}{2} + c\left(T_2 - T_1\right) = h_2 - h_1 \end{cases} \tag{51}$$

Where the x_1 and x_2 subscripts describe the left and right states, respectively. Solving the linear system, the coefficients are

$$\begin{cases} a = \frac{3\left(C_{p_2} + C_{p_1}\right)}{(T_2 - T_1)^2} - \frac{6(h_2 - h_1)}{(T_2 - T_1)^3} \\ b = -\frac{2\left[(C_{p_2} + 2C_{p_1})T_2 + (2C_{p_2} + C_{p_1})T_1\right]}{(T_2 - T_1)^2} + \frac{6(T_2 + T_1)(h_2 - h_1)}{(T_2 - T_1)^3} \\ c = \frac{C_{p_1}T_2(T_2 + 2T_1) + C_{p_2}T_1(T_1 + 2T_2)}{(T_2 - T_1)^2} - \frac{6T_1T_2(h_2 - h_1)}{(T_2 - T_1)^3} \end{cases} \tag{52}$$

This can be simplified to

$$\begin{cases} a = 3B - A \\ b = \frac{-2(C_{p_1}T_2 + C_{p_2}T_2)}{(T_2 - T_1)^2} + (T_2 + T_1)(A - 2B) \\ c = \frac{C_{p_1}{T_2}^2 + C_{p_2}{T_1}^2}{(T_2 - T_1)^2} + T_1T_2(2B - A) \end{cases} \tag{53}$$

$$A = \frac{6(h_2 - h_1)}{(T_2 - T_1)^3} \tag{54}$$

$$B = \frac{C_{p_2} + C_{p_1}}{(T_2 - T_1)^2} \tag{55}$$

5 Block Jacobi Adjoint Decoupling

The primal flow equations are solved using the decoupled scheme based on the work by Candler, et. al. In doing this the conserved variables are split from

$$\mathbf{U} = \begin{pmatrix} \rho_1 \\ \vdots \\ \rho_{ns} \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix} \tag{56}$$

into

$$\mathbf{U}' = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}, \qquad \hat{\mathbf{U}} = \begin{pmatrix} \rho_1 \\ \vdots \\ \rho_{ns} \end{pmatrix} \tag{57}$$

By doing this, the flow equations can be solved implicitly using an approximate, first-order jacobian to drive the flow solution to a converged steady-state. In practice, this is done by essentially formulating

$$\frac{V}{\Delta t}\mathbf{I} + \begin{pmatrix} \frac{\partial \mathbf{R}_\rho}{\partial \rho} & \frac{\partial \mathbf{R}_\rho}{\partial \rho \mathbf{u}} & \frac{\partial \mathbf{R}_\rho}{\partial \rho E} \\ \frac{\partial \mathbf{R}_{\rho \mathbf{u}}}{\partial \rho} & \frac{\partial \mathbf{R}_{\rho \mathbf{u}}}{\partial \rho \mathbf{u}} & \frac{\partial \mathbf{R}_{\rho \mathbf{u}}}{\partial \rho E} \\ \frac{\partial \mathbf{R}_{\rho E}}{\partial \rho} & \frac{\partial \mathbf{R}_{\rho E}}{\partial \rho \mathbf{u}} & \frac{\partial \mathbf{R}_{\rho E}}{\partial \rho E} \end{pmatrix} \begin{pmatrix} \Delta \rho \\ \Delta \rho \mathbf{u} \\ \Delta \rho E \end{pmatrix} = \begin{pmatrix} \mathbf{R}_\rho \\ \mathbf{R}_{\rho \mathbf{u}} \\ \mathbf{R}_{\rho E} \end{pmatrix} \tag{58}$$

to solve for the mixture flow variables, and formulating

$$\frac{V}{\Delta t}\mathbf{I} + \begin{pmatrix} \frac{\partial \mathbf{R}_{\rho_1}}{\partial c_1} & \cdots & \frac{\partial \mathbf{R}_{\rho_1}}{\partial c_{ns}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{R}_{\rho_{ns}}}{\partial c_1} & \cdots & \frac{\partial \mathbf{R}_{\rho_{ns}}}{\partial c_{ns}} \end{pmatrix} \begin{pmatrix} \Delta c_1 \\ \vdots \\ \Delta c_{ns} \end{pmatrix} = \begin{pmatrix} \mathbf{R}_{\rho_1} \\ \vdots \\ \mathbf{R}_{\rho_{ns}} \end{pmatrix} \tag{59}$$

Note that the correction terms required to maintain $\sum_{s=1}^{ns} c_s = 1$ and $\sum_{s=1}^{ns} \delta c_s = 0$ are omitted in Eq. (59) only for brevity in this explanation. Examining Eq.s (58-59) shows that there are clearly some physical dependencies being omitted, namely $\frac{\partial \mathbf{R}_{\rho_s}}{\partial \rho}$, $\frac{\partial \mathbf{R}_\rho}{\partial c_s}$, $\frac{\partial \mathbf{R}_{\rho \mathbf{u}}}{\partial c_s}$, and $\frac{\partial \mathbf{R}_{\rho E}}{\partial c_s}$. It has been demonstrated that omitting this dependencies does not hinder convergence the primal solver; however, the adjoint requires an exact linearization of the converged steady-state solution.

The discrete adjoint formulation is given as

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\right]^T \mathbf{\Lambda} = -\frac{\partial f}{\partial \mathbf{Q}} \tag{60}$$

where $\mathbf{Q}$ is a vector of conserved variables, and f is the cost function (i.e. lift, drag, etc.). There is now an apparent need to reconcile the two sets of conserved variables in Eq. (57) with $\mathbf{Q}$. The most intuitive and staightforward way to do this is to forgo solving for the species mass ρ_s in lieu of the species mass fraction c_s. Thus, $\mathbf{Q}$ can be expressed as

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \\ c_1 \\ \vdots \\ c_{ns} \end{pmatrix} \tag{61}$$

This allows the linearizations derived in Eq.s (58-59) to be used in the adjoint formulation, by augmenting them with the previously omitted linearizations. Replacing $\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}$ with the fully system, the adjoint system becomes

$$\begin{pmatrix} \frac{\partial \mathbf{R}_\rho}{\partial \rho}^T & \frac{\partial \mathbf{R}_\rho}{\partial \rho \mathbf{u}}^T & \frac{\partial \mathbf{R}_\rho}{\partial \rho E}^T & \frac{\partial \mathbf{R}_\rho}{\partial c_s}^T \\ \frac{\partial \mathbf{R}_{\rho \mathbf{u}}}{\partial \rho}^T & \frac{\partial \mathbf{R}_{\rho \mathbf{u}}}{\partial \rho \mathbf{u}}^T & \frac{\partial \mathbf{R}_{\rho \mathbf{u}}}{\partial \rho E}^T & \frac{\partial \mathbf{R}_{\rho \mathbf{u}}}{\partial c_s}^T \\ \frac{\partial \mathbf{R}_{\rho E}}{\partial \rho}^T & \frac{\partial \mathbf{R}_{\rho E}}{\partial \rho \mathbf{u}}^T & \frac{\partial \mathbf{R}_{\rho E}}{\partial \rho E}^T & \frac{\partial \mathbf{R}_{\rho E}}{\partial c_s}^T \\ \frac{\partial \mathbf{R}_{\rho_s}}{\partial \rho}^T & \frac{\partial \mathbf{R}_{\rho_s}}{\partial \rho \mathbf{u}}^T & \frac{\partial \mathbf{R}_{\rho_s}}{\partial \rho E}^T & \frac{\partial \mathbf{R}_{\rho_s}}{\partial c_s}^T \end{pmatrix} \begin{pmatrix} \Lambda_\rho \\ \Lambda_{\rho \mathbf{u}} \\ \Lambda_{\rho E} \\ \Lambda_{c_s} \end{pmatrix} = - \begin{pmatrix} \frac{\partial f}{\partial \rho} \\ \frac{\partial f}{\partial \rho \mathbf{u}} \\ \frac{\partial f}{\partial \rho E} \\ \frac{\partial f}{\partial c_s} \end{pmatrix} \tag{62}$$

Thus the jacobian in Eq. (62) is the completed one of Eq.s (58-59). While this is useful, the point of decoupling the species equations from the mixture equations was to speed up the linear solver and save memory. Solving Eq. (62) undermines both of these goals, so an alternative solution strategy must be formulated. If a block jacobi scheme is employed, the system can be decoupled once again as

$$\begin{pmatrix} \frac{\partial \mathbf{R}_\rho}{\partial \rho}^T & \frac{\partial \mathbf{R}_\rho}{\partial \rho \mathbf{u}}^T & \frac{\partial \mathbf{R}_\rho}{\partial \rho E}^T \\ \frac{\partial \mathbf{R}_{\rho \mathbf{u}}}{\partial \rho}^T & \frac{\partial \mathbf{R}_{\rho \mathbf{u}}}{\partial \rho \mathbf{u}}^T & \frac{\partial \mathbf{R}_{\rho \mathbf{u}}}{\partial \rho E}^T \\ \frac{\partial \mathbf{R}_{\rho E}}{\partial \rho}^T & \frac{\partial \mathbf{R}_{\rho E}}{\partial \rho \mathbf{u}}^T & \frac{\partial \mathbf{R}_{\rho E}}{\partial \rho E}^T \end{pmatrix} \begin{pmatrix} \Lambda_\rho \\ \Lambda_{\rho \mathbf{u}} \\ \Lambda_{\rho E} \end{pmatrix} = - \begin{pmatrix} \frac{\partial f}{\partial \rho} \\ \frac{\partial f}{\partial \rho \mathbf{u}} \\ \frac{\partial f}{\partial \rho E} \end{pmatrix} - \begin{pmatrix} \frac{\partial \mathbf{R}_\rho}{\partial c_s}^T \\ \frac{\partial \mathbf{R}_{\rho \mathbf{u}}}{\partial c_s}^T \\ \frac{\partial \mathbf{R}_{\rho E}}{\partial c_s}^T \end{pmatrix} \Lambda_{c_s} \tag{63}$$

$$\frac{\partial \mathbf{R}_{\rho_s}}{\partial c_s}^T \Lambda_{c_s} = -\frac{\partial f}{\partial c_s} - \frac{\partial \mathbf{R}_{\rho_s}}{\partial \rho}^T \Lambda_\rho - \frac{\partial \mathbf{R}_{\rho_s}}{\partial \rho \mathbf{u}}^T \Lambda_{\rho \mathbf{u}} - \frac{\partial \mathbf{R}_{\rho_s}}{\partial \rho E}^T \Lambda_{\rho E} \tag{64}$$

If a time-like derivative is added to the adjoint, the solution of the costate variables, Λ, can be time marched similar to the primal flow solver

$$\left[\frac{V}{\Delta t} \mathbf{I} + \frac{\partial \mathbf{R}}{\partial \mathbf{Q}}^T \right] \Delta \Lambda = -\frac{\partial f}{\partial \mathbf{Q}} - \frac{\partial \mathbf{R}}{\partial \mathbf{Q}}^T \Lambda \tag{65}$$

Thus, the first system in Eq. (63) becomes

$$\left[\frac{V}{\Delta t}\mathbf{I}+\begin{pmatrix} \frac{\partial \mathbf{R}_{\rho}}{\partial \rho}^{T} & \frac{\partial \mathbf{R}_{\rho}}{\partial \rho \mathbf{u}}^{T} & \frac{\partial \mathbf{R}_{\rho}}{\partial \rho E}^{T} \\ \frac{\partial \mathbf{R}_{\rho \mathbf{u}}}{\partial \rho}^{T} & \frac{\partial \mathbf{R}_{\rho \mathbf{u}}}{\partial \rho \mathbf{u}}^{T} & \frac{\partial \mathbf{R}_{\rho \mathbf{u}}}{\partial \rho E}^{T} \\ \frac{\partial \mathbf{R}_{\rho E}}{\partial \rho}^{T} & \frac{\partial \mathbf{R}_{\rho E}}{\partial \rho \mathbf{u}}^{T} & \frac{\partial \mathbf{R}_{\rho E}}{\partial \rho E}^{T} \end{pmatrix}\right]\begin{pmatrix} \Delta\Lambda_{\rho} \\ \Delta\Lambda_{\rho \mathbf{u}} \\ \Delta\Lambda_{\rho E} \end{pmatrix} = \\ -\begin{pmatrix} \frac{\partial f}{\partial \rho} \\ \frac{\partial f}{\partial \rho \mathbf{u}} \\ \frac{\partial f}{\partial \rho E} \end{pmatrix} - \begin{pmatrix} \frac{\partial \mathbf{R}_{\rho}}{\partial \rho}^{T} & \frac{\partial \mathbf{R}_{\rho}}{\partial \rho \mathbf{u}}^{T} & \frac{\partial \mathbf{R}_{\rho}}{\partial \rho E}^{T} \\ \frac{\partial \mathbf{R}_{\rho \mathbf{u}}}{\partial \rho}^{T} & \frac{\partial \mathbf{R}_{\rho \mathbf{u}}}{\partial \rho \mathbf{u}}^{T} & \frac{\partial \mathbf{R}_{\rho \mathbf{u}}}{\partial \rho E}^{T} \\ \frac{\partial \mathbf{R}_{\rho E}}{\partial \rho}^{T} & \frac{\partial \mathbf{R}_{\rho E}}{\partial \rho \mathbf{u}}^{T} & \frac{\partial \mathbf{R}_{\rho E}}{\partial \rho E}^{T} \end{pmatrix}\begin{pmatrix} \Lambda_{\rho} \\ \Lambda_{\rho \mathbf{u}} \\ \Lambda_{\rho E} \end{pmatrix} - \begin{pmatrix} \frac{\partial \mathbf{R}_{\rho}}{\partial c_s}^{T} \\ \frac{\partial \mathbf{R}_{\rho \mathbf{u}}}{\partial c_s}^{T} \\ \frac{\partial \mathbf{R}_{\rho E}}{\partial c_s}^{T} \end{pmatrix}\Lambda_{c_s} \tag{66}$$

and the second system in Eq. (64) becomes

$$\left(\frac{V}{\Delta t}\mathbf{I}+\frac{\partial \mathbf{R}_{\rho_s}}{\partial c_s}^{T}\right)\Delta\Lambda_{c_s} = -\frac{\partial f}{\partial c_s}-\frac{\partial \mathbf{R}_{\rho_s}}{\partial c_s}^{T}\Lambda_{c_s}-\frac{\partial \mathbf{R}_{\rho_s}}{\partial \rho}^{T}\Lambda_{\rho}-\frac{\partial \mathbf{R}_{\rho_s}}{\partial \rho \mathbf{u}}^{T}\Lambda_{\rho \mathbf{u}}-\frac{\partial \mathbf{R}_{\rho_s}}{\partial \rho E}^{T}\Lambda_{\rho E} \tag{67}$$

This is advantageous, because the jacobians on the left hand side (LHS) can be the first-order approximate jacobians that were used to solve the primal flow equations; hence, all of the benefits of the diagonal block matricies that are exploited to reduce the linear solver cost and overall memory now apply to the adjoint.

6 Steps in Flow/Adjoint Solve

The method of solving the primal flow conservation equations in a decoupled fashion begins by introducing a change of variable. The conserved variables for a fully coupled system

$$\mathbf{U} = \begin{pmatrix} \rho_1 \\ \vdots \\ \rho_{ns} \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix} \tag{68}$$

Can be split into two sets of conserved variables

$$\mathbf{U}' = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}, \qquad \hat{\mathbf{U}} = \begin{pmatrix} \rho_1 \\ \vdots \\ \rho_{ns} \end{pmatrix} \tag{69}$$

Note the constraint

$$\rho = \sum_{s=1}^{N_s} \rho_s \tag{70}$$

that must be explicitly enforced between $\mathbf{U}'$ and $\hat{\mathbf{U}}$. This is effectively the result of introducing the additional variable ρ. Likewise, the residual formed via the summation of fluxes for each governing equation at each node can be split in a similar fashion. For the fully coupled system, this is

$$\mathbf{R} = \begin{pmatrix} \mathbf{R}_{\rho_1} \\ \vdots \\ \mathbf{R}_{\rho_{ns}} \\ \mathbf{R}_{\rho u} \\ \mathbf{R}_{\rho v} \\ \mathbf{R}_{\rho w} \\ \mathbf{R}_{\rho E} \end{pmatrix} \tag{71}$$

and can be split into two residual vectors for the decoupled scheme

$$\mathbf{R}' = \begin{pmatrix} \mathbf{R}_{\rho} \\ \mathbf{R}_{\rho u} \\ \mathbf{R}_{\rho v} \\ \mathbf{R}_{\rho w} \\ \mathbf{R}_{\rho E} \end{pmatrix}, \qquad \hat{\mathbf{R}} = \begin{pmatrix} \mathbf{R}_{\rho_1} \\ \vdots \\ \mathbf{R}_{\rho_{ns}} \end{pmatrix} \tag{72}$$

There is a similar constraint between these systems as the conserved variables in that

$$\mathbf{R}_{\rho} = \sum_{s=1}^{N_s} \mathbf{R}_{\rho_s} \tag{73}$$

The mapping of the decoupled variable set $\mathbf{U}'$ to the fully coupled variable set $\mathbf{U}$ is done by

$$\begin{pmatrix} d\rho_1 \\ \vdots \\ d\rho_{ns} \\ d\rho u \\ d\rho v \\ d\rho w \\ d\rho E \end{pmatrix} = \begin{pmatrix} c_1 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{ns} & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} d\rho \\ d\rho u \\ d\rho v \\ d\rho w \\ d\rho E \end{pmatrix} \tag{74}$$

Likewise, the mapping of the decoupled variables $\hat{\mathbf{U}}$ to the fully coupled variables $\mathbf{U}$ is done by

$$\begin{pmatrix} d\rho_1 \\ \vdots \\ d\rho_{ns} \\ d\rho u \\ d\rho v \\ d\rho w \\ d\rho E \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} d\rho_1 \\ \vdots \\ d\rho_{ns} \end{pmatrix} \tag{75}$$

Because the total density ρ is solved for in $\mathbf{U}'$, the sum of $\hat{\mathbf{U}}$ cannot change. In practice, this means that the primal solver does not solve for $\hat{\mathbf{U}}$, but rather for the mass fractions c_s which we define as

$$c_s = \frac{\rho_s}{\sum_{s=1}^{N_s} \rho_s} = \frac{\rho_s}{\rho} \tag{76}$$

thus, in lieu of $\hat{\mathbf{U}}$, a new vector $\hat{\mathbf{V}}$ is solved for

$$\hat{\mathbf{V}} = \begin{pmatrix} c_1 \\ \vdots \\ c_{ns} \end{pmatrix} \tag{77}$$

This requires a new mapping to convert $\hat{\mathbf{V}}$ to $\mathbf{U}$

$$\begin{pmatrix} d\rho_1 \\ \vdots \\ d\rho_{ns} \\ d\rho u \\ d\rho v \\ d\rho w \\ d\rho E \end{pmatrix} = \begin{pmatrix} \rho & 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & \rho \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} dc_1 \\ \vdots \\ dc_{ns} \end{pmatrix} \tag{78}$$

A mapping is also required to convert between the equations of the fully coupled system and decoupled systems. To map $\mathbf{R}$ to $\mathbf{R}'$

$$\begin{pmatrix} \mathbf{R}_\rho \\ \mathbf{R}_{\rho u} \\ \mathbf{R}_{\rho v} \\ \mathbf{R}_{\rho w} \\ \mathbf{R}_{\rho E} \end{pmatrix} = \begin{pmatrix} 1 & \dots & 1 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 1 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 1 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 1 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{R}_{\rho_1} \\ \vdots \\ \mathbf{R}_{\rho_{ns}} \\ \mathbf{R}_{\rho u} \\ \mathbf{R}_{\rho v} \\ \mathbf{R}_{\rho w} \\ \mathbf{R}_{\rho E} \end{pmatrix} \tag{79}$$

The mapping from $\mathbf{R}$ to $\hat{\mathbf{R}}$ is simply

$$\begin{pmatrix} \mathbf{R}_{\rho_1} \\ \vdots \\ \mathbf{R}_{\rho_{ns}} \end{pmatrix} = \begin{pmatrix} 1 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{R}_{\rho_1} \\ \vdots \\ \mathbf{R}_{\rho_{ns}} \\ \mathbf{R}_{\rho u} \\ \mathbf{R}_{\rho v} \\ \mathbf{R}_{\rho w} \\ \mathbf{R}_{\rho E} \end{pmatrix} \tag{80}$$

With this defined, the sequence of solving the primal problem begins with solving for $\mathbf{U}'$

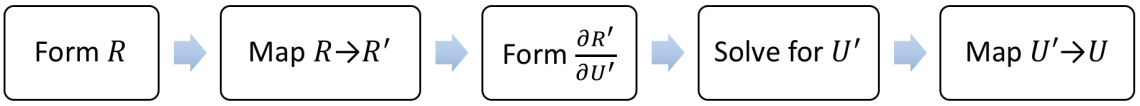

Figure 2: Primal solve flow for $\mathbf{U}'$ variable set

Next the variable set $\hat{\mathbf{U}}$ is solved for

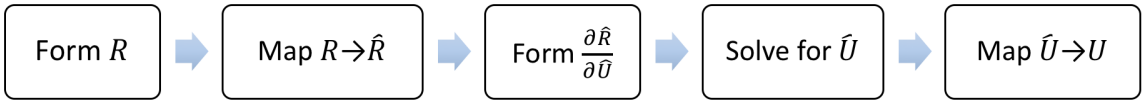

Figure 3: Primal solve flow for $\hat{\mathbf{U}}$ variable set