# Notes on the Fundamental Equations of PHASTA

James Wright

October 29, 2019

## 1 Shorthand

$$\phi_{,t} = \frac{\partial \phi}{\partial t} \tag{1}$$

$$\phi_{,i} = \frac{\partial \phi}{\partial x_i} \tag{2}$$

$$u_{i,t} = \frac{\partial u_i}{\partial t} \tag{3}$$

$$u_{i,j} = \frac{\partial u_i}{\partial x_j} \tag{4}$$

$$[\phi u_i]_{,j} = \frac{\partial \phi u_i}{\partial x_j} \tag{5}$$

## 2 Fundamental Fluid Equations

Unsteady Compressible Navier-Stokes (UCNS) equations:

### 2.1 Traditional Form

**Continuity**

$$\rho_{,t} + \left[\rho u_j\right]_{,j} = 0 \tag{6}$$

**Momentum**

$$[\rho u_i]_{,t} + \left[\rho u_i u_j\right]_{,j} + p_{,i} = \tau_{ij,j} + b_i \tag{7}$$

**Energy**

$$[\rho e_{tot}]_{,t} + \left[\rho e_{tot} u_j\right]_{,j} + \left[\rho u_j\right]_{,j} = \left[\tau_{ij} u_j\right]_{,j} + b_i u_j + r + q_{i,i} \tag{8}$$

## 2.2 Conservative Vectorized Form

$$\boldsymbol{U} \equiv \begin{Bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho e_{tot} \end{Bmatrix} = \begin{Bmatrix} \rho \\ \rho u_j \\ \rho e_{tot} \end{Bmatrix} \tag{9}$$

Flux Vector:

$$\begin{aligned} \boldsymbol{F}_i &= \underbrace{\begin{Bmatrix} \rho u_i \\ \rho u_i u_j \\ \rho u_i e_{tot} \end{Bmatrix} + \begin{Bmatrix} 0 \\ p\delta_{ij} \\ u_i p \end{Bmatrix}}_{\text{Advective Flux}} \underbrace{- \begin{Bmatrix} 0 \\ \tau_{ij} \\ \tau_{ij} u_j \end{Bmatrix} + \begin{Bmatrix} 0 \\ \boldsymbol{0} \\ q_{i,i} \end{Bmatrix}}_{\text{Diffusive Flux}} \\ &= \boldsymbol{F}_i^{\text{adv}} + \boldsymbol{F}_i^{\text{dif}} \end{aligned} \tag{10}$$

Also:

$$\boldsymbol{F}_i^{adv} = u_i \boldsymbol{U} + \begin{Bmatrix} 0 \\ p\delta_{ij} \\ u_i p \end{Bmatrix} \tag{11}$$

Source Vector:

$$\boldsymbol{\mathfrak{F}} = \begin{Bmatrix} 0 \\ b_j \\ b_j u_j + r \end{Bmatrix} \tag{12}$$

These terms combine together to form:

$$\boldsymbol{U}_{,t} + \boldsymbol{F}_{i,i} = \boldsymbol{\mathfrak{F}} \tag{13}$$

# 3 Finite Element Discretization

Before discritizing, we must first setup the UCNS equations. First, rearrange eq. (13) into a residual form:

$$\boldsymbol{U}_{,t} + \boldsymbol{F}_{i,i} - \boldsymbol{\mathfrak{F}} = \boldsymbol{0} \tag{14}$$

Next multiply by weight/test functions, $\boldsymbol{W}$. Note that $\boldsymbol{0}$ is now just a scalar 0.

$$\boldsymbol{W} \cdot (\boldsymbol{U}_{,t} + \boldsymbol{F}_{i,i} - \boldsymbol{\mathfrak{F}}) = 0 \tag{15}$$

To create the **Weak Form** of the NS equations, integrate over the domain $\Omega$:

$$\int_\Omega \boldsymbol{W} \cdot (\boldsymbol{U}_{,t} + \boldsymbol{F}_{i,i} - \boldsymbol{\mathfrak{F}}) \, \mathrm{d}\Omega = 0 \tag{16}$$

Next, perform integration by parts and Gauss's theorem on $\boldsymbol{F}_{i,i}$:

$$\int_\Omega \left\{ \boldsymbol{W} \cdot \boldsymbol{U}_{,t} - \boldsymbol{W}_{,i} \cdot \boldsymbol{F}_i - \boldsymbol{W} \cdot \boldsymbol{\mathfrak{F}} \right\} \mathrm{d}\Omega + \int_\Gamma \boldsymbol{W} \cdot \boldsymbol{F}_i \cdot \hat{n}_i \, \mathrm{d}\Gamma = 0 \tag{17}$$

where $\Gamma$ is the boundary of the domain $\Omega$ and $\hat{n}_i$ is the normal unit vector of the boundary surface. Equation (17) represents the **Weak UCNS in IBP Form**.

## 3.1 Domain Discretization

Define nodes, points, and elements.

### 3.1.1 Shape Function Decomposition

Define a set of functions $N$ that are a basis for the weight functions. In the case of the **Galerkin Form**, the basis/shape functions $N$ are the same for the weight function and the solution function. We can decompose some constant $\phi$ into

$$\phi(\boldsymbol{x}) = \sum_{A=1}^{n_n} N_A(\boldsymbol{x}) \phi_A \tag{18}$$

where $A$ is the index of each node and $\phi_A$ is the value of $\phi$ at each node $A$. This decomposition can be extrapolated to our unknowns: $\boldsymbol{U}$, $\boldsymbol{U}_{,t}$, and $\boldsymbol{W}$:

$$\boldsymbol{U}(\boldsymbol{x}) = \sum_{A=1}^{n_n} N_A(\boldsymbol{x}) \boldsymbol{U}_A \tag{19a}$$

$$\boldsymbol{U}_{,t}(\boldsymbol{x}) = \sum_{A=1}^{n_n} N_A(\boldsymbol{x}) \boldsymbol{U}_{A,t} \tag{19b}$$

$$\boldsymbol{W}(\boldsymbol{x}) = \sum_{B=1}^{n_n} N_B(\boldsymbol{x}) \boldsymbol{W}_B \tag{19c}$$

Since we are working in Galerkin, $N_A(\boldsymbol{x}) = N_B(\boldsymbol{x})$ for $A = B$ (ie, at the same node).