Notes on the Fundamental Equations of PHASTA

James Wright

December 13, 2019

1 Shorthand

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

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

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

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

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

2 Theory

2.1 Fundamental Fluid Equations

Unsteady Compressible Navier-Stokes (UCNS) equations:

2.1.1 Traditional (Strong) Form

Continuity

$$\rho_{,t} + \left[\rho u_j\right]_{,j} = 0 \tag{1}$$

Momentum

$$[\rho u_i]_{,t} + \left[\rho u_i u_j\right]_{,j} + p_{,i} = \tau_{ij,j} + b_i \tag{2}$$

Energy

$$[\rho e_{tot}]_{,t} + \left[\rho e_{tot} u_j\right]_{,j} + \left[\rho u_j\right]_{,j} = \left[\tau_{ij} u_j\right]_{,j} + b_i u_j + r + q_{i,i} \tag{3}$$

Constituitive Equations

$$q_i = -\kappa T_{,i} \tag{4}$$

2.1.2 Conservative Vectorized Form

$$\boldsymbol{U} \equiv \begin{Bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho e_{tot} \end{Bmatrix} = \begin{Bmatrix} \rho \\ \rho u_j \\ \rho e_{tot} \end{Bmatrix} \tag{5}$$

Flux Vector:

$$\begin{aligned} \boldsymbol{F}_i &= \underbrace{\begin{Bmatrix} \rho u_i \\ \rho u_i u_j \\ \rho u_i e_{tot} \end{Bmatrix} + \begin{Bmatrix} 0 \\ p\delta_{ij} \\ u_i p \end{Bmatrix}}_{\text{Advective Flux}} \underbrace{- \begin{Bmatrix} 0 \\ \tau_{ij} \\ \tau_{ij} u_j \end{Bmatrix} + \begin{Bmatrix} 0 \\ \boldsymbol{0} \\ q_{i,i} \end{Bmatrix}}_{\text{Diffusive Flux}} \\ &= \boldsymbol{F}_i^{\text{adv}} + \boldsymbol{F}_i^{\text{dif}} \end{aligned} \tag{6}$$

Also:

$$\boldsymbol{F}_i^{adv} = u_i \boldsymbol{U} + \begin{Bmatrix} 0 \\ p\delta_{ij} \\ u_i p \end{Bmatrix} \tag{7}$$

Source Vector:

$$\boldsymbol{\mathfrak{F}} = \begin{Bmatrix} 0 \\ b_j \\ b_j u_j + r \end{Bmatrix} \tag{8}$$

These terms combine together to form:

$$\boldsymbol{U}_{,t} + \boldsymbol{F}_{i,i} = \boldsymbol{\mathfrak{F}} \tag{9}$$

2.2 Finite Element Discretization

Before discretizing, we must first setup the UCNS equations. First, rearrange eq. (9) into a residual form:

$$\boldsymbol{U}_{,t} + \boldsymbol{F}_{i,i} - \boldsymbol{\mathfrak{F}} = \boldsymbol{0} \tag{10}$$

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

$$\boldsymbol{W} \cdot (\boldsymbol{U}_{,t} + \boldsymbol{F}_{i,i} - \boldsymbol{\mathfrak{F}}) = 0 \tag{11}$$

To create the **Weak Form** of the NS equations, integrate over the domain Ω:

$$\int_\Omega \boldsymbol{W} \cdot (\boldsymbol{U}_{,t} + \boldsymbol{F}_{i,i} - \boldsymbol{\mathfrak{F}}) \, \mathrm{d}\Omega = 0 \tag{12}$$

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

$$\int_\Omega \left\{ \boldsymbol{W} \cdot \boldsymbol{U}_{,t} - \boldsymbol{W}_{,i} \cdot \boldsymbol{F}_i(\boldsymbol{U}, \boldsymbol{U}_{,i}) - \boldsymbol{W} \cdot \boldsymbol{\mathfrak{F}}(\boldsymbol{U}) \right\} \mathrm{d}\Omega + \int_\Gamma \boldsymbol{W} \cdot \boldsymbol{F}_i(\boldsymbol{U}, \boldsymbol{U}_{,i}) \cdot \hat{n}_i \, \mathrm{d}\Gamma = 0 \tag{13}$$

where Γ is the boundary of the domain Ω and $\hat{n}_i$ is the normal unit vector of the boundary surface. Equation (13) represents the **Weak UCNS in IBP Form**. The fact that $\boldsymbol{F}_i$ and $\boldsymbol{\mathfrak{F}}$ are functions of $\boldsymbol{U}$ and $\boldsymbol{U}_{,i}$ is explicitly shown here.

2.2.1 Domain Discretization

Define nodes, points, and elements.

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 ϕ into

$$\phi(\boldsymbol{x}) = \sum_{A=1}^{n_n} N_A(\boldsymbol{x})\phi_A \tag{14}$$

where A is the index of each node, ϕ_A is the value of ϕ at each node A, and n_n is the number of nodes used to discretize Ω. Note that ϕ on the LHS is continuous, but ϕ_A are discrete scalar values for every A. This decomposition can be extrapolated to our unknowns: $\boldsymbol{U}$, $\boldsymbol{U}_{,t}$, and $\boldsymbol{W}$:

$$\boldsymbol{U}(\boldsymbol{x}) = \sum_{A=1}^{n_n} N_A(\boldsymbol{x})\boldsymbol{U}_A \tag{15a}$$

$$\boldsymbol{U}_{,t}(\boldsymbol{x}) = \sum_{A=1}^{n_n} N_A(\boldsymbol{x})\boldsymbol{U}_{A,t} \tag{15b}$$

$$\boldsymbol{W}(\boldsymbol{x}) = \sum_{B=1}^{n_n} N_B(\boldsymbol{x})\boldsymbol{W}_B \tag{15c}$$

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

Discretize UCNS Let's substitute the expressions in eq. (15) into the weak UCNS form in eq. (13).

$$\begin{aligned}\int_\Omega \Bigg\{ \sum_{B=1}^{n_n} N_B(\boldsymbol{x})\boldsymbol{W}_B \cdot \sum_{A=1}^{n_n} N_A(\boldsymbol{x})\boldsymbol{U}_{A,t} - \sum_{B=1}^{n_n} N_{B,i}(\boldsymbol{x})\boldsymbol{W}_B\boldsymbol{F}_i\Bigg(\sum_{A=1}^{n_n} N_A(\boldsymbol{x})\boldsymbol{U}_A,\ \sum_{A=1}^{n_n} N_{A,i}(\boldsymbol{x})\boldsymbol{U}_A\Bigg) \\ - \sum_{B=1}^{n_n} \boldsymbol{W}_B\boldsymbol{\mathfrak{F}}\Bigg(\sum_{A=1}^{n_n} N_A(\boldsymbol{x})\boldsymbol{U}_A\Bigg)\Bigg\}\,\mathrm{d}\Omega \\ + \int_\Gamma \sum_{B=1}^{n_n} \boldsymbol{W}_B \cdot \boldsymbol{F}_i\Bigg(\sum_{A=1}^{n_n} N_A(\boldsymbol{x})\boldsymbol{U}_A,\ \sum_{A=1}^{n_n} N_{A,i}(\boldsymbol{x})\boldsymbol{U}_A\Bigg)\hat{n}_i\,\mathrm{d}\Gamma = 0\end{aligned} \tag{16}$$

Note that $\boldsymbol{U}_{,i} = [\sum_{A=1}^{n_n} N_A(\boldsymbol{x})\boldsymbol{U}_A]_{,i}$, but since $\boldsymbol{U}_A$ is not a function of $\boldsymbol{x}$ (it is only dependent on A), the sum and $\boldsymbol{U}_A$ can be taken out of the brackets, leaving $N_A(\boldsymbol{x})$ to be derived: $\boldsymbol{U}_{,i} = \sum_{A=1}^{n_n} N_{A,i}(\boldsymbol{x})\boldsymbol{U}_A$.

Since $\boldsymbol{W}_B$ is a vector of constants, we can take it out of the integral, along with its respective summation:

$$\sum_{B=1}^{n_n} \boldsymbol{W}_B \Bigg\{ \int_\Omega \Bigg\{ N_B(\boldsymbol{x}) \cdot \sum_{A=1}^{n_n} N_A(\boldsymbol{x}) \boldsymbol{U}_{A,t} - N_{B,i}(\boldsymbol{x}) \boldsymbol{F}_i \Bigg(\sum_{A=1}^{n_n} N_A(\boldsymbol{x}) \boldsymbol{U}_A,\ \sum_{A=1}^{n_n} N_{A,i}(\boldsymbol{x}) \boldsymbol{U}_A \Bigg) - \boldsymbol{\mathfrak{F}} \Bigg(\sum_{A=1}^{n_n} N_A(\boldsymbol{x}) \boldsymbol{U}_A \Bigg) \Bigg\} \, \mathrm{d}\Omega + \int_\Gamma \boldsymbol{F}_i \Bigg(\sum_{A=1}^{n_n} N_A(\boldsymbol{x}) \boldsymbol{U}_A,\ \sum_{A=1}^{n_n} N_{A,i}(\boldsymbol{x}) \boldsymbol{U}_A \Bigg) \hat{n}_i \, \mathrm{d}\Gamma \Bigg\} = 0 \tag{17}$$

$$eq.\ (17) \Rightarrow \sum_{B=1}^{n_n} \boldsymbol{W}_B \cdot \boldsymbol{G}_B = 0 \tag{18}$$

If we assume that $\boldsymbol{W}_B$ is arbitrary, then $\boldsymbol{G}_B = 0$.

2.3 Local/Element level Formulation

For brevity, we will abbreviate the inputs of $\boldsymbol{F}_i$ and $\boldsymbol{\mathfrak{F}}$ to their continuous inputs: $\boldsymbol{F}_i(\boldsymbol{U}, \boldsymbol{U}_i)$ and $\boldsymbol{\mathfrak{F}}(\boldsymbol{U})$.

$$\boldsymbol{G}_b^e = \int_{\Omega^e} \Bigg\{ N_b^e(\boldsymbol{x}) \Bigg[\sum_{a=1}^{n_{en}} N_a^e(\boldsymbol{x}) \boldsymbol{U}_{a,t}^e - \boldsymbol{\mathfrak{F}}(\boldsymbol{U}) \Bigg] - N_{b,i}^e(\boldsymbol{x}) \boldsymbol{F}_i(\boldsymbol{U}, \boldsymbol{U}_{,i}) \Bigg\} \, \mathrm{d}\Omega^e + \int_{\Gamma^e} N_b^e(\boldsymbol{x}) \boldsymbol{F}_i(\boldsymbol{U}, \boldsymbol{U}_{,i}) \hat{n}_i \, \mathrm{d}\Gamma^e \tag{19}$$

Notes:

1. n_{en} is the number of nodes per element
2. ^{e} is the element ID number ($e = \{1...n_n\}$),
3. $_a$ is the node ID number relative to the current element, $a = \{1...n_{en}\}$
4. $\boldsymbol{U} = \sum_{a=1}^{n_{en}} N_a^e(\boldsymbol{x}) \boldsymbol{U}_a^e$, $\boldsymbol{U}_i = \sum_{a=1}^{n_{en}} N_{a,i}^e(\boldsymbol{x}) \boldsymbol{U}_a^e$
5. $\Gamma^e \subset \Gamma$. In words, Γ^e is a subset (small part) of the global Γ (ie. is not the boundary of Ω^e)

Assembly (Local to Global) The values at the local level must be brought up to the global level, in a process called assembly. This will be denoted by an $\mathbb{A}$. For a 1-D problem with 3 elements (and 4 nodes), this takes the form of:

$$\boldsymbol{G}_B = \mathbb{A}\, \boldsymbol{G}_b^e \Rightarrow \begin{cases} \boldsymbol{G}_1 = \boldsymbol{G}_1^1 \\ \boldsymbol{G}_2 = \boldsymbol{G}_2^1 + \boldsymbol{G}_1^2 \\ \boldsymbol{G}_3 = \boldsymbol{G}_2^2 + \boldsymbol{G}_1^3 \\ \boldsymbol{G}_4 = \boldsymbol{G}_2^3 \end{cases} \tag{20}$$

Each $\boldsymbol{G}_a^e$ in the array represents a single N_a shape function.

3 Random Notes

RHS and LHS in the code refer to the Newton's method variation (see page 12b and 13 in notes).

RHS is the stabilized residual term:

$$\hat{\boldsymbol{G}}_B \tag{21}$$

and LHS is the mass matrix

$$\sum \frac{\partial \hat{\boldsymbol{G}}_B}{\partial \boldsymbol{Y}} \tag{22}$$

3.1 Term Definition

Term	Definition
`rlYi`	$\boldsymbol{A}_i \boldsymbol{Y}_{,i}$
`ri`	building of residual RHS (I think)
`EGmass`	mass matrix, $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{Y}}$

3.2 Meaning of `n`...

Term	Definition	Source/Relevant Reference
`nsd`	number of spacial dimensions	`common/common.h#343`
`nflow`	number of flow variables (ie. size of $\boldsymbol{Y}$)	?
`nshape`	number of interior element shape functions	`common/common.h#444`
`ngauss`	number of interior element integration points	`common/common.h#447`
`npro`	number of elements processed in a single call of `e3.f`	Jansen lecture
`npro`	number of virtual processors for the current block	`common/common/h#586`
`nen`	maximum number of element nodes	`common/common.h#341`
`nQpt`	number of quadrature points per element	`common/shp4t.f#14`
`nshl`	number of shape functions per element	`common/genblkPosix.f#70`
`nshg`	global number of shape functions	`common/common.h#354`
`nenl`	number of element nodes for current block	`common/common.h#382`
`nedof`	total number of degrees of freedom	`common/e3.f#35,344`