

Notes on the Fundamental Equations of PHASTA

James Wright

October 30, 2019

1 Shorthand

$$\phi_{,t} = \frac{\partial \phi}{\partial t} \quad (1)$$

$$\phi_{,i} = \frac{\partial \phi}{\partial x_i} \quad (2)$$

$$u_{i,t} = \frac{\partial u_i}{\partial t} \quad (3)$$

$$u_{i,j} = \frac{\partial u_i}{\partial x_j} \quad (4)$$

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

2 Fundamental Fluid Equations

Unsteady Compressible Navier-Stokes (UCNS) equations:

2.1 Traditional Form

Continuity

$$\rho_{,t} + [\rho u_j]_{,j} = 0 \quad (6)$$

Momentum

$$[\rho u_i]_{,t} + [\rho u_i u_j]_{,j} + p_{,i} = \tau_{ij,j} + b_i \quad (7)$$

Energy

$$[\rho e_{tot}]_{,t} + [\rho e_{tot} u_j]_{,j} + [\rho u_j]_{,j} = [\tau_{ij} u_j]_{,j} + b_i u_j + r + q_{i,i} \quad (8)$$

2.2 Conservative Vectorized Form

$$\mathbf{U} \equiv \begin{Bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho e_{tot} \end{Bmatrix} = \begin{Bmatrix} \rho \\ \rho u_j \\ \rho e_{tot} \end{Bmatrix} \quad (9)$$

Flux Vector:

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

Also:

$$\mathbf{F}_i^{\text{adv}} = u_i \mathbf{U} + \begin{Bmatrix} 0 \\ p \delta_{ij} \\ u_i p \end{Bmatrix} \quad (11)$$

Source Vector:

$$\mathcal{F} = \begin{Bmatrix} 0 \\ b_j \\ b_j u_j + r \end{Bmatrix} \quad (12)$$

These terms combine together to form:

$$\mathbf{U}_{,t} + \mathbf{F}_{i,i} = \mathcal{F} \quad (13)$$

3 Finite Element Discretization

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

$$\mathbf{U}_{,t} + \mathbf{F}_{i,i} - \mathcal{F} = \mathbf{0} \quad (14)$$

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

$$\mathbf{W} \cdot (\mathbf{U}_{,t} + \mathbf{F}_{i,i} - \mathcal{F}) = 0 \quad (15)$$

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

$$\int_{\Omega} \mathbf{W} \cdot (\mathbf{U}_{,t} + \mathbf{F}_{i,i} - \mathcal{F}) \, d\Omega = 0 \quad (16)$$

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

$$\int_{\Omega} \{ \mathbf{W} \cdot \mathbf{U}_{,t} - \mathbf{W}_{,i} \cdot \mathbf{F}_i(\mathbf{U}, \mathbf{U}_{,i}) - \mathbf{W} \cdot \mathcal{F}(\mathbf{U}) \} d\Omega + \int_{\Gamma} \mathbf{W} \cdot \mathbf{F}_i(\mathbf{U}, \mathbf{U}_{,i}) \cdot \hat{n}_i d\Gamma = 0 \quad (17)$$

where Γ is the boundary of the domain Ω and \hat{n}_i is the normal unit vector of the boundary surface. Equation (17) represents the **Weak UCNS in IBP Form**. The fact that \mathbf{F}_i and \mathcal{F} are functions of \mathbf{U} and $\mathbf{U}_{,i}$ is explicitly shown here.

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 ϕ into

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

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: \mathbf{U} , $\mathbf{U}_{,t}$, and \mathbf{W} :

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

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

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

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

3.1.2 Discretize UCNS

Let's substitute the expressions in eq. (19) into the weak UCNS form in eq. (17).

$$\begin{aligned}
& \int_{\Omega} \left\{ \sum_{B=1}^{n_n} N_B(\mathbf{x}) \mathbf{W}_B \cdot \sum_{A=1}^{n_n} N_A(\mathbf{x}) \mathbf{U}_{A,t} - \sum_{B=1}^{n_n} N_{B,i}(\mathbf{x}) \mathbf{W}_B \mathbf{F}_i \left(\sum_{A=1}^{n_n} N_A(\mathbf{x}) \mathbf{U}_A, \sum_{A=1}^{n_n} N_{A,i}(\mathbf{x}) \mathbf{U}_A \right) \right. \\
& \quad \left. - \sum_{B=1}^{n_n} \mathbf{W}_B \mathcal{F} \left(\sum_{A=1}^{n_n} N_A(\mathbf{x}) \mathbf{U}_A \right) \right\} d\Omega \\
& + \int_{\Gamma} \sum_{B=1}^{n_n} \mathbf{W}_B \cdot \mathbf{F}_i \left(\sum_{A=1}^{n_n} N_A(\mathbf{x}) \mathbf{U}_A, \sum_{A=1}^{n_n} N_{A,i}(\mathbf{x}) \mathbf{U}_A \right) \hat{n}_i d\Gamma = 0 \quad (20)
\end{aligned}$$

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

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

$$\begin{aligned}
& \sum_{B=1}^{n_n} \mathbf{W}_B \left\{ \int_{\Omega} \left\{ N_B(\mathbf{x}) \cdot \sum_{A=1}^{n_n} N_A(\mathbf{x}) \mathbf{U}_{A,t} - N_{B,i}(\mathbf{x}) \mathbf{F}_i \left(\sum_{A=1}^{n_n} N_A(\mathbf{x}) \mathbf{U}_A, \sum_{A=1}^{n_n} N_{A,i}(\mathbf{x}) \mathbf{U}_A \right) \right. \right. \\
& \quad \left. \left. - \mathcal{F} \left(\sum_{A=1}^{n_n} N_A(\mathbf{x}) \mathbf{U}_A \right) \right\} d\Omega \right. \\
& \quad \left. + \int_{\Gamma} \mathbf{F}_i \left(\sum_{A=1}^{n_n} N_A(\mathbf{x}) \mathbf{U}_A, \sum_{A=1}^{n_n} N_{A,i}(\mathbf{x}) \mathbf{U}_A \right) \hat{n}_i d\Gamma \right\} = 0 \quad (21)
\end{aligned}$$

$$eq. (21) \Rightarrow \sum_{B=1}^{n_n} \mathbf{W}_B \cdot \mathbf{G}_B = 0 \quad (22)$$

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

3.1.3 Local/Element level Formulation

For brevity, we will abbreviate the inputs of \mathbf{F}_i and \mathcal{F} to their continuos inputs: $\mathbf{F}_i(\mathbf{U}, \mathbf{U}_i)$ and $\mathcal{F}(\mathbf{U})$.

$$\begin{aligned}
\mathbf{G}_b^e = & \int_{\Omega^e} \left\{ N_b^e(\mathbf{x}) \left[\sum_{a=1}^{n_{en}} N_a^e(\mathbf{x}) \mathbf{U}_{a,t}^e - \mathcal{F}(\mathbf{U}) \right] - N_{b,i}^e(\mathbf{x}) \mathbf{F}_i(\mathbf{U}_a^e, \mathbf{U}_{a,i}^e) \right\} d\Omega^e \\
& + \int_{\Gamma^e} N_b^e(\mathbf{x}) \mathbf{F}_i(\mathbf{U}_a^e, \mathbf{U}_{a,i}^e) \hat{n}_i d\Gamma^e \quad (23)
\end{aligned}$$

Notes:

1. n_{en} is the number of nodes per element
2. e is the element ID number ($e = \{1 \dots n_n\}$),

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

4 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:

$$\hat{\mathbf{G}}_B \quad (24)$$

and LHS is the mass matrix

$$\sum \frac{d\hat{\mathbf{G}}_B}{d\mathbf{Y}} \quad (25)$$

4.1 Term Definition

Term	Definition
<code>rlYi</code>	$\mathbf{A}_i \mathbf{Y}_{,i}$

4.2 Meaning of n . . .

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