ASEN 5331 - HW3

James Wright

October 24, 2019

Using the code that is posted on the web page, associate the following concepts to both your notes and to the code. By associate, I mean a one-to-one correspondence between the variables in the notes and the variables in the code (complete with the range of their indices when they are arrays). Also, identify the routines where each of these variables get updated and thereby carry out the algorithms we have described in the notes (again mapping the algorithms from the notes to routines that carry them out)

0.1 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`

1 Gradient of shape function in the parent domain

$$N_{a,\xi_l}(\boldsymbol{\xi}^k) \tag{1}$$

Represented by `shgl (nsd,nen,nQpt)` in `common/shpt10t.f`. It appears that the `shgl` is defined in either `common/{shp10t.f,shp4t.f}`. Note this never gets called, but all other references use `shgl` as an input, so I can't find where else `shgl` would be defined.

Note also there are different definitions to the size of `shgl` in different functions:

`shgl (nsd,nshl,ngauss)`	`common/e3.f`
`shgl (npro,nsd,nshape)`	`common/e3qvar.f`
`shgl (nsd, nen)`	`common/e3ivar.f`

Using the definition in `common/e3.f`, `nsd` $=\ell$, `nshl` $=a$, `ngauss`$=k$

2 Gradient of shape function in real domain

$$N_{a,i} = \frac{\partial N_a}{\partial x_i} \tag{2}$$

Possibly `shg(npro,nshl,nsd)` is computed in `compressible/e3metric.f#64`

Note that `compressible/e3qvar.f` defines `shg(npro,nshape,nsd)` but `compressible/e3ivar.f` defines `shg(npro,nshl,nsd)` but

3 Metric tensor that maps §1 to §2

$$\boldsymbol{\xi_{j,i}} \tag{3}$$

for

$$N_{a,i} = N_{a,\xi_j}\boldsymbol{\xi_{j,i}} \tag{4}$$

Written as `dxidx(npro,nsd,nsd)` in `common/e3metric.f`. The index would then be a, j, i.

4 Global Residual

$$\boldsymbol{G}_A = \mathbb{A}\boldsymbol{G}^e_a \tag{5}$$

Stored as `res(nshg,nflow)` which is assembled by the subroutine `common/local.f` which is called from `compressible/asigmr.f#84`.

The `local` subroutine just performs the $\mathbb{A}$ operator.

5 Element Level Residual

$$\boldsymbol{G}^e_a \tag{6}$$

Defined as `rl(npro,nshl,nflow)` in `common/e3.f`. `e3.f` calls `e3wmlt.f`, where `rl` is actually computed in `compressible/e3wmlt.f#66`.

It is a collection of 5x1 vectors, one for each `a` and `e`, where `e=npro` and `a=nshl`

6 Element Level tangent matrix

$$\frac{\mathrm{d}\boldsymbol{G}^e_a}{\mathrm{d}\boldsymbol{Y}_b} \tag{7}$$

Appears to be stored as `EGmass(npro,nedof,nedof)` in `compressible/e3conv.f#314` where `e = npro` and `{a,b} = nedof`.

This is built by inputting the jacobian of the advective flux ($\boldsymbol{A}_i = \boldsymbol{F}_{i,\boldsymbol{Y}}$ written in code as `Ai`) which is calculated in `compressible/e3mtrx.f`, multiplying this by the real space gradient of the shape function in lines `compressible/e3conv.f#201-306` to make the `AiNbi` tensor. `AiNbi` is then multiplied by `shape`

7 The gradient of the solution vector

$$\boldsymbol{Y}_{,i} \tag{8}$$

Appears to be named as `g[1..3]yi(npro,nflow)` and is calculated in `compressible/e3ivar.f#243`.

It is calculated by the sum of the products of the real space gradient shape function `shg` and the element primitive variables `yl`.

8 Jacobian of the map from real domain to parent domain

$$\frac{\partial x}{\partial \xi} \tag{9}$$

Appears to be `dxdxi(npro,nsd,nsd)` in `common/e3metric.f#20`

It is calculated by the multiplying the nodal coordinates `xl` by the gradient of the local shape function `shgl`.

9 The advective/convective flux

$$\boldsymbol{F}_i^{\text{adv}} \tag{10}$$

Defined as `ri (npro, nflow*(nsd+1))` in `compressible/e3conv.f#74`. It is calculated directly using the definition of the advective flux

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

`ri` is formatted such that the $\boldsymbol{F}_1^{\text{adv}}$ is `ri(:,1:5)`

10 The Jacobian of the Advective/Convective Flux

$$\boldsymbol{A}_i = \boldsymbol{F}_{i,\boldsymbol{Y}} \tag{12}$$

Defined in `compressible/e3mtrx.f#132` as `A[1..3] (npro,nflow,nflow)`, it is calculated in a similar direct way as $\boldsymbol{F}_i^{\text{adv}}$, while additionally using `rYli` which is calculated in `compressible/e3conv.f#100`. Note that `rYli` is mascaraded as `e2p, e3p, e4p, drdp,` and `drdT` inside `compressible/e3mtrx.f`.