SU/PG Implementation

Nicholas K. Burgess

February 16, 2014

1 Introduction

Implementing the SU/PG method is a variant of the continuous Galerkin method, which enforces stability by adding a term to the test function. This method is written as

$$\sum_{e\in\mathcal{T}_h}\int_{\Omega_k}\phi_i\frac{\partial\mathbf{u}_h}{\partial t}d\Omega_e-\sum_{e\in\mathcal{T}_h}\int_{\Omega_k}\nabla\phi_i\cdot\left(\vec{\mathbf{F}}_c\left(\mathbf{u}_h\right)-\vec{\mathbf{F}}_v\left(\mathbf{u}_h,\nabla\mathbf{u}_h\right)\right)+\phi_i\mathbf{S}\left(\mathbf{u}_h,\nabla\mathbf{u}_h\right)d\Omega_e+$$
$$\sum_{e\in\mathcal{T}_h}\int_{\Omega_k}\nabla\phi_i\cdot\frac{\partial\vec{\mathbf{F}}_c\left(\mathbf{u}_h\right)}{\partial\mathbf{u}_h}[\tau]\left(\frac{\partial\mathbf{u}_h}{\partial t}+\nabla\cdot\left(\vec{\mathbf{F}}_c\left(\mathbf{u}_h\right)-\vec{\mathbf{F}}_v\left(\mathbf{u}_h,\nabla\mathbf{u}_h\right)\right)+\mathbf{S}\left(\mathbf{u}_h,\nabla\mathbf{u}_h\right)\right)d\Omega_e+$$
$$\sum_{b\in\mathcal{B}_h}\int_{\Gamma^b}\vec{n}\cdot\left(\vec{\mathbf{F}}_c\left(\mathbf{u}_h\right)-\vec{\mathbf{F}}_v\left(\mathbf{u}_h,\nabla\mathbf{u}_h\right)\right)ds \tag{1}$$

The above equation involves gradients of test functions ϕ_i in the physical space $\vec{x}\in\mathbb{R}^d$ where d is the number of physical dimensions. The basis functions are normally defined in a reference space with $\vec{\xi}\in[-1,1]^d$. One can define the physical coordinates $\vec{x}$ using a mapping function $\vec{x}\left(\vec{\xi}\right):[-1,1]^d\mapsto\mathbb{R}^d$. This definition allows one to write the gradient in the physical space as

$$\nabla=\frac{\partial\vec{\xi}}{\partial\vec{x}}\cdot\nabla_{\vec{\xi}} \tag{2}$$

However, the relation $\vec{\xi}(\vec{x})$ is unknown. Therefore, one normally uses the following relationship

$$\begin{aligned}\frac{\partial\vec{x}}{\partial\vec{\xi}}&:=[J]\\\frac{\partial\vec{x}}{\partial\vec{\xi}}&:=[J]^{-1}\end{aligned} \tag{3}$$

Therefore the gradiient in the physical space can be written as

$$\nabla=[J]^{-1}\nabla_{\vec{\xi}} \tag{4}$$

Now consider the dot product of the gradient (i.e. divergence) with a vector $\vec{\mathbf{F}}$ is written as

$$\begin{aligned}\nabla\cdot\vec{\mathbf{F}} &= [J]^{-1}\nabla_{\xi}\cdot\vec{\mathbf{F}}\\ \nabla\cdot\vec{\mathbf{F}} &= \frac{\partial\xi_j}{\partial x_i}\frac{\partial}{\partial\xi_j}\mathbf{F}_i\end{aligned}\tag{5}$$

which can be re-arranged as

$$\nabla\cdot\vec{\mathbf{F}} = \frac{\partial}{\partial\xi_j}\frac{\partial\xi_j}{\partial x_i}\mathbf{F}_i \tag{6}$$

If one defines a new flux-vector $\mathbf{E}_j = \frac{\partial\xi_j}{\partial x_i}\mathbf{F}_i$. The equation simply can be written as

$$\mathbf{E}_j = n_i\mathbf{F}_i \tag{7}$$

When programing this method one uses the above equation and sets $n_i = \frac{\partial\xi_j}{\partial x_i}$ for a particular $\mathbf{E}_j$. Now considering that the divergence operator is just the dot product of the gradient with a vector one can re-write many of the operation in Eq. (1) in the spirit of the above manipulations. So Eq. (1) can be re-written as

$$\begin{aligned}&\sum_{e\in\mathcal{T}_h}\int_{\Omega_k}\phi_i\frac{\partial\mathbf{u}_h}{\partial t}d\Omega_e - \sum_{e\in\mathcal{T}_h}\int_{\Omega_k}\nabla_{\vec{\xi}}\phi_i\cdot\Big(\vec{\mathbf{E}}_c(\mathbf{u}_h)-\vec{\mathbf{E}}_v(\mathbf{u}_h,\nabla\mathbf{u}_h)\Big)+\phi_i\mathbf{S}(\mathbf{u}_h,\nabla\mathbf{u}_h)\,d\Omega_e+\\ &\sum_{e\in\mathcal{T}_h}\int_{\Omega_k}\nabla_{\vec{\xi}}\phi_i\cdot\frac{\partial\vec{\mathbf{E}}_c(\mathbf{u}_h)}{\partial\mathbf{u}_h}[\tau]\left(\frac{\partial\mathbf{u}_h}{\partial t}+\nabla_{\vec{\xi}}\cdot\Big(\vec{\mathbf{E}}_c(\mathbf{u}_h)-\vec{\mathbf{E}}_v(\mathbf{u}_h,\nabla\mathbf{u}_h)\Big)+\mathbf{S}(\mathbf{u}_h,\nabla\mathbf{u}_h)\right)d\Omega_e+\\ &\sum_{b\in\mathcal{B}_h}\int_{\Gamma^b}\vec{n}\cdot\Big(\vec{\mathbf{F}}_c(\mathbf{u}_h)-\vec{\mathbf{F}}_v(\mathbf{u}_h,\nabla\mathbf{u}_h)\Big)\,ds\end{aligned}\tag{8}$$

Examing equation indicates how many functions are required to compute the SU/PG residual. In practice one needs only a few functions that when used in the proper sequence can generate the SU/PG residual. To simplify the concepts we will write the discretization in terms a general fluxes $\mathbf{F}$ and $\mathbf{E}$ both of which can in general depend on $\mathbf{u}_h$ and $\nabla\mathbf{u}_h$.

$$\begin{aligned}&\sum_{e\in\mathcal{T}_h}\int_{\Omega_k}\phi_i\frac{\partial\mathbf{u}_h}{\partial t}d\Omega_e - \sum_{e\in\mathcal{T}_h}\int_{\Omega_k}\nabla_{\vec{\xi}}\phi_i\cdot\vec{\mathbf{E}}+\phi_i\mathbf{S}(\mathbf{u}_h,\nabla\mathbf{u}_h)\,d\Omega_e+\\ &\sum_{e\in\mathcal{T}_h}\int_{\Omega_k}\nabla_{\vec{\xi}}\phi_i\cdot\frac{\partial\vec{\mathbf{E}}_c(\mathbf{u}_h)}{\partial\mathbf{u}_h}[\tau]\left(\frac{\partial\mathbf{u}_h}{\partial t}+\nabla_{\vec{\xi}}\cdot\vec{\mathbf{E}}+\mathbf{S}(\mathbf{u}_h,\nabla\mathbf{u}_h)\right)d\Omega_e+\\ &\sum_{b\in\mathcal{B}_h}\int_{\Gamma^b}\vec{n}\cdot\vec{\mathbf{F}}ds\end{aligned}\tag{9}$$

Due to the non-linearity of the fluxes the methods of computing the divergence is as follows

$$\nabla_{\vec{\xi}}\cdot\vec{\mathbf{E}} = \frac{\partial\vec{\mathbf{E}}}{\partial\mathbf{u}_h}\cdot\nabla_{\vec{\xi}}\mathbf{u}_h+ \tag{10}$$

If one carelly examines these equations one can deduce that all operations required by the SU/PG method effectively become computations of the form

$$\mathbf{E}_j = n_i \mathbf{F}_i \tag{11}$$

and operations of the flux jacobian of $\mathbf{E}$ on a vector V over the number of equations.

$$\frac{\partial \mathbf{E}}{\partial \mathbf{u}_h} \cdot \mathbf{V} = n_i \frac{\partial \mathbf{F}_i}{\partial \mathbf{u}_h} \cdot \mathbf{V} \tag{12}$$

Finally a method is required to compute the product $[\tau]\,\mathbf{V}$. However, since an explicit expression is only availible for $[\tau]^1$. Therefore, $[\tau]$ is never explicitly formed rather it's product onto a vector is formed by recalling

$$[\tau]\,V = \left([\tau]^{-1}\right)^{-1} V = W \tag{13}$$

which is simpliy the solution of

$$[\tau]^{-1}\,W = V; \tag{14}$$

Thus the product of $[\tau]$V is implemented as a linear solve operation. The SU/PG residual is formed using the following algorithm

Algorithm 1 :SU/PG Residual Formation Algorithm

$\mathbf{R}(:) = 0$
for qp = 0; qp $<$ nqp; qp++ **do**
  $\nabla \cdot \mathbf{F} = 0$
  for j = 0; j $<$ d; j++ **do**
    Form $\vec{n} = [J(:,d)]^{-1}$
    Compute $\mathbf{E}_j = \vec{\mathbf{F}} \cdot \vec{n}$
    for i = 0; i $<$ NDOF; i++ **do**
      $\mathbf{R}_i += \frac{\partial \phi_i}{\partial \xi_j} \mathbf{E}_j w_q(qp) Det(J)$
    end for
    Compute $\nabla \cdot \vec{\mathbf{F}} += \frac{\partial \mathbf{E}_j}{\partial \mathbf{u}_h} \frac{\partial \mathbf{u}_h}{\partial \xi_j}$
  end for
  Compute $[\tau]^{-1}$
  Compute $[\tau]\,\nabla \cdot \vec{\mathbf{F}}$ via solving $[\tau]^{-1}\,\mathbf{x} = \nabla \cdot \nabla \vec{\mathbf{F}}$
  for j = 0; j $<$ d; j++ **do**
    Form $\vec{n} = [J(:,d)]^{-1}$
    Compute $D = \frac{\partial \mathbf{E}_j}{\partial \mathbf{u}_h} \mathbf{x}$
    for i = 0; i $<$ NDOF; i++ **do**
      $\mathbf{R}_i += \frac{\partial \phi_i}{\partial \xi_j} \mathbf{D} w_q(qp) Det(J)$
    end for
  end for
end for

Examination of the algorithm shows that one requires only 3 functions:

1. Compute $\mathbf{E}_j$
2. Compute $\frac{\partial \mathbf{E}_j}{\partial \mathbf{u}_h} \cdot \mathbf{y}$
3. Compute $[\tau]$.

While it may seem that there is a missing function for inverting the τ matrix, this functionality has been provided as part of the square matrix class.