

The details of the numerical setup are presented in Section ??.

We wish to use $Q_1 \times Q_1$ element, which, unless stabilised, violates the LBB stability condition and therefore is unusable. Stabilisation can be of two types: least-squares [?, ?, ?, ?], or by means of an additional term in the weak form as first introduced in [?, ?]. It is further analysed in [?, ?, ?, ?, ?]. Note that an equal-order velocity-pressure formulation that does not exhibit spurious pressure modes (without stabilisation) has been presented in [?].

This element corresponds to bilinear velocities, bilinear pressure (equal order interpolation for both velocity and pressure) which is very convenient in terms of data structures since all dofs are colocated.

In geodynamics, it is used in the Rhea code [?, ?] and in Gale [?]. It is also used in [?] in its stabilised form, in conjunction with AMR. This element is quickly discussed at page 217 of Volker John's book [?].

The stabilisation term \mathbb{C} enters the Stokes matrix in the (2,2) position:

$$\begin{pmatrix} \mathbb{K} & \mathbb{G} \\ \mathbb{G}^T & \mathbb{C} \end{pmatrix} \cdot \begin{pmatrix} \mathcal{V} \\ \mathcal{P} \end{pmatrix} = \begin{pmatrix} f \\ h \end{pmatrix}$$

The purpose of the \mathbb{C} term is to stabilise the linear system. It is given by:

$$\mathbb{C}(p, q) = \sum_e \int_{\Omega_e} \frac{1}{\eta} (p - \Pi p)(q - \Pi q) d\Omega$$

where Π is the L^2 -projection onto the space of element-wise constant functions:

$$\Pi p = \frac{1}{|\Omega_e|} \int_{\Omega_e} p d\Omega$$

Because of the stabilisation matrix \mathbb{C} , the numerical solution satisfies the incompressibility condition only approximately. Local mesh refinement helps to control these unwanted effects [?, ?]. Since \mathbb{K} and \mathbb{C} are symmetric matrices, the Stokes system is then an indefinite symmetric system.

One can further expand the above expression for the \mathbb{C} term:

$$\mathbb{C}(p, q) = \sum_e \int_{\Omega_e} \frac{1}{\eta} (p - \Pi p)(q - \Pi q) d\Omega \quad (1)$$

$$= \sum_e \int_{\Omega_e} \frac{1}{\eta} [pq - (\Pi p)q - (\Pi q)p + (\Pi p)(\Pi q)] d\Omega \quad (2)$$

$$= \sum_e \frac{1}{\eta_e} \left[\int_{\Omega_e} pq d\Omega - \int_{\Omega_e} (\Pi p)q d\Omega - \int_{\Omega_e} (\Pi q)p d\Omega + \int_{\Omega_e} (\Pi p)(\Pi q) d\Omega \right] \quad (3)$$

$$= \sum_e \frac{1}{\eta_e} \left[\int_{\Omega_e} pq d\Omega - (\Pi p) \int_{\Omega_e} q d\Omega - (\Pi q) \int_{\Omega_e} p d\Omega + (\Pi p)(\Pi q) \int_{\Omega_e} d\Omega \right] \quad (4)$$

$$= \sum_e \frac{1}{\eta_e} \left[\int_{\Omega_e} pq d\Omega - (\Pi p)|\Omega_e|(\Pi q) - (\Pi q)|\Omega_e|(\Pi p) + (\Pi p)(\Pi q)|\Omega_e| \right] \quad (5)$$

$$= \sum_e \frac{1}{\eta_e} \left[\int_{\Omega_e} pq d\Omega - |\Omega_e|(\Pi p)(\Pi q) \right] \quad (6)$$

where we have used the fact that on each element Πp is constant. The left term will obviously yield a Q_1 mass matrix (scaled by the elemental viscosities). Note that this approach is not used in practice as we'll see hereafter.

The pressure inside an element is given by

$$p^h(\vec{x}) = \sum_k N_k^p(\vec{x}) p_k$$

so that

$$\Pi p^h = \frac{1}{|\Omega_e|} \int_{\Omega_e} \sum_k N_k^p p_k d\Omega = \sum_k \left(\underbrace{\frac{1}{|\Omega_e|} \int_{\Omega_e} N_k^p d\Omega}_{\tilde{N}_k^p} \right) p_k \quad (7)$$

$$(8)$$

and then

$$p^h - \Pi p^h = \sum_k N_k^p(\vec{x}) p_k - \sum_k \tilde{N}_k^p p_k = \sum_k (N_k^p(\vec{x}) - \tilde{N}_k^p) p_k$$

The algorithm is straightforward and as follows: In the loop over elements, a) Compute the average of each shape function $N_k^p(\vec{x})$ over the element; b) Subtract this average to the shape function; c) Build mass matrix with modified/offset shape functions (taking in account the viscosity).

In the case of rectangular elements of size h_x, h_y , Πp simplifies even more:

$$\tilde{N}_k^p = \frac{1}{|\Omega_e|} \int_{\Omega_e} N_k^p(\vec{x}) d\Omega = \frac{1}{h_x h_y} \frac{h_x h_y}{4} \int_{-1}^{+1} \int_{-1}^{+1} N_k^p(r, s) dr ds = \frac{1}{4} \int_{-1}^{+1} \int_{-1}^{+1} N_k^p(r, s) dr ds \quad (9)$$

It is easy to show that the average of the Q_1 shape functions over the reference element is 1, so that $\tilde{N}_k^p = 1/4$. This explains why in the code we have:

```
Navrg = np.zeros(m, dtype=np.float64)
Navrg[0]=0.25
Navrg[1]=0.25
Navrg[2]=0.25
Navrg[3]=0.25
```

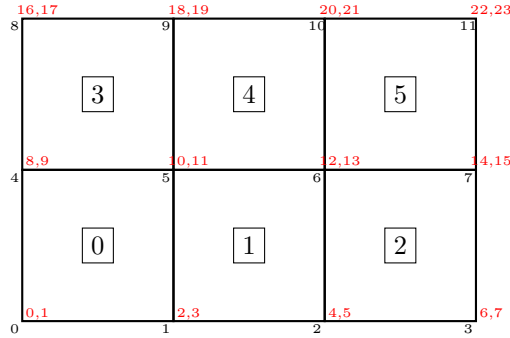
The elemental \mathbb{C} matrix is then computed like a mass matrix. Inside the loop over quadrature points, we do:

```
Nvect[0,0:m]=N[0:m]-Navrg[0:m]
C_el+=Nvect.T.dot(Nvect)*jacob*weightq/viscosity(xq,yq,case)
```

It is then assembled inside the big FEM matrix

```
for k1 in range(0,m):
    for k2 in range(0,m):
        C_mat[icon[k1,iel],icon[k2,iel]]+=C_el[k1,k2]
```

Non-zero pattern of the \mathbb{C} matrix: Let us take a simple example: a 3x2 element grid.



The \mathbb{K} matrix is of size $NfemV \times NfemV$ with $NfemV = ndofV \times nnp = 2 \times 12 = 24$. The \mathbb{G} matrix is of size $NfemV \times NfemP$ with $NfemP = ndofP \times nnp = 1 \times 12 = 12$. The \mathbb{C} matrix is of size $NfemP \times NfemP$.

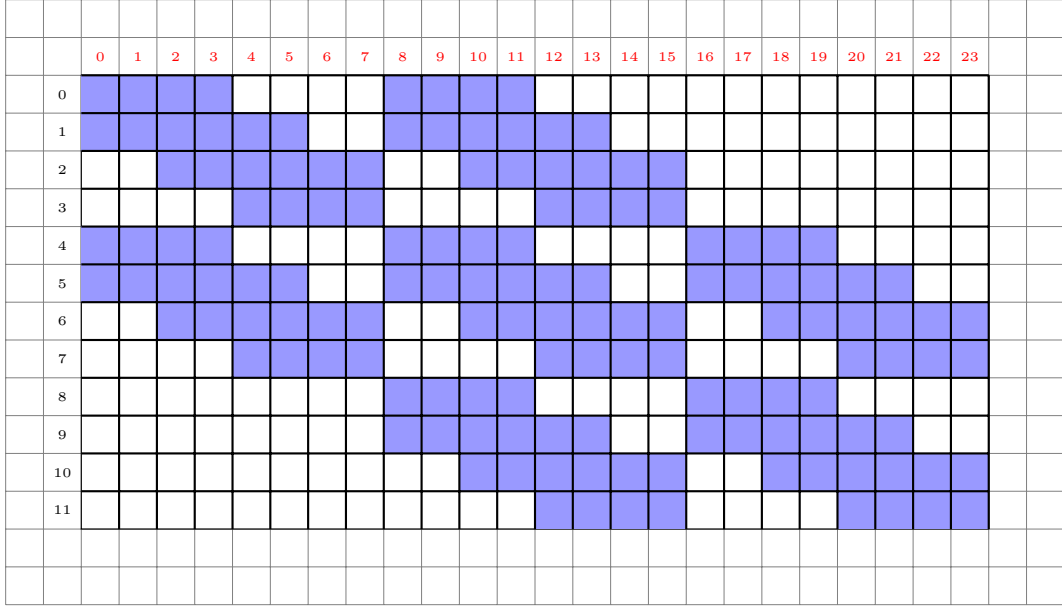
A corner pdof sees 4 vdofs, a side pdof sees 12 vdofs and an inside pdof sees 18 vdofs, so that the total number of nonzeros in \mathbb{G} can be computed as follows:

$$NZ_{\mathbb{G}} = \underbrace{4}_{\text{corners}} + \underbrace{2(nnx-2)*12}_{2\text{hor.sides}} + \underbrace{2(nny-2)*12}_{2\text{vert.sides}} + \underbrace{(nnx-2)(nny-2)*18}_{\text{insidenodes}}$$

Concretely,

- pdof #0 sees vdofs 0,1,2,3,8,9,10,11
- pdof #1 sees vdofs 0,1,2,3,4,5,8,9,10,11,12,13
- pdof #5 sees vdofs 0,1,2,3,4,5,8,9,10,11,12,13,16,17,18,19,20,21

so that the \mathbb{G}^T matrix non-zero structure then is as follows:

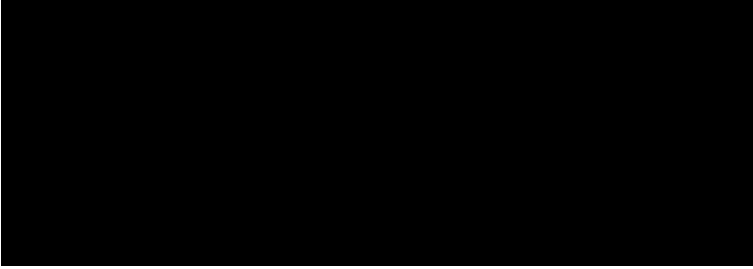


We impose $\int p dV = 0$ which means that the following constraint is added to the Stokes matrix:

$$\begin{pmatrix} \mathbb{K} & \mathbb{G} & 0 \\ \mathbb{G}^T & \mathbb{C} & \mathbb{L} \\ 0 & \mathbb{L}^T & 0 \end{pmatrix} \cdot \begin{pmatrix} \mathcal{V} \\ \mathcal{P} \\ \lambda \end{pmatrix} = \begin{pmatrix} f \\ h \\ 0 \end{pmatrix}$$

0.1 The Donea & Huerta benchmark

As in [?] we solve the benchmark problem presented in section ??.



0.2 The Dohrmann & Bochev benchmark

As in [?] we solve the benchmark problem presented in section ??.



compare my rates with original paper!

0.3 The falling block experiment

The setup is described in [?].

