MAC Discretization

of Incompressible NS Equations

A. Donev, CFD Spring 2013

Recall the projected (pressure-free) formulation of the NS equations:

$$v_t = P\left[-v\cdot\nabla v + \nu\nabla^2 v + f\right]$$

where
$$Pv = v - \nabla(\nabla^{-2})\nabla\cdot v$$
$$P \equiv I - \nabla(\nabla^{-2})\nabla\cdot$$

How do we discretize this on a finite volume / difference grid in 2D?

We need to decide where:

- velocity lives (what its grid is)
- divergence lives (scalar grid)

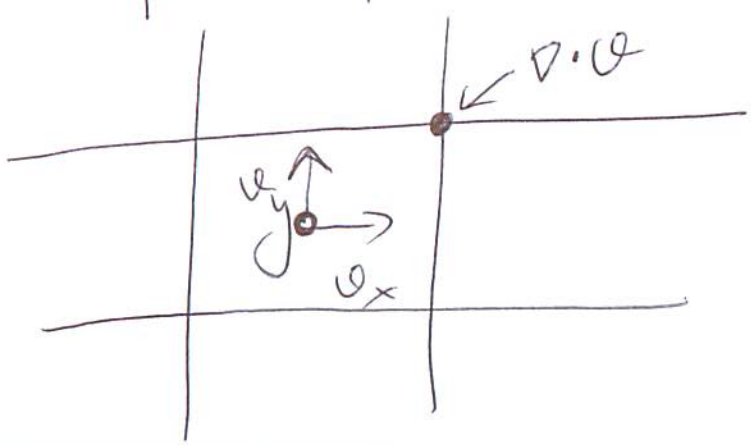
Options include:

1

<u>Fully centered</u>:

- v_x, v_y and $\nabla\cdot v$ all represented on the same grid (<u>cell-centered</u>)

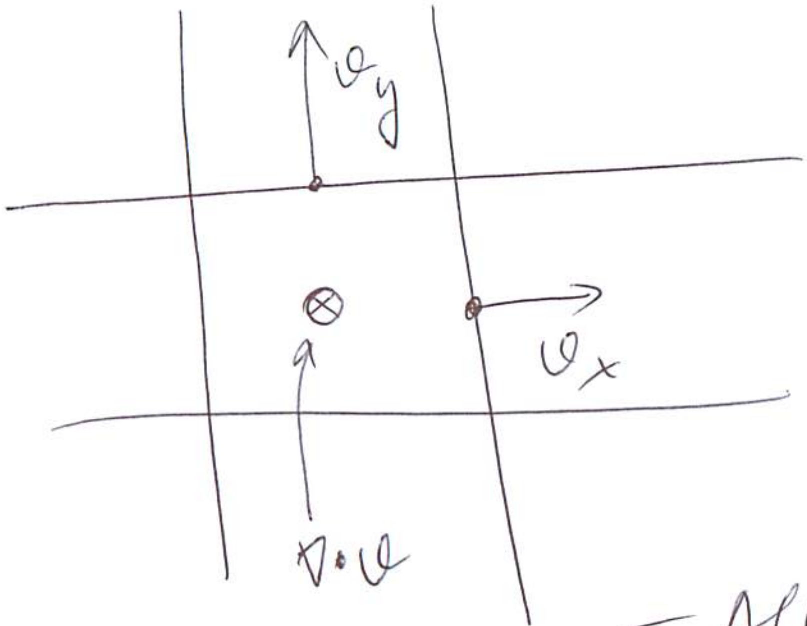
2



<u>Node-based</u>

- v_x, v_y at cell centers
- $\nabla\cdot v$ at cell <u>nodes</u>

3 Staggered grid or "Marker-and-Cell" MAC grid



- All vector fields split on 2 grids, one for x components, one for y components, face-centered
- All scalars cell-centered

All three grids have a history and advantages/disadvantages. Similarly in finite-element methods there are different elements (velocity-pressure spaces)

Once you decide the grid for v and $\nabla \cdot v$, define discrete difference operators to approximate the continuum differential ones:

$$\mathbb{P} = I - G L^{-1} D$$

$\mathbb{P}$: discrete projection

G: discrete Gradient

L: discrete Laplacian

D: discrete divergence

D : vector $\longrightarrow$ scalar

L : scalar $\longleftrightarrow$ scalar

G : scalar $\rightarrow$ vector

A key property of P is that it is idempotent, as any true projection has to be:

$$P^2 \vartheta = P(P\vartheta) = 0 \quad \Rightarrow \quad \boxed{P^2 = P}$$

because

$$\nabla \cdot (P\vartheta) = \nabla\cdot\vartheta - (\nabla\cdot\nabla)\nabla^{-2}(\nabla\cdot\vartheta) =$$

$$= \nabla\cdot\vartheta - \nabla^2\nabla^{-2}(\nabla\cdot\vartheta) = 0$$

We want to mimic this in the discrete projection $\mathbb{P}$

$$\mathbb{P}^2 = \mathbb{P}$$

One simple and good way to achieve this is to mimick: ⑥

$$\boxed{DG \equiv L}$$

We would also like L to be <u>symmetric negative semidefinite</u>:

$$\boxed{G = -D^*}$$ ← adjoint operator

So once we know D, we know "everything" and can construct $\mathbb{P}$.

We have choices: $v = (u, v)$

A Fully-centered

1 $$(\nabla \cdot v)_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y} = (Dv)_{i,j}$$

This defines a matrix $D_{i,j;i',j'}$

2 $G = -D^*$

$$(\nabla \Pi)_{i,j} = \begin{bmatrix} (\Pi_{i+1,j} - \Pi_{i-1,j})/2\Delta x \\ (\Pi_{i,j+1} - \Pi_{i,j-1})/2\Delta y \end{bmatrix}$$

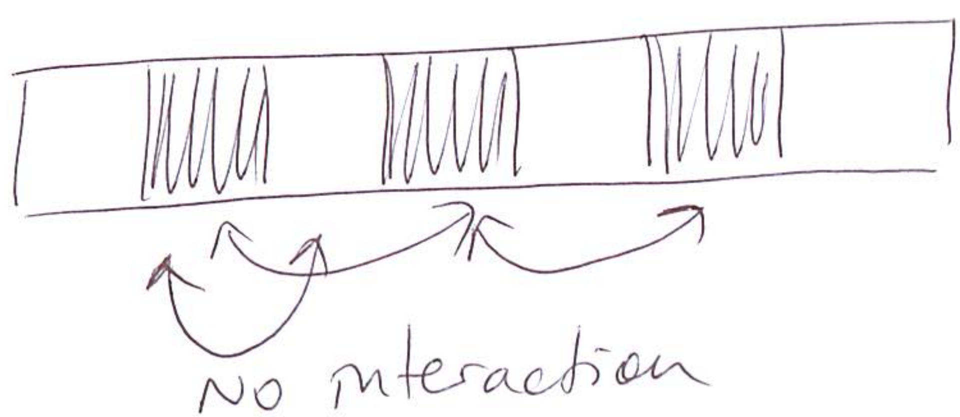
3 $L = DG$

$$(L\Pi)_{i,j} = \begin{matrix} (\Pi_{i+2,j} - 2\Pi_{i,j} + \Pi_{i-2,j})/(2\Delta x)^2 + \\ (\Pi_{i,j+2} - 2\Pi_{i,j} + \Pi_{i,j-2})/(2\Delta y)^2 \end{matrix}$$

So the Laplacian we get is not the usual one but rather a <u>wider Laplacian</u>:

$$\frac{d^2 f}{dx^2} \simeq \frac{f(x+2\Delta x) - 2f(x) + f(x-2\Delta x)}{(2\Delta x)^2}$$

For this Laplacian, the <u>odd</u> and the <u>even</u> points are completely <u>decoupled</u>



This means that the Laplacian $L = DG$ has a <u>nontrivial null space</u> (checkerboard modes)

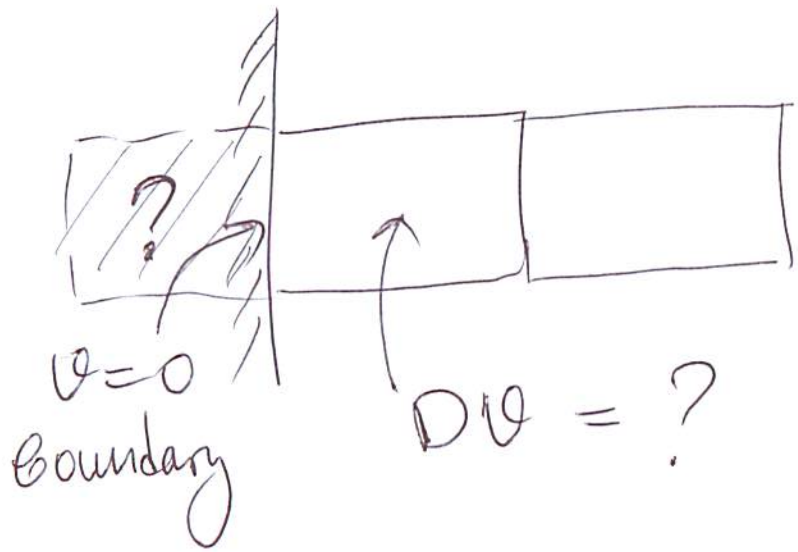
$$u = \begin{array}{|c|c|c|c|c|c|} \hline \alpha & \beta & \alpha & \beta & \alpha & \beta \\ \hline \beta & \alpha & \beta & \alpha & \beta & \alpha \\ \hline \alpha & \beta & \alpha & \beta & \alpha & \beta \\ \hline \beta & \alpha & \beta & \alpha & \beta & \alpha \\ \hline \end{array} \Rightarrow \boxed{Lu = 0}$$

This leads to difficulties when trying to implement L^{-1} in

$$\mathbb{P} = I - G L^{-1} D$$

(e.g. special multigrid is required)

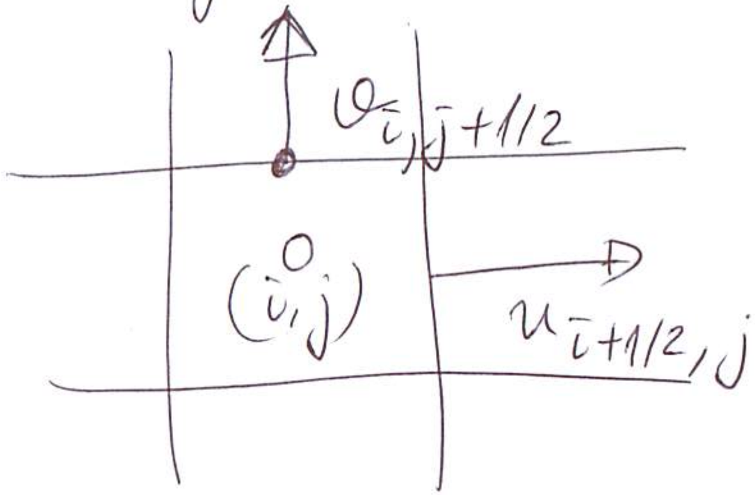
Note that this cell-centered grid also requires ghost-cell values to compute Dv near boundaries.



This is often expressed as: L^{-1} requires BCs (e.g. Neumann BCs)

Similar checkerboard modes exist even if Dv is node-centered (but that is better in general)

B) The staggered grid was designed to avoid these issues



$$(Dv)_{i,j} = \frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y}$$

$$G = -D^*$$

$$(G\pi)^x_{i+1/2,j} = \frac{\pi_{i+1,j} - \pi_{i-1,j}}{\Delta x}$$

cell centered

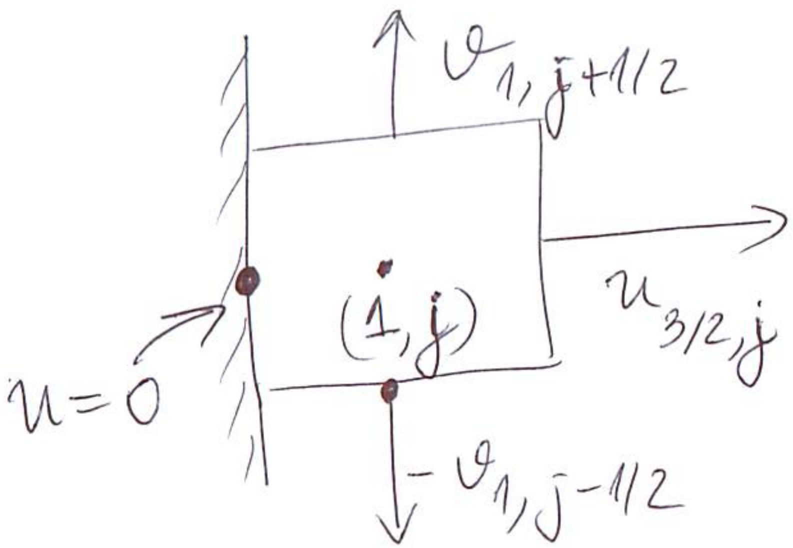
and similarly for y

Importantly, now $L = DG$ is the usual "compact" Laplacian:

$$(L\pi)_{i,j} = (\pi_{i+1,j} - 2\pi_{i,j} + \pi_{i-1,j})/(2\Delta x)^2 + (\pi_{i,j+1} - 2\pi_{i,j} + \pi_{i,j-1})/(2\Delta y)^2$$

which has only a trivial null space $\pi = \text{const.}$ so there is no checker board modes!

Furthermore, for no-slip BCs no ghost cells are needed: The BCs are handled naturally.



$$(Dv)_{1,j} = \frac{u_{3/2,j} - 0}{\Delta x} + \frac{v_{1,j+1/2} - v_{1,j-1/2}}{\Delta y}$$

One can also think of $L = DG$ coming with Neumann BCs but this is only useful in implementations as a trick. In reality no BCs for pressure are needed near no-slip boundaries. (i.e, impenetrable boundaries)

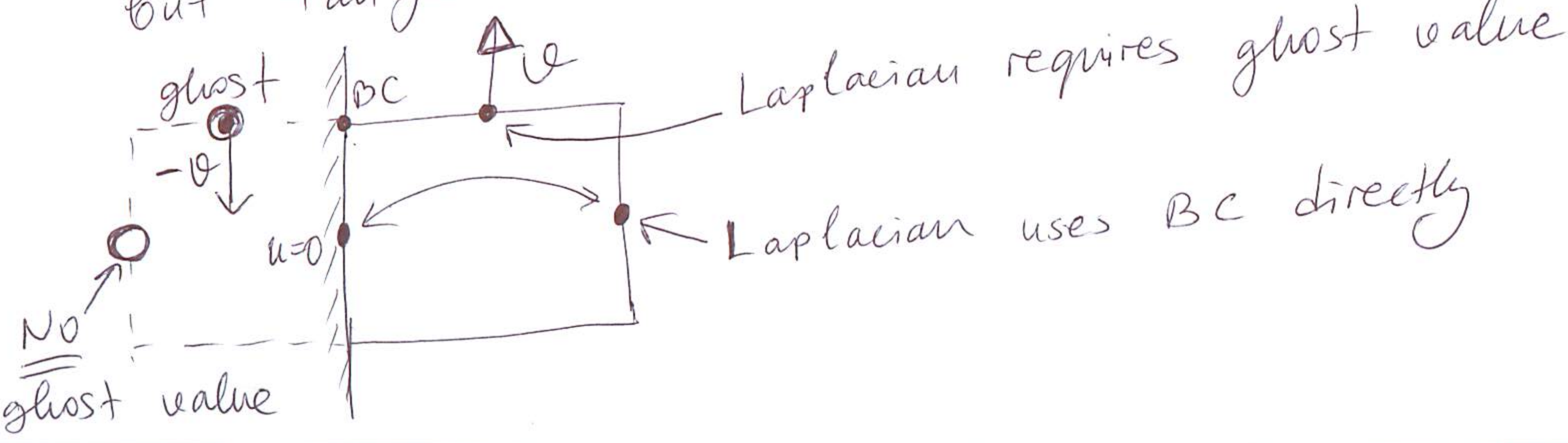
Recall

$$v_t = \mathbb{P}\left[-v\cdot\nabla v + \nu \nabla^2 v + f \right]$$

How to discretize this?

easy to discretize (including boundaries)

Note that near boundaries the normal component of velocity requires no special handling (no ghost cells), but tangential does:

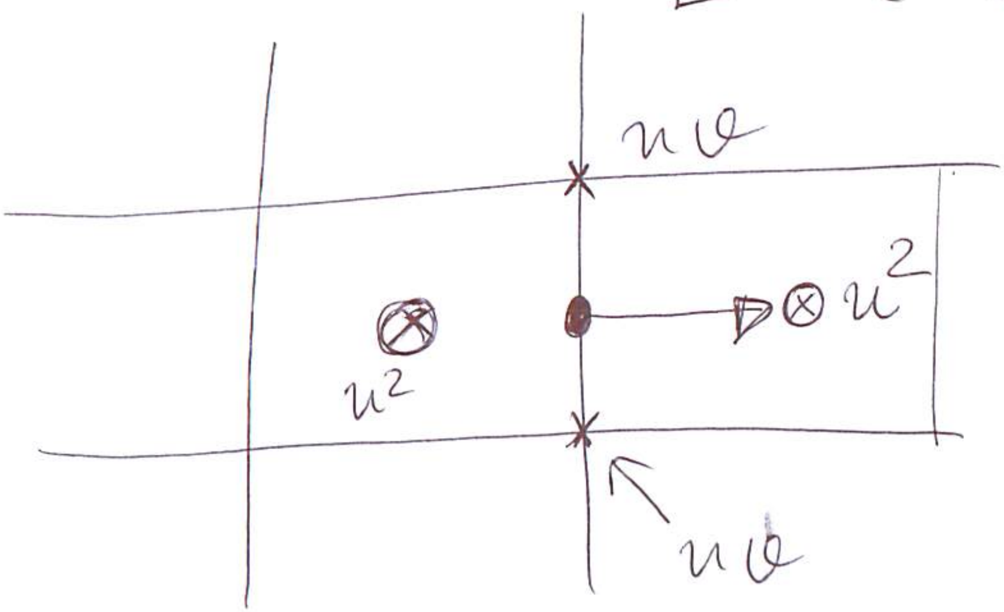


Advection

There are many ways to discretize advection, from limiters to simple centered differences.

In 2D: $-u\cdot\nabla u = \underbrace{-\nabla\cdot(u\otimes u^T)}_{\text{conservative}}$

$$\Rightarrow \quad -u\cdot\nabla u = \begin{bmatrix} -(u^2)_x - (uv)_y \\ -(uv)_x - (v^2)_y \end{bmatrix}$$



$$\left[(u^2)_x\right]_{i+1/2,j} = \frac{(u^2)_{i+1,j} - (u^2)_{i,j}}{\Delta x}$$

So for this to work we need to define u^2 and v^2 at cell centers and (uv) at nodes of the grid. (16)

We can do this by centered differences: by using simple averaging:

$$(u^2)_{i,j} = \left(\frac{u_{i+1/2,j} + u_{i-1/2,j}}{2}\right)^2$$

$$(uv)_{i+1/2,j+1/2} = \left(\frac{u_{i+1/2,j+1} + u_{i+1/2,j}}{2}\right)\left(\frac{v_{i+1,j+1/2} + v_{i,j+1/2}}{2}\right)$$

To do <u>upwinding</u>, one defines

$$\left[(u^2)\right]_{i,j} = \underbrace{\left(\frac{u_{i+1/2,j}+u_{i-1/2,j}}{2}\right)}_{\text{if } \bar{u}_{i,j} \geq 0} \cdot u_{i-1/2,j}$$

down-stream value

$$(uv)_{i+1/2,j+1/2} = \left\{ u_{i-1/2,j} \cdot \left(\frac{v_{i+1,j+1/2}+v_{i,j+1/2}}{2}\right) \right.$$

for $(uv)_y$ if $\bar{v} \geq 0$

upwind in direction of derivative

One can combine these sorts of expressions as, for example

$$(u^2)_{i,j} = \frac{u_{i+1/2,j} + u_{i-1/2,j}}{2} \cdot \begin{cases} \frac{1-\gamma}{2} u_{i+1/2,j} + \frac{1+\gamma}{2} u_{i-1/2,j} & \text{if } \bar{u}_{i,j} \geq 0 \\ \frac{1+\gamma}{2} u_{i+1/2,j} + \frac{1-\gamma}{2} u_{i-1/2,j} & \text{otherwise} \end{cases}$$

where $\gamma = 0$ gives centered (second order) differencing, and $\gamma = 1$ gives upwinding. (conservative).

These are just examples, there are alternative ways.

Aside: MAC grid can be seen as a special type of finite-element basis on rectangular elements. (13)

In order for saddle-point problem to be well-defined, we need Schur $= BA^{-1}B^T$ to be invertible uniformly in h (grid spacing)!

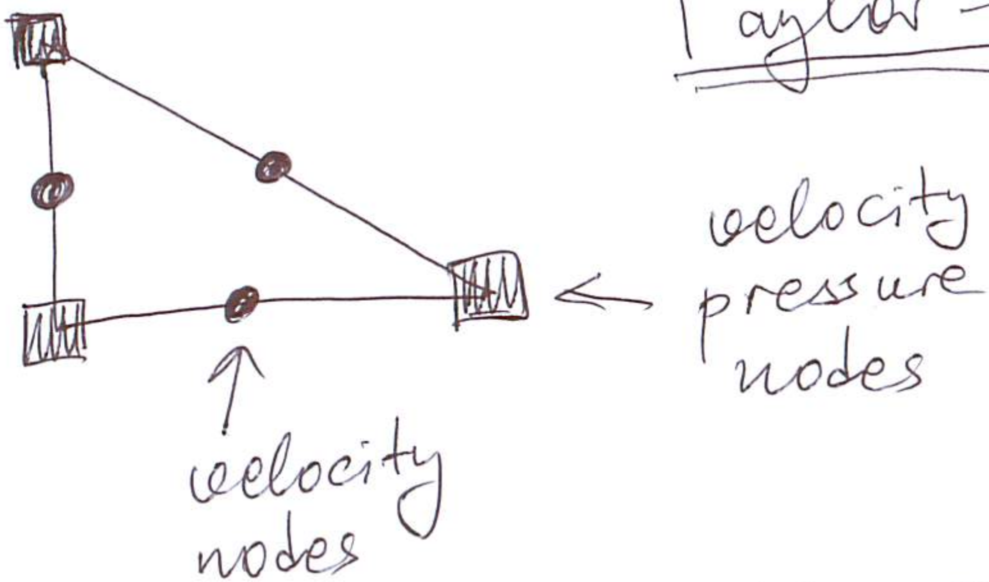
Precisely, we need the Babuška-Brezzi condition

$$\inf_{p \in P_h} \sup_{v \in V_h} \frac{(\nabla \cdot v, p)}{\|v\|_{V_h} \|p\|_{P_h}} \geq \gamma > 0$$

which is satisfied by MAC grid (14)

On triangular grids we have

Taylor-Hood element



6 velocity nodes

3 pressure nodes

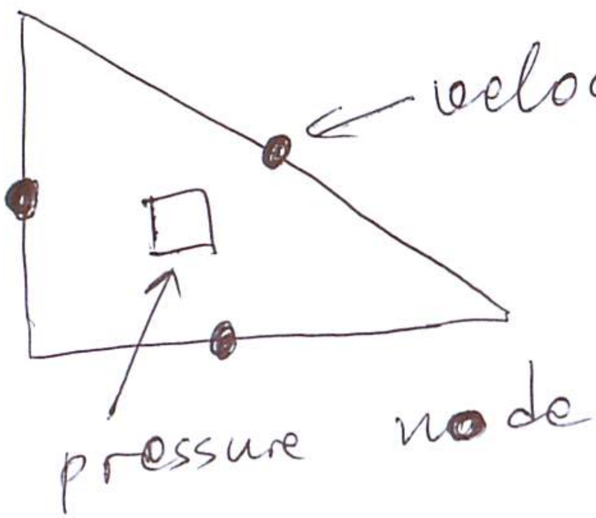
Each node is shared however

Quadratic in v:

$$v \to a + bx + cy + dxy + ex^2 + fy^2$$

Linear in P: $p \to \alpha + \beta x + \gamma y$

Or the lower-order
Crouzeix-Raviart element



3 velocity nodes
1 pressure node

Note: At nodes shared by elements the values must match → this ensures certain continuity of variables between adjacent elements.