POLITECNICO DI MILANO

Scuola di Ingegneria Industriale e dell'Informazione

Corso di Laurea Magistrale in Ingegneria Matematica

Formulazione mista per flussi in mezzi porosi fratturati: approssimazione con le Differenze Finite Mimetiche

Relatore
Prof. Luca Formaggia

Correlatore
Dr.ssa Anna Scotti

Candidata
Federica Sottocasa
Matricola 820073

Anno Accademico 2014-2015

Alla mia famiglia e a Claudio.

Sommario

Questo lavoro riguarda l'analisi di un modello che descrive il flusso in un mezzo poroso fratturato mediante le equazioni di Darcy in forma mista sia nel mezzo poroso che nelle fratture e le opportune condizioni di accoppiamento. Si suppone che le fratture abbiano spessore piccolo rispetto alla scala del mezzo e vengono infatti descritte mediante un modello ridotto. È stato definito un setting funzionale adatto ad introdurre la formulazione debole del problema accoppiato anche nel caso di un network di fratture e sono state provate esistenza e unicità della soluzione del problema modello supponendo di avere una sola frattura immersa nel dominio. Le Differenze Finite Mimetiche costituiscono la tecnica di discretizzazione utilizzata sia nel mezzo poroso che nelle fratture.
Oltre alla parte teorica è stato sviluppato un codice che si occupa del calcolo della pressione e della velocità del fluido nel mezzo e nelle fratture mediante la creazione della mesh, l'assemblaggio del sistema derivante dalla discretizzazione e la risoluzione di quest'ultimo dopo aver imposto le condizioni al bordo e di accoppiamento. Nei punti di intersezione vengono imposte condizioni di conservazione del flusso e continuità della pressione. Infine è stata effettuata un'analisi sugli errori e sugli ordini di convergenza di pressione nel mezzo e nelle fratture e di velocità nel mezzo.

Abstract

This work concerns the analysis of a model which describes the flow in a fractured porous media by the Darcy equations in mixed form both in the porous media and in the fractures, and the appropriate coupling conditions. These fractures are supposed to have a small thickness with respect to the media scale and indeed they are described by a reduced model. A suitable functional setting has been defined to introduce the weak formulation of the coupled problem also in the case of a fracture network, and existence and uniqueness of the solution to the model problem have been proved in the case of a single immersed fracture. Mimetic Finite Differences are the discretization technique both in the porous media and in the fractures.
In addition to the theoretical part a code has been developed to compute the pressure and the velocity of the fluid in the media and in the fractures by the creation of the mesh, the assembling of the system resulting from the discretization and its resolution after imposing boundary and coupling conditions. Conditions of flow conservation and pressure continuity are imposed at the intersection points. Finally, the analysis of pressure and velocity errors and of convergence orders has been carried out.

Indice

Introduzione

La simulazione di flussi sotterranei è di grande interesse in molti ambiti, ad esempio nella produzione di energia geotermica, nell'estrazione del petrolio, nella gestione di rifiuti nucleari e nei processi di bonifica. Spesso gli sforzi causati dai movimenti delle placche tettoniche o dalle attività umane producono fratture. La presenza di tali fratture può avere impatti differenti sul flusso: esse possono determinare una direzione preferenziale o formare barriere impermeabili, a seconda del tipo di materiale che le riempie.
Solitamente nella modellizzazione della frattura si sostituisce la regione d-dimensionale che occupa con un'interfaccia $(d-1)$-dimensionale, dando vita al cosiddetto *modello ridotto*. Tale modello viene proposto in [13] per una frattura che attraversa completamente il dominio e in [1] per una frattura immersa. Se non si utilizzasse il modello ridotto sarebbe necessaria una mesh molto fitta nelle fratture con un elevato costo computazionale. La legge di Darcy [3] è il modello matematico più utilizzato per la descrizione del flusso in un mezzo poroso ma anche in una frattura.
Inoltre tale modello si può presentare in forma mista o in forma primale, ad esempio in [1] si tratta il modello in forma primale sia nel mezzo che nella frattura, in [13] in forma mista in entrambi e in [2] in forma mista nel mezzo e primale nella frattura.

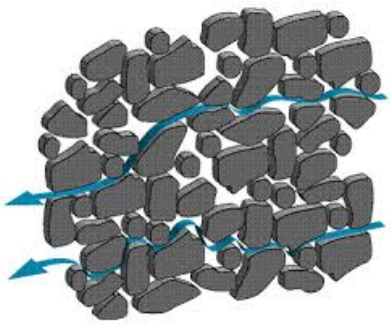


Figura 1: Un mezzo poroso.

In questa tesi è stato studiato il flusso di un fluido in un mezzo poroso in cui sono presenti delle fratture immerse. Il modello è stato descritto uti-

lizzando la legge di Darcy in forma mista sia nel mezzo che nelle fratture. Questo lavoro differisce da [13] poiché sono state considerate fratture immerse e da [1] poiché è stata usata la formulazione mista sia nel mezzo che nelle fratture.
Inoltre è stato interessante osservare cosa succede in presenza di un network di fratture intersecanti. A tal proposito, nel lavoro [9], sono state proposte alcune condizioni da imporre all'intersezione. In [5] l'autore ha considerato un network di fratture parzialmente immerso in cui almeno un estremo raggiunge il bordo del dominio, mentre in questa tesi il modello è stato esteso ad un network di fratture completamente immerse giungendo alla formulazione debole del problema.

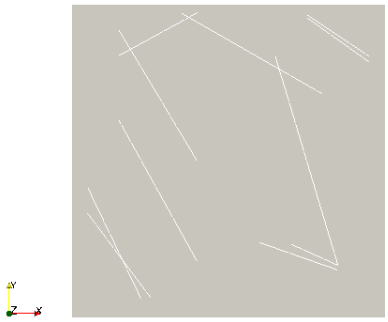


Figura 2: Un network di fratture.

La parte teorica innovativa della tesi riguarda la dimostrazione di buona posizione del problema di Darcy in forma mista sia nel mezzo che in un'unica frattura immersa nel dominio. Si tratta quindi di una dimostrazione più generale di quella affrontata in [13] che analizza solo il caso di una frattura che attraversa il dominio e che presenta difficoltà tecniche superiori a causa della maggiore rilevanza del termine di accoppiamento nel determinare la buona posizione del problema.

Per quanto riguarda le tecniche di discretizzazione, finora in letteratura sono state analizzate diverse possibilità, ad esempio in [8] si utilizzano gli XFEM, in [5] uno schema del tipo gradiente, in [1] i Volumi Finiti e in [2] le Differenze Finite Mimetiche per la discretizzazione nel mezzo e i Volumi Finiti nelle fratture. In questa tesi sarà utilizzata una discretizzazione con le Differenze Finite Mimetiche, non solo nel mezzo poroso, ma anche nelle fratture. L'utilizzo delle Differenze Finite Mimetiche è aumentato molto recentemente poiché sono estremamente flessibili e capaci di preservare le proprietà del modello, oltre ad avere ottime proprietà di convergenza su griglie poligonali molto generali.

La tesi ha anche una notevole componente implementativa, infatti si è proseguito il lavoro di N. Verzotti [18] introducendo la discretizzazione con le Mimetiche anche nelle fratture. Si è cercato di sfruttare al meglio quanto sviluppato precedentemente, di introdurre nuovi aspetti nel modo più coerente possibile e di uniformare il codice precedente al nuovo approccio "interamente mimetico". Il codice, utilizzato per calcolare la pressione e la velocità del fluido sia nel mezzo che nelle fratture, si occupa della costruzione della mesh, della definizione delle condizioni al bordo, dell'assemblaggio delle matrici e del termine noto del problema, della risoluzione del sistema, del calcolo degli errori e dell'esportazione della soluzione in un formato visualizzabile.
Il codice è stato esteso e generalizzato. È stato sviluppato utilizzando nuove tecniche del C++11 e librerie efficienti per la gestione delle matrici, quali le Eigen, e per la descrizione della geometria della mesh, CGAL in particolare.

La tesi presenta quindi una parte teorica ed una implementativa ed è strutturata come riportato in seguito.
Nel *capitolo 1* si effettua un'introduzione teorica sul problema di Darcy in forma mista sia nel mezzo poroso che nelle fratture.
Nel *capitolo 2* si fornisce il setting funzionale per la descrizione della formulazione debole del problema accoppiato sia nel caso di una sola frattura immersa che di un network di fratture immerse e si esegue la dimostrazione di esistenza e unicità della soluzione nel caso di una sola frattura immersa.
Nel *capitolo 3* si discute sull'applicazione delle Differenze Finite Mimetiche al problema, in particolare si analizza la discretizzazione e la formulazione algebrica del problema.
Nel *capitolo 4* si analizzano i dettagli implementativi e le novità introdotte nel codice.
Nel *capitolo 5* si discutono i risultati numerici ottenuti, mostrando casi che variano per il tipo di mesh, le condizioni di accoppiamento, il network di fratture, le permeabilità, la presenza di forzanti e il tipo di discretizzazione nelle fratture. In particolare si analizzano casi con fratture immerse ed intersecanti, si studia l'influenza di alcuni parametri fisici del modello sulla soluzione e si verifica l'ordine di convergenza del metodo.

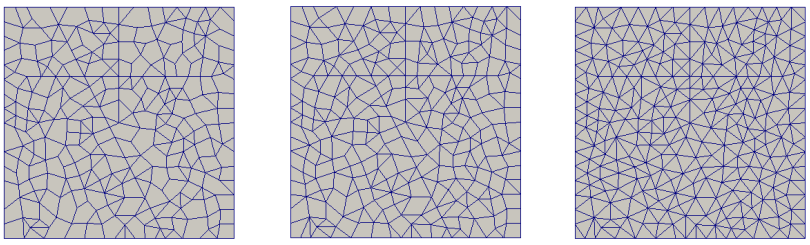


Figura 3: Esempi di mesh.

Capitolo 1

Le equazioni di Darcy

In questo capitolo verrà descritto il modello fisico e matematico delle equazioni di Darcy in forma mista, sia nel mezzo poroso sia nella frattura e l'accoppiamento dei due problemi. Sarà discusso il caso di fratture immerse che si intersecano indicando le condizioni che occorre imporre all'intersezione. I passi saranno quindi i seguenti:

- Modello nel mezzo poroso.
- Modello nella frattura.
- Accoppiamento dei due modelli.
- Generalizzazione al caso di un network di fratture.

1.1 Il modello fisico-matematico

Nel 1856 Darcy fece un esperimento sul flusso dell'acqua attraverso la sabbia e capì che il moto del fluido era dovuto alla pressione di quest'ultimo e alla gravità. Egli ambientò il suo esperimento in un particolare mezzo poroso, la sabbia, ma vi sono altri esempi di mezzi porosi come la ghiaia, l'argilla e soprattutto le rocce sedimentarie.
Quanto studiato da Darcy è un problema di interesse in molti rami della scienza, in particolare per lo studio del moto dei fluidi nel sottosuolo. Solitamente quando si deve studiare il moto di un fluido si ricorre alle equazioni di Navier-Stokes. In questo caso però, avendo a che fare con un mezzo poroso, si avrebbe la necessità di risolvere la scala dei pori, spesso inferiore di diversi ordini di grandezza alla grande scala, di interesse pratico. Si utilizza quindi l'equazione di Darcy che modellizza il fenomeno alla mesoscala, intermedia tra la grande scala del dominio e la piccola scala dei pori. Tale legge è stata ricavata sperimentalmente, come detto in precedenza, ma si può ottenere anche tramite un metodo di omogeneizzazione partendo dalle equazioni di

Navier-Stokes.

Una caratteristica importante di un mezzo poroso è la sua *porosità*, che rappresenta la misura dello spazio dei pori, calcolato come il rapporto tra il volume dei pori e il volume totale. Un altro parametro fondamentale è la *permeabilità*, che misura quanto facilmente il fluido si muova attraverso il mezzo. In questa tesi si considererà un fluido Newtoniano che occupa tutto lo spazio a disposizione, ossia quello dei pori e costituito da una sola fase. In generale lo spazio dei pori potrebbe essere riempito da uno o più liquidi e gas ma per semplicità si tratterà il caso di fluido monofase.

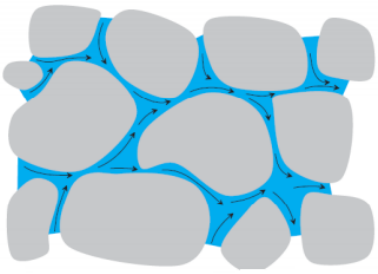


Figura 1.1: Un mezzo poroso.

Figura 1.2: Una roccia porosa.

Figura 1.3: Un mezzo poroso.

Sia Ω un dominio aperto e limitato, allora la *Legge di Darcy* afferma che:

$$\mathbf{u} = -\frac{\mathbf{K}}{\mu}(\nabla p - \rho\,\mathbf{g}) \qquad \text{in } \Omega, \tag{1.1}$$

dove:

- $\mathbf{u}$ rappresenta la velocità di Darcy del fluido, da interpretare come una media della velocità microscopica nei pori,
- $\mathbf{K}$ è il tensore di permeabilità del mezzo poroso,
- μ è la viscosità del fluido,
- p indica la pressione del fluido,

- $\mathbf{g}$ è l'accelerazione di gravità, più precisamente $\mathbf{g} = g\,\mathbf{e}_z$ dove $\mathbf{e}_z$ rappresenta il versore orientato lungo la verticale. Nel sistema di riferimento adottato z è infatti la coordinata verticale crescente con la profondità.

Se le variazioni di quota sono trascurabili rispetto al termine di pressione è possibile evitare di considerare il termine dovuto all'accelerazione di gravità nell'equazione (1.1). Se la densità è costante trascurare il termine di gravità non introduce una perdita di generalità poiché la (1.1) si può riscrivere:

$$\mathbf{u} = -\frac{\mathbf{K}}{\mu}\nabla\psi \qquad \text{con} \quad \psi = p - \rho\, g\, z \qquad \text{in } \Omega.$$

In seguito il tensore $\mathbf{K}$ terrà conto oltre che della permeabilità del mezzo anche della viscosità e della densità del fluido; sarà chiamato ancora permeabilità anche se più propriamente si tratterà di una conducibilità.
Si ipotizza che il tensore $\mathbf{K}$ sia simmetrico e uniformemente ellittico, cioè che esistano due costanti positive tali che:

$$c_1 \|\mathbf{u}\|^2 \leq \mathbf{u}^T \mathbf{K}\, \mathbf{u} \leq c_2 \|\mathbf{u}\|^2 \qquad \text{q.o. in } \Omega.$$

Accanto alla legge di Darcy per descrivere il moto di un fluido in un mezzo poroso occorre introdurre la *Legge di Conservazione della Massa* che, nella sua forma più generale, si presenta nel seguente modo:

$$\frac{\partial(\phi\,\rho)}{\partial t} + \operatorname{div}(\rho\,\mathbf{u}) = \rho\, f \qquad \text{in } \Omega \tag{1.2}$$

dove:

- ϕ rappresenta la porosità del mezzo,
- ρ la densità volumetrica del fluido,
- f una forzante esterna per unità di volume, per esempio legata a iniezione o estrazione del fluido.

Se si ipotizza che il fluido sia a densità costante e a porosità che non varia nel tempo la (1.2) si semplifica notevolmente, diventando:

$$\operatorname{div}\mathbf{u} = f \qquad \text{in } \Omega. \tag{1.3}$$

Nel caso più generale il modello può essere accoppiato con le equazioni della poroelasticità che descrivano la variazione di porosità del mezzo, particolarmente complicata soprattutto se il mezzo non è omogeneo.

Le equazioni da risolvere nel mezzo poroso sono quindi:

$$\begin{cases} \operatorname{div} \mathbf{u} = f & \text{in } \Omega \\ \mathbf{u} + \mathbf{K}\nabla p = 0 & \text{in } \Omega \\ p = g_P & \text{su } \partial\Omega_P \\ \mathbf{u} \cdot \mathbf{n} = g_{\mathbf{u}} & \text{su } \partial\Omega_{\mathbf{u}} \end{cases} \tag{1.4}$$

dove sono state introdotte delle opportune condizioni al bordo, suddiviso in $\partial\Omega_P$ su cui sono imposte condizioni sulla pressione e in $\partial\Omega_{\mathbf{u}}$ con condizioni sulla velocità.
Il modello riportato in (1.4) si presenta in forma mista, un'alternativa sarebbe l'analisi del modello in forma primale, descritto in [1].

In questa tesi si tratterà un mezzo poroso con fratture immerse ed intersecanti. Per descrivere il problema nelle fratture si adotterà il modello di Darcy in forma mista analogo al precedente. Occorre dapprima introdurre la notazione utilizzata e le approssimazioni adottate per descrivere le fratture. Per ora si suppone di avere una sola frattura.
L'uso del modello (1.4) in presenza di fratture è problematico. Infatti le fratture hanno una dimensione, lo spessore, molto più piccola rispetto al diametro di Ω. Inoltre la permeabilità nelle fratture può variare di diversi ordini di grandezza rispetto a quella del mezzo. Quindi utilizzare direttamente la (1.4) richiederebbe una mesh estremamente fine nelle zone con fratture a costi computazionali eccessivamente elevati. Se lo spessore della frattura è trascurabile rispetto alla scala del mezzo si adotta un *Modello Ridotto*, in cui la frattura è una varietà di codimensione uno. Nei casi analizzati si avrà sempre a che fare con domini bidimensionali in cui la frattura sarà considerata come un'interfaccia unidimensionale. In generale la frattura potrebbe essere una qualsiasi linea regolare, per semplicità si assume che sia un segmento di retta.

La notazione introdotta si riferisce alla figura (1.4):

- Con Γ si indica la frattura. Questa può attraversare completamente il dominio o essere parzialmente o completamente immersa. Negli ultimi due casi, più interessanti dal punto di vista dell'analisi, si assumerà che Γ possa essere estesa fino al bordo partizionando Ω in due sottodomini, indicati con Ω^+ e Ω^- e Lipschtz. La frattura estesa verrà indicata con $\widetilde{\Gamma}$. In questo lavoro si considererà il caso di una frattura parzialmente o completamente immersa, in quanto il caso di frattura che attraversa il dominio è già stato trattato in [8], [13] e [18]. La normale alla frattura $\mathbf{n}_\Gamma$ esce dal sotto-dominio Ω^+ e coincide con $\mathbf{n}^+$.

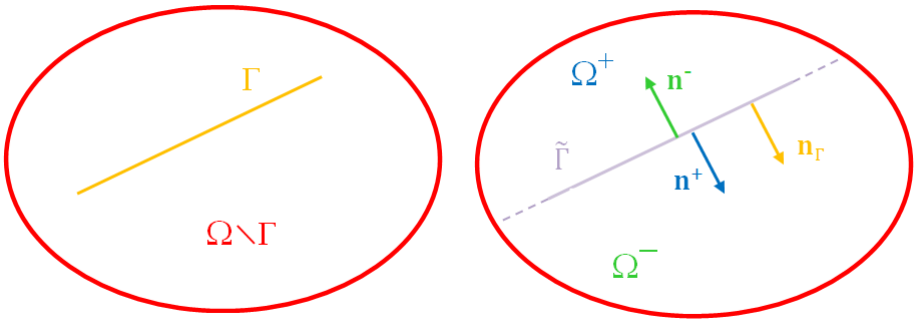


Figura 1.4: Notazione.

- $[\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]$ e $\{\mathbf{u}\cdot\mathbf{n}_\Gamma\}$ indicano rispettivamente il salto e la media della velocità $\mathbf{u}$ attraverso la frattura e sono definiti come segue:

$$[\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!] = \mathbf{u}^+\cdot\mathbf{n}_\Gamma - \mathbf{u}^-\cdot\mathbf{n}_\Gamma \tag{1.5}$$

$$\{\mathbf{u}\cdot\mathbf{n}_\Gamma\} = \frac{\mathbf{u}^+\cdot\mathbf{n}_\Gamma + \mathbf{u}^-\cdot\mathbf{n}_\Gamma}{2}. \tag{1.6}$$

- $\hat{p}$ e $\hat{\mathbf{u}}$ sono rispettivamente la pressione e il flusso del fluido nella frattura e considerando l_Γ la larghezza e $\boldsymbol{\tau}$ la direzione tangente alla frattura, si ha:

$$\hat{\mathbf{u}} = \int_{-\frac{l_\Gamma}{2}}^{\frac{l_\Gamma}{2}} \hat{\mathbf{v}}\cdot\boldsymbol{\tau}$$

  dove $\hat{\mathbf{v}}$ indica la velocità nella frattura.
  Allo stesso modo si definisce:

$$\hat{p} = \frac{1}{l_\Gamma}\int_{-\frac{l_\Gamma}{2}}^{\frac{l_\Gamma}{2}} p_f$$

  dove $\hat{p}$ è data dalla pressione nella frattura p_f integrata lungo la direzione normale alla frattura.

- Il tensore di permeabilità $\hat{\mathbf{K}}$ della frattura ha, per ipotesi, solo due componenti non nulle in un sistema di coordinate localmente allineato con la frattura, infatti per la diversa conformazione della matrice solida e della frattura, in quest'ultima il fluido può scorrere in direzione normale o tangenziale. Si suppone dunque che nel sistema di coordinate allineate alla frattura il tensore di permeabilità abbia la seguente forma:

$$\hat{\mathbf{K}} = \begin{bmatrix} \hat{K}_{\mathbf{n}} & 0 \\ 0 & \hat{K}_{\boldsymbol{\tau}} \end{bmatrix}.$$

Nel sistema di riferimento (x, z) la permeabilità nella frattura è descritta da $\hat{K}_{\mathbf{n}}\,\mathbf{n}_\Gamma \otimes \mathbf{n}_\Gamma + \hat{K}_{\boldsymbol{\tau}}\,(I - \mathbf{n}_\Gamma \otimes \mathbf{n}_\Gamma)$.

- Si introducono, per semplificare la notazione, le seguenti due quantità:

$$\eta = \frac{l_\Gamma}{\hat{K}_{\mathbf{n}}} \qquad \text{e} \qquad \hat{K} = l_\Gamma\,\hat{K}_{\boldsymbol{\tau}}.$$

- Si indica con Ω_Γ il dominio $\Omega \backslash \Gamma$, in cui si esclude la frattura dal mezzo poroso.

Dopo tutte queste precisazioni il modello in forma mista per il problema di Darcy nella frattura è il seguente:

$$\begin{cases} \mathrm{div}_{\boldsymbol{\tau}}\,\hat{\mathbf{u}} = \hat{f} + [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!] & \text{in } \Gamma \\ \hat{K}^{-1}\,\hat{\mathbf{u}} + \nabla_{\boldsymbol{\tau}}\,\hat{p} = 0 & \text{in } \Gamma \\ \hat{\mathbf{u}}\cdot\boldsymbol{\tau} = \hat{g}_{\mathbf{u}} & \text{su } \partial\Gamma_{\hat{\mathbf{u}}} \\ \hat{p} = \hat{g}_P & \text{su } \partial\Gamma_{\hat{p}} \end{cases} \tag{1.7}$$

dove:

- $\mathrm{div}_{\boldsymbol{\tau}}\,\mathbf{v} = \mathrm{div}\,\mathbf{v} - \nabla(\mathbf{v}\cdot\mathbf{n}_\Gamma)\cdot\mathbf{n}_\Gamma$ e $\nabla_{\boldsymbol{\tau}}\,q = \nabla q - (\nabla q\cdot\mathbf{n}_\Gamma)\,\mathbf{n}_\Gamma$.
- $\hat{f}$ indica un termine forzante nella frattura ed è pari a $l_\Gamma\,h$ se h è il termine sorgente per unità di volume nella frattura.
- $[\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]$ è un termine sorgente addizionale che tiene conto del contributo del flusso tra i sotto-domini e la frattura.

Tale modello è stato dettagliatamente ricavato in [13]. La prima equazione rappresenta la conservazione della massa all'interno della frattura e la seconda la legge di Darcy in direzione tangenziale alla frattura. Sono state imposte due condizioni generali sul bordo della frattura; solitamente nel caso di frattura immersa vengono applicate condizioni sulla velocità omogenee, poiché la larghezza l_Γ della frattura è piccola e di conseguenza il trasferimento di massa attraverso le estremità può essere trascurato rispetto a quello trasversale. Nel caso in cui un estremo della frattura raggiunga il bordo potrebbe capitare di voler imporre una condizione sulla pressione.

Si noti che, nel caso di un mezzo poroso fratturato, nella matrice solida vale ancora la legge di Darcy (1.4), ambientata però in Ω_Γ. A questo punto occorre accoppiare il problema nel mezzo e quello nella frattura, aggiungendo altre due equazioni che legano i due modelli nel seguente modo, come riportato in [13]:

$$\begin{cases} \eta\,\{\mathbf{u}\cdot\mathbf{n}_\Gamma\} = [\![p]\!] & \text{su } \Gamma \\ \eta\,\dfrac{2\xi-1}{4}\,[\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!] = \{p\} - \hat{p} & \text{su } \Gamma. \end{cases} \tag{1.8}$$

ξ è un parametro che teoricamente può variare nell'intervallo $[0, 1]$, ma la buona posizione del problema si otterrà per ξ maggiore di un certo valore dipendente da alcuni parametri fisici della frattura e del mezzo. Il valore ottimale di ξ è 0.75 che corrisponde ad assumere una variazione parabolica della pressione trasversalmente alla frattura.

Quindi il modello da discretizzare è il seguente:

$$\begin{cases} \operatorname{div} \mathbf{u} = f & \text{in } \Omega_\Gamma \\ \mathbf{K}^{-1}\mathbf{u} + \nabla p = 0 & \text{in } \Omega_\Gamma \\ p = g_P & \text{su } \partial\Omega_P \\ \mathbf{u} \cdot \mathbf{n} = g_{\mathbf{u}} & \text{su } \partial\Omega_{\mathbf{u}} \\ \operatorname{div}_{\boldsymbol{\tau}} \hat{\mathbf{u}} = \hat{f} + [\![\mathbf{u} \cdot \mathbf{n}_\Gamma]\!] & \text{in } \Gamma \\ \hat{K}^{-1}\,\hat{\mathbf{u}} + \nabla_{\boldsymbol{\tau}} \hat{p} = 0 & \text{in } \Gamma \\ \hat{\mathbf{u}} \cdot \boldsymbol{\tau} = \hat{g}_{\mathbf{u}} & \text{su } \partial\Gamma_{\hat{\mathbf{u}}} \\ \hat{p} = \hat{g}_P & \text{su } \partial\Gamma_{\hat{p}} \\ \eta\,\{\mathbf{u} \cdot \mathbf{n}_\Gamma\} = [\![p]\!] & \text{su } \Gamma \\ \eta\,\dfrac{2\xi - 1}{4}\,[\![\mathbf{u} \cdot \mathbf{n}_\Gamma]\!] = \{p\} - \hat{p} & \text{su } \Gamma. \end{cases} \tag{1.9}$$

Per la presenza della forma mista sia nel mezzo poroso che nella frattura, ci sono quattro incognite, due scalari e due vettoriali, la pressione nel mezzo p, la velocità nel mezzo $\mathbf{u}$, la pressione nella frattura $\hat{p}$ e il flusso nella frattura $\hat{\mathbf{u}}$.

Sarebbe stato possibile utilizzare anche altri tipi di formulazioni, ad esempio in [1] si usa la formulazione primale del problema di Darcy sia nel mezzo che nella frattura, in [2] si utilizza la formulazione mista nel mezzo e primale nella frattura. In questa tesi si è scelto di utilizzare la formulazione mista sia nel mezzo che nella frattura poiché fornisce una miglior rappresentazione del flusso. Per la derivazione del modello si rimanda a [13].

1.2 Un network di fratture immerse

In questa sezione si riporteranno con particolare dettaglio la notazione e le convenzioni adottate per gestire un network di fratture. Più precisamente verrà considerato un network di fratture immerse, ossia un network di fratture i cui estremi non intersecano i bordi del dominio; se almeno un estremo raggiunge il bordo si parla di network di fratture parzialmente immerso.

Si indica con Γ il network composto da un insieme di M_Γ fratture:

$$\Gamma = \bigcup_{k=1}^{M_\Gamma} \overline{\gamma}_k$$

dove $\overline{\gamma}_k$ rappresenta la $k-$esima frattura ed è un segmento aperto in $\mathbb{R}^2$.

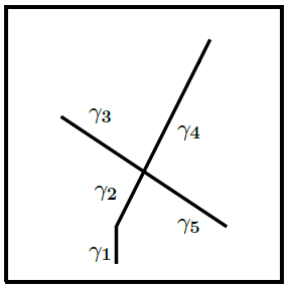


Figura 1.5: Un network di fratture immerse in un dominio bidimensionale.

Come si osserva nella figura (1.5) vale la seguente assunzione:

$$\forall\, j \neq k \quad \overline{\gamma}_k \cap \overline{\gamma}_j \,=\, \partial\gamma_k \cap \partial\gamma_j = i_{kj}, \tag{1.10}$$

che significa che le fratture possono incontrarsi solo ai loro estremi.
Dalla (1.10) si osserva che i_{kj} indica il punto di intersezione tra la frattura k e la frattura j, che può essere vuoto se le due non si intersecano.
Si assume che l'angolo formato da una coppia di fratture intersecanti sia limitato dal basso con un angolo positivo. Non si esclude la possibilità che le fratture possano raggiungere il bordo, è quindi opportuno ipotizzare anche che l'angolo formato tra la frattura e il bordo del dominio sia limitato come nel caso precedente. Una conseguenza di queste ipotesi è che il numero di fratture che si intersecano in un stesso punto è limitato.
Si indica con I l'insieme dei punti di intersezione presenti nel network, quindi $I = \cup\, i_{kj}$ e si identificano i seguenti insiemi:

$$\partial\gamma_k^P \,=\, \partial\gamma_k \cap \partial\Omega_P \qquad \partial\gamma_k^{\mathbf{u}} \,=\, \partial\gamma_k \cap \partial\Omega_{\mathbf{u}}$$

$$\partial\gamma_k^I \,=\, \partial\gamma_k \cap I \qquad \partial\gamma_k^F \,=\, \partial\gamma_k \setminus \left(\bigcup_{s=P,\mathbf{u},I} \partial\gamma_k^s \right),$$

dove $\partial\gamma_k^F$ raccoglie gli estremi della frattura γ_k che sono completamente immersi nel dominio e che non sono punti di intersezione.
Chiaramente alcuni di questi insiemi possono essere vuoti e l'unione di tutti costituisce $\partial\gamma_k$. Per $s = P, \mathbf{u}, F$ si definisce $I^s = \bigcup_{k=1}^{M_\Gamma} \partial\gamma_k^s$. Dato un punto di intersezione $i \in I$ si indica con S_i l'insieme delle fratture γ_k che si intersecano in i cioè quelle fratture per cui $\partial\gamma_k^I \cap i \neq \emptyset$.

Su ciascuna frattura vengono identificate due parti γ_k^+ e γ_k^- e le due normali associate $\mathbf{n}_k^+$ e $\mathbf{n}_k^- = -\mathbf{n}_k^+$. Inoltre si identifica la normale della frattura $\mathbf{n}_k = \mathbf{n}_k^+$. Per semplificare la notazione si indicano con $\mathbf{n}_\Gamma^\pm$ i vettori normali al network di fratture, per cui $\mathbf{n}_\Gamma^\pm = \mathbf{n}_k^\pm(\mathbf{x})$ se $\mathbf{x} \in \gamma_k$. Analogamente si definisce $\mathbf{n}_\Gamma = \mathbf{n}_\Gamma^+$.
Data una funzione f appartenente a Ω_Γ si indica con $f^\pm$ la sua traccia su $\Gamma^\pm = \bigcup_{k=1}^{M_\Gamma} \gamma_k^\pm$. Ciò consente di estendere gli operatori di media e salto sul network nel seguente modo:

$$\{f\} = \frac{1}{2}(f^+ + f^-) \qquad\qquad [\![f]\!] = f^+ - f^-.$$

Si estendono le definizioni di $\hat{\mathbf{K}}$, $\hat{K}_{\mathbf{n}}$ e $\hat{K}_{\boldsymbol{\tau}}$ in maniera naturale. Infine si indica con $\boldsymbol{\tau}_k$ la tangente unitaria a γ_k.

È quindi possibile estendere formalmente il problema (1.9) ad un network di fratture immerse. Occorre però imporre delle condizioni nei punti di intersezione, in particolare saranno utilizzate la continuità della pressione e la conservazione del flusso. Altre possibili condizioni da applicare all'intersezione sono state studiate in [9].
Il problema da risolvere assume quindi la seguente forma:

$$\begin{cases}
\operatorname{div} \mathbf{u} = f & \text{in } \Omega_\Gamma \\
\mathbf{K}^{-1}\mathbf{u} + \nabla p = 0 & \text{in } \Omega_\Gamma \\
p = g_P & \text{su } \partial\Omega_P \\
\mathbf{u} \cdot \mathbf{n} = g_{\mathbf{u}} & \text{su } \partial\Omega_{\mathbf{u}} \\
\operatorname{div}_{\boldsymbol{\tau}} \hat{\mathbf{u}} = \hat{f} + [\![\mathbf{u} \cdot \mathbf{n}_\Gamma]\!] & \text{in } \Gamma \\
\hat{K}^{-1}\,\hat{\mathbf{u}} + \nabla_{\boldsymbol{\tau}}\,\hat{p} = 0 & \text{in } \Gamma \\
\hat{\mathbf{u}} \cdot \boldsymbol{\tau} = \hat{g}_{\mathbf{u}} & \text{su } I^{\mathbf{u}} \\
\hat{\mathbf{u}} \cdot \boldsymbol{\tau} = 0 & \text{su } I^F \\
\hat{p} = \hat{g}_P & \text{su } I^P \\
\eta\,\{\mathbf{u} \cdot \mathbf{n}_\Gamma\} = [\![p]\!] & \text{su } \Gamma \\
\eta\,\dfrac{2\xi - 1}{4}\,[\![\mathbf{u} \cdot \mathbf{n}_\Gamma]\!] = \{p\} - \hat{p} & \text{su } \Gamma \\
\hat{p}_k = p_i & \text{in } i \quad \forall\, \gamma_k \in S_i, \quad \forall\, i \in I \\
\displaystyle\sum_{k:\,\gamma_k \in S_i} \hat{\mathbf{u}}_k \cdot \boldsymbol{\tau}_k|_i = 0 & \text{in } i \quad \forall\, i \in I
\end{cases} \tag{1.11}$$

dove con $\hat{p}_k$ e $\hat{\mathbf{u}}_k$ si indicano rispettivamente la pressione e il flusso nella frattura γ_k e con p_i la pressione nel punto di intersezione $i \in I$.

Si è ipotizzato, per l'analisi teorica, che sugli estremi immersi delle fratture ci siano condizioni sulla velocità omogenee.

Nel successivo capitolo si fornirà la formulazione debole di tale problema. Si entrerà maggiormente nel dettaglio per il caso di una sola frattura immersa ma si forniranno degli spunti anche per il network di fratture. In particolare sarà interessante osservare come il problema nel network si possa ricondurre ad un problema in forma debole del tutto analogo a quello per la singola frattura.

Capitolo 2

Buona posizione del problema

In questo capitolo verrà effettuata la dimostrazione di buona posizione della formulazione debole del problema (1.9) descritto nel capitolo precedente. Sarà opportuno introdurre un adeguato setting funzionale per definire correttamente la formulazione debole e in seguito si procederà con la dimostrazione di esistenza e unicità della soluzione. La dimostrazione rappresenta uno degli aspetti più innovativi della tesi, in particolare per la parte legata alla validità della condizione inf-sup. Nell'articolo [13] è stata analizzata la buona posizione del problema in forma mista, sia nella frattura che nel mezzo poroso, ma è stato trattato solamente un dominio con una frattura che lo attraversa completamente. In questo capitolo si considererà invece il caso di una frattura immersa nel dominio. Già in [1] è stato descritto il setting funzionale per questa configurazione, ma utilizzando una formulazione primale sia nel mezzo che nella frattura.

2.1 Gli operatori di traccia

Prima di procedere con la formulazione debole del problema (1.9) occorre precisare nel dettaglio il setting funzionale. Infatti si ha a che fare con una frattura immersa nel dominio ed è opportuno introdurre spazi e operatori adeguati, analogamente a quanto fatto nell'articolo [1].
In generale si potrebbero avere fratture che attraversano il dominio, fratture parzialmente immerse, con un estremo che raggiunge il bordo del dominio e l'altro immerso o fratture completamente immerse. Il modello che verrà presentato è generale ed è valido per ciascuno dei tre tipi di frattura. Nell'analisi condotta in questo capitolo ci si concentrerà però sul caso di frattura completamente immersa, essendo quello più critico dei tre possibili. Inoltre il caso di una frattura che attraversa il dominio è già stato ampiamente trattato, in particolare in [13] e quello di una frattura parzialmente immersa è semplicemente un problema intermedio tra gli altri due.

Figura 2.1: In ordine una frattura immersa, una parzialmente immersa e una che attraversa il dominio.

Si indicheranno con i seguenti simboli le norme utilizzate nel capitolo:

- $\|\cdot\|_{L^2(\Omega_\Gamma)}$ la norma L^2 su Ω_Γ,
- $\|\cdot\|_{H^1(\Omega_\Gamma)}$ la norma H^1 su Ω_Γ,
- $\|\cdot\|_{H^{\frac{1}{2}}(U)}$ la norma dello spazio $H^{\frac{1}{2}}(U)$ dove $U \in \partial\Omega$.

Innanzitutto verranno forniti dei dettagli a proposito degli operatori di traccia. Gli operatori di traccia da $H^1(\Omega_\Gamma)$ a $H^{\frac{1}{2}}(\partial\Omega_P)$ o $H^{\frac{1}{2}}(\partial\Omega_{\mathbf{u}})$ sono definiti in modo classico come se il dominio fosse continuo, essendo la frattura immersa. Occorre concentrarsi maggiormente sulla definizione della traccia sulla frattura. Si riportano, a tal proposito, i risultati ottenuti in [1].
Innanzitutto si definiscono due operatori di traccia lineari e continui γ^+ e γ^- da $H^1(\Omega_\Gamma)$ a $H^{\frac{1}{2}}(\Gamma)$, restringendo a Γ gli operatori standard di traccia su $\widetilde{\Gamma}^+$ e $\widetilde{\Gamma}^-$ definiti per i due sottodomini Ω^+ e Ω^- introdotti nel capitolo precedente. Questa definizione non dipende dal modo in cui i due sottodomini sono costruiti estendendo la frattura.

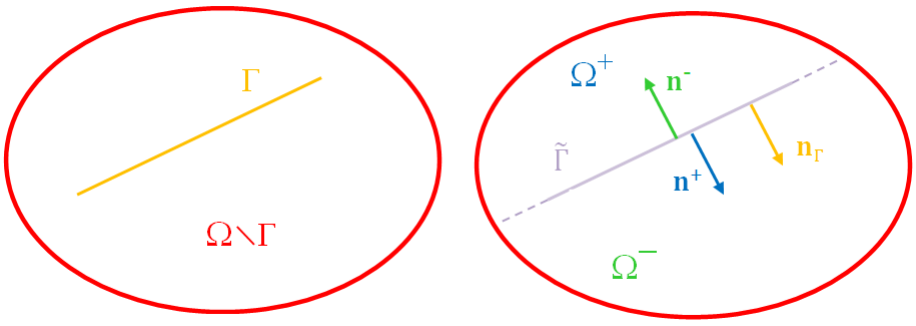


Figura 2.2: Notazione.

Si introduce una coordinata sulla frattura Γ per cui i suoi punti sono descritti

da $\mathbf{x} = \mathbf{x}(s),\ s \in \left[0,\ |\Gamma|\right]$ e si definisce lo spazio seguente:

$$V_\Gamma = \left\{ g \in H^{\frac{1}{2}}(\Gamma)\ :\ \frac{|g(s)|^2}{s\left(|\Gamma| - s\right)} \in L^1(\Gamma) \right\}$$

dove $s \in \Gamma$.
Tale spazio è spesso indicato con $H^{\frac{1}{2}}_{00}(\Gamma)$ ed è dotato della seguente norma:

$$\|g\|^2_{V_\Gamma} = \|g\|^2_{H^{\frac{1}{2}}(\Gamma)} + \int_0^{|\Gamma|} \frac{|g(s)|^2}{s\left(|\Gamma| - s\right)}\, ds.$$

Si indica con V'_Γ il suo duale, mentre con $H^{-\frac{1}{2}}(\Gamma)$ si indica il duale di $H^{\frac{1}{2}}(\Gamma)$.

Proposizione 2.1.1. *L'operatore di traccia γ_Γ su Γ:*

$$\gamma_\Gamma : p \in H^1(\Omega_\Gamma) \to \gamma_\Gamma(p) = (\gamma^+(p), \gamma^-(p)) \in H^{\frac{1}{2}}(\Gamma) \times H^{\frac{1}{2}}(\Gamma),$$

è continuo da $H^1(\Omega_\Gamma)$ sullo spazio:

$$\begin{aligned} T_\Gamma &= \left\{ \mathbf{g} = (g^+, g^-) \in \left(H^{\frac{1}{2}}(\Gamma)\right)^2,\ \int_0^{|\Gamma|} \frac{[\![\mathbf{g}]\!]^2}{s\left(|\Gamma| - s\right)}\, ds < +\infty \right\} \\ &= \left\{ \mathbf{g} \in \left(H^{\frac{1}{2}}(\Gamma)\right)^2,\ [\![\mathbf{g}]\!] \in V_\Gamma \right\}, \end{aligned}$$

dotato della norma:

$$\|\mathbf{g}\|_{T_\Gamma} = \left(\|g^+\|^2_{H^{\frac{1}{2}}(\Gamma)} + \|g^-\|^2_{H^{\frac{1}{2}}(\Gamma)} + \int_0^{|\Gamma|} \frac{[\![\mathbf{g}]\!]^2}{s\left(|\Gamma| - s\right)}\, ds \right)^{\frac{1}{2}}.$$

Inoltre esiste un operatore di rilevamento lineare continuo

$$R_\Gamma : T_\Gamma \to H^1_P(\Omega_\Gamma)$$

che è l'inverso destro dell'operatore di traccia, cioè:

$$\gamma_\Gamma \circ R_\Gamma = Id_{T_\Gamma}$$

dove $H^1_P(\Omega_\Gamma) = \left\{ p \in H^1(\Omega_\Gamma),\ p = 0 \text{ su } \partial\Omega \right\}$.
In particolare:

$$\exists\, C_\Gamma > 0 :\ \|R_\Gamma(\mathbf{g})\|_{H^1_P(\Omega_\Gamma)} \leq C_\Gamma \|\mathbf{g}\|_{T_\Gamma}.$$

Infine C^∞_c è denso in $\ker \gamma_\Gamma \cap H^1_P(\Omega_\Gamma)$.

Con tali risultati, riportati in [1], è possibile definire la traccia normale su Γ per ogni vettore $\mathbf{v} \in H_{\text{div}}(\Omega_\Gamma)$ come un elemento dello spazio duale $T_\Gamma{}'$, segue quindi la seguente proposizione:

Proposizione 2.1.2. *Per ogni* $\mathbf{v} \in H_{\mathrm{div}}(\Omega_\Gamma)$*, la mappa* $\mathbf{v}\cdot\mathbf{n} \in T_\Gamma{}'$ *è definita come:*

$$\mathbf{g} = (g^+, g^-) \in T_\Gamma \to \langle \mathbf{v}\cdot\mathbf{n},\ \mathbf{g}\rangle_{T_\Gamma{}',T_\Gamma}$$

dove

$$\langle \mathbf{v}\cdot\mathbf{n},\ \mathbf{g}\rangle_{T_\Gamma{}',T_\Gamma} = \int_{\Omega_\Gamma} \mathbf{v}\cdot\nabla R_\Gamma(\mathbf{g})\,dx + \int_{\Omega_\Gamma} \mathrm{div}\,\mathbf{v}\, R_\Gamma(\mathbf{g})\,dx.$$

Tale mappa è lineare e continua e non dipende dalla scelta di R_Γ*. Inoltre esiste un unico elemento* $[\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!]$ *in* $H^{-\frac{1}{2}}(\Gamma)$ *e un unico elemento* $\{\mathbf{v}\cdot\mathbf{n}_\Gamma\}$ *in* V_Γ' *tali che:*

$$\big\langle \mathbf{v}\cdot\mathbf{n},\ \mathbf{g}\big\rangle_{T_\Gamma{}',T_\Gamma} = \big\langle\, [\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!],\ \{\mathbf{g}\}\,\big\rangle_{H^{-\frac{1}{2}}(\Gamma),H^{\frac{1}{2}}(\Gamma)} + \big\langle\, \{\mathbf{v}\cdot\mathbf{n}_\Gamma\},\ [\![\mathbf{g}]\!]\,\big\rangle_{V_\Gamma',V_\Gamma}. \quad (2.1)$$

La dimostrazione di tale risultato si trova in [1].

Si noti che nel caso in cui $[\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!] \in L^2(\Gamma)$ e $\{\mathbf{v}\cdot\mathbf{n}_\Gamma\} \in L^2(\Gamma)$ la (2.1) si può riscrivere nel seguente modo:

$$\big\langle \mathbf{v}\cdot\mathbf{n},\ \mathbf{g}\big\rangle_{T_\Gamma{}',T_\Gamma} = \int_\Gamma [\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!]\,\{g\} + \int_\Gamma \{\mathbf{v}\cdot\mathbf{n}_\Gamma\}\,[\![g]\!]$$

dove $[\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!]$ e $\{\mathbf{v}\cdot\mathbf{n}_\Gamma\}$ sono le definizioni di salto e media (1.5)-(1.6) di $\mathbf{v}\cdot\mathbf{n}_\Gamma$ attraverso la frattura Γ.

2.2 La formulazione debole del problema

Per giungere alla formulazione debole del problema (1.9) si procede formalmente supponendo che tutte le funzioni siano sufficientemente regolari. Con tale approccio i passaggi seguenti risulteranno giustificati e si definiranno solo in seguito gli opportuni spazi funzionali in cui ricercare la soluzione.

Si consideri una funzione test $\mathbf{v} : \Omega_\Gamma \to \mathbb{R}^2$ sufficientemente regolare tale che $\mathbf{v}\cdot\mathbf{n}\,|_{\partial\Omega_\mathbf{u}} = 0$ e $\mathbf{v}^+ = \mathbf{v}^-$ su $\widetilde{\Gamma}\setminus\Gamma$ quindi con salto $[\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!]$ su $\widetilde{\Gamma}\setminus\Gamma$ pari a zero. Si moltiplichi l'equazione di Darcy nel mezzo per tale test $\mathbf{v}$ e successivamente la si integri su Ω_Γ:

$$\int_{\Omega_\Gamma} \mathbf{K}^{-1}\,\mathbf{u}\cdot\mathbf{v} + \int_{\Omega_\Gamma} \mathbf{v}\cdot\nabla p = 0. \quad (2.2)$$

Si utilizzerà la seguente formula di Stokes:

$$\int_{\Omega_\Gamma} \mathbf{v}\cdot\nabla p = -\int_{\Omega_\Gamma} p\,\mathrm{div}\,\mathbf{v} + \int_\Gamma [\![p\,\mathbf{v}\cdot\mathbf{n}_\Gamma]\!] + \int_{\partial\Omega_p} g_P\,\mathbf{v}\cdot\mathbf{n}.$$

Il termine $\int_\Gamma [\![p\,\mathbf{v}\cdot\mathbf{n}_\Gamma]\!]$ può essere visto come somma di due contributi:

$$\int_\Gamma [\![p\,\mathbf{v}\cdot\mathbf{n}_\Gamma]\!] = \int_\Gamma [\![p]\!]\{\mathbf{v}\cdot\mathbf{n}_\Gamma\} + \int_\Gamma [\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!]\{p\}. \tag{2.3}$$

Utilizzando la definizione di η e di $\xi_0 = \dfrac{2\xi-1}{4}$ le condizioni di accoppiamento (1.8) possono essere così riscritte:

$$\begin{cases} \eta\,\{\mathbf{u}\cdot\mathbf{n}_\Gamma\} = [\![p]\!] \\ \eta\,\xi_0\,[\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!] = \{p\} - \hat{p} \end{cases}$$

e sostituendole nella (2.3) si giunge a:

$$\int_\Gamma [\![p\,\mathbf{v}\cdot\mathbf{n}_\Gamma]\!] = \int_\Gamma \eta\,\{\mathbf{u}\cdot\mathbf{n}_\Gamma\}\{\mathbf{v}\cdot\mathbf{n}_\Gamma\} + \int_\Gamma \xi_0\,\eta\,[\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]\,[\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!] + \int_\Gamma \hat{p}\,[\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!].$$

Perciò riprendendo l'equazione di Darcy (2.2) si ottiene:

$$\begin{aligned} &\int_{\Omega_\Gamma} \mathbf{K}^{-1}\,\mathbf{u}\cdot\mathbf{v} - \int_{\Omega_\Gamma} p\,\mathrm{div}\,\mathbf{v} + \int_\Gamma \eta\,\{\mathbf{u}\cdot\mathbf{n}_\Gamma\}\,\{\mathbf{v}\cdot\mathbf{n}_\Gamma\} + \\ &+ \int_\Gamma \xi_0\,\eta\,[\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]\,[\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!] + \int_\Gamma \hat{p}\,[\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!] = -\int_{\partial\Omega_P} g_P\,\mathbf{v}\cdot\mathbf{n}. \end{aligned}$$

Come per la prima equazione si procede analogamente per l'equazione di conservazione della massa nel mezzo poroso, moltiplicandola per una funzione test sufficientemente regolare q ed integrandola sul dominio Ω_Γ:

$$\int_{\Omega_\Gamma} \mathrm{div}\,\mathbf{u}\,q = \int_{\Omega_\Gamma} f\,q.$$

Si agisce in maniera del tutto analoga a quanto fatto nel mezzo poroso anche nella frattura. Si moltiplica l'equazione di Darcy per una funzione $\hat{\mathbf{v}} : \Gamma \to \mathbb{R}$ sufficientemente regolare e tale che $\hat{\mathbf{v}}\cdot\boldsymbol{\tau} = 0$ su $\partial\Gamma_{\mathbf{u}}$ e la si integra su Γ:

$$\int_\Gamma \hat{K}^{-1}\,\hat{\mathbf{u}}\cdot\hat{\mathbf{v}} + \int_\Gamma \nabla_{\boldsymbol{\tau}}\,\hat{p}\cdot\hat{\mathbf{v}} = 0.$$

Integrando per parti si ottiene:

$$\int_\Gamma \hat{K}^{-1}\,\hat{\mathbf{u}}\cdot\hat{\mathbf{v}} - \int_\Gamma \hat{p}\,\mathrm{div}_{\boldsymbol{\tau}}\,\hat{\mathbf{v}} + \int_{\partial\Gamma_{\hat{p}}} \hat{g}_P\,\hat{\mathbf{v}}\cdot\boldsymbol{\tau} = 0.$$

Infine si consideri l'equazione di conservazione della massa nella frattura e la si moltiplichi per una funzione test sufficientemente regolare $\hat{q} : \Gamma \to \mathbb{R}$:

$$\int_\Gamma \mathrm{div}_{\boldsymbol{\tau}}\,\hat{\mathbf{u}}\,\hat{q} - \int_\Gamma [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]\,\hat{q} = \int_\Gamma \hat{f}\,\hat{q}.$$

In conclusione questa derivazione formale porta alla seguente formulazione:

Trovare $\mathbf{u}$, $\hat{\mathbf{u}}$, p e $\hat{p}$ tali che:

$$\begin{aligned}
&\int_{\Omega_\Gamma} \mathbf{K}^{-1}\,\mathbf{u}\cdot\mathbf{v} - \int_{\Omega_\Gamma} p\,\mathrm{div}\,\mathbf{v} + \int_\Gamma \eta\,\{\mathbf{u}\cdot\mathbf{n}_\Gamma\}\,\{\mathbf{v}\cdot\mathbf{n}_\Gamma\} + \\
&\qquad + \xi_0 \int_\Gamma \eta\,[\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]\,[\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!] + \int_\Gamma \hat{p}\,[\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!] = -\int_{\partial\Omega_P} g_P\,\mathbf{v}\cdot\mathbf{n} \qquad (2.4)\\
&\int_{\Omega_\Gamma} \mathrm{div}\,\mathbf{u}\,q = \int_{\Omega_\Gamma} f\,q\\
&\int_\Gamma \hat{K}^{-1}\hat{\mathbf{u}}\cdot\hat{\mathbf{v}} - \int_\Gamma \hat{p}\,\mathrm{div}_{\boldsymbol{\tau}}\,\hat{\mathbf{v}} = -\int_{\partial\Gamma_{\hat{p}}} \hat{g}_P\,\hat{\mathbf{v}}\cdot\boldsymbol{\tau}\\
&\int_\Gamma \mathrm{div}_{\boldsymbol{\tau}}\,\hat{\mathbf{u}}\,\hat{q} - \int_\Gamma [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]\,\hat{q} = \int_\Gamma \hat{f}\,\hat{q}
\end{aligned}$$

per ogni $\mathbf{v}$, $\hat{\mathbf{v}}$, q e $\hat{q}$ sufficientemente regolari.

2.2.1 Identificazione degli spazi funzionali

Si procede ora completando la derivazione della formulazione debole con l'identificazione degli opportuni spazi funzionali.
La presenza del terzo e del quarto integrale nella (2.4) mostra che è necessario richiedere una particolare regolarità per $[\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]$ e $\{\mathbf{u}\cdot\mathbf{n}_\Gamma\}$. Ciò equivale a ipotizzare che i termini $\mathbf{u}^+\cdot\mathbf{n}_\Gamma$ e $\mathbf{u}^-\cdot\mathbf{n}_\Gamma$ siano sufficientemente regolari. Per questo si introduce la definizione del seguente sottospazio di $H_{\mathrm{div}}(\Omega_\Gamma)$:

$$V = \Big\{\mathbf{u}\in H_{\mathrm{div}}(\Omega_\Gamma)\ :\ \mathbf{u}^+\cdot\mathbf{n}_\Gamma|_\Gamma\ \in\ L^2(\Gamma)\ \ \mathrm{e}\ \ \mathbf{u}^-\cdot\mathbf{n}_\Gamma|_\Gamma\ \in\ L^2(\Gamma)\Big\}$$

e dei suoi sottoinsiemi:

$$\begin{aligned}
V_{g_{\mathbf{u}}} &= \Big\{\mathbf{u}\in V\ :\ \mathbf{u}\cdot\mathbf{n} = g_{\mathbf{u}}\ \ \mathrm{su}\ \ \partial\Omega_{\mathbf{u}}\Big\}\\
V_0 &= \Big\{\mathbf{u}\in V\ :\ \mathbf{u}\cdot\mathbf{n} = 0\ \ \mathrm{su}\ \ \partial\Omega_{\mathbf{u}}\Big\},
\end{aligned}$$

V_0 è evidentemente uno spazio lineare e quindi sottospazio di $H_{\mathrm{div}}(\Omega_\Gamma)$.
Si assume che $g_{\mathbf{u}}$ sia sufficientemente regolare per cui che esista un rilevamento $R(g_{\mathbf{u}})\in H_{\mathrm{div}}(\Omega_\Gamma)$ la cui traccia coincida con $g_{\mathbf{u}}$, di conseguenza $V_{g_{\mathbf{u}}}$ è uno spazio affine, infatti $V_{g_{\mathbf{u}}} = V_0 + R(g_{\mathbf{u}})$.
Lo spazio V viene dotato del seguente prodotto interno:

$$(\mathbf{u},\mathbf{v})_V = \int_{\Omega_\Gamma} \mathbf{u}\,\mathbf{v} + \int_{\Omega_\Gamma} \mathrm{div}\,\mathbf{u}\,\mathrm{div}\,\mathbf{v} + \int_\Gamma [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]\,[\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!] + \int_\Gamma \{\mathbf{u}\cdot\mathbf{n}_\Gamma\}\,\{\mathbf{v}\cdot\mathbf{n}_\Gamma\}$$

e della norma indotta $\| \mathbf{u} \| = \sqrt{(\mathbf{u}, \mathbf{u})_V}$.
Inoltre nel mezzo poroso si richiede che:

$$p \in Q = L^2(\Omega_\Gamma).$$

Nella frattura gli spazi di riferimento per il flusso e la pressione sono:

$$\hat{V} = H_{\mathrm{div}}(\Gamma) \qquad \hat{Q} = L^2(\Gamma)$$
$$\hat{V}_{\hat{g}_\mathbf{u}} = \left\{ \mathbf{u} \in \hat{V} : \ \mathbf{u} \cdot \boldsymbol{\tau} = \hat{g}_\mathbf{u} \ \text{ su } \ \partial\Gamma_{\hat{\mathbf{u}}} \right\}.$$

Nel caso di $\hat{g}_\mathbf{u} = 0$, $\hat{V}_{\hat{g}_\mathbf{u}}$ si indica con $\hat{V}_0$ ed è un sottospazio lineare di $\hat{V}$. In generale $\hat{V}_{\hat{g}_\mathbf{u}}$ è uno spazio affine.
La norma e il prodotto interno dello spazio $\hat{V}$ sono quelli standard dello spazio $H_{\mathrm{div}}(\Gamma)$.
Si introducono poi gli spazi globali per velocità e pressione:

$$W = \left\{ (\mathbf{v}, \hat{\mathbf{v}}) \in \ V_{g_\mathbf{u}} \times \hat{V}_{\hat{g}_\mathbf{u}} \right\} \qquad \text{e} \qquad M = \left\{ (q, \hat{q}) \in \ Q \times \hat{Q} \right\} \tag{2.5}$$

dotati delle seguenti norme:

$$\begin{aligned}
\| (q, \hat{q}) \|_M^2 &= \| q \|_{L^2(\Omega_\Gamma)}^2 + \| \hat{q} \|_{L^2(\Gamma)}^2 \\
\| (\mathbf{v}, \hat{\mathbf{v}}) \|_W^2 &= \| \mathbf{v} \|_V^2 + \| \hat{\mathbf{v}} \|_{\hat{V}}^2 = \\
&= \| \mathbf{v} \|_{L^2(\Omega_\Gamma)}^2 + \| \hat{\mathbf{v}} \|_{L^2(\Gamma)}^2 + \| \operatorname{div} \mathbf{v} \|_{L^2(\Omega_\Gamma)}^2 + \\
&+ \| \operatorname{div}_{\boldsymbol{\tau}} \hat{\mathbf{v}} \|_{L^2(\Gamma)}^2 + \| [\![\mathbf{v} \cdot \mathbf{n}_\Gamma]\!] \|_{L^2(\Gamma)}^2 + \| \{ \mathbf{v} \cdot \mathbf{n}_\Gamma \} \|_{L^2(\Gamma)}^2.
\end{aligned} \tag{2.6}$$

Alternativamente la norma nello spazio W si definisce nel seguente modo:

$$\begin{aligned}
\| (\mathbf{v}, \hat{\mathbf{v}}) \|_W^2 &= \| \mathbf{v} \|_{L^2(\Omega_\Gamma)}^2 + \| \hat{\mathbf{v}} \|_{L^2(\Gamma)}^2 + \| \operatorname{div} \mathbf{v} \|_{L^2(\Omega_\Gamma)}^2 + \\
&+ \| \operatorname{div}_{\boldsymbol{\tau}} \hat{\mathbf{v}} \|_{L^2(\Gamma)}^2 + \| \mathbf{v}^+ \cdot \mathbf{n}_\Gamma \|_{L^2(\Gamma)}^2 + \| \mathbf{v}^- \cdot \mathbf{n}_\Gamma \|_{L^2(\Gamma)}^2.
\end{aligned} \tag{2.7}$$

Vale infatti il seguente risultato:

Lemma 2.2.1. *Le norme (2.6) e (2.7) sono equivalenti.*

Dimostrazione. Si pone $\mathbf{v}^\pm \cdot \mathbf{n}^\pm = a^\pm$. Segue che:

$$\{a\}^2 + [a]^2 = \frac{5\,(a^+)^2}{4} + \frac{5\,(a^-)^2}{4} - \frac{3\,a^+\,a^-}{2}. \tag{2.8}$$

Grazie alla disuguaglianza di Young vale che:

$$-\frac{3\,|a^+|\,|a^-|}{2} \geq -\left(\frac{3\,(a^+)^2}{4} + \frac{3\,(a^-)^2}{4} \right).$$

Applicando la precedente a (2.8) si ha che:

$$\{a\}^2 + [a]^2 \geq \frac{5\,(a^+)^2}{4} + \frac{5\,(a^-)^2}{4} - \frac{3\,|a^+|\,|a^-|}{2} \geq \frac{1}{2}\left((a^+)^2 + (a^-)^2\right).$$

Utilizzando la disuguaglianza di Young nella sua forma base, si ottiene una maggiorazione per (2.8):

$$\{a\}^2 + [a]^2 \leq \frac{5\,(a^+)^2}{4} + \frac{5\,(a^-)^2}{4} + \frac{3\,|a^+|\,|a^-|}{2} \leq 2\left((a^+)^2 + (a^-)^2\right).$$

Vale quindi la seguente catena di disuguaglianze:

$$\frac{1}{2}\left((a^+)^2 + (a^-)^2\right) \leq \{a\}^2 + [a]^2 \leq 2\left((a^+)^2 + (a^-)^2\right). \tag{2.9}$$

□

L'equivalenza permetterà di utilizzare la norma (2.6) o (2.7) a seconda della convenienza.

Si definiscono le seguenti forme bilineari:

$$\begin{aligned} a\left((\mathbf{u},\hat{\mathbf{u}}),\,(\mathbf{v},\hat{\mathbf{v}})\right) \;=\; & \int_{\Omega_\Gamma} \mathbf{K}^{-1}\,\mathbf{u}\cdot\mathbf{v} \;+\; \int_\Gamma \eta\,\{\mathbf{u}\cdot\mathbf{n}_\Gamma\}\,\{\mathbf{v}\cdot\mathbf{n}_\Gamma\} + \\ & +\,\xi_0 \int_\Gamma \eta\,[\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!][\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!] \;+\; \int_\Gamma \hat{K}^{-1}\,\hat{\mathbf{u}}\cdot\hat{\mathbf{v}} \end{aligned}$$

$$b\left((\mathbf{v},\hat{\mathbf{v}}),\,(p,\hat{p})\right) \;=\; -\int_{\Omega_\Gamma} \operatorname{div}\mathbf{v}\,p \;-\; \int_\Gamma \operatorname{div}_{\boldsymbol{\tau}}\,\hat{\mathbf{v}}\,\hat{p} \;+\; \int_\Gamma \hat{p}\,[\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!]$$

e i funzionali lineari (si assume che $(f,\,\hat{f})$ e $(g_P,\,\hat{g}_P)$ siano sufficientemente regolari, per esempio che $(f,\,\hat{f}) \in M$ e $(g_P,\,\hat{g}_P) \in H^{\frac{1}{2}}(\partial\Omega_P) \times H^{\frac{1}{2}}(\partial\Gamma_{\hat{p}})$):

$$F(q) \;=\; -\int_{\Omega_\Gamma} f\,q \qquad\qquad \hat{F}(\hat{q}) \;=\; -\int_\Gamma \hat{f}\,\hat{q}$$

$$T(\mathbf{v}) \;=\; -\int_{\partial\Omega_P} g_P\,\mathbf{v}\cdot\mathbf{n} \qquad\qquad \hat{T}(\hat{\mathbf{v}}) \;=\; -\int_{\partial\Gamma_{\hat{p}}} \hat{g}_P\,\hat{\mathbf{v}}\cdot\boldsymbol{\tau}$$

dove gli ultimi due integrali sono intesi nel senso di dualità.

La formulazione debole nel caso di frattura immersa è quindi:

Trovare $((\mathbf{u},\hat{\mathbf{u}}),\,(p,\hat{p})) \in W \times M$ *tali che:*

$$\begin{aligned} a\left((\mathbf{u},\hat{\mathbf{u}}),\,(\mathbf{v},\hat{\mathbf{v}})\right) + b\left((\mathbf{v},\hat{\mathbf{v}}),\,(p,\hat{p})\right) \;&=\; T(\mathbf{v}) + \hat{T}(\hat{\mathbf{v}}) \qquad \forall\,(\mathbf{v},\hat{\mathbf{v}}) \in V_0 \times \hat{V}_0 \\ b\left((\mathbf{u},\hat{\mathbf{u}}),\,(q,\hat{q})\right) \;&=\; F(q) + \hat{F}(\hat{q}) \qquad \forall\,(q,\hat{q}) \in Q \times \hat{Q}. \end{aligned} \tag{2.10}$$

2.3 Esistenza e unicità della soluzione

In questa sezione si dimostrerà che la formulazione debole (2.10) ammette una e una sola soluzione con dipendenza continua dai dati. A tale scopo si userà un risultato classico della teoria dei problemi di punto-sella, riportato in [16]. Per la dimostrazione occorre verificare la continuità delle forme bilineari $a\big((\mathbf{u},\hat{\mathbf{u}}),\,(\mathbf{v},\hat{\mathbf{v}})\big)$ e $b\big((\mathbf{v},\hat{\mathbf{v}}),\,(p,\hat{p})\big)$ e dei funzionali lineari $T(\mathbf{v})+\hat{T}(\hat{\mathbf{v}})$ e $F(q)+\hat{F}(\hat{q})$ e la coercività di $a\big((\mathbf{u},\hat{\mathbf{u}}),\,(\mathbf{v},\hat{\mathbf{v}})\big)$. Si dovrà dimostrare anche la validità della condizione inf-sup: questa è la parte particolarmente innovativa ed articolata della dimostrazione. Come già detto, la dimostrazione si limiterà al caso della singola frattura completamente immersa, che è anche il più interessante ed originale dal punto di vista dell'analisi.

Per semplicità, ma senza perdere di generalità, si considerano condizioni sulla pressione omogenee al bordo del dominio. Per la frattura invece il fatto che sia immersa induce condizioni omogenee sul flusso al bordo.

$$\begin{cases} p = 0 & \text{su } \partial\Omega \\ \hat{\mathbf{u}}\cdot\boldsymbol{\tau} = 0 & \text{su } \partial\Gamma. \end{cases}$$

Si ipotizza inoltre che valga la seguente disuguaglianza di traccia:

$$\|\,\mathbf{u}^+\cdot\mathbf{n}^+\,\|^2_{L^2(\Gamma)} + \|\,\mathbf{u}^-\cdot\mathbf{n}^-\,\|^2_{L^2(\Gamma)} \le C\,\|\,\mathbf{u}\,\|^2_{H_{\mathrm{div}}(\Omega_\Gamma)} \quad \forall\,\mathbf{u}\in V. \tag{2.11}$$

Questa ipotesi è ragionevole anche se la sua dimostrazione rigorosa è ancora in fase di elaborazione. Grazie alla (2.9) la (2.11) si può presentare anche nel seguente modo, più utile per il seguito:

$$\|\,[\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]\,\|^2_{L^2(\Gamma)} + \|\,\{\mathbf{u}\cdot\mathbf{n}_\Gamma\}\,\|^2_{L^2(\Gamma)} \le C_{tr}\,\|\,\mathbf{u}\,\|^2_{H_{\mathrm{div}}(\Omega_\Gamma)} \quad \forall\,\mathbf{u}\in V. \tag{2.12}$$

Il dominio e la frattura si considerano sufficientemente regolari. Vale quindi il seguente teorema di esistenza e unicità della soluzione.

Teorema 2.3.1 (**Esistenza e unicità della soluzione**). *Si ipotizza che lo spettro del tensore di permeabilità* $\mathbf{K}$ *(simmetrico e definito positivo) e il parametro* $\hat{K}$ *appartengano rispettivamente agli intervalli* $[K_{min},\,K_{max}]$ *e* $[\hat{K}_{min},\,\hat{K}_{max}]$*, con* $K_{min}>0$ *e* $\hat{K}_{min}>0$ *e che* η *appartenga all'intervallo* $[\eta_{min},\,\eta_{max}]$*, con* $\eta_{min}>0$*. Si assume inoltre che* $\xi > \frac{1}{2}-2\,\alpha\,K_{max}^{-1}\,C_{tr}^{-1}\,\eta_{max}^{-1}$ *con* $\alpha\in(0,\,1)$.
Allora il problema (2.10) è ben posto.

Le seguenti sezioni tratteranno i vari punti della dimostrazione, in particolare sarà provata la continuità e la coercività delle forme bilineari e in seguito la validità della condizione inf-sup.

2.3.1 Continuità e coercività delle forme bilineari

Verranno riportati in seguito i risultati sulla continuità e sulla coercività delle forme bilineari del problema (2.10).

Lemma 2.3.2. *La forma bilineare* $a\big((\mathbf{u},\hat{\mathbf{u}}),\,(\mathbf{v},\hat{\mathbf{v}})\big)$ *è continua.*

Dimostrazione. Grazie alle ipotesi fatte sui parametri del problema segue che:

$$\begin{aligned}|\,a\big((\mathbf{u},\hat{\mathbf{u}}),\,(\mathbf{v},\hat{\mathbf{v}})\big)\,| \le{}& K_{min}^{-1}\,\|\,\mathbf{u}\,\|_{L^2(\Omega_\Gamma)}\,\|\,\mathbf{v}\,\|_{L^2(\Omega_\Gamma)} + \\ &+\eta_{max}\,\|\,\{\mathbf{u}\cdot\mathbf{n}_\Gamma\}\,\|_{L^2(\Gamma)}\,\|\,\{\mathbf{v}\cdot\mathbf{n}_\Gamma\}\,\|_{L^2(\Gamma)} + \\ &+\hat{K}_{min}^{-1}\,\|\,\hat{\mathbf{u}}\,\|_{L^2(\Gamma)}\,\|\,\hat{\mathbf{v}}\,\|_{L^2(\Gamma)} + \\ &+|\xi_0|\,\eta_{max}\,\|\,[\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]\,\|_{L^2(\Gamma)}\,\|\,[\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!]\,\|_{L^2(\Gamma)}.\end{aligned}$$

Si osservi che ξ_0 potrebbe essere negativo.
La forma bilineare $a\big((\mathbf{u},\hat{\mathbf{u}}),\,(\mathbf{v},\hat{\mathbf{v}})\big)$ è quindi continua:

$$|\,a\big((\mathbf{u},\hat{\mathbf{u}}),\,(\mathbf{v},\hat{\mathbf{v}})\big)\,| \le C\,\|\,(\mathbf{u},\hat{\mathbf{u}})\,\|_W\,\|\,(\mathbf{v},\hat{\mathbf{v}})\,\|_W$$

con $C = \max\{\,K_{min}^{-1},\ \hat{K}_{min}^{-1},\ \eta_{max}\,\}$. □

Lemma 2.3.3. *La forma bilineare* $b\big((\mathbf{u},\hat{\mathbf{u}}),\,(q,\hat{q})\big)$ *è continua.*

Dimostrazione.

$$\begin{aligned}b\big((\mathbf{u},\hat{\mathbf{u}}),\,(q,\hat{q})\big) \le{}& \|\,\mathrm{div}\,\mathbf{u}\,\|_{L^2(\Omega_\Gamma)}\,\|\,q\,\|_{L^2(\Omega_\Gamma)} + \|\,\mathrm{div}_{\boldsymbol{\tau}}\,\hat{\mathbf{u}}\,\|_{L^2(\Gamma)}\,\|\,\hat{q}\,\|_{L^2(\Gamma)} + \\ &+\|\,\hat{q}\,\|_{L^2(\Gamma)}\,\|\,[\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]\,\|_{L^2(\Gamma)} \\ \le{}& \|\,(\mathbf{u},\hat{\mathbf{u}})\,\|_W\,\|\,(q,\hat{q})\,\|_M.\end{aligned}$$

□

Lemma 2.3.4. *Se* $\xi_0 > -\alpha\,K_{max}^{-1}\,C_{tr}^{-1}\,\eta_{max}^{-1}$ *per un* $\alpha \in (0,1)$ *la forma bilineare* $a\big((\mathbf{u},\hat{\mathbf{u}}),\,(\mathbf{v},\hat{\mathbf{v}})\big)$ *è coerciva su*

$$\widetilde{W} = \Big\{(\mathbf{v},\hat{\mathbf{v}}) \in W :\ b\big((\mathbf{v},\hat{\mathbf{v}}),\,(r,\hat{r})\big) = 0 \quad \forall\,(r,\hat{r}) \in M\Big\}.$$

Dimostrazione. Dimostrare che la forma bilineare $a\big((\mathbf{u},\hat{\mathbf{u}}),\,(\mathbf{v},\hat{\mathbf{v}})\big)$ è coerciva su $\widetilde{W}$ equivale a provare che esiste una costante C_a tale che:

$$\inf_{(\mathbf{u},\hat{\mathbf{u}})\in\widetilde{W}} \frac{a\big((\mathbf{u},\hat{\mathbf{u}}),\,(\mathbf{u},\hat{\mathbf{u}})\big)}{\|\,(\mathbf{u},\hat{\mathbf{u}})\,\|^2_{\widetilde{W}}} \ge C_a.$$

Se $(\mathbf{u}, \hat{\mathbf{u}}) \in \widetilde{W}$ si ha che:

$$\int_{\Omega_\Gamma} \operatorname{div} \mathbf{u}\, r \,+\, \int_\Gamma \operatorname{div}_{\boldsymbol{\tau}} \hat{\mathbf{u}}\, \hat{r} \,-\, \int_\Gamma [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]\, \hat{r} = 0 \quad \forall\, (r,\hat{r}) \in M.$$

Quindi segue che:

$$\operatorname{div} \mathbf{u} = 0 \qquad \text{e} \qquad \operatorname{div}_{\boldsymbol{\tau}} \hat{\mathbf{u}} = [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]. \tag{2.13}$$

Di conseguenza sostituendo le (2.13) nella definizione della norma (2.7) si ha che $\forall\, (\mathbf{u}, \hat{\mathbf{u}}) \in \widetilde{W}$:

$$\begin{aligned} \|\, (\mathbf{u}, \hat{\mathbf{u}})\, \|^2_W = \,& \|\, \mathbf{u}\, \|^2_{L^2(\Omega_\Gamma)} \,+\, \|\, \hat{\mathbf{u}}\, \|^2_{L^2(\Gamma)} \,+\, \|\, [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]\, \|^2_{L^2(\Gamma)} + \\ & + \|\, \mathbf{u}^+\cdot\mathbf{n}_\Gamma\, \|^2_{L^2(\Gamma)} \,+\, \|\, \mathbf{u}^-\cdot\mathbf{n}_\Gamma\, \|^2_{L^2(\Gamma)}. \end{aligned}$$

Ora si consideri $(\mathbf{u}, \hat{\mathbf{u}}) \in \widetilde{W}$ e il termine:

$$\begin{aligned} a\left((\mathbf{u}, \hat{\mathbf{u}}), (\mathbf{u}, \hat{\mathbf{u}})\right) \,=\, & \int_{\Omega_\Gamma} \mathbf{K}^{-1}\, \mathbf{u}\cdot\mathbf{u} \,+\, \int_\Gamma \eta\, \{\mathbf{u}\cdot\mathbf{n}_\Gamma\}^2 + \\ & + \,\xi_0 \int_\Gamma \eta\, [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]^2 \,+\, \int_\Gamma \hat{K}^{-1}\, \hat{\mathbf{u}}\cdot\hat{\mathbf{u}}. \end{aligned} \tag{2.14}$$

Grazie all'ipotesi su $\mathbf{K}$ e $\hat{K}$ il primo e l'ultimo termine di (2.14) vengono stimati semplicemente:

$$\begin{aligned} &\int_{\Omega_\Gamma} \mathbf{K}^{-1}\, \mathbf{u}\cdot\mathbf{u} \geq K^{-1}_{max}\, \|\, \mathbf{u}\, \|^2_{L^2(\Omega_\Gamma)} \\ &\int_\Gamma \hat{K}^{-1}\, \hat{\mathbf{u}}\cdot\hat{\mathbf{u}} \geq \hat{K}^{-1}_{max}\, \|\, \hat{\mathbf{u}}\, \|^2_{L^2(\Gamma)}. \end{aligned} \tag{2.15}$$

Riprendendo la (2.15) ed utilizzando la (2.13) e la disuguaglianza di traccia (2.12) vale la seguente minorazione:

$$\begin{aligned} &\int_{\Omega_\Gamma} \mathbf{K}^{-1}\, \mathbf{u}\,\cdot\,\mathbf{u} \,\geq\, K^{-1}_{max}\, \|\, \mathbf{u}\, \|^2_{L^2(\Omega_\Gamma)} \,=\, K^{-1}_{max}\, \|\, \mathbf{u}\, \|^2_{H_{\mathrm{div}}(\Omega_\Gamma)} \,= \\ &= \alpha\, K^{-1}_{max}\, \|\, \mathbf{u}\, \|^2_{H_{\mathrm{div}}(\Omega_\Gamma)} + (1-\alpha)\, K^{-1}_{max}\, \|\, \mathbf{u}\, \|^2_{H_{\mathrm{div}}(\Omega_\Gamma)} \,\geq \\ &\geq \alpha\, K^{-1}_{max}\, C^{-1}_{tr} \Big(\|\, [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]\, \|^2_{L^2(\Gamma)} \,+\, \|\, \{\mathbf{u}\cdot\mathbf{n}_\Gamma\}\, \|^2_{L^2(\Gamma)} \Big) + \\ &\qquad + \,(1-\alpha)\, K^{-1}_{max}\, \|\, \mathbf{u}\, \|^2_{L^2(\Omega_\Gamma)} \end{aligned}$$

dove $\alpha \in (0,1)$.
Grazie all'ipotesi su η i termini con salto e media di (2.14) vengono stimati

nel seguente modo:

$$\int_\Gamma \eta \{\mathbf{u}\cdot\mathbf{n}_\Gamma\}^2 \geq \eta_{min} \| \{\mathbf{u}\cdot\mathbf{n}_\Gamma\} \|^2_{L^2(\Gamma)}$$

$$\xi_0 \int_\Gamma \eta [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]^2 \geq \begin{cases} \xi_0\, \eta_{min} \| [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!] \|^2_{L^2(\Gamma)} & \text{se } \xi_0 > 0 \\ \xi_0\, \eta_{max} \| [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!] \|^2_{L^2(\Gamma)} & \text{se } \xi_0 \leq 0. \end{cases} \tag{2.16}$$

Si osservi che:

$$\xi_0\, \eta_{max} \| [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!] \|^2_{L^2(\Gamma)} = \xi_0 \frac{\eta_{max}}{\eta_{min}} \eta_{min} \| [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!] \|^2_{L^2(\Gamma)}.$$

Si introduce quindi la seguente funzione:

$$\sigma(\xi_0) = \begin{cases} \xi_0 & \text{se } \xi_0 > 0 \\ \xi_0 \dfrac{\eta_{max}}{\eta_{min}} & \text{se } \xi_0 \leq 0. \end{cases}$$

Conseguentemente la (2.16) si riformula nel seguente modo:

$$\xi_0 \int_\Gamma \eta [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]^2 \geq \eta_{min}\, \sigma(\xi_0) \| [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!] \|^2_{L^2(\Gamma)}.$$

Per quanto affermato finora vale che:

$$\begin{aligned} a\big((\mathbf{u},\hat{\mathbf{u}}),(\mathbf{u},\hat{\mathbf{u}})\big) \geq\ & \alpha\, K_{max}^{-1}\, C_{tr}^{-1} \Big(\| [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!] \|^2_{L^2(\Gamma)} + \| \{\mathbf{u}\cdot\mathbf{n}_\Gamma\} \|^2_{L^2(\Gamma)} \Big) + \\ & + (1-\alpha)\, K_{max}^{-1} \| \mathbf{u} \|^2_{L^2(\Omega_\Gamma)} + \hat{K}_{max}^{-1} \| \hat{\mathbf{u}} \|^2_{L^2(\Gamma)} + \\ & + \eta_{min} \| \{\mathbf{u}\cdot\mathbf{n}_\Gamma\} \|^2_{L^2(\Gamma)} + \\ & + \sigma(\xi_0)\, \eta_{min} \| [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!] \|^2_{L^2(\Gamma)}. \end{aligned}$$

Quindi:

$$\begin{aligned} a\big((\mathbf{u},\hat{\mathbf{u}}),(\mathbf{u},\hat{\mathbf{u}})\big) \geq\ & (1-\alpha)\, K_{max}^{-1} \| \mathbf{u} \|^2_{L^2(\Omega_\Gamma)} + \hat{K}_{max}^{-1} \| \hat{\mathbf{u}} \|^2_{L^2(\Gamma)} + \\ & + \Big(\eta_{min} + \alpha\, K_{max}^{-1}\, C_{tr}^{-1} \Big) \| \{\mathbf{u}\cdot\mathbf{n}_\Gamma\} \|^2_{L^2(\Gamma)} + \\ & + \Big(\sigma(\xi_0)\, \eta_{min} + \alpha\, K_{max}^{-1}\, C_{tr}^{-1} \Big) \| [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!] \|^2_{L^2(\Gamma)}. \end{aligned}$$

Infine si ottiene la coercività della forma bilineare grazie all'equivalenza tra norme mostrata nel lemma (2.2.1):

$$a\big((\mathbf{u},\hat{\mathbf{u}}),(\mathbf{u},\hat{\mathbf{u}})\big) \geq C_a \| (\mathbf{u},\, \hat{\mathbf{u}}) \|^2_W \qquad \forall\, (\mathbf{u},\, \hat{\mathbf{u}}) \in \widetilde{W}.$$

dove:

$$C_a \,=\, \min\Big\{(1-\alpha)\; K_{max}^{-1},\, \Big(\sigma(\xi_0)\,\eta_{min} + \alpha\, K_{max}^{-1}\, C_{tr}^{-1}\Big),\, \hat{K}_{max}^{-1}\Big\}.$$

Affinché tale forma bilineare sia coerciva è necessario che la costante ottenuta sia positiva, ciò equivale a richiedere che:

$$\sigma(\xi_0) \,>\, -\alpha\, K_{max}^{-1}\, C_{tr}^{-1}\, \eta_{min}^{-1}.$$

Ciò implica che:

$$\xi_0 \,>\, -\alpha\, K_{max}^{-1}\, C_{tr}^{-1}\, \eta_{max}^{-1}.$$

Esplicitando ξ_0 si ottiene la seguente condizione sul parametro di accoppiamento ξ:

$$\xi \,>\, \frac{1}{2} - 2\,\alpha\, K_{max}^{-1}\, C_{tr}^{-1}\, \eta_{max}^{-1}.$$

□

Osservazione 2.3.5. *Nella dimostrazione in [13] si considera una frattura che attraversa il dominio e si richiede inoltre che $\xi > \frac{1}{2}$ per ottenere la coercività della forma bilineare a $\big((\mathbf{u},\hat{\mathbf{u}}),(\mathbf{v},\hat{\mathbf{v}})\big)$. Nella dimostrazione riportata in questo capitolo non solo il risultato è stato esteso al caso di frattura immersa ma si è mostrato che la condizione $\xi > \frac{1}{2}$ può essere rilassata, in particolare il caso $\xi_0 = 0$ corrispondente a $\xi = \frac{1}{2}$ è sempre ammissibile.*

2.3.2 Continuità dei funzionali lineari

Secondo il teorema di esistenza e unicità della soluzione di un problema punto-sella occorre anche mostrare che i funzionali che compaiono a termine noto della formulazione siano lineari e continui. Quindi secondo la (2.10) è opportuno verificare che $T(\mathbf{v}) + \hat{T}(\hat{\mathbf{v}})$ e $F(q) + \hat{F}(\hat{q})$ siano funzionali lineari e continui.

Lemma 2.3.6. *Il funzionale $T(\mathbf{v}) + \hat{T}(\hat{\mathbf{v}})$ è lineare e continuo.*

Dimostrazione. Si è ipotizzato precedentemente che $(g_P,\, \hat{g}_P) \in H^{\frac{1}{2}}(\partial\Omega_P) \times H^{\frac{1}{2}}(\partial\Gamma_{\hat{p}})$.
La linearità di tale funzionale è ovvia e la continuità è quindi conseguenza della dualità tra $H^{\frac{1}{2}}$ e $H^{-\frac{1}{2}}$. □

Lemma 2.3.7. *Il funzionale $F(q) + \hat{F}(\hat{q})$ è lineare e continuo.*

Dimostrazione. Dato che $(f,\, \hat{f}) \in M$ la linearità di tale funzionale è ovvia e la dimostrazione della continuità segue dalla disuguaglianza di Cauchy-Schwarz. □

2.3.3 Dimostrazione della condizione inf-sup

A questo punto si deve dimostrare la validità della condizione inf-sup.

Lemma 2.3.8. *La forma bilineare* $b\big((\mathbf{v},\hat{\mathbf{v}}),(q,\hat{q})\big)$ *rispetta la condizione inf-sup, cioè esiste una costante* C_b *tale che:*

$$\inf_{(q,\hat{q})\in M}\ \sup_{(\mathbf{v},\hat{\mathbf{v}})\in W}\ \frac{b\big((\mathbf{v},\hat{\mathbf{v}}),(q,\hat{q})\big)}{\|\,(q,\hat{q})\,\|_M\,\|\,(\mathbf{v},\hat{\mathbf{v}})\,\|_W}\ \geq\ C_b. \tag{2.17}$$

Dimostrazione. La condizione (2.17) può essere ridefinita nel seguente modo:

$$\forall\ (q,\hat{q})\in M\quad \exists\ (\mathbf{v},\hat{\mathbf{v}})\in W:\quad b\big((\mathbf{v},\hat{\mathbf{v}}),(q,\hat{q})\big)\ \geq\ C_b\,\|\,(q,\hat{q})\,\|_M\,\|\,(\mathbf{v},\hat{\mathbf{v}})\,\|_W.$$

Per dimostrare la validità di tale condizione per prima cosa si introducono dei problemi ausiliari opportunamente definiti e successivamente si sfruttano le proprietà di regolarità ellittica e di stabilità delle soluzioni di tali problemi.

I problemi introdotti in seguito serviranno in particolare per trovare delle funzioni $(\mathbf{v},\hat{\mathbf{v}})\in W$ opportune, costruite in funzione di q e $\hat{q}$, da sostituire nella $b\big((\mathbf{v},\hat{\mathbf{v}}),(q,\hat{q})\big)$, dato che nella riformulazione della (2.17) si richiede l'esistenza di una $(\mathbf{v},\hat{\mathbf{v}})\in W$ che rispetti la condizione.
Si introduce innanzitutto un problema definito nella frattura.
Dato $\hat{q}\in L^2(\Gamma)$ sia $\hat{\varphi}$ soluzione del seguente problema:

$$\begin{cases} -\triangle_\tau\,\hat{\varphi} = \hat{q}\ -\ |\Gamma|^{-1}\int_\Gamma \hat{q} & \text{su}\ \ \Gamma \\ \dfrac{\partial\hat{\varphi}}{\partial\mathbf{n}_\Gamma} = 0 & \text{su}\ \ \partial\Gamma. \end{cases} \tag{2.18}$$

Per avere una soluzione unica $\hat{\varphi}$ occorre che $\int_\Gamma\hat{\varphi}=0$ cioè che $\hat{\varphi}\in H^1(\Gamma)\setminus\mathbb{R}$ che rappresenta lo spazio delle funzioni appartenenti ad $H^1(\Gamma)$ a media nulla.
La condizione di compatibilità è verificata; si ha infatti che:

$$\int_\Gamma\Big(\hat{q}\ -\ |\Gamma|^{-1}\int_\Gamma\hat{q}\Big)=0.$$

Si ricorda il seguente:

Lemma 2.3.9. *La soluzione* $\hat{\varphi}\in H^1(\Gamma)\setminus\mathbb{R}$ *di (2.18) esiste ed è unica. Inoltre vale la seguente disuguaglianza:*

$$\|\,\hat{\varphi}\,\|_{H^1(\Gamma)}\ \leq\ \hat{C}_s\ \left\|\,\hat{q}-|\Gamma|^{-1}\int_\Gamma q\,\right\|_{L^2(\Gamma)}. \tag{2.19}$$

Si definisce $\hat{\mathbf{v}}=\nabla_\tau\,\hat{\varphi}$ da cui segue che $\mathrm{div}_{\boldsymbol{\tau}}\,\hat{\mathbf{v}}=-\hat{q}\ +\ |\Gamma|^{-1}\int_\Gamma\hat{q}$.
La forma bilineare $b\big((\mathbf{v},\hat{\mathbf{v}}),(q,\hat{q})\big)$ diventa:

$$b\big((\mathbf{v},\hat{\mathbf{v}}),(q,\hat{q})\big)=-\int_{\Omega_\Gamma}\mathrm{div}\,\mathbf{v}\,q+\int_\Gamma\hat{q}^2-\int_\Gamma\Big(|\Gamma|^{-1}\,\hat{q}\int_\Gamma\hat{q}\Big)+\int_\Gamma[\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!]\,\hat{q}. \tag{2.20}$$

Si introduce ora un problema definito all'interno del dominio. Si noti che poiché Γ è a d-misura nulla, $L^2(\Omega_\Gamma)$ può essere identificato con $L^2(\Omega)$. Questo problema viene quindi posto in Ω e non in Ω_Γ.
Dato $q \in L^2(\Omega)$ si cerca φ soluzione di:

$$\begin{cases} -\triangle\varphi = q & \text{su } \Omega \\ \varphi = 0 & \text{su } \partial\Omega. \end{cases} \tag{2.21}$$

Si ricorda che:

Lemma 2.3.10. *La soluzione di (2.21) esiste ed è unica, inoltre vale la seguente disuguaglianza:*

$$\| \varphi \|_{H^1(\Omega)} \leq C_s \| q \|_{L^2(\Omega)}.$$

Osservazione 2.3.11. *Una funzione* $\mathbf{v} \in H_{\text{div}}(\Omega)$ *può essere identificata con una funzione di* $H_{\text{div}}(\Omega_\Gamma)$ *con* $[\![\mathbf{v} \cdot \mathbf{n}_\Gamma]\!] = 0$.

Come già affermato in precedenza si assume che sia possibile estendere la frattura fino al bordo in modo da dividere il dominio in due sottodomini. Ai due problemi precedenti si aggiunge un terzo problema da risolvere nel sottodominio Ω^+:

$$\begin{cases} -\triangle\varphi^+ = 0 & \text{su } \Omega^+ \\ \varphi^+ = 0 & \text{su } \partial\Omega^+ \setminus \Gamma \\ \dfrac{\partial\varphi^+}{\partial\mathbf{n}_\Gamma} = f & \text{su } \Gamma. \end{cases} \tag{2.22}$$

Per quest'ultimo problema si richiede che $f \in L^2(\Gamma)$ e che il suo supporto sia $\overline{\Gamma}$. Si darà in seguito una forma esplicita ad f in modo da eliminare il termine di salto nella forma bilineare (2.20) con il termine con la media di $\hat{q}$.
Si procede quindi con la determinazione di una $(\mathbf{v}, \hat{\mathbf{v}}) \in W$ ponendo:

$$\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_2$$

dove:

$$\mathbf{v}_1 = \nabla\varphi \qquad \text{e} \qquad \mathbf{v}_2 = \begin{cases} \nabla\varphi^+ & \text{in } \Omega^+ \\ 0 & \text{in } \Omega^-. \end{cases}$$

Per quanto affermato nell'osservazione 2.3.11 è possibile identificare la funzione $\mathbf{v}_1$ definita su Ω con una funzione in Ω_Γ con salto nullo attraverso la frattura. Ciò verrà utilizzato in seguito per calcolare $[\![\mathbf{v_1} \cdot \mathbf{n}_\Gamma]\!]$.

Lemma 2.3.12. $\mathbf{v}_2$ *appartiene a* V.

Dimostrazione. $\mathbf{v}_2|_{\Omega^+} \in H_{\mathrm{div}}(\Omega^+)$ e $\mathbf{v}_2|_{\Omega^-} \in H_{\mathrm{div}}(\Omega^-)$. Con $\widetilde{\Gamma}$ si indica la frattura Γ estesa fino ai bordi del dominio. Inoltre $\mathbf{v}_2^+ \cdot \mathbf{n}_\Gamma|_{\widetilde{\Gamma}} \in L^2(\widetilde{\Gamma})$ da cui segue che $\mathbf{v}_2^+ \cdot \mathbf{n}_\Gamma|_\Gamma \in L^2(\Gamma)$.
Da tali osservazioni si giunge alla tesi, ossia $\mathbf{v}_2 \in V$. Infatti la norma è limitata essendo tutti i suoi termini limitati:

$$\begin{aligned}
&\| \mathbf{v}_2 \|_V^2 = \\
&= \| \mathbf{v}_2 \|_{H_{\mathrm{div}}(\Omega^+)}^2 + \| \mathbf{v}_2 \|_{H_{\mathrm{div}}(\Omega^-)}^2 + \int_{\widetilde{\Gamma}} (\mathbf{v}_2^+ \cdot \mathbf{n}_\Gamma)^2 + \int_{\widetilde{\Gamma}} (-\mathbf{v}_2^- \cdot \mathbf{n}_\Gamma)^2 = \\
&= \| \mathbf{v}_2 \|_{H_{\mathrm{div}}(\Omega^+)}^2 + \| \mathbf{v}_2 \|_{H_{\mathrm{div}}(\Omega^-)}^2 + \int_{\Gamma} (\mathbf{v}_2^+ \cdot \mathbf{n}_\Gamma)^2 \leq \infty
\end{aligned}$$

dove si sono usate la seguente:

$$\mathbf{v}_2^+ \cdot \mathbf{n}_\Gamma - \mathbf{v}_2^- \cdot \mathbf{n}_\Gamma = 0 \qquad \text{su } \widetilde{\Gamma} \setminus \Gamma$$

che si riduce a

$$\mathbf{v}_2^+ \cdot \mathbf{n}_\Gamma = 0 \qquad \text{su } \widetilde{\Gamma} \setminus \Gamma$$

e l'equivalenza tra norme. □

Si torna ora all'analisi della forma bilineare $b\left((\mathbf{v}, \hat{\mathbf{v}}), (q, \hat{q})\right)$ e in particolare alle conseguenze dovute alla sostituzione della funzione $\mathbf{v}$ in essa.
Dalla definizione di $\mathbf{v}$ segue che:

$$\mathrm{div}\, \mathbf{v} = \mathrm{div}\, (\mathbf{v}_1 + \mathbf{v}_2) = \begin{cases} \triangle \varphi + \triangle \varphi^+ = -q & \text{in } \Omega^+ \\ \triangle \varphi = -q & \text{in } \Omega^- \end{cases}$$

e il termine di salto assume la seguente forma:

$$\begin{aligned}
[\![\mathbf{v} \cdot \mathbf{n}_\Gamma]\!] = [\![(\mathbf{v_1} + \mathbf{v_2}) \cdot \mathbf{n}_\Gamma]\!] = [\![\mathbf{v_1} \cdot \mathbf{n}_\Gamma]\!] + [\![\mathbf{v_2} \cdot \mathbf{n}_\Gamma]\!] = [\![\mathbf{v_2} \cdot \mathbf{n}_\Gamma]\!] = \\
= \mathbf{v}_2^+ \cdot \mathbf{n}_\Gamma - \mathbf{v}_2^- \cdot \mathbf{n}_\Gamma = \mathbf{v}_2^+ \cdot \mathbf{n}_\Gamma = \nabla \varphi^+ \cdot \mathbf{n}_\Gamma = f.
\end{aligned}$$

Quindi il valore della forma bilineare $b\left((\mathbf{v}, \hat{\mathbf{v}}), (q, \hat{q})\right)$ è:

$$\begin{aligned}
b\left((\mathbf{v}, \hat{\mathbf{v}}), (q, \hat{q})\right) &= \int_{\Omega_\Gamma} q^2 + \int_\Gamma \hat{q}^2 - \int_\Gamma \left(|\Gamma|^{-1} \hat{q} \int_\Gamma \hat{q} \right) + \int_\Gamma [\![\mathbf{v} \cdot \mathbf{n}_\Gamma]\!] \hat{q} = \\
&= \| (q, \hat{q}) \|_M^2 - |\Gamma|^{-1} \left(\int_\Gamma \hat{q} \right)^2 + \int_\Gamma f \hat{q}. \qquad (2.23)
\end{aligned}$$

Affinché gli ultimi due termini della forma bilineare precedente si annullino uno con l'altro si pone:

$$f = \begin{cases} |\Gamma|^{-1} \int_\Gamma \hat{q} & \text{su } \Gamma \\ 0 & \text{su } \partial\Omega^+ \setminus \Gamma. \end{cases}$$

Con tale scelta di f si eliminano gli ultimi due termini di (2.23) che diventa:

$$b\left((\mathbf{v},\hat{\mathbf{v}}),(q,\hat{q})\right) = \|\,(q,\hat{q})\,\|_M^2.$$

Ora occorre verificare che si riesca ad ottenere una stima del tipo

$$\|\,(\mathbf{v},\,\hat{\mathbf{v}})\,\|_W \leq C\,\|\,(q,\hat{q})\,\|_M$$

per gestire il denominatore nella condizione inf-sup (2.17).
Si cercherà quindi di ottenere una maggiorazione opportuna per ciascuno dei membri della norma $\|\cdot\|_W$ di cui si ricorda la definizione:

$$\|\,(\mathbf{v},\,\hat{\mathbf{v}})\,\|_W^2 = \|\,\mathbf{v}\,\|_{L^2(\Omega_\Gamma)}^2 + \|\,\hat{\mathbf{v}}\,\|_{L^2(\Gamma)}^2 + \|\,\mathrm{div}\,\mathbf{v}\,\|_{L^2(\Omega_\Gamma)}^2 + \|\,\mathrm{div}_{\boldsymbol{\tau}}\,\hat{\mathbf{v}}\,\|_{L^2(\Gamma)}^2 +$$
$$+ \|\,\mathbf{v}^+\cdot\mathbf{n}_\Gamma\,\|_{L^2(\Gamma)}^2 + \|\,\mathbf{v}^-\cdot\mathbf{n}_\Gamma\,\|_{L^2(\Gamma)}^2.$$

Per ottenere delle maggiorazioni verranno utilizzati dei risultati di regolarità ellittica e di stabilità. Per il problema (2.18) vale il risultato di stabilità (2.19), il cui secondo membro può essere così riscritto:

$$\left\|\,\hat{q} - |\Gamma|^{-1}\int_\Gamma \hat{q}\,\right\|_{L^2(\Gamma)}^2 = \int_\Gamma\left(\,\hat{q} - |\Gamma|^{-1}\int_\Gamma \hat{q}\,\right)^2 = \tag{2.24}$$

$$= \int_\Gamma\left\{\,\hat{q}^2 - 2\,|\Gamma|^{-1}\,\hat{q}\int_\Gamma\hat{q} + |\Gamma|^{-2}\left(\int_\Gamma\hat{q}\right)^2\right\} =$$

$$= \int_\Gamma \hat{q}^2 - 2\,|\Gamma|^{-1}\left(\int_\Gamma\hat{q}\right)^2 + |\Gamma|^{-2}\left(\int_\Gamma\hat{q}\right)^2|\Gamma| = \int_\Gamma\hat{q}^2 - |\Gamma|^{-1}\left(\int_\Gamma\hat{q}\right)^2 =$$

$$= \|\,\hat{q}\,\|_{L^2(\Gamma)}^2 - |\Gamma|^{-1}\|\,\hat{q}\,\|_{L^1(\Gamma)}^2.$$

Quindi per il problema (2.18) vale che:

$$\|\,\hat{\varphi}\,\|_{H^1(\Gamma)} \leq \hat{C}_s\left(\|\,\hat{q}\,\|_{L^2(\Gamma)}^2 - |\Gamma|^{-1}\|\,\hat{q}\,\|_{L^1(\Gamma)}^2\right). \tag{2.25}$$

Per il problema (2.21) valgono i seguenti risultati di regolarità ellittica:

$$\|\,\varphi\,\|_{H^2(\Omega_\Gamma)} \leq C_e\,\|\,q\,\|_{L^2(\Omega_\Gamma)}$$
$$\|\,\nabla\varphi\cdot\mathbf{n}\,\|_{L^2(\partial\Omega)} \leq C_n\,\|\,\varphi\,\|_{H^1(\Omega)} \tag{2.26}$$

e la seguente stima di stabilità:

$$\|\,\varphi\,\|_{H^1(\Omega_\Gamma)} \leq C_s\,\|\,q\,\|_{L^2(\Omega_\Gamma)}. \tag{2.27}$$

Per il problema (2.22) vale la seguente stima di stabilità:

$$\|\,\varphi^+\,\|_{H^1(\Omega^+)} \leq C^+\,\|\,f\,\|_{L^2(\Gamma)}. \tag{2.28}$$

Occorre ora maggiorare $\|\,(\mathbf{v},\,\hat{\mathbf{v}})\,\|_W$, analizzando un termine alla volta:

- Si considera $\| \mathbf{v} \|^2_{L^2(\Omega_\Gamma)}$:

$$\| \mathbf{v} \|^2_{L^2(\Omega_\Gamma)} = \| \mathbf{v}_1 + \mathbf{v}_2 \|^2_{L^2(\Omega_\Gamma)} \leq \left(\| \mathbf{v}_1 \|_{L^2(\Omega_\Gamma)} + \| \mathbf{v}_2 \|_{L^2(\Omega_\Gamma)} \right)^2$$
$$\leq 2 \| \mathbf{v}_1 \|^2_{L^2(\Omega_\Gamma)} + 2 \| \mathbf{v}_2 \|^2_{L^2(\Omega_\Gamma)}$$

  in cui:

$$\| \mathbf{v}_1 \|^2_{L^2(\Omega_\Gamma)} = \| \nabla \varphi \|^2_{L^2(\Omega_\Gamma)} \leq \| \varphi \|^2_{H^1(\Omega_\Gamma)} \leq C_s \| q \|^2_{L^2(\Omega_\Gamma)}$$

  grazie alla (2.27).

$$\| \mathbf{v}_2 \|^2_{L^2(\Omega_\Gamma)} = \| \nabla \varphi^+ \|^2_{L^2(\Omega^+)} \leq \| \varphi^+ \|^2_{H^1(\Omega^+)} \leq C^+ \| f \|^2_{L^2(\Gamma)}$$

  secondo la (2.28).

- Si analizza $\| \hat{\mathbf{v}} \|^2_{L^2(\Gamma)}$:

$$\| \hat{\mathbf{v}} \|^2_{L^2(\Gamma)} = \| \nabla \hat{\varphi} \|^2_{L^2(\Gamma)} \leq \| \hat{\varphi} \|^2_{H^1(\Gamma)}$$
$$\leq \hat{C}_s \left(\| \hat{q} \|^2_{L^2(\Gamma)} - |\Gamma|^{-1} \| \hat{q} \|^2_{L^1(\Gamma)} \right)$$

  grazie alla relazione di stabilità (2.25).

- Si tratta ora $\| \operatorname{div} \mathbf{v} \|^2_{L^2(\Omega_\Gamma)}$:

$$\| \operatorname{div} \mathbf{v} \|^2_{L^2(\Omega_\Gamma)} = \| \operatorname{div} \mathbf{v}_1 + \operatorname{div} \mathbf{v}_2 \|^2_{L^2(\Omega_\Gamma)}$$
$$\leq \left(\| \operatorname{div} \mathbf{v}_1 \|_{L^2(\Omega_\Gamma)} + \| \operatorname{div} \mathbf{v}_2 \|_{L^2(\Omega_\Gamma)} \right)^2$$
$$\leq 2 \| \operatorname{div} \mathbf{v}_1 \|^2_{L^2(\Omega_\Gamma)} + 2 \| \operatorname{div} \mathbf{v}_2 \|^2_{L^2(\Omega_\Gamma)}$$

  in cui:

$$\| \operatorname{div} \mathbf{v}_1 \|^2_{L^2(\Omega_\Gamma)} = \| \triangle \varphi \|^2_{L^2(\Omega_\Gamma)} = \| q \|^2_{L^2(\Omega_\Gamma)}$$

  Mentre

$$\| \operatorname{div} \mathbf{v}_2 \|^2_{L^2(\Omega_\Gamma)} = 0$$

  per costruzione di $\mathbf{v}_2$.

- Si considera ora:

$$\| \operatorname{div}_{\boldsymbol{\tau}} \hat{\mathbf{v}} \|^2_{L^2(\Gamma)} = \| \triangle_{\boldsymbol{\tau}} \hat{\varphi} \|^2_{L^2(\Gamma)} = \left\| \hat{q} - |\Gamma|^{-1} \int_\Gamma \hat{q} \right\|^2_{L^2(\Gamma)}$$
$$= \| \hat{q} \|^2_{L^2(\Gamma)} - |\Gamma|^{-1} \| \hat{q} \|^2_{L^1(\Gamma)}.$$

- Si tratta $\| \mathbf{v}^+ \cdot \mathbf{n}_\Gamma \|^2_{L^2(\Gamma)}$:

$$\begin{aligned} \| \mathbf{v}^+ \cdot \mathbf{n}_\Gamma \|^2_{L^2(\Gamma)} &\leq 2 \| \mathbf{v}_1^+ \cdot \mathbf{n}_\Gamma \|^2_{L^2(\Gamma)} + 2 \| \mathbf{v}_2^+ \cdot \mathbf{n}_\Gamma \|^2_{L^2(\Gamma)} \\ &= 2 \| \nabla \varphi \cdot \mathbf{n}_\Gamma \|^2_{L^2(\Gamma)} + 2 \| \nabla \varphi^+ \cdot \mathbf{n}_\Gamma \|^2_{L^2(\Gamma)} \\ &\leq C'_n \| q \|^2_{L^2(\Omega_\Gamma)} + 2 \| f \|^2_{L^2(\Gamma)} \end{aligned}$$

  grazie alla (2.26).

- Infine si ha:

$$\begin{aligned} \| \mathbf{v}^- \cdot \mathbf{n}_\Gamma \|^2_{L^2(\Gamma)} &\leq 2 \| \mathbf{v}_1^- \cdot \mathbf{n}_\Gamma \|^2_{L^2(\Gamma)} + 2 \| \mathbf{v}_2^- \cdot \mathbf{n}_\Gamma \|^2_{L^2(\Gamma)} \\ &= 2 \| \nabla \varphi \cdot \mathbf{n}_\Gamma \|^2_{L^2(\Gamma)} \leq C'_n \| q \|^2_{L^2(\Omega_\Gamma)} \end{aligned}$$

  secondo la (2.26).

Sostituendo l'espressione di f nelle norme delle disuguaglianze precedenti si ottiene:

$$\| f \|^2_{L^2(\Gamma)} = \left\| |\Gamma|^{-1} \int_\Gamma \hat{q} \right\|^2_{L^2(\Gamma)} = |\Gamma|^{-1} \left(\int_\Gamma \hat{q} \right)^2 = |\Gamma|^{-1} \| \hat{q} \|^2_{L^1(\Gamma)}.$$

Dalle precedenti disuguaglianze si giunge a:

$$\| (\mathbf{v}, \hat{\mathbf{v}}) \|^2_W \leq C_1 \| q \|^2_{L^2(\Omega_\Gamma)} + C_2 |\Gamma|^{-1} \| \hat{q} \|^2_{L^1(\Gamma)} + C_3 \| \hat{q} \|^2_{L^2(\Gamma)}.$$

Utilizzando la disuguaglianza di Holder si ottiene

$$|\Gamma|^{-1} \| \hat{q} \|^2_{L^1(\Gamma)} \leq |\Gamma|^{-1} \| \hat{q} \|^2_{L^2(\Gamma)} \| 1 \|^2_{L^2(\Gamma)} \leq |\Gamma|^{-1} |\Gamma| \| \hat{q} \|^2_{L^2(\Gamma)}$$

e si giunge alla seguente disuguaglianza:

$$\| (\mathbf{v}, \hat{\mathbf{v}}) \|^2_W \leq C_1 \| q \|^2_{L^2(\Omega_\Gamma)} + C_4 \| \hat{q} \|^2_{L^2(\Gamma)} \leq C \| (q, \hat{q}) \|^2_M$$

con C_1, C_4 e C costanti positive.

Considerando i risultati ottenuti per ogni $(q, \hat{q}) \in M$ è stata trovata una funzione $(\mathbf{v}, \hat{\mathbf{v}}) \in W$ tale che:

$$\frac{b((\mathbf{v}, \hat{\mathbf{v}}), (q, \hat{q}))}{\| (q, \hat{q}) \|_M \| (\mathbf{v}, \hat{\mathbf{v}}) \|_W} = \frac{\| (q, \hat{q}) \|^2_M}{\| (q, \hat{q}) \|_M \| (\mathbf{v}, \hat{\mathbf{v}}) \|_W} \geq \frac{\| (q, \hat{q}) \|^2_M}{\sqrt{C} \| (q, \hat{q}) \|^2_M} \geq C_b.$$

□

La dimostrazione è quindi conclusa per il caso di una sola frattura immersa.

2.4 Intersezioni tra fratture

Il caso dell'intersezione tra fratture viene analizzato in questa specifica sezione poiché richiede qualche accorgimento in più. Infatti nei punti di intersezione tra fratture occorre imporre alcune condizioni aggiuntive che, come visto nel capitolo precedente, riguardano la continuità della pressione e la conservazione del flusso.
In questa sezione si vuole giungere alla formulazione debole del problema (1.11). Come nel caso di una sola frattura immersa, occorre definire un opportuno setting funzionale, si introducono così i seguenti spazi, dove si usa la notazione introdotta nel paragrafo (1.2):

$$V = \left\{\mathbf{u} \in H_{\mathrm{div}}(\Omega_\Gamma) \,:\, \mathbf{u}^+ \cdot \mathbf{n}_\Gamma \,\in\, L^2(\Gamma) \;\text{ e }\; \mathbf{u}^- \cdot \mathbf{n}_\Gamma \,\in\, L^2(\Gamma)\right\}$$

$$V_{g_\mathbf{u}} = \left\{\mathbf{u} \in V \,:\, \mathbf{u} \cdot \mathbf{n} = g_\mathbf{u} \;\text{ su }\; \partial\Omega_\mathbf{u}\right\}$$

$$Q = L^2(\Omega_\Gamma)$$

$$\hat{V} = \left\{\hat{\mathbf{u}} \,:\, \hat{\mathbf{u}}_k = \hat{\mathbf{u}}|_{\gamma_k} \in H_{\mathrm{div}}(\gamma_k)\ \forall\, \gamma_k,\ \sum_{j:\,\gamma_j \in S_i} \hat{\mathbf{u}}_j \cdot \boldsymbol{\tau}_j|_i = 0 \quad \forall\, i \in I\right\} \tag{2.29}$$

$$\hat{V}_{\hat{g}_\mathbf{u}} = \left\{\hat{\mathbf{u}} \in \hat{V} \;:\; \hat{\mathbf{u}} \cdot \boldsymbol{\tau} = \hat{g}_\mathbf{u} \;\text{ su }\; I^\mathbf{u}\right\}$$

$$\hat{Q} = \left\{\hat{q} \,:\, \hat{q}_k = \hat{q}|_{\gamma_k} \in L^2(\gamma_k)\ \forall\, \gamma_k\right\}.$$

Inoltre si definiscono i seguenti spazi globali per velocità e pressione:

$$W = \left\{(\mathbf{v},\, \hat{\mathbf{v}}) \in V_{g_\mathbf{u}} \times \hat{V}_{\hat{g}_\mathbf{u}}\right\} \qquad \text{e} \qquad M = \left\{(q,\, \hat{q}) \in Q \times \hat{Q}\right\}.$$

Occorre precisare che con $\mathbf{u}^+ \cdot \mathbf{n}_\Gamma \,\in\, L^2(\Gamma)$ si richiede che $\mathbf{u}^+ \cdot \mathbf{n}_\Gamma|_{\gamma_k} \,\in\, L^2(\gamma_k)$ per ciascuna frattura $\gamma_k \in \Gamma$ e con $\mathbf{u}^- \cdot \mathbf{n}_\Gamma \,\in\, L^2(\Gamma)$ che $\mathbf{u}^- \cdot \mathbf{n}_\Gamma|_{\gamma_k} \,\in\, L^2(\gamma_k)$ per ciascuna frattura $\gamma_k \in \Gamma$.
L'introduzione della condizione di conservazione del flusso nella definizione dello spazio $\hat{V}$ (2.29) porta a delle semplificazioni nella derivazione della formulazione debole di (1.11). È opportuno rivedere tale condizione nel senso delle distribuzioni:

$$\sum_{k:\,\gamma_k \in S_i} \langle\, \hat{\mathbf{u}}_k \cdot \boldsymbol{\tau}_k,\, \varphi \,\rangle_{H^{-\frac{1}{2}}, H^{\frac{1}{2}}} = 0 \qquad \forall\, i \in I.$$

Più precisamente, $\forall\, i \in I$:

$$\sum_{k:\,\gamma_k \in S_i} \left(\int_{\gamma_k} \hat{\mathbf{u}}_k \cdot \nabla_{\boldsymbol{\tau}} \varphi \,+\, \int_{\gamma_k} \mathrm{div}_{\boldsymbol{\tau}} \hat{\mathbf{u}}_k\, \varphi \right) = 0 \qquad \forall\, \varphi \in H^1(\gamma_k)\ \forall\, \gamma_k \in S_i$$

dove $\varphi = 0$ su $\partial\gamma_k \setminus I$ per ciascuna frattura $\gamma_k \in S_i$.

In seguito alle considerazioni precedenti la formulazione debole del problema (1.11) è la seguente:

Trovare $\big((\mathbf{u},\,\hat{\mathbf{u}}),\,(p,\,\hat{p})\big) \in W \times M$ *tali che:*

$$\int_{\Omega_\Gamma} \mathbf{K}^{-1}\,\mathbf{u}\cdot\mathbf{v} - \int_{\Omega_\Gamma} \operatorname{div}\mathbf{v}\,p + \int_\Gamma \eta\,\{\mathbf{u}\cdot\mathbf{n}_\Gamma\}\,\{\mathbf{v}\cdot\mathbf{n}_\Gamma\} + \\ + \int_\Gamma \xi_0\,\eta\,[\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]\,[\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!] + \int_\Gamma \hat{p}\,[\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!] = -\int_{\partial\Omega_P} g_P\,\mathbf{v}\cdot\mathbf{n} \qquad \forall\,\mathbf{v}\in V_0$$

$$\int_{\Omega_\Gamma} \operatorname{div}\mathbf{u}\,q = \int_{\Omega_\Gamma} f\,q \qquad \forall\,q\in Q$$

$$\int_\Gamma \hat{K}^{-1}\,\hat{\mathbf{u}}\cdot\hat{\mathbf{v}} - \int_\Gamma \operatorname{div}_{\boldsymbol{\tau}}\,\hat{\mathbf{v}}\,\hat{p} = -\int_{I^P} \hat{g}_P\,\hat{\mathbf{v}}\cdot\boldsymbol{\tau} \qquad \forall\,\hat{\mathbf{v}}\in\hat{V}_0$$

$$\int_\Gamma \operatorname{div}_{\boldsymbol{\tau}}\,\hat{\mathbf{u}}\,\hat{q} - \int_\Gamma [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]\,\hat{q} = \int_\Gamma \hat{f}\,\hat{q} \qquad \forall\,\hat{q}\in\hat{Q}.$$

Si introducono le seguenti forme bilineari:

$$a\big((\mathbf{u},\hat{\mathbf{u}}),\,(\mathbf{v},\hat{\mathbf{v}})\big) = \int_{\Omega_\Gamma} \mathbf{K}^{-1}\,\mathbf{u}\cdot\mathbf{v} + \int_\Gamma \eta\,\{\mathbf{u}\cdot\mathbf{n}_\Gamma\}\,\{\mathbf{v}\cdot\mathbf{n}_\Gamma\} + \\ + \int_\Gamma \xi_0\,\eta\,[\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]\,[\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!] + \int_\Gamma \hat{K}^{-1}\,\hat{\mathbf{u}}\cdot\hat{\mathbf{v}}$$

$$b\big((\mathbf{v},\hat{\mathbf{v}}),\,(p,\hat{p})\big) = -\int_{\Omega_\Gamma} \operatorname{div}\mathbf{v}\,p - \int_\Gamma \operatorname{div}_{\boldsymbol{\tau}}\,\hat{\mathbf{v}}\,\hat{p} + \int_\Gamma \hat{p}\,[\![\mathbf{v}\cdot\mathbf{n}_\Gamma]\!]$$

e i seguenti funzionali lineari:

$$F(q) = -\int_{\Omega_\Gamma} f\,q \qquad \hat{F}(\hat{q}) = -\int_\Gamma \hat{f}\,\hat{q}$$

$$T(\mathbf{v}) = -\int_{\partial\Omega_P} g_P\,\mathbf{v}\cdot\mathbf{n} \qquad \hat{T}(\hat{\mathbf{v}}) = -\int_{I^P} \hat{g}_P\,\hat{\mathbf{v}}\cdot\boldsymbol{\tau}.$$

La formulazione debole nel caso di network di fratture immerse risulta quindi:

Trovare $\big((\mathbf{u},\hat{\mathbf{u}}),\,(p,\hat{p})\big) \in W \times M$ *tali che:*

$$a\big((\mathbf{u},\hat{\mathbf{u}}),\,(\mathbf{v},\hat{\mathbf{v}})\big) + b\big((\mathbf{v},\hat{\mathbf{v}}),\,(p,\hat{p})\big) = T(\mathbf{v}) + \hat{T}(\hat{\mathbf{v}}) \qquad \forall\,(\mathbf{v},\hat{\mathbf{v}})\in V_0\times\hat{V}_0$$

$$b\big((\mathbf{u},\hat{\mathbf{u}}),\,(q,\hat{q})\big) = F(q) + \hat{F}(\hat{q}) \qquad \forall\,(q,\hat{q})\in Q\times\hat{Q}.$$

Si può osservare che, grazie alla notazione introdotta e alla definizione del setting funzionale, la formulazione debole del problema (1.11), che presenta un network di fratture immerse, assume una forma del tutto analoga alla formulazione debole (2.10) del problema con una sola frattura immersa. La dimostrazione della buona posizione del problema con il network di fratture immerse è ancora in fase di elaborazione ma ci sono tutti i presupposti per sostenere che avrà una struttura simile a quella per una sola frattura immersa.

Capitolo 3

Discretizzazione del problema

In questo capitolo si forniranno le principali idee relative alle Differenze Finite Mimetiche applicate al problema (1.9). Verranno quindi richiamati e generalizzati i risultati riportati in [4] per giungere alla formulazione debole discretizzata del problema in esame. Successivamente verrà descritta la formulazione algebrica del problema globale, utile dal punto di vista implementativo. La matrice globale verrà assemblata a partire dai singoli blocchi e si dovranno imporre correttamente le condizioni di accoppiamento e le condizioni al bordo. A fine capitolo si discuterà su come affrontare la questione delle intersezioni tra fratture.

3.1 Un esempio

In questa sezione si analizzerà nel dettaglio la discretizzazione con le Differenze Finite Mimetiche dell'equazione di Darcy unidimensionale, utile per apprendere i principi alla base dello schema.

Il problema da risolvere è il seguente:

$$\begin{cases} -\dfrac{d}{dx}\left(K\,\dfrac{dp}{dx}\right) = b \qquad \text{in } (0,1) \\ p(0) = p(1) = 0 \end{cases} \tag{3.1}$$

dove b è un termine sorgente sufficientemente regolare.
Si può riscrivere l'equazione del secondo ordine (3.1) come un sistema di due equazioni del primo ordine nel seguente modo:

$$\begin{cases} u = -K\,\dfrac{dp}{dx} \\ \dfrac{du}{dx} = b. \end{cases} \tag{3.2}$$

L'equazione di Darcy in forma primale (3.1) è stata trasformata nell'equazione in forma mista (3.2).
Si consideri una griglia uniforme con $n+1$ nodi $x_i = (i-1)\Delta x$, $i = 1, \cdots, n+1$ e $\Delta x = 1/n$.
Si definisce la funzione discreta u_h appartenente a $\mathbb{R}^{n+1}$ per approssimare la funzione continua u nei nodi della mesh:

$$u_h = \left(u_i\right)_{i=1}^{n+1} \qquad \text{e} \qquad u_i \approx u(x_i).$$

Una definizione alternativa e più generale di u_i è:

$$u_i \approx \frac{1}{\Delta x} \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} u(x)\, dx.$$

Si procede analogamente per la funzione p, che viene approssimata da $p_h \in \mathbb{R}^n$ al centro degli elementi della mesh. Si ha che:

$$p_h = \left(p_{i+1/2}\right)_{i=1}^{n} \qquad \text{e} \qquad p_{i+1/2} \approx p(x_{i+1/2}).$$

Quindi è stata introdotta una discretizzazione *face-based* per u e *cell-based* per p. La discretizzazione *face-based* nel caso 1-D consiste infatti nell'associare un grado di libertà ad ogni vertice, quindi il campo discreto u_h può essere visto come l'approssimazione di $u(x)$ nei vertici della mesh. La discretizzazione *cell-based* nel caso 1-D consiste invece nell'associare un numero ad ogni elemento della mesh. Il campo discreto p_h può essere visto come l'approssimazione di una funzione scalare continua $p(x)$ al centro di ciascun elemento.

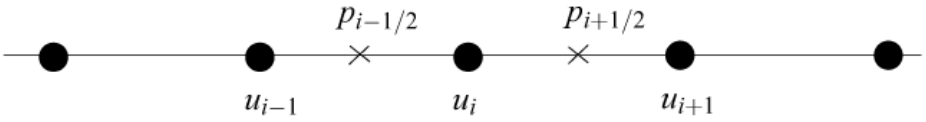


Figura 3.1: I punti rappresentano i vertici e le x i centri delle facce.

La discretizzazione con le Differenze Finite del problema (3.2) è:

$$\begin{cases} u_i = -K_i \dfrac{p_{i+1/2} - p_{i-1/2}}{\Delta x}, & i = 1, \cdots, n+1 \\ \dfrac{u_{i+1} - u_i}{\Delta x} = b(x_{i+1/2}), & i = 1, \cdots, n \end{cases} \tag{3.3}$$

con $p_{1/2} = p_{n+3/2} = 0$ condizioni al bordo e K_i approssimazione della permeabilità K sull'elemento i della mesh, per esempio $K_i = \frac{1}{x_{i+1} - x_i} \int_{x_i}^{x_{i+1}} K$.

Introducendo i simboli div_h e $\widetilde{\nabla}_h$ per gli operatori di divergenza e flusso discreti le (3.3) si possono riscrivere:

$$\begin{cases} u_h = -\widetilde{\nabla}_h\, p_h \\ \mathrm{div}_h\, u_h = b_h \end{cases}$$

dove $b_h = (b_{i+1/2})_{i=1}^n$ e gli operatori sono definiti come segue:

$$(\widetilde{\nabla}_h\, p_h)_i = K_i\, \frac{p_{i+1/2} - p_{i-1/2}}{\Delta x} \qquad (\mathrm{div}_h\, u_h)_{i+1/2} = \frac{u_{i+1} - u_i}{\Delta x}. \tag{3.4}$$

Vale la seguente relazione di dualità discreta:

$$\Delta x \sum_{i=1}^{n+1} \frac{p_{i+1/2} - p_{i-1/2}}{\Delta x}\, u_i = -\Delta x \sum_{i=1}^{n} \frac{u_{i+1} - u_i}{\Delta x}\, p_{i+1/2}$$

ed utilizzando le definizioni (3.4) segue che:

$$\Delta x \sum_{i=1}^{n+1} K_i^{-1}\, (\widetilde{\nabla}_h\, p_h)_i\, u_i = -\Delta x \sum_{i=1}^{n} (\mathrm{div}_h\, u_h)_{i+1/2}\, p_{i+1/2}. \tag{3.5}$$

La precedente è una formula di integrazione per parti discreta, è quindi l'analogo discreto della formula di Green moltiplicata per opportuni coefficienti:

$$\int_0^1 \hat{K}^{-1}\, u\, \hat{K}\, \frac{dp}{ds}\, ds = -\int_0^1 p\, \frac{du}{ds}\, ds \qquad \forall\, u \in H^1(0,1),\ \forall\, p \in H_0^1(0,1).$$

Tale espressione corrisponde alla relazione di dualità tra flusso $\hat{K}\, \dfrac{dp}{ds}$ e divergenza nel caso unidimensionale, dove per il flusso si usa un prodotto interno L^2 pesato con $\hat{K}^{-1}$. L'espressione (3.5) ne rappresenta l'equivalente discreto.

Osservazione 3.1.1. *La dualità tra gradiente discreto e divergenza discreta rappresenta una proprietà naturale dello schema a Differenze Finite. Uno schema mimetico deve però avere anche altre proprietà.*

Innanzitutto uno schema mimetico deve conservare la proprietà di dualità discreta anche in due o tre dimensioni e su griglie composte da poligoni e poliedri arbitrari. Ciò risulta impossibile se si discretizzano gli operatori gradiente e divergenza in maniera indipendente l'uno dall'altro. Questo punto si può elaborare usando la definizione formale della dualità discreta (3.5), riscrivendola come:

$$[\widetilde{\nabla}_h\, p_h, u_h]_{\mathscr{F}_h} \;=\; -[p_h, \mathrm{div}_h\, u_h]_{\mathscr{P}_h} \qquad \forall\, p_h \in \mathscr{P}_h,\ \forall\, u_h \in \mathscr{F}_h.$$

In questo caso $\mathscr{P}_h = \mathbb{R}^n$ e $\mathscr{F}_h = \mathbb{R}^{n+1}$ sono gli spazi in cui sono definite p_h e u_h rispettivamente e $[\cdot,\cdot]$ indica un opportuno prodotto interno in questi spazi. Nel caso di questo esempio

$$[p_h,\, q_h]_{\mathscr{P}_h} = \sum_{i=1}^{n} \Delta x\; p_{i+1/2}\; q_{i+1/2} \qquad \forall\; p_h,\; q_h \in \mathscr{P}_h$$

$$[u_h,\, v_h]_{\mathscr{F}_h} = \sum_{i=1}^{n+1} \Delta x\; K_i^{-1}\, u_i\; v_i \qquad \forall\; u_h,\; v_h \in \mathscr{F}_h.$$

Tali prodotti scalari si possono esprimere anche in forma matriciale nel seguente modo:

$$[p_h,\, q_h]_{\mathscr{P}_h} = q_h^T\; \mathbf{M}_{\mathscr{P}_h}\; p_h \qquad [u_h,\, v_h]_{\mathscr{F}_h} = v_h^T\; \mathbf{M}_{\mathscr{F}_h}\; u_h. \tag{3.6}$$

Data la semplice struttura delle matrici dei prodotti interni mimetici riportati in (3.6) si giunge alla seguente espressione per il gradiente discreto:

$$\widetilde{\nabla}_h \;= -\mathbf{M}_{\mathscr{F}_h}^{-1}\; \mathrm{div}_h^T\; \mathbf{M}_{\mathscr{P}_h}.$$

Si osserva quindi che l'operatore divergenza discreta e le matrici di massa rappresentano in maniera univoca l'operatore gradiente discreto.
In questo caso la costruzione delle matrici di massa è molto semplice, infatti sono entrambe diagonali e caratterizzate nel seguente modo:

$$(\mathbf{M}_{\mathscr{F}_h})_{ii} = \Delta x\, K_i^{-1} \quad \text{e} \quad (\mathbf{M}_{\mathscr{P}_h})_{ii} = \Delta x.$$

In generale però, la costruzione di tali matrici è più complessa, pertanto si procederà con la scomposizione nelle loro componenti locali. Sarà mostrato il procedimento per la matrice $M_{\mathscr{F}_h}$ di questo semplice esempio, assemblata a partire dalle matrici locali $\mathbf{M}_i$ nel seguente modo:

$$\mathbf{M}_{\mathscr{F}_h} = \sum_{i=1}^{n} \mathscr{N}_i^T\, \mathbf{M}_i\, \mathscr{N}_i \tag{3.7}$$

dove le matrici $\mathscr{N}_i$ sono identiche a quelle usate nel metodo degli Elementi Finiti e sono composte soltanto da 0 e 1 per indicare in quali righe e colonne della matrice globale vanno inseriti gli elementi della matrice locale $\mathbf{M}_i$.
Le matrici locali $\mathbf{M}_i$ sono in questo caso di dimensione 2×2 ed hanno la stessa interpretazione delle matrici globali, infatti vale che:

$$(v_i, v_{i+1})\, \mathbf{M}_i \begin{pmatrix} u_i \\ u_{i+1} \end{pmatrix} \approx \int_{x_i}^{x_{i+1}} K^{-1}\, v\, u\, dx.$$

Nel caso unidimensionale la matrice globale è una matrice scalare per cui si verifica facilmente che la seguente rispetta la (3.7):

$$\mathbf{M}_i^{DF} = \frac{\Delta x}{2\, K_i} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Con un calcolo diretto si mostra che:

$$(v_i, v_{i+1})\,\mathbf{M}_i^{DF}\begin{pmatrix} u_i \\ u_{i+1}\end{pmatrix} = \frac{\Delta x}{2}\,K_i^{-1}\,(v_i\,u_i + v_{i+1}\,u_{i+1}) = \\ = \int_{x_i}^{x_{i+1}} K^{-1}\,v\,u\,dx \;+\; O\big((\Delta x)^3\big).$$

Quindi la matrice di massa locale svolge il ruolo di una regola di quadratura.

Si può mostrare che la matrice locale $\mathbf{M}_i$ può essere derivata da due condizioni, la consistenza e la stabilità, generalizzabili a dimensioni arbitrarie. Se si sostituisce la funzione v con una costante v^0 e la funzione u con una funzione lineare u^1 e si assume che v^0 sia pari alla media di v nell'intervallo $[x_i, x_{i+1}]$ e che u^1 assuma i valori u_i e u_{i+1} agli estremi dell'intervallo allora si ha che:

$$\int_{x_i}^{x_{i+1}} K^{-1}\,v^0\,u^1\,dx = \int_{x_i}^{x_{i+1}} K^{-1}\,v\,u\,dx \;+\; O\big((\Delta x)^2\big).$$

Questa relazione permette di collegare gli integrali al prodotto interno locale, secondo la seguente identità:

$$\int_{x_i}^{x_{i+1}} K^{-1}\,v^0\,u^1\,dx = \frac{\Delta x}{2}\,K_i^{-1}\,v^0\,(u_i + u_{i+1}) = (v^0, v^0)\,\mathbf{M}_i^{DF}\begin{pmatrix} u_i \\ u_{i+1}\end{pmatrix}$$

che vale per ogni v^0, u_i e u_{i+1}.

Definizione 3.1.1. La matrice 2×2, $\mathbf{M}_i$, simmetrica e definita positiva, soddisfa la condizione di consistenza dello schema mimetico se

$$(v^0, v^0)\,\mathbf{M}_i\begin{pmatrix} u_i \\ u_{i+1}\end{pmatrix} = \int_{x_i}^{x_{i+1}} K^{-1}\,v^0\,u^1\,dx \qquad \forall\, v^0,\ \forall\,(u_i, u_{i+1})$$

dove u^1 è la funzione lineare $u^1 = u_i + \frac{x - x_i}{\Delta x}(u_{i+1} - u_i)$ e v^0 è una funzione costante.

Un'altra possibilità di matrice locale da cui assemblare $\mathbf{M}_{\mathscr{F}_h}$ che soddisfa la condizione di consistenza è la matrice degli Elementi Finiti di Raviart-Thomas di ordine zero 1-D:

$$\mathbf{M}_i^{RV} = \frac{\Delta x}{6\,K_i}\begin{bmatrix} 2 & 1 \\ 1 & 2\end{bmatrix}. \tag{3.8}$$

Inoltre la matrice locale $\mathbf{M}_i$ deve rispettare la condizione di stabilità, che equivale a richiedere che sia una matrice definita positiva.

Definizione 3.1.2. La matrice $\mathbf{M}_i$ soddisfa la condizione di stabilità se esistono due costanti positive c_1 e c_2 tali che:

$$c_1\,\Delta x\,(u_i^2+u_{i+1}^2) \leq (u_i\,,\,u_{i+1})\,\mathbf{M}_i \begin{pmatrix} u_i \\ u_{i+1} \end{pmatrix} \leq c_2\,\Delta x\,(u_i^2+u_{i+1}^2) \quad \forall\,(u_i\,,\,u_{i+1}).$$

Tale condizione è soddisfatta dalla matrice $\mathbf{M}_i^{RV}$ con costanti $c_1 = K_i^{-1}$ e $c_2 = 3\,K_i^{-1}$. Anche $\mathbf{M}_i^{DF}$ la soddisfa con entrambe le costanti pari ad K_i^{-1}.

Tale esempio, oltre ad evidenziare i concetti alla base delle Differenze Finite Mimetiche, fornisce tutti gli elementi necessari per la discretizzazione del problema di Darcy nella frattura.

3.2 Le Differenze Finite Mimetiche

La tecnica di discretizzazione usata è quella delle Differenze Finite Mimetiche sia nella frattura che nel mezzo poroso. Per quanto riguarda il mezzo poroso si entrerà nel dettaglio, mentre per l'applicazione alla frattura, che viene trattata come un oggetto monodimensionale, si fa riferimento all'esempio della sezione precedente.

Le Differenze Finite Mimetiche applicate al problema (1.9) prevedono i seguenti passi:

1. Introduzione di una opportuna griglia di calcolo.
2. Definizione dei gradi di libertà e dei relativi spazi.
3. Determinazione degli operatori mimetici primali e derivati.
4. Costruzione dei prodotti interni mimetici e della loro rappresentazione matriciale.
5. Espressione della condizione di consistenza mediante una forma algebrica lineare.

3.2.1 La griglia di calcolo e i gradi di libertà

Si consideri un dominio $\Omega \subset \mathbb{R}^2$ e la sua approssimazione discreta $\Omega_h \subset \Omega$. Occorre richiedere che Ω_h sia una approssimazione poligonale di Ω e che tutti i vertici di $\partial\Omega_h$ appartengano anche al bordo di Ω. Ω_h rappresenta anche la suddivisione in elementi del dominio computazionale e si ipotizza che tale partizione sia conforme. E' necessario inoltre che la mesh soddisfi due proprietà importanti:

Proprietà 3.2.1. *Esistono due numeri reali e positivi N^s e ρ_s e una sottopartizione conforme di Ω_h, chiamata T_h e costituita da triangoli, tali che:*

- *ciascun poligono $P \subset \Omega_h$ ammette una decomposizione $T_{h|P}$ costituita al più da N^s elementi;*

- *ciascun triangolo $T \in T_h$ è regolare. Infatti il rapporto tra il raggio della circonferenza inscritta nel triangolo r_T e il diametro dell'elemento h_T è limitato dal basso:*

$$\frac{r_T}{h_T} \geq \rho_S > 0.$$

Queste due proprietà impongono restrizioni sulla forma degli elementi ammissibili per evitare alcune situazioni problematiche. Resta comunque ampia la varietà di poligoni di forma regolare accettabili.

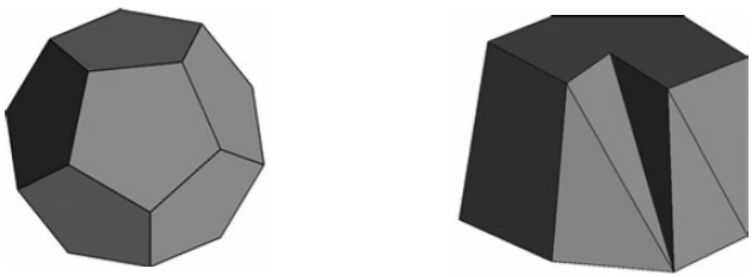


Figura 3.2: Esempi 3D: a sinistra un poliedro convesso di forma regolare, a destra degenere e non convesso.

Talvolta si tende a considerare un'ulteriore condizione, più restrittiva a proposito della forma degli elementi della mesh:

Proprietà 3.2.2. *Ogni poligono P è un insieme stellato rispetto a un punto $\bar{\mathbf{x}}_P \in P$ e ogni faccia f è un insieme stellato rispetto ad un punto $\bar{\mathbf{x}}_f \in f$. Per di più la sottopartizione triangolare T_h è semplice.*

Prima di specificare le conseguenze delle proprietà (3.2.1) è opportuno fare alcune precisazioni sulla notazione. Si indica una faccia di un elemento P con f e un vertice con v. In 2D $|P|$ e $|f|$ indicano rispettivamente l'area dell'elemento P e la lunghezza della faccia f. h_P rappresenta il diametro dell'elemento P. $\mathbf{n}_f$ è il vettore unitario normale alla faccia f, univocamente definito, e $\mathbf{x}_f$ rappresenta il centro della faccia. $\mathbf{x}_P$ indica il centro dell'elemento e $\mathbf{x}_v$ è il vettore delle coordinate del nodo v. Inoltre gli insiemi di nodi, facce ed elementi della mesh vengono rispettivamente indicati con $\mathscr{V}$, $\mathscr{F}$ e $\mathscr{P}$.

Le proprietà (3.2.1) hanno delle importanti conseguenze utilizzate nelle derivazioni teoriche successive:

1. Esistono due interi positivi $\mathscr{N}^{\mathscr{F}}$ e $\mathscr{N}^{\mathscr{V}}$ che dipendono solo da $\mathscr{N}^S$ tali che ogni elemento P ha al più $\mathscr{N}^{\mathscr{F}}$ facce e ogni faccia f ha al più $\mathscr{N}^{\mathscr{V}}$ vertici.

2. Le quantità geometriche relative a ciascun elemento $P \in \Omega_h$ sono uniformemente limitate dall'alto e dal basso. Più precisamente esistono due costanti a_* e l_* che dipendono solo da $\mathscr{N}^S$ e da ρ_S tali che per ciascuna faccia $f \in \partial P$ e per ogni elemento P vale:

$$a_* \, h_P^d \leq |P| \qquad \text{e} \qquad l_* \, h_P^{d-1} \leq |f|$$

   dove d indica la dimensione del dominio.

3. Esiste una costante b_* dipendente solo da $\mathscr{N}^S$ e da ρ_S tale che per ogni $P \in \Omega_h$ e per ogni $T \in T_{h|P}$ vale che:

$$h_P \leq b_* \, h_T.$$

4. Esiste una costante C indipendente da h_P e tale che:

$$\sum_{f \in \partial P} \|\phi\|^2_{L^2(f)} \leq C \left(h_P^{-1} \, \|\phi\|^2_{L^2(P)} + h_P \, |\phi|^2_{H^1(P)} \right)$$

   per ogni funzione $\phi \in H^1(P)$.

5. **Stime di approssimazione:** sia $m \in \mathbb{N}$. Esiste una costante C^{Int} indipendente da h_P tale che per ogni funzione $q \in H^{s+1}(P)$ con $s \in \mathbb{R}$ e $0 \leq s \leq m$ esiste una approssimazione polinomiale $q_P^{(m)} \in \mathbb{P}_m(P)$ tale che:

$$\|q - q_P^{(m)}\|_{L^2(P)} + \sum_{k=1}^{[s]} h_P^k \, |q - q_P^{(m)}|_{H^k(P)} \leq C^{Int} \, h_P^{s+1} \, |q|_{H^{s+1}(P)}$$

   dove $[s]$ è la parte intera di s.

Per le dimostrazioni di tali proprietà si rimanda a [4].

Dopo aver introdotto la mesh è di fondamentale importanza definire i gradi di libertà, in generale è possibile scegliere tra quattro tipi che corrispondono ai vari oggetti della griglia, cioè vertici, lati, facce e celle. In questo caso nel mezzo poroso è opportuno utilizzare un campo discreto *cell-based* p_h per descrivere la pressione e un campo *face-based* $\mathbf{u}_h$ per descrivere la velocità. Il primo campo associa un grado di libertà ad ogni cella $P \in \Omega_h$ della mesh, il secondo ad ogni faccia f della griglia. L'insieme dei gradi di libertà è lo spazio $\mathscr{P}_h$ nel primo caso e lo spazio $\mathscr{F}_h$ nel secondo. La collezione dei gradi di libertà associati a ciascuna cella forma il vettore algebrico $p_h \in \mathscr{P}_h$

che può essere interpretato come l'approssimazione di una funzione scalare $p(\mathbf{x})$ nei centri degli elementi della mesh o come l'approssimazione del suo valore medio. Analogamente il valore della velocità associato ad ogni faccia $f \in \mathscr{F}$ è indicato con u_f e l'insieme di tutti i gradi di libertà costituisce il vettore $\mathbf{u}_h$. Il valore u_f rappresenta il flusso medio nella direzione normale $\mathbf{n}_f$ attraverso la faccia f della mesh.

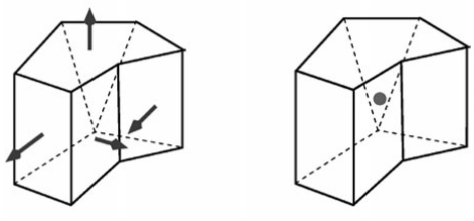


Figura 3.3: A sinistra un campo *face-based*, a destra *cell-based*.

Questo tipo di discretizzazione è quello di ordine più basso, infatti è accurato al primo ordine per la velocità e al secondo ordine per la pressione se quest'ultima appartiene ad $H^2(\Omega)$, come mostrato in [4]. Se la mesh è costituita interamente da triangoli, nel caso 2-D, il metodo delle Differenze Finite Mimetiche è una generalizzazione del metodo degli *Elementi Finiti di Raviart-Thomas*.

Si è ipotizzato precedentemente che il dominio sia bidimensionale e per ora si assume che ci sia una sola frattura, il caso di un network di fratture sarà affrontato successivamente. Poiché è stato adottato il *Modello Ridotto* la frattura ha uno spessore trascurabile rispetto al mezzo e viene quindi considerata come un'interfaccia unidimensionale approssimata con le facce della mesh. La griglia, sia nell'analisi teorica che nei casi test numerici, è conforme alle fratture e la suddivisione della frattura in elementi viene indicata con Γ_h. Per quanto riguarda la discretizzazione nella frattura si procede come indicato nell'esempio iniziale, con una discretizzazione *face-based* per la velocità nella frattura e *cell-based* per la pressione. Gli spazi mimetici vengono indicati con $\hat{\mathscr{P}}_h$ per i gradi di libertà associati alla pressione e $\hat{\mathscr{F}}_h$ per quelli legati alla velocità.

Osservazione 3.2.3. *Il mezzo poroso corrisponde al dominio Ω_Γ che risulta privato della frattura. Dovendo rappresentare salti e medie attraverso Γ risulta quindi naturale sdoppiare i gradi di libertà del campo di velocità in Ω_Γ in prossimità di Γ. Inoltre è stato ipotizzato che la griglia sia conforme alla frattura, di conseguenza a una faccia di frattura corrispondono ben tre gradi di libertà, due per la velocità nel mezzo poroso e uno per la pressione nella frattura.*

Lo sdoppiamento dei gradi di libertà nel mezzo poroso è rappresentativo della presenza di discontinuità in esso: quando il fluido incontra la frattura subisce delle variazioni e il flusso non si conserva attraverso le facce.
Questo sdoppiamento è di fondamentale importanza per introdurre la dipendenza dei problemi nel mezzo e nella frattura.

3.2.2 Gli operatori di proiezione

Gli operatori di proiezione trasferiscono funzioni sufficientemente regolari a valori scalari o vettoriali in uno spazio discreto, che nel mezzo poroso potrà essere $\mathscr{P}_h$ o $\mathscr{F}_h$. Forniscono quindi l'approssimazione discreta di un campo scalare o vettoriale. Se si indica con $\mathscr{S}_h$ il generico spazio discreto, allora l'operatore di proiezione su tale spazio è indicato con $\Pi^{\mathscr{S}}$ e nel caso di restrizione dell'operatore all'elemento P della mesh quest'ultimo viene indicato con $\Pi_P^{\mathscr{S}}$.
L'operatore di proiezione sullo spazio delle facce si applica in generale ad una funzione $\mathbf{u}(\mathbf{x})$ a valori vettoriali, sufficientemente regolare e tale che l'integrale sulle facce della mesh delle sue componenti normali sia ben definito. L'operatore di proiezione agisce tra i seguenti spazi: $\Pi^{\mathscr{F}} : X \to \mathscr{F}_h$ ed è stabile per funzioni vettoriali appartenenti al seguente spazio:

$$X = \left\{\mathbf{v} \in (L^2(\Omega_\Gamma))^d,\ s > 2,\ \operatorname{div}\mathbf{v} \in L^2(\Omega_\Gamma)\right\}.$$

L'approssimazione di $\mathbf{u}(\mathbf{x})$, $\mathbf{u}_h \in \mathscr{F}_h$, si ottiene applicando l'operatore di proiezione $\Pi^{\mathscr{F}}$, definito nel seguente modo:

$$\mathbf{u}_h = \Pi^{\mathscr{F}}(\mathbf{u}) = (u_f)_{f\in\mathscr{F}}, \qquad u_f = \frac{1}{|f|}\int_f \mathbf{u}\cdot\mathbf{n}_f\, dS$$

dove $\mathbf{n}_f$ è il vettore normale unitario ortogonale alla faccia f e $\mathscr{F}$ è l'insieme delle facce della mesh su cui si ha un grado di libertà.
Nel caso di facce appartenenti a Γ si hanno due gradi di libertà per ciascuna di esse:

$$u_{f\pm} = \frac{1}{|f|}\int_f \mathbf{u}^{\pm}\cdot\mathbf{n}_f\, dS.$$

Convenzionalmente è stato stabilito che la faccia $+$ è quella che appartiene alla cella caratterizzata dalla normale di frattura concorde con la normale alla faccia uscente dalla cella.
Si noti che l'operatore di proiezione è suriettivo.
L'operatore di proiezione ristretto all'elemento P della mesh è definito in maniera analoga:

$$\Pi_P^{\mathscr{F}}(\mathbf{u}) = (u_f)_{f\in\partial P}.$$

È conveniente introdurre il flusso $u_{P,f}$ attraverso f nella direzione $\mathbf{n}_{P,f}$ che indica la normale uscente dalla faccia $f \in \partial P$. Ovviamente vale che

$u_{P,f} = \alpha_{P,f}\, u_f$ dove $\alpha_{P,f} = \mathbf{n}_{P,f} \cdot \mathbf{n}_f$.

L'operatore di proiezione sullo spazio $\mathscr{P}_h$ è definito concettualmente in modo simile. Solitamente si richiede che $p(\mathbf{x})$ sia una funzione scalare sufficientemente regolare, tale che il suo integrale su un sottoinsieme compatto di Ω esista. In questo caso $p(\mathbf{x})$ appartiene ad $L^2(\Omega_\Gamma)$ e l'operatore di proiezione agisce tra i seguenti spazi: $\Pi^{\mathscr{P}} : L^2(\Omega_\Gamma) \to \mathscr{P}_h$. Allora l'approssimazione di $p(\mathbf{x})$, $p_h \in \mathscr{P}_h$, è definita nel seguente modo:

$$p_h = \Pi^{\mathscr{P}}(p) = (p_P)_{P \in \mathscr{P}}, \qquad p_P = \frac{1}{|P|} \int_P p \, dV$$

dove $\mathscr{P}$ è l'insieme degli elementi della mesh.
La dimensione di $\mathscr{P}_h$ è pari al numero degli elementi della mesh.

3.2.3 Gli operatori mimetici

Occorre ora definire gli operatori mimetici, per prima cosa si introducono gli operatori primali e in seguito si ottengono i derivati utilizzando la formula di integrazione per parti o il teorema di Stokes a seconda della dimensione del dominio. Saranno analizzati in seguito la divergenza discreta come operatore primale e il flusso discreto come operatore derivato corrispondente. In realtà quale sia l'operatore primale e quale il duale dipende dalla scelta degli spazi discreti.

Il teorema di Stokes su una cella bidimensionale P assume la seguente forma:

$$\int_P \operatorname{div} \mathbf{u} \, dV = \int_{\partial P} \mathbf{u} \cdot \mathbf{n}_P \, dS \tag{3.9}$$

dove $\mathbf{n}_P$ è il vettore normale unitario uscente dalla cella.
La (3.9) suggerisce in realtà una scelta naturale per i gradi di libertà e la loro relazione con i diversi oggetti della mesh.
Per identificare l'operatore derivato, nel caso continuo, si usa il Teorema di Green su Ω_Γ:

$$\int_{\Omega_\Gamma} p \operatorname{div} \mathbf{u} \, dV = -\int_{\Omega_\Gamma} \nabla p \cdot \mathbf{u} \, dV + \int_\Gamma [\![p \, \mathbf{u} \cdot \mathbf{n}_\Gamma]\!] \, dS + \int_{\partial\Omega} p \, \mathbf{u} \cdot \mathbf{n} \, dS.$$

Nel caso di condizioni omogenee su $\partial\Omega$ non è presente l'ultimo termine, in tal caso la precedente uguaglianza può essere riformulata nel seguente modo:

$$\int_{\Omega_\Gamma} p \operatorname{div} \mathbf{u} \, dV = -\int_{\Omega_\Gamma} \nabla p \cdot \mathbf{u} \, dV + \int_\Gamma [\![p]\!] \{ \mathbf{u} \cdot \mathbf{n}_\Gamma \} \, dS + \int_\Gamma \{ p \} [\![\mathbf{u} \cdot \mathbf{n}_\Gamma]\!] \, dS.$$

Nell'approccio mimetico la dualità tra gli operatori viene spesso combinata con i coefficienti del problema. Per esempio supponendo che $\mathbf{K}$ sia un tensore

definito positivo la precedente può essere riscritta nel seguente modo:

$$\int_{\Omega_\Gamma} p \operatorname{div} \mathbf{u}\, dV = -\int_{\Omega_\Gamma} \mathbf{K}^{-1} (\mathbf{K}\nabla p) \cdot \mathbf{u}\, dV + \int_\Gamma [\![p]\!] \{\mathbf{u}\cdot\mathbf{n}_\Gamma\}\, dS + \\ + \int_\Gamma \{p\} [\![\mathbf{u}\cdot\mathbf{n}_\Gamma]\!]\, dS. \tag{3.10}$$

Questa è una relazione di dualità pesata tra i due operatori primale, div, e derivato, $\mathbf{K}\nabla$, a livello continuo.

A partire dalla (3.9) segue la definizione della divergenza discreta applicata al campo discreto $\mathbf{u}_h$ su ciascuna cella P:

$$(\operatorname{div}_h \mathbf{u}_h)_P = \frac{1}{|P|} \sum_{f\in\mathscr{F}_P} \alpha_{P,f} |f| u_f \tag{3.11}$$

dove, come già detto, $\alpha_{P,f} = \mathbf{n}_{P,f}\cdot\mathbf{n}_f$ assume il valore ± 1 a seconda dell'orientazione relativa della faccia.
Quindi $\operatorname{div}_h \mathbf{u}_h \in \mathscr{P}_h$ e div_h è un operatore lineare tale che $\operatorname{div}_h : \mathscr{F}_h \to \mathscr{P}_h$.

Nell'ambito discreto vale una relazione di dualità analoga alla (3.10) esprimibile nel seguente modo:

$$[\operatorname{div}_h \mathbf{v}_h, p_h]_{\mathscr{P}_h} = -\big[\mathbf{v}_h, \widetilde{\nabla}_h p_h\big]_{\mathscr{F}_h} + \sum_{f\in\Gamma_h} |f| [\![p_f]\!] \{v_f\} + \sum_{f\in\Gamma_h} |f| \{p_f\} [\![v_f]\!]. \tag{3.12}$$

La dualità fa infatti intervenire tre termini, il primo è quello classico che rappresenta la componente legata alla matrice solida e i due termini addizionali forniscono il contributo dovuto alla presenza della frattura.
$\{v_f\}$ rappresenta la media dei gradi di libertà della velocità sulla faccia di frattura f e $[\![v_f]\!]$ il salto:

$$\{v_f\} = \frac{v_{f+} + v_{f-}}{2} \qquad\qquad [\![v_f]\!] = v_{f+} - v_{f-}.$$

La pressione viene invece discretizzata sulle celle della mesh e con $[\![p_f]\!]$ si indica il salto delle pressioni fra le celle che hanno in comune la faccia di frattura f e analogamente con $\{p_f\}$ si indica la media delle pressioni delle due celle che condividono la faccia f:

$$\{p_f\} = \frac{p_f^+ + p_f^-}{2} \qquad\qquad [\![p_f]\!] = p_f^+ - p_f^-.$$

La relazione (3.12) definisce in maniera univoca l'operatore duale della divergenza discreta, il flusso discreto $\widetilde{\nabla}_h = -\operatorname{div}_h^*$.

Il prodotto interno di cui sono dotati gli spazi $\mathscr{F}_h$ e $\mathscr{P}_h$ si può rappresentare con una matrice simmetrica e definita positiva, indicata con $\mathbf{M}_{\mathscr{F}}$ per il primo e $\mathbf{M}_{\mathscr{P}}$ per il secondo, nel modo seguente:

$$\begin{aligned} [\mathbf{u}_h,\, \mathbf{v}_h]_{\mathscr{F}_h} &= \mathbf{u}_h^T\, \mathbf{M}_{\mathscr{F}}\, \mathbf{v}_h \\ [p_h,\, q_h]_{\mathscr{P}_h} &= p_h^T\, \mathbf{M}_{\mathscr{P}}\, q_h. \end{aligned} \tag{3.13}$$

Utilizzando nella (3.12) la definizione dei prodotti interni (3.13) si ottiene:

$$\begin{aligned} \mathbf{v}_h^T\, \mathrm{div}_h^T\, \mathbf{M}_{\mathscr{P}}\, p_h = &- \mathbf{v}_h^T\, \mathbf{M}_{\mathscr{F}}\, \widetilde{\nabla}_h\, p_h + \sum_{f \in \Gamma_h} |f|\, [\![p_f]\!]\, \{v_f\} + \\ &+ \sum_{f \in \Gamma_h} |f|\, \{p_f\}\, [\![v_f]\!] \quad \forall\, \mathbf{v}_h \in \mathscr{F}_h \text{ e } \forall\, p_h \in \mathscr{P}_h \end{aligned}$$

dove div_h^T è l'operatore trasposto di div_h.

3.2.4 I prodotti interni mimetici

A questo punto il cuore del discorso riguarda la costruzione esplicita dei prodotti interni mimetici. Per prima cosa si definirà il prodotto interno locale dello spazio $\mathscr{F}_{h,P}$ e quindi si procederà alla costruzione della matrice locale $\mathbf{M}_P$:

$$[\mathbf{u}_P, \mathbf{v}_P]_P = \mathbf{u}_P^T\, \mathbf{M}_P\, \mathbf{v}_P.$$

Solo alla fine della sezione si indicherà il prodotto globale banalmente derivato da quello locale. Per definire il prodotto mimetico dello spazio $\mathscr{P}_h$ non si incontreranno particolari difficoltà e verrà riportato insieme all'altro prodotto globale.

È opportuno scegliere un prodotto interno che soddisfi le condizioni di consistenza e di stabilità, così da avere un metodo convergente. Riformulando queste condizioni sarà possibile ricavare le matrici e proseguire nella discretizzazione del problema.
Innanzitutto si introduce una breve descrizione teorica di queste due condizioni che successivamente saranno reinterpretate negli opportuni spazi mimetici.
Con $\mathscr{S}_h$ si indica uno qualsiasi degli spazi mimetici introdotti, con X uno degli spazi continui di interesse e con $\Pi_P^{\mathscr{S}} : X_{|P} \to \mathscr{S}_{h,P}$ l'operatore di proiezione ristretto all'elemento P. $\mathscr{T}_P$ rappresenta lo spazio delle funzioni test e dal momento che si vogliono costruire schemi di ordine basso è composto da funzioni scalari o vettoriali costanti a tratti. Si introduce inoltre lo spazio S_P caratterizzato dalle seguenti proprietà:

1. L'operatore di proiezione locale $\Pi_P^{\mathscr{S}}$ è suriettivo da S_P a $\mathscr{S}_h$.

2. Lo spazio delle funzioni test $\mathscr{T}_P$ è contenuto in S_P.

3. Siano $q \in \mathscr{T}_P$ e $v \in S_P$, allora l'integrale $\int_P q\, v\, dV$ può essere calcolato esattamente usando i gradi di libertà, ossia le componenti dei vettori $\Pi_P^{\mathscr{L}}(v)$ e $\Pi_P^{\mathscr{L}}(q)$.

La prima di queste condizioni afferma che lo spazio S_P deve essere sufficientemente ricco, la seconda è connessa all'accuratezza dello schema mimetico e la terza è legata al problema in esame e permetterà di scrivere in modo algebrico la condizione di consistenza.

Innanzitutto si costruiscono i prodotti interni locali che permetteranno poi la definizione di quelli globali. In questo caso di spazio generico il prodotto interno globale è definito a partire da quello locale nel seguente modo:

$$[u_h,\, v_h]_{\mathscr{S}_h} = \sum_{P \in \Omega_h} [u_{h,P},\, v_{h,P}]_{\mathscr{S}_{h,P}} \qquad \forall\, u_h,\, v_h \in \mathscr{S}_h$$

dove $u_{h,P}$ e $v_{h,P}$ sono le restrizioni di u_h e v_h all'elemento P.

Definizione 3.2.1 (**Condizione di consistenza**)**.** Il prodotto interno mimetico locale soddisfa la condizione di consistenza se vale la seguente:

$$[\Pi_P^{\mathscr{L}}(q),\, \Pi_P^{\mathscr{L}}(\nu)]_{\mathscr{S}_{h,P}} = \int_P q\, \nu\, dV \qquad \forall\, q \in \mathscr{T}_P,\ \forall\, \nu \in S_P.$$

Per una cella poliedrica può capitare che lo spazio di tutti i vettori $\Pi_P^{\mathscr{L}}(q)$ sia più piccolo dello spazio $\mathscr{S}_{h,P}$. Se ciò si verifica la condizione di consistenza non definisce il prodotto interno in maniera univoca e per evitare instabilità è opportuno introdurre un'ulteriore condizione:

Definizione 3.2.2 (**Condizione di stabilità**)**.** Il prodotto interno mimetico soddisfa la condizione di stabilità se esistono due costanti positive C_1 e C_2 indipendenti da P e da $v_{h,P}$ tali che:

$$C_1\, |P|\, ||v_{h,P}||^2 \leq [v_{h,P},\, v_{h,P}]_{\mathscr{S}_{h,P}} \leq C_2\, |P|\, ||v_{h,P}||^2 \qquad \forall\, v_{h,P} \in \mathscr{S}_{h,P}.$$

Dopo queste nozioni puramente generali si riadattano le definizioni introdotte agli spazi utilizzati per il problema in considerazione. La condizione di stabilità assume la seguente forma:

Definizione 3.2.3 (S)**.** Esistono due costanti positive C_1 e C_2, indipendenti dalla dimensione della mesh h, tali che per ogni elemento P:

$$C_1\, |P| \sum_{f \in \partial P} |v_f|^2 \leq [\mathbf{v}_P,\, \mathbf{v}_P]_P \leq C_2\, |P| \sum_{f \in \partial P} |v_f|^2 \qquad \forall\, \mathbf{v}_P \in \mathscr{F}_{h,P}.$$

Tale condizione afferma che il prodotto interno $[\cdot,\cdot]_P$ è coercivo, ossia che è nullo se e solo se $\mathbf{v}_P = 0$. Inoltre il limite superiore ed inferiore mostrano che il prodotto scala come $|P|$: ciò è piuttosto intuitivo poiché tale prodotto scalare approssima un integrale di volume su P.
Prima di specificare la condizione di consistenza al caso in esame è opportuno introdurre il seguente spazio:

$$S_P = \{\, \mathbf{v} \in (L^s(P))^d,\ s > 2,\ \mathrm{div}\,\mathbf{v} = cost,\ \mathbf{v}\cdot\mathbf{n}_f = cost \quad \forall\, f \in \partial P\}.$$

Si può dimostrare che S_P soddisfa le condizioni richieste, a tal proposito si consulti [4]. La condizione di consistenza assume allora la seguente forma:

Definizione 3.2.4 (C)**.** Per ogni funzione vettoriale $\mathbf{v} \in S_P$, per ogni polinomio lineare q e per ogni elemento P vale che:

$$\left[\Pi_P^{\mathscr{F}}(\,\mathbf{K}_P\nabla q\,),\,\Pi_P^{\mathscr{F}}(\mathbf{v})\right]_P = \int_P \mathbf{K}_P^{-1}(\,\mathbf{K}_P\,\nabla q\,)\cdot\mathbf{v}\,dV. \tag{3.14}$$

$\mathbf{K}_P$ è l'approssimazione del tensore di diffusione $\mathbf{K}$ sull'elemento P ed è definito come:

$$\mathbf{K}_P = \frac{1}{|P|}\int_P \mathbf{K}\,dV.$$

Se $\mathbf{K}$ è sufficientemente regolare spesso si preferisce approssimarlo con il suo valore calcolato nel baricentro della cella:

$$\mathbf{K}_P = \mathbf{K}(\mathbf{x}_P).$$

In effetti bisogna assicurarsi che $\mathbf{K}_P$ sia una approssimazione al primo ordine di $\mathbf{K}$, più precisamente che valga:

$$||\mathbf{K}_P - \mathbf{K}||_{L^\infty(P)} \le C\,h_P \qquad \forall\, P \in \Omega_h \tag{3.15}$$

dove h_P è il diametro del poligono P.
Se $\mathbf{K}$ è Lipschitz e continuo allora la validità della (3.15) è assicurata.
Si noti che il prodotto interno nello spazio dei flussi è definito con il peso tensoriale $\mathbf{K}_P^{-1}$.
Le definizioni degli operatori di proiezione unite a quella della divergenza discreta forniscono il seguente risultato:

Lemma 3.2.4. *Sia* $X(\Omega) = \left\{\mathbf{v} \in (L^s(\Omega_\Gamma))^d,\ s \ge 2,\ \mathrm{div}\,\mathbf{v} \in L^2(\Omega_\Gamma)\right\}$*, per ogni* $\mathbf{v} \in X$ *vale:*

$$\Pi^{\mathscr{P}}\left(\mathrm{div}\,(\mathbf{v})\right) = \mathrm{div}_h\,(\Pi^{\mathscr{F}}\,\mathbf{v}).$$

La condizione di consistenza fornisce la proprietà di accuratezza del metodo e per renderla utile occorre modificare il termine di destra della (3.14) in modo che sia indipendente dal valore di $\mathbf{v}$ nell'elemento P. Integrando

per parti e utilizzando le proprietà dello spazio S_P si possono effettuare i seguenti passaggi:

$$\begin{aligned}\int_P \mathbf{K}_P^{-1}(\mathbf{K}_P\,\nabla q)\cdot\mathbf{v}\,dV \;&=\; \int_P \nabla q\cdot\mathbf{v}\,dV \;=\\ &=\; -\int_P q\,\mathrm{div}\,\mathbf{v}\,dV \;+\; \sum_{f\in\partial P}\int_f \mathbf{v}\cdot\mathbf{n}_{P,f}\,q\,dS \;=\\ &=\; -\mathrm{div}_P\big(\Pi_P^{\mathscr{F}}(\mathbf{v})\big)\int_P q\,dV \;+\; \sum_{f\in\partial P}\alpha_{P,f}\,v_f\int_f q\,dS\end{aligned} \tag{3.16}$$

dove $\mathrm{div}_P\left(\Pi_P^{\mathscr{F}}(\mathbf{v})\right) \;=\; (\mathrm{div}\,\mathbf{v})_{|P} \;=\; \textit{cost}$ e $\alpha_{P,f}\,v_f = \mathbf{v}\cdot\mathbf{n}_{P,f} = \textit{cost}$.
Le componenti normali di $\mathbf{v}$ sulle facce f sono tutto ciò che serve per calcolare l'integrale. Tali valori sono i gradi di libertà dello schema numerico a cui si sta facendo riferimento. Questi concetti svolgeranno un ruolo cruciale nei passaggi successivi.

Definite nel dettaglio le due condizioni si può iniziare a costruire il prodotto interno, quello locale si può rappresentare mediante una matrice nel seguente modo:

$$[\mathbf{u}_P,\,\mathbf{v}_P]_P = \mathbf{u}_P^T\,\mathbf{M}_P\,\mathbf{v}_P.$$

Si applicherà ora la condizione di consistenza, scegliendo $q\in\mathbb{P}^1(P)/\mathbb{R}$. Avere scelto q a media nulla non è restrittivo, infatti nel caso q pari a costante si ritrova la definizione di divergenza discreta. In questo modo l'integrale di volume della (3.16) si semplifica e la condizione di consistenza si riduce a:

$$\left[\Pi_P^{\mathscr{F}}\left(\mathbf{K}_P\,\nabla q\right),\,\Pi_P^{\mathscr{F}}(\mathbf{v})\right]_P \;=\; \sum_{f\in\partial P}\alpha_{P,f}\,v_f\int_f q\,dS. \tag{3.17}$$

Si considerino allora le seguenti funzioni polinomiali:

$$q^1(x,y) = x - x_P \qquad \text{e} \qquad q^2(x,y) = y - y_P,$$

dove $\mathbf{x}_P = (x_P, y_P)$ è il baricentro della faccia P. Esse costituiscono una base per $\mathbb{P}_1(P)/\mathbb{R}$.
Si introduca il vettore:

$$N_j = \Pi_P^{\mathscr{F}}(\,\mathbf{K}_P\,\nabla q^j\,), \qquad j = 1,2.$$

Se si numerano le facce di P da 1 a $N_P^{\mathscr{F}}$ è possibile esplicitare l'i-esima componente del vettore N_j, $j=1,2$:

$$(N_j)_i \;=\; \frac{1}{|f_i|}\int_{f_i}\mathbf{n}_{f_i}\cdot\mathbf{K}_P\nabla\,q^j\,dS \;=\; \mathbf{n}_{f_i}^T\,\mathbf{K}_P\cdot\nabla\,q^j \;=\; \mathbf{n}_{f_i}^T\,\mathbf{K}_P\cdot\mathbf{e}_j$$

dove $\mathbf{e}_j$ è il versore unitario associato alla coordinata j e $i = 1,\cdots,N_P^{\mathscr{F}}$.
Si definisce così la matrice di dimensione $N_P^{\mathscr{F}}\times 2$ tale che $\mathbf{N}_P = [\mathbf{N}_1,\,\mathbf{N}_2]$.

Poiché ∇q^j è un vettore di due dimensioni con un uno al j-esimo posto e zero altrove, la i-esima riga della matrice $\mathbf{N}_P$ è data da $\mathbf{n}_{f_i}^T \mathbf{K}_P$ ed in complesso risulta:

$$\mathbf{N}_P = \begin{bmatrix} \mathbf{n}_{f_1}^T \\ \mathbf{n}_{f_2}^T \\ \vdots \\ \mathbf{n}_{f_{N_P^{\mathscr{F}}}}^T \end{bmatrix} \mathbf{K}_P.$$

Di conseguenza è possibile riformulare la (3.17) come:

$$\left[\Pi_P^{\mathscr{F}} \left(\mathbf{K}_P \nabla q^j \right), \Pi_P^{\mathscr{F}}(\mathbf{v}) \right]_P = \left(\Pi_P^{\mathscr{F}}(\mathbf{v}) \right)^T \mathbf{M}_P \mathbf{N}_j = \left(\Pi_P^{\mathscr{F}}(\mathbf{v}) \right)^T \mathbf{R}_j$$

dove $\mathbf{R}_j$ è definita nel seguente modo:

$$(\mathbf{R}_j)_i = \alpha_{P,f_i} \int_{f_i} q^j \, dS.$$

Tale relazione deve valere per qualsiasi campo vettoriale discreto $\Pi_P^{\mathscr{F}}(\mathbf{v})$. Il vettore $\mathbf{R}_j$ dipende da q^j e dalla geometria della cella P e analogamente a $\mathbf{N}_P$ si può definire $\mathbf{R}_P = [\mathbf{R}_1, \mathbf{R}_2]$. Dato che $\Pi_P^{\mathscr{F}}(\mathbf{v})$ è arbitrario si ottengono le seguenti due relazioni:

$$\mathbf{M}_p \mathbf{N}_j = \mathbf{R}_j, \qquad j = 1, 2$$

che in forma compatta diventano:

$$\mathbf{M}_p \mathbf{N}_P = \mathbf{R}_P. \tag{3.18}$$

Quindi dalla condizione di consistenza è stata ottenuta una relazione algebrica; di fondamentale importanza ora è attribuire a $\mathbf{R}_P$ una formula esplicita e soprattutto implementabile. Nella definizione di $(\mathbf{R}_j)_i$ si richiede il calcolo dell'integrale di una funzione lineare, che equivale al valore della funzione nel baricentro moltiplicato per l'area della faccia; quindi $\mathbf{R}_P$ diventa:

$$\mathbf{R}_P = \begin{bmatrix} \alpha_{P,f_1} |f_1| (\mathbf{x}_{f_1} - \mathbf{x}_P)^T \\ \alpha_{P,f_2} |f_2| (\mathbf{x}_{f_2} - \mathbf{x}_P)^T \\ \vdots \\ \alpha_{P,f_{N_P^{\mathscr{F}}}} |f_{N_P^{\mathscr{F}}}| (\mathbf{x}_{f_{N_P^{\mathscr{F}}}} - \mathbf{x}_P)^T \end{bmatrix}.$$

Vale il seguente risultato:

Lemma 3.2.5. *Per ogni cella poliedrica $|P|$ si ha che:*

$$\mathbf{N}_P^T \mathbf{R}_P = \mathbf{K}_P |P|.$$

Per la dimostrazione di tale risultato si rimanda a [4].

Si verifica facilmente che tutte le matrici della forma sotto riportata rispettano la (3.18):

$$\mathbf{M}_P = \mathbf{R}_P \left(\mathbf{R}_P^T \mathbf{N}_P\right)^{-1} \mathbf{R}_P^T + \mathbf{M}_P^{(1)} \tag{3.19}$$

se $\mathbf{M}_P^{(1)}$ è una matrice simmetrica e semi-definita positiva tale che:

$$\ker(\mathbf{M}_P^{(1)}) = \mathrm{img}(\mathbf{N}_P).$$

La scelta di questa matrice non è unica, infatti la formula (3.19) rappresenta una famiglia di matrici e quindi di schemi numerici. Una possibilità effettiva per tale matrice può essere il proiettore ortogonale scalato definito nel seguente modo:

$$\mathbf{M}_P^{(1)} = \gamma_P \left(I - \mathbf{N}_P \left(\mathbf{N}_P^T \mathbf{N}_P\right)^{-1}\right) \mathbf{N}_P^T \tag{3.20}$$

dove:

$$\gamma_p = \frac{1}{\mathscr{N}_P^{\mathscr{F}} |P|} Tr(\mathbf{R}_P \mathbf{K}_P^{-1} \mathbf{R}_P^T).$$

Dato che la matrice $\mathbf{R}_P \left(\mathbf{R}_P^T \mathbf{N}_P\right)^{-1} \mathbf{R}_P^T$ è simmetrica per definizione ma non è in generale definita positiva non rispetta la condizione di stabilità. Quindi è stato necessario aggiungere la matrice $\mathbf{M}_P^{(1)}$ per stabilizzare il metodo.

Teorema 3.2.6. *Siano valide le assunzioni sulla mesh (3.2.1); in aggiunta valga la (3.2.2) con $\bar{\mathbf{x}}_P = \mathbf{x}_P$ baricentro di P e $\bar{\mathbf{x}}_f = \mathbf{x}_f$ baricentro della faccia f, per ogni faccia $f \in P$. Allora la matrice del prodotto interno $\mathbf{M}_P$ data da (3.19) e da (3.20) soddisfa la condizione di stabilità (S) con costanti C_1 e C_2 che dipendono solo dalla dimensione del dominio d, dalle costanti di regolarità della mesh che appaiono in (3.2.1) e dalle costanti di ellitticità che limitano lo spettro di $\mathbf{K}_P$.*

Per la dimostrazione di tale risultato si rimanda a [4].

Finora l'analisi si è concentrata sulla costruzione del prodotto interno locale legato al prodotto mimetico dello spazio $\mathscr{F}_h$. A partire da questo è possibile costruire il prodotto interno globale grazie alla seguente relazione:

$$[\mathbf{u}_h, \mathbf{v}_h]_{\mathscr{F}_h} = \sum_{P \in \Omega_h} [\mathbf{u}_P, \mathbf{v}_P]_P.$$

Il prodotto interno dello spazio $\mathscr{P}_h$ è più immediato, infatti ha la seguente forma:

$$[p_h, q_h]_{\mathscr{P}_h} = \sum_{P \in \Omega_h} |P| \, p_P \, q_P.$$

3.3 La discretizzazione

In questa sezione sarà introdotta la discretizzazione mimetica della formulazione debole (2.10). Si ricorda che gli operatori discreti approssimano quelli continui nel seguente modo nel mezzo poroso:

$$\mathrm{div}_h \approx \mathrm{div} \qquad \text{e} \qquad \widetilde{\nabla}_h \approx \mathbf{K}\,\nabla$$

e nel seguente nella frattura:

$$\mathrm{div}_{\boldsymbol{\tau},h} \approx \mathrm{div}_{\boldsymbol{\tau}} \qquad \text{e} \qquad \widetilde{\nabla}_{\boldsymbol{\tau},h} \approx \hat{K}\,\nabla_{\boldsymbol{\tau}}.$$

La formulazione variazionale discreta del problema permette di chiarire come implementare i termini di bordo. Per prima cosa si definiscono gli spazi opportuni:

$$\mathscr{F}_{h,g} = \left\{ \mathbf{v}_h \in \mathscr{F}_h \ : \ v_f = \frac{1}{|f|}\int_f g_{\mathbf{u}}\, dS \quad \forall\, f \in \mathscr{F}^{\mathbf{u}} \right\}$$

dove $\mathscr{F}^{\mathbf{u}} \subset \mathscr{F}$ è l'insieme delle facce contenute in $\partial\Omega_{\mathbf{u}}$. Analogamente si definisce l'insieme delle facce appartenenti a $\partial\Omega_P$ come $\mathscr{F}^P \subset \mathscr{F}$.
Se $g_P \in L^1(\partial\Omega_P)$ si introduce il funzionale lineare:

$$\langle g_P, \mathbf{v}_h \rangle_h = \sum_{f \in \mathscr{F}^P} v_f \int_f g_P\, dS.$$

Inoltre si indica l'insieme delle facce che rappresentano il bordo della Γ con $\mathscr{F}^{\Gamma} \subset \mathscr{F}$.
Avendo a che fare anche nella frattura con una formulazione mista del problema, si impongono le condizioni sulla velocità in maniera forte, costruendo uno spazio che le ingloba e quelle sulla pressione in forma debole. Per affrontare la questione si definisce lo spazio affine:

$$\hat{\mathscr{F}}_{h,g} = \left\{ \hat{\mathbf{u}}_h \in \hat{\mathscr{F}}_h \ : \ \hat{u}_i = \hat{g}_{\mathbf{u}}(x_i) \quad \forall\, x_i \in \hat{\mathscr{F}}^{\mathbf{u}} \right\}$$

dove $\hat{\mathscr{F}}^{\mathbf{u}} \subset \hat{\mathscr{F}}$ è l'insieme dei nodi x_i contenuti in $\partial\Gamma_{\hat{\mathbf{u}}}$. Analogamente si indica con $\hat{\mathscr{F}}^P \in \hat{\mathscr{F}}$ l'insieme dei nodi appartenenti a $\partial\Gamma_{\hat{p}}$.

La formulazione debole (2.10) discretizzata con le Mimetiche diventa:

Trovare $(\mathbf{u}_h,\, p_h) \in \mathscr{F}_{h,g} \times \mathscr{P}_h$ *e* $(\hat{\mathbf{u}}_h,\, \hat{p}_h) \in \hat{\mathscr{F}}_{h,g} \times \hat{\mathscr{P}}_h$ *tali che:*

$$
\begin{aligned}
&\big[\,\mathbf{u}_h,\, \mathbf{v}_h\,\big]_{\mathscr{F}_h} - \big[\,p_h,\, \mathrm{div}_h\, \mathbf{v}_h\,\big]_{\mathscr{P}_h} + \sum_{f\in\mathscr{F}^\Gamma} |f|\,[\![v_f]\!]\,\hat{p}_f + \sum_{f\in\mathscr{F}^\Gamma} \eta\,|f|\,\{u_f\}\,\{v_f\} + \\
&\qquad + \sum_{f\in\mathscr{F}^\Gamma} \xi_0\,\eta\,|f|\,[\![u_f]\!]\,[\![v_f]\!] = -\langle\, g_P,\, \mathbf{v}_h\,\rangle_h \qquad\qquad \forall\, \mathbf{v}_h \in \mathscr{F}_{h,0}, \\
&\big[\,\mathrm{div}_h\, \mathbf{u}_h,\, q_h\,\big]_{\mathscr{P}_h} = \big[\,f^I,\, q_h\,\big]_{\mathscr{P}_h} \qquad\qquad \forall\, q_h \in \mathscr{P}_h, \\
&\big[\,\hat{\mathbf{u}}_h,\, \hat{\mathbf{v}}_h\,\big]_{\hat{\mathscr{F}}_h} - \big[\,\hat{p}_h,\, \mathrm{div}_h\, \hat{\mathbf{v}}_h\,\big]_{\hat{\mathscr{P}}_h} = -\sum_{x_i\in\hat{\mathscr{F}}^P} \hat{g}_P(x_i)\,\hat{\mathbf{v}}_i\cdot\boldsymbol{\tau} \qquad \forall\, \hat{\mathbf{v}}_h \in \hat{\mathscr{F}}_{h,0}, \\
&\big[\,\mathrm{div}_{\boldsymbol{\tau},h}\, \hat{\mathbf{u}}_h,\, \hat{q}_h\,\big]_{\hat{\mathscr{P}}_h} - \sum_{f\in\mathscr{F}^\Gamma} |f|\,[\![u_f]\!]\,\hat{q}_f = \big[\,\hat{f}^I,\, \hat{q}_h\,\big]_{\hat{\mathscr{P}}_h} \qquad \forall\, \hat{q}_h \in \hat{\mathscr{P}}_h
\end{aligned}
\tag{3.21}
$$

dove $f^I = \Pi^{\mathscr{P}}(f)$ e $\hat{f}^I = \Pi^{\hat{\mathscr{P}}}(\hat{f})$ sono le interpolazioni di f e $\hat{f}$ sugli spazi $\mathscr{P}_h$ e $\hat{\mathscr{P}}_h$.
$[\![u_f]\!]$ indica il salto dei gradi di libertà del campo di velocità sulla faccia di frattura f e $\{u_f\}$ la media, come già detto in precedenza. $\hat{p}_f$ è il grado di libertà della pressione nella frattura sulla faccia f.
Si noti che non appare il duale della divergenza, perché è implicitamente definito dalle forme bilineari. Inoltre il problema (3.21) ingloba direttamente le condizioni al bordo e quelle di accoppiamento.

3.4 La formulazione algebrica

Dopo le considerazioni fatte e l'analisi del metodo di discretizzazione applicato al problema in esame si hanno tutti gli elementi per fornirne la formulazione algebrica. L'obiettivo è ricavare dal sistema la pressione e la velocità del fluido nel mezzo poroso e nella frattura; occorre quindi assemblare la matrice globale e il vettore del termine noto. La tecnica adottata per imporre le condizioni al bordo sulla velocità opera a livello algebrico.
La formulazione algebrica di (3.21), prima dell'imposizione delle condizioni sulla velocità, è:

$$
\begin{bmatrix} M_o & B^T & 0 & C^T \\ B & 0 & 0 & 0 \\ 0 & 0 & \hat{M}_o & \hat{B}^T \\ \hat{C} & 0 & \hat{B} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \\ \hat{\mathbf{u}} \\ \hat{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} \mathbf{g}_o \\ -\mathbf{h} \\ \hat{\mathbf{g}}_o \\ -\hat{\mathbf{h}} \end{bmatrix}
\tag{3.22}
$$

dove:

- $\mathbf{u}$ e $\mathbf{p}$ sono i gradi di libertà del problema in velocità e pressione nel mezzo poroso rispettivamente. Con N_F si indica il numero di tutte le

facce della mesh a cui va aggiunto il numero delle facce di frattura per lo sdoppiamento dei gradi di libertà. Con N_C si indica il numero di tutte le celle della mesh.

- $M_o \in \mathbb{R}^{N_F} \times \mathbb{R}^{N_F}$ è la matrice globale del prodotto interno in $\mathscr{F}_h$, assemblata a partire dalle matrici locali $\mathbf{M}_P$, a cui vanno aggiunti alcuni contributi dovuti ai seguenti termini presenti nella formulazione:

$$\sum_{f \in \mathscr{F}^\Gamma} |f| \{u_f\} \{v_f\} + \sum_{f \in \mathscr{F}^\Gamma} \xi_0 |f| [\![u_f]\!] [\![v_f]\!] \tag{3.23}$$

dove $\mathbf{v}_h$ è una generica funzione test.
Più precisamente ciascun termine con le due medie contribuisce ad incrementare quattro elementi della matrice M_o ed analogo è l'effetto di ciascun termine con i due salti attraverso la frattura. Per esempio se si sviluppa quest'ultimo si ottengono i seguenti contributi:

$$[\![u_f]\!] [\![v_f]\!] = \left\{ u_{f+} v_{f+} + u_{f-} v_{f-} - u_{f+} v_{f-} - u_{f-} v_{f+} \right\}$$

dove u_{f+} e u_{f-} sono i due gradi di libertà associati a ciascuna faccia di frattura in virtù dello sdoppiamento del singolo grado di libertà u_f. L'elemento $f1, f2$ della matrice è dato dall'elemento corrispondente della matrice locale, indicato con gli indici i, j e da alcuni contributi da aggiungere in determinati casi:

$$[M_o]_{f1,f2} = \begin{cases} [\mathbf{M}_P]_{ij} & \text{se } f1 \text{ e } f2 \notin \mathscr{F}^{\Gamma^+} \cup \mathscr{F}^{\Gamma^-} \\ [\mathbf{M}_P]_{ij} + \dfrac{\eta |f|}{4} + \eta |f| \xi_0 & \text{se } f1 \in \mathscr{F}^{\Gamma^+} \text{ e } f1 = f2 \\ [\mathbf{M}_P]_{ij} + \dfrac{\eta |f|}{4} - \eta |f| \xi_0 & \text{se } f1 \in \mathscr{F}^{\Gamma^+} \text{ e } f2 = f1 + N_f \\ [\mathbf{M}_P]_{ij} + \dfrac{\eta |f|}{4} - \eta |f| \xi_0 & \text{se } f1 \in \mathscr{F}^{\Gamma^-} \text{ e } f1 = f2 + N_f \\ [\mathbf{M}_P]_{ij} + \dfrac{\eta |f|}{4} + \eta |f| \xi_0 & \text{se } f1 \in \mathscr{F}^{\Gamma^-} \text{ e } f1 = f2 \end{cases}$$

dove N_f è il numero di facce della mesh e $\mathscr{F}^{\Gamma+}$ e $\mathscr{F}^{\Gamma-}$ sono gli insiemi degli identificativi delle facce di frattura appartenenti alle celle + e alle celle - rispettivamente. Sommando N_f all'identificativo di una faccia appartenente a $\mathscr{F}^{\Gamma+}$ si ottiene l'identificativo della faccia corrispondente appartenente alla cella -.

Osservazione 3.4.1. *Le dimensioni della matrice M_o aumentano rispetto all'analogo problema senza frattura, alle "classiche" facce vengono aggiunte un numero di facce pari a quelle che rappresentano la frattura. Analogamente, rispetto al problema senza frattura, si devono modificare anche le matrici B e B^T per quanto riguarda rispettivamente il numero di colonne e righe, come pure il termine noto* $\mathbf{h}$.

- $B \in \mathbb{R}^{N_C} \times \mathbb{R}^{N_F}$ è la matrice che rappresenta l'operatore divergenza ed è assemblata sfruttando la definizione della divergenza discreta (3.11):

$$-[\mathrm{div}_h\, \mathbf{u}_h,\, p_h]_{\mathscr{P}_h} = -\sum_{P\in\Omega_h} |P|\, p_P\, \frac{1}{|P|} \sum_{f\in\mathscr{F}_P} \alpha_{P,f}\, |f|\, \mathbf{u}_P = \\ = -\sum_{P\in\Omega_h} p_P \sum_{f\in\mathscr{F}_P} \alpha_{P,f}\, |f|\, \mathbf{u}_P = \mathbf{p}^T\, B\, \mathbf{u}.$$

- $B^T \in \mathbb{R}^{N_F} \times \mathbb{R}^{N_C}$ è la trasposta della precedente.

- $\hat{\mathbf{u}}$ e $\hat{\mathbf{p}}$ sono i gradi di libertà in velocità e pressione nella frattura.

- La matrice $\hat{M}_o$ è la matrice di massa del problema nella frattura e deriva dal termine $[\hat{\mathbf{u}}_h, \hat{\mathbf{v}}_h]_{\hat{\mathscr{F}}_h}$ della formulazione. Si tratta di una matrice tridiagonale costituita da valori che dipendono da alcuni parametri fisici del problema. E' inoltre simmetrica e ha le stesse proprietà della matrice di massa degli Elementi Finiti 1-D.

- La matrice $\hat{B}$ è legata al termine $-[\mathrm{div}_{\boldsymbol{\tau},h}\, \hat{\mathbf{u}}_h,\, \hat{q}_h]_{\hat{\mathscr{P}}_h}$ ed è assemblata sfruttando la seguente definizione di divergenza discreta nella frattura:

$$(\mathrm{div}_{\boldsymbol{\tau},h}\, \hat{\mathbf{u}}_h)_i = \frac{\hat{u}_{i+1} - \hat{u}_i}{|f|}$$

  dove $|f|$ rappresenta la lunghezza della faccia in considerazione.
  Inserendo nella formulazione debole tale definizione si giunge alla semplificazione dei termini con la divergenza:

$$[\mathrm{div}_{\boldsymbol{\tau},h}\, \hat{\mathbf{u}}_h,\, \hat{q}_h]_{\hat{\mathscr{P}}_h} = \sum_{f\in\Gamma_h} |f|\, \mathrm{div}_{\boldsymbol{\tau},h}\, \hat{\mathbf{u}}_h\, \hat{q}_h = \sum_{f\in\Gamma_h} (\hat{u}_{i+1} - \hat{u}_i)\, \hat{q}_h.$$

  Algebricamente ciò si traduce in una matrice divergenza nella frattura composta da -1 e 1; si tratta quindi di una matrice molto semplice, costituita da valori che non dipendono dai parametri fisici del problema:

$$\hat{B}_{ij} = \begin{cases} 1 & \text{se } i = j \\ -1 & \text{se } j = i+1 \\ 0 & \text{altrimenti.} \end{cases}$$

- La matrice $\hat{C}$ è legata al termine di salto presente nell'equazione di conservazione della massa nella frattura, più precisamente deriva dal seguente termine della formulazione debole discretizzata con le Mimetiche:

$$\sum_{f\in\mathscr{F}^\Gamma} |f|\, [\![u_f]\!]\, \hat{q}_f = \sum_{f\in\mathscr{F}^\Gamma} |f|\, (u_{f+} - u_{f-})\, \hat{q}_f$$

dove $\hat{q}_h$ è una generica funzione test appartenente a $\hat{\mathscr{P}}_h$.
Più precisamente indicando con Id_f l'identificativo della faccia come elemento della mesh, con IdF_f l'identificativo della faccia come elemento della frattura e con N_f il numero totale delle facce della mesh si ha che:

$$\hat{C}_{ij} = \begin{cases} |f| & \text{se } i = IdF_f,\ j = Id_f \\ -|f| & \text{se } i = IdF_f,\ j = N_f + Id_f \\ 0 & \text{altrimenti.} \end{cases}$$

- C^T è legata al termine presente nella formulazione debole derivante dalla sostituzione delle condizioni di accoppiamento:

$$\sum_{f \in \mathscr{F}^\Gamma} |f| \, [\![v_f]\!] \, \hat{p}_f,$$

che fornisce due contributi per ciascuna faccia di frattura.
Più precisamente la matrice assume la seguente forma:

$$C^T_{ij} = \begin{cases} |f| & \text{se } i = Id_f,\ j = IdF_f \\ -|f| & \text{se } i = N_f + Id_f,\ j = IdF_f \\ 0 & \text{altrimenti.} \end{cases}$$

Si nota che nel caso di imposizione in forma debole delle condizioni di accoppiamento $\hat{C}$ ha esattamente come trasposta C^T.

- Il termine noto del sistema (3.22) è composto dal vettore $-\mathbf{h}$, legato all'interpolazione della forzante f sullo spazio mimetico $\mathscr{P}_h$ e da $\mathbf{g}_o$, legato all'imposizione delle condizioni al bordo sulla pressione:

$$[\mathbf{g}_o]_i = \begin{cases} -\int_{f_i} g_P & \text{se } f_i \in \mathscr{F}^p \\ 0 & \text{altrimenti.} \end{cases}$$

- La parte di termine noto $\hat{\mathbf{g}}_o$ consiste in termini non nulli nel caso di condizioni sulla pressione non omogenee. Supponendo di avere ad entrambi i vertici della frattura condizioni sulla pressione il vettore $\hat{\mathbf{g}}_o$ è dato da:

$$\hat{\mathbf{g}}_o = \begin{pmatrix} \hat{g}_p(x_1) \\ 0 \\ \vdots \\ 0 \\ -\hat{g}_p(x_{n+1}) \end{pmatrix}$$

ipotizzando che il vettore tangente alla frattura sia orientato dal primo all'ultimo vertice della frattura stessa.

- La parte di termine noto $-\hat{\mathbf{h}}$, come anche $-\mathbf{h}$, è preceduta dal segno meno semplicemente perché è stato cambiato il segno dell'equazione della divergenza in modo che il problema abbia una struttura simmetrica. Tale vettore è il risultato dell'interpolazione della forzante $\hat{f}$ nello spazio mimetico $\hat{\mathscr{P}}_h$. Dal punto di vista pratico si calcola l'interpolazione della funzione su ciascuna cella della partizione sulla frattura e si compone così questa parte del vettore termine noto.

Nel mezzo poroso le condizioni al bordo sulla velocità vengono imposte in maniera forte, modificando M_o, B^T e $\mathbf{g}_o$. La tecnica utilizzata per l'imposizione di una condizione sulla velocità prevede la modifica delle righe di M_o corrispondenti alle facce di bordo su cui vale tale condizione nel seguente modo:

$$M_{ij} = \begin{cases} M_{o,ij} & \text{se } f_i \notin \mathscr{F}^{\mathbf{u}} \\ 0 & \text{se } f_i \in \mathscr{F}^{\mathbf{u}} \text{ e } i \neq j \\ 1 & \text{se } f_i \in \mathscr{F}^{\mathbf{u}} \text{ e } i = j. \end{cases}$$

Nel caso in cui $f_i \in \mathscr{F}^{\mathbf{u}}$ e $i = j$ si potrebbe porre l'elemento M_{ij} pari ad un numero diverso da 1 per evitare di sbilanciare troppo la matrice.
Inoltre è necessario annullare le righe di B^T corrispondenti alle facce di bordo su cui vale una condizione sulla velocità:

$$D_{ij} = \begin{cases} B^T_{ij} & \text{se } f_i \notin \mathscr{F}^{\mathbf{u}} \\ 0 & \text{se } f_i \in \mathscr{F}^{\mathbf{u}}. \end{cases}$$

Quindi si ottengono due nuove matrici, M non simmetrica e D coincidente con la trasposta di B se e solo se vengono imposte solo condizioni sulla pressione. Si noti che il vettore dei gradi di libertà $\mathbf{u}$ rappresenta la velocità su tutte le facce della mesh.
La parte di termine noto $\mathbf{g}$ è invece caratterizzata nel seguente modo:

$$\mathbf{g}_i = \begin{cases} g_{\mathbf{u},i} & \text{se } f_i \in \mathscr{F}^{\mathbf{u}} \\ -\int_{f_i} g_P & \text{se } f_i \in \mathscr{F}^{P} \\ 0 & \text{altrimenti} \end{cases} \tag{3.24}$$

dove $g_{\mathbf{u},i}$ rappresenta $g_{\mathbf{u}}$ valutata al centro della faccia f_i.

Per l'imposizione delle condizioni al bordo della frattura si utilizza un approccio del tutto analogo a quello del mezzo poroso. Le condizioni sulla pressione non determinano alcun cambiamento nelle matrici del sistema, quelle sulla velocità invece sono imposte in maniera forte, modificando le righe delle matrici $\hat{M}_o$ e $\hat{B}^T$ e del termine $\hat{\mathbf{g}}_o$. La matrice di massa della

frattura diventa:

$$\hat{M}_{ij} = \begin{cases} \hat{M}_{o,ij} & \text{se } x_i \notin \hat{\mathscr{F}}^{\mathbf{u}} \\ 0 & \text{se } x_i \in \hat{\mathscr{F}}^{\mathbf{u}} \text{ e } i \neq j \\ 1 & \text{se } x_i \in \hat{\mathscr{F}}^{\mathbf{u}} \text{ e } i = j. \end{cases}$$

Inoltre vengono annullate le righe di $\hat{B}^T$ corrispondenti ai nodi di bordo su cui vale una condizione sulla velocità:

$$\hat{D}_{ij} = \begin{cases} \hat{B}^T_{ij} & \text{se } x_i \notin \hat{\mathscr{F}}^{\mathbf{u}} \\ 0 & \text{se } x_i \in \hat{\mathscr{F}}^{\mathbf{u}}. \end{cases}$$

Si ottengono due nuove matrici, $\hat{M}$ non simmetrica e $\hat{D}$ coincidente con la trasposta di $\hat{B}$ se e solo se vengono imposte solo condizioni sulla pressione. Si definisce con $\hat{\mathbf{g}}$ la parte di termine noto corrispondente a $\hat{\mathbf{g}}_o$ dopo l'imposizione delle condizioni sulla velocità, qualora ci fossero. Nel caso di condizioni sulla velocità non omogenee al bordo della frattura $\hat{\mathbf{g}}$ assume la seguente forma:

$$\hat{\mathbf{g}} = \begin{pmatrix} \hat{g}_{\mathbf{u}}(x_1) \\ 0 \\ \vdots \\ 0 \\ \hat{g}_{\mathbf{u}}(x_{n+1}) \end{pmatrix}.$$

Osservazione 3.4.2. *Nel caso di intersezioni tra fratture sarà necessario uno sdoppiamento dei gradi di libertà riguardanti la velocità nella frattura, se ne discuterà nelle sezioni successive.*

Quindi la formulazione algebrica di (3.22) dopo l'imposizione delle condizioni al bordo sulla velocità, se presenti, diventa:

$$\begin{bmatrix} M & D & 0 & C^T \\ B & 0 & 0 & 0 \\ 0 & 0 & \hat{M} & \hat{D} \\ \hat{C} & 0 & \hat{B} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \\ \hat{\mathbf{u}} \\ \hat{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ -\mathbf{h} \\ \hat{\mathbf{g}} \\ -\hat{\mathbf{h}} \end{bmatrix}. \tag{3.25}$$

Si noti che la struttura a blocchi di tale sistema sarà valida anche nel caso di un network di fratture.
Nei test numerici è stato risolto il problema monolitico con un metodo diretto, ma se le dimensioni crescono notevolmente può rivelarsi necessario l'utilizzo di un metodo iterativo. Nel caso di uso di metodi iterativi potrebbe essere utile assemblare ciascuna delle singole sotto-matrici indipendentemente, di fronte alla necessità di precondizionare il sistema.

Accoppiamento in forma forte

Finora è stato considerato l'accoppiamento in forma debole dei due problemi nel mezzo e nella frattura. Ciò ha determinato la presenza di termini aggiuntivi nella formulazione debole. Se invece si imponessero le condizioni di accoppiamento in forma forte, nella formulazione debole si trascurerebbero i termini aggiuntivi e si imporrebbero le seguenti direttamente sulla matrice globale del problema:

$$\begin{aligned} [\,\xi\,\mathbf{u}^+\cdot\mathbf{n}^+ - (1-\xi)\,\mathbf{u}^-\cdot\mathbf{n}^-\,] &= (p^+ - \hat{p}) \quad \text{su } \Gamma \\ \eta\,[\,(1-\xi)\,\mathbf{u}^+\cdot\mathbf{n}^+ - \xi\,\mathbf{u}^-\cdot\mathbf{n}^-\,] &= (\hat{p} - p^-) \quad \text{su } \Gamma. \end{aligned} \tag{3.26}$$

Il sistema diventerebbe il seguente:

$$\begin{bmatrix} \widetilde{M} & \widetilde{D} & 0 & C^T \\ B & 0 & 0 & 0 \\ 0 & 0 & \hat{M} & \hat{D} \\ \hat{C} & 0 & \hat{B} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \\ \hat{\mathbf{u}} \\ \hat{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ -\mathbf{h} \\ \hat{\mathbf{g}} \\ -\hat{\mathbf{h}} \end{bmatrix}$$

dove:

- $\widetilde{M}$ è la matrice globale del prodotto interno nello spazio $\mathscr{F}_h$ in cui le righe corrispondenti alle facce della frattura vengono cancellate e sostituite da termini opportuni per riprodurre correttamente le condizioni (3.26). I termini a sinistra dell'uguale influenzeranno questa matrice.
- $\widetilde{D}$ corrisponde alla matrice D precedentemente introdotta in cui vengono cancellate le righe della matrice corrispondenti alle facce della frattura e sostituite da termini opportuni per riprodurre correttamente le condizioni (3.26). Più precisamente in questa matrice sarà riprodotto il contributo di p_+ e di p_-.
- C^T è legata esclusivamente all'applicazione delle condizioni di accoppiamento, in questo caso conterrà il contributo determinato da $\hat{p}$ presente nelle (3.26).
  Con l'imposizione delle condizioni di accoppiamento in maniera forte C^T non coincide con la trasposta di $\hat{C}$.

Nei test numerici effettuati si è rilevato che l'accoppiamento debole, oltre ad essere più coerente con la formulazione debole del problema, è più efficace di quello in forma forte.

3.5 Intersezioni tra fratture

Finora il network di fratture è stato considerato solo da un punto di vista teorico, sarà opportuno fornire anche dei dettagli utili per lo sviluppo del

codice.
Innanzitutto l'intersezione coincide sicuramente con un vertice della mesh per come questa viene costruita. Il grado di libertà associato a questo vertice viene inizialmente sdoppiato, definendo un grado di libertà differente per ciascuna frattura coinvolta nell'intersezione. Ciò non è sufficiente, infatti in prossimità di una intersezione il grado di libertà di ciascuna frattura viene ulteriormente sdoppiato per rappresentare il flusso "spazialmente prima e dopo" l'intersezione. In corrispondenza di un vertice in cui si incontrano due fratture si hanno quindi quattro gradi di libertà, due per frattura.
Per imporre la conservazione del flusso nel punto di intersezione i a livello implementativo si utilizza una strategia differente da quella teorica, infatti è particolarmente difficile riuscire a realizzare uno spazio come $\hat{V}$, definito nella (2.29). Occorre dunque procedere in maniera differente. Si aggiunge l'equazione di conservazione del flusso a quelle del modello per ciascuna intersezione $i \in I$ e p_i diventa il moltiplicatore di Lagrange di questo vincolo. Di conseguenza la matrice globale del sistema ha una riga in più per ciascuna intersezione presente. Anche il vettore delle incognite presenterà alcune righe aggiuntive pari al numero delle intersezioni. Nella formulazione debole occorre considerare però il contributo legato ai punti di intersezione, quindi comparirà per ciascuna intersezione $i \in I$ un termine del tipo:

$$\sum_{k:\, \gamma_k \in S_i} \hat{\mathbf{v}}_k \cdot \boldsymbol{\tau}_k \hat{p}_i \,|_i.$$

Ciò corrisponde a considerare la condizione di continuità della pressione imposta debolmente, come se fosse una condizione al bordo sulla pressione che si richiede sia pari a p_i su ciascuno dei gradi di libertà all'intersezione. Ciò si riflette a livello algebrico nell'aggiunta, per ciascuna intersezione, di una colonna che coincide con la trasposta della riga aggiunta per rappresentare il vincolo sui flussi alla stessa intersezione.

In questo capitolo sono state analizzate le idee alla base dei metodi mimetici nel caso monodimensionale e bidimensionale, in particolare applicate al problema di Darcy. Si è introdotta una opportuna discretizzazione che ha permesso di esprimere il problema (1.9) in forma algebrica. Sono stati forniti anche dettagli implementativi, in particolare riguardo al problema nel mezzo, alle condizioni di accoppiamento e alla presenza di intersezioni tra fratture. Quest'ultimo aspetto verrà approfondito maggiormente nel capitolo successivo.

Capitolo 4

Dettagli implementativi

In questo capitolo si forniranno i dettagli implementativi riguardanti il codice che ha permesso di sviluppare il modello numerico descritto nei precedenti capitoli e quindi di calcolare pressione e velocità del fluido nel mezzo poroso e nelle fratture. In particolare si analizzeranno la struttura del codice, le tecniche di implementazione, le librerie utilizzate e gli aspetti più interessanti legati alla sintassi del C++11.

Le novità introdotte in questa tesi rispetto al codice preesistente sono:

- La discretizzazione mediante le Differenze Finite Mimetiche nelle fratture.
- Una classe che rappresenta le fratture.
- L'imposizione di opportune condizioni nei punti di intersezione tra fratture con la nuova discretizzazione mediante le Differenze Finite Mimetiche.
- La generalizzazione delle condizioni al bordo, essendo necessario imporle anche agli estremi delle fratture.
- La lettura da file di tutte le informazioni in input riguardanti il mezzo poroso e le fratture.
- Il calcolo degli errori, oltre che della pressione nel mezzo poroso, anche del flusso e della pressione nelle fratture.

Il codice è piuttosto articolato ed è stato sviluppato partendo dal lavoro di L. Turconi per la creazione della mesh, di F. Della Porta per lo sviluppo del solutore a Volumi Finiti e di N. Verzotti per l'implementazione delle Differenze Finite Mimetiche nel mezzo poroso. La prima versione effettuava la discretizzazione utilizzando i Volumi Finiti, successivamente sono state introdotte le Differenze Finite Mimetiche nel mezzo poroso e ora queste saranno estese

anche alle fratture.
Si analizzeranno in particolare la struttura della mesh creata nel dominio, le classi che permettono l'assemblaggio delle matrici, i dettagli legati all'imposizione delle condizioni al bordo e di accoppiamento e la gestione delle intersezioni. Oltre a questi aspetti particolarmente interessanti si accennerà al metodo di risoluzione del sistema ottenuto e al calcolo degli errori.
Hanno svolto un ruolo importante Paraview e Matlab, rispettivamente per visualizzare la soluzione e rappresentare l'andamento degli errori.

4.1 La mesh

Il primo aspetto di cui ci si occupa è la costruzione della mesh mediante la classe template Rigid_Mesh<T>. Tale classe prende spunto dalla libreria Mesh2D che è stata costruita utilizzando la libreria CGAL. L'introduzione della classe Rigid_Mesh è dovuta all'inadeguatezza della libreria Mesh2D ad assemblare velocemente una matrice. Infatti Mesh2D permette di creare una griglia che può essere raffinata e modificata ma le sue strutture dati non sono efficienti nell'assemblare una matrice. Di conseguenza è stata creata la classe Rigid_Mesh, che come suggerisce il nome, presenta caratteristiche opposte alla Mesh2D, è quindi particolarmente efficace nella costruzione delle matrici, ma poco flessibile sulla struttura della mesh che non può essere modificata una volta creata.

La classe Rigid_Mesh è stata realizzata con la prospettiva di generalizzare il codice al caso 3-D, di conseguenza è una classe template il cui parametro riguarda la dimensione. Più precisamente il parametro template appartiene alla classe template Geometry::Dimension il cui parametro N rappresenta le dimensioni del dominio computazionale. Tale classe presenta due specializzazioni, $N = 2$ e $N = 3$, in ciascuna delle quali vengono definiti mediante dei typedef punti, segmenti ed altri elementi geometrici nella dimensione corretta. Ci si trova di fronte al classico caso di *type trait*, grazie al fatto che gli enum vengono valutati al momento della compilazione. Stabilire la dimensione a livello del compilatore permette una maggior efficienza del codice e un minor rischio di errori. Si riporta la struttura di tale classe perché riveste un ruolo particolarmente interessante, mostrando la specificazione sia per il caso 2-D che per il 3-D, ma tralasciando alcune definizioni.

```
namespace Geometry{
 template<int N>
 struct Dimension {};

 template<>
 struct Dimension<2>
```

```
  {
    typedef Geometry::Vector2D Generic_Vector;
    typedef Geometry::Mesh2D::Edge2D Generic_Facet;
    typedef Geometry::Mesh2D::Cell2D Generic_Cell;
    typedef Geometry::Mesh2D Generic_Mesh;
    typedef Geometry::FractureNetwork2D
                              FractureNetwork;
    enum {dim=2};
  };

  template<>
  struct Dimension<3>
  {
    typedef Geometry::Vector3D Generic_Vector;
    typedef Geometry::Mesh3D::Facet3D Generic_Facet;
    typedef Geometry::Mesh3D::Cell3D Generic_Cell;
    typedef Geometry::Mesh3D Generic_Mesh;
    typedef Geometry::FractureNetwork3D
                              FractureNetwork;
    enum {dim=3};
  };
}
```

La presenza di tale classe permette di gestire il caso bidimensionale e quello tridimensionale allo stesso modo, ad esempio il costruttore della classe Rigid_Mesh prende in input una Generic_Mesh e non deve essere specificato nei due casi.

La classe Rigid_Mesh permette di costruire la griglia su cui risolvere il problema. Tale classe contiene al suo interno altre classi, che consentono la definizione di facce, del loro tipo e di celle. Accanto a questa struttura base, in questo progetto è stata aggiunta la classe Fracture per gestire più agevolmente la discretizzazione con le Differenze Finite Mimetiche nelle fratture e in particolare per la necessità di avere i vertici di ciascuna frattura ordinati.

```
template <class T>
class Rigid_Mesh{

public:
  class Facet;
  class Cell;
  class Facet_ID;
  class Regular_Facet : public Facet_ID;
  class Border_Facet: public Facet_ID;
```

```
    class Fracture_Facet: public Facet_ID;
    class Fracture;

protected:
    std::vector<Point> M_nodes;
    std::vector<Facet> M_facets;
    std::vector<Cell> M_cells;
    std::vector<Regular_Facet> M_internalFacets;
    std::vector<Border_Facet> M_borderFacets;
    std::vector<Fracture_Facet> M_fractureFacets;
    std::vector<Generic_Vector> M_fracturesNormal;
    std::vector<UInt> M_Criter_ref;
    Domain<T> M_domain;

    UInt M_Nfracture;
    std::vector<Fracture> M_fractures;
    std::vector<UInt> M_interVertexes;
    UInt M_gdl;
};
```

In questo frammento di codice sono riportati, oltre alle classi presenti ed innestate nella Rigid_Mesh, anche i suoi attributi. Alcuni di questi sono rappresentati da vettori che contengono le facce della mesh, i nodi visti come punti, le celle, le facce interne, le facce di bordo, le facce che appartengono alle fratture e le normali alle fratture. L'attributo M_Criter_ref contiene gli identificativi delle celle +, in cui la normale della faccia di frattura deve essere concorde a quella di riferimento presente in M_fracturesNormal.
Di fronte all'obiettivo di implementare il metodo delle Differenze Finite Mimetiche nelle fratture sono stati introdotti nuovi attributi: M_Nfracture che indica il numero di fratture e M_fractures che è il vettore delle fratture della mesh. Inoltre sono state aggiunte informazioni sulle intersezioni, raccogliendo nel vettore M_interVertexes gli identificativi dei vertici che costituiscono delle intersezioni e specificando con l'attributo M_gdl il numero di gradi di libertà addizionali in presenza di intersezioni. Tali informazioni saranno di fondamentale importanza per assemblare in maniera corretta le matrici. Prima di entrare nel dettaglio di queste novità si cercherà di chiarire la struttura della mesh.

4.1.1 La classe Facet

Si esporrà nel dettaglio la struttura della classe Facet per mostrare un esempio della tecnica e dell'approccio usato, molto di quanto osservato sarà comune ad altre classi.

```
class Facet{
```

```
public:
  Facet (const Generic_Facet & generic_facet,
         Rigid_Mesh<T> * const mesh,
         const std::map<UInt, UInt> &old_to_new_map,
         const UInt m_id, const F_Type m_ftype);

  Facet (const Facet& facet,
          Geometry::Rigid_Mesh<T> *const mesh);
  Facet (const Facet& facet);
  Facet () = delete;
  ~Facet () = default;
};
```

Tale frammento di codice riporta il costruttore di una faccia, che richiede in input una Generic_Facet, di cui si è parlato precedentemente, il puntatore alla Rigid_Mesh a cui appartiene la faccia, una mappa old_to_new_map che permette di individuare le celle che tale faccia separa e di convertire il loro identificativo della Mesh2D in quello della nuova mesh rigida, l'identificativo della faccia e una std::tuple che contiene informazioni sull'identificativo e sul tipo di faccia. Più precisamente la std::tuple F_Type è costruita nel seguente modo:

```
enum Facet_Type {Internal, Border, Fractured};
typedef std::tuple<Facet_Type, UInt> F_Type;
```

Il costruttore vuoto è stato eliminato come in tutte le altre classi per non avere metodi esterni che possano cambiare i membri protetti. Il distruttore è definito di default e sono stati creati due costruttori di copia che differiscono per il numero di parametri in input. Il primo viene chiamato dal costruttore di copia della Rigid_Mesh e costruisce la copia di una faccia che appartiene ad un'altra Rigid_Mesh, mentre il secondo costruisce la copia di una faccia appartenente alla stessa Rigid_Mesh.

I metodi di tipo *get* della classe permettono di ottenere informazioni sulle variabili protette. Si riportano alcuni esempi di metodi di tipo *get* e gli attributi della classe:

```
public:
  Geometry::Rigid_Mesh<T> *const getMesh () const
     { return M_mesh; }

  const std::vector<UInt> & getVertexesIds () const
     { return M_Vertexes_Ids; }
  friend class Rigid_Mesh<T>;

protected:
```

```
  Geometry::Rigid_Mesh<T> * M_mesh;
  UInt M_Id;
  std::vector<UInt> M_Vertexes_Ids;
  std::vector<UInt> M_separatedCells_Ids;
  Real M_size;
  Generic_Point M_center;
  Generic_Vector M_UnsignedNormal;
  F_Type M_fType;
```

Gli attributi M_center e M_UnsignedNormal servono per l'assemblaggio delle matrici, il primo contiene il baricentro della faccia e il secondo il vettore normale alla faccia. Il significato degli altri attributi è facilmente intuibile o è stato spiegato nella descrizione del costruttore. La classe è inoltre friend della classe in cui è innestata ciò significa che la classe Rigid_Mesh può accedere agli elementi protetti e privati di Facet.

Infine vi sono altri metodi di tipo protected, chiamati dal costruttore, che consentono il calcolo della normale M_UnsignedNormal, differente nel caso 2-D o 3-D, il calcolo del centro della faccia e il cambio di ordine dei vertici. Il metodo changeVertexesOrder viene chiamato dal costruttore della mesh quando si riempie il vettore delle facce di una cella per avere l'orientazione di ciascuna faccia corrispondente a quella della cella. Infatti se ciò non si verifica tale metodo riordina opportunamente i vertici della faccia.

```
protected:
  void computeUnsignedNormal (const Dimension<3> );
  void computeUnsignedNormal (const Dimension<2> );
  void computeCenter ();
  void changeVertexesOrder()
   {std::swap(M_Vertexes_Ids[0],M_Vertexes_Ids[1]);}
};
```

4.1.2 La classe Cell

Dopo questa dettagliata descrizione, si analizzerà un'altra classe innestata in quella della mesh, la classe Cell. Tale classe consente un'ottima gestione delle celle della mesh che svolgono un ruolo fondamentale nell'implementazione delle Differenze Finite Mimetiche nel mezzo poroso, in quanto per il campo di pressione si è scelta una discretizzazione *cell-based*. Le celle possono essere costituite da un numero arbitrario di facce ma occorre che siano poligonali o poliedriche. Tale classe ha la seguente struttura:

```
class Cell{
public:
  typename T::Generic_Vector outerNormalToEdge
```

```
      (const UInt & idv1, const UInt & idv2) const;

  friend class Rigid_Mesh<T>;

protected:
  Geometry::Rigid_Mesh<T> * M_mesh;
  UInt M_Id;
  std::vector<UInt> M_Vertexes_Ids;
  std::vector<UInt> M_Neighbors_Ids;
  Generic_Point M_centroid;
  Real M_volume;
  std::vector<Facet> M_CFacets;
};
```

Per quanto riguarda i costruttori di copia e il distruttore valgono considerazioni analoghe a quelle della classe precedente e il costruttore prende in input una Generic_Cell, un puntatore alla mesh e l'identificativo della cella. I metodi di tipo *get* seguono la stessa logica della classe precedente, trasferendo le informazioni riguardo le variabili protette della classe.
Grazie alla definizione della classe come friend della Rigid_Mesh è possibile agire in quest'ultima per modificare i vicini della cella, definiti nell'attributo M_Neighbors_Ids, grazie al metodo AdjustCellNeighbors.
Interessante per l'implementazione delle Differenze Finite Mimetiche è il metodo pubblico outerNormalToEdge che calcola la normale, esterna alla cella, di una faccia della cella stessa, prendendo in ingresso gli identificativi dei vertici della faccia.

4.1.3 La classe FacetID

Si descriveranno ora le altre classi riguardanti le facce della Rigid_Mesh che svolgono un ruolo significativo per distinguere e raggruppare le facce senza caratteristiche particolari, da quelle di bordo e da quelle che appartengono ad una frattura. Si osserva la presenza di una classe base Facet_ID da cui ereditano le altre tre classi, come mostrato in un estratto di codice precedente. Le classi derivate ereditano attributi e metodi ma ne hanno anche di nuovi. In questo caso ciascuna delle tre classi derivate presenterà caratteristiche differenti. Si riporta la struttura comune:

```
class Facet_ID{
protected:
  UInt Facet_Id;
  Rigid_Mesh<T>* M_mesh;
public:
  Facet_ID (const UInt facet_id,
     Geometry::Rigid_Mesh<T> *const mesh);
```

```
  Facet_ID () = delete;
  virtual ~Facet_ID () {};

  const Generic_Vector getUNormal () const
     {return M_mesh->getFacetsVector()
     [Facet_Id].getUnsignedNormal();}
};
```

Dal precedente estratto di codice si osserva che la classe base contiene due attributi, uno fornisce l'identificativo della faccia vista come un elemento della classe Facet della Rigid_Mesh e l'altro un puntatore alla mesh a cui la faccia appartiene. Il costruttore si occupa infatti di questi due attributi.

Si procede con l'osservazione delle classi derivate:

```
class Regular_Facet : public Facet_ID {
public:
  Regular_Facet (const UInt facet_Id,
    Geometry::Rigid_Mesh<T> *const mesh);

  Regular_Facet (const Regular_Facet& regular_facet,
    Geometry::Rigid_Mesh<T> *const mesh);
  Regular_Facet (const Regular_Facet &);
  ~Regular_Facet () = default;
};
```

Come ovvio tale classe non aggiunge informazioni, avendo a che fare con "facce regolari".

Oltre ad ereditare dalla classe base, la classe Border_Facet contiene in aggiunta l'indice del bordo del dominio a cui appartiene la faccia. Si richiede che tale indice venga anche passato al costruttore della classe. Ovviamente nell'implementazione del costruttore si farà riferimento al costruttore della classe base.

Accanto a queste due classi figlie la terza è quella che aggiunge il maggior numero di informazioni, infatti oltre ad un identificativo specifico della faccia di frattura si incorporano alcune informazioni relative alla frattura.

```
class Fracture_Facet: public Facet_ID {
private:
  UInt Cells_number;
  UInt M_Id;
  Real M_permeability_tang;
  Real M_permeability_norm;
```

```
  Real M_aperture;
  std::vector<UInt> Fracture_Ids;
  std::map<Fracture_Juncture,
           std::vector<UInt>> Fracture_Neighbors;
};
```

In questo caso oltre alla mesh rigida si passa al costruttore la Generic_Mesh poiché contiene le informazioni sui parametri della frattura. Gli attributi aggiuntivi sulla frattura riguardano la permeabilità normale e tangenziale, l'apertura e l'identificativo della frattura a cui la faccia appartiene, che svolge un ruolo fondamentale per gestire le intersezioni. Inoltre è presente nella classe una mappa Fracture_Neighbors, di significativa importanza nell'ordinamento delle facce della frattura, in cui Fracture_Juncture rappresenta l'identificativo del vertice che si sta considerando e std::vector<UInt> gli identificativi delle facce aventi in comune con quella della classe il vertice in considerazione.

4.1.4 La classe Fracture

Si analizzerà di seguito una nuova classe introdotta per agevolare la discretizzazione nelle fratture con le Differenze Finite Mimetiche. Per implementare correttamente le matrici $\hat{M}$, $\hat{B}$ e $\hat{D}$ del sistema (3.25) si è rivelato necessario possedere i vertici e le facce delle fratture ordinati. Finora non ci si è occupati di ciò poiché nelle fratture venivano utilizzati i Volumi Finiti. Inoltre avere gli elementi delle fratture ordinati ha permesso di disporre facilmente di quanto necessario per calcolare i termini delle matrici, ma tali dettagli implementativi saranno riportati in seguito.
La classe è caratterizzata nel seguente modo:

```
class Fracture{
public:
  Fracture (const UInt Id,
            Geometry::Rigid_Mesh<T> *const mesh,
            const Generic_Mesh &generic_mesh);

  Fracture (const Fracture &);
  Fracture (const Fracture & f,
            Geometry::Rigid_Mesh<T> *const mesh);
  ~Fracture () = default;
  friend class Rigid_Mesh<T>;

protected:
  Geometry::Rigid_Mesh<T>* M_mesh;
  UInt M_Id;
  UInt M_nfacets;
```

```
  UInt M_nvertexes;
  std::map<UInt,UInt> M_orderedVertexes;
  std::map<UInt,UInt> M_orderedFacets;
  Real M_lf;
  Real M_kt;
  Real M_kn;
  UInt M_nInter;
  std::vector<UInt> M_Inter;

protected:
  void orderFacetsVertexes
      (const Generic_Mesh &generic_mesh);
};
```

Come in tutte le altre classi vi è un puntatore alla mesh a cui la frattura appartiene, inoltre vi sono informazioni sul numero di vertici e di facce della frattura in considerazione e sui parametri, quali permeabilità e apertura. Sono presenti nella classe due attributi che svolgono un ruolo fondamentale per la gestione delle intersezioni: M_nInter che indica il numero di intersezioni presenti nella frattura in considerazione e M_Inter che è un vettore contenente gli identificativi dei vertici di intersezione di questa frattura.
Gli elementi più importanti sono però le due mappe che contengono le facce e i vertici ordinati e in cui il primo elemento è la posizione nella frattura e il secondo rappresenta l'identificativo del vertice o della faccia di frattura, a seconda della mappa che si sta considerando. Per ordinare vertici e facce è stato utilizzato il metodo protetto, chiamato nel costruttore, orderFacetsVertexes che prende in input la mesh generica. Infatti nella costruzione della Mesh2D vengono ordinati i vertici delle fratture, di conseguenza si è ritenuto opportuno utilizzare tale parte di codice già esistente e completarla con l'ordinamento delle facce di frattura, non ancora trattato. L'implementazione del metodo orderFacetsVertexes viene eseguita solo per il caso bidimensionale mediante una specializzazione.

Osservazione 4.1.1. *I vertici sono ordinati a partire da quello con ascissa inferiore e nel caso di frattura verticale da quello con ordinata inferiore. Di conseguenza le facce sono ordinate allo stesso modo. Tale osservazione è di fondamentale importanza per l'imposizione delle condizioni al bordo, che dovrà essere coerente con l'ordinamento dei vertici.*

L'introduzione della classe Fracture ha avuto come conseguenza l'aggiunta di un attributo nella classe Rigid_Mesh, ossia un std::vector<Fracture>, un vettore di fratture.

4.1.5 La classe Rigid_Mesh

Finora sono state analizzate le varie classi presenti nella Rigid_Mesh senza porre attenzione alla classe stessa, essa presenta alcuni attributi che sono stati indicati ad inizio capitolo.
Per quanto riguarda i costruttori della Rigid_Mesh valgono osservazioni analoghe a quelle delle classi precedenti:

```
Rigid_Mesh (Generic_Mesh & generic_mesh,
            Domain<T>& domain);
Rigid_Mesh (const Rigid_Mesh &mymesh);
Rigid_Mesh () = delete;
~Rigid_Mesh () = default;
```

Di fondamentale importanza è il parametro di input del costruttore Generic_Mesh che corrisponde alla Mesh2D nel caso bidimensionale e a cui la Rigid_Mesh si ispira. Tale costruttore chiama altri metodi protetti della classe in cui si costruiscono i vettori di facce, celle e fratture, si calcolano le intersezioni, si ordinano i vertici e le facce di ogni frattura, si definiscono i vicini delle facce e si stabiliscono molte altre informazioni. Si riporta qui di seguito la definizione di due di questi metodi, che fanno parte delle novità di questo progetto e che si occupano di costruire il vettore delle fratture e di individuare i vertici di intersezione.

```
protected:
 void FractureBuilder (Generic_Mesh & generic_mesh);
 void ComputeIntersectionVertexes();
```

La classe comprende anche dei metodi di tipo *get* per estrarre le sue variabili protette.

Si è cercato in questa sezione di dare informazioni principalmente sulla struttura del codice, tralasciando i dettagli di implementazione. Si osserva un approccio omogeneo e coerente in tutto lo sviluppo di questa classe così articolata. Tale struttura sarà fondamentale per permettere lo sviluppo successivo del codice.

4.2 Il dominio del problema

La classe Domain contiene i bordi del dominio e svolge un importante ruolo per l'assegnazione delle condizioni al bordo. Tale classe include un vettore di elementi della classe DomainBorder, ciascuno dei quali rappresenta un tratto di bordo. La classe DomainBorder ha infatti come attributi un Generic_Border, che nel caso di dominio 2-D è un segmento e un identificativo per il tratto di bordo a cui corrisponde.

Un ruolo interessante è svolto dal seguente metodo:

```
const UInt getBorderId
  (const std::vector<Point2D> facetVertexes) const;
```

Questo utilizza metodi della libreria CGAL, tra cui il metodo bool has_on (const Point_2< Kernel > & p) const, che verifica se un vertice appartiene o meno ad un segmento di bordo. Per una descrizione più approfondita si rimanda al sito [19].

Queste due classi, seppur semplici, svolgono un importante ruolo per la definizione delle condizioni al bordo del dominio, descritte nella successiva sezione.

4.3 Le condizioni al bordo

Fondamentali per la risoluzione di una qualsiasi equazione a derivate parziali sono le condizioni al bordo. Un'altra novità di questo codice è la generalizzazione delle condizioni al bordo, infatti se nelle versioni precedenti non era necessario imporle ai bordi delle fratture in maniera esplicita, ora lo diventa. Solitamente se la frattura è immersa si impongono condizioni omogenee sulla velocità, ma potrebbe capitare che la frattura arrivi al bordo del dominio e che si voglia imporre una condizione sulla pressione. Quindi il codice tratta casi sia con condizioni sulla velocità che con condizioni sulla pressione.

Per generalizzare è stata introdotta una classe base da cui derivano due classi figlie, una per gestire le condizioni al bordo del mezzo poroso e l'altra per gestire quelle delle fratture. La differenza sostanziale tra le due classi riguarda dove sono imposte queste condizioni, infatti nel mezzo poroso sono applicate sul dominio, appartenente alla classe Domain, mentre nelle fratture sul network stesso. Di conseguenza si ha una classe base:

```
template <class T>
class BCond
{
typedef typename T::Generic_Point Generic_Point;
public:
  class BorderBC {
  protected:
     UInt m_Id;
     BCType bcType;
     std::function<D_Real(Generic_Point)> BC;

  public:
```

```
      BorderBC (UInt Id, BCType bctype, std::function
                <D_Real(Generic_Point)> const & bc);

      BorderBC (const BorderBC&) = default;
      virtual ~BorderBC () {};
      friend class BCond;
    };

  BCond (std::vector<BorderBC>& borderbc);
  BCond (const BCond&) = default;
  virtual ~BCond () {};

protected:
  std::vector<BorderBC> BordersBCVector;
};
```

La classe BCond contiene al suo interno la classe BorderBC in cui viene definita ciascuna condizione al bordo mediante un identificativo, la funzione che descrive la condizione con un *function-wrapper* e il tipo grazie a:

```
enum BCType{Dirichlet, Neumann};
```

Le condizioni al bordo vere e proprie non sono altro che un vettore costituito da elementi della classe BorderBC. Inoltre si noti che la classe BCond è una classe template.

La prima classe che deriva da BCond è caratterizzata nel seguente modo:

```
template <class T>
class MediaBC: public BCond<T>
{
typedef typename BCond<T>::BorderBC bbc;
public:
  MediaBC (std::vector<bbc> & borderbc,
           Geometry::Domain<T>& domain);

protected:
  Geometry::Domain<T>& M_domain;
};
```

Tale classe, come già accennato, si differenzia dalla precedente per l'aggiunta di una variabile che riguarda il dominio del problema. Nel costruttore di tale classe viene effettuato il riordinamento delle condizioni secondo l'indice del bordo a cui sono assegnate, tale procedura viene eseguita utilizzando aspetti nuovi del C++11:

```
auto cmp = [](bbc bc1,bbc bc2)
         {return (bc1.getId()<bc2.getId());};
std::sort (this->BordersBCVector.begin(),
           this->BordersBCVector.end(), cmp);
```

Come si può notare viene usata una *lambda-function* e l'algoritmo di ordinamento std::sort.

L'altra classe derivata è la seguente:

```
template <class T>
class FractureBC: public BCond<T>
{
typedef typename T::FractureNetwork FractureNetwork;
public:
FractureBC(std::vector<typename BCond<T>::BorderBC>&
           borderbc, FractureNetwork & fb);

protected:
  FractureNetwork FB;
};
```

Tale classe al suo interno presenta un nuovo attributo: il network di fratture, che viene utilizzato nel costruttore per riordinare le condizioni al bordo coerentemente con l'ordinamento dei vertici.

Si tratta di un buon esempio di classe base da cui ereditano due figlie che differiscono sostanzialmente per "l'ambiente" in cui sono poste. Nel capitolo sui risultati numerici sarà interessante vedere cosa succede nel caso di condizioni sulla pressione imposte ai bordi di una frattura immersa e analizzare il caso di una frattura semi-immersa riproducendo i casi test introdotti in [1].

4.4 Le matrici delle Differenze Finite Mimetiche

In questa sezione si analizzerà l'implementazione delle varie matrici che sono state introdotte nel capitolo precedente e che costituiscono il problema (3.25). Per prima cosa si discuterà l'approccio usato nell'assemblaggio delle matrici del mezzo poroso, successivamente della frattura e infine dell'accoppiamento. In questa sezione sarà possibile capire quanto siano differenti le Differenze Finite Mimetiche nel caso monodimensionale e bidimensionale, infatti oltre alle differenze teoriche annunciate nei capitoli precedenti qui si noteranno quelle implementative. Nonostante siano presenti delle differenze tutte queste matrici sono accomunate dalla derivazione da un'unica classe che contiene due metodi puramente virtuali, implementati solo nelle classi

derivate. Si inizierà con l'analizzare questa classe generale e poi si entrerà nel dettaglio delle varie classi.

4.4.1 La classe base MatrixHandler

Questa classe è la base di tutte le matrici che vengono implementate in questo codice, infatti non ha dimensione fissata ma il suo costruttore richiede in input il numero di righe e colonne, che dipende dal tipo di discretizzazione scelto. Le dimensioni non sono state introdotte come parametri template perché comunque la matrice viene dimensionata al momento dell'esecuzione.

```
enum DiscretizationType
   {D_Cell, D_Facet, D_Fracture, D_Vertex};

MatrixHandler (
  const Geometry::Rigid_Mesh<T> & rigid_mesh,
  DType dtype_row, DType dtype_col, BCond<T> &bc );
```

Come al solito il costruttore vuoto viene eliminato, in questa classe viene eliminato anche il costruttore di copia e il distruttore è virtuale essendo una classe base. Le dimensioni possibili per le matrici che derivano da questa classe sono quattro:

- D_Cell indica un numero di righe o colonne pari a quello delle celle della mesh.
- D_Facet indica un numero di righe o colonne pari a quello delle facce della mesh a cui va aggiunto il numero delle facce appartenenti alle fratture a causa dello sdoppiamento dei gradi di libertà sulla velocità del mezzo poroso in corrispondenza di tali facce.
- D_Fracture indica un numero di righe o colonne pari a quello delle facce in corrispondenza di fratture della mesh a cui va aggiunto il numero delle intersezioni.
- D_Vertex è leggermente più complicato e pari a:

  ```
  rigid_mesh.getFractureFacetsVector().size() +
  + rigid_mesh.getNfracture() +
  + rigid_mesh.getGdl()
  ```

  ossia pari al numero dei nodi delle fratture a cui va aggiunto il valore dell'attributo M_gdl, che considera il raddoppio dei gradi di libertà alle intersezioni; per esempio se si ha una sola intersezione tra due fratture M_gdl sarà pari a due, se tra tre fratture sarà pari a tre. Come già spiegato nel capitolo precedente occorre raddoppiare i gradi di libertà in corrispondenza delle intersezioni per ciascuna frattura coinvolta.

La matrice globale è composta da varie matrici con specifiche dimensioni:

$$\begin{bmatrix} \underset{D\_Fa \times D\_Fa}{\mathbf{M}} & \underset{D\_Fa \times D\_Ce}{\mathbf{D}} & 0 & \underset{D\_Fa \times D\_Fr}{\mathbf{C^T}} \\ \underset{D\_Ce \times D\_Fa}{\mathbf{B}} & 0 & 0 & 0 \\ 0 & 0 & \underset{D\_Ve \times D\_Ve}{\hat{\mathbf{M}}} & \underset{D\_Ve \times D\_Fr}{\hat{\mathbf{D}}} \\ \underset{D\_Fr \times D\_Fa}{\hat{\mathbf{C}}} & 0 & \underset{D\_Fr \times D\_Ve}{\hat{\mathbf{B}}} & 0 \end{bmatrix}$$

Da tale classe base è possibile ricavare informazioni sulla matrice che si sta assemblando grazie ai metodi *get*, che danno informazioni sui seguenti attributi:

```
protected:
  const Geometry::Rigid_Mesh<T> & M_mesh;
  DType M_policy_row;
  DType M_policy_col;
  D_UInt M_size_rows;
  D_UInt M_size_cols;
  std::unique_ptr<SpMat> M_Matrix;
  BCond<T>& m_Bc;
```

Ovviamente la classe contiene informazioni anche sulle condizioni al bordo poiché queste modificano le matrici o danno contributi ai termini noti e sono fondamentali in fase di assemblaggio.

Matrix_Handler è una classe astratta, contiene infatti due metodi *pure virtual*, implementati solo nelle classi figlie e in cui saranno scritte le righe fondamentali del codice:

```
virtual void assemble()=0;
virtual void fillMatrix(SpMat &MAT)=0;
```

La differenza tra i due è che il metodo fillMatrix riempie la matrice globale che riceve in input inserendo gli elementi nelle posizioni corrette, mentre assemble costruisce la matrice. La presenza del metodo assemble potrebbe rivelarsi utile di fronte alla necessità di costruire un precondizionatore. Per ora si preferisce risolvere il sistema monolitico e quindi assemblare le matrici

usando fillMatrix.
La matrice che il costruttore della classe crea è del seguente tipo:

```
typedef Eigen::SparseMatrix<D_Real> SpMat;
```

Si tratta di una rappresentazione matriciale versatile, migliore del formato compresso, infatti tutti i valori non nulli sono immagazzinati in un singolo buffer. Più precisamente i valori diversi da zero di ciascuna colonna sono memorizzati come una coppia valore e indice della riga. Inoltre, a differenza del formato compresso, ci possono essere spazi aggiuntivi tra valori non nulli di due successive colonne in modo tale che l'inserimento di un nuovo elemento possa essere fatto risparmiando sull'allocazione di memoria e sulle copie. Il tipo dei coefficienti è posto pari a double e non avendo messo il secondo parametro, di default la matrice è memorizzata con una procedura *column-major*. Tra le matrici di questa classe è possibile effettuare un gran numero di operazioni, si rimanda al sito [20]. Si osservi però che l'attributo che rappresenta la matrice è uno *smart-pointer* ad essa, ciò è dovuto al fatto che non si può avere all'interno di una classe una matrice del tipo descritto sopra a meno che non occupi un numero preciso di byte. Di conseguenza per poter sfruttare le potenzialità delle Eigen è stato utilizzato uno *smart-pointer* che libera la memoria al momento della sua eliminazione.

E' stata analizzata la classe mostrandone le sue peculiarità e la sua grande versatilità, apprezzando in particolar modo la libreria Eigen. In realtà è possibile utilizzare questa classe per qualsiasi problema differenziale definito su una Rigid_Mesh.

4.4.2 La classe MatrixMglob

Si tratterà ora l'analisi dettagliata delle matrici introdotte con la discretizzazione mediante le Differenze Finite Mimetiche. Innanzitutto si descriverà la matrice di massa, relativa alla velocità, nel mezzo poroso. Per implementare tale matrice di fondamentale importanza è la parte teorica sviluppata nel capitolo precedente. E' infatti necessario assemblarla a partire dalle matrici locali, esattamente secondo la logica esposta nella teoria: definizione dei prodotti interni globali e delle matrici ad essi associate a partire dai prodotti locali e dalle corrispondenti matrici locali.
La classe ovviamente eredita da Matrix_Handler, quindi avrà metodi e attributi della classe base, a cui viene aggiunta la seguente struttura:

```
template <class T>
class Matrix_Mglob: public MatrixHandler<T>
{
  typedef typename T::Generic_Point Generic_Point;
  typedef std::function<D_Real(Generic_Point)> func;
```

```
public:
  Matrix_Mglob (
    const Geometry::Rigid_Mesh<T> &rigid_mesh,
    MediaBC<T>& Bc, func kappa_11, func kappa_12,
    func kappa_22, bool weak = true);
  Matrix_Mglob(const Matrix_Mglob&) = delete;
  Matrix_Mglob() = delete;
  ~Matrix_Mglob() = default;
  void setPsi(D_Real const & psi){ M_psi = psi; }

  void assemble();
  void fillMatrix(SpMat &MAT);
  void computeBC();

protected:
  std::unique_ptr<Vector> _b;
  func K11;
  func K12;
  bool M_weak;
  D_Real M_psi;
  static D_Real const M_defaultPsi;
};
template<class T>
D_Real const Matrix_Mglob<T>::M_defaultPsi = 0.75;
```

Il costruttore della matrice richiede in input la Rigid_Mesh del problema, le condizioni al bordo del mezzo poroso e i valori del tensore di permeabilità, necessari per implementare le matrici locali. Quest'ultimi possono essere funzioni del punto e per cui sono rappresentati da una std::function. E' presente la dichiarazione dei due metodi che nella classe base sono dichiarati *virtual* e di cui si analizzerà poi l'implementazione. Inoltre la classe contiene uno *smart-pointer* ad un Vector definito nel seguente modo:

```
typedef Eigen::VectorXd Vector;
```

Questo vettore viene riempito con i termini derivanti dalle condizioni al bordo, tale procedura viene eseguita dal metodo void computeBC().
L'attributo weak fornisce informazioni su come applicare le condizioni di accoppiamento, se in forma debole o forte e sarà presente anche nelle altre classi; se non specificato di default saranno applicate le condizioni in forma debole.

Per quanto riguarda l'implementazione vera e propria, occorre per prima cosa effettuare alcune precisazioni sulla convenzione adottata per gestire lo sdoppiamento dei gradi di libertà della velocità in prossimità delle facce di

frattura. Infatti ad ogni faccia di frattura corrispondono due righe e due colonne della matrice e si è deciso di aggiungere i gradi di libertà derivanti dallo sdoppiamento in fondo alla matrice. Occorre ora distinguere le due celle che toccano la faccia di frattura sdoppiata. Si è stabilito che la cella + è quella che ha la normale di faccia concorde a quella di riferimento della mesh, contenuta nel vettore M_fracturesNormal. L'altra sarà la cella -, coerentemente con quanto affermato in precedenza. Poiché tale procedimento dovrà essere eseguito più volte, sono stati memorizzati nel vettore M_Criter_ref, attributo protetto di Rigid_Mesh, gli indici delle celle +.

Il metodo fillMatrix di questa classe consiste nei seguenti passaggi:

1. Si esegue un ciclo sulle celle della mesh nel quale:

   - vengono assemblate le matrici locali. Ciascuna di esse è quadrata di dimensioni pari al numero di facce della cella relativamente a cui è costruita. Inoltre ogni matrice locale viene assemblata, come riportato nella teoria del capitolo precedente, a partire da altre due matrici locali R_P e N_P. Infatti la classe per implementare le matrici locali, che non eredita da Matrix_Handler, possiede i seguenti metodi e attributi:

     ```
     public:
       void assemble();
     private:
       void assemble_R();
       void assemble_N();

       Matrix M_Matrix_M;
       Matrix M_Matrix_R;
       Matrix M_Matrix_N;
     ```

     Si noti che la classe locale non è una classe template essendo valida solo nel caso 2-D e che si è scelto di utilizzare le matrici dense delle Eigen avendo a che fare con piccole dimensioni.
   - Si calcola la corretta posizione degli elementi locali nella matrice globale, tenendo conto dello sdoppiamento dei gradi di libertà sulle facce di frattura.
   - Si stabilisce il segno dei termini, moltiplicandoli per i segni corrispondenti alle facce in questione. Quest'ultimi sono determinati confrontando l'ordine dei vertici della faccia vista come elemento della cella e come componente della Rigid_Mesh.

2. Si impongono le condizioni di accoppiamento. Se si sceglie di farlo in forma forte occorre modificare le righe della matrice associate alle facce di frattura, cancellando i valori inseriti precedentemente mediante

il comando prune delle Eigen ed inserendo i termini che riproducono le condizioni (3.26). Se invece si impongono in forma debole, bisogna solamente aggiungere i contributi riportati in (3.23) nelle posizioni opportune della matrice.

3. Si modificano le righe della matrice corrispondenti alle facce di bordo su cui sono imposte condizioni sulla velocità, cancellando tali righe e sostituendole semplicemente con un uno in posizione diagonale.

Si riporta il codice che tratta quest'ultimo aspetto:

```
for (auto facet_it:
     this->M_mesh.getBorderFacetsVector())
{
  UInt borderId = facet_it.getBorderId();
  UInt facetId = facet_it.getFacetId();
  if(this->m_Bc.getBordersBCVector()
                [borderId].getBCType() == Neumann)
  {
     MAT.prune([facetId](UInt i, UInt, Real)
                { return ( i != facetId ); });
     MAT.insert(facetId, facetId) = 1;
  }
}
```

Tale frammento di codice mostra l'utilizzo dei seguenti due comandi delle Eigen:

```
void prune ( const KeepFunc & keep = KeepFunc() )
Scalar& insert ( Index row, Index col )
```

Il primo cancella gli elementi della matrice che non soddisfano il predicato keep, che deve essere un *functor* e che in questo caso è definito mediante una *lambda-function*. Il secondo inserisce un elemento nella posizione (row,col) della matrice.

Il metodo computeBC individua tra i lati di bordo quelli che possiedono una condizione sulla velocità o sulla pressione e costruisce la parte $\mathbf{g}$ del termine noto come riportato nella (3.24). Se la condizione è sulla velocità viene inserito nella posizione opportuna del vettore _b il termine $g_{\mathbf{u}}$ valutato al centro della faccia. Se invece la condizione è sulla pressione si calcola in maniera approssimata, con il metodo dei trapezi, l'integrale di g_P e lo si inserisce in _b.

Il metodo assemble che si occupa della singola matrice, non presenta differenze dal metodo fillMatrix a meno di alcuni metodi delle Eigen. Infatti

per riempire la matrice occorre aggiungere elementi ad un vettore ausiliario definito nel seguente modo:

```
std::vector<Triplet> Matrix_elements;
```

dove:

```
typedef Eigen::Triplet<D_Real> Triplet;
```

La classe template Triplet delle Eigen è una struttura perfetta per contenere le informazioni necessarie per costruire una matrice, ossia riga, colonna e valore dell'elemento. Durante il ciclo sulle celle della mesh vengono aggiunti elementi al vettore e solo una volta che tutti i valori sono stati inseriti si usa il comando per riempire la matrice:

```
this->M_Matrix->setFromTriplets(
    Matrix_elements.begin(), Matrix_elements.end());
```

Si osservi che nel caso di imposizione debole delle condizioni di accoppiamento si aggiungono al vettore Matrix_elements anche i termini (3.23). Se le condizioni di accoppiamento sono imposte in forma forte invece si costruisce la matrice e poi si modificano le righe usando prune e insert come per l'imposizione delle condizioni sulla velocità.

Questa sezione è stata analizzata con il massimo dettaglio in quanto esplicativa sull'applicazione delle Differenze Finite Mimetiche nel mezzo poroso, si proseguirà più rapidi nella sezione successiva, in cui la costruzione della matrice sarà più semplice.

4.4.3 La classe DivOperator

La classe DivOperator deriva da Matrix_Handler e si occupa della costruzione della matrice associata all'operatore divergenza e della sua trasposta. Il costruttore della classe si basa su quello della classe da cui deriva, a meno dell'inizializzazione del parametro che indica il tipo di imposizione delle condizioni di accoppiamento. Questa infatti è l'unica variabile aggiuntiva della classe.

Questa classe ha a che fare sia con la matrice B che con la D del sistema (3.25), quindi oltre al metodo assemble è necessario implementare un metodo che permetta di ottenere la trasposta opportunamente modificata. I metodi della classe sono quindi:

```
void assemble();
void fillMatrix(SpMat &MAT);
SpMat getMatrix_transpose();
```

Il metodo fillMatrix riempie le posizioni corrette della matrice data in input effettuando un ciclo sulle celle e per ciascuna cella sulle sue facce e calcolando il valore da inserire, dato dalla dimensione della faccia in considerazione moltiplicata per il segno opportuno. La parte della matrice che corrisponde a B^T viene però modificata, infatti occorre annullare le righe corrispondenti alle facce di bordo con condizioni sulla velocità ed imporre le condizioni di accoppiamento (3.26) se sono in forma forte. Nella maggior parte dei casi test si sceglierà l'imposizione in forma debole.

Il metodo assemble si occupa dell'implementazione della matrice B.
Per assemblare la matrice D del sistema (3.25) è necessario utilizzare il metodo getMatrix_transpose che per prima cosa, grazie agli strumenti forniti dalle Eigen, determina la trasposta della matrice della classe e successivamente la corregge per imporre condizioni sulla velocità e di accoppiamento se in forma forte.

4.4.4 La classe MatrixMglobFrac

In questa sezione si inizierà la descrizione della parte di codice legata alla risoluzione con le Differenze Finite Mimetiche del problema di Darcy nelle fratture. Per prima cosa si analizzerà la classe MatrixMglobFrac. Anche tale classe deriva da Matrix_Handler: ciò è stato possibile grazie alla generalizzazione della classe delle condizioni al bordo descritta in precedenza. Infatti il costruttore della classe Matrix_Handler richiede che le condizioni al bordo appartengano alla classe madre, in questo modo è possibile fornire le condizioni della classe figlia MediaBC per le due matrici del mezzo poroso e le condizioni dell'altra classe figlia, FractureBC, per le due matrici della frattura. Tale approccio consente di evitare ripetizioni del codice, infatti se non fosse stato generalizzato in questo modo sarebbe stato necessario implementare altre due classi, una per le condizioni al bordo e una analoga a Matrix_Handler ma per le fratture.

Il cuore della classe è l'implementazione dei due soliti metodi virtuali della classe base. Le classi nella frattura sono ancora in versione template, in vista di una futura generalizzazione del codice, ma al momento i metodi di assemblaggio delle matrici sono stati implementati come specificazioni, in quanto la versione presente risulta adatta al caso bidimensionale.

Per quanto riguarda i costruttori della classe valgono osservazioni analoghe a quelle delle classi precedenti:

```
Matrix_Mglob_Frac(
    const Geometry::Rigid_Mesh<T> &rigid_mesh,
    FractureBC<T>& Bc);
```

```
Matrix_Mglob_Frac(const Matrix_Mglob_Frac&)=delete;
Matrix_Mglob_Frac() = delete;
~Matrix_Mglob_Frac() = default;
```

Il costruttore riceve in ingresso solamente la mesh e le condizioni al bordo, infatti grazie alla creazione della classe Fracture sarà possibile ottenere le informazioni sui parametri delle fratture ogniqualvolta sia necessario.

Il resto della classe è composto dai metodi di assemblaggio e da quello per il calcolo del termine noto.

```
public:
  void assemble();
  void fillMatrix(SpMat &MAT);
  void computeBC();

protected:
  std::unique_ptr<Vector> _b;
```

Svolge un ruolo importante in questa classe, come pure nella MatrixMglob, la presenza di uno *smart pointer* che viene riempito con i termini legati alle condizioni al bordo.

Nel metodo fillMatrix, per ciascuna frattura presente nel mezzo, vengono effettuati i seguenti passi:

1. Si esegue un ciclo sulle facce ordinate della frattura in cui:

   - si calcolano gli elementi della matrice locale M_i^{RT} (3.8), considerando gli opportuni fattori moltiplicativi legati alla dimensione della faccia, alla permeabilità tangenziale e all'apertura della frattura.
   - Si inseriscono i valori della matrice locale nella posizione corretta della matrice data in input. In tale passaggio occorre controllare che la faccia in considerazione non abbia vertici di intersezione. La matrice locale viene inserita nella globale secondo le tecniche tradizionali dei metodi degli Elementi Finiti, per cui sulla diagonale si sommano i contributi legati a due matrici locali. Ciò non deve avvenire qualora ci si trovi di fronte ad una intersezione perché il grado di libertà viene sdoppiato.
     A prescindere dai fattori legati agli aspetti fisici del problema, la matrice senza intersezioni assume la forma riportata nella pagina seguente.

$$\frac{1}{6}\begin{pmatrix} 2 & 1 & 0 & \dots & \dots & \dots & 0 \\ 1 & 4 & 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & 4 & 1 & 0 & \dots & 0 \\ \dots & \dots & \ddots & \ddots & \ddots & \dots & \dots \\ \dots & \dots & \dots & \ddots & \ddots & \ddots & \dots \\ 0 & \dots & \dots & 0 & 1 & 4 & 1 \\ 0 & \dots & \dots & \dots & 0 & 1 & 2 \end{pmatrix}.$$

Supponendo la presenza di un'intersezione nel vertice tra la seconda e la terza faccia, la matrice assume la forma seguente:

$$\frac{1}{6}\begin{pmatrix} 2 & 1 & 0 & \dots & \dots & \dots & \dots & 0 \\ 1 & 4 & 1 & 0 & \dots & \dots & \dots & 0 \\ 0 & 1 & \mathbf{2} & 0 & 0 & \dots & \dots & 0 \\ \dots & \dots & 0 & \mathbf{2} & 1 & 0 & \dots & \dots \\ \dots & \dots & 0 & 1 & 4 & 1 & 0 & \dots \\ \dots & \dots & \dots & \dots & \ddots & \ddots & \ddots & \dots \\ 0 & \dots & \dots & \dots & 0 & 1 & 4 & 1 \\ 0 & \dots & \dots & \dots & \dots & 0 & 1 & 2 \end{pmatrix}.$$

Lo sdoppiamento del grado di libertà corrisponde ad una visione della frattura come composta da due fratture. Infatti la presenza dell'intersezione comporta delle discontinuità, di conseguenza è necessario distinguere il flusso prima e dopo l'intersezione.
Dato che per ora non si considera il caso di biforcazioni, non sarà possibile avere intersezioni che coinvolgano il primo e l'ultimo vertice delle fratture.

2. Si impongono le condizioni al bordo sulla velocità con la stessa procedura descritta nel dettaglio per la matrice M, quindi con la cancellazione delle righe corrispondenti ai vertici di bordo con tali condizioni e l'inserimento di un uno sulla diagonale. In seguito al riordinamento dei vertici, quelli di bordo sono il primo e l'ultimo, quindi è solo necessario verificare se su questi due sia stata imposta una condizione sulla velocità.

3. Si aggiornano la riga e la colonna da cui partire per proseguire con il riempimento della matrice per la successiva frattura. Bisogna tenere conto di eventuali raddoppi dei gradi di libertà, che non vengono posti in fondo alla matrice al contrario di quanto accade nel mezzo poroso circostante.

Il metodo assemble assembla la sotto-matrice $\hat{M}$ in maniera analoga a quanto fatto nel metodo fillMatrix con le uniche differenze legate ai metodi

delle matrici sparse delle Eigen di cui si è parlato in precedenza.

Il metodo computeBC si propone di aggiornare il termine noto coerentemente con le condizioni al bordo. Il termine $\hat{\mathbf{g}}$ del (3.25) assume quindi una delle seguenti forme a seconda del tipo di condizioni:

$$\hat{\mathbf{g}} = \begin{pmatrix} \hat{g}_p(x_1) \\ 0 \\ \vdots \\ 0 \\ -\hat{g}_p(x_{n+1}) \end{pmatrix}, \quad \begin{pmatrix} \hat{g}_{\mathbf{u}}(x_1) \\ 0 \\ \vdots \\ 0 \\ \hat{g}_{\mathbf{u}}(x_{n+1}) \end{pmatrix}, \quad \begin{pmatrix} \hat{g}_p(x_1) \\ 0 \\ \vdots \\ 0 \\ \hat{g}_{\mathbf{u}}(x_{n+1}) \end{pmatrix}, \quad \begin{pmatrix} \hat{g}_{\mathbf{u}}(x_1) \\ 0 \\ \vdots \\ 0 \\ -\hat{g}_p(x_{n+1}) \end{pmatrix}.$$

Tale classe presenta quindi la stessa struttura delle altre classi ma introduce novità legate al fatto che la frattura si può vedere come un oggetto unidimensionale.

4.4.5 La classe DivOperatorFrac

La classe DivOperatorFrac presenta la solita struttura e si occupa dell'assemblaggio sia di $\hat{B}$ che della sua trasposta. La matrice $\hat{B}$ è particolarmente semplice da assemblare essendo in un caso monodimensionale e avendo come gradi di libertà della pressione i valori al centro delle facce. La struttura della matrice nel caso di assenza di intersezioni è la seguente:

$$\hat{B} = \begin{pmatrix} 1 & -1 & 0 & \dots & \dots & \dots \\ 0 & 1 & -1 & 0 & \dots & \dots \\ \dots & \dots & \ddots & \ddots & \dots & \dots \\ \dots & \dots & 0 & 1 & -1 & 0 \\ \dots & \dots & \dots & 0 & 1 & -1 \end{pmatrix}.$$

Viene utilizzato il solito approccio nel definire costruttori e distruttore, il costruttore in particolare prende in ingresso semplicemente la mesh e le condizioni al bordo, esattamente come nella classe precedente. Non sono presenti attributi aggiuntivi e il cuore della classe sta nei soliti due metodi.
In presenza di intersezioni la matrice $\hat{B}$ ha delle colonne aggiuntive e la matrice $\hat{D}$ delle righe aggiuntive, in accordo con lo sdoppiamento dei gradi di libertà nei punti di intersezione. Se ci fosse una intersezione nel secondo vertice della frattura, la matrice $\hat{B}$ assumerebbe la seguente struttura:

$$\hat{B} = \begin{pmatrix} 1 & -1 & 0 & \dots & \dots & \dots & \dots \\ 0 & 0 & \mathbf{1} & -1 & 0 & \dots & \dots \\ 0 & 0 & 0 & 1 & -1 & 0 & \dots \\ \dots & \dots & \dots & \ddots & \ddots & \ddots & \dots \\ \dots & \dots & \dots & \dots & 0 & 1 & -1 \end{pmatrix}. \tag{4.1}$$

Tale classe assume però un ruolo fondamentale in presenza di intersezioni e la matrice (4.1) viene ulteriormente modificata. Per esempio se si ha un network caratterizzato da due fratture intersecanti in un punto occorre aggiungere alla matrice $\hat{B}$ una riga per rappresentare il vincolo della conservazione del flusso. È opportuno anche aggiungere una colonna a $\hat{D}$ per l'imposizione della continuità della pressione in forma debole. Si osserva che la colonna aggiuntiva di $\hat{D}$ è esattamente la trasposta della riga addizionale in $\hat{B}$. Per questo si è deciso di implementare i vincoli di intersezione con tali matrici. Ciò ha permesso inoltre di evitare ripetizioni di alcune operazioni, in particolare l'individuazione delle posizioni opportune della matrice in cui inserire i termini delle condizioni.

Osservazione 4.4.1. *Grazie all'ordinamento dei vertici e all'ipotesi che la tangente sia diretta dal vertice iniziale al vertice finale della frattura, risulta particolarmente semplice imporre la condizione di conservazione del flusso. Supponendo di avere una frattura suddivisa in due parti* Γ_1 *e* Γ_3 *e l'altra suddivisa in* Γ_2 *e* Γ_4 *e che* Γ_1 *e* Γ_2 *contengano vertici precedenti nell'ordinamento di ciascuna frattura, la condizione diventa:*

$$\hat{u}_1 - \hat{u}_3 + \hat{u}_2 - \hat{u}_4 = 0 \tag{4.2}$$

dove con u_i *si indica il grado di libertà nel vertice di intersezione del ramo* i *di frattura.*

Dopo questa importante osservazione si riportano i passi principali da eseguire per ciascuna frattura nell'assemblaggio della matrice:

1. Si esegue un ciclo sulle facce ordinate in cui:
   - si valuta se il secondo vertice della faccia è un vertice di intersezione, se lo è si raddoppia il grado di libertà e si tiene traccia della posizione, utile per l'imposizione delle condizioni,
   - si inseriscono gli elementi nella posizione corretta, sia quelli relativi alla matrice $\hat{B}$ che quelli riguardanti la sua trasposta.
2. Si annullano le righe di $\hat{B}^T$ che corrispondono ai vertici su cui sono imposte condizioni sulla velocità.
3. Si inseriscono i termini derivanti dalle condizioni di intersezione; è possibile avere anche più intersezioni.

Anche in questo caso è possibile costruire le singole matrici con due metodi, uno che fornisce la $\hat{B}$ e l'altro la trasposta opportunamente modificata.

Nonostante la semplicità delle matrici $\hat{B}$ e $\hat{D}$, tale classe diventa particolarmente interessante in presenza di intersezioni, ma le condizioni vengono imposte agevolmente grazie all'ordinamento dei vertici effettuato in precedenza.

4.4.6 La classe MatrixFrac

Tale classe si occupa dell'assemblaggio delle matrici C^T e $\hat{C}$ e presenta caratteristiche del tutto simili alle classi precedenti. Come affermato nel capitolo precedente, la prima delle matrici sopra riportate riguarda l'accoppiamento e può assumere due forme diverse, a seconda che sia in forma debole o forte, la seconda corrisponde al termine di salto della velocità presente nel modello per le fratture. Le due sono esattamente l'una la trasposta dell'altra nel caso di imposizione debole delle condizioni. Entrambe sono caratterizzate dal fatto di essere matrici fortemente sparse.
Nell'assemblaggio, per ciascuna frattura, si esegue un ciclo sulle sue facce ordinate e si calcola il termine da inserire.
Anche per questa classe si implementano i soliti due metodi e in aggiunta il metodo che permette di ottenere la trasposta.

4.4.7 Il termine noto e la classe Quadrature

Accanto alla matrice globale del sistema occorre implementare il termine noto, questo viene fatto solitamente nel main ed è costruito accorpando vari elementi:

```
Darcy::Vector RHS (dim);
// dim sono le dimensioni della matrice
RHS << g, -f, gf, -h, in;
```

Nel termine noto si hanno due contributi legati alle condizioni al bordo del mezzo poroso e delle fratture, g e gf rispettivamente:

```
auto g = MMM.getBCVector();
auto gf = T.getBCVector();
```

C'è un contributo, in, legato all'imposizione della conservazione del flusso alle intersezioni:

```
Darcy::Vector in = Eigen::VectorXd::Zero
  (myrmesh.getInterVertexes().size());
```

Vi sono due contributi, -f e -h, legati alle forzanti, Source nel mezzo poroso e SourceFr nella frattura:

```
Darcy::Quadrature<R2> quadrature (myrmesh,
       Darcy::CentroidQuadrature<R2>(),
       Darcy::MiddlePoint(),
       Darcy::VectorialTrapezium());

Darcy::Vector f(myrmesh.getCellsVector().size());
f = quadrature.CellIntegrate (Source);
```

```
Darcy::Vector h=Eigen::VectorXd::Zero
  (myrmesh.getFractureFacetsVector().size());
buildSourceFracture (sourceFr, myrmesh,
  quadrature, h);
```

Per calcolare opportunamente questi termini vengono utilizzate delle formule di quadratura, implementate nella classe template QuadratureRule. Si nota infatti che nella costruzione di quadrature vengono passati tre argomenti: il primo riguarda la formula di quadratura per calcolare l'integrale approssimato della sorgente nel mezzo poroso, il secondo per fare lo stesso nella frattura e il terzo per calcolare un'eventuale interpolazione del flusso nel caso di valutazione dell'errore. In realtà esistono due costruttori nella classe Quadrature che differiscono per la presenza o meno della formula di quadratura da utilizzare per il calcolo dell'interpolazione della soluzione esatta del flusso.
Il contributo delle forzanti nelle fratture al termine noto viene calcolato mediante la funzione buildSourceFracture in cui per ciascuna frattura:

- si interpreta la sorgente letta da file,
- si calcola l'integrale utilizzando la formula di quadratura stabilita in precedenza,
- si riempie la parte di vettore corrispondente alla frattura considerata che andrà a contribuire al termine noto.

La classe Quadrature svolge un ruolo importante per il calcolo degli errori in quanto, oltre ad integrali, consente di calcolare norme ed interpolazioni delle funzioni esatte mediante i metodi:

```
Vector CellApprox (const std::function
       <D_Real(Generic_Point)>& func);
Vector FractureApprox (const std::function
       <D_Real(Generic_Point)>& func);
D_Real L2Norm (const Vector& Integrand);
D_Real L2NormFrac (const Vector& Integrand);

Vector VectorFaceApprox (
 const std::function<D_Real(Generic_Point)>& funcx,
 const std::function<D_Real(Generic_Point)>& funcy);
```

Quest'ultimo metodo permette di calcolare l'interpolazione della componente normale di una funzione vettoriale le cui componenti sono funcx e funcy.

La struttura della classe Quadrature è cambiata rispetto alle versioni precedenti in quanto si è introdotto il calcolo dell'errore del flusso nel mezzo

poroso e della pressione nella frattura. Anche la classe template QuadratureRule è stata ampliata con l'aggiunta di nuove regole di quadrature, tra cui quella del punto medio e del trapezio.

4.5 Il main

Dopo aver analizzato nel dettaglio le matrici derivanti dalla discretizzazione mediante Differenze Finite Mimetiche e la loro implementazione, si costruirà il sistema per calcolare pressione e flusso nel mezzo e nelle fratture. Novità importante è la possibilità di passare tutti i parametri del problema da file. La possibilità di cambiare solo ed esclusivamente il file di dati permette di eseguire casi test e verifiche molto rapidamente poiché non si deve ricompilare ogni volta il codice. Si analizzerà ora la struttura di un tipico file main, per prima cosa ci si occupa della lettura da file delle informazioni che riguardano il mezzo poroso.

```
auto dataFile=readOptions(argc,argv);
auto parameters=readParametersFromFile(dataFile);
```

La variabile parameters è caratterizzata dalla seguente:

```
struct Parameters{
 bool weak; // Condizioni di accoppiamento
 D_Real size; // Massima lunghezza dei lati
 D_Real angle; // Ampiezza minima dell'angolo
 std::string fractureFileName;// File della frattura
 D_Real psi; // Parametro di accoppiamento
 std::string outputDir; // Cartella degli output
 std::string outputFileName; // Nome degli output
 std::string matrixDir; // Cartella delle matrici
 std::string method; // Metodo da usare
 std::string solverType; // Tipo di solutore
 UInt numEdges; // Numero dei lati del dominio
 std::vector<double> vertexes;// Vertici del dominio
 std::vector<std::string> K; // Permeabilità
 std::string source; // Forzante del mezzo
 std::vector<std::string> BC; // Funzione delle BC
 std::vector<std::string> BCType; // Tipo delle BC
 std::string exactPressure; // Pressione esatta
 std::vector<std::string> exactVelocity; // Velocità
                                         // esatta
};
```

Tale struct definisce la variabile di tipo Parameters che ingloba al suo interno un gran numero di informazioni, tutte lette da file grazie al metodo sopra

citato. Per la lettura del file è stato usato GetPot, molto pratico e semplice. Inoltre è necessario definire le funzioni che rappresentano le condizioni al bordo, le sorgenti e le soluzioni esatte, che possono dipendere dalle coordinate del punto e da alcuni parametri fisici del problema. É stato opportuno quindi introdurre un parser e delle ulteriori funzioni per interpretare correttamente le stringhe inserite nel file dei dati.

```
LifeV::scalar_type readFunction (
  const Geometry::Point2D & p, const std::string s);
LifeV::scalar_type readFunction (
  const Geometry::Point2D & p, const std::string s,
  const double Kt, const double df = 0.0,
  const double Kn = 0.0);

typedef Darcy::BCType type;
type convertBCType (const std::string s);
```

Dal precedente estratto di codice si osserva che si hanno due versioni della stessa funzione, differenti solo per i diversi parametri, si usa la prima quando non c'è dipendenza dagli aspetti fisici nella stringa da interpretare e la seconda qualora ci fosse. L'altra funzione riportata, convertBCType consente di convertire la stringa contenente informazioni sul tipo di condizioni al bordo in modo che queste diventino del tipo Darcy::BCType.

Dopo la lettura di questi parametri si costruisce il dominio:

```
std::vector<Geometry::Point2D> vertexes;
std::vector<Geometry::Domain<R2>::DomainBorder>
                                  domainborders;
buildDomain (parameters, vertexes, domainborders);
Geometry::Domain<R2 > domain (domainborders);
```

La funzione buildDomain è stata creata per alleggerire il main e si occupa della costruzione dei vertici e dei bordi del dominio. Infatti i vertici letti da file vanno a costituire dei punti, questi punti formano poi dei bordi, questi bordi determinano infine il dominio.
Successivamente vengono fissati gli altri parametri che sono letti da file.
Si impongono poi le condizioni al bordo del dominio, qui rientrano in gioco le funzioni necessarie per interpretare quanto inserito da file. Viene assemblata ciascuna condizione singolarmente e poi vengono accorpate per formare le Darcy::MediaBC<R2>. Anche in questo caso è stata creata una funzione per alleggerire il codice.

Dopo aver definito il mezzo occorre leggere da file le informazioni sulla frattura. In questo caso non viene utilizzato GetPot ma la lettura da file

tradizionale. La struttura della funzione che permette di leggere da file è la seguente:

```
bool readFractureFile (std::string const & filename,
  Geometry::FractureNetwork2D & fn,
  std::vector<std::string> & sourceF,
  std::vector<std::string> & exactSolution,
  std::vector<std::pair<std::string,
  std::string>> & BC );
```

Si ottengono informazioni sul numero di fratture, sui vertici e sui parametri fisici di queste, sulla forzante, sulla pressione esatta qualora sia nota e sulle condizioni al bordo.
Nel caso di discretizzazione con il metodo dei Volumi Finiti si utilizza la seguente funzione per leggere i dati da file:

```
Geometry::FractureNetwork2D fn;
std::vector<std::string> sourceFr;
std::vector<std::string> pExFrac;
Darcy::pImposedValues_t pImposedValues;

readFractureFile(parameters.fractureFileName,
    fn, sourceFr, pExFrac, pImposedValues);
```

L'unica differenza nella lettura del file riguarda il fatto che nel caso di Differenze Finite Mimetiche vengono imposte le condizioni agli estremi delle fratture, mentre nel caso di Volumi Finiti si può imporre la pressione di una frattura.

Successivamente si costruisce la triangolazione, la mesh2D e infine la Rigid_Mesh.
Con un procedimento analogo al precedente si stabiliscono le condizioni al bordo delle fratture ed in questo caso è fondamentale introdurre l'interpretazione anche dei parametri fisici.
Si procede poi con l'assemblaggio delle matrici. In tutti i test effettuati si utilizza il metodo fillMatrix per assemblare la matrice globale e si definisce il termine noto globale come mostrato in una delle sezioni precedenti. La risoluzione del sistema globale avviene mediante un solutore, definito nel file e scelto tra parecchi metodi forniti dalle Eigen. In particolare si userà UMFPACK dato che la matrice globale non è necessariamente simmetrica e definita positiva.
Infine occorre estrarre la pressione ed esportare la soluzione in un formato visualizzabile con altri software.

Per il calcolo dell'errore è stato necessario effettuare l'interpolazione delle soluzioni esatte negli spazi mimetici e utilizzare gli strumenti forniti dalla

classe Quadrature. I risultati numerici saranno mostrati nel capitolo seguente.
Inoltre è stata implementata la possibilità di scegliere da file se utilizzare il metodo dei Volumi Finiti o delle Differenze Finite Mimetiche nelle fratture. Il confronto sarà mostrato nel capitolo seguente.

In questo capitolo sono stati forniti dettagli sulla struttura del codice e sono state motivate le scelte implementative effettuate. Inoltre ci si è concentrati spesso sulla descrizione degli strumenti forniti dalle Eigen e sul loro utilizzo nel codice. Si è cercato di ampliare, generalizzare e adattare al meglio il codice già esistente alle novità introdotte sia dal punto di vista della discretizzazione che dell'analisi dei risultati.

Capitolo 5

Risultati Numerici

In questo capitolo verranno proposti alcuni casi per testare il codice descritto nel capitolo precedente in diverse situazioni. In particolare saranno analizzati modelli con fratture immerse ed intersecanti e verrà mostrato anche il caso di più fratture che si intersecano in uno stesso punto. Sarà interessante osservare l'andamento della pressione al variare della permeabilità e confrontare l'uso delle Differenze Finite Mimetiche con l'uso dei Volumi Finiti nelle fratture. Per uno dei casi test, tratto da [2], si calcoleranno gli errori di pressione e velocità e gli ordini di convergenza del metodo delle Differenze Finite Mimetiche.

5.1 Una frattura semi-immersa

Questo primo caso test è stato proposto in [1] e si concentra sull'analisi di una frattura semi-immersa al variare dei valori di permeabilità normale e tangenziale della frattura e delle condizioni al bordo. Tali variazioni consentiranno di osservare il comportamento di una frattura permeabile, di una impermeabile e di una semi-permeabile.

Il dominio del problema è il quadrato unitario in tutti e tre i casi e la frattura è verticale e si estende dal centro al bordo superiore del dominio:

$$\Omega = [0,1] \times [0,1] \qquad \text{e} \qquad \Gamma = \{0.5\} \times [0.5,1].$$

Il tensore di permeabilità del mezzo poroso coincide con la matrice identità. L'apertura della frattura è costante pari a $l_\Gamma = 0.01$ e le permeabilità della frattura saranno specificate nei tre casi.
Il parametro di accoppiamento ha valore $\xi = 0.75$ e le condizioni sono imposte in forma debole.
Nell'immagine (5.1) sono raccolte le informazioni sulle condizioni al bordo imposte nei tre casi che saranno analizzati.

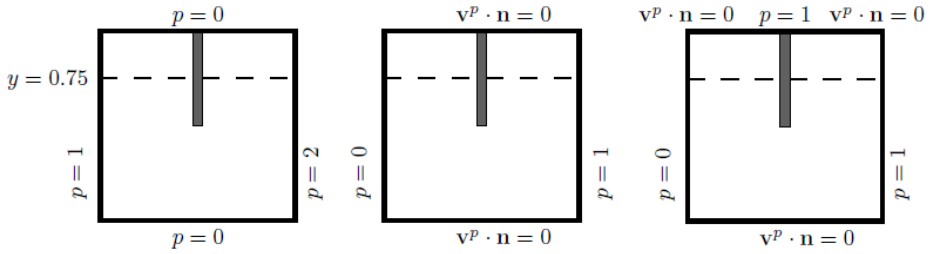


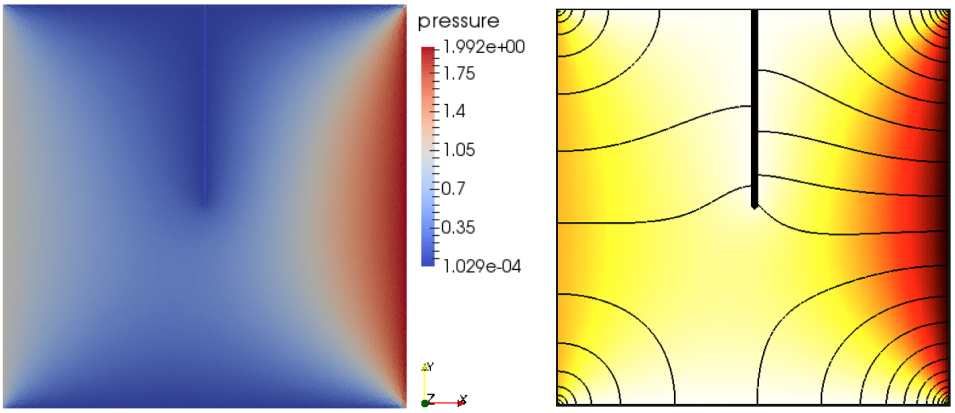
Figura 5.1: Tre casi test.

Caso test 5.1.1 (Frattura permeabile). Questo primo caso è caratterizzato da una permeabilità normale inferiore a quella tangenziale: $\hat{K_n} = 100$ e $\hat{K_\tau} = 10^6$, in questo modo $l_\Gamma \hat{K_\tau}$ e $\hat{K_n}/l_\Gamma$ sono dello stesso ordine di grandezza.
Sono state imposte condizioni sulla pressione su tutto il bordo, pari ad 1 su $\partial\Omega_{P,1} = \{0\} \times [0,1]$, a 2 su $\partial\Omega_{P,2} = \{1\} \times [0,1]$ ed omogenee su $\partial\Omega_{P,0} = [0,1] \times \{0,1\}$. Sull'estremo immerso della frattura è stata imposta una condizione omogenea sulla velocità e sull'altro omogenea sulla pressione. Il problema da risolvere è:

$$
\begin{cases}
\operatorname{div} \mathbf{u} = 0 & \text{in } \Omega \\
\mathbf{K}^{-1}\mathbf{u} + \nabla p = 0 & \text{in } \Omega \\
p = 0 & \text{su } \partial\Omega_{P,0} \\
p = 1 & \text{su } \partial\Omega_{P,1} \\
p = 2 & \text{su } \partial\Omega_{P,2} \\
\operatorname{div}_{\boldsymbol{\tau}} \hat{\mathbf{u}} = [\![\mathbf{u} \cdot \mathbf{n}_\Gamma]\!] & \text{in } \Gamma \\
\hat{K}^{-1}\, \hat{\mathbf{u}} + \nabla_{\boldsymbol{\tau}}\, \hat{p} = 0 & \text{in } \Gamma \\
\hat{\mathbf{u}} \cdot \boldsymbol{\tau} = 0 & \text{su } \partial\Gamma_{\hat{\mathbf{u}}} \\
\hat{p} = 0 & \text{su } \partial\Gamma_{\hat{p}} \\
\eta \{\mathbf{u} \cdot \mathbf{n}_\Gamma\} = [\![p]\!] & \text{su } \Gamma \\
\eta\, \xi_0 [\![\mathbf{u} \cdot \mathbf{n}_\Gamma]\!] = \{p\} - \hat{p} & \text{su } \Gamma.
\end{cases}
$$

La pressione risultante è riportata nelle figure (5.2) della pagina seguente. La velocità nel dominio non è facilmente rappresentabile, in quanto al contrario del caso degli Elementi Finiti, con le Differenze Finite Mimetiche non esistono funzioni di base. Esistono degli operatori di ricostruzione teorici ma la loro applicazione in casi generali esula dallo scopo di questo lavoro.

Nelle seguenti figure vengono confrontati i risultati ottenuti con il codice descritto nel capitolo precedente e quelli riportati nell'articolo [1] in cui è stato effettuato il medesimo caso test.



(a) Codice (b) Articolo

Figura 5.2: Pressione nel dominio.

La frattura è molto permeabile in direzione tangenziale e il fluido è attratto da essa, di conseguenza le linee di flusso sono dirette verso la frattura. Solitamente si ha un flusso di massa non continuo attraverso una frattura altamente permeabile.
Grazie ai filtri di Paraview la figura (5.2(a)) può essere così rivista, evidenziando il minimo di pressione vicino alla frattura:

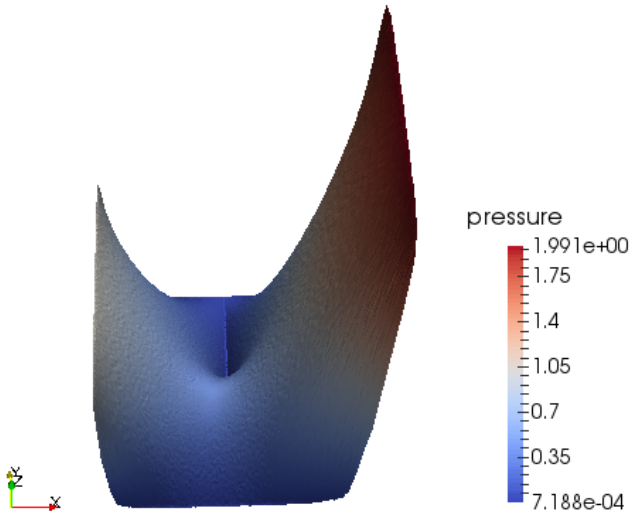
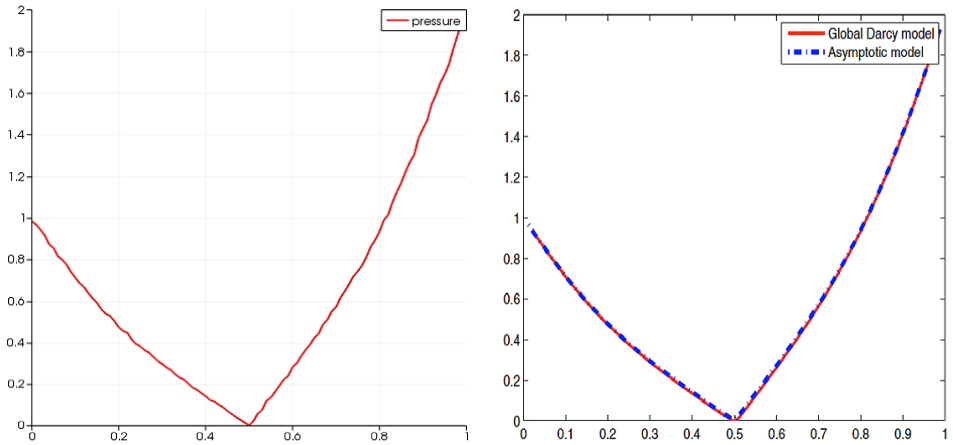


Figura 5.3: Pressione nel dominio.

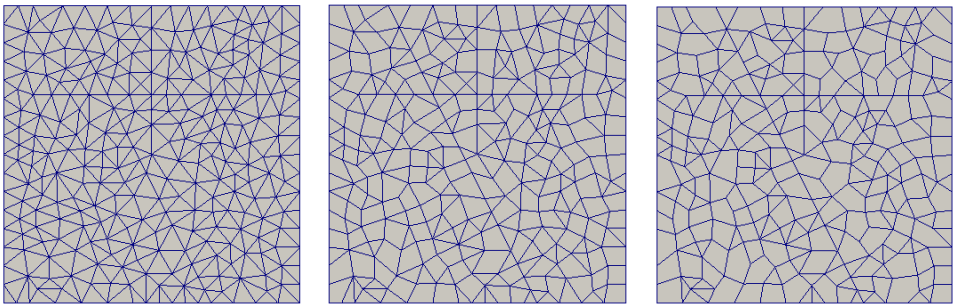


(a) Codice (b) Articolo

Figura 5.4: Pressione sulla linea $y = 0.75$.

Nei grafici (5.4) si riporta l'andamento della pressione in funzione della coordinata x lungo la linea $\{(x, y) \in \Omega : \{y = 0.75 \text{ e } 0 < x < 1\}\}$. La figura (5.4(b)), relativa all'articolo [1], mostra un confronto tra i risultati ottenuti con il modello di Darcy d-dimensionale nella frattura e con il modello ridotto. In entrambi i grafici si osserva che in prossimità della frattura la pressione scende bruscamente a zero. Più precisamente la pressione lungo tutta la frattura si mantiene costante e pari a zero.

Osservazione 5.1.1. *Nella maggior parte dei casi test proposti sarà utilizzata una griglia di tipo misto, formata a partire da una griglia triangolare unendo in modo casuale coppie di elementi per formare dei quadrilateri. I nuovi elementi formati possono essere uniti ancora ad un triangolo, in questo modo nella mesh ci saranno elementi con tre, quattro e cinque lati.*



(a) 3 (b) 3 o 4 (c) 3, 4 o 5

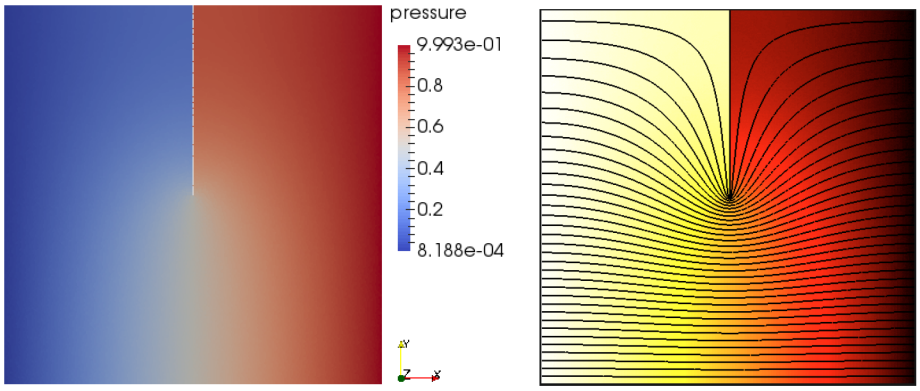
Figura 5.5: Tipi di griglie (i numeri indicano quanti lati hanno gli elementi che ne fanno parte).

I comandi che permettono di fare ciò sono gli ultimi due di quelli riportati

di seguito per la costruzione della Mesh2D:

```
std::unique_ptr<Geometry::Mesh2D>
        mesh (new Geometry::Mesh2D);
mesh -> addTriangulation (*Triang);
mesh -> genericUnifyTriangleMFD ();
mesh -> genericUnifyCellMFD ();
```

Caso test 5.1.2 (Frattura impermeabile). Questo secondo caso è caratterizzato dalla permeabilità nella frattura inferiore alla permeabilità del mezzo poroso, infatti $\hat{K_n} = \hat{K_\tau} = 10^{-7}$ e di conseguenza $l_\Gamma \hat{K_\tau} = 10^{-9}$ e $\hat{K_n}/l_\Gamma = 10^{-5}$.
Sono state imposte condizioni sulla velocità omogenee su $\partial\Omega_{\mathbf{u}} = [0,1]\times\{0,1\}$ e sulla pressione omogenee su $\partial\Omega_{P,0} = \{0\} \times [0,1]$ e pari ad 1 su $\partial\Omega_{P,1} = \{1\} \times [0,1]$. Agli estremi della frattura sono state imposte condizioni sulla velocità omogenee, in quanto un estremo è immerso e l'altro tocca il bordo superiore su cui è imposta una condizione analoga.
Nei grafici riportati in seguito si osserva che la pressione non è continua attraverso la frattura e si viene infatti a creare un salto di pressione. Nella frattura stessa la pressione si mantiene circa costante pari a 0.5. Dalla figura (5.6(b)) si nota che le linee di flusso, a differenza del caso precedente, non entrano nella frattura ma la evitano, ci si trova infatti di fronte al caso di una frattura impermeabile.



(a) Codice (b) Articolo

Figura 5.6: Pressione nel dominio.

Si osserva che i risultati ottenuti con il codice descritto nel capitolo precedente e quelli riportati in [1] sono del tutto coerenti anche per questo caso.

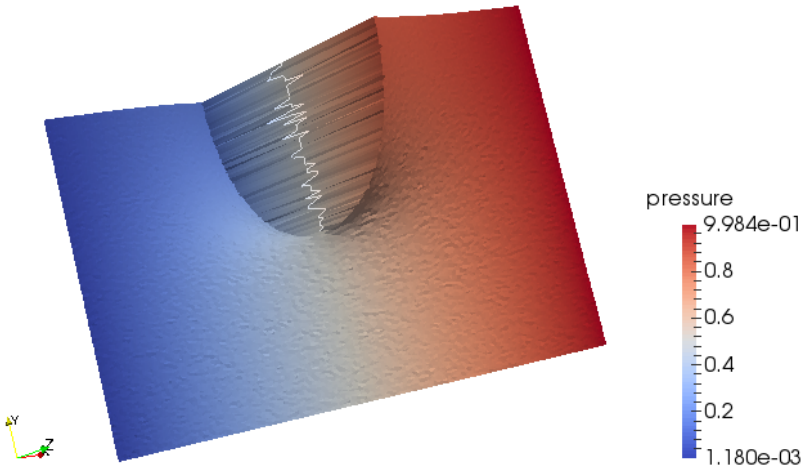
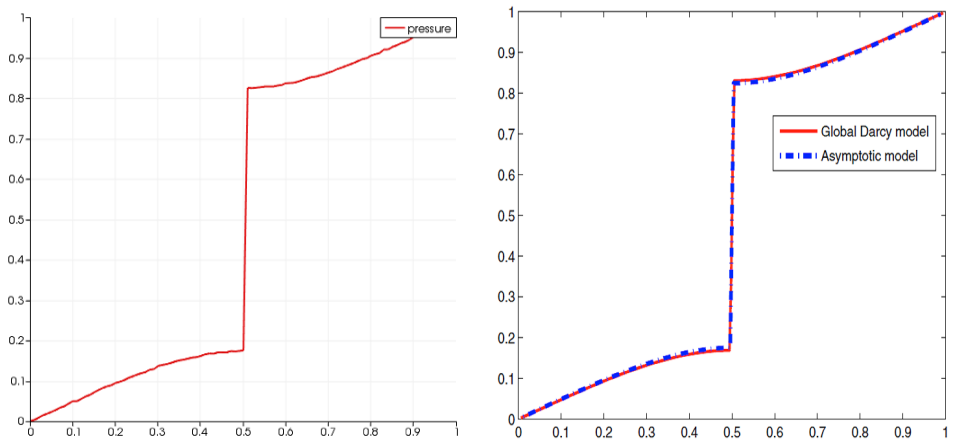


Figura 5.7: Pressione nel dominio.

La pressione nella frattura è una funzione definita a tratti per il tipo di discretizzazione scelto. In fase di visualizzazione viene interpolata dal software grafico Paraview e nelle immagini appare come una linea spezzata.

I grafici seguenti mostrano l'andamento della pressione lungo la linea $\{(x, y) \in \Omega : \{y = 0.75 \text{ e } 0 < x < 1\}\}$ ed evidenziano la presenza del salto di pressione.

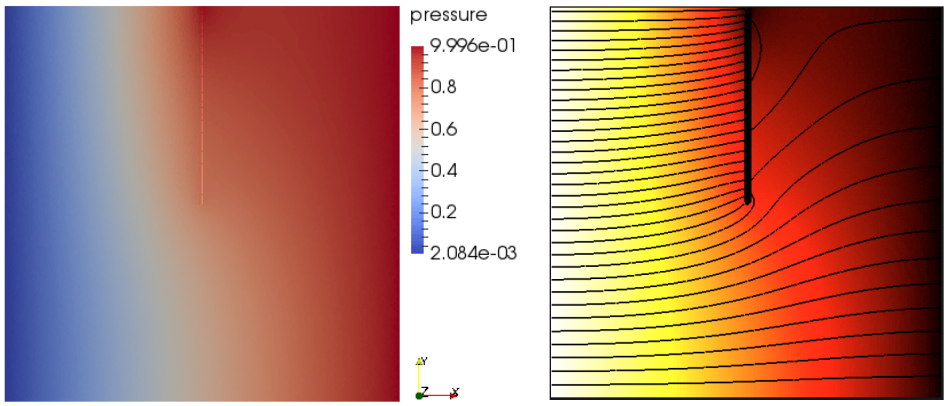


(a) Codice (b) Articolo

Figura 5.8: Pressione sulla linea $y = 0.75$.

Caso test 5.1.3 (Frattura semi-permeabile). In questo caso si verifiche-

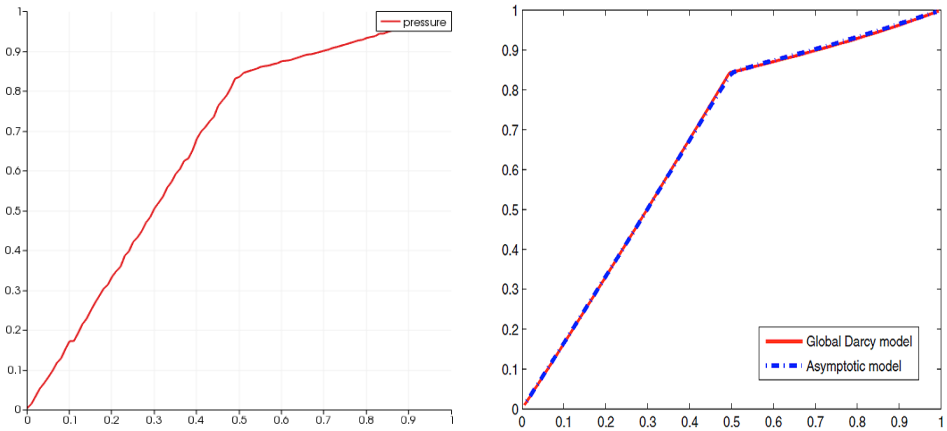
rà una situazione intermedia tra le precedenti, i parametri sono posti pari a $\hat{K}_n = \hat{K}_\tau = 100$, in modo che $\mathbf{K}$ e $l_\Gamma \hat{K}_\tau$ siano dello stesso ordine di grandezza e che il rapporto tra $\hat{K}_n / l_\Gamma$ e i valori di $\mathbf{K}$ sia alto.
Le condizioni imposte ai bordi del dominio sono identiche a quelle del caso precedente; per quanto riguarda la frattura, sull'estremo immerso è applicata una condizione sulla velocità omogenea e su quello in prossimità del bordo superiore del dominio la pressione è posta ad 1.
In questo caso si ottengono i seguenti risultati:



(a) Codice (b) Articolo

Figura 5.9: Pressione nel dominio.

Nella frattura il valore della pressione cresce da 0.78 a 1 circa. L'andamento della pressione lungo la linea $\{(x, y) \in \Omega : \{y = 0.75 \text{ e } 0 < x < 1\}\}$ è:



(a) Codice (b) Articolo

Figura 5.10: Pressione sulla linea $y = 0.75$.

5.2 Intersezioni tra fratture

In questa sezione si analizzerà il caso di intersezioni tra fratture, in particolare si mostrerà che più fratture possono incontrarsi in uno stesso punto. Nel caso test trattato il dominio è un quadrato unitario in cui sono presenti tre fratture con due intersezioni. La prima frattura si estende dal punto $(0.2, 1)$ a $(0.7, 0)$, la seconda da $(0.5, 0.5)$ a $(0.5, 1)$ e la terza da $(0.1, 0.7)$ a $(0.9, 0.7)$. Quindi la prima frattura attraversa il dominio, la seconda è semi-immersa e la terza è completamente immersa.
I bordi del dominio sono divisi in $\partial\Omega_{\mathbf{u}} = \{0, 1\} \times [0, 1]$ su cui si applica una condizione sulla velocità omogenea e in $\partial\Omega_P = [0, 1] \times \{0, 1\}$ dove si pone la pressione pari a $p(x, y) = y$. Sugli estremi delle fratture sono imposte condizioni sulla velocità omogenee nel caso siano immersi, altrimenti delle condizioni coerenti con quelle imposte al bordo. Per esempio nel vertice $(0.5, 1)$ è stata applicata una condizione sulla pressione; si è rivelato quindi utile introdurre nel codice la possibilità di imporre condizioni non solo sulla velocità ma anche sulla pressione agli estremi delle fratture.
Il tensore di permeabilità nel mezzo è sempre pari alla matrice identità, quello nella frattura cambierà nei vari casi proposti. La frattura presenta un'apertura $l_\Gamma = 0.01$. Il parametro di accoppiamento ha valore $\xi = 0.75$ e le condizioni sono imposte in forma debole.

Caso test 5.2.1 (Presenza di una forzante). In questo primo caso si suppone di avere una forzante nel mezzo poroso pari a $f(x, y) = 4$.
Il tensore di permeabilità nella frattura viene posto pari alla matrice identità, in modo da simulare un mezzo omogeneo senza fratture. Per la pressione si ottiene:

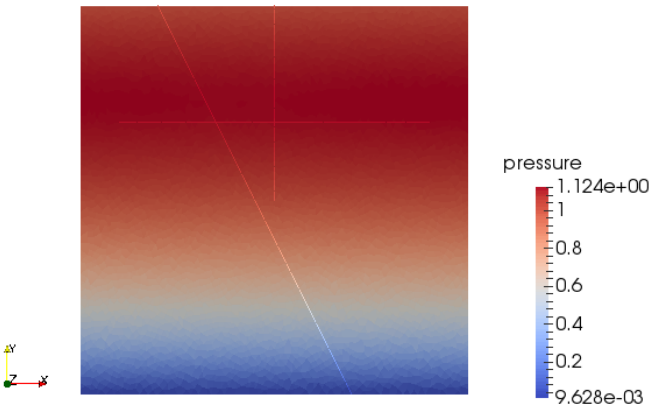


Figura 5.11: Pressione nel dominio.

La visione in 3D grazie ai filtri di Paraview consente di osservare meglio cosa succede in presenza di una forzante.

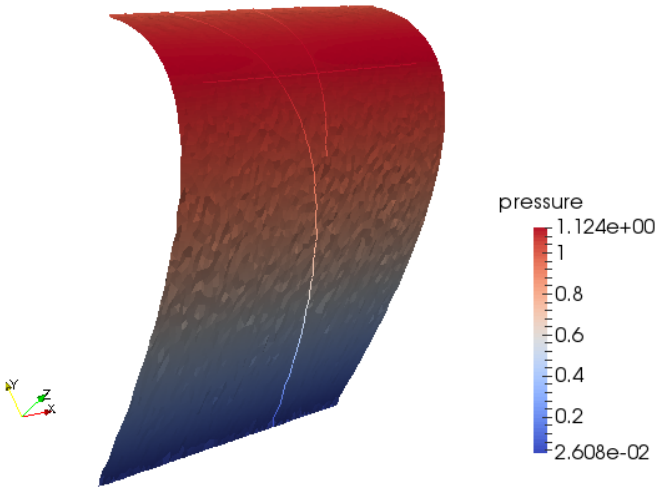


Figura 5.12: Pressione nel dominio.

Caso test 5.2.2 (Permeabilità diverse). In questo secondo caso si pone la forzante pari a zero e si fissano le permeabilità della frattura orizzontale pari a $\hat{K}_\tau = \hat{K}_n = 10^{-7}$. La frattura orizzontale risulta così impermeabile. Le permeabilità, tangenziali e normali, delle altre due fratture restano pari ad 1. Si ottengono i seguenti risultati:

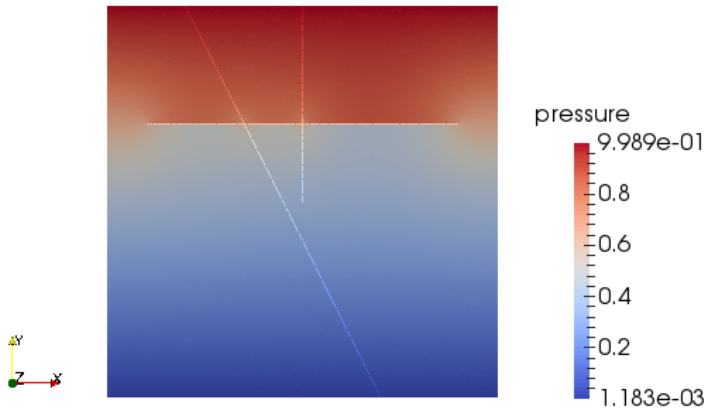


Figura 5.13: Pressione nel dominio.

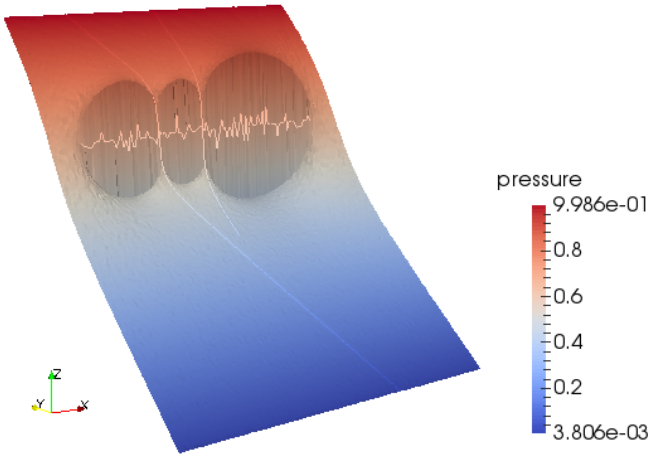


Figura 5.14: Pressione nel dominio.

5.2.1 Fratture intersecanti in un unico punto

Si considererà ora il caso particolare di tre fratture che si intersecano in un unico punto. Si tratta di un test del tutto simile al precedente, con stesso dominio, dati e condizioni al bordo. Leggermente diversa è la posizione delle fratture, infatti la prima si estende dal punto $(0.1, 0.9)$ al $(0.9, 0.1)$, la seconda da $(0.5, 0.4)$ a $(0.5, 1)$ e la terza da $(0.1, 0.5)$ a $(0.9, 0.5)$. La frattura orizzontale ha valori di permeabilità tipici di una frattura impermeabile, $\hat{K}_\tau = \hat{K}_n = 10^{-7}$ mentre le altre hanno permeabilità unitarie. Sui bordi immersi delle fratture si impongono condizioni sulla velocità omogenee mentre si pone la pressione pari ad 1 sul vertice $(0.5, 1)$. Il risultato è il seguente:

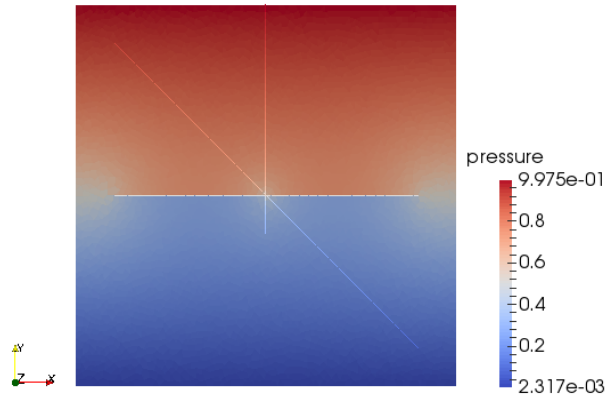


Figura 5.15: Tre fratture intersecanti in un unico punto.

Tale test mostra che il codice è adatto a studiare l'andamento della pressione anche in un caso come questo, con più fratture intersecanti in un unico punto.

5.3 Confronto tra Volumi Finiti e Differenze Finite Mimetiche

In questa sezione verrà proposto un caso test su un network di fratture piuttosto complicato, formato da undici fratture immerse nel quadrato unitario. Inoltre sarà effettuato un confronto tra la discretizzazione con i Volumi Finiti e con le Differenze Finite Mimetiche nelle fratture.

Il dominio del test effettuato è il quadrato unitario e il network di fratture si può osservare in una delle figure riportate in seguito.
La permeabilità del mezzo poroso è pari alla matrice identità mentre quella della frattura viene fatta variare. La larghezza della frattura è sempre pari a $l_\Gamma = 0.01$. Le condizioni di accoppiamento sono imposte ancora in forma debole e il parametro di accoppiamento è pari a $\xi = 0.75$.
Ai bordi del dominio sono imposte condizioni omogenee sulla pressione e agli estremi delle fratture condizioni omogenee sulla velocità essendo tutte immerse. In questo caso si ha una forzante nel mezzo poroso pari a:

$$f := \begin{cases} 10 & \text{se} \quad (x-0.10)^2 + (y-0.10)^2 < 0.04 \\ -10 & \text{se} \quad (x-0.90)^2 + (y-0.90)^2 < 0.04. \end{cases}$$

Nel caso di permeabilità nella frattura pari alla matrice identità e discretizzazione con le Differenze Finite Mimetiche si ottiene:

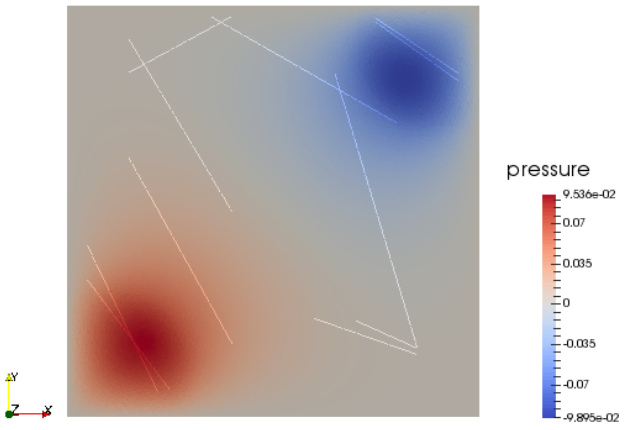
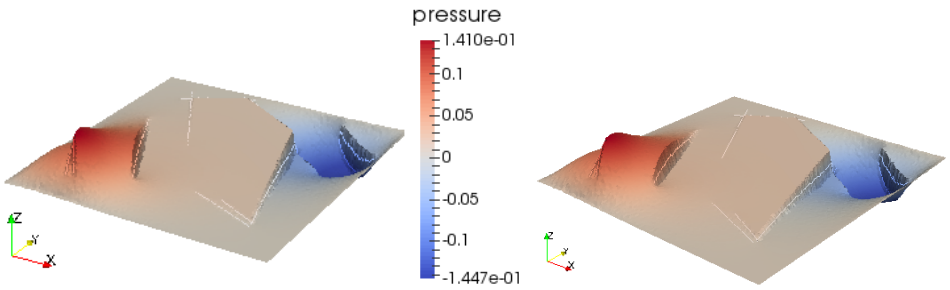


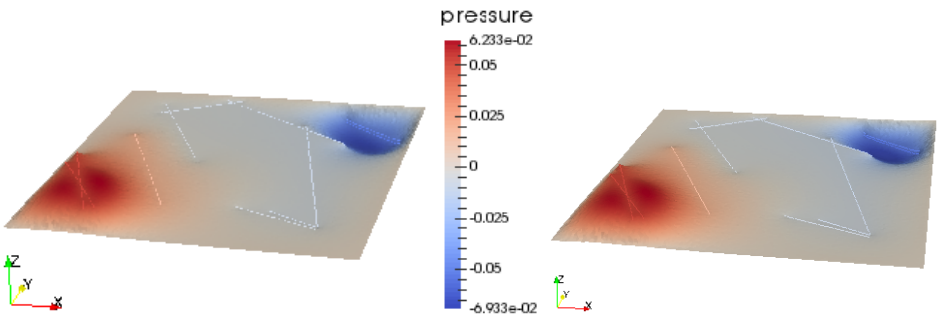
Figura 5.16: Permeabilità: $\hat{K_n} = 1$ e $\hat{K_\tau} = 1$.

Si osserva che i risultati ottenuti con le Differenze Finite Mimetiche nelle fratture sono coerenti con quelli ottenuti con i Volumi Finiti al variare della permeabilità nelle fratture.



(a) Differenze Finite Mimetiche (b) Volumi Finiti

Figura 5.17: Permeabilità: $\hat{K_n} = \hat{K_\tau} = 10^{-3}$.



(a) Differenze Finite Mimetiche (b) Volumi Finiti

Figura 5.18: Permeabilità: $\hat{K_n} = \hat{K_\tau} = 10^{3}$.

Osservazione 5.3.1. *Si è cercato di verificare a livello numerico i risultati teorici ottenuti precedentemente, in particolare affinché il problema in considerazione sia ben posto il parametro di accoppiamento ξ deve essere strettamente maggiore di un preciso valore. Sono stati effettuati casi test con valori del parametro prossimi allo zero e sono stati ottenuti comunque buoni risultati. Dal punto di vista teorico se ξ è inferiore a quel preciso valore si perde la coercività della forma bilineare con conseguente molteplicità di soluzioni in velocità, ma probabilmente lo schema numerico è in grado di individuare una di queste. Nel caso di $\xi = 0.5$, corrispondente ad assumere una variazione lineare della pressione trasversalmente alle fratture, si osserva invece in diversi casi un andamento differente della soluzione in pressione.*

5.4 Ordini di convergenza

In questa sezione si analizzerà, con qualche modifica, un caso test riportato nell'articolo [2] di cui è nota la soluzione esatta. In [2] la frattura attraversa il dominio mentre in questo test sarà immersa. Essendo nota la soluzione esatta sarà possibile calcolare gli errori e gli ordini di convergenza.

Il dominio del problema è il quadrato $\Omega = [-1, 1] \times [-1, 1]$ e la frattura è orizzontale ed immersa $\Gamma = [-0.9, 0.9] \times \{0\}$.
I tensori di permeabilità del mezzo poroso e della frattura sono pari alla matrice identità e l'apertura della frattura è $l_\Gamma = 0.01$.
Le condizioni di accoppiamento sono applicate in forma debole con parametro $\xi = 0.75$.
Inoltre è presente un termine sorgente nella frattura $\hat{f} = l_\Gamma \cos(x)$.
La pressione esatta è data da:

$$p = \begin{cases} \cos(x)\cosh(y) & \text{in } \Omega \\ \cos(x) & \text{su } \Gamma. \end{cases}$$

Prima di esaminare i risultati ottenuti ed in particolare gli ordini di convergenza è opportuno introdurre alcuni risultati teorici a supporto di quelli sperimentali. Innanzitutto per le stime dell'errore occorre utilizzare le norme indotte dai prodotti scalari degli spazi mimetici:

$$|||q_h|||^2_{\mathscr{P}_h} := [q_h, q_h]_{\mathscr{P}_h} = \sum_{P \in \Omega_h} |P||q_P|^2 \qquad \forall\, q_h \in \mathscr{P}_h$$
$$|||\mathbf{v}_h|||^2_{\mathscr{F}_h} := [\mathbf{v}_h, \mathbf{v}_h]_{\mathscr{F}_h} = \mathbf{v}_h^T \mathbf{M}\, \mathbf{v}_h \qquad \forall\, \mathbf{v}_h \in \mathscr{F}_h.$$

Inoltre è opportuno ricordare che l'interpolazione della soluzione esatta per il calcolo dell'errore viene effettuata secondo la teoria mimetica come una media su ciascuna cella per la pressione e come una media della componente normale su ciascuna faccia per la velocità.
In questa analisi verranno calcolati gli errori relativi di pressione nel mezzo poroso, velocità nel mezzo poroso e pressione nella frattura. In seguito si calcoleranno gli ordini di convergenza del metodo utilizzando gli strumenti forniti da Matlab. I risultati teorici al riguardo sono:

Teorema 5.4.1. *Sia $(\mathbf{u}, p)$ con $p \in H^2(\Omega)$, la soluzione esatta del problema e $(\mathbf{u}_h, p_h) \in \mathscr{F}_h \times \mathscr{P}_h$ la soluzione del problema discreto, allora valgono le seguenti stime:*

$$|||p_h - p^I|||_{\mathscr{P}_h} \leq C_1\, h\, ||p||_{H^2(\Omega)}$$
$$|||\mathbf{u}_h - \mathbf{u}^I|||_{\mathscr{P}_h} \leq C_2\, h\, ||p||_{H^2(\Omega)}$$

con C_1 e C_2 costanti positive indipendenti da h.

Teorema 5.4.2. *Sia* $(\mathbf{u}, p)$ *con* $p \in H^2(\Omega)$*, la soluzione esatta del problema, definito in un dominio* Ω *convesso e con condizioni di Dirichlet omogenee al bordo. Inoltre sia il tensore di diffusione* $\mathbf{K}$ *costante a tratti e il termine sorgente* f *appartenente ad* $H^1(\Omega)$*. Infine sia* $(\mathbf{u}_h, p_h) \in \mathscr{F}_h \times \mathscr{P}_h$ *la soluzione del problema discreto. Allora esiste una costante positiva* C *indipendente da* h *tale che:*

$$|||p_h - p^I|||_{\mathscr{P}_h} \leq C\, h^2 \left(||p||_{H^2(\Omega)} + |f|_{H^1(\Omega)} \right).$$

Per le dimostrazioni di questi teoremi si rimanda a [4].

In virtù dei risultati teorici riportati precedentemente sono stati effettuati dei confronti tra diversi tipi di griglia, triangolare o mista e tra diverse possibilità di imposizione delle condizioni di accoppiamento, in forma forte o debole. Il caso test, al variare degli aspetti indicati, è stato effettuato più volte, raffinando progressivamente la griglia. Questa è stata definita con un passo spaziale $h = 1/N$ e il valore di N è stato raddoppiato ad ogni nuova simulazione $N = 2, 4, 8, 16, 32, 64$.

Innanzitutto è stato calcolato e analizzato l'errore della pressione nel mezzo poroso. I dati ottenuti nelle varie simulazioni sono riportati nei seguenti grafici:

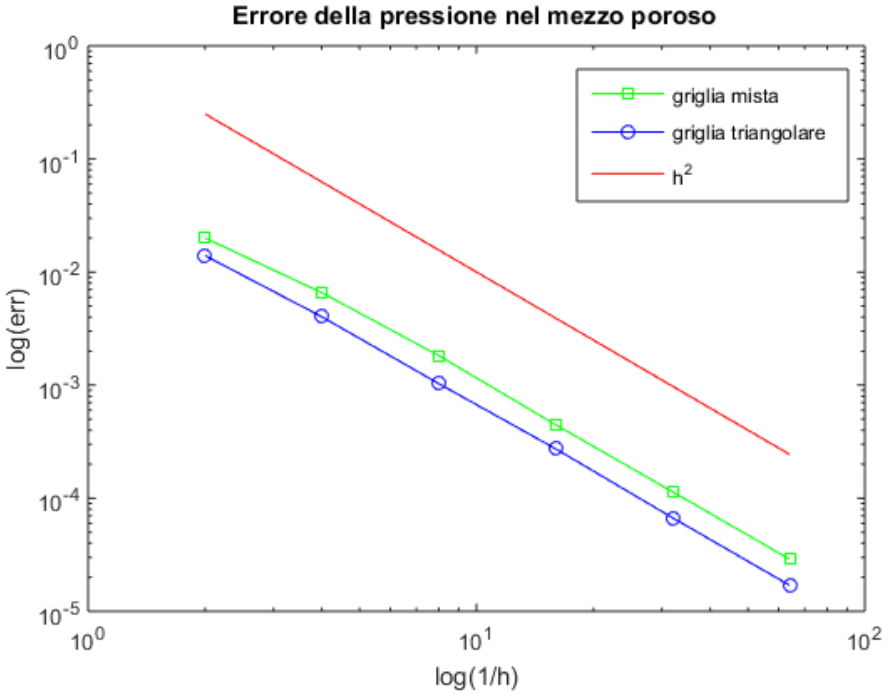


Figura 5.19: Errore relativo della pressione in funzione della dimensione della mesh: confronto del tipo di griglia.

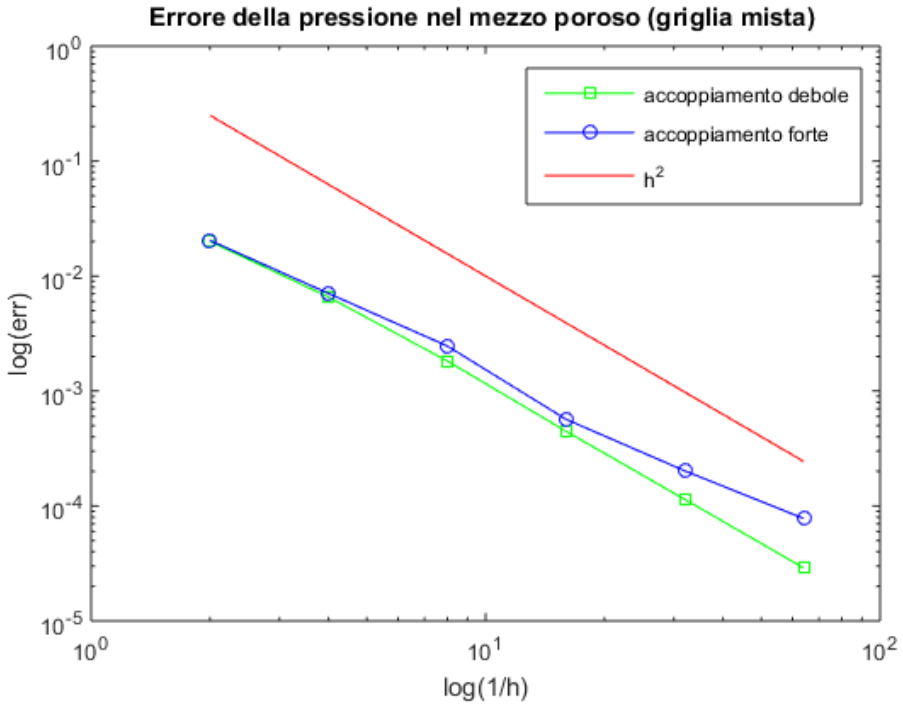


Figura 5.20: Errore relativo della pressione in funzione della dimensione della mesh: confronto del tipo di accoppiamento.

Nella seguente tabella sono indicati gli ordini di convergenza medi ottenuti dai dati dei grafici precedenti:

Accoppiamento	Debole	Forte
Ordine	1.9070	1.6481

Griglia	Mista	Triangolare
Ordine	1.9070	1.9443

Tabella 5.1: Tassi di convergenza medi della pressione.

Si osserva che non c'è una grande differenza tra l'ordine di convergenza con l'utilizzo di una griglia mista e quello con una triangolare, mentre è immediato notare che l'accoppiamento in forma debole favorisce una convergenza migliore di quello forte. Si suppone che ciò sia dovuto al fatto che con l'imposizione delle condizioni in forma debole le matrici C^T e $\hat{C}$ del sistema (3.25) sono esattamente una la trasposta dell'altra, cosa che non accade con l'accoppiamento in forma forte.
Inoltre le condizioni di accoppiamento applicate in forma debole non comportano la cancellazione di righe della matrice e l'inserimento degli elementi che le riproducono ma l'aggiunta di alcuni contributi a termini della matrice già presenti, che non vengono quindi cancellati.

E' stato analizzato anche l'errore della velocità, ciò è stato possibile calcolando la soluzione esatta a partire dalla pressione esatta riportata nella (5.4). Inoltre sfruttando il fatto che la frattura sia orizzontale e che la componente della velocità nella frattura in direzione verticale sia nulla si è constatato che l'interpolazione della soluzione esatta nelle facce sdoppiate è nulla.
Avendo quindi la soluzione esatta sono stati calcolati errori ed ordini di convergenza, giungendo ai seguenti risultati:

Accoppiamento	Debole	Forte
Ordine	1.0662	0.8047

Griglia	Mista	Triangolare
Ordine	1.0662	1.3471

Tabella 5.2: Tassi di convergenza medi della velocità.

I risultati ottenuti sono in accordo con quanto atteso dalla teoria, infatti si ottiene un ordine di convergenza pari circa ad uno.
Si osserva anche in questo caso che l'accoppiamento debole fornisce una migliore convergenza rispetto a quello forte, per la stessa motivazione data in precedenza. In questo caso risulta più performante l'utilizzo di una griglia triangolare piuttosto che mista.
I dati analizzati sono riportati nei grafici seguenti:

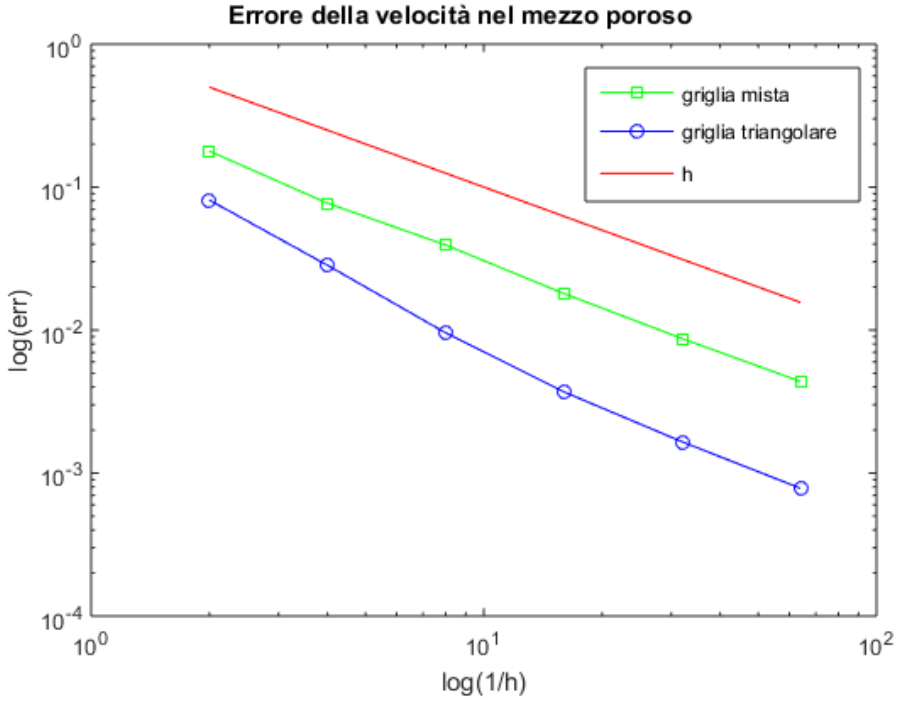


Figura 5.21: Errore relativo della velocità in funzione della dimensione della mesh: confronto del tipo di griglia.

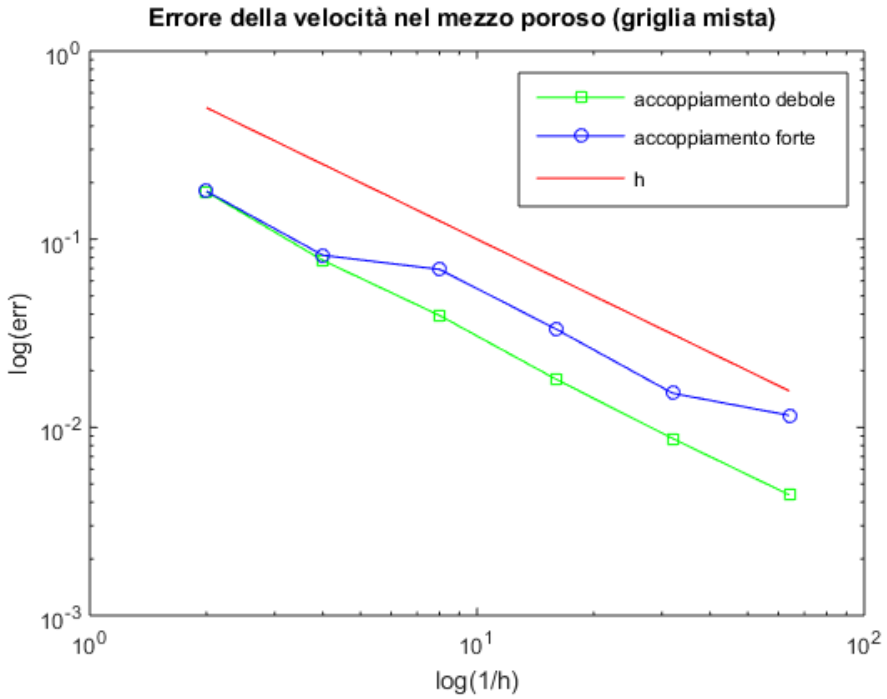


Figura 5.22: Errore relativo della velocità in funzione della dimensione della mesh: confronto del tipo accoppiamento.

Il calcolo degli errori e degli ordini di convergenza è stato effettuato anche per la pressione nella frattura. Per quanto affermato nei capitoli precedenti le Differenze Finite Mimetiche nel caso monodimensionale coincidono con gli Elementi Finiti. Quindi la discretizzazione *cell-based* del campo di pressione nella frattura coincide con la tecnica degli Elementi Finiti di ordine zero. La stima sull'errore con l'utilizzo di questi ultimi prevederebbe un andamento lineare al variare di h. Ciò è coerente con quanto ottenuto, come si osserva nella seguente tabella.

Accoppiamento	Debole	Forte
Ordine	2.0081	1.1881

Griglia	Mista	Triangolare
Ordine	2.0081	1.8083

Tabella 5.3: Tassi di convergenza medi della pressione nella frattura.

Si nota che la griglia mista determina una miglior convergenza rispetto a quella triangolare. Inoltre, coerentemente con quanto osservato finora, l'accoppiamento debole risulta migliore di quello forte. Tali risultati sono mostrati anche nei grafici della pagina seguente. Si ottengono quindi risultati di convergenza addirittura migliori rispetto a quelli previsti dalla teoria, si parla anche in questo caso di superconvergenza della pressione.

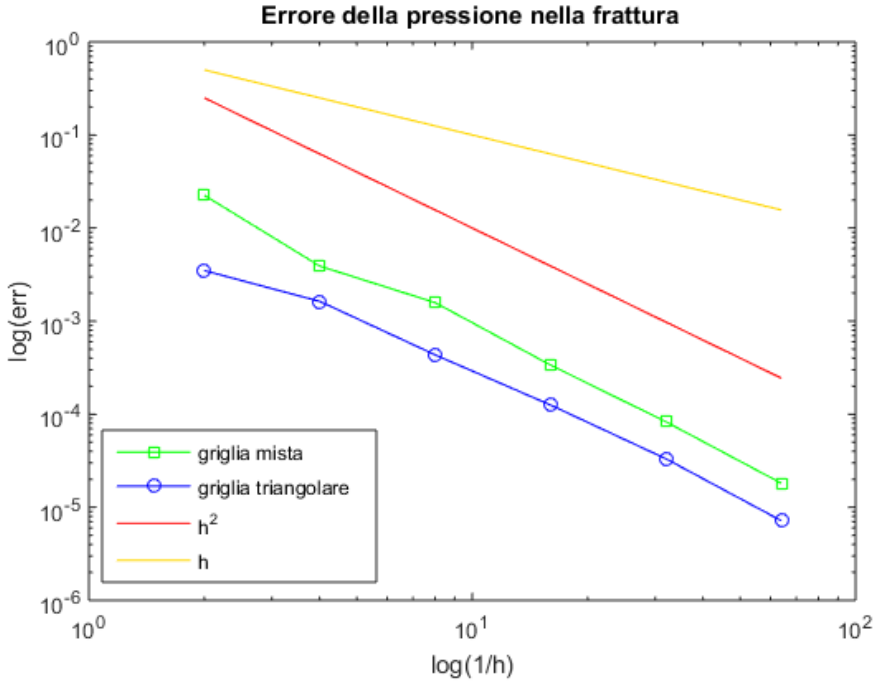


Figura 5.23: Errore relativo della pressione nella frattura in funzione della dimensione della mesh: confronto del tipo di griglia.

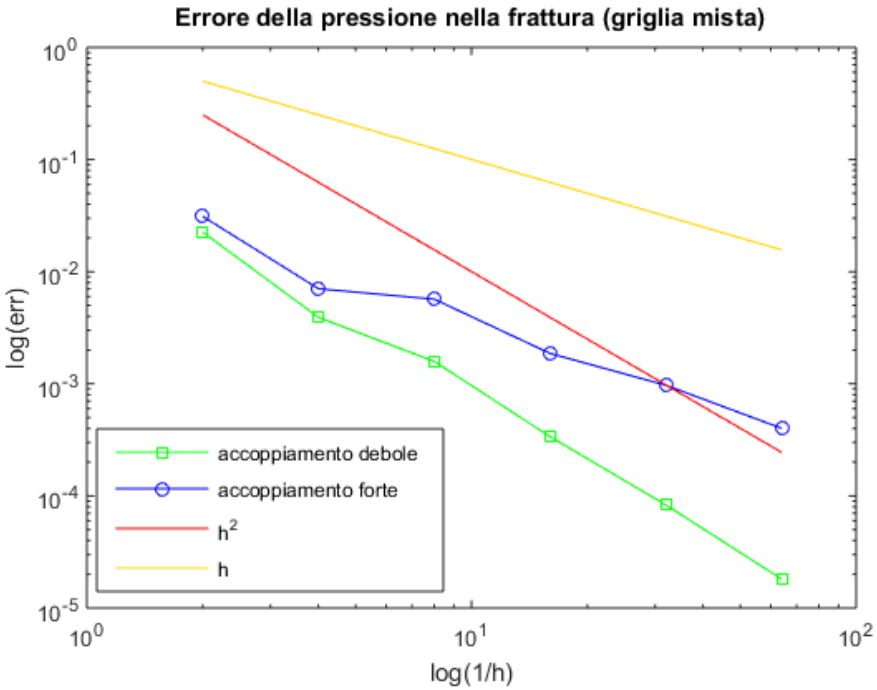


Figura 5.24: Errore relativo della pressione nella frattura in funzione della dimensione della mesh: confronto del tipo accoppiamento.

L'analisi fatta in questo capitolo ha permesso di fare chiarezza su quali casi test sia possibile affrontare con il codice descritto nel capitolo precedente. Inoltre si è verificato che l'andamento dell'errore delle quantità calcolate è coerente con quanto atteso dalla teoria. Si è potuto constatare, nell'ultimo test, che è persino migliore. Sono stati effettuati altri test per verificare la correttezza del codice, ma sono stati descritti quelli più significativi ed esplicativi. Per ulteriori test si rimanda al codice direttamente.

Conclusioni e sviluppi futuri

In questa tesi è stato analizzato il flusso in un mezzo poroso fratturato sia da un punto di vista teorico, con la dimostrazione di alcune proprietà del modello utilizzato per descriverlo, che da un punto di vista numerico, con lo sviluppo di un codice per il calcolo della pressione e della velocità del fluido.

I lavori [13] e [1] sono stati fondamentali per lo studio teorico del modello che descrive il flusso di un fluido in un mezzo poroso fratturato. È stato utilizzato il modello di Darcy in forma mista sia nella matrice solida che nelle fratture e i due modelli sono stati poi opportunamente accoppiati, come descritto nel Capitolo 1. Dopo aver introdotto il modello è stato proposto un setting funzionale adatto a descrivere il problema, in particolare nel caso di una frattura immersa nel dominio e di un network di fratture immerse. In seguito è stata derivata la formulazione debole del problema accoppiato e sono stati identificati i corretti spazi funzionali in cui cercare la soluzione debole.

Uno degli obiettivi raggiunti nella tesi è stata la dimostrazione della buona posizione del problema di Darcy in forma mista sia nel mezzo che nella frattura. Nel caso di una frattura che attraversa il dominio tale dimostrazione è stata proposta in [13], in questa tesi è stato invece analizzato il caso più complesso di una frattura immersa nel dominio. Si suppone inoltre che sia possibile estendere la dimostrazione al caso di un network di fratture, ma tale generalizzazione è ancora in fase di elaborazione. È stato inoltre esteso il risultato al caso in cui il parametro di chiusura $\xi \leq \frac{1}{2}$ e si è verificato numericamente che effettivamente la condizione $\xi > \frac{1}{2}$ non è necessaria.

In seguito il problema continuo è stato discretizzato con le Differenze Finite Mimetiche. In [2] è stata trattata la discretizzazione con le Differenze Finite Mimetiche nel mezzo poroso e con i Volumi Finiti nel network mentre in questa tesi sono state utilizzate le Differenze Finite Mimetiche in entrambi. Questo è un altro degli aspetti interessanti, le Differenze Finite Mimetiche sono infatti una tecnica di discretizzazione particolarmente flessibile e capace di preservare le proprietà del modello continuo. Inoltre consentono l'utilizzo di griglie con elementi generici che possono quindi adattarsi bene alle frat-

ture ed hanno ottime proprietà di convergenza.

È stato poi sviluppato, a partire da un lavoro già esistente, il codice in C++11 per il calcolo della pressione e del flusso di un fluido che attraversa un mezzo poroso fratturato. Il codice si occupa principalmente della costruzione della mesh, dell'implementazione del sistema derivante dalla discretizzazione e della sua risoluzione.
L'introduzione della formulazione mista per il problema di Darcy nella frattura e la relativa discretizzazione con le Differenze Finite Mimetiche hanno determinato l'implementazione delle classi per l'assemblaggio delle nuove matrici e per la gestione delle condizioni al bordo oltre che del dominio anche delle fratture. Inoltre a differenza del codice con discretizzazione mediante i Volumi Finiti nelle fratture, si è rivelato necessario imporre esplicitamente delle condizioni all'intersezione tra fratture.

Sono stati effettuati molti casi test per verificare la validità del codice e le diverse situazioni che può trattare. È infatti adatto a fratture immerse, semi-immerse o che attraversano il dominio, inoltre è in grado di trattare network di fratture complicati con intersezioni tra due o più fratture. Grazie all'introduzione della lettura da file dei dati del problema è diventato molto più agevole eseguire un test e soprattutto modificare i dati per osservare diversi comportamenti. Sono stati valutati gli errori di pressione nel mezzo e nella frattura e di flusso nel mezzo e gli ordini di convergenza sono perfettamente in accordo con quelli attesi secondo la teoria. Si è constatato che l'accoppiamento in forma debole risulta migliore rispetto a quello in forma forte e che i risultati ottenuti con la tecnica delle Differenze Finite Mimetiche sono in accordo con quelli ottenuti con i Volumi Finiti nelle fratture.

Innanzitutto una possibile estensione di questa tesi riguarda la generalizzazione del modello utilizzato, infatti sono state fatte numerose ipotesi prima di giungere alla definizione del problema. Si potrebbero considerare l'effetto della gravità nella legge di Darcy, le interazioni dovute ad un fluido bifase o l'evoluzione temporale di un fluido comprimibile. Si potrebbe anche generalizzare il modello per la descrizione del flusso nelle fratture, per esempio non trascurando lo spessore di queste geometricamente, ossia facendo un modello equidimensionale con elementi molto anisotropi nelle fratture, sfruttando la flessibilità delle Differenze Finite Mimetiche rispetto alla discretizzazione spaziale.
Ancora più interessante sarebbe estendere il modello ed il metodo numerico utilizzato al caso di un mezzo poroso fratturato tridimensionale. L'estensione al 3D presenta una complessità maggiore sia dal punto di vista puramente geometrico e implementativo, sia modellistico. In 3D infatti l'intersezione tra due fratture non è un singolo punto, ma una linea, e non è tutt'ora chiaro se la chiusura corretta del modello richieda la soluzione di un flusso di Darcy

bidimensionale lungo tale intersezione.
Finora nel codice è stato utilizzato un metodo diretto per risolvere il sistema corrispondente alla formulazione algebrica del problema. Sarebbe interessante introdurre la risoluzione del sistema con dei metodi iterativi. In questo caso risulterebbe particolarmente interessante la definizione di un opportuno precondizionatore del sistema.
Molto interessante sarebbe anche estendere il tipo di fratture trattabili. Attualmente infatti il codice gestisce il caso di più fratture che si intersecano in un punto ma in futuro si potrebbe renderlo adatto a risolvere anche delle biforcazioni.
Un altro ambito di sviluppo potrebbe riguardare l'imposizione di diverse condizioni all'intersezione, più complicate e meno restrittive di quelle imposte finora nel codice, rifacendosi a quanto studiato in [9].

Bibliografia

[1] P. Angot, F. Boyer, F. Hubert, *Asymptotic and numerical modelling of flows in fractured porous media*, ESAIM: Mathematical Modelling and Numerical Analysis, 43(2):239–275, 2009.

[2] P. F. Antonietti, L. Formaggia, A. Scotti, M. Verani, N. Verzotti, *Mimetic finite difference approximation of flows in fractured porous media*, ESAIM: Mathematical Modelling and Numerical Analysis, 2015.

[3] J. Bear, C.-F. Tsang, G. de Marsily, *Flow and contaminant transport in fractured rock*, San Diego, Academic Press, 1993.

[4] L. Beirão da Veiga, K. Lipnikov, G. Manzini, *The Mimetic Finite Difference Method for Elliptic Problems*, Switzerland, Springer, 2014.

[5] K. Brenner, M. Groza, C. Guichard, G. Lebeau, R. Masson, *Gradient discretization of hybrid dimensional Darcy flows in fractured porous media*, In J. Fuhrmann, M. Ohlberger, and C. Rohde, editors, Finite Volumes for Complex Applications VII-Elliptic, Parabolic and Hyperbolic Problems, pages 527–535, Springer, 2014.

[6] F. Brezzi, K. Lipnikov, M. Shashkov, *Convergence of the mimetic finite difference method for diffusion problems on polyhedral meshes*, SIAM Journal on Numerical Analysis, 43(5):1872–1896, 2005.

[7] F. Brezzi, K. Lipnikov, V. Simoncini, *A family of mimetic finite difference methods on polygonal and polyhedral meshes*, Mathematical Models and Methods in Applied Sciences, 15(10):1533–1551, 2005.

[8] C. D'Angelo, A. Scotti, *A mixed finite element method for Darcy flow in fractured porous media with non-matching grids*, ESAIM: Mathematical Modelling and Numerical Analysis, 46(02):465–489, 2012.

[9] L. Formaggia, A. Fumagalli, A. Scotti, P. Ruffo, *A reduced model for Darcy's problem in networks of fractures*, ESAIM: Mathematical Modelling and Numerical Analysis, 48(4):1089-1116, 2014.

[10] L. Formaggia, F. Saleri, A. Veneziani, *Solving Numerical PDEs: Problems, Applications, Exercises*, Italy, Springer, 2012.

[11] A. Fumagalli, *Numerical modelling of flows in fractured porous media by the XFEM method*, PhD Thesis, Politecnico di Milano, 2012.

[12] S. B. Lippman, J. Lajoie, B. E. Moo, *C++ Primer*, Fifth Edition, United States of America, Addison-Wesley, 2012.

[13] V. Martin, J. Jaffrè, J. E. Roberts, *Modeling fractures and barriers as interfaces for flow in porous media*, SIAM Journal on Scientific Computing, 26(5):1667–1691, 2005.

[14] S. Prata, *C++ Primer Plus*, Sixth Edition, United States of America, Addison-Wesley, 2012.

[15] A. Quarteroni, *Modellistica Numerica per Problemi Differenziali*, Quarta edizione, Springer, Milano, 2008.

[16] A. Quarteroni, F. Saleri, R. Sacco, *Matematica Numerica*, Terza edizione, Springer, Italia, 2008.

[17] S. Salsa, *Equazioni a derivate parziali. Metodi, modelli e applicazioni*, Seconda edizione, Springer, Milano, 2011.

[18] N. Verzotti, *Flusso in un mezzo poroso fratturato: approssimazione numerica tramite Differenze Finite Mimetiche*, Tesi Magistrale, Politecnico di Milano, 2014.

[19] CGAL, http://www.cgal.org.

[20] Eigen v3, http://eigen.tuxfamily.org.

[21] GetPot: Input File and Command Line Parsing, http://getpot.sourceforge.net.

Ringraziamenti

Desidero ringraziare tutti coloro che mi hanno sostenuta nella realizzazione della tesi. Grazie alla grande esperienza dei professori che mi hanno guidata in questo percorso e al sostegno delle persone a me care sono riuscita a raggiungere questo fondamentale traguardo.

Anzitutto desidero ringraziare il Professor Luca Formaggia, Relatore della tesi, per i preziosi insegnamenti e per i numerosi consigli. Il suo corso e le ore passate a discutere della mia tesi sono state per me un'occasione per imparare tanto e per ottenere numerosi spunti. Ringrazio il Professore anche per la sua disponibilità a rispondere ai miei quesiti, alle mie curiosità e anche alle mie incertezze. Un grazie sentito va alla Dottoressa Anna Scotti, Correlatrice della tesi, per avermi insegnato un approccio pratico che nei corsi non si impara, dato preziosi suggerimenti e dedicato molto del suo tempo. La ringrazio anche per il supporto e per la sua estrema gentilezza e disponibilità. Un grazie particolare alla Professoressa Paola Antonietti per avere condiviso le sue conoscenze teoriche e per i preziosi suggerimenti dati nei vari incontri. Ringrazio davvero i professori che mi hanno guidata perché mi hanno fatta sentire parte del loro gruppo e mi hanno coinvolta nella loro attività di ricerca. Un grazie va anche ai dottorandi e ai ricercatori con cui ho trascorso giornate di lavoro, in particolare ringrazio Stefano Zonca per i preziosi consigli.

Un ringraziamento immenso va ai miei genitori per il grande sostegno e per l'enorme fiducia che hanno dimostrato nei miei confronti in questi anni. Mi hanno aiutata e incoraggiata nei momenti difficili, mi hanno infuso sicurezza e autostima e soprattutto hanno sempre creduto in me. Mi hanno permesso di condurre il percorso di studi nella maniera più serena possibile ed oltre ad avere condiviso con me le gioie e le grandi soddisfazioni di questi anni, hanno sopportato le mie lamentele e le mie preoccupazioni. Davvero un grazie di cuore.
Proseguo con mia sorella Veronica che ringrazio per avermi regalato un sorriso sia nei momenti sereni che in quelli difficili e soprattutto per essere stata oltre che una sorella, un'amica e una coinquilina fantastica. Lei e Samuele mi sono sempre stati vicini ascoltandomi e dandomi buoni consigli.

Ringrazio anche zii e nonne per avermi sempre incoraggiata e sostenuta.

Un ringraziamento speciale va a Claudio per la pazienza, la disponibilità e il sostegno che mi ha sempre dimostrato. È stato un punto di riferimento per me negli ultimi due anni e mezzo, ha sopportato le mie lamentele e condiviso le mie gioie, sempre pronto ad ascoltarmi, ad incoraggiarmi e a starmi accanto nei momenti piú difficili. Un percorso impegnativo, quello della magistrale in Ingegneria Matematica, ma ringrazio Claudio per essere stato pazientemente accanto a me e per avere sempre creduto in me. Grazie di cuore anche a lui per avere rispettato i miei impegni e le mie esigenze e soprattutto lo ringrazio per tutte le volte che è riuscito a farmi dimenticare i problemi della giornata e a farmi sorridere.

Ringrazio la mia cara amica Giada per avermi ascoltata, supportata ed incoraggiata e per avere condiviso molto con me in questi ultimi anni. La ringrazio per essere stata come una sorella maggiore e soprattutto per avere pienamente avuto fiducia in me. Un grazie anche per avermi reso partecipe di tanti momenti speciali dopo la nascita della piccola Gaia.
Grazie alle mie amiche di sempre, Silvia, Giulia, Mara, Alyssa e Jasmine con cui ho sempre avuto modo di confrontarmi e di confidarmi ma anche di divertirmi e di condividere bei momenti insieme. Le ringrazio per il sostegno che mi hanno dimostrato e per essermi state vicine in diversi momenti del mio percorso universitario. In particolare ringrazio Silvia per le tante chiacchierate, Giulia per le numerose risate, Jasmine per la sua generosità, Alyssa per la sua simpatia e Mara per essere cresciuta con me.
Un grazie anche agli amici che purtroppo ho visto meno in questo periodo ma che so che ci saranno sempre, Elisa e Marco.
Grazie a Luca, Giorgia, Silvia, Stefania e Lorenzo per avere condiviso il percorso di studi con me. Un grazie particolare va a Luca, compagno di banco insostituibile e amico sincero. Il suo supporto è stato fondamentale in molte situazioni, in particolare nel seguire lezioni e preparare esami. Con lui ho condiviso momenti felici e festeggiato ottimi risultati ma anche affrontato situazioni stressanti e pomeriggi di studio intenso. Ringrazio Luca per avere spesso alleggerito giornate pesanti e per avermi spesso trasmesso la sua serenità. Un grazie sentito va alla mia amica Giorgia, una persona disponibile e generosa, che è stata capace di ascoltarmi e incoraggiarmi in molte situazioni e soprattutto di rallegrare numerose giornate. Ringrazio anche Silvia che è stata un'amica eccezionale in questi anni, sempre pronta ad aiutarmi e consigliarmi ma anche a ridere e a divertirsi insieme. Un grazie va a Stefania per le tante giornate che abbiamo passato insieme nell'effettuare importanti progetti con determinazione e confrontando le nostre idee. Ringrazio anche Lorenzo, un amico sincero e disponibile, sempre allegro e pronto a scherzare.
Grazie ai miei amici della squadra di pallavolo che mi hanno regalato momenti sereni e di divertimento. Ringrazio anche le ragazze dell'under14 perché

mi hanno dato tante soddisfazioni.

È stato fondamentale avere accanto tutte queste persone speciali, ciascuna a modo suo, per raggiungere l'obiettivo finale. Grazie.