

POLITECNICO DI TORINO



Corso di Laurea in ingegneria meccanica

Tesi di Laurea Magistrale

**Studio di un algoritmo per la
risoluzione di reti di
teleriscaldamento di grandi
dimensioni**

Relatore

Prof. Vittorio Verda

Correlatore:

Ing. Martina Capone

Laureando

Daniela Faruolo

Luglio 2024

Sommario

Nel presente lavoro di tesi, è stato sviluppato e testato un metodo per la risoluzione efficiente del problema fluidodinamico delle reti di teleriscaldamento, con particolare attenzione alla riduzione dei tempi di calcolo. Le reti di grandi dimensioni, ossia con un elevato numero di nodi e rami, sono caratterizzate dalla risoluzione complessiva della rete inefficace e lenta. L'approccio di cui si discuterà sfrutta un metodo iterativo per la riduzione dei tempi. Per costruire un modello matematico accurato della rete che descrivesse il comportamento del fluido nei condotti sono state utilizzate le equazioni di continuità e della quantità di moto. Per risolverlo, invece, sono stati implementati in MATLAB due algoritmi di risoluzione esistenti: il SIMPLE e il Fixed Point, i quali sono stati applicati, inizialmente, alla sola rete di trasporto per verificarne l'affidabilità e l'accuratezza. Successivamente, i due algoritmi sono stati utilizzati su diverse reti di dimensioni crescenti per analizzare l'andamento dei tempi di calcolo in funzione della dimensione della rete. Questo ha permesso di identificare la relazione tra la dimensione della rete e i tempi di calcolo, conducendo alla ricerca di un metodo mirato per ridurli. Vista la complessità di calcolo si è deciso di dividere la rete generica in due. Durante questo processo, era cruciale trovare un criterio per imporre condizioni al contorno coerenti a questa divisione. Pertanto, è stato implementato un algoritmo per iterare sulle condizioni al contorno fino a garantire coerenza tra i valori di pressione e portata del fluido per entrambe le sottoreti. I risultati delle simulazioni hanno dimostrato che il metodo proposto riduce significativamente i tempi di calcolo, soprattutto per reti con un numero di nodi superiore a circa 2600 nodi, evidenziando l'efficacia del metodo implementato.

Indice

Elenco delle figure	VI
Elenco delle tabelle	VIII
Elenco dei listati	IX
1 Introduzione e scopo del lavoro	1
1.1 Contesto	1
1.2 Obiettivi della ricerca	3
1.3 Domande di ricerca	4
2 Algoritmi di risoluzione	6
2.1 Teorie di base sulla fluidodinamica	7
2.1.1 Equazione di continuità	8
2.1.2 Equazione della quantità di moto	11
2.2 The SIMPLE algorithm	14
2.3 Fixed point algorithm	16
3 Metodologia	20
3.1 Disegno della ricerca	20
3.1.1 Risoluzione rete di trasporto	22
3.1.2 Condizioni al contorno	23
3.1.3 Procedura di analisi di dati	24
3.1.4 Creazione sottorete	26
3.2 Metodologia per il calcolo dei tempi	27
4 Riduzione dei tempi di calcolo	32
4.1 Criterio di suddivisione della rete	33

4.2	Condizioni al contorno	36
4.3	Metodo Iterativo di Convergenza e Implementazione del Codice . .	39
5	Conclusioni	45
5.1	Reti utilizzate	45
5.1.1	Rete 1	46
5.1.2	Rete 2	46
5.1.3	Rete 3	46
5.1.4	Rete 4	47
5.1.5	Rete 5	47
5.2	Confronto dei tempi di calcolo	47
A	Appendice	50
	Riferimenti bibliografici	66

Elenco delle figure

2.1	Elemento infinitesimo di fluido [8]	7
2.2	Conservazione della massa [8]	8
2.3	Sezione rete	9
2.4	Rete generica: bilancio nodo [6]	10
2.5	sforzi elemento infinitesimale fluido [8]	12
2.6	Rete generica: bilancio ramo [6]	12
2.7	Diagramma SIMPLE [6]	17
2.8	Diagramma Fixed Point [6]	18
3.1	Rete di trasporto	21
3.2	Portate Estratte	22
3.3	Portate rete di trasporto	25
3.4	Pressioni rete di trasporto	26
3.5	Sottorete	27
3.6	Ricostruzione Rete	28
3.7	Tempi di calcolo	30
4.1	Nodi di taglio	35
4.2	Sottorete Inferiore	35
4.3	Sottorete Superiore	35
4.4	Portate Sottorete inferiore	37
4.5	Andamento Pressioni	40
4.6	Andamento Portate	41
4.7	Confronto Pressioni determinate con i due metodi	42
4.8	Confronto Portate determinate con i due metodi	43
4.9	Diagramma del Metodo	44

5.1	Confronto tempi di calcolo	48
5.2	Riduzione dei tempi di calcolo con il metodo sviluppato	49

Elenco delle tabelle

5.1	Numero di nodi nelle reti esaminate	45
5.2	Numero di nodi e tempi di calcolo per le diverse reti	47

Elenco dei listati

A.1	Funzione trova_colonne_zero	50
A.2	Codice MATLAB per la risoluzione della rete di trasporto	50
A.3	Codice MATLAB per la risoluzione della rete ampliata	51
A.4	Risoluzione Sottorete 1	53
A.5	Risoluzione Sottorete 2	55
A.6	Codice per la risoluzione della rete divisa in due	56
A.7	Risoluzione Semirete inferiore per reti di grandi dimensioni	57
A.8	Risoluzione Semirete superiore per reti di grandi dimensioni	60
A.9	Risoluzione rete di grandi dimensioni	63

Capitolo 1

Introduzione e scopo del lavoro

1.1 Contesto

Il teleriscaldamento è un sistema avanzato che distribuisce energia termica a distanza da una centrale alle utenze attraverso una rete di tubazioni in cui circola acqua calda o vapore. Questo sistema offre numerosi vantaggi rispetto ai tradizionali sistemi di riscaldamento, in particolare nelle zone in cui la densità degli edifici è elevata. Riduce i costi legati al consumo dei combustibili attualmente in commercio eliminando la necessità di utilizzare un generatore di calore per ogni edificio. [5]

Per secoli, gli edifici sono stati riscaldati utilizzando il calore prodotto per mezzo di diversi combustibili, come ad esempio il carbone. All'inizio del 1900 si è iniziato ad adottare sistemi di riscaldamento più avanzati. Negli Stati Uniti i combustibili vennero sostituiti dal vapore, prima per la produzione di energia elettrica e poi per il riscaldamento degli edifici. Parallelamente in Europa, venivano realizzati sistemi ad acqua calda per la produzione di calore. Solo alla fine del XX secolo in alcuni paesi europei iniziarono a diffondersi i primi sistemi di teleriscaldamento per cercare di abbassare le emissioni di CO₂, in risposta alla crisi energetica. Questo sviluppo interessò in particolare il Regno Unito dove, nei primi anni 2000, vennero sviluppati, in diverse città, sistemi di riscaldamento avanzati che utilizzavano fonti di calore innovative come l'energia geotermica e i impianti di recupero dei rifiuti. [7]

Negli ultimi anni, l'attenzione verso l'efficienza energetica e la sostenibilità ambientale è cresciuta significativamente, spingendo verso l'adozione di tecnologie avanzate per migliorare le prestazioni dei sistemi di teleriscaldamento. Per questo motivo i sistemi di riscaldamento moderni utilizzano fonti di calore rinnovabili. I sistemi di teleriscaldamento moderni mirano a utilizzare come energia termica fonti di calore di scarto provenienti da altri processi. In questo modo è possibile recuperare calore altrimenti disperso nell'ambiente, migliorando l'efficienza complessiva del sistema di teleriscaldamento.[3]

I sistemi di teleriscaldamento sono visti come uno strumento particolarmente efficace per la riduzione delle emissioni di CO₂ nel settore del riscaldamento. Oltre all'utilizzo di energia di scarto come fonte energetica, consentono anche l'integrazione di energia proveniente da fonti rinnovabili. Energia solare, eolica o geotermica, possono essere utilizzate per generare elettricità che può essere utilizzata per il funzionamento dei sistemi di teleriscaldamento [4]

. L'utilizzo della fluidodinamica computazionale si è diffusa in maniera sempre più capillare negli ultimi decenni per studiare il comportamento dei fluidi. Prima della sua diffusione, la risoluzione dei problemi fluidodinamici complessi era di competenza di persone altamente specializzate nel settore. L'accesso alle tecnologie innovative ai più ha permesso l'uso di algoritmi CFD da parte di ingegneri per studi di ricerca, sviluppo e progettazione in ambito industriale. Nell'ambito del teleriscaldamento, disporre di un modello che consenta di simulare accuratamente le diverse condizioni operative è essenziale per monitorare e migliorare il funzionamento della rete termofluidodinamica. La simulazione CFD consente di valutare l'influenza di eventuali variazioni nei parametri quali pressione, temperatura del fluido senza dover necessariamente fare interventi sulla rete. Questo ha consentito di prevedere e prevenire potenziali problemi, migliorando l'efficienza del sistema. Questo ha reso possibile l'identificazione delle condizioni operative ottimali che massimizzano l'efficienza minimizzando le perdite. [2].

I sistemi di teleriscaldamento è necessario che abbiano tempi di risposta brevi alle variazioni delle richieste di energia. Disporre di algoritmi veloci permette di adattarsi rapidamente a questi cambiamenti, accelerando anche i tempi legati allo sviluppo stesso della rete. Le reti di teleriscaldamento possono interessare aree urbane molto estese e in continua crescita, per questo motivo è importante che gli algoritmi siano scalabili e in grado di simulare reti anche di grandi dimensioni senza tempi proibitivi. La presente ricerca ha l'obiettivo di sviluppare un metodo più efficace

per risolvere il problema termofluidodinamico della rete di teleriscaldamento di Torino. Questo metodo mira a ottimizzare la risoluzione del sistema, superando le problematiche legate alle dimensioni e alla complessità della rete. Questo non solo potrebbe incrementare l'efficienza operativa del sistema di teleriscaldamento, ma anche contribuire a una maggiore sostenibilità ambientale e a una riduzione dei costi energetici legati al sistema di riscaldamento. L'importanza di questo studio è legata allo sviluppo di un algoritmo che può essere implementato non solo per la rete di teleriscaldamento di Torino, ma anche per quella di altre città.

1.2 Obiettivi della ricerca

Nello sviluppo del seguente lavoro verrà sviluppato un algoritmo di risoluzione che sarà verificato attraverso simulazioni su reti di diverse dimensioni.

I risultati delle simulazioni saranno confrontati per valutarne l'impatto in termini di efficienza e tempi di calcolo, con l'obiettivo di trovare soluzioni ottimali per la gestione della rete di teleriscaldamento riducendo la complessità di calcolo legata alle dimensioni del sistema. In particolare, la ricerca si propone di raggiungere diversi obiettivi. Un problema fondamentale riscontrato è che le dimensioni della rete di teleriscaldamento di Torino sono estremamente grandi, il che rende difficile la risoluzione del problema fluidodinamico con gli strumenti attualmente disponibili. In particolare, MATLAB incontra notevoli difficoltà in termini di tempi di calcolo nella gestione e risoluzione di algoritmi così complessi a causa del numero di equazioni da risolvere legate alle dimensioni della rete. Questo comporta problemi di ottimizzazione e tempi di calcolo molto elevati, rendendo l'approccio tradizionale inefficace per applicazioni pratiche su larga scala. La fase iniziale di questa ricerca è analizzare l'attuale configurazione della rete, divisa in rete di trasporto e reti di distribuzione, andando a individuare quali sono le dimensioni oltre le quali i tempi di calcolo diventano un problema. La rete di trasporto che è costituita dai tubi primari che collegano le centrali di produzione dell'energia ai nodi principali, mentre le reti di distribuzione, sono collegate alla rete principale e sono costituite da tubazioni secondarie che raggiungono le utenze finali trasportando quindi il calore dalla rete di trasporto a queste. Una volta analizzato il problema, il passo successivo consiste nella creazione e simulazione di algoritmi computazionali che descrivano il comportamento della rete, tenendo conto delle variabili operative e delle condizioni al contorno. L'obiettivo finale è sviluppare una strategia che permetta di risolvere

la rete in modo più veloce e ottimizzato, riducendo i tempi di calcolo e migliorando l'efficienza complessiva del processo. Il funzionamento dei modelli proposti sarà verificato attraverso simulazioni su reti di diverse dimensioni e i risultati delle simulazioni saranno confrontati per valutarne l'impatto in termini di tempi di calcolo. In particolare, si intende determinare se le soluzioni proposte possono essere applicate efficacemente per la gestione della rete di teleriscaldamento di Torino, migliorandone la gestibilità e riducendo i costi computazionali. Il fine ultimo è migliorare l'efficienza dei software attualmente in uso per la gestione complessiva della rete che comprende quelle di trasporto e distribuzione, offrendo un'unica soluzione più veloce per l'intero sistema di teleriscaldamento.

1.3 Domande di ricerca

In che modo le dimensioni della rete di teleriscaldamento influenzano le prestazioni degli algoritmi di risoluzione fluidodinamica utilizzati?

Il problema della creazione di un algoritmo che sia in grado di risolvere il problema termofluidodinamico sull'intera rete di teleriscaldamento è legato alla complessità computazionale del problema, in quanto la complessità e il numero delle equazioni che descrivono il comportamento fluidodinamico dei flussi di calore nella rete richiedono algoritmi altamente efficienti per garantire tempi di calcolo accettabili. Per questo motivo possono essere necessari tempi molto lunghi che renderebbero le simulazioni non eseguibili in tempo reale. L'algoritmo dovrà essere studiato in modo tale da adattarsi a reti di dimensioni e complessità diverse, questo è fondamentale affinché sia garantita l'applicabilità del codice a scenari realistici. Una mancata scalabilità del codice può limitarne l'utilità stessa. Per realizzare un codice che produca risultati affidabili e accurati è necessario verificarlo e ciò è possibile attraverso la comparazione dei risultati ottenuti dall'esecuzione del codice con dati sperimentali o soluzioni analitiche note, in modo da confermare l'attendibilità delle simulazioni. Nel caso di mancata validazione nei risultati prodotti dall'algoritmo, l'applicabilità pratica potrebbe essere limitata. Una mancata validazione può compromettere la fiducia nei risultati prodotti dal codice e limitarne l'applicabilità pratica. Per questo motivo, in tutte le fasi della ricerca, gli algoritmi utilizzati per simulare la rete e quelli sviluppati per la riduzione dei tempi di calcolo, prima di essere utilizzati su reti grandi, sono stati prima verificati sulla rete di trasporto per la quale si ha un numero di dati a disposizione più elevato. Un altro aspetto fondamentale è la

definizione delle condizioni al contorno, infatti piccole variazioni di queste possono comportare significativi cambiamenti sulle portate e le pressioni. Per questo motivo errori o approssimazioni durante la definizione di quest'ultime potrebbero portare a risultati non affidabili. Per questo motivo sono state considerati criteri di convergenza molto stringenti, per evitare variazioni troppo grandi rispetto alle reali condizioni di funzionamento. Quali sono le strategie di ottimizzazione che permetterebbero di ridurre i tempi di calcolo per le reti di grandi dimensioni?

Si è pensato di considerare la rete complessiva come suddivisa in sottoreti, al fine di ottimizzare l'esecuzione dell'algoritmo e accelerare il processo di risoluzione garantendo maggiore efficienza. Per valutare l'efficacia delle soluzioni ottenute si può tenere conto di diversi parametri tra cui:

- Velocità di esecuzione: una riduzione dei tempi di calcolo indica un miglioramento delle prestazioni;
- Valutazione dei tempi di calcolo in funzione del metodo utilizzato e delle dimensioni della rete considerata;
- Precisione e quindi verifica dell'accuratezza dei risultati, in quanto questa non dipende dalla riduzione dei tempi

Dal punto di vista delle difficoltà tecniche e computazionali, oltre alla dimensione della rete e quindi dei tempi di calcolo, ci sono l'incompletezza dei dati che rende difficile ottenere risultati affidabili e precisi oltre che la loro comparazione con parametri sperimentali. Inoltre potrebbe essere un problema l'integrazione del nuovo algoritmo con sistemi attualmente esistenti. Quali sono i benefici che possono essere ottenuti dall'adozione di un metodo di risoluzione ottimizzato e unico per l'intera rete di teleriscaldamento di Torino?

Un sistema unificato consente di gestire in modo più efficiente e rapido gli interventi di manutenzioni e le operazioni quotidiane sulla rete. Ciò significa ridurre i tempi di intervento in caso di guasti o problemi, migliorando i tempi di risposta complessiva del sistema. Ottimizzare la gestione della rete permette di ridurre le perdite e di ottimizzare l'utilizzo delle risorse energetiche andando a ridurre i costi e l'impatto ambientale.

Capitolo 2

Algoritmi di risoluzione

In questo capitolo, verranno esaminati in dettaglio gli algoritmi utilizzati per risolvere il problema fluidodinamico relativo alle reti di teleriscaldamento. Prima di procedere con lo sviluppo dell'algoritmo per velocizzare la risoluzione delle reti di grandi dimensioni, è essenziale introdurre gli algoritmi necessari per risolvere i problemi termofluidodinamici all'interno di tali reti. Per arrivare alla definizione degli algoritmi che delineano la metodologia per determinare le portate e le pressioni che garantiscono il corretto funzionamento delle reti, è necessario fare una panoramica della teoria che sta alla base del moto dei fluidi nei condotti. Nella prima parte di questo capitolo, verranno discusse le teorie fondamentali della fluidodinamica, concentrandosi in particolare sull'equazione di continuità e sull'equazione del momento. Queste equazioni sono essenziali per lo studio che seguirà e forniranno le basi teoriche necessarie per comprendere e implementare gli algoritmi di risoluzione. Questo processo richiede un passaggio dalla teoria fluidodinamica alla formulazione e implementazione degli algoritmi stessi. Successivamente, verranno illustrati i due algoritmi di risoluzione fondamentali: il metodo SIMPLE e il metodo del Punto Fisso. Il metodo SIMPLE è particolarmente importante perché permette di risolvere le equazioni di Navier-Stokes in modo efficiente. Questo metodo semi-implicito collega le equazioni della pressione e della velocità, aggiornando iterativamente i campi di velocità e pressione fino a raggiungere la convergenza. Nella pratica, il metodo SIMPLE inizia impostando un valore iniziale arbitrario per la variabile di pressione. Successivamente, attraverso iterazioni successive, calcola una correzione per le variabili di velocità e pressione, fino a quando la differenza tra le soluzioni successive è sufficientemente piccola, indicando che la soluzione ha convergenza.

Tuttavia, poiché le equazioni coinvolte sono non lineari, è necessario un approccio mirato per calcolare accuratamente queste correzioni. Per questo motivo, in combinazione con il SIMPLE, viene impiegato l'algoritmo del FIXED Point. Il metodo del FIXED POINT, è un approccio iterativo per risolvere equazioni non lineari, si basa sull'applicazione iterativa di una funzione di aggiornamento che converge verso la soluzione desiderata, sarà meglio spiegato in seguito, nel caso specifico, viene utilizzato per calcolare le correzioni che, ad ogni iterazione, devono essere applicate alle variabili di pressione e portata affinché si raggiunga la convergenza e quindi si arrivi alla soluzione del sistema. Gli algoritmi definiti verranno successivamente applicati alla rete di trasporto del teleriscaldamento della città di Torino. Questo servirà a verificare l'affidabilità e l'accuratezza delle simulazioni, consentendo di confrontare i risultati ottenuti con i dati reali e di valutare l'efficacia dei metodi proposti nel contesto di una rete complessa e di grande scala.

2.1 Teorie di base sulla fluidodinamica

Le equazioni che descrivono i principi fondamentali della fluidodinamica sono le equazioni di Navier-Stokes. In particolare, in questo studio sono state utilizzate le equazioni di continuità e quella di conservazione della quantità di moto. Nel seguito, per la scrittura della forma generale delle equazioni della fluidodinamica, si farà riferimento all'elemento infinitesimo di fluido in figura 2.1.

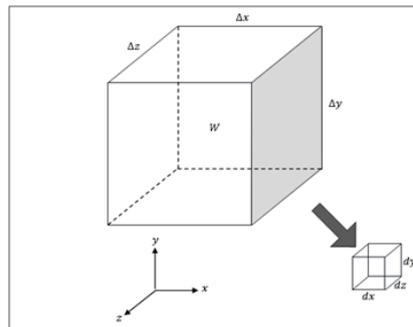


Figura 2.1: Elemento infinitesimo di fluido [8]

2.1.1 Equazione di continuità

L'equazione di continuità afferma che, considerato un qualsiasi volume di controllo di un fluido, la massa entrante e uscente dal volume di controllo in un intervallo di tempo Δt è uguale alla variazione interna al volume nello stesso intervallo di tempo. In generale si può scrivere:

$$G_{us} - G_{in} = \frac{\partial m}{\partial t} \quad (2.1)$$

Per un volume di controllo infinitesimo, come quello rappresentato in figura 2.2,

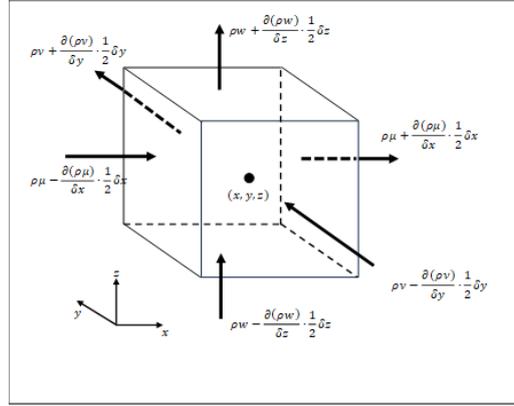


Figura 2.2: Conservazione della massa [8]

vale:

$$\int_A \rho v \cdot n dA + \int_W \frac{\partial \rho}{\partial t} dx dy dz = 0 \quad (2.2)$$

Nel caso in esame delle reti di teleriscaldamento, tuttavia, si fa riferimento alla forma discreta dell'equazione di continuità tenendo conto, inoltre, che la massa non varia nel tempo e quindi il caso considerato è stazionario. L'equazione diventa quindi:

$$\sum_{in} (\rho_{in} v_{1,in} S_{in} + G_{inj,i}) - \sum_{us} (\rho_{us} v_{1,us} S_{us} + G_{ex,i}) = 0 \quad (2.3)$$

Dove le G con il pedice inj ed ex rappresentano rispettivamente le portate provenienti dalle centrali termiche e quelle estratte nei nodi i che corrispondono a quelle che raggiungono le utenze come in figura 2.3.

La rete di teleriscaldamento può essere rappresentata come un grafo, dove i nodi rappresentano i punti in cui confluiscono più condotti e i collegamenti rappresentano i condotti che trasportano il fluido alle reti di distribuzione o alle utenze. Nella teoria dei grafi orientati, ogni arco ha una direzione associata. Nel contesto della

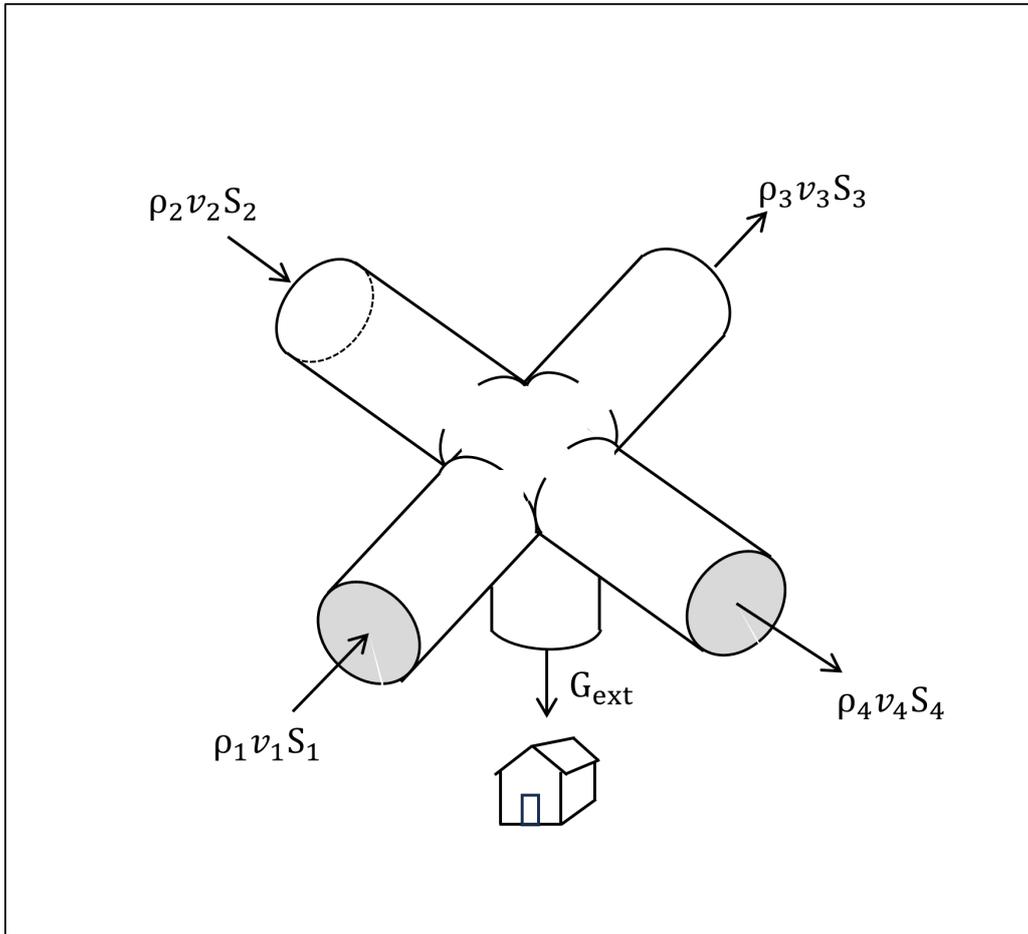


Figura 2.3: Sezione rete

rete di teleriscaldamento, ogni arco può essere orientato in base al verso della portata di fluido. Gli archi entranti, dove il fluido entra in un nodo, possono essere rappresentati con una freccia e un peso associato di -1 . Gli archi uscenti, dove il fluido esce da un nodo, possono essere rappresentati con una freccia e un peso di $+1$.

Utilizzando la matrice di incidenza, che rappresenta dal punto di vista matematico le connessioni tra i nodi e gli archi di un grafo orientato, è possibile incorporare i valori -1 e $+1$ per rappresentare le portate in entrata e in uscita da ciascun nodo. In particolare, le colonne corrispondono agli archi e le righe ai nodi. [1]

Questa teoria può essere utilizzata per analizzare la logica della rete di teleriscaldamento. Ad esempio, è possibile determinare se esistono percorsi alternativi per il trasporto del fluido in caso di guasti o interruzioni, migliorando il funzionamento.

Si consideri ora una piccolissima parte di una generica rete rappresentata in figura 2.4, costituita da sei nodi e cinque rami, in cui sono riportati anche i versi delle portate in ciascun ramo e prendiamo come volume di controllo quello in rosso.

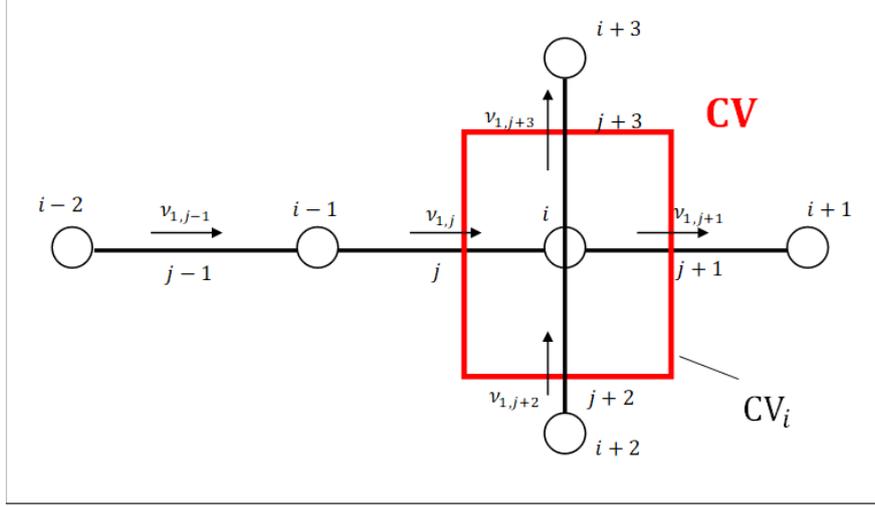


Figura 2.4: Rete generica: bilancio nodo [6]

Il bilancio di massa attraverso il V.C. è dato da:

$$\rho_j v_{1,j} S_j + \rho_{j+2} v_{1,j+2} S_{j+2} + G_{inj,i} - \rho_{j+3} v_{1,j+3} S_{j+3} - \rho_{j+1} v_{1,j+1} S_{j+1} - G_{ext,i} = 0 \quad (2.4)$$

Per semplificare l'equazione, si considerano i flussi di massa al posto delle velocità nei rami e, relativamente ai nodi, si valuta il bilancio netto tra le immissioni e le estrazioni.

$$G_{extr,i} = G_{ext,i} - G_{inj,i} \quad (2.5)$$

Per cui si può riscrivere l'equazione di continuità come:

$$\sum_j G_j + G_{ext,i} = 0 \quad (2.6)$$

Per i sistemi discreti, come in questo caso, si può scrivere l'equazione di continuità in forma matriciale:

$$AG + G_{ext} = 0 \quad (2.7)$$

Dove A è la matrice di incidenza, G il vettore dei flussi di massa nei rami e G_{ext} il vettore delle portate estratte.

Con riferimento alla figura 2.4 si può scrivere:

$$\begin{bmatrix} +1 & 0 & 0 & 0 & 0 \\ -1 & +1 & 0 & 0 & 0 \\ 0 & -1 & +1 & -1 & +1 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} G_{J-1} \\ G_J \\ G_{J+1} \\ G_{J+2} \\ G_{J+3} \end{bmatrix} + \begin{bmatrix} G_{\text{ext},i-2} \\ G_{\text{ext},i-1} \\ G_{\text{ext},i} \\ G_{\text{ext},i+1} \\ G_{\text{ext},i+2} \\ G_{\text{ext},i+3} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.8)$$

Le portate uscenti dal volume di controllo sono state considerate positive, mentre quelle entranti sono state considerate negative [6].

2.1.2 Equazione della quantità di moto

L'equazione della quantità di moto, o del momento, afferma che la variazione del momento di una particella di fluido è uguale alla somma delle forze che agiscono su di essa.

$$\begin{bmatrix} \text{variazione} \\ \text{del momento} \end{bmatrix} = \begin{bmatrix} \text{variazione della quantità di moto} \\ \text{dovuta alla convezione} \end{bmatrix} + \begin{bmatrix} \text{forze} \\ \text{esterne} \end{bmatrix} \quad (2.9)$$

Dove:

$$\begin{bmatrix} \text{variazione} \\ \text{del momento} \end{bmatrix} = \frac{d(\rho v)}{dt} \quad (2.10)$$

$$\begin{bmatrix} \text{variazione della quantità di moto} \\ \text{dovuta alla convezione} \end{bmatrix} = -(\nabla \cdot \rho \mathbf{v}\mathbf{v})dV \quad (2.11)$$

Il termine legato alle forze esterne può essere diviso in forze di superficie e di volume per cui:

$$\begin{bmatrix} \text{forze} \\ \text{di superficie} \end{bmatrix} = -(\nabla \cdot \boldsymbol{\tau}) \cdot dV - (\nabla p) \cdot dV \quad (2.12)$$

la cui formulazione può essere ricavata dalla figura 2.5.

$$\begin{bmatrix} \text{forze} \\ \text{di volume} \end{bmatrix} = \rho \mathbf{g} \quad (2.13)$$

Possiamo quindi scrivere l'equazione della quantità di moto, per un fluido Newtoniano:

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \mu \nabla^2 \mathbf{v} + \rho \mathbf{g} \quad (2.14)$$

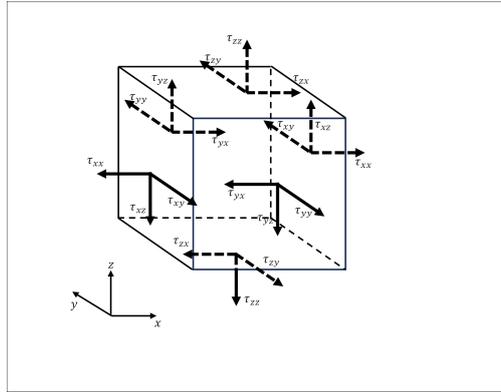


Figura 2.5: sforzi elemento infinitesimale fluido [8]

[2]. Nel caso particolare della rete di teleriscaldamento, i contributi legati agli sforzi tensoriali del fluido possono essere sostituiti dalla forma delle perdite di carico del fluido, concentrate e distribuite, all'interno di un condotto. Anche in questo caso si fa riferimento alla forma discreta dell'equazione del caso stazionario, scritta per la rete riportata sopra, ma in questo caso il volume di controllo a cui si fa riferimento è quello riportato in verde

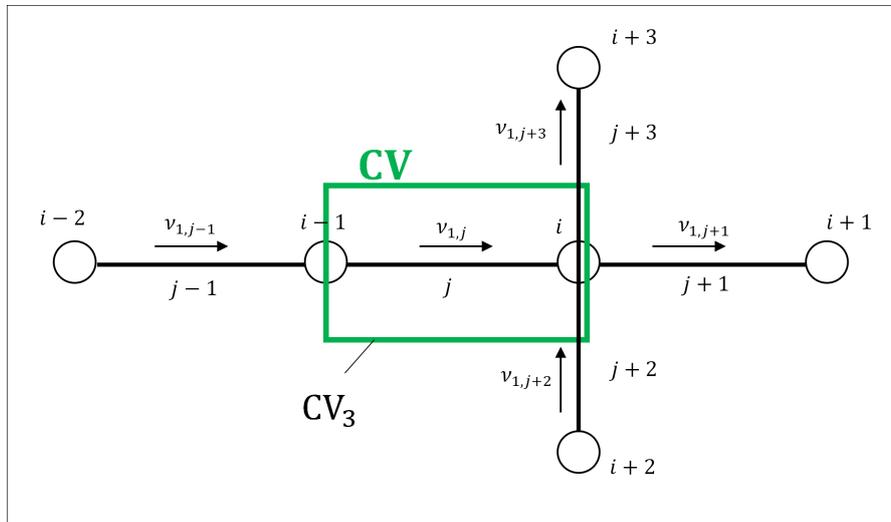


Figura 2.6: Rete generica: bilancio ramo [6]

$$P_i + P_{i-1} + \frac{1}{2} \rho_j v_{i,j}^2 \left(f_j \frac{L_j}{D_{h,j}} + \sum_k \beta_{k,j} \right) - \Delta P_{pump,j} = 0 \quad (2.15)$$

in questa abbiamo che

P_i = pressioni relative ai nodi

$v_{1,j}$ = velocità del fluido nei rami j

f_j = fattore di attrito

L_j = lunghezza dei rami

D_j = diametro condotti

β = coefficienti di perdita di carico

ΔP_{pump} = dovuta alla presenza di eventuali pompaggi nei rami considerati.

In particolare tra parentesi i contributi moltiplicati per f rappresentano le perdite di carico distribuite, mentre i contributi moltiplicati per β rappresentano le perdite di carico concentrate. I valori di beta utilizzati nella trattazione sono:

- giunzioni a T, $\beta=2$
- curve di 90° , $\beta=0.9$
- curve di 45° , $\beta=0.3$

Il fattore di attrito f , è stato considerato pari a 0.017.

Anche in questo caso è possibile semplificare l'equazione considerando che:

$$v_{1,j} = \frac{G_j}{S_j \rho_j} \quad (2.16)$$

$$R_j = \frac{1}{2} \frac{\left(f_j \frac{L_j}{D_{h,j}} + \sum_k \beta_{k,j} \right)}{\rho_j S_j^2} G_j \quad (2.17)$$

L'equazione diventa:

$$(P_{i-1} - P_i) = R_j G_j - \Delta P_{\text{pump},j} \quad (2.18)$$

In forma matriciale diventa:

$$A^T P = R G - t \quad (2.19)$$

$$R \triangleq \begin{bmatrix} R_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R_{NB} \end{bmatrix} \quad (2.20)$$

Volendo esplicitare il vettore delle portate si ha:

$$G = Y A^T P + Y t \quad (2.21)$$

$$Y \triangleq \begin{bmatrix} \frac{1}{R_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{R_{NB}} \end{bmatrix} \quad (2.22)$$

Per il caso considerato:

$$\begin{aligned}
 \begin{bmatrix} G_{J-1} \\ G_J \\ G_{J+1} \\ G_{J+2} \\ G_{J+3} \end{bmatrix} &= \begin{bmatrix} Y_{j-1} & 0 & 0 & 0 & 0 \\ 0 & Y_j & 0 & 0 & 0 \\ 0 & 0 & Y_{j+1} & 0 & 0 \\ 0 & 0 & 0 & Y_{j+2} & 0 \\ 0 & 0 & 0 & 0 & Y_{j+3} \end{bmatrix} \\
 &\times \begin{bmatrix} +1 & -1 & 0 & 0 & 0 \\ 0 & +1 & -1 & 0 & 0 \\ 0 & 0 & +1 & -1 & 0 \\ 0 & 0 & 0 & +1 & -1 \\ 0 & 0 & 0 & 0 & +1 \end{bmatrix} \times \begin{bmatrix} P_{i-2} \\ P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{bmatrix} \\
 &+ \begin{bmatrix} Y_{j-1} & 0 & 0 & 0 & 0 \\ 0 & Y_j & 0 & 0 & 0 \\ 0 & 0 & Y_{j+1} & 0 & 0 \\ 0 & 0 & 0 & Y_{j+2} & 0 \\ 0 & 0 & 0 & 0 & Y_{j+3} \end{bmatrix} \times \begin{bmatrix} t_{J-1} \\ t_J \\ t_{J+1} \\ t_{J+2} \\ t_{J+3} \end{bmatrix} \quad (2.23)
 \end{aligned}$$

[6]

2.2 The SIMPLE algorithm

Le equazioni di Navier-Stokes sono non lineari, per questo la loro risoluzione analitica è molto complicata. Nel caso specifico, le reti che si sono considerate possono avere migliaia di punti di griglia, ciascuno con un insieme di equazioni associate. Risolvere analiticamente questo problema richiederebbe un numero inimmaginabile di calcoli manuali, rendendo l'approccio analitico impraticabile. Il metodo SIMPLE è particolarmente efficace per risolvere numericamente le equazioni di Navier-Stokes per fluidi incomprimibili.

L'algoritmo SIMPLE (chiamato così in quanto significa Semi-Implicit Method for Pressure-Linked equations) utilizza la seguente logica.

Si comincia ipotizzando un vettore randomico per il campo delle pressioni P^* , in modo da poter inizializzare l'algoritmo e determinare successivamente il campo delle portate associate risolvendo l'equazione del momento 2.21.

Gli elementi definiti per la risoluzione dell'algoritmo sono:

- AA è la matrice di incidenza della rete.
- Y è una matrice sparsa utilizzata per rappresentare le conduttanze del sistema.
- Gstar è un vettore inizializzato a 1, rappresentante le portate ipotizzate.
- G è un vettore inizializzato a 1, rappresentante le portate esatte.
- P è il campo delle pressioni iniziali, generato come un vettore di numeri casuali moltiplicati per 10^5 per ottenere valori nell'ordine di grandezza desiderato.

A questo punto sottraendo l'equazione del momento “vera”, cioè quella per la valutazione dei valori esatti di pressioni e portate, da quella con i valori ipotizzati si ottiene:

$$\begin{aligned} \mathbf{G} &= \mathbf{Y} \cdot \mathbf{A}^T \cdot \mathbf{P} + \mathbf{Y} \cdot \mathbf{t} = \\ \mathbf{G}^* &= \mathbf{Y}^* \cdot \mathbf{A}^T \cdot \mathbf{P}^* + \mathbf{Y}^* \cdot \mathbf{t} \end{aligned} \quad (2.24)$$

$$\mathbf{G} - \mathbf{G}^* = \mathbf{Y} \cdot \mathbf{A}^T \cdot \mathbf{P} - \mathbf{Y}^* \cdot \mathbf{A}^T \cdot \mathbf{P}^* + \mathbf{Y} \cdot \mathbf{t} - \mathbf{Y}^* \cdot \mathbf{t}$$

In generale il termine Y dipende da G ma si assume che la non linearità di Y sia molto debole in modo tale da poter scrivere $Y(G) = Y(G^*)$. Quest'assunzione porta all'equazione che mette in relazione la correzione sulla portata con quella sulla pressione.

$$\mathbf{G}' = \mathbf{Y}^* \cdot \mathbf{A}^T \cdot \mathbf{P}' \quad (2.25)$$

L'equazione di continuità scritta con riferimento alla correzione G' diventa:

$$\mathbf{A} \cdot \mathbf{G}' - \mathbf{A} \cdot \mathbf{G}^* - \mathbf{G}_{ext} \quad (2.26)$$

E sostituendo nell'equazione 2.27 si ottiene l'equazione in cui si esplicita la correzione P' .

$$\mathbf{A} \cdot \mathbf{Y}^* \cdot \mathbf{A}^T \cdot \mathbf{P}' = -\mathbf{A} \cdot \mathbf{G}^* - \mathbf{G}_{ext} \quad (2.27)$$

Con $\mathbf{A}\mathbf{Y}^*\mathbf{A}^T\mathbf{P}' = \mathbf{H}$ e $\mathbf{A}\mathbf{G}^* - \mathbf{G}_{ext} = \mathbf{b}$

$$\mathbf{H}\mathbf{P}' = \mathbf{b} \quad (2.28)$$

I fattori DatiSolver.SottSIMPLEp e DatiSolver.SottSIMPLEg vengono utilizzati per la stabilità del processo iterativo, per evitare che ad ogni iterazione le variabili siano troppo diverse da quelle dell'iterazione precedente. La condizione di convergenza per l'algoritmo SIMPLE è che l'errore medio tra la differenza dei vettori delle

pressioni, soluzioni dell'equazione della quantità di moto tra l'iterazione corrente e quella precedente, nonché l'errore medio dalla risoluzione dell'equazione di continuità, siano entrambi inferiori a un valore piccolo scelto arbitrariamente. Gli step di questo algoritmo sono quindi:

1. Si ipotizzano dei valori di P^* e G^*
2. Si risolvono le equazioni con i valori ipotizzati
3. Si calcolano le correzioni P' e G'
4. Si trovano i nuovi valori di G^* e P^*
5. Se i valori di G e P trovati soddisfano la condizione di convergenza l'algoritmo si interrompe, altrimenti si ritorna al punto 2.

Possono essere schematizzati come in figura 2.7.

2.3 Fixed point algorithm

Dal paragrafo precedente si può osservare che le equazioni con i valori ipotizzati della pressione e della portata per la risoluzione del SIMPLE, sono non lineari. Per questo motivo, per poter calcolare i valori delle correzioni sulle due variabili, si utilizza il metodo del punto fisso per la risoluzione delle stesse. Questo metodo funziona nel seguente modo: si considera l'espressione delle portate ricavata dall'equazione del momento 2.21.

$$G = Y A^T P + Y t$$

Il punto fisso cerca il vettore delle portate G tale per cui $G = \phi(G)$ dove ϕ è una funzione derivata da 2.21.

Si crea un algoritmo che iterativamente aggiorna la variabile della portata G in questo modo:

$$G_{k+1} = \phi(G_k) \tag{2.29}$$

fino alla convergenza. L'algoritmo va inizializzato scegliendo arbitrariamente un vettore delle portate.

Con:

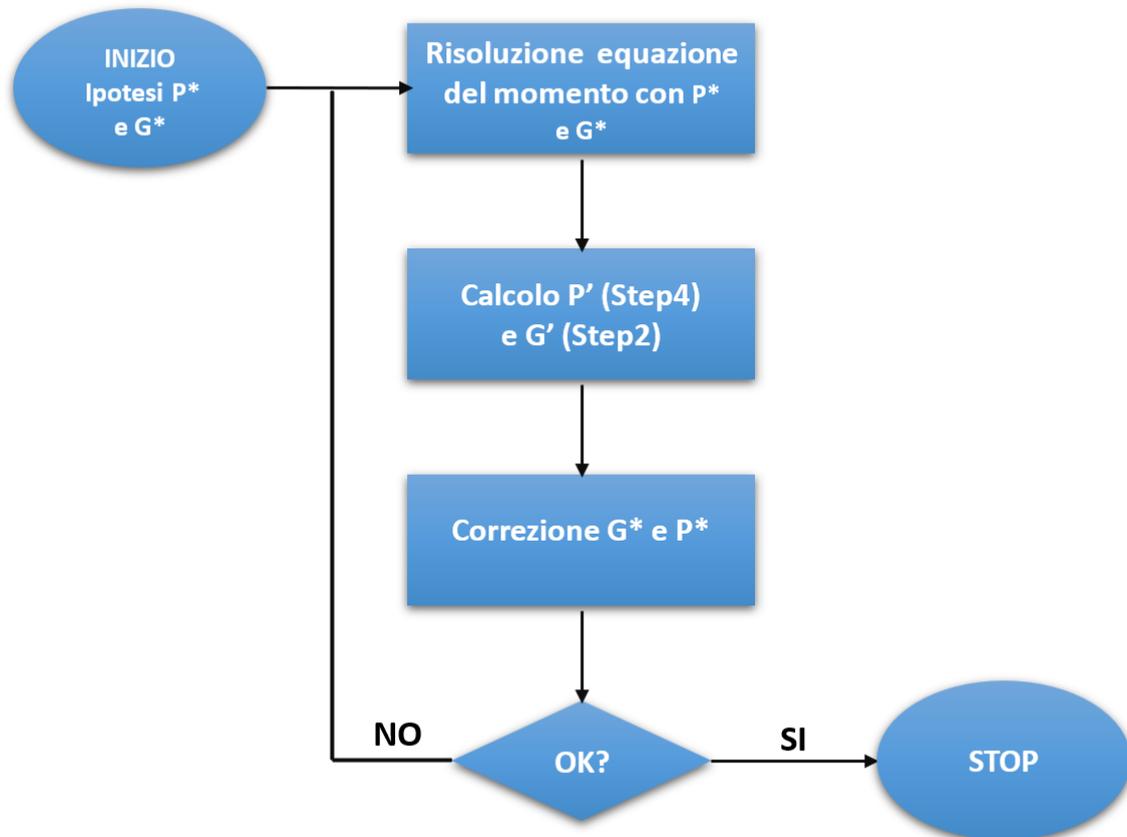


Figura 2.7: Diagramma SIMPLE [6]

- NetData.ff= fattore di attrito
- NetData.LL=lunghenze dei rami
- NetData.DD=diametro dei tubi
- NetData.beta=fattori di perdita di carico
- NetData.Sezione=sezioni dei tubi

Per evitare problemi di convergenza, inoltre, si utilizzano due coefficienti di sotto-relazione λ_1 e λ_2 , che garantiscono la stabilità delle iterazioni, in particolare sono tali che:

$$\lambda_1 + \lambda_2 = 1 \quad (2.30)$$

La loro funzione è quella di ridurre la variazione tra la soluzione trovata e quella dell'iterazione precedente, rendendo il processo iterativo più graduale come mostrato di seguito:

L'algoritmo si interrompe quando converge e cioè quando:

$$\max(|G_{k+1} - g(G_k)|) < \epsilon \quad (2.31)$$

Dove epsilon è un numero arbitrariamente piccolo, in questo caso è stato scelto pari a $\text{DatiSolver.TolFP}=10^{-4}$.

In figura 5.1 è spiegato come funziona il Fixed Point.

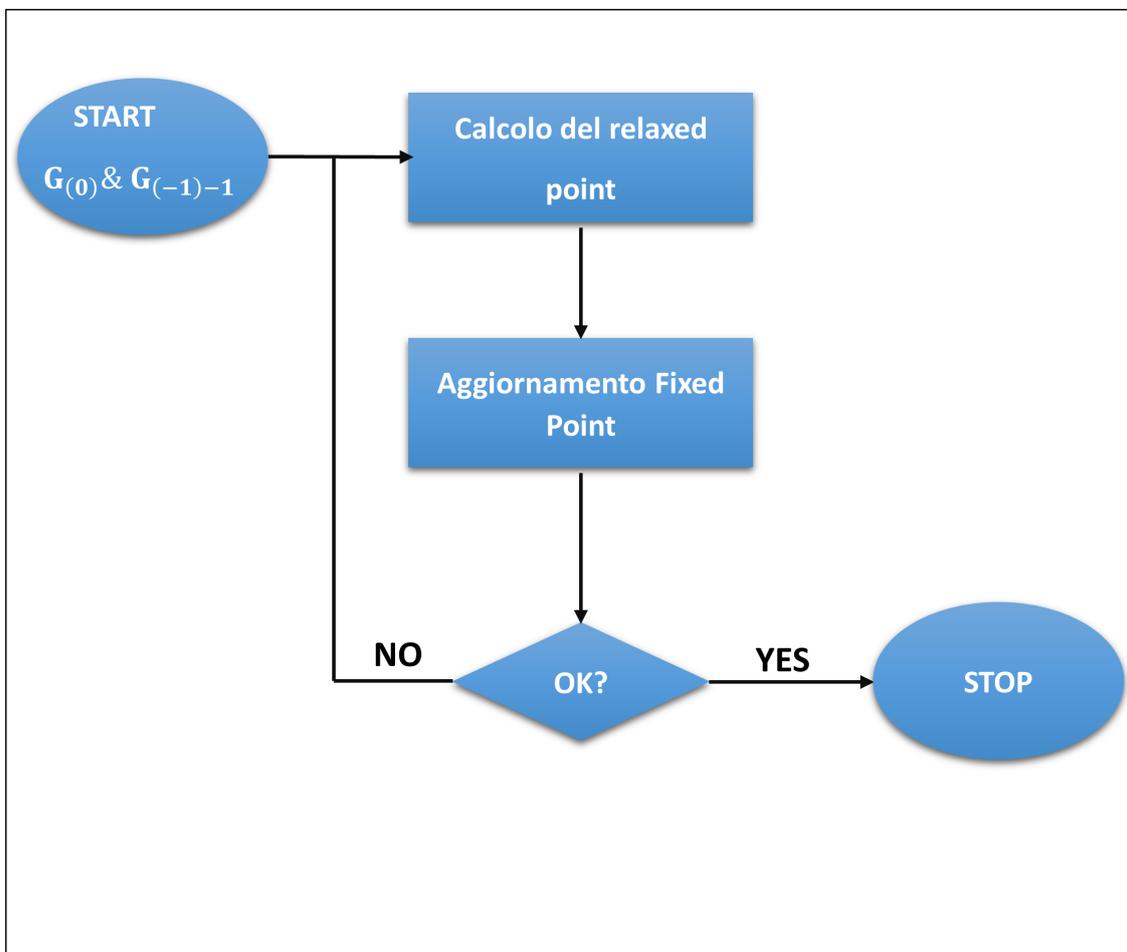


Figura 2.8: Diagramma Fixed Point [6]

Capitolo 3

Metodologia

Nel capitolo che segue, viene presentata la metodologia seguita per lo sviluppo di un algoritmo di simulazione per la risoluzione della rete di teleriscaldamento. Il problema è stato affrontato in modo sistematico e rigoroso cercando di rispondere alle domande di ricerca presentate precedentemente puntando all'ottenimento di risultati validi ed affidabili.

Il capitolo si articola in diverse sezioni che coprono l'intero processo di ricerca, dalla pianificazione iniziale fino all'implementazione e validazione degli algoritmi di risoluzione.

La sezione sull'algoritmo di risoluzione del problema fluidodinamico rappresenta un punto focale della metodologia, in quanto illustra in dettaglio il processo di sviluppo e di validazione dell'algoritmo, con particolare attenzione alle difficoltà legate alla risoluzione di reti di grandi dimensioni.

3.1 Disegno della ricerca

L'obiettivo principale di questo capitolo è quello di sviluppare un algoritmo efficace per risolvere il problema termofluidodinamico della rete cercando di risolvere i problemi legati alle grandi dimensioni della stessa che comportano notevoli difficoltà di ottimizzazione e tempi di calcolo molto lunghi.

Nella fase iniziale della ricerca sono stati valutati i tempi di calcolo relativi alla risoluzione della sola rete di trasporto, e successivamente per valutare come variassero i tempi di calcolo in funzione del numero di rami e di nodi è stata utilizzata

una rete di supporto realizzata in modo tale da sostituire i nodi di estrazione della rete di trasporto. Man mano che le dimensioni della rete aumentavano, è stato possibile osservare e valutare l'impatto sui tempi di calcolo, permettendo un'analisi dettagliata per lo sviluppo di un algoritmo di risoluzione efficace.

In figura 3.1 è rappresentata la rete di trasporto a cui si fa riferimento. Questa

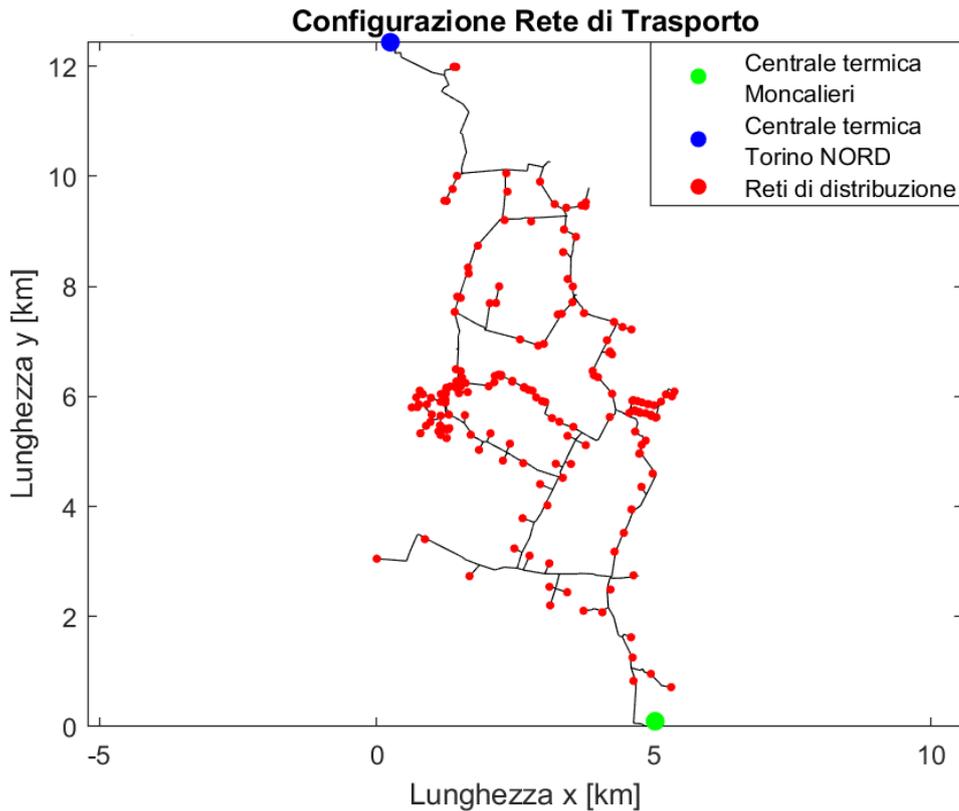


Figura 3.1: Rete di trasporto

rete è costituita da 703 rami e 695 nodi, di cui 182 sono nodi collegati alle reti di distribuzione, inizialmente ipotizzati come nodi di estrazione, e rappresentate in rosso in figura.

Il calore viene trasportato alle reti di distribuzione e successivamente queste lo trasportano fino alle utenze. In blu è rappresentata la centrale termica di Torino Nord e in verde, quella di Moncalieri.

Sebbene ci siano più centrali termiche, queste sono le uniche due che verranno considerate.

Dalle condizioni di funzionamento ipotizzate, la pressione imposta dalla centrale

termica di Moncalieri serve per far funzionare la rete con una pressione di 2.5 bar, mentre si è ipotizzato che la centrale di Torino Nord fornisca il 33% della portata richiesta dalle reti di distribuzione.

In figura 3.2 sono rappresentate le portate richieste dalle 182 reti di distribuzione che dal punto di vista operativo, si è considerato che le portate fossero pari al 42% di quelle nominali.

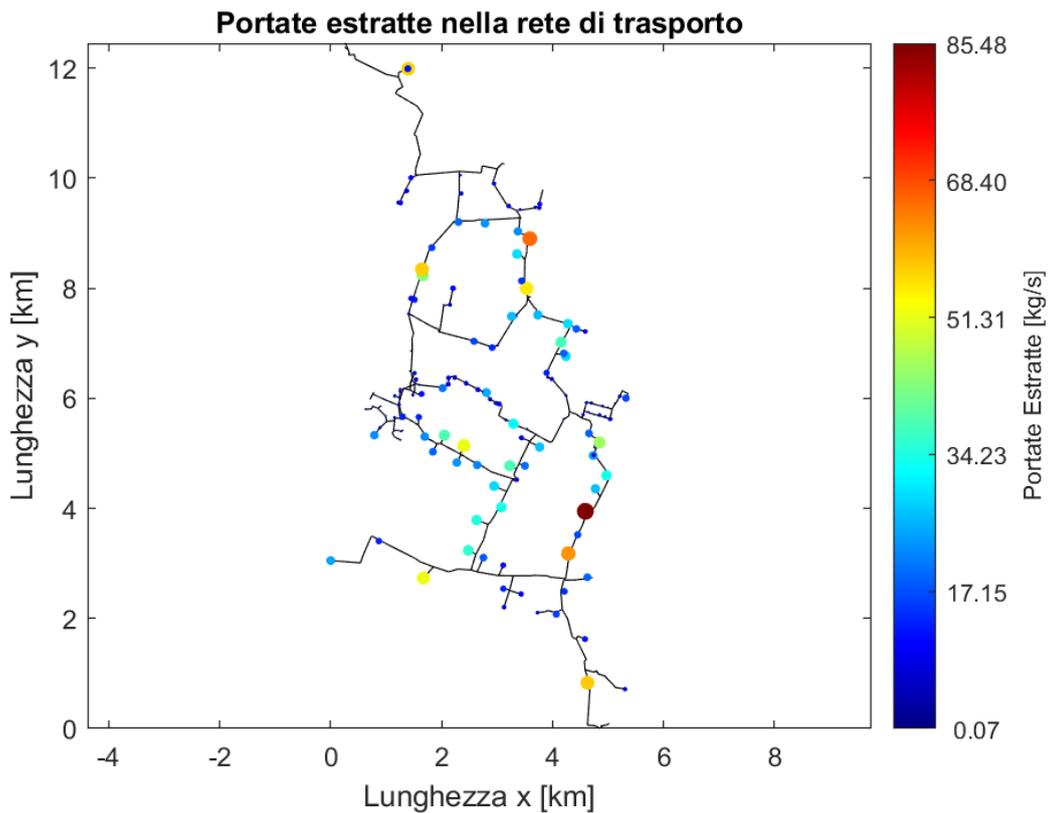


Figura 3.2: Portate Estratte

3.1.1 Risoluzione rete di trasporto

Dal capitolo 2 si sono ricavate le equazioni di continuità e del momento in forma discreta.

Le equazioni del bilancio di massa in gioco sono in numero pari al numero dei rami della rete, e quelle del bilancio del momento sono pari al numero dei nodi della rete. Per il caso in esame della rete di trasporto, il numero di equazioni disponibili è di $703+695=1398$. D'altra parte, il numero di incognite è dato dal numero dei nodi

in cui sono sconosciute le pressioni e dal numero di rami in cui non sono note le portate, risultando in totale 1398 incognite.

Per poter risolvere le equazioni, bisogna considerare la comprimibilità del fluido. Se non fosse così, la pressione risulterebbe slegata dal bilancio di massa, creando un sistema di equazioni in cui la pressione appare solo nell'equazione del momento e non è direttamente correlata al bilancio di massa. Questo disaccoppiamento comporta che la soluzione per la pressione potrebbe non soddisfare le condizioni imposte dalla continuità.

Con la condizione di incomprimibilità, invece, i campi di velocità e pressione sono accoppiati, garantendo che la pressione che soddisfa l'equazione del momento corrisponda alla velocità che soddisfa l'equazione di continuità. Questo accoppiamento permette di determinare univocamente il campo di pressione una volta noto il campo di velocità e viceversa.

3.1.2 Condizioni al contorno

L'equazione del momento considera il gradiente di pressione, ossia una differenza di pressione. Per risolvere il sistema, è quindi necessario imporre almeno una pressione su un nodo.

Per fare ciò, bisogna inserire nel vettore delle pressioni il questo valore nella posizione corrispondente al nodo con pressione nota P_i .

$$P_{(k)}^* = \begin{bmatrix} P_1^* \\ P_2^* \\ \vdots \\ P_i \\ \vdots \\ P_n^* \end{bmatrix} \quad (3.1)$$

Nel codice MATLAB per l'algoritmo del SIMPLE l'aggiornamento del vettore delle pressioni avviene come di seguito:

```
1 P(NodePBC) = Pimposta;
```

Dove NodePBC è il nodo relativo alla centrale termica di Moncalieri, a cui si fa riferimento con NetData.PowerPlants.Nodes(1) nei listati in appendice.

Analogamente, la matrice H deve essere aggiornata impostando tutti i valori della

riga corrispondente al nodo con pressione nota a zero, eccetto quello sulla diagonale principale.

$$\mathbf{H} = \begin{bmatrix} H_{11} & H_{12} & \cdots & \cdots & \cdots & H_{1n} \\ H_{21} & H_{22} & \cdots & \cdots & \cdots & H_{2n} \\ 0 & 0 & 0 & \mathbf{H}_{ii} & 0 & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ H_{n1} & H_{n2} & \cdots & \cdots & \cdots & H_{nn} \end{bmatrix} \quad (3.2)$$

Infine anche il vettore b deve essere modificato, imponendo il valore zero nella posizione del nodo considerato.

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ 0 \\ \vdots \\ b_n \end{bmatrix} \quad (3.3)$$

Che nel codice MATLAB è:

```
1 dd(NodePBC) = 0;
```

3.1.3 Procedura di analisi di dati

Definite le funzioni di base del SIMPLE e del FIXED POINT, per la definizione dei vettori delle portate e delle pressioni, si è proceduto con l'analisi della rete di trasporto.

Si sono imposte le condizioni di estrazione sui nodi relativi alle reti di distribuzione. Nei listati in appendice si fa riferimento al nodo relativo alla centrale termica di Torino NORD con `NetData.PowerPlants.Nodes(2)`. Le condizioni sulla centrale termica di Moncalieri sono invece di pressione e come si è discusso sopra si è imposto un valore `Pimposta=2.5` bar. Una volta impostate le condizioni di funzionamento della rete dalla risoluzione del SIMPLE, la cui impostazione completa è nel listato A.2, si ottengono i vettori relativi alle pressioni nei nodi e alle portate nei rami, Questi risultati sono rappresentati nelle figure 3.3 e 3.4.

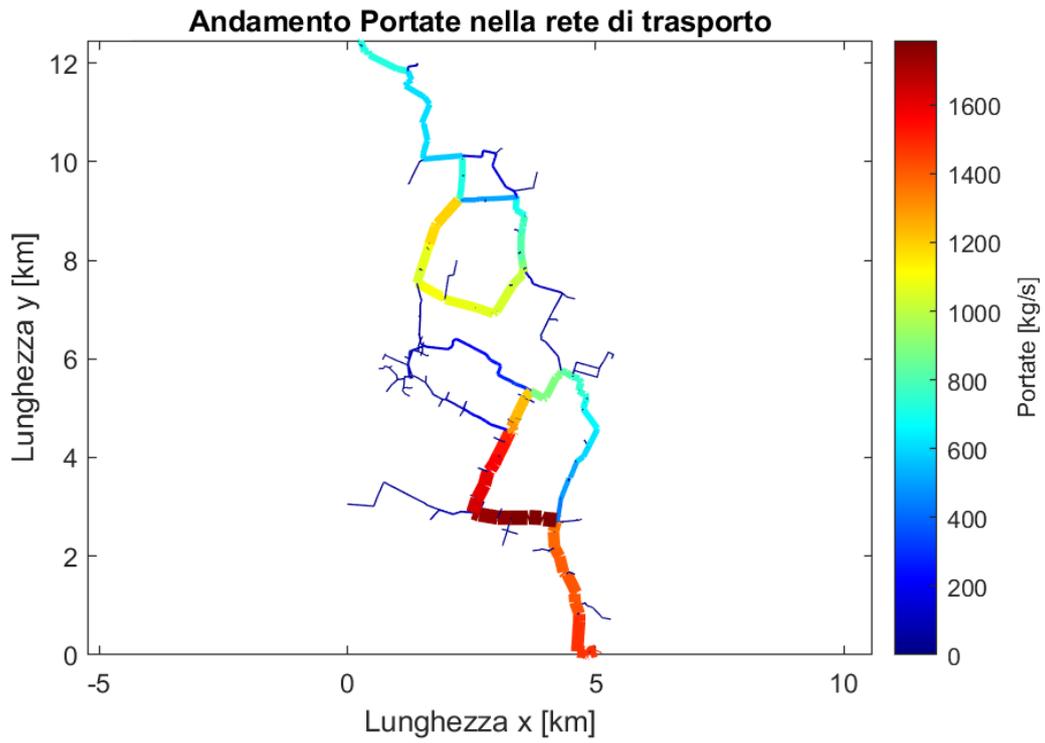


Figura 3.3: Portate rete di trasporto

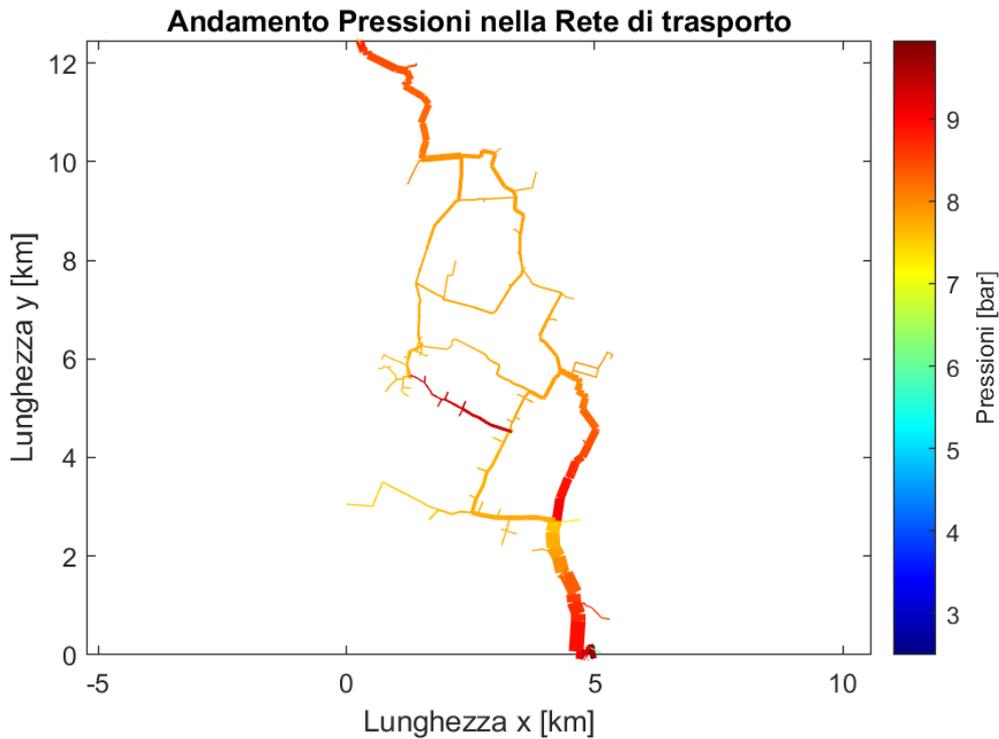


Figura 3.4: Pressioni rete di trasporto

3.1.4 Creazione sottorete

Per rispondere alla domanda relativa all'andamento dei tempi di calcolo in funzione del numero di rami e di nodi si è proceduto come segue.

È stata realizzata una rete costituita da 48 nodi e 47 rami, mostrata in figura 3.5, per la quale sono stati ipotizzati dei versi delle portate e dei nodi di estrazione. L'idea iniziale è stata quella di aggiungere iterativamente una nuova rete alla rete di trasporto esistente, sostituendo di volta in volta uno dei nodi di estrazione della rete di trasporto.

Questo metodo ha permesso di ricostruire la topologia delle reti di distribuzione, che non era disponibile in precedenza.

La distribuzione delle nuove portate estratte è stata effettuata partendo dalla portata originariamente estratta in ciascun nodo. Questa è stata suddivisa tra i nuovi nodi della rete, seguendo il seguente criterio di redistribuzione:

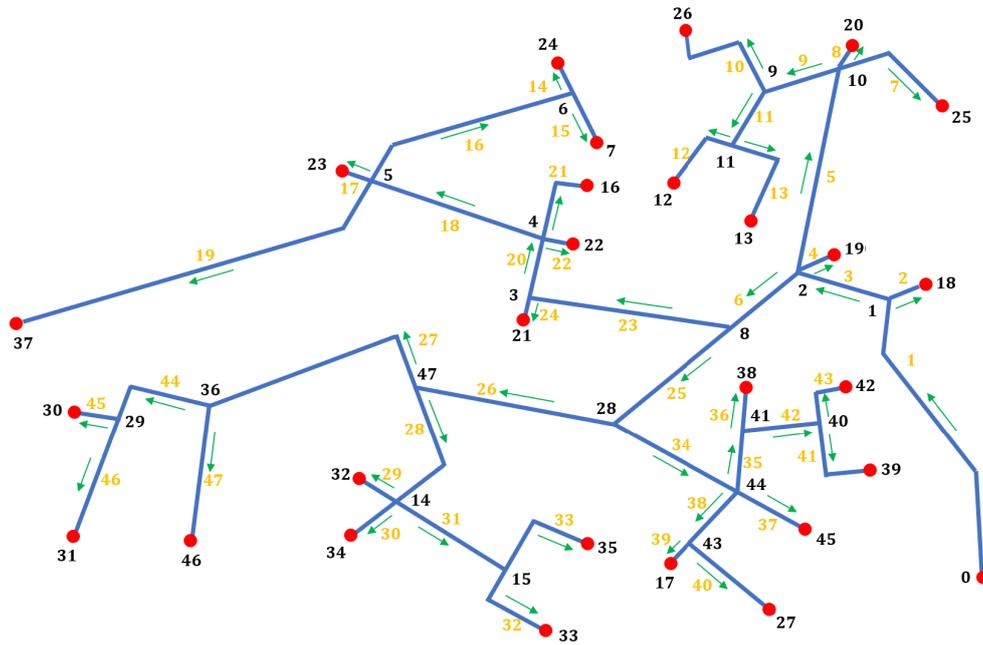


Figura 3.5: Sottorete

- 1. Si seleziona il nodo della rete di trasporto da sostituire
- 2. Per ripartire la portata sui nodi di estrazione della nuova rete, si crea un vettore di ripartizione. Questo vettore assegna un fattore di ripartizione a ciascun nodo di estrazione, garantendo che la somma di tutti i fattori di ripartizione sia pari a 1. In questo modo, si assicura che l'intera portata sia distribuita proporzionalmente tra i nodi di estrazione.
- 3. Si moltiplica la portata estratta per il vettore di ripartizione
- 4. Si assegnano le portate ai nuovi nodi di estrazione

3.2 Metodologia per il calcolo dei tempi

Per valutare i tempi di calcolo al variare del numero di nodi, si sono sostituiti di volta in volta alcuni nodi di estrazione come mostrato in figura 3.6.

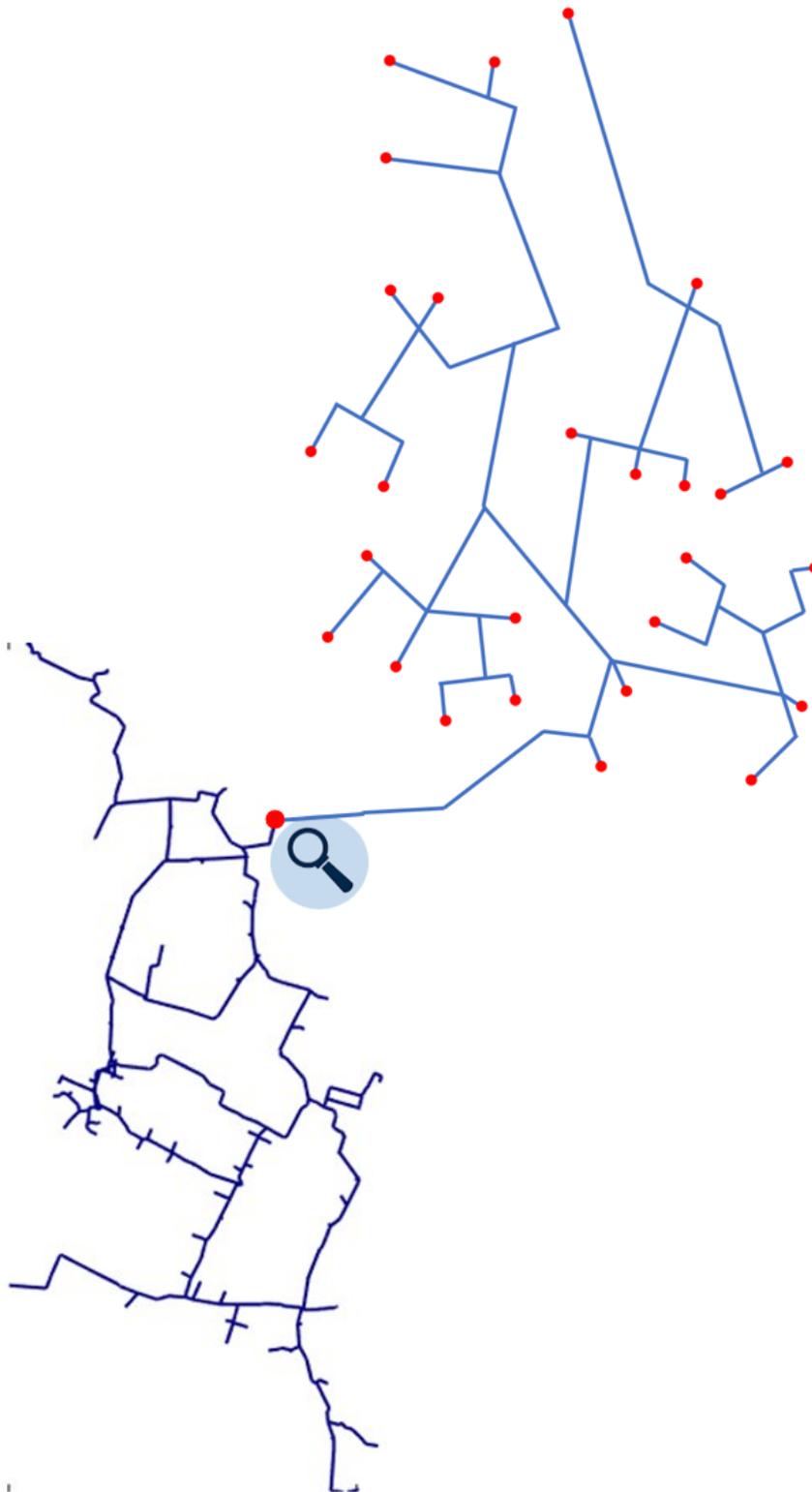


Figura 3.6: Ricostruzione Rete
28

La conseguenza di questa operazione è stata quindi responsabile di alcuni cambiamenti nei vettori e nelle matrici in gioco.

Per riflettere la nuova configurazione bisogna aggiornare la matrice di incidenza. Per fare ciò è necessario effettuare due operazioni principali:

1. Aggiunta di righe e colonne: Per ogni sottorete aggiunta, si devono aggiungere righe e colonne alla matrice di incidenza originale.
2. Sostituzione degli elementi: Una volta che le righe e le colonne sono state aggiunte, è necessario aggiornare gli elementi della matrice di incidenza.

Più nello specifico: La matrice di incidenza è stata modificata per riflettere la nuova configurazione della rete.

- La matrice di incidenza originale è stata ingrandita per aggiungere i dati relativi alla sottorete. sono state aggiunte tante righe quante colonne quanti sono rispettivamente il numero di nodi e di rami della sottorete
- La matrice di incidenza principale è stata modificata per tenere conto del fatto che il nodo di estrazione originario non esiste più come nodo di estrazione. Di conseguenza, la portata estratta da quel nodo è stata impostata a zero.
- Il nodo di estrazione diventa un nodo di transito attraverso il quale la portata fluisce verso i rami della sottorete. Per riflettere questa modifica, nella matrice di incidenza, il nodo originario avrà un valore di -1 per indicare che la portata in uscita da quel nodo confluisce nei rami della sottorete.

Il vettore dei diametri relativo alla sottorete è stato valutato selezionando i diametri tra quelli normati e considerando una velocità massima del fluido di 1.5 m/s. Successivamente, questi diametri sono stati aggiunti, di volta in volta, al vettore dei diametri relativo alla rete di trasporto così come i vettori delle perdite di carico, delle lunghezze dei rami e dei pompaggi.

Questa metodologia permette di simulare l'aumento della complessità della rete, dovuta all'aumento dell'equazioni da dover risolvere, e di osservare come i tempi di calcolo variano in relazione al numero di nodi.

Una volta definiti tutti i parametri relativi alla rete, è possibile procedere alla risoluzione di questa e valutare i tempi di calcolo associati, utilizzando il codice di riferimento, listato A.3.

I valori di output sono stati raccolti in un vettore che ha consentito di creare un grafico numero di nodi - tempo di calcolo, come mostrato nella figura 3.7.

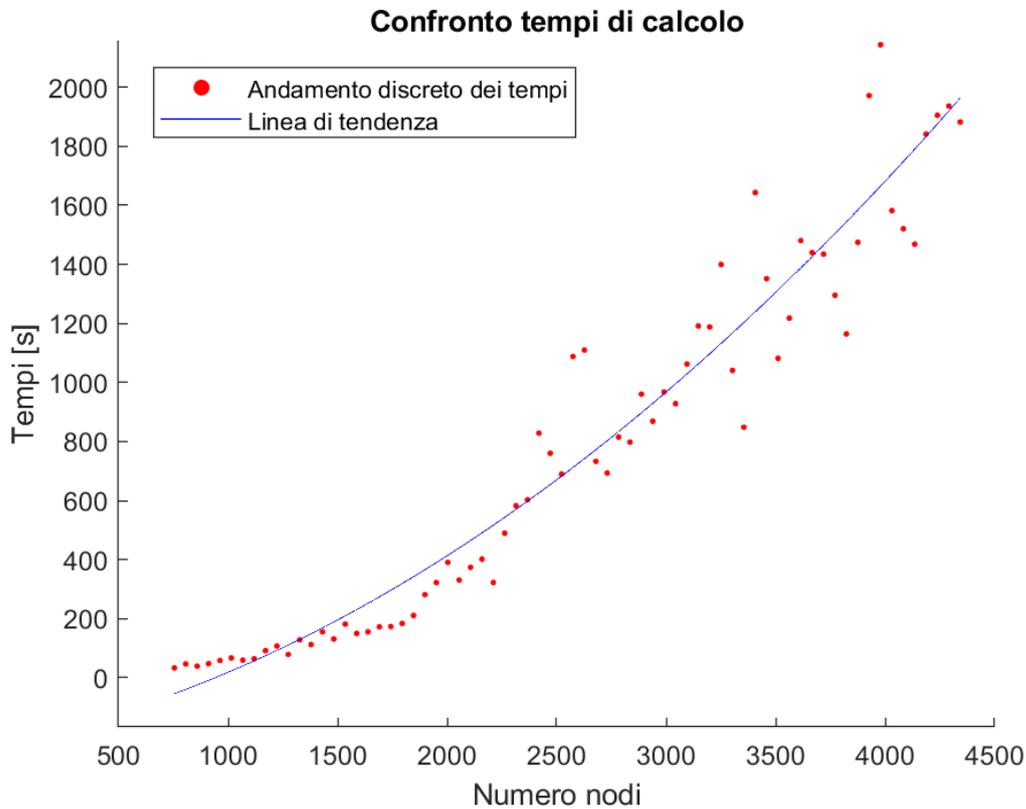


Figura 3.7: Tempi di calcolo

Dove l'equazione della linea di tendenza può essere approssimata come:

$$y = 8 \times 10^{-5}x^2 + 0.1546x - 216.45 \quad (3.4)$$

Questa equazione rappresenta una parabola con concavità verso l'alto, il che implica che il tempo di calcolo y cresce sempre più rapidamente con l'aumentare del numero di nodi x . Quando si affronta un problema di calcolo su una rete con X nodi, suddividere il problema in due reti separate, ciascuna con $X/2$ nodi, porta a una riduzione complessiva del tempo di calcolo. Se si considera una rete con 3000 nodi risolvere il problema due volte con metà dei nodi (1500 nodi per volta) richiede un tempo totale di 390.9 s, mentre risolvere tutto insieme con 3000 nodi richiede un tempo di 967.35 s. Il processo appena descritto è riportato per intero nel listato A.2.

Capitolo 4

Riduzione dei tempi di calcolo

In questo capitolo, verranno trattate le metodologie iterative che permettono di ridurre significativamente i tempi di calcolo, mantenendo al contempo un elevato livello di accuratezza.

Verrà descritto il processo di suddivisione della rete in dettaglio, i criteri utilizzati per scegliere i punti di taglio e l'implementazione di approcci iterativi per aggiornare le condizioni al contorno relative a pressioni e portate fino a raggiungere la convergenza.

La prima fase della ricerca ha avuto come obiettivo lo studio di una soluzione che potesse velocizzare i tempi di calcolo associati a reti di teleriscaldamento di grandi dimensioni.

Come si è visto dal capitolo precedente, i tempi di calcolo aumentano in maniera approssimativamente quadratica rispetto al numero di nodi. Per i motivi illustrati nel capitolo 1, è quindi necessario ricercare dei metodi che possano portare alla risoluzione del problema fluidodinamico in tempi brevi.

L'idea principale per raggiungere questo obiettivo è stata quella di suddividere la rete in due parti.

In modo più specifico, se il numero di nodi raddoppia, il tempo necessario per risolvere il sistema di equazioni associato può aumentare di un fattore molto superiore a due, spesso diventando irrisolvibile per reti di grandi dimensioni. Questo aumento

quadratico è dovuto alla complessità computazionale nella risoluzione dell'equazione di continuità e del momento, che sono alla base del funzionamento della rete.

D'altronde, quando si ha a che fare con un certo numero di nodi, la dimensione della matrice di incidenza cresce rapidamente e, di conseguenza, aumenta il numero di operazioni necessarie per la risoluzione.

Per questo scopo, la prima rete a cui si è fatto riferimento è quella di trasporto, per la quale si hanno maggiori dati a disposizione ed è più facile verificare l'eventuale veridicità del metodo.

Di fatto, se il metodo funziona per questa rete, si può provare ad applicarlo ad altre e vedere come si comporta in altre situazioni.

4.1 Criterio di suddivisione della rete

Durante la fase di suddivisione della rete, sono stati seguiti alcuni punti chiave:

- Sono state utilizzate le coordinate cartesiane dei nodi che costituiscono la rete di trasporto per la visualizzazione e l'analisi della struttura e della distribuzione spaziale della rete stessa. Queste coordinate consentono di rappresentare graficamente la rete, facilitando l'individuazione visiva dei punti ottimali per la suddivisione.

La scelta dei punti di taglio tramite un'analisi visiva si è rivelata la strategia più intuitiva ed efficace per dividere la rete in modo bilanciato, evitando punti critici che potrebbero complicare la risoluzione delle reti.

- La Matrice di Incidenza, generalmente denotata come A , è una matrice in cui le righe rappresentano i nodi e le colonne rappresentano i rami della rete. Ogni elemento a_{ij} della matrice assume il valore di:

1. 1 se il nodo i è l'inizio del ramo j
2. (-1) se il nodo i è la fine del ramo j
3. 0 se il nodo i non è collegato al ramo j

Questa struttura permette di definire chiaramente le connessioni tra i nodi e i rami facilitando l'individuazione dei rami e dei nodi su cui è possibile intervenire. Questo ha permesso di capire su quali righe e colonne della matrice era possibile intervenire per realizzare una divisione efficace.

- Durante la suddivisione della rete, i dati relativi alle portate valutate nei rami sono stati utilizzati per impostare le condizioni al contorno nei punti scelti per la divisione, garantendo che le portate siano coerenti tra le due sottoreti. Questo significa che il flusso totale in ingresso e in uscita dai punti di divisione deve essere mantenuto costante per evitare sbilanciamenti che potrebbero compromettere il funzionamento della rete.
- Nei punti di divisione, si sono impostate le condizioni di pressione che hanno garantito la connessione tra le due sottoreti. Questo ha richiesto l'imposizione di pressioni specifiche nei nodi coinvolti nel taglio, per mantenere l'equilibrio del sistema.

I criteri adottati per la selezione dei punti di taglio hanno incluso due diversi fattori chiave:

1. Minimizzazione delle Interconnessioni: si è preferito scegliere come nodi di taglio quelli con meno interconnessioni poiché riducono la complessità nella gestione delle variabili di flusso. Meno interconnessioni significano meno variabili da monitorare e regolare, semplificando così il bilanciamento delle portate tra le due sottoreti risultanti dalla divisione. Questo approccio aiuta a mantenere la semplicità del sistema e facilita la convergenza durante il processo iterativo di risoluzione.
2. Facilità di lettura delle variabili: I nodi scelti dovevano essere posizionati in punti della rete in cui fosse relativamente semplice impostare e valutare le portate e le pressioni. Questo permette di mantenere condizioni di contorno ben definite e coerenti tra le due sottoreti.

Una volta identificati i punti di taglio, come mostrato in figura 4.1, la rete è stata suddivisa in due sottoreti distinte, ciascuna delle quali include una porzione specifica dei nodi e dei rami della rete originale. Questa suddivisione è stata guidata dalle coordinate cartesiane dei nodi, assicurando una divisione logica e bilanciata attraverso l'implementazione di un codice specifico. Gli step principali di questa implementazione sono illustrati in seguito.

Nella Sottorete Inferiore, in figura 4.2, sono stati inclusi tutti i nodi situati al di sotto dei punti di taglio selezionati. Utilizzando le coordinate associate ai nodi, è stato possibile identificare con precisione quali nodi facessero parte della Sottorete 1 il cui codice di definizione è il A.4 in appendice, dove il nodo Nrif è il nodo 1 in

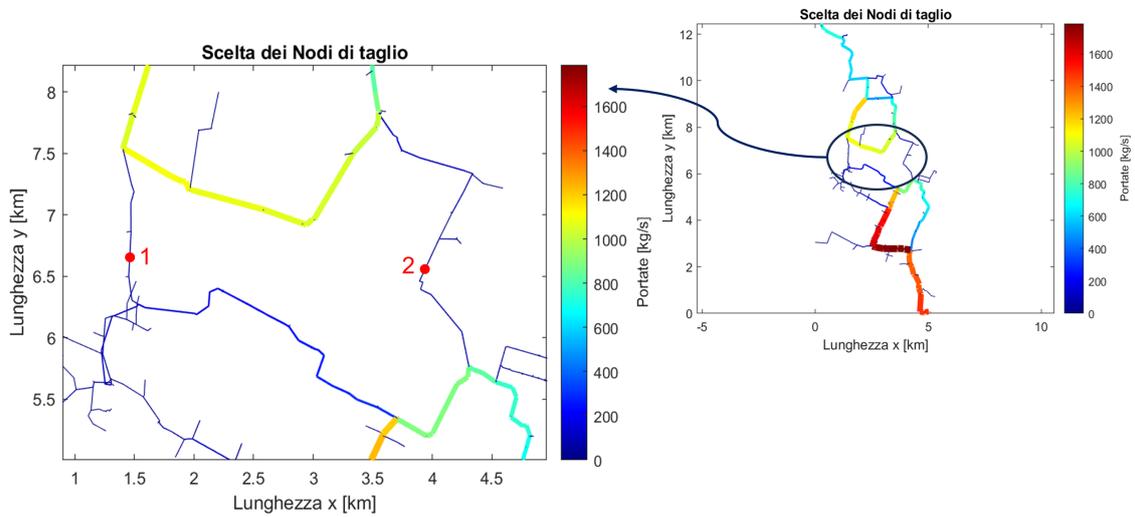


Figura 4.1: Nodi di taglio

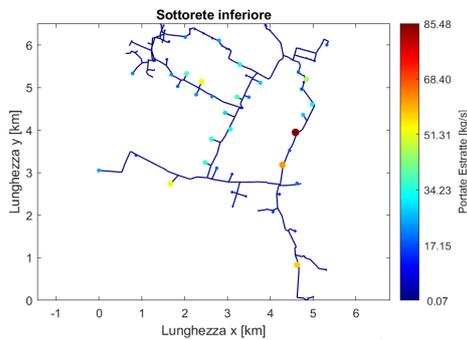


Figura 4.2: Sottorete Inferiore

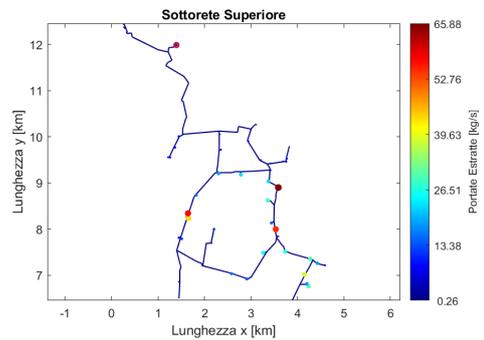


Figura 4.3: Sottorete Superiore

figura 4.1 ed è stato selezionato come punto di riferimento per identificare i nodi appartenenti alla Sottorete 1.

Si è preferito dividere la rete in punti con il minor numero di interconnessioni, al fine di minimizzare le interazioni complesse tra le due sottoreti. La Figura 4.1 mostra i nodi scelti per la divisione.

In particolare, nel nodo a destra è presente un ramo aggiuntivo che collega la rete

a un'utenza; questo è stato attentamente considerato nella valutazione della ripartizione delle portate.

Per prima cosa sono stati individuati tutti i nodi situati nella parte inferiore della rete rispetto al nodo di riferimento.

Tra questi sono stati individuati quelli che i nodi di estrazione tra quelli che erano di estrazione in modo tale da poterne tenere conto in un secondo momento per la valutazione dei nodi da sostituire per ampliare la rete.

Nella Sottorete Superiore, in figura 4.3, sono stati inclusi tutti i nodi situati al di sopra dei punti di taglio.

Analogamente alla Sottorete 1, la determinazione dei nodi appartenenti alla Sottorete 2 è stata guidata dalle coordinate cartesiane, che hanno permesso una chiara distinzione tra le due sottoreti.

Per poter risolvere le due Sottoreti è stato ovviamente necessario imporre delle condizioni al contorno sui nodi di taglio 1 e 2.

Per la Sottorete 1, le portate valutate nei rami e le pressioni nei nodi sono state impostate considerando il flusso totale e le condizioni al contorno imposte nei punti di taglio. È stato essenziale assicurare che le portate di estrazione e di immissione, mostrate in figura 4.4 fossero bilanciate correttamente per mantenere la stabilità del sistema.

4.2 Condizioni al contorno

Una volta ottenuta la separazione fisica delle due sottoreti, si è passati all'imposizione delle condizioni al contorno relative ai nuovi nodi generati dalla suddivisione della rete. Le variabili considerate per impostare le condizioni al contorno sono la pressione e la portata estratta sui nodi 1 e 2.

Per la Sottorete 1, le condizioni al contorno sono imposte sui nodi 1 e 2 in termini di portata estratta.

Osservando la figura 4.4, è possibile formulare il bilancio di massa relativo all'intera rete.

La somma delle portate estratte nei nodi 1 e 2, deve corrispondere alla differenza tra la somma delle portate immesse, il cui unico contributo in questo caso è dato

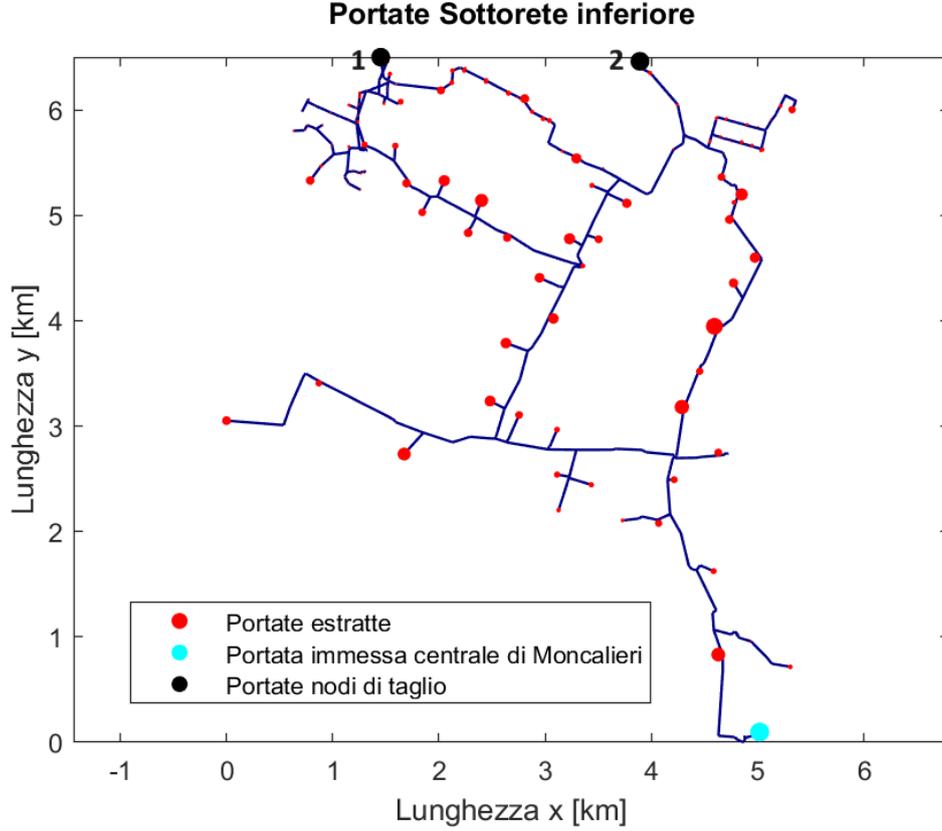


Figura 4.4: Portate Sottorete inferiore

dalla centrale di Moncalieri, e le altre portate estratte dalla Sottorete 1:

$$G_{\text{ext},1} + G_{\text{ext},2} = \sum_i G_{\text{estratta},i} - \sum_i G_{\text{immessa},i} \quad (4.1)$$

Si decide di imporre una portata randomica sul nodo 1, denominata G_1 , e per differenza si calcola e impone la portata estratta sul nodo 2.

Questo approccio consente di mantenere il bilancio di massa e garantisce la corretta distribuzione delle portate nella Sottorete 1.

$$G_{\text{ext},2} = \sum_i G_{\text{estratta},i} - \sum_i G_{\text{immessa},i} - G_1 \quad (4.2)$$

Il passaggio dalla rete di trasporto alle due Sottoreti ha comportato un cambiamento negli indici e nelle relazioni tra rami e nodi rispetto alla configurazione originale della rete. Di conseguenza, è stato necessario sviluppare strategie per identificare e riferirsi ai nodi di interesse nella nuova configurazione della Sottorete,

garantendo che tutti gli step di calcolo si riferissero ai nodi e rami corretti.

Per aggiornare la matrice di incidenza per la Sottorete 1, il primo step è stato quello di eliminare le righe corrispondenti ai nodi costituenti la Sottorete 2. Successivamente, sono state esaminate le colonne rimaste: quelle che sono diventate completamente composte da zeri nella matrice indicano i rami che non sono più connessi ai nodi rimanenti della rete principale dopo l'eliminazione della Sottorete 2.

Questo implica che in questi rami non passa più alcuna portata, di conseguenza devono essere riferiti alla Sottorete 2.

Una volta impostate queste condizioni, è possibile risolvere il problema termofluidodinamico sulla Sottorete 1. La risoluzione fornisce il vettore delle pressioni, in particolare i valori relativi ai nodi 1 e 2 saranno utilizzati per la Sottorete 2 come condizioni al contorno sui corrispondenti nodi. Questo permette di risolvere correttamente il sistema anche per la Sottorete 2.

Vengono valutate le pressioni relative ai nodi 1 e 2 trovate risolvendo la Sottorete 1 come spiegato di seguito:

- Il vettore delle pressioni è stato inizializzato come un vettore di zeri, con dimensione uguale a quella del numero di nodi della rete di trasporto per rappresentare tutte le posizioni dei nodi nella rete originale.
- Si sono individuati gli indici dei nodi che fanno parte della Sottorete 1.
- I valori di pressione ottenuti dalla risoluzione della sottorete 1 sono stati inseriti nel vettore di zeri in corrispondenza dei nodi al punto sopra.
- Si sono potute individuare le pressioni dei nodi di taglio 1 e 2 per imporle come condizioni al contorno sulla sottorete 2.

Si è proceduto poi alla risoluzione della Sottorete 2 allo stesso modo. Il codice completo per la risoluzione della Sottorete 2 è riportato in appendice A.5.

Va osservato che nel processo di divisione le portate di estrazione e quelle immesse dalle centrali termiche nella rete di partenza non cambiano rispetto a come erano state impostate nel Capitolo 2, assicurando che il bilancio complessivo della rete rimanga invariato.

4.3 Metodo Iterativo di Convergenza e Implementazione del Codice

A questo punto si è valutata la convergenza del metodo utilizzato nella rete di trasporto e verificata la correttezza.

La logica seuita è stata:

1. Impostazione del ciclo per le iterazioni di convergenza.

La convergenza del metodo è stata imposta utilizzando un ciclo while. Durante ogni iterazione, la variabile della portata estratta sul nodo di taglio 1 è stata aggiornata prendendo la media tra il valore precedentemente impostato e quello ottenuto dalla risoluzione della sottorete 2.

2. Verifica del funzionamento del metodo sulla rete di trasporto.

Prima di valutare i tempi di calcolo per una rete di grandi dimensioni, il metodo è stato testato su una rete di trasporto più semplice per assicurarsi del suo funzionamento corretto.

3. Test del metodo con portata esatta.

È stato imposto il valore di Portata estratta sul nodo di taglio 1 pari a 38.33 kg/s, che rappresenta la portata che passerebbe attraverso quel nodo se la rete fosse risolta come unica. Questo valore è stato utilizzato per risolvere in sequenza le due semireti:

- Prima, la semirete 1
- Successivamente, la semirete 2

In figura 4.5 e 4.6 sono riportati i grafici relativi ai valori all'andamento delle pressioni e delle portate con i due diversi metodi. Si è quindi proceduto a valutare la differenza tra i valori ottenuti nel Capitolo 2 dalla risoluzione della rete di trasporto, e quelli ottenuti con l'approccio separato come mostrato nel grafico nelle figure 4.7 e 4.8.

Nei grafici riportati, i punti rappresentano le variabili ottenute risolvendo la rete con due diversi approcci: come un'unica unità e con il metodo di convergenza suddividendo la rete in due metà. Nel primo grafico, si considerano le pressioni, mentre nel secondo le portate. Si nota come questi punti si allineino lungo la bisettrice del grafico, indicando una forte corrispondenza tra i valori ottenuti dai due metodi. Da

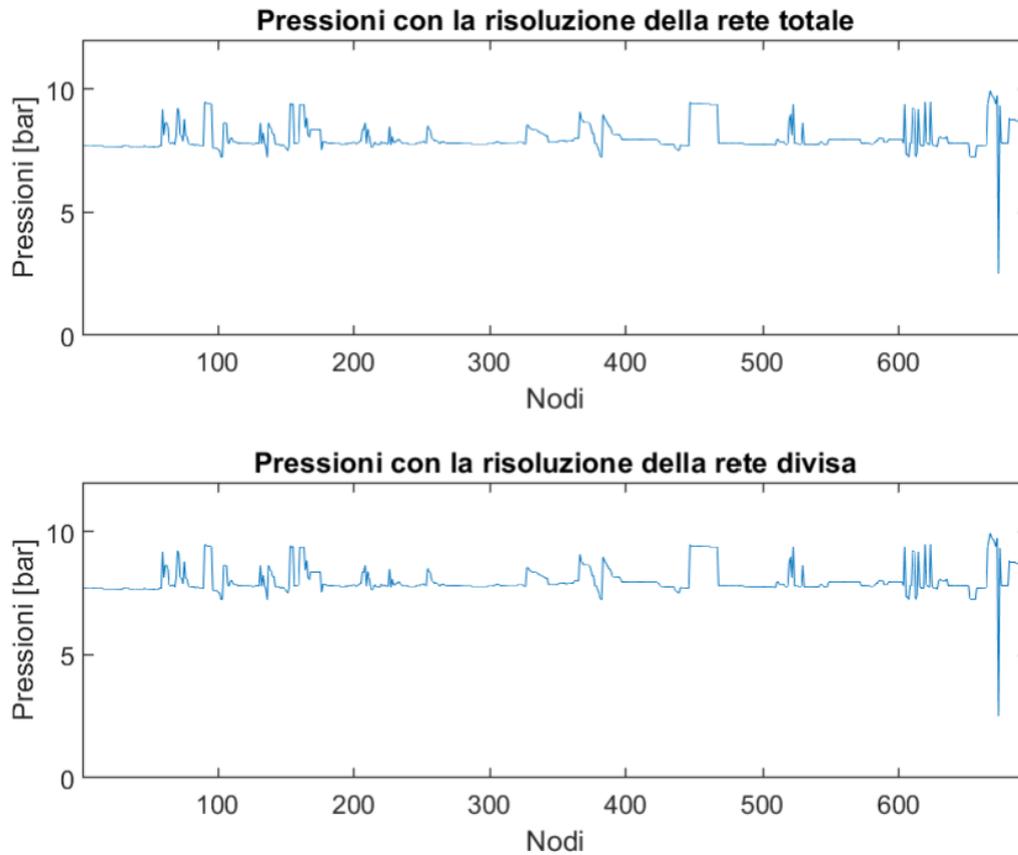


Figura 4.5: Andamento Pressioni

quanto detto si può osservare che i valori in output sono identici, il massimo errore commesso è dello 0.2%, dimostrando che l'approccio descritto fornisce soluzioni conformi con la realtà.

La corrispondenza dei valori ottenuti con quelli teorici conferma che il metodo descritto è valido e può essere applicato anche a reti di dimensioni maggiori, garantendo risultati accurati.

Il ciclo iterativo, per mostrare la convergenza delle soluzioni, funziona con un algoritmo di tipo while rappresentato in figura 4.9 e descritto nel listato A.6. Il ciclo while si interrompe quando la differenza tra la portata impostata sul nodo di taglio 1 durante la risoluzione della Sottorete 1 e la portata calcolata dalla risoluzione della Sottorete 2, che fluisce attraverso il ramo collegato al nodo di taglio 1, diventa molto piccola.

Questo ha permesso di ottenere una soluzione della rete totale che coincidesse con

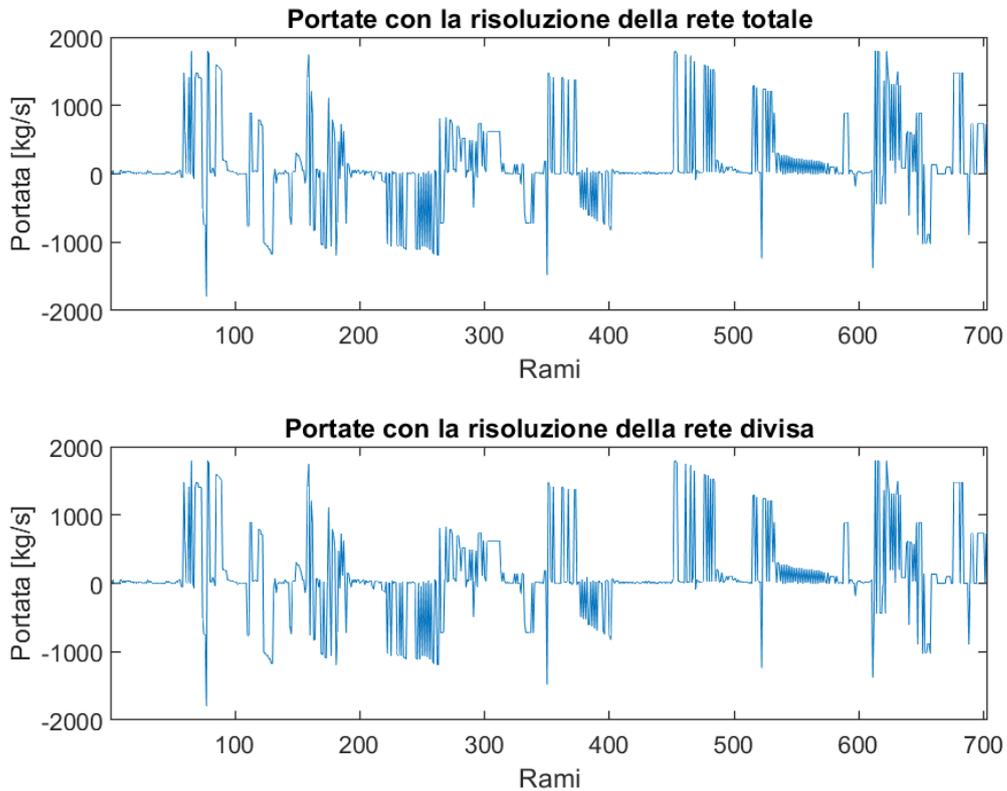


Figura 4.6: Andamento Portate

la risoluzione complessiva.

Si è quindi inizializzata la portata imposta con un valore arbitrario. Successivamente, sono state eseguite nell'ordine le risoluzioni delle Sottoreti 1 e 2 e sono stati importati i vettori delle pressioni e delle portate relativi ad entrambe le sottoreti per poter calcolare l'errore commesso.

La verifica sull'errore è stata fatta anche per le pressioni per cui è stata calcolata la differenza tra i valori dei nodi comuni delle due sottoreti, con l'obiettivo di assicurare la continuità tra le due, si osserva però che il ciclo risultava molto più sensibile alle variazioni di portata.

La prima prova è stata eseguita sulla rete di trasporto. Questo processo iterativo ha permesso di raggiungere la convergenza, garantendo che le soluzioni delle due sottoreti fossero coerenti e rappresentassero accuratamente il comportamento della rete totale. La continua revisione e aggiustamento delle portate e delle pressioni nei nodi comuni ha assicurato che la soluzione finale soddisfacesse i criteri di continuità

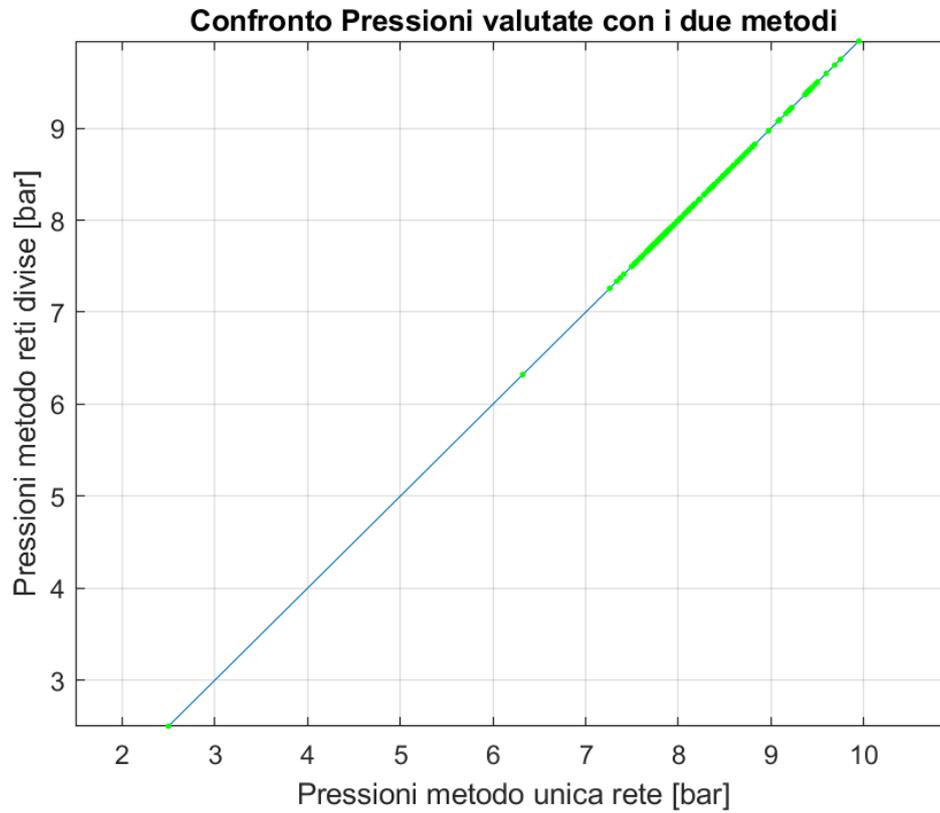


Figura 4.7: Confronto Pressioni determinate con i due metodi

e congruenza tra le due sottoreti.

Grazie a questo approccio, è stato quindi possibile applicare il metodo a reti di dimensioni maggiori e valutare i relativi tempi di calcolo. I risultati di queste valutazioni verranno discussi nel capitolo successivo.

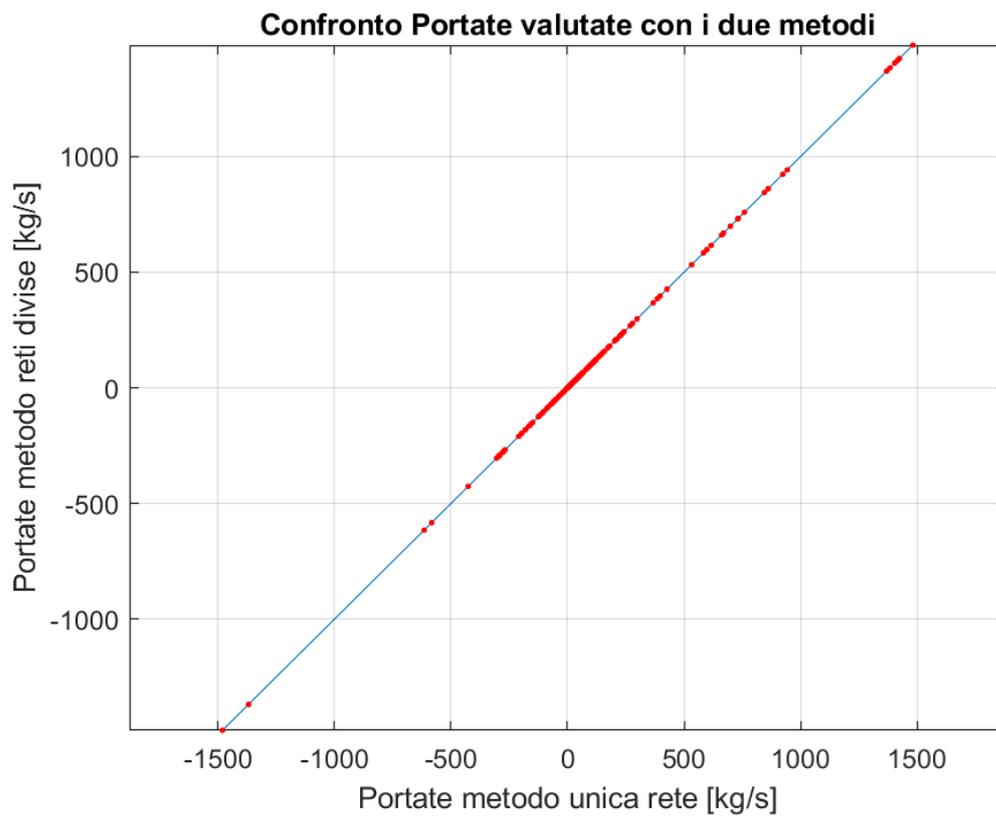


Figura 4.8: Confronto Portate determinate con i due metodi

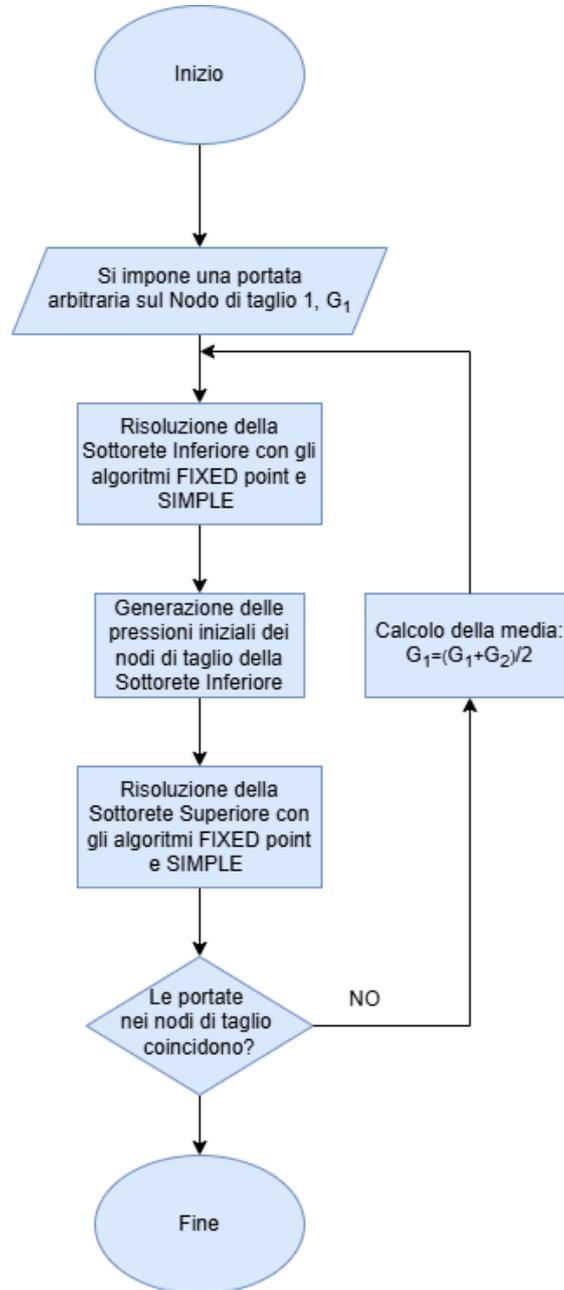


Figura 4.9: Diagramma del Metodo

Capitolo 5

Conclusioni

In questo lavoro, è stata studiata la risoluzione del problema termofluidodinamico relativo alle di reti di teleriscaldamento.

La metodologia adottata ha vistp la suddivisione della rete in sottoreti, e la conseguente riduzione della complessità computazionale Il metodo definito è stato verificato sulla rete di trasporto iniziale, confrontando i risultati ottenuti risolvendo la rete complessiva con quelli ottenuti risolvendo la rete divisa a metà, mostrando che il metodo produce risultati coerenti in entrambi i casi. Questo conferma l'affidabilità del nostro approccio nell'ottenere soluzioni accurate con l'algoritmo implementato e consente il suo utilizzo per diverse tipologie di rete.

5.1 Reti utilizzate

Le reti considerate per la valuatzione dei tempi di calcolo hanno il numero di nodi rappresentato in tabella 5.1. Queste sono state create sostituendo progressivamente

Rete	Nodi
1	1319
2	1943
3	2567
4	3191
5	3815

Tabella 5.1: Numero di nodi nelle reti esaminate

12, 24, 36, 48 e infine 60 nodi di estrazione della rete di trasporto con la sottorete

da 47 nodi descritta nel capitolo 2. I nodi di estrazione sono stati selezionati in modo tale che metà di essi ricadessero nella semirete inferiore e metà nella semirete superiore. Questo approccio ha garantito una divisione equilibrata per quanto riguarda il numero di nodi delle due sottoreti.

In questo paragrafo sono descritti i tempi di calcolo relativi alle reti sopra menzionate, fornendo dettagli specifici per ciascuna rete e spiegandone le caratteristiche. Per tutte le reti considerate, i codici di riferimento per la semirete inferiore e superiore sono rispettivamente i codici A.7 e A.8, riportati in appendice. Il codice per la risoluzione complessiva è invece A.9.

Sebbene questi facciano riferimento alla rete 1 di 1319 nodi, rappresentano la logica utilizzata per tutte le reti. I dettagli delle differenze nella risoluzione delle diverse reti sono spiegati in seguito. Per la risoluzione del metodo separato, il listato a cui si fa riferimento è il A.6; cambiano le righe 4, 5, 29 e 30 in cui si specifica il codice di riferimento a seconda della rete considerata.

5.1.1 Rete 1

La prima rete studiata è composta da 1319 nodi. I codici di riferimento per risolvere questa rete sono quelli menzionati precedentemente. Analizzando i tempi di calcolo, si è osservato che il primo metodo richiede 86 secondi, mentre il metodo sviluppato ne richiede solo 76. Questo indica un risparmio di tempo dell'11,63%.

5.1.2 Rete 2

La seconda rete oggetto di studio è formata da 1943 nodi. Dall'analisi dei tempi, emerge che il primo metodo richiede 245 secondi, mentre il metodo sviluppato ne richiede solo 220 secondi, evidenziando un risparmio di tempo del 11,5%.

5.1.3 Rete 3

La terza rete presa in considerazione conta 2567 nodi. Dall'analisi dei tempi, emerge che il primo metodo richiede 431 secondi, mentre il metodo sviluppato ne impiega soltanto 371 secondi, mostrando un risparmio di tempo del 13,93%.

5.1.4 Rete 4

La quarta rete oggetto di studio è composta da 3191 nodi. Dall'analisi dei tempi, emerge che il primo metodo richiede 753 secondi, mentre il metodo sviluppato impiega solo 621 secondi, indicando un risparmio di tempo del 17,53%.

5.1.5 Rete 5

La quinta rete considerata consiste di 3815 nodi. Dall'analisi dei tempi, emerge che il primo metodo richiede 1367 secondi, mentre il metodo sviluppato ne richiede solo 985 secondi, mostrando un risparmio di tempo del 27,92%.

5.2 Confronto dei tempi di calcolo

A questo punto è possibile trarre le conclusioni del lavoro svolto. come mostrato dalla tabella 5.2, e dal relativo grafico si nota come il metodo sviluppato nel lavoro di tesi risulta essere conveniente in particolare all'aumentare del numero di nodi della rete di teleriscaldamento.

Rete	Nodi	Tempo Totale [s]	Tempo Divisa [s]
1	1319	86	76
2	1943	226	200
3	2567	431	371
4	3191	753	621
5	3815	1367	985

Tabella 5.2: Numero di nodi e tempi di calcolo per le diverse reti

Come si può osservare dalla figura 5.2, per reti con un numero relativamente basso di nodi, in particolare 1319, il metodo implementato riduce il tempo di calcolo di circa il 10%. Questa riduzione diventa ancora più rilevante per reti di dimensioni maggiori, in particolare arriva a quasi il 28% per la rete con 3815 nodi.

Questo dimostra che il metodo proposto è particolarmente vantaggioso per la simulazione di reti di grandi dimensioni.

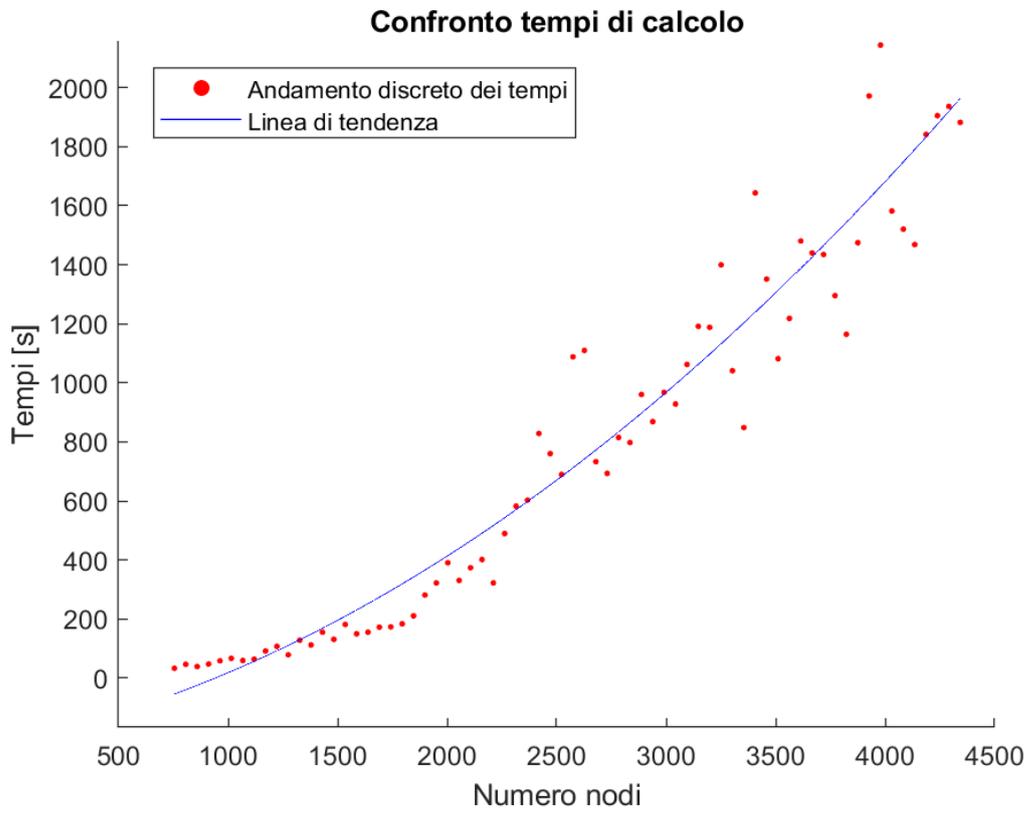


Figura 5.1: Confronto tempi di calcolo

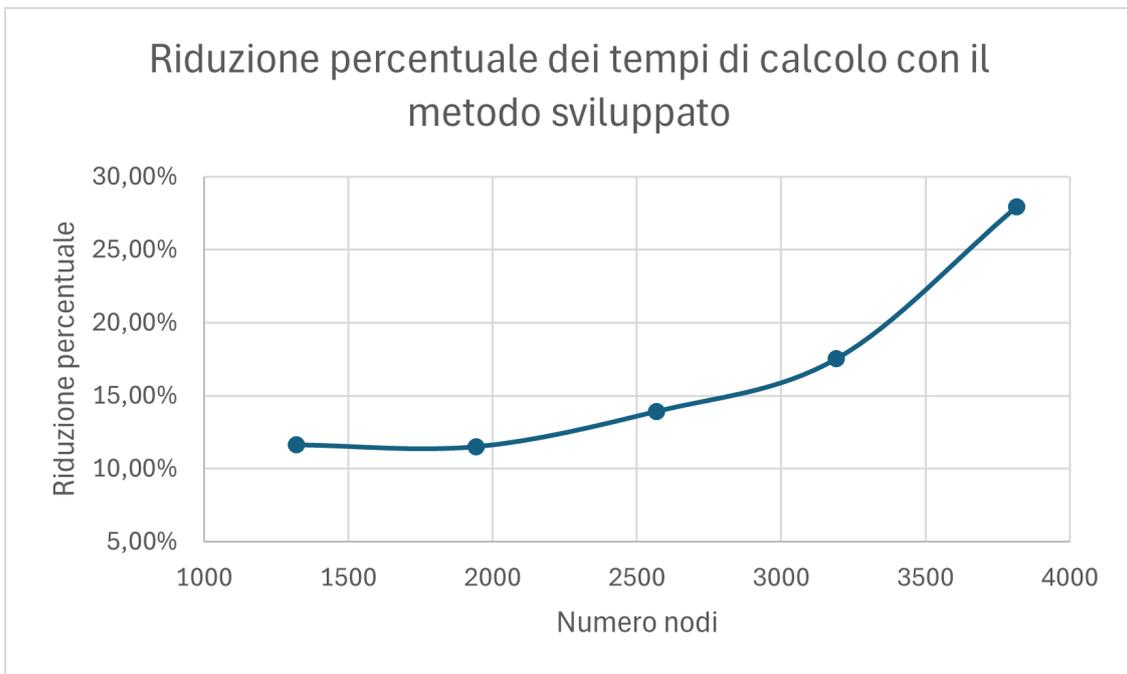


Figura 5.2: Riduzione dei tempi di calcolo con il metodo sviluppato

Appendice A

Appendice

```
1 function zero_columns_indices = trova_colonne_zero(matrice)
2     % Trova gli indici delle colonne contenenti tutti zeri
   nella matrice
3
4     % Inizializza un vettore per memorizzare gli indici delle
   colonne
5     zero_columns_indices = [];
6
7     % Ottieni il numero di colonne nella matrice
8     num_colonne = size(matrice, 2);
9
10    % Loop attraverso tutte le colonne della matrice
11    for i = 1:num_colonne
12        % Verifica se tutti gli elementi della colonna sono
   zero
13        if all(matrice(:, i) == 0)
14            % Se s , aggiungi l'indice della colonna ai
   risultati
15            zero_columns_indices = [zero_columns_indices, i];
16        end
17    end
18 end
```

Listato A.1: Funzione trova_colonne_zero

```
1 clear
2 clc
3 load("NetData.mat")
4 NetData = DatiTrasporto;
5 DatiSolver = load("DatiSolver.mat");
```

```

6 G_ext = zeros(695,1);
7 G_ext(NetData.bar(:,2)) = 0.42 * NetData.bar(:,3);
8 G_ext(NetData.PowerPlants.Nodes(2)) = -0.33 * 0.42 *
    sum(NetData.bar(:,3));
9 NetData.ff = 0.007;
10 NetData.tt = NetData.tt_mandata;
11 NetData.Sezione = (NetData.DD).^2 / 4 * pi;
12 NetData.rr_g = ((NetData.ff .* NetData.LL ./ NetData.DD) +
    NetData.beta) ./ (2000 .* NetData.Sezione.^2);
13 NodePBC = NetData.PowerPlants.Nodes(1);
14 Pimposta = 2.5e5;
15 tic
16 [P, G, Y, Err, res] = SIMPLE(NetData, G_ext, NodePBC,
    DatiSolver, Pimposta);
17 toc

```

Listato A.2: Codice MATLAB per la risoluzione della rete di trasporto

```

1 clear;
2 clc;
3 load("NetData.mat");
4 NetData = DatiTrasporto;
5 DatiSolver = load("DatiSolver.mat");
6 G_bar = 0.42 * NetData.bar(:,3);
7 NodePBC = NetData.PowerPlants.Nodes(1);
8 NetData.ff = 0.007;
9 B = NetData.A;
10 NetData.Sezione = NetData.DD.^2 / 4 * pi;
11 NetData.tt = NetData.tt_mandata;
12 NetData.rr_g = ((NetData.ff .* NetData.LL ./ NetData.DD) +
    NetData.beta) ./ (2000 .* NetData.Sezione.^2);
13 Pimposta = 2e5;
14 Ai = xlsread("Ai.xlsx", "B3:BF54");
15 db = xlsread("Ai.xlsx", "B60:E60");
16 Ai(:, db) = [];
17 %% Calcolo diametri sottorete
18 DDs = zeros(52, 150);
19 Ai2 = [zeros(size(Ai)); zeros(1, 52)];
20 Ai2(end, 1) = 1;
21 Ai2(1:end-1, 1:end) = Ai;
22 n_ext = xlsread("Ai.xlsx", "B59:AB59");
23 Gext_s = zeros(53, 30);
24 Gs = zeros(52, 30);

```

```

25 for i = 1:150
26     Gext_s(end, i) = -0.42 * abs(NetData.bar(i, 3));
27     n = length(n_ext);
28     vettore = linspace(0.03, 1, n);
29     vettore_normalizzato = vettore / sum(vettore);
30     Gext_s(n_ext, i) = -Gext_s(end, i) .* vettore_normalizzato;
31     Gs(:, i) = Ai2 \ (-Gext_s(:, i));
32     rho_H2O = 990; %[kg/m3]
33     vel_max = 1.5; %[m/s]
34     diam_uni = 10^-3 * [0 0 50 75 100 125 250 300 350 400 450
35         500 600 800]';
36     DDs(:, i) = 2 * sqrt(abs(Gs(:, i)) / (pi * rho_H2O *
37         vel_max)); %[m]
38     for k = 1:length(DDs(:, i))
39         diff = diam_uni - DDs(k, i);
40         ind = find(diff > 0, 1);
41         DDs(k, i) = diam_uni(ind);
42     end
43 end
44 %% Risoluzione rete ampliata
45 Gext_s(end, :) = [];
46 Tnn = zeros(150, 1);
47 NNR = zeros(150, 1);
48 for iii = 1:150
49     dimensione1 = [695 + iii * length(Ai(:, 1)) 703 + iii *
50         length(Ai(1, :))];
51     B = zeros(dimensione1);
52     B(1:695, 1:703) = NetData.A(1:695, 1:703);
53     G_ext = zeros(695 + iii * length(Ai(:, 1)), 1);
54     for l11 = 1:iii
55         B(696 + (l11 - 1) * length(Ai(1, :)):695 + l11 *
56             length(Ai(:, 1)), 704 + (l11 - 1) * length(Ai(1,
57                 :)):703 + l11 * length(Ai(1, :))) = Ai;
58         B(NetData.bar(l11, 2), 703 + 1 + (l11 - 1) *
59             length(Ai(:, 1))) = 1;
60         G_ext(NetData.bar(:, 2)) = G_bar;
61         G_ext(NetData.PowerPlants.Nodes(2)) = -0.33 *
62             sum(G_bar);
63         G_ext(696 + (l11 - 1) * length(Ai(:, 1)):695 + l11 *
64             length(Ai(:, 1))) = Gext_s(:, l11);
65     end
66 end
67 NetData.A = B;

```

```

59 NetData.DD = [NetData.DD; DDs(:, iii)];
60 NetData.LL = [NetData.LL; NetData.LL(1:length(DDs(:, 1)))];
61 NetData.tt_mandata = [NetData.tt_mandata;
    zeros(length(DDs(:, 1)), 1)];
62 NetData.zz = [NetData.zz; zeros(length(Ai(1, :)), 1)];
63 NetData.beta = [NetData.beta; 2 * ones(length(DDs(:, 1)),
    1)];
64 NetData.Sezione = NetData.DD.^2 / 4 * pi;
65 NetData.tt = NetData.tt_mandata;
66 NetData.nn = 695 + iii * length(Ai(:, 1));
67 NetData.nr = 703 + iii * length(Ai(1, :));
68 NetData.rr_g = ((NetData.ff .* NetData.LL ./ NetData.DD) +
    NetData.beta) ./ (2000 .* NetData.Sezione.^2);
69 tic;
70 [P, G, Y, Err, res] = SIMPLE(NetData, G_ext, NodePBC,
    DatiSolver, Pimposta);
71 toc;
72 Tnn(iii) = toc;
73 NNR(iii) = NetData.nr;
74 resssss(iii) = res;
75 end
76 xlswrite("Tnn.xlsx", Tnn);
77 xlswrite("NNR.xlsx", NNR);
78 xlswrite("res.xlsx", resssss);

```

Listato A.3: Codice MATLAB per la risoluzione della rete ampliata

```

1 Gplus11 = 33.38;
2 load('NetData.mat')
3 NetData = DatiTrasporto;
4 DatiSolver = load('DatiSolver.mat');
5 % load('sharedData.mat', 'Gplus11')
6 Nrif = NetData.Coord(516,2);
7 ri = find(NetData.Coord(:,2) <= Nrif);
8 differenza = NetData.Coord(:,2) - Nrif;
9 NodePBC = 667;
10 P1 = zeros(695,1);
11 % P1(267) = 1;
12 P1(NodePBC) = 1;
13 P1 = P1(ri);
14 P1NODEPBC = find(P1 == 1);
15 C = sort(differenza, 'descend');
16 [ind, ~] = find(C <= 0);

```

```

17 D = C(ind(1:2)) + Nrif;
18 [ind2, ~] = find(ismember(NetData.Coord(:,2), D));
19 [ind3, ~] = find(C > 0);
20 G_retsup = find(ismember(NetData.Coord(:,2), C(ind3) + Nrif));
21 [ci, ~] = find(ismember(NetData.bar(:,2), ri));
22 Gexsup = sum(NetData.bar(:,3)) - sum(NetData.bar(ci,3));
23 G_ext = zeros(695,1);
24 G_ext(NetData.PowerPlants.Nodes(1)) = -0.42 *
    sum(NetData.bar(:,3));
25 ind2_dx = find(NetData.Coord(:,2) ==
    max(NetData.Coord(find(NetData.Coord(:,2) < 6464115 &
    NetData.Coord(:,1) > 2.5e6),2)));
26 G_ext(NetData.bar(:,2)) = 0.42 * NetData.bar(:,3);
27 G_ext(516) = Gplus11 * 0.42 * Gexsup;
28 G_ext(267) = (1 - Gplus11) * 0.42 * Gexsup;
29 G_ext = G_ext(ri);
30 NetData.bar = NetData.bar(ci,:);
31 [ind7, ~] =
    find(NetData.Coord(NetData.bar(find(ismember(NetData.bar(:,2),
    G_retsup)),2),1) < 2784100);
32 %
    plot(NetData.Coord(NetData.bar(:,2),1),NetData.Coord(NetData.bar(:,2),2),'o');
    hold on;
33 % NetData.Coord = NetData.Coord(ri,:);
34 NetData.A = NetData.A(ri,:);
35 zero_columns_indices = trova_colonne_zero(NetData.A);
36 CC = find(sum(NetData.A,1));
37 OO = 1:1:703;
38 X = setdiff(OO, [zero_columns_indices, CC]);
39 NetData.A(:, [zero_columns_indices, CC]) = [];
40 % n1 = find(G_ext == Gplus11 * 0.42 * Gexsup);
41 NetData.ff = 0.007;
42 NetData.tt_mandata([zero_columns_indices, CC]) = [];
43 NetData.tt = NetData.tt_mandata;
44 NetData.DD([zero_columns_indices, CC]) = [];
45 NetData.Sezione = (NetData.DD).^2 / 4 * pi;
46 NetData.beta([zero_columns_indices, CC]) = [];
47 NetData.LL([zero_columns_indices, CC]) = [];
48 NetData.nn = length(NetData.A(:,1));
49 NetData.nr = length(NetData.A(1,:));
50 NetData.zz = NetData.zz(ri);

```

```

51 NetData.rr_g = ((NetData.ff .* NetData.LL ./ NetData.DD) +
    NetData.beta) ./ (2000 .* NetData.Sezione.^2);
52 NodePBC = P1NODEPBC;
53 % NodePBC2 = P1NODEPBC(1);
54 Pimposta = 2.5e5;
55 [P, G, Y, Err, res] = SIMPLE(NetData, G_ext, NodePBC,
    DatiSolver, Pimposta);

```

Listato A.4: Risoluzione Sottorete 1

```

1 % Inizializzazione del vettore delle pressioni della Sottorete
  2
2 P1 = zeros(695, 1);
3 P1(516) = 1;
4 P1(267) = 1;
5 P1 = P1(ri);
6 P1NODEPBC = find(P1 == 1);
7 [ci, val] = find(ismember(NetData.bar(:, 2), ri));
8
9 % Impostazione delle portate estratte
10 G_ext = zeros(695, 1);
11 G_ext(NetData.bar(:, 2)) = 0.42 * NetData.bar(:, 3);
12 G_ext(NetData.PowerPlants.Nodes(2)) = -0.33 * 0.42 *
    sum(NetData.bar(:, 3));
13 G_ext = G_ext(ri);
14 NetData.bar = NetData.bar(ci, :);
15 NetData.Coord = NetData.Coord(ri, :);
16 NetData.A = NetData.A(ri, :);
17
18 % Rimozione delle colonne vuote dalla matrice di incidenza
19 zero_columns_indices = trova_colonne_zero(NetData.A);
20 CC = find(sum(NetData.A, 1));
21 OO = 1:1:703;
22 X = setdiff(OO, [zero_columns_indices, CC]);
23 NetData.A(:, [zero_columns_indices, CC]) = [];
24
25 % Parametri della rete
26 NetData.ff = 0.007;
27 NetData.tt_mandata([zero_columns_indices, CC]) = [];
28 NetData.tt = NetData.tt_mandata;
29 NetData.DD([zero_columns_indices, CC]) = [];
30 NetData.Sezione = (NetData.DD).^2 / 4 * pi;
31 NetData.beta([zero_columns_indices, CC]) = [];

```

```

32 NetData.LL([zero_columns_indices, CC]) = [];
33 NetData.nn = length(NetData.A(:, 1));
34 NetData.nr = length(NetData.A(1, :));
35 NetData.zz = NetData.zz(ri);
36 NetData.rr_g = ((NetData.ff .* NetData.LL ./ NetData.DD) +
    NetData.beta) ./ (2000 .* NetData.Sezione.^2);
37
38 % Condizioni al contorno di pressione sui nodi 267 e 516
39 Pimposta = Pcc1;
40 Pimposta2 = Pcc3;
41 NodePBC = P1NODEPBC(1);
42 NodePBC2 = P1NODEPBC(2);
43 [P, G, Y, Err, res] = SIMPLE2(NetData, G_ext, NodePBC,
    NodePBC2, DatiSolver, Pimposta, Pimposta2);

```

Listato A.5: Risoluzione Sottorete 2

```

1 clear
2 clc
3 Gplus11 = 20;
4 run('Semirete_giu21.m')
5 run('Semirete_su21.m')
6
7 ri = xlsread("ri.xlsx");
8 ri2 = xlsread("ri2.xlsx");
9 X = xlsread("X.xlsx");
10 X2 = xlsread("X2.xlsx");
11 Psu = xlsread("PPPsu.xlsx");
12 Pgiu = xlsread("PPPgiu.xlsx");
13 Gsu = xlsread("GGGsu.xlsx");
14 Ggiu = xlsread("GGGgiu.xlsx");
15
16 P11(ri, 1) = Psu;
17 P12(ri2, 1) = Pgiu;
18 G11(X, 1) = Gsu;
19 G12(X2, 1) = Ggiu;
20
21 err2 = abs(G11(593) - (G12(194) - G12(592)));
22 err1 = abs(P11(267) - P12(267));
23 pl = 1;
24
25 tic
26 while err2 > 0.01

```

```

27     pl = pl + 1;
28     Gplus11 = (Gplus11 + Gcc3) / 2;
29     run('Semirete_giu21.m')
30     run('Semirete_su21.m')
31
32     Psu = xlsread("PPPsu.xlsx");
33     Pgiu = xlsread("PPPgiu.xlsx");
34     Gsu = xlsread("GGGsu.xlsx");
35     Ggiu = xlsread("GGGgiu.xlsx");
36
37     P11 = zeros(695, 1);
38     G11 = zeros(703, 1);
39
40     P11(ri, 1) = Psu;
41     P12(ri2, 1) = Pgiu;
42     G11(X, 1) = Gsu;
43     G12(X2, 1) = Ggiu;
44
45     err2 = abs(G11(593) - (G12(194) - G12(592)));
46     err1 = abs(P11(267) - P12(267));
47 end
48 toc

```

Listato A.6: Codice per la risoluzione della rete divisa in due

```

1 load("NetData.mat")
2 NetData = DatiTrasporto;
3 DatiSolver = load("DatiSolver.mat");
4 Nrif = NetData.Coord(516, 2);
5 ri = find(NetData.Coord(:, 2) <= Nrif);
6 differenza = NetData.Coord(:, 2) - Nrif;
7 NodePBC = NetData.PowerPlants.Nodes(1);
8 P1 = zeros(695, 1);
9 P1(NodePBC) = 1;
10 P1 = P1(ri);
11 P1NODEPBC = find(P1 == 1);
12 C = sort(differenza, 'descend');
13 finto = zeros(695, 1);
14 finto(267) = 1;
15 finto(516) = 1;
16 finto = finto(ri);
17 node_inc = find(finto == 1);
18 [ind, ~] = find(C <= 0);

```

```

19 D = C(ind(1:2)) + Nrif;
20 [ind2, ~] = find(ismember(NetData.Coord(:, 2), D));
21 [ind3, ~] = find(C > 0);
22 G_retsup = find(ismember(NetData.Coord(:, 2), C(ind3) + Nrif));
23 [ci, val] = find(ismember(NetData.bar(:, 2), ri));
24 di = zeros(695, 1);
25 di(NetData.bar(ci, 2)) = 1;
26 di = di(ri);
27 vi = find(di == 1);
28 Gexsup = sum(NetData.bar(:, 3)) - sum(NetData.bar(ci, 3));
29 G_ext = zeros(695, 1);
30 ind2_dx = find(NetData.Coord(:, 2) ==
    max(NetData.Coord(find(NetData.Coord(:, 2) < 6464115 &
    NetData.Coord(:, 1) > 2.5e6), 2)));
31 G_ext(NetData.bar(:, 2)) = 0.42 * NetData.bar(:, 3);
32 G_ext = G_ext(ri);
33 G_ext(node_inc(2)) = 0.42 * 0.67 * sum(NetData.bar(:, 3)) -
    sum(G_ext) - Gplus11;
34 G_ext(node_inc(1)) = Gplus11;
35 NetData.bar = NetData.bar(ci, :);
36 [ind7, ~] =
    find(NetData.Coord(NetData.bar(find(ismember(NetData.bar(:,
    2), G_retsup)), 2), 1) < 2784100);
37 NetData.Coord = NetData.Coord(ri, :);
38 NetData.A = NetData.A(ri, :);
39 zero_columns_indices = trova_colonne_zero(NetData.A);
40 CC = find(sum(NetData.A, 1));
41 OO = 1:703;
42 X = setdiff(OO, [zero_columns_indices, CC]);
43 NetData.A(:, [zero_columns_indices, CC]) = [];
44 NetData.ff = 0.007;
45 NetData.tt_mandata([zero_columns_indices, CC]) = [];
46 NetData.tt = NetData.tt_mandata;
47 NetData.DD([zero_columns_indices, CC]) = [];
48 NetData.Sezione = (NetData.DD).^2 / 4 * pi;
49 NetData.beta([zero_columns_indices, CC]) = [];
50 NetData.LL([zero_columns_indices, CC]) = [];
51 NetData.nn = length(NetData.A(:, 1));
52 NetData.nr = length(NetData.A(1, :));
53 NetData.zz = NetData.zz(ri);
54 Ai = xlsread("Ai.xlsx", "B3:BF54");
55 db = xlsread("Ai.xlsx", "B60:E60");

```

```

56 Ai(:, db) = [];
57 DDs = zeros(52, 6);
58 Ai2 = [zeros(size(Ai)); zeros(1, 52)];
59 Ai2(end, 1) = 1;
60 Ai2(1:end-1, 1:end) = Ai;
61 n_ext = xlsread("Ai.xlsx", "B59:AB59");
62 Gext_s = zeros(53, 6);
63 Gs = zeros(52, 6);
64 for i = 1:6
65     Gext_s(end, i) = -0.42 * abs(NetData.bar(i, 3));
66     G_ext(vi(i)) = 0;
67     n = length(n_ext);
68     vettore = linspace(0.03, 1, n);
69     vettore_normalizzato = vettore / sum(vettore);
70     Gext_s(n_ext, i) = -Gext_s(end, i) .* vettore_normalizzato;
71     Gs(:, i) = Ai2 \ (-Gext_s(:, i));
72     rho_H2O = 990; %[kg/m3]
73     vel_max = 1.5; %[m/s]
74     diam_uni = 10^-3 * [0 0 50 75 100 125 250 300 350 400 450
75         500 600 800]';
76     DDs(:, i) = 2 * sqrt(abs(Gs(:, i)) / (pi * rho_H2O *
77         vel_max)); %[m]
78     for k = 1:length(DDs(:, i))
79         diff = diam_uni - DDs(k, i);
80         ind = find(diff > 0, 1); %controllare
81         DDs(k, i) = diam_uni(ind);
82     end
83 end
84 Gext_s(end, :) = [];
85 Dngiu = reshape(DDs, length(DDs(:, 1)) * length(DDs(1, :)), 1);
86 dimensione1 = [length(NetData.A(:, 1)) + 12 * length(Ai(:,
87     1)), length(NetData.A(1, :)) + 12 * length(Ai(1, :))];
88 B = zeros(dimensione1);
89 B(1:length(NetData.A(:, 1)), 1:length(NetData.A(1, :))) =
90     NetData.A(1:length(NetData.A(:, 1)), 1:length(NetData.A(1,
91     :)));
92 for l11 = 1:6
93     B(length(NetData.A(:, 1)) + 1 + (l11 - 1) * length(Ai(1,
94         :)):length(NetData.A(:, 1)) + l11 * length(Ai(:, 1)),
95         length(NetData.A(1, :)) + 1 + (l11 - 1) * length(Ai(1,
96         :)):length(NetData.A(1, :)) + l11 * length(Ai(1, :))) =
97         Ai;

```

```

89     B(vi(111), length(NetData.A(1, :)) + 1 + (111 - 1) *
        length(Ai(:, 1))) = -1;
90 end
91 NetData.A = B;
92 NetData.DD = [NetData.DD; Dngiu];
93 NetData.LL = [NetData.LL; 12 * ones(length(Dngiu), 1)];
94 NetData.nr = length(NetData.A(1, :));
95 NetData.nn = length(NetData.A(:, 1));
96 NetData.tt = [NetData.tt_mandata; zeros(length(Dngiu), 1)];
97 NetData.beta = [NetData.beta; 3 * ones(length(Dngiu), 1)];
98 NetData.zz = [NetData.zz; zeros(12 * length(Ai(1, :)), 1)];
99 NetData.Sezione = NetData.DD.^2 / 4 * pi;
100 Gngiu = (reshape(Gext_s, length(Gext_s(:, 1)) *
        length(Gext_s(1, :)), 1));
101 G_ext = [G_ext; Gngiu];
102 NetData.rr_g = ((NetData.ff .* NetData.LL ./ NetData.DD) +
        NetData.beta) ./ (2000 * NetData.Sezione.^2);
103 NodePBC = P1NODEPBC;
104 Pimposta = 2.5e5;
105 [P, G, Y, Err, res] = SIMPLE(NetData, G_ext, NodePBC,
        DatiSolver, Pimposta);
106 xlswrite("PPPgiu.xlsx", P);
107 xlswrite("GGGgiu.xlsx", G);
108 PP = zeros(length(NetData.A(1, :)), 1);
109 PP(ri) = P(1:length(ri));
110 Pcc2 = PP(267);
111 GG = zeros(length(NetData.A(:, 1)), 1);
112 GG(X) = G(1:length(X));
113 Pcc1 = PP(516);
114 Pcc3 = PP(267);
115 Gcc1 = GG(155) - GG(201);
116 PPGiu = xlswrite("Pgiu.xlsx", PP);

```

Listato A.7: Risoluzione Semirete inferiore per reti di grandi dimensioni

```

1 load("NetData.mat")
2 NetData = DatiTrasporto;
3 DatiSolver = load("DatiSolver.mat");
4 mini = NetData.Coord(516, 2);
5 ri = [find(NetData.Coord(:, 2) >= mini); 267];
6 ri = sort(ri);
7 ind2_1 = 267;
8 P1 = zeros(695, 1);

```

```
9 P1(516) = 1;
10 P1(267) = 1;
11 P1 = P1(ri);
12 P1NODEPBC = find(P1 == 1);
13 [ci, val3] = find(ismember(NetData.bar(:, 2), ri));
14 di = zeros(695, 1);
15 di(NetData.bar(ci, 2)) = 1;
16 di = di(ri);
17 vi = find(di == 1);
18 G_ext = zeros(695, 1);
19 G_ext(NetData.bar(:, 2)) = 0.42 * NetData.bar(:, 3);
20 G_ext(NetData.PowerPlants.Nodes(2)) = -0.33 * 0.42 *
    sum(NetData.bar(:, 3));
21 G_ext = G_ext(ri);
22 NetData.bar = NetData.bar(ci, :);
23 NetData.A = NetData.A(ri, :);
24 zero_columns_indices = trova_colonne_zero(NetData.A);
25 CC = find(sum(NetData.A, 1));
26 OO = 1:703;
27 X = setdiff(OO, [zero_columns_indices, CC]);
28 NetData.A(:, [zero_columns_indices, CC]) = [];
29 NetData.ff = 0.007;
30 NetData.tt_mandata([zero_columns_indices, CC]) = [];
31 NetData.tt = NetData.tt_mandata;
32 NetData.DD([zero_columns_indices, CC]) = [];
33 NetData.Sezione = (NetData.DD).^2 / 4 * pi;
34 NetData.beta([zero_columns_indices, CC]) = [];
35 NetData.LL([zero_columns_indices, CC]) = [];
36 NetData.nn = length(NetData.A(:, 1));
37 NetData.nr = length(NetData.A(1, :));
38 NetData.zz = NetData.zz(ri);
39 Ai = xlsread("Ai.xlsx", "B3:BF54");
40 db = xlsread("Ai.xlsx", "B60:E60");
41 Ai(:, db) = [];
42 %% calcolo diametri sottorete
43 DDs = zeros(52, 6);
44 Ai2 = [zeros(size(Ai)); zeros(1, 52)];
45 Ai2(end, 1) = 1;
46 Ai2(1:end-1, 1:end) = Ai;
47 n_ext = xlsread("Ai.xlsx", "B59:AB59");
48 Gext_s = zeros(53, 6);
49 Gs = zeros(52, 6);
```

```

50 for i = 1:6
51     Gext_s(end, i) = -0.42 * abs(NetData.bar(i, 3));
52     G_ext(vi(i)) = 0;
53     n = length(n_ext);
54     vettore = linspace(0.03, 1, n);
55     vettore_normalizzato = vettore / sum(vettore);
56     Gext_s(n_ext, i) = -Gext_s(end, i) .* vettore_normalizzato;
57     Gs(:, i) = Ai2 \ (-Gext_s(:, i));
58     rho_H2O = 990;           % [kg/m3]
59     vel_max = 1.5;          % [m/s]
60     diam_uni = 10^-3 * [0 0 50 75 100 125 250 300 350 400 450
        500 600 800]';
61     DDs(:, i) = 2 * sqrt(abs(Gs(:, i)) / (pi * rho_H2O *
        vel_max));           % [m]
62     for k = 1:length(DDs(:, i))
63         diff = diam_uni - DDs(k, i);
64         ind = find(diff > 0, 1); % controllare
65         DDs(k, i) = diam_uni(ind);
66     end
67 end
68 Gext_s(end, :) = [];
69 Dngiu = reshape(DDs, length(DDs(:, 1)) * length(DDs(1, :)), 1);
70 dimensione1 = [length(NetData.A(:, 1)) + 12 * length(Ai(:,
        1)), length(NetData.A(1, :)) + 12 * length(Ai(1, :))];
71 B = zeros(dimensione1);
72 B(1:length(NetData.A(:, 1)), 1:length(NetData.A(1, :))) =
        NetData.A(1:length(NetData.A(:, 1)), 1:length(NetData.A(1,
        :)));
73 for l11 = 1:6
74     B(length(NetData.A(:, 1)) + 1 + (l11 - 1) * length(Ai(1,
        :)):length(NetData.A(:, 1)) + l11 * length(Ai(:, 1)),
        length(NetData.A(1, :)) + 1 + (l11 - 1) * length(Ai(1,
        :)):length(NetData.A(1, :)) + l11 * length(Ai(1, :))) =
        Ai;
75     B(vi(l11), length(NetData.A(1, :)) + 1 + (l11 - 1) *
        length(Ai(:, 1))) = -1;
76 end
77 NetData.A = B;
78 NetData.DD = [NetData.DD; Dngiu];
79 NetData.LL = [NetData.LL; 12 * ones(length(Dngiu), 1)];
80 NetData.nr = length(NetData.A(1, :));
81 NetData.nn = length(NetData.A(:, 1));

```

```

82 NetData.tt = [NetData.tt_mandata; zeros(length(Dngiu), 1)];
83 NetData.beta = [NetData.beta; 3 * ones(length(Dngiu), 1)];
84 NetData.zz = [NetData.zz; zeros(6 * length(Ai(1, :)), 1)];
85 NetData.Sezione = NetData.DD.^2 / 4 * pi;
86 Gngiu = (reshape(Gext_s, length(Gext_s(:, 1)) *
      length(Gext_s(1, :)), 1));
87 G_ext = [G_ext; Gngiu];
88 NetData.Sezione = NetData.DD.^2 / 4 * pi;
89 NetData.rr_g = ((NetData.ff .* NetData.LL ./ NetData.DD) +
      NetData.beta) ./ (2000 * NetData.Sezione.^2);
90 Pimposta = Pcc1;
91 Pimposta2 = Pcc3;
92 NodePBC = P1NODEPBC(2);
93 NodePBC2 = P1NODEPBC(1);
94 [P, G, Y, Err, res] = SIMPLE2(NetData, G_ext, NodePBC,
      NodePBC2, DatiSolver, Pimposta, Pimposta2);
95 xlswrite("PPPsu.xlsx", P);
96 xlswrite("GGGsu.xlsx", G);
97 PP = zeros(length(NetData.A(1, :)), 1);
98 PP(ri) = P(1:length(ri));
99 Pcc2 = PP(267);
100 GG = zeros(length(NetData.A(:, 1)), 1);
101 GG(X) = G(1:length(X));
102 Gcc2 = GG(593);
103 Gcc3 = GG(202);

```

Listato A.8: Risoluzione Semirete superiore per reti di grandi dimensioni

```

1 clear
2 clc
3 ci = [80, 81, 82, 83, 84, 85, 86, 87, 93, 94, 95, 96, 97, 98,
      99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110,
      111, 112, 113, 115, 116, 164, 165, 166, 167, 168, 169, 170,
      171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182];
4 load("NetData.mat")
5 NetData = DatiTrasporto;
6 DatiSolver = load("DatiSolver.mat");
7 G_ext = zeros(695, 1);
8 G_ext(NetData.bar(:, 2)) = 0.42 * NetData.bar(:, 3);
9 G_ext(NetData.PowerPlants.Nodes(2)) = -0.33 * 0.42 *
      sum(NetData.bar(:, 3));
10 NetData.ff = 0.007;
11 NetData.tt = NetData.tt_mandata;

```

```

12 NodePBC = NetData.PowerPlants.Nodes(1);
13 Pimposta = 2.5e5;
14 Ai = xlsread("Ai.xlsx", "B3:BF54");
15 db = xlsread("Ai.xlsx", "B60:E60");
16 Ai(:, db) = [];
17
18 %% calcolo diametri sottorete
19 DDs = zeros(52, 12);
20 Ai2 = [zeros(size(Ai)); zeros(1, 52)];
21 Ai2(end, 1) = 1;
22 Ai2(1:end-1, 1:end) = Ai;
23 n_ext = xlsread("Ai.xlsx", "B59:AB59");
24 Gext_s = zeros(53, 12);
25 Gs = zeros(52, 12);
26 for i = 1:12
27     if i <= 6
28         Gext_s(end, i) = -0.42 * abs(NetData.bar(i, 3));
29     else
30         Gext_s(end, i) = -0.42 * abs(NetData.bar(ci(i-6), 3));
31     end
32     n = length(n_ext);
33     vettore = linspace(0.02, 1, n);
34     vettore_normalizzato = (vettore / sum(vettore));
35     Gext_s(n_ext, i) = -(Gext_s(end, i) .*
36         vettore_normalizzato);
37     Gs(:, i) = Ai2 \ (-Gext_s(:, i));
38     rho_H2O = 990;    %[kg/m3]
39     vel_max = 1.5;   %[m/s]
40     diam_uni = 10^-3 * [0 0 50 75 100 125 250 300 350 400 450
41         500 600 800]';
42     DDs(:, i) = 2 * sqrt(abs(Gs(:, i)) / (pi * rho_H2O *
43         vel_max));    %[m]
44     for k = 1:length(DDs(:, i))
45         diff = diam_uni - DDs(k, i);
46         ind = find(diff > 0, 1); % controllare
47         DDs(k, i) = diam_uni(ind);
48     end
49 end
50 Gext_s(end, :) = [];
51 G_ext(NetData.bar(1:6, 2)) = 0;
52 G_ext(NetData.bar(ci(1:6), 2)) = 0;
53 Dngiu = reshape(DDs, length(DDs(:, 1)) * length(DDs(1, :)), 1);

```

```

51 dimensione1 = [length(NetData.A(:, 1)) + 12 * length(Ai(:,
    1)), length(NetData.A(1, :)) + 12 * length(Ai(1, :))];
52 B = zeros(dimensione1);
53 B(1:length(NetData.A(:, 1)), 1:length(NetData.A(1, :))) =
    NetData.A(1:length(NetData.A(:, 1)), 1:length(NetData.A(1,
    :)));
54 for l11 = 1:12
55     B(length(NetData.A(:, 1)) + 1 + (l11-1) * length(Ai(1,
        :)):length(NetData.A(:, 1)) + l11 * length(Ai(:, 1)),
        length(NetData.A(1, :)) + 1 + (l11-1) * length(Ai(1,
        :)):length(NetData.A(1, :)) + l11 * length(Ai(1, :))) =
        Ai;
56
57     if l11 <= 6
58         B(NetData.bar((l11), 2), length(NetData.A(1, :)) + 1 +
            (l11-1) * length(Ai(:, 1))) = -1;
59     else
60         B(NetData.bar(ci(l11-6), 2), length(NetData.A(1, :)) +
            1 + (l11-1) * length(Ai(:, 1))) = -1;
61     end
62 end
63 NetData.A = B;
64 NetData.DD = [NetData.DD; Dngiu];
65 NetData.LL = [NetData.LL; 12 .* ones(length(Dngiu), 1)];
66 NetData.nr = length(NetData.A(1, :));
67 NetData.nn = length(NetData.A(:, 1));
68 NetData.tt = [NetData.tt_mandata; zeros(length(Dngiu), 1)];
69 NetData.beta = [NetData.beta; 3 * ones(length(Dngiu), 1)];
70 NetData.zz = [NetData.zz; zeros(12 * length(Ai(1, :)), 1)];
71 NetData.Sezione = NetData.DD.^2 / 4 * pi;
72 Gngiu = (reshape(Gext_s, length(Gext_s(:, 1)) *
    length(Gext_s(1, :)), 1)) *
    length(Gext_s(1, :)), 1));
73 G_ext = [G_ext; Gngiu];
74 NetData.rr_g = ((NetData.ff .* NetData.LL ./ NetData.DD) +
    NetData.beta) ./ (2000 .* NetData.Sezione.^2);
75 tic
76 [P, G, Y, Err, res] = SIMPLE(NetData, G_ext, NodePBC,
    DatiSolver, Pimposta);
77 toc

```

Listato A.9: Risoluzione rete di grandi dimensioni

Bibliografia

- [1] Francesco Bottacin. *Note del Corso di Teoria dei Grafi*. Francesco Bottacin, 2008.
- [2] W. Malalasekera H. Versteeg. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method (2nd Edition)*. Prentice Hall, 2007. ISBN: 9780131274983.
- [3] Henrik Lund et al. «Future District Heating Systems and Technologies: On the role of Smart Energy Systems and 4th Generation District Heating». In: *Energy* (2018).
- [4] Thomas Lickleder Saltanat Kuntuarova. «Design and simulation of district heating networks: A review of modelling approaches and tools». In: *Energy* (2024).
- [5] Stierhoff K.L. Taylor N.R. *District heating Handbook*. International District Heating Association Washington, D.C. ISBN: 0961083808.
- [6] Vittorio Verda. *SIMPLE and linear systems*. Presentazione PowerPoint, Numerical Design of Thermal Systems. 2022.
- [7] Robin Wiltshire. *Advanced district heating and cooling (DHC) systems*. Elsevier : WoodHead Publishing, 2016. ISBN: 9781782423959.
- [8] John M.Cimbala Yunus A. Çengel. *Meccanica dei fluidi*. 2007.