

## Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
<b>2</b>	<b>File di input.</b>	<b>4</b>
2.1	Il file <code>boundary.gridname.</code> . . . . .	4
2.2	Il file <code>domain.gridname</code> . . . . .	18
<b>3</b>	<b>Informazioni intermedie</b>	<b>21</b>
3.1	File intermedi . . . . .	21
3.2	Esecuzione di <i>HMesh<sup>2D</sup></i> . . . . .	25
<b>4</b>	<b>File di Output</b>	<b>26</b>

# 1 Introduzione

$HMesh^{2D}$  é un generatore di griglie ibride bidimensionali costituite di triangoli e quadrilateri ed é stato realizzato per condurre simulazioni numeriche ad alti numeri di Reynolds che richiedono griglie adatte al calcolo di uno strato limite sottile (fig. 1).

La sua caratteristica principale é la separazione delle porzioni di dominio in cui vengono collocati gli uni o gli altri elementi: nelle regioni di strato limite sono generalmente preferiti i quadrilateri allungati, mentre altrove sono maggiormente indicati i triangoli di forma equilatera .

La costruzione della griglia procede in due fasi consecutive:

- generazione della griglia di strato limite, con l'introduzione di elementi misti attraverso un metodo a fronte avanzante,
- copertura della restante parte di dominio con una triangolazione di Delaunay attraverso l'algoritmo di Bowyer-Watson.

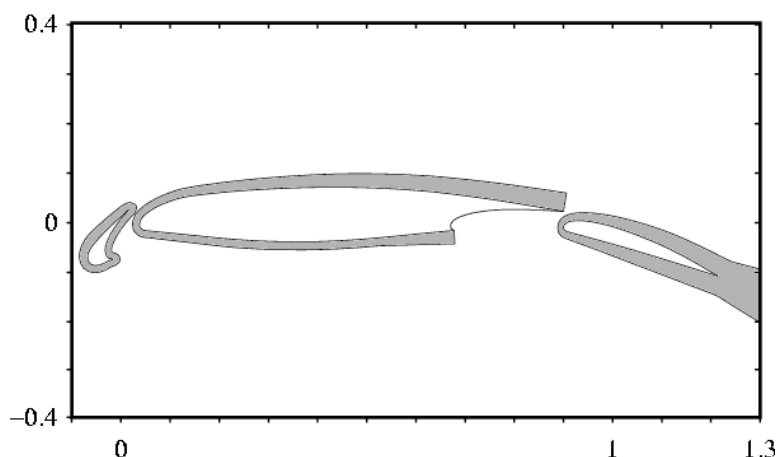


Figura 1: Suddivisione del dominio

Nella regione di strato limite vengono introdotti in prevalenza quadrilateri, mentre l'uso dei triangoli é funzionale al corretto avanzamento del fronte. Un esempio delle capacità del generatore  $HMesh^{2D}$  é riportato in fig. 2; in relazione a tale esempio verranno descritti nei paragrafi successivi i dati richiesti per la generazione della griglia.

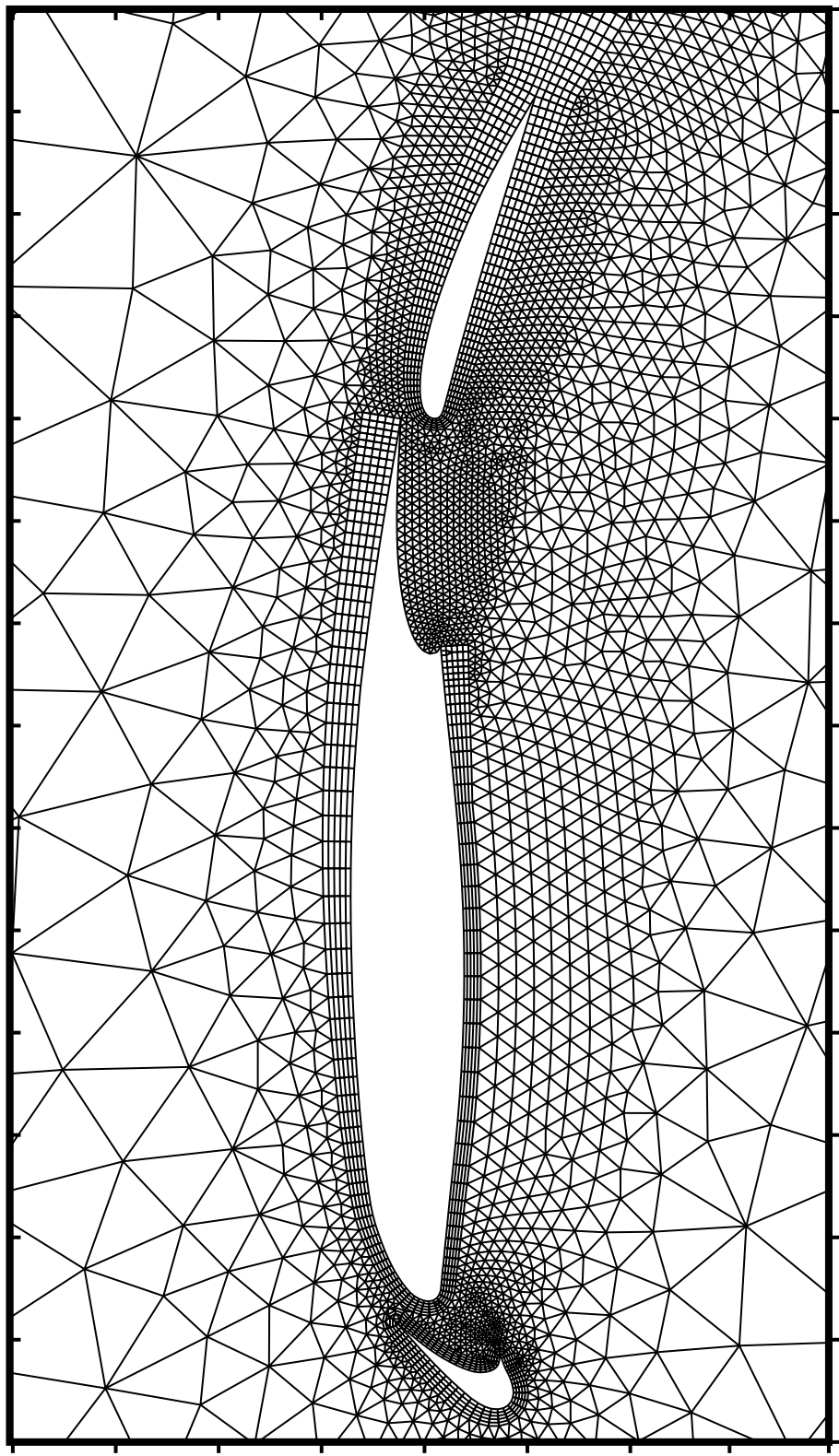


Figura 2: Profilo ipersostentato con griglia di strato limite e adattamento di scia

File di input	File intermedi	File di output
boundary.gridname domain.gridname	knots.gridname geometry.gridname quadplot splot	nodes.gridname grid.gridname backplot.gridname gridplot.gridname

## 2 File di input.

I dati necessari alla generazione della griglia sono contenuti all'interno di due file che devono essere preparati prima di eseguire *HMesh*<sup>2D</sup>:

- boundary.gridname
- domain.gridname

dove l'estensione **gridname** deve essere identica e viene mantenuta per tutti i file di output relativi alla griglia generata. Quando i due file precedenti siano completi di tutte le informazioni necessarie, *HMesh*<sup>2D</sup> viene avviato dal comando `./hmesh2d.exe` che richiede di immettere la sola estensione **gridname** dei file.

### 2.1 Il file boundary.gridname.

Nelle pagine 15-17 viene riportato nella sua completezza il file **boundary.iper** relativo all'esempio illustrato, mentre nel seguito viene descritta in dettaglio ognuna delle sue parti: il file é composto da due righe iniziali che contengono alcuni dati generali e da blocchi consecutivi di informazioni che hanno lo scopo di definire i contorni del dominio all'interno del quale é richiesta la costruzione della griglia.

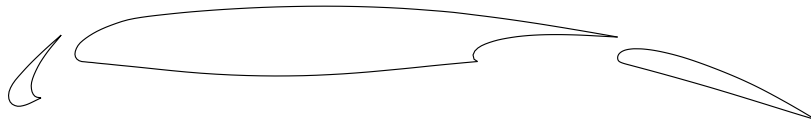


Figura 3: Profilo alare ipersostentato

I contorni sono rappresentati da linee chiuse, come visibile in figura 3, dove non é riportata la circonferenza che identifica il contorno esterno; per

comodit di rappresentazione, possibile suddividere ciascuna linea in pi parti consecutive che, nel seguito, prendono il nome di *lati* e sono identificate da un indice intero progressivo ( $\square$  in fig. 4).

Ogni lato possiede un punto iniziale ed un punto finale (due *vertici*), che vengono a loro volta contrassegnati da un indice intero progressivo ( $\circ$  in fig. 4): se i due punti sono coincidenti, il lato é *chiuso* su s stesso ed il vertice é unico.

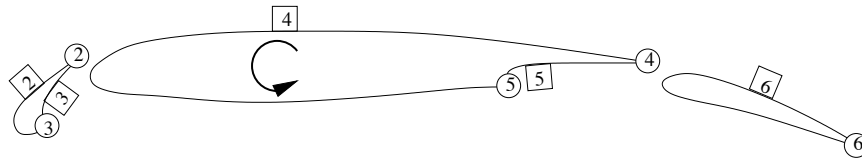


Figura 4: Topologia dei contorni

Nelle due righe iniziali del file `boundary` si trova pertanto `NVert`, che rappresenta il numero complessivo di vertici riconoscibile nel numero `NEdge` di lati che descrivono il contorno (fig. 4); per la trattazione dei dati rimanenti si rimanda ai paragrafi successivi.

NVert	NEdge	Nlayer	Height_block	Condens_angle	Insert_mtd
6	6	5	30	40	0

---

Ciascun lato viene descritto da un blocco di informazioni che si compone di due parti:

- la prima descrive la geometria del lato,
- la seconda stabilisce il numero e la collocazione dei punti necessari ad imporre le condizioni al contorno del problema differenziale.

### Geometria del contorno.

La forma del lato può essere interpolata a partire da un insieme di punti dei quali siano disponibili le coordinate; in tal caso l'opzione `data` serve a specificare che i punti si trovano, ad esempio, nel file `slat1` della directory corrente, mentre `translation` e `rotation` precedono rispettivamente i valori di traslazione e di rotazione in blocco di tali punti.

```

BEGIN (SLAT LEFT)
#
data
slat1
    translation (Dx Dy)
    -0.009 0.0
    rotation (degrees)
    -7.0
#

```

Il file `slat1` contiene nella prima riga il numero totale di punti contenuti nel file e per ciascuna delle righe successive le coordinate cartesiane di un punto.

```

36
2.7025000000000000E-02  1.8988000000000001E-02
1.9713999999999999E-02  1.3891000000000001E-02
1.3998000000000000E-02  1.0012000000000000E-02
.....

```

I punti devono essere elencati in modo tale che attraverso la loro successione si percorra il lato lasciando a destra il dominio interessato dalla griglia (fig. 4); nell'esempio si percorre la parte anteriore dello slat a partire del bordo di uscita superiore.

Se non é disponibile un file di punti, é possibile in alternativa descrivere il lato secondo tre forme elementari: il segmento di retta, l'arco di cerchio e l'arco di ellisse.

```

BEGIN (CIRCLE OUT)
#
circle
    101 1.0 0.0 3.0 360.0 0.0
#

```

Le tre opportunità sono rispettivamente identificate dall'opzione `line`, `circle`, `ellipse` alla quale segue una riga di parametri:

- `line`  
 numero di punti, coordinate del punto iniziale, coordinate del punto finale
- `circle`  
 numero di punti, coordinate del centro, raggio, angolo iniziale, angolo finale (in gradi)

- ellipse  
numero di punti, coordinate del centro, lunghezza dei semiassi  $x$  e  $y$ ,  
angolo iniziale, angolo finale (in gradi)

Anche in questo caso il verso di percorrenza del lato fornisce un'indicazione implicita sulla collocazione del dominio di interesse per la griglia.

Il numero di punti indicati per descrivere il lato deve essere sufficientemente elevato; in caso contrario può essere difficoltoso riuscire ad imporre efficacemente le successive condizioni di metrica.

### Metrica sul contorno.

La seconda parte di informazioni relative al lato ha lo scopo di definirne la topologia e la metrica; inoltre contiene alcune indicazioni necessarie alla prima fase di generazione della griglia mista.

```
BEGIN (FLAP)
:
:
:
Edge BegVert EndVert N | Adv_front? Delta Ref.Leng Propag.Law Par
6      6         6      5 | 1          0.1      2         2         1.5
First layer elements: | Wake options: Wake final elements:
Sdata Hdata Idata Height | Type Length Angle Length Height
0.00 0.009 1 0.007 1 2 3 0.1 0.07
0.25 0.01 1 0.005 0 0 0 0 0
0.51 0.002 1 0.002 0 0 0 0 0
0.75 0.01 1 0.005 0 0 0 0 0
1.00 0.009 1 0.007 0 0 0 0 0
END
```

---

Sotto la prima riga evidenziata sono disposti alcuni dati riguardanti la topologia del lato ed infine i parametri essenziali per governare l'avanzamento del fronte che ricopre con la griglia la regione di strato limite:

Edge	numero del lato
BegVert	numero del vertice iniziale
EndVert	numero del vertice finale
N	numero dei punti di imposizione della metrica ( $\geq 2$ )
Adv-Front?	fronte avanzante dal lato (0 = no, 1 = sì)
Delta	parametro per la variazione delle distanze di inserimento dei punti
Ref.Leng.	base per il calcolo della lunghezza di riferimento (1, 2, 3)
Propag.Law	legge di variazione della lunghezza(1, 2, 3)
Par.	parametro per la variazione della lunghezza

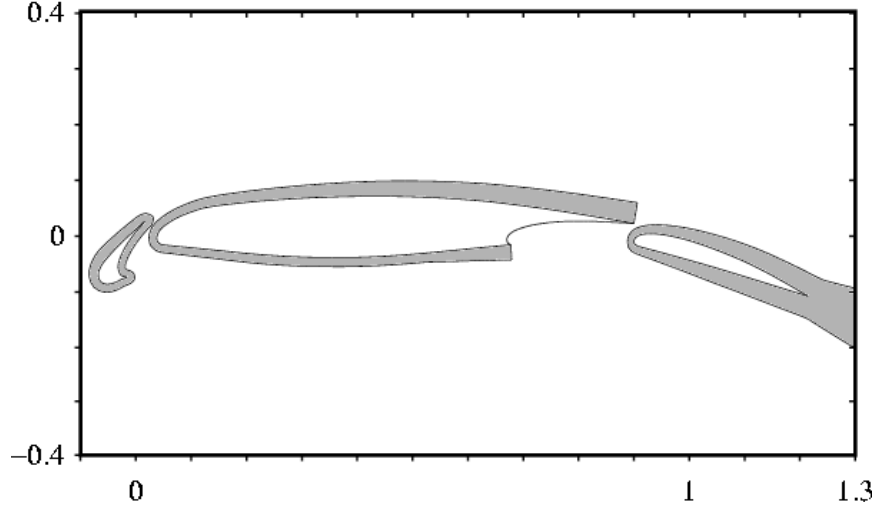


Figura 5: Regioni di interesse per la griglia di quadrilateri

Se **Adv-Front?** assume valore unitario il lato viene duplicato e proiettato verso l'interno del dominio in modo iterativo; il numero di iterazioni ha un valore unico assegnato dalla variabile **Nlayer** nella seconda riga del file **boundary** ed equivale al numero di strati di elementi da sovrapporre al lato.

I lati consecutivi (che hanno un vertice in comune) sui quali é richiesto l'avanzamento del fronte vengono riuniti in un solo fronte, che può essere chiuso o aperto. Nel primo caso i vertici del fronte coincidono, mentre nel secondo sono distinti.

La distanza di proiezione viene governata dalla legge:

$$h_i = h_i(1 + \delta)^i \quad i = 1, N_{layer}$$

che dipende dal parametro **delta**.

La direzione di proiezione é strettamente ortogonale al contorno ovvero al passo precedente del fronte; per consentire la deviazione dalla direzione normale é sufficiente impostare il parametro **insert\_mtd** pari ad 1 nella seconda riga del file (altrimenti il valore é pari a 0). La deviazione viene governata dalla differenza di lunghezza tra quadrilateri adiacenti.

Il parametro **N** indica il numero di condizioni necessarie a definire completamente le dimensioni del primo strato di elementi costruiti dal fronte avanzante sul lato. Le condizioni di metrica vengono imposte a diversi valori di ascissa curvilinea definita sul lato includendo in ogni caso gli estremi 0.0 e 1.0 (quindi  $N \geq 2$ ).



<b>Sdata</b>	ascissa curvilinea normalizzata
<b>Hdata</b>	base dei quadrilateri o lato dei triangoli equilateri
<b>Idata</b>	tipo di interpolazione dei dati di metrica
<b>Height</b>	altezza dei quadrilateri

Se non é prevista la costruzione della griglia di strato limite sul lato in questione il parametro **Hdata** identifica il lato dei triangoli equilateri da collocare in prossimità del valore di **Sdata** e **Height** non viene letto (con esso anche i successivi parametri della riga).

Introducendo invece l'avanzamento del fronte, **Height** rappresenta l'altezza del primo strato di quadrilateri inseriti (che varia con **Delta** negli strati successivi).

I dati assegnati ai diversi valori di **Sdata** vengono interpolati nelle ascisse intermedie allo scopo di definire con continuità la metrica del lato; sono disponibili tre modalità di interpolazione: lineare, sinusoidale e geometrica (corrispondenti a **Idata** = 1, 2, 3).

Se ad una qualsiasi delle ascisse **Sdata** intermedie tra 0 e 1 non si desidera imporre un valore a **Height** é sufficiente assegnarvi un valore negativo (le condizioni su **Hdata** sono, nelle griglie di strato limite, più numerose e stringenti rispetto a **Height**).

*HMesh*<sup>2D</sup> distribuisce sulla geometria del lato definito nella prima fase un numero di punti adeguato a soddisfare le condizioni assegnate da **Hdata**; a questo scopo, dopo aver effettuato l'interpolazione, ne esegue il campionamento sui punti che descrivono la geometria (i punti contenuti nel file **slat1** oppure il numero di punti impiegati per definire le semplici geometrie **line**, **circle**, **ellipse**).

É importante garantire un numero adeguato di punti nella descrizione della geometria poichè in caso contrario molte delle informazioni di metrica imposte possono essere smarrite nel processo di interpolazione e campionamento (fig. 6).

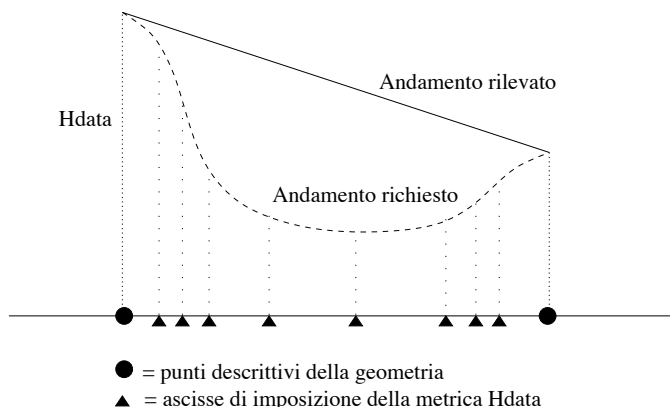


Figura 6: Visibilità delle richieste di metrica

### Adattamento in presenza di scie.

Le ascisse **Sdata** sono impiegate per l'introduzione di una regione di griglia di "strato limite" all'interno di una eventuale scia (**Type** pari a 1); a questo scopo si esegue una estensione del contorno con area nulla e lunghezza assegnata **Length** (fig. 2.1).

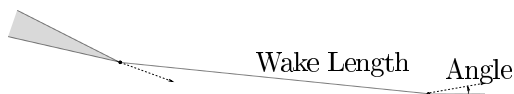


Figura 7: Estensione del contorno per la presenza della scia

I dati necessari alla definizione della griglia di scia sono rilevati sul contorno di partenza e sull'estremità libera, mentre nelle regioni intermedie vengono interpolati linearmente.

La direzione della "linea" che supporta la griglia di scia è inizialmente ortogonale al contorno, mentre nella parte finale viene specificata da **Angle** (in gradi, positivo antiorario rispetto all'asse x).

Gli elementi del primo strato inserito hanno dimensioni note in corrispondenza del contorno e vengono specificate solamente nell'estremo libero (**Length** e **Height**, lunghezza ed altezza degli ultimi quadrilateri). Eseguite queste operazioni preliminari, *HMesh<sup>2D</sup>* prosegue come per tutti gli altri contorni (fig. 2.1).

Specificando **Type**= 2 è possibile utilizzare la stessa procedura modificata per l'impiego ottimale su contorni non spigolosi (ad esempio, nella ottimizzazione della griglia presso il punto di ristagno di un profilo alare).

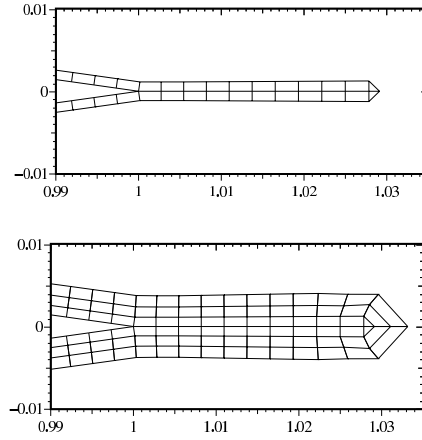


Figura 8: Generazione della griglia di scia

### L'uso dei triangoli nella prima fase di generazione.

$HMesh^{2D}$  dispone attraverso l'avanzamento del fronte successivi strati di punti ovvero di quadrilateri; in presenza di contorni convessi o concavi la disposizione dei punti può provocare rispettivamente la dispersione o la concentrazione, che viene risolta attraverso l'impiego limitato di triangoli al posto dei quadrilateri.

Nel caso di un contorno convesso l'introduzione dei triangoli avviene quando la base di un quadrilatero supera una lunghezza massima di riferimento (fig. 9).

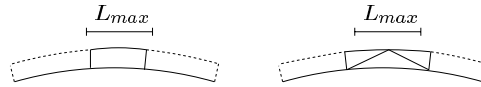


Figura 9: Triangoli per contorni convessi

Tale lunghezza viene definita per ogni quadrilatero del primo strato e viene modificata nei successivi passi di avanzamento del fronte. Il parametro `Ref.Leng` seleziona la modalità di calcolo della lunghezza per il primo strato di quadrilateri (non definitiva) come:

- 1 base del quadrilatero
- 2 base media di tre quadrilateri adiacenti
- 3 base media di tutti i quadrilateri del lato di appartenenza

Il parametro **Propag.Law** permette di scegliere la legge di variazione della lunghezza, che viene applicata a tutti gli strati di elementi introdotti (compreso il primo, fornendo la lunghezza definitiva) e prevede le seguenti opzioni governate dal parametro fisso **Par**:

- 1  $L_{max} = \text{Ref.Leng} * \text{Par}$
- 2  $L_{max} = \text{Ref.Leng} * (1 + \text{Par})^i$ ,  $i = 1, \text{Nlayer}$
- 3  $L_{max} = \text{Ref.Leng} * [1 + (\text{Par} - 1) * (\frac{i}{\text{Nlayer}})]$   $i = 1, \text{Nlayer}$

L'opzione 3 corrisponde all'assegnazione di una lunghezza massima unica per tutti i quadrilateri dell'ultimo fronte, che viene interpolata linearmente negli strati intermedi.

Sono previsti interventi correttivi in presenza di contorni spigolosi attraverso l'inserimento nel primo strato di elementi di un triangolo (per spigoli il cui angolo sia compreso tra  $100^\circ$  e  $60^\circ$ ) o due triangoli affiancati (per spigoli compresi tra  $60^\circ$  e  $0^\circ$ ); negli strati successivi l'inserimento degli elementi procede nel modo normale (fig. 10).

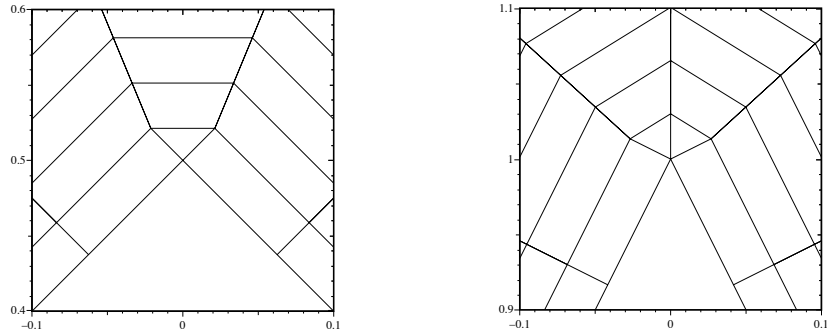


Figura 10: biforcazioni e triforcazioni

In presenza di contorni concavi, l'uso dei triangoli viene gestito in modo unitario, senza distinguere tra i diversi lati del contorno, da un parametro collocato nella seconda riga del file (**Condense\_angle**, affiancato da **Height\_block**).

I contorni concavi provocano la concentrazione dei punti del fronte e l'uso dei triangoli viene interpretato come strumento per limitarne gli effetti. Un quadrilatero in corrispondenza del quale la concentrazione dei punti risulti eccessiva viene trasformato in un triangolo annullando la lunghezza del sul lato "superiore" (*condensazione* dei due punti di estremità).

Il triangolo risultante é caratterizzato dall'angolo di vertice "superiore"; **Condense\_angle** ne rappresenta il limite inferiore per tutti i triangoli di

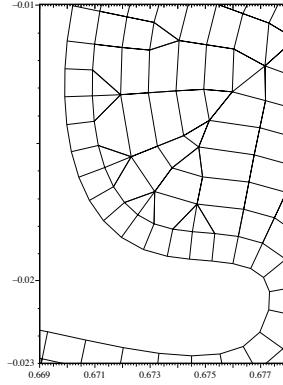


Figura 11: Condensazioni su due e tre punti

condensazione e la sua variazione permette di controllare sia la densità dei punti distribuiti che la qualità dei triangoli correttivi.

Poichè la velocità con la quale i punti del fronte vengono concentrati è proporzionale alla concavità del contorno, il valore di **Condense\_angle** deve crescere di conseguenza; in caso contrario il fronte può essere soggetto a comportamenti non adeguati (soprattutto per forti concavità e valori di **delta** elevati) ed interrompere il proprio funzionamento.

Infine, la condensazione può interessare uno o al massimo due quadrilateri adiacenti; pertanto i diversi interventi correttivi vengono comunque separati da un quadrilatero che mantiene la propria identità (fig. 11).

L'assegnazione di **Condense\_angle** equivale a limitare le dimensioni caratteristiche dei quadrilateri collocati da un fronte concavo; la seguente tabella fornisce l'equivalenza tra l'allungamento (rapporto tra base ed altezza) di un quadrilatero ed il corrispondente valore limite per l'angolo di vertice del triangolo di condensazione.

Allungamento quadrilatero	Angolo di vertice del quadrilatero condensato
0.175	10°
0.352	20°
0.535	30°
0.728	40°
0.932	50°
1.154	60°
1.400	70°
1.678	80°
2.000	90°
2.383	100°
2.856	110°

L'angolo `Height_block` segue lo stesso principio, ma viene impiegato per limitare inferiormente l'allungamento dei quadrilateri conseguente alla variazione delle altezze di inserimento; quando un quadrilatero nello strato corrente ha un allungamento troppo ridotto (ovvero troppo allungato nella direzione di avanzamento del fronte), l'altezza di inserimento dell'intero strato viene bloccata e non sono più consentiti gli aggiornamenti negli strati successivi. Questo parametro consente di ottenere, per valori prossimi a 60°, un passaggio graduale nelle dimensioni degli elementi dalla griglia mista alla triangolazione (da quadrati a triangoli equilateri); al contrario, per valori tendenti a 0° non viene posto alcun vincolo.

NVert	NEdge	Nlayer	Height_block	Condens_angle	Insert_mtd
6	6	5	30	40	0

---

```

BEGIN (CIRCLE OUT)
#
circle
    101 1.0 0.0 3.0 360.0 0.0
#
Edge BegVert EndVert N | Adv_front? Delta Ref.Leng Propag.Law Par
1 1 1 2 | 0 0 0 0 0
First layer elements: | Wake options: Wake final elements:
Sdata Hdata Idata Height | Type Length Angle Length Height
0.0 0.5 1 0 | 0 0 0 0 0
1.0 0.5 1 0 | 0 0 0 0 0
END

```

---

```

BEGIN (SLAT LEFT)
#
data
slat1
    translation
        -0.009 0.0
    rotation
        -7
#
Edge BegVert EndVert N | Adv_front? Delta Ref.Leng Propag.Law Par
2 2 3 4 | 1 0.0 2 1 2.2
First layer elements: | Wake options: Wake final elements:
Sdata Hdata Idata Height | Type Length Angle Length Height
0.00 0.005 1 0.003 | 0 0 0 0 0
0.45 0.009 1 0.004 | 0 0 0 0 0
0.73 0.005 1 0.003 | 0 0 0 0 0
1.00 0.003 1 0.0025 | 0 0 0 0 0
END

```

---

```

BEGIN (SLAT RIGHT)
#
data
slat2
    translation
        -0.009 0.0
    rotation
        -7
#
Edge BegVert EndVert N | Adv_front? Delta Ref.Leng Propag.Law Par
2 3 2 3 | 1 0.0 2 1 2.2
First layer elements: | Wake options: Wake final elements:
Sdata Hdata Idata Height | Type Length Angle Length Height
0.00 0.003 1 0.0025 | 0 0 0 0 0

```

```

0.48  0.008  1    0.0025  0    0          0    0    0
1.00  0.004  1    0.0025  0    0          0    0    0
END
-----
BEGIN (MAIN LONG)
#
data
mainlong
  translation
    0.0 0.0
  rotation
    0
#
Edge  BegVert  EndVert  N  |  Adv_front?  Delta  Ref.Leng  Propag.Law  Par
3     4         5         6  |  1           0.1    2         2         1.2
First layer elements:  |  Wake options:  Wake final elements:
Sdata  Hdata  Idata  Height  |  Type  Length  Angle  Length  Height
0.00   0.006  1    0.007  0    0          0    0    0
0.30   0.028  1    0.005  0    0          0    0    0
0.585  0.004  1    0.0025  0    0          0    0    0
0.70   0.025  1    0.003  0    0          0    0    0
0.90   0.02   1    0.003  0    0          0    0    0
1.00   0.004  1    0.005  0    0          0    0    0
END
-----
BEGIN (MAIN SHORT)
#
data
mainshort
  translation
    0.0 0.0
  rotation
    0
#
Edge  BegVert  EndVert  N  |  Adv_front?  Delta  Ref.Leng  Propag.Law  Par
4     5         4         5  |  0           0    0         0         0
First layer elements:  |  Wake options:  Wake final elements:
Sdata  Hdata  Idata  Height  |  Type  Length  Angle  Length  Height
0.00   0.007  1    0      0    0          0    0    0
0.30   0.007  1    0      0    0          0    0    0
0.50   0.01   1    0      0    0          0    0    0
0.70   0.008  1    0      0    0          0    0    0
1.00   0.007  1    0      0    0          0    0    0
END
-----
BEGIN (FLAP)
#
data
flap

```



```

translation
0.0 0.0
rotation
0
#
Edge BegVert EndVert N | Adv_front? Delta Ref.Leng Propag.Law Par
6 6 6 5 | 1 0.1 2 2 1.5
First layer elements: | Wake options: Wake final elements:
Sdata Hdata Idata Height | Type Length Angle Length Height
0.00 0.009 1 0.007 | 1 2 3 0.1 0.07
0.25 0.01 1 0.005 | 0 0 0 0 0
0.51 0.002 1 0.002 | 0 0 0 0 0
0.75 0.01 1 0.005 | 0 0 0 0 0
1.00 0.009 1 0.007 | 0 0 0 0 0
END
-----

```

Il file boundary.iper

## 2.2 Il file `domain.gridname`

La maggior parte delle informazioni necessarie al funzionamento della seconda parte di generazione della griglia é contenuta nel file descritto in precedenza.

Il triangolatore rileva la spaziatura dei punti distribuiti sui lati, siano essi appartenenti a contorni veri o a contorni fittizi (coincidenti con l'ultimo passo di avanzamento del fronte, fig. 12); la distanza tra i punti adiacenti fornisce una misura per la dimensione caratteristica dei triangoli da costruire nella rimanente parte del dominio.

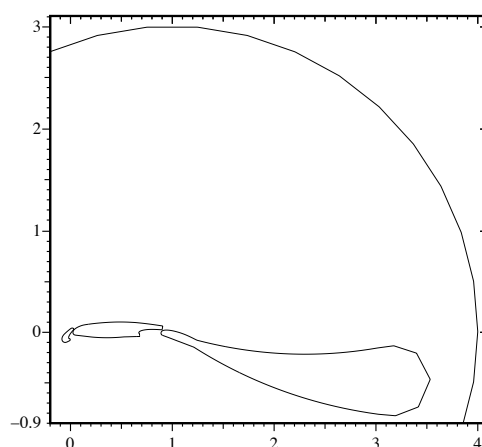


Figura 12: Dominio per la triangolazione

La taglia dei triangoli viene quindi rilevata sul contorno e diffusa all'interno del dominio lungo le linee della griglia contenuta nel file di output `backplot.iper`; questa griglia (fig. 13) é ottenuta triangolando l'insieme dei soli punti di contorno con l'usuale algoritmo di Bowyer-Watson.

Il triangolatore adotta una procedura a fronte avanzante che prevede l'inserimento iterativo di insiemi di punti all'interno della backgrid; la loro posizione é determinata in modo tale che il conseguente aggiornamento della backgrid produca triangoli conformi alle indicazioni di metrica rilevate. Il numero massimo di iterazioni é specificato dal parametro `Delaunay Steps` (di valore in genere elevato) ma se la triangolazione é soddisfacente il numero di iterazioni effettivamente eseguite può essere inferiore. Al loro termine é possibile attivare una procedura di regolarizzazione della griglia (`Laplacian Smoothing=1`).

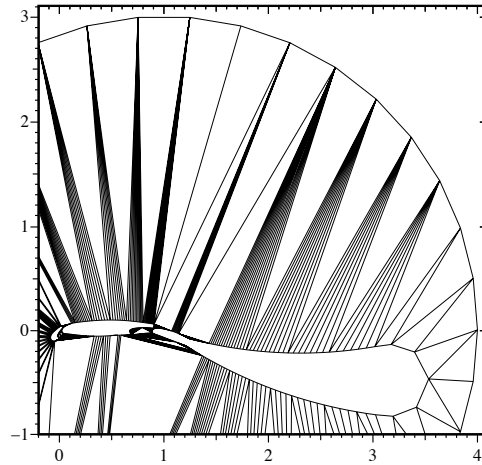


Figura 13: Backgrid

```

Delaunay Steps      Laplacian Smoothing(1/0)
1000                1
#
Backgrid points:    N
                   1
                   X      Y      Hdata
                   0.03   -0.06  0.02
#
Steiner points:     N
                   0
                   X      Y
#

```

La forma della triangolazione può essere condizionata principalmente attraverso il controllo della spaziatura dei punti sul contorno; tuttavia alcune parti del contorno che interessano il triangolatore sono fittizie e su di esse non é possibile agire direttamente.

L'uso dei 'punti di backgrid' permette di modificare la forma della backgrid e la diffusione delle informazioni di metrica che avviene lungo i suoi segmenti. Il parametro **N** quantifica i punti di backgrid che devono essere elencati in colonna completi delle coordinate **X**, **Y** e della dimensione **Hdata** da assegnare localmente ai triangoli.

Le figure 14 e 15 confrontano la griglia dell'esempio rispettivamente senza l'uso e con l'uso del punto di backgrid specificato nel file **domain.ip**.

I punti di backgrid hanno il duplice effetto di attrarre su di sé una parte

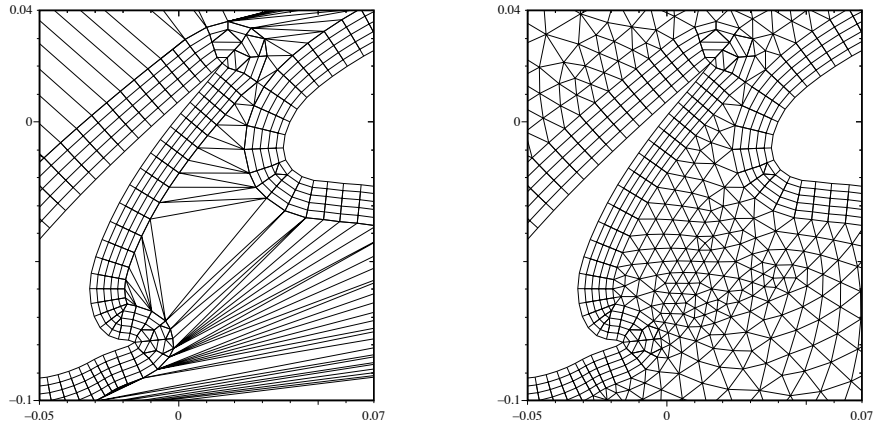


Figura 14: Griglia di fondo e finale senza punto di backgrid

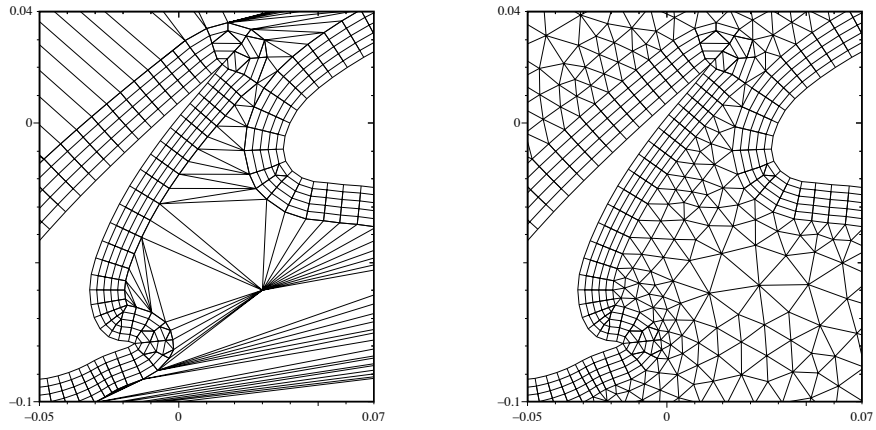


Figura 15: Griglia di fondo e finale con il punto di backgrid

dei segmenti della griglia di fondo che partono dal contorno e di vincolare direttamente l'interpolazione della metrica su di essi.

I punti di Steiner non richiedono invece l'assegnazione di una metrica (che rilevano dall'interpolazione già esistente) e limitano la propria influenza al primo effetto.

## 3 Informazioni intermedie

### 3.1 File intermedi

**Il file `knots.gridname`:** questo file contiene i punti attraverso i quali é stata descritta la geometria dei contorni, suddividisi per lati di appartenenza e specificando in quale modo essi sono stati ottenuti.

<code>n_curv</code>	numero di lati
<code>dim</code>	numero di dimensioni del lato(2)
<code>points</code>	numero di punti del lato
<code>idc</code>	0 per <code>data</code> e <code>line</code> , 2 per <code>circle</code> ed <code>ellipse</code>
<code>x, y</code>	coordinate dei punti

**Il file `geometry.gridname`:** Contiene per ciascun lato i dati impiegati per il calcolo delle curve interpolanti.

<code>n_curv</code>	numero di lati
<code>nd</code>	numero di dimensioni del lato (2)
<code>ns</code>	numero di punti del lato
<code>s</code>	ascissa curvilinea del punto
<code>x1, x2</code>	coordinate <code>x</code> e <code>y</code> del punto
<code>xs1, xs2</code>	derivate delle coordinate <code>x</code> ed <code>y</code> rispetto ad <code>s</code>

**I file `splot` e `quadplot`:** questi file, come `backplot` e `gridplot`, consentono la visualizzazione della griglia. I dati contenuti al loro interno rappresentano l'insieme dei segmenti di griglia e sono riportati nel formato PLOTMTV.

Il file `splot` rappresenta i contorni del dominio con la relativa distribuzione di punti; per ognuno di essi viene riportato un indice che ne esprime, in millesimi, l'ascissa curvilinea (normalizzata a 1) sul lato di appartenenza. Questa visualizzazione facilita la ricerca delle ascisse curvilinee `Sdata` alle quali imporre le condizioni di metrica.

Il file `quadplot` viene completato al termine della prima fase di generazione della griglia e contiene i contorni del dominio e la griglia priva della successiva triangolazione.

```

Problem name ?
iper
~~~~~ Boundary description ~~~~~
Loading boundary data:
Boundary loaded
Boundary grid...

Segment 1 points = 39
Segment 2 points = 36
Segment 3 points = 24
Segment 4 points = 115
Segment 5 points = 34
Segment 6 points = 88

~~~~~ BOUNDARY LAYER GRID ~~~~~

Initial front description
edge 1 : begins at vertex 1 ,ends at vertex 1
edge 2 : begins at vertex 2 ,ends at vertex 3
edge 3 : begins at vertex 3 ,ends at vertex 2
edge 4 : begins at vertex 4 ,ends at vertex 5
edge 5 : begins at vertex 5 ,ends at vertex 4
edge 6 : begins at vertex 6 ,ends at vertex 6

Closed & structured boundary 1 is made of 2 edges
Open & structured boundary 2 is made of 1 edges
Closed & structured boundary 3 is made of 1 edges
which_e 2 3 4 6 0 0
h_many_edg 2 1 1 0 0 0
.....
Front advancing: step n.
--- 1
sharp edge! 2 triangles added, p = 1.5580000000000000E-02 2.1040000000000000E-02
sharp edge! 2 triangles added, p = -1.5759853193284514E-02 -7.9113864450005997E-02
wake added
--- 2
--- 3
--- 4
--- 5
Structured mesh done!
.....
Interface with Delaunay Triangulator:
New boundaries:
edge 1 type: Delaunay
edge 2 type: b.layer
edge 4 type: b.layer
edge 5 type: Delaunay
edge 6 type: b.layer

```

Open boundaries corrections:

edge 3:

beginning side: points added to b.layer front

final side: points added to b.layer front

Structured grid plotted on file quadplot

~~~~~ DELAUNAY TRIANGULATION ~~~~~

Background grid...

b pnt number = 478 478

b spx number = 488 488

d pnt number = 466 466

d spx number = 931 931

Backgrid points...

Advancing Front :

| STEP | NFRO | NADD | MAX_LENGTH |
|------|------|------|------------|
| 1    | 456  | 447  | 8.89       |
| 2    | 432  | 415  | 6.45       |
| 3    | 412  | 392  | 4.66       |
| 4    | 385  | 357  | 3.34       |
| 5    | 368  | 308  | 2.36       |
| 6    | 320  | 273  | 1.67       |
| 7    | 280  | 246  | 1.14       |
| 8    | 248  | 224  | 0.79       |
| 9    | 236  | 198  | 0.51       |
| 10   | 205  | 175  | 0.41       |
| 11   | 169  | 132  | 0.26       |
| 12   | 122  | 88   | 0.17       |
| 13   | 92   | 72   | 0.09       |
| 14   | 71   | 55   | 0.05       |
| 15   | 53   | 37   | 0.02       |
| 16   | 38   | 30   | 0.01       |
| 17   | 28   | 18   | 0.01       |
| 18   | 20   | 11   | 0.00       |
| 19   | 6    | 2    | 0.00       |
| 20   | 0    | 0    | 0.00       |

Front end.

Corners splitted

Triangulation Laplacian Smoothing

1 0.21540E+00

2 0.30907E-01

3 0.11317E-01

4 0.63629E-02

5 0.43913E-02

6 0.32918E-02

7 0.25879E-02

```

      8 0.20968E-02
      9 0.17379E-02
     10 0.14660E-02
~~~~~ GRIDS INTERFACE & SAVING ~~~~~
Building triangulation arrays...
Building b.layer arrays...
Coupling arrays...
Interface done!
Grid saved! (nodes and grid)
~~~~~ PLOT FILES ~~~~~
Splot      done
Quadplot   done
Backplot   done
Gridplot   done
~~~~~

```



### 3.2 Esecuzione di *HMesh*<sup>2D</sup>

L'esecuzione del programma produce sullo schermo una serie di informazioni (pagg. 22-24) raggruppate secondo le diverse fasi di generazione della griglia:

1. descrizione dei contorni:
  - caricamento dei dati contenuti nel file `boundary.gridname`
  - distribuzione dei punti sulla geometria secondo le indicazioni di metrica (viene riportato il numero finale di punti per ciascun lato)
2. generazione griglia di strato limite:
  - descrizione topologia dei contorni
  - raggruppamento dei contorni consecutivi nel fronte
  - avanzamento del fronte
  - interfaccia con il triangolatore:
    - trasferimento ultimo fronte e lati normali
    - integrazioni e correzioni sui lati aperti
  - scrittura file grafico quadplot
3. triangolazione di Delaunay:
  - generazione griglia di backgrid
  - introduzione punti di backgrid e di Steiner
  - introduzione dei punti secondo l'avanzamento di un fronte (sono riportati per ogni passo il numero di punti indicati per l'introduzione `NFR0`, il numero di punti effettivamente introdotti `NADD` ed il segmento di lunghezza massima riscontrato nella griglia rapportato alle condizioni locali di metrica `MAX_LENGTH`)
  - frammentazione triangoli 'angolari'
  - 10 iterazioni dello smoothing Laplaciano (viene riportato il residuo ad ogni passo)
4. costruzione array contenenti i dati delle due fasi di generazione della griglia e interfacciamento
5. salvataggio dei dati nei file `nodes` e `grid`
6. compilazione dei file per la visualizzazione della griglia

## 4 File di Output

**Il file `nodes.gridname`:** contiene la descrizione finale dei punti della griglia; la prima parte tratta tutti i punti nell'ambito bidimensionale della griglia, mentre la seconda parte descrive la collocazione dei punti di contorno nell'ambito monodimensionale dei lati di appartenenza e la loro corrispondenza con i loro punti immagine nella griglia bidimensionale.

|                    |                                                                                        |
|--------------------|----------------------------------------------------------------------------------------|
| <code>np_D</code>  | numero di punti appartenenti al dominio (incluso <code>np_B</code> )                   |
| <code>np_B</code>  | numero di punti appartenenti al contorno                                               |
| <code>idx</code>   | indice del punto (di dominio o di contorno)                                            |
| <code>rr</code>    | coordinate del punto                                                                   |
| <code>jd_jb</code> | punto <code>jd</code> del dominio corrispondente al punto <code>jb</code> del contorno |
| <code>bound</code> | indice del lato di appartenenza del punto                                              |

**Il file `grid.gridname`:** contiene la descrizione finale degli elementi di griglia, prima nell'ambito bidimensionale ed infine nell'ambiente dei lati(dove gli elementi non sono quadrilateri o triangoli, ma segmenti).

|                    |                                                                  |
|--------------------|------------------------------------------------------------------|
| <code>ne_D</code>  | numero di elementi del dominio                                   |
| <code>ne_B</code>  | numero di elementi del contorno                                  |
| <code>idx</code>   | indice dell'elemento(di dominio o di contorno)                   |
| <code>type</code>  | tipo di elemento: 3 = quadrilatero, 2 = triangolo, 1 = segmento  |
| <code>j_m</code>   | matrice di connessione elemento $\Rightarrow$ nodi               |
| <code>ma_m</code>  | matrice di connessione elemento $\Rightarrow$ elementi adiacenti |
| <code>bound</code> | indice del lato di appartenenza del segmento                     |

Le figure 16 e 17 mostrano la descrizione della connettività degli elementi, così come appare nel file `grid.gridname`; si noti che gli indici dei punti sono riportati percorrendo i lati dell'elemento in senso antiorario e che la definizione degli elementi adiacenti é diversa per i triangoli e i quadrilateri. Infine, quando un elemento é disposto sul contorno del dominio, il suo adiacente fittizio é identificato da un indice nullo.

La connettività dei segmenti segue lo schema adottato nei triangoli, con la differenza che i punti dell'elemento sono solamente due.

**I file `backplot.gridname` e `gridplot.gridname`:** il file `backplot.gridname` consente la visualizzazione della griglia generata dal fronte avanzante e della `backgrid` che precede la realizzazione della griglia finale; quest'ultima é visibile nel file `gridplot.gridname`. Entrambi i file sono presentati nel formato PLOTMTV.

| # | IDX  | TYPE |     |  |
|---|------|------|-----|--|
| # | J_M  |      |     |  |
| # | MA_M |      |     |  |
|   | 1    | 2    |     |  |
|   | 4    | 316  | 317 |  |
|   | 8    | 5    | 0   |  |

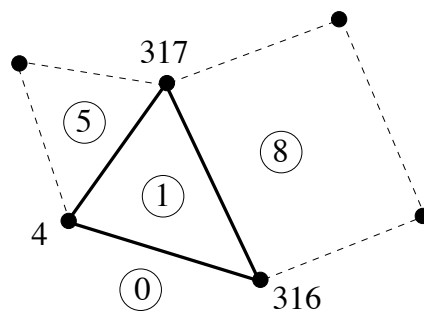


Figura 16: Connettività di un triangolo

|    |    |    |    |
|----|----|----|----|
| :  | :  |    |    |
| :  | :  |    |    |
| :  | :  |    |    |
| 27 | 3  |    |    |
| 10 | 39 | 40 | 11 |
| 26 | 65 | 28 | 0  |

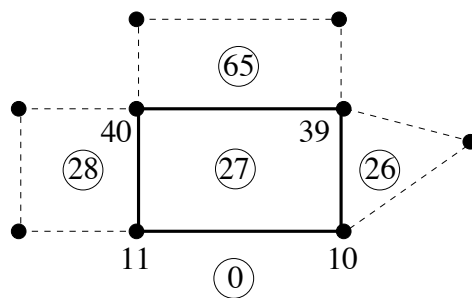


Figura 17: Connettività di un quadrilatero