

May 21, 2018

Specifiche Progetti Matlab 2018 Programmazione e Calcolo Scientifico

Problema 1

Sia assegnata una triangolazione attraverso i file con la seguente struttura (file ottenuti con il programma Triangle <https://www.cs.cmu.edu/~quake/triangle.html>):

quad.1.node

```
13  2  0  1
  1    0  0    1
  2    1  0    1
  3    1  1    1
  4    0  1    1
  5    0.5 0.5    0
  6    0  0.5    1
  7    0.5 0    1
  8    0.5 1    1
  9    0.25 0.25    0
 10    1  0.5    1
 11    0.25 0.75    0
 12    0.75 0.75    0
 13    0.75 0.25    0
# Generated by ./triangle -pqena0.1 quad.poly
```

quad.1.ele

```
16  3  0
  1      5    13    10
  2      9     5     6
  3     11     8     4
  4      4     6    11
  5     13     5     7
  6      5     9     7
  7      1     7     9
  8     12     3     8
  9      6     1     9
 10     10     3    12
 11      5     8    11
 12      6     5    11
 13      5    10    12
 14      8     5    12
 15      2    10    13
 16      7     2    13
# Generated by ./triangle -pqena0.1 quad.poly
```

quad.1.edge

```
28 1
  1  5 13 0
  2 13 10 0
  3 10  5 0
  4  9  5 0
  5  5  6 0
  6  6  9 0
  7 11  8 0
  8  8  4 1
  9  4 11 0
10  4  6 1
11  6 11 0
12  5  7 0
13  7 13 0
14  9  7 0
15  1  7 1
16  9  1 0
17 12  3 0
18  3  8 1
19  8 12 0
20  6  1 1
21 10  3 1
22 12 10 0
23  5  8 0
24 11  5 0
25 12  5 0
26  2 10 1
27 13  2 0
28  7  2 1
```

Generated by ./triangle -pqena0.1 quad.poly

quad.1.neigh

```
16 3
  1  15 13 5
  2  12  9 6
  3  -1  4 11
  4  12  3 -1
  5   6 16 1
  6   7  5 2
  7   6  9 -1
  8  -1 14 10
  9   7  2 -1
10   8 13 -1
11   3 12 14
12  11  4 2
13  10 14 1
14  13  8 11
15   1 16 -1
```

```
16      15  5  -1
# Generated by ./triangle -pqena0.1 quad.poly
```

e un file che definisce dei segmenti nel piano: tracce.trace

```
4 2 0 0
1 -1.0 -2.0
2  2.0  4.0
3  0.5  2.0
4  0.8  0.2
2 0
1 1 2
2 3 4
```

Calcolare le intersezioni tra le tracce e gli elementi della triangolazione salvando le informazioni come segue.

- Per ciascuna traccia individuare l'elenco dei triangoli tagliati dalla traccia. (Se la traccia tocca il triangolo in un vertice o ricopre un intero lato il triangolo non si intende tagliato. Se la traccia ricopre parte di un lato o ha un estremo all'interno di un lato invece è da ritenersi tagliato).
- Per ciascuna traccia individuare l'elenco dei triangoli che condividono almeno un vertice con un triangolo tagliato e per questi memorizzare anche quali vertici o lati sono condivisi con un triangolo tagliato.
- Per ciascun triangolo tagliato creare i sottopoligoni ottenuti tagliando il triangolo con la traccia. Se la traccia termina nel triangolo prolungare il taglio fino ad un lato/vertice del triangolo.
- Per ciascun triangolo tagliato creare una sottotriangolazione conforme alla traccia **ed al suo eventuale prolungamento**.
- Salvare le coordinate curvilinee delle intersezioni della traccia con i segmenti o punti della triangolazione.

Problema 2

Sia assegnata una tetraedrizzazione attraverso i file con la seguente struttura (file ottenuti con il programma Tetgen <http://wias-berlin.de/software/index.jsp?id=TetGen&lang=1>):

barra.1.node

First line: <# of points> <dimension (3)> <# of attributes>
<boundary markers (0 or 1)>

Remaining lines list # of points:

<point #> <x> <y> <z> [attributes] [boundary marker]

...

```
12 3 0 0
 1 1 0 0
 2 1 5 0
 3 1 5 -1.5
 4 1 0 -1.5
 5 -1 0 0
 6 -1 5 0
 7 -1 5 -1.5
 8 -1 0 -1.5
 9 1 2.5 0
10 1 2.5 -1.5
11 -1 2.5 0
12 -1 2.5 -1.5
```

Generated by ./tetgen -pkfennqa25.0 barra.poly

barra.1.edge

First line: <# of edges> <boundary marker (0 or 1)>

Remaining lines list # of edges:

<edge #> <endpoint> <endpoint> ... [boundary marker] ... <tet>

...

```
33 1
 1 12 9 0 1
 2 12 2 0 1
 3 10 2 0 1
 4 10 9 0 1
 5 10 12 0 1
 6 2 9 1 1
 7 12 11 0 2
 8 2 11 0 2
 9 11 9 0 2
10 11 6 1 3
11 12 6 0 3
12 6 2 1 3
13 7 2 0 4
14 7 3 1 4
15 12 3 0 4
16 12 7 1 4
```

17	3	2	1	4
18	10	1	0	5
19	10	4	1	5
20	12	4	0	5
21	12	1	0	5
22	4	1	1	5
23	3	10	1	6
24	11	1	0	7
25	1	9	1	7
26	1	8	0	9
27	12	8	1	9
28	4	8	1	9
29	8	5	1	10
30	12	5	0	10
31	5	1	1	10
32	5	11	1	11
33	6	7	1	12

Generated by ./tetgen -pkfennqa25.0 barra.poly

Ciascuna delle righe del corpo del file contiene: indice del lato, indice del nodo iniziale, indice del nodo finale , marker do bordo (0 il lato è interno o 1 se il lato è di bordo), indice di un tetraedro contenete il lato.

barra.1.face

First line: <# of faces> <boundary marker (0 or 1)>

Remaining lines list # of faces:

<face #> <node> <node> <node> ... [boundary marker] ... <tet left> <tet right>

...

34	1						
	1	12	9	10	0	1	8
	2	12	2	9	0	1	2
	3	10	2	12	0	1	6
	4	10	9	2	1	1	-1
	5	12	11	9	0	2	7
	6	2	11	12	0	2	3
	7	2	9	11	3	2	-1
	8	11	6	2	3	3	-1
	9	12	6	11	2	3	-1
	10	12	2	6	0	3	12
	11	7	2	12	0	4	12
	12	7	3	2	6	4	-1
	13	12	3	7	4	4	-1
	14	12	2	3	0	4	6
	15	10	1	12	0	5	8
	16	10	4	1	1	5	-1
	17	12	4	10	4	5	-1
	18	12	1	4	0	5	9
	19	2	3	10	1	6	-1
	20	12	10	3	4	6	-1

21	12	1	9	0	7	8
22	11	1	12	0	7	11
23	11	9	1	3	7	-1
24	1	9	10	1	8	-1
25	1	8	12	0	9	10
26	1	4	8	5	9	-1
27	12	8	4	4	9	-1
28	8	5	1	5	10	-1
29	12	5	8	2	10	-1
30	12	1	5	0	10	11
31	1	5	11	3	11	-1
32	12	11	5	2	11	-1
33	2	6	7	6	12	-1
34	12	7	6	2	12	-1

Generated by ./tetgen -pkfennqa25.0 barra.poly

La terza colonna da destra contiene un marker di bordo: 0 significa che la faccia è interna, i che la faccia è sulla faccia i del dominio originale. Le ultime due colonne contengono l'indice dei tetraedi che condividono la faccia (-1 significa che non c'è tetraedro).

barra.1.ele

First line: <# of tetrahedra> <nodes per tet. (4 or 10)>
 <region attribute (0 or 1)>

Remaining lines list # of tetrahedra:

<tetrahedron #> <node> <node> ... <node> [attribute]

...

12	4	0		
1	2	10	9	12
2	11	2	9	12
3	6	12	2	11
4	3	12	2	7
5	4	12	1	10
6	3	12	10	2
7	1	11	9	12
8	10	1	9	12
9	4	12	8	1
10	5	12	1	8
11	5	12	11	1
12	6	12	7	2

Generated by ./tetgen -pkfennqa25.0 barra.poly

barra.1.neigh

First line: <# of tetrahedra> 4

Following lines list # of neighbors:

<tetrahedra #> <neighbor> <neighbor> <neighbor> <neighbor>

...

12	4			
1	8	2	6	-1

```

2      1      7      3      -1
3      2      -1     -1     12
4      12     -1     -1      6
5      8      -1     -1      9
6      1      -1      4     -1
7      2      8      11     -1
8      7      1      5     -1
9      10     -1      5     -1
10     9      -1     -1     11
11     7      -1     10     -1
12     4      -1      3     -1
# Generated by ./tetgen -pkfennqa25.0 barra.poly
    e un file che definisce dei poligoni convessi nello spazio fract.pol
First line: <# of points> <dimension (3)> <# of attributes>
           <boundary markers (0 or 1)>
Remaining lines list # of points:
<point #> <x> <y> <z> [attributes] [boundary marker]
...
One line: <# of fractures> [boundary marker]
Following lines list # of fractures:
<fracture #>   <# of corners> <corner 1> ... <corner #> [boundary marker]
...
4 3 0 0
1  0 0  2
2  0 0 -2
3 -2 4  2
4 -2 4 -2
1 0
1 4 1 2 4 3

```

Ripetere le stesse operazioni eseguite sulla triangolazione.

- Per ciascuna frattura individuare l'elenco dei tetraedri tagliati dalla frattura. (Se la frattura tocca il tetraedro in un vertice, un lato o ricopre una intera faccia il tetraedro non si intende tagliato. Se la frattura ricopre parte di una faccia o lato invece o se ha un vertice all'interno di un lato o faccia del tetraedro, questo è da ritenersi tagliato).
- Per ciascuna frattura individuare l'elenco dei tetraedri che condividono almeno un vertice con un tetraedro tagliato e per questi memorizzare anche quali vertici o lati sono condivisi con un tetraedro tagliato.
- Per ciascun tetraedro tagliato creare i sottopoliedri ottenuti tagliando il tetraedro con la frattura. Se la frattura termina nel tetraedro prolungare il taglio fino ad una faccia/lato/vertice del tetraedro. Se la frattura non è passante per il tetraedro tagliato distinguere sulla faccia del taglio la porzione di faccia contenuta nella frattura da quella esterna.
- Suddividere la frattura in poligoni ottenuti dall'intersezione con i tetraedri.

Si noti che il contenuto dei file è puramente rappresentativo per indicare il formato in cui i dati vengono forniti. Elaborare i file di input che si ritengono utili per testare il funzionamento del programma nei vari casi possibili.