

Presentazione progetto

Bessi Ludovico, Boscolo Claudio,
Cataldo Luca, Licari Valentina
A.A. 2017/2018
Programmazione e calcolo scientifico

22/05/2018

Indice

1	Progetto 2D	2
1.1	Lettura file e strutture dati	2
1.1.1	Strutture dati	2
1.1.2	Funzioni di riempimento	4
2	Fratture	4
3	C++	4

1 Progetto 2D

Obiettivo della parte bidimensionale del progetto scritto in matlab é, data una triangolazione e un segmento definito "traccia", portare a termine le seguenti richieste:

- Individuare quali triangoli di partenza vengono tagliati dalla traccia
- Individuare tutti i triangoli vicini ai triangoli tagliati, i.e. tutti quelli che condividono un vertice o un lato con un triangolo tagliato
- Costruire sottopoligoni a partire dalle informazioni ottenute analizzando la posizione della traccia rispetto alla triangolazione di partenza.
- Costruire sottotriangoli a partire dalle informazioni ottenute analizzando la posizione della traccia rispetto alla triangolazione di partenza.
- Salvare le informazioni relative alle intersezioni tra la traccia e la triangolazione, i.e. le ascisse curvilinee dei punti di intersezione.

1.1 Lettura file e strutture dati

In questa sezione vengono brevemente presentate le scelte adottate per salvare le informazioni del file relativo alla triangolazione. Per generare le strutture dati di *node* e *info_trace* é stato utilizzato il comando *repmat* di matlab.

1.1.1 Strutture dati

Node

Si é scelto di salvare nella struttura *node* le coordinate dei punti della triangolazione del piano cartesiano 2D, un "side" del punto rispetto alla traccia (un indicatore che può assumere valore 0,1,-1,2) che esprime dove si trovi il punto rispetto alla traccia. Il campo *side* assume valore '0' se la posizione del punto é non nota, '1' o '-1' se il punto si trova alla sinistra o alla destra della traccia e '2' se si trova sulla retta della traccia. Il campo "tol" presenta una tolleranza relativa al problema, calcolata tenendo conto delle dimensioni dei lati della triangolazione e verrà meglio descritta in seguito. 'Edges' e 'Triangles' sono vettori che contengono rispettivamente gli indici dei segmenti e dei triangoli che hanno l'n-esimo nodo della struttura dati come estremo. 'tot_edges' e 'tot_triangles' contengono rispettivamente il numero di segmenti e il numero di triangoli condivisi da ogni nodo della struttura dati. L'ultimo campo di *node* contiene l'ascissa curvilinea *s* del punto nel caso in cui questo si trovi sulla traccia.

Triangle

É una matrice di `n_triangles` righe e 10 colonne. Le colonne 1,2,3 contengono i vertici del triangolo (indici dei nodi). Le colonne 4,5,6 contengono degli status relativi ai lati dell' `n`-esimo triangolo. Gli status indicano le possibili posizioni del segmento rispetto alla traccia (interno,secante,non incidente). É importante osservare che la colonna 4 della matrice contiene lo status del segmento formato dai nodi 2-3 ; la colonna 5 della matrice contiene lo status del segmento formato dai nodi 1-3 ; la colonna 6 della matrice contiene lo status del segmento formato dai nodi 1-2. Le colonne 7,8,9 contengono le eventuali intersezioni dei segmenti del triangolo con la traccia, seguendo una numerazione analoga a quella poco prima descritta. La colonna 10, ultima, contiene un flag che monitora l'operazione di messa in coda del triangolo la quale serve per verificare se é tagliato.

Edge

É una matrice contenente i segmenti indicizzati della triangolazione, le due colonne della matrice contengono gli indici degli estremi del segmento.

Neigh

É una matrice le cui colonne contengono i vicini del triangolo `n`-esimo.

Trace vertex

É una matrice le cui colonne contengono le *coordinate* degli estremi della traccia.

Trace

É una matrice le cui colonne contengono gli indici degli estremi della traccia riferiti alla matrice `'trace_vertex'`.

Info trace

É una struttura che contiene, per ogni traccia, le informazioni relative alle richieste del problema. Il campo `'cut_tri'` é una sottostruttura di `'info_trace'` e contiene le informazioni relative ai triangoli tagliati. Per ogni triangolo tagliato di questa sottostruttura vengono salvati in `'points'` i punti in `'poly_1'` i punti del primo poligono che si forma dopo il taglio, in `'poly_2'` i punti del secondo poligono che si forma dopo il taglio, in `'tri'` le informazioni relative alla sottotriangolazione, `'tri'` é una matrice di dimensione iniziale 3x3 (numero plausibile di sottotriangoli) che descrive utilizzando i punti le cui coordinate sono salvate in `points`, i triangoli della sottotriangolazione.

1.1.2 Funzioni di riempimento

2 Fratture

... delle nostre ossa

3 C++

`std::vector` per tutto