

Programmazione e Calcolo Scientifico - a.a. 2018/2019

Progetto C++

Per lo svolgimento di questo progetto verranno forniti:

- Classi per la generazione di una mesh triangolare in un dominio poligonale. I triangoli sono sempre “orientati” in senso **antiorario**.
- Classi per la memorizzazione di una mesh e per eseguire su essa operazioni elementari di modifica, consultazione. Possibili criteri di raffinamento:
 - Newest vertex bisection: si taglia a metà il lato opposto al vertice 2 del triangolo, che è il lato tra il vertice 0 e il vertice 1 del triangolo.
 - * Si parte ruotando la numerazione dei nodi in modo che il vertice 0 sia il nodo di inizio del lato più lungo (nell’orientazione antioraria del triangolo).
 - * Ad ogni passo si taglia il triangolo connettendo il vertice 2 con il punto medio del lato opposto.
 - * I due triangoli figli vengono generati in modo che il nuovo nodo inserito sia il vertice 2 per i due traingoli generati.
 - Longest edge bisection: si taglia a metà il lato pi’u lungo del triangolo.
 - * Si parte ruotando la numerazione dei nodi in modo che il vertice 0 sia il nodo di inizio del lato più lungo (nell’orientazione antioraria del triangolo).
 - * Ad ogni passo si taglia il triangolo connettendo il vertice opposto al lato più lungo con il punto medio del lato più lungo.
 - * I due triangoli figli vengono generati in modo che il vertice 2 per ciascuno di essi sia il vertice opposto al lato più lungo (che è sempre il lato 0).

Chiamiamo lato “base” il lato opposto al newest vertex o il lato più lungo a seconda del criterio di raffinamento scelto.

Scrivere:

1. Una fuzione che marchi le basi dei triangoli marcati per il raffinamento.
2. Una funzione che raffini la mesh triangolare in modo da ottenere una mesh conforme. I lati della triangolazione che vengono spezzati diventano inattivi, vanno tagliati i triangoli finché hanno almeno un lato marcato o inattivo. Questo processo si ripete iterativamente finché non ci sono più lati o triangoli da raffinare. Contestualmente si aggiornano anche le informazioni di vicinanza per i lati della triangolazione.
3. Una funzione che aggiorna le informazioni di vicinanza per i triangoli.
4. Una funzione per il raffinamento uniforme della triangolazione (ogni triangolo è diviso in 4 triangoli simili unendo i punti medi dei lati del triangolo padre).

Suggerimenti

Algoritmo di completion

- Ogniqualevolta un triangolo ha tutti i lati marcati per il raffinamento viene diviso in 4 triangoli simili anche se si sta operando con newest vertex o longest edge bisection.
- Procedere a marcare tutte le basi dei triangoli che hanno almeno un lato marcato.

- In caso di raffinamento “longest edge” è possibile introdurre una variante al passo precedente. Possiamo considerare base anche un lato che non sia necessariamente il più lungo, ma abbia una lunghezza che è inferiore a quella del lato più lungo di una percentuale fissata (esempio 15%). Se questo lato del triangolo risulta marcato per il raffinamento e quello più lungo no ed utilizzando quel lato per il raffinamento si può evitare la propagazione del raffinamento si usa questo lato come base e non si marca il lato più lungo.