

Programmazione e Calcolo Scientifico

Presentazione Progetto PCS 2025

Giovannetti Martino, Mastalia Adriano, Sala Leonardo

Matematica per l'Ingegneria
Politecnico di Torino

November 4, 2025

1. Struttura del software
2. Funzionalità del software

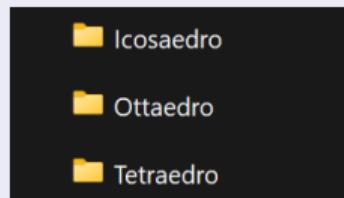
Impostazione del problema

Consegna del progetto

Generare un poliedro geodetico dando in input i numeri della notazione di Wenninger $\{p, q\}_{(b,c)}$. Eventualmente, generare il duale e cercare il cammino minimo.

Gestione dei file di input

Creazione nella cartella del progetto di una directory */SolidiPlatonici* contenente 3 sottocartelle, una per tipo di poliedro platonico di nostro interesse.



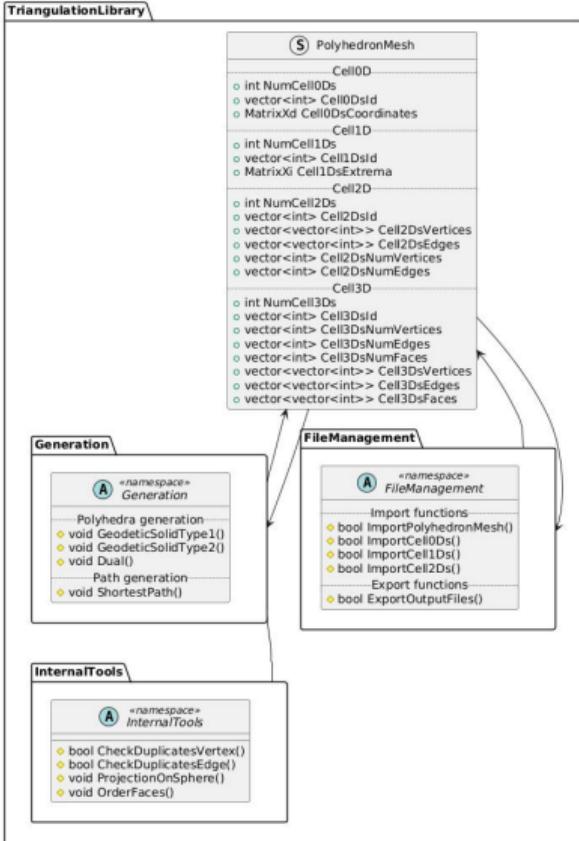
Contenuto della directory
/SolidiPlatonici

ID;X;Y;Z	ID;Origin;End
0;1;1;1	0;0;1
1;1;-1;-1	1;1;2
2;-1;1;-1	2;2;0
3;-1;-1;1	3;2;3
	4;3;0
	5;3;1

ID;Vertex1;Vertex2;Vertex3;Edge1;Edge2;Edge3;
0;0;1;2;0;1;2
1;0;2;3;2;3;4
2;1;2;3;1;3;5
3;0;1;3;0;5;4

Contenuto dei file *Cell0Ds.txt*,
Cell1Ds.txt, *Cell2Ds.txt* in
/SolidiPlatonici/Tetraedro

Struct e funzionalità implementate

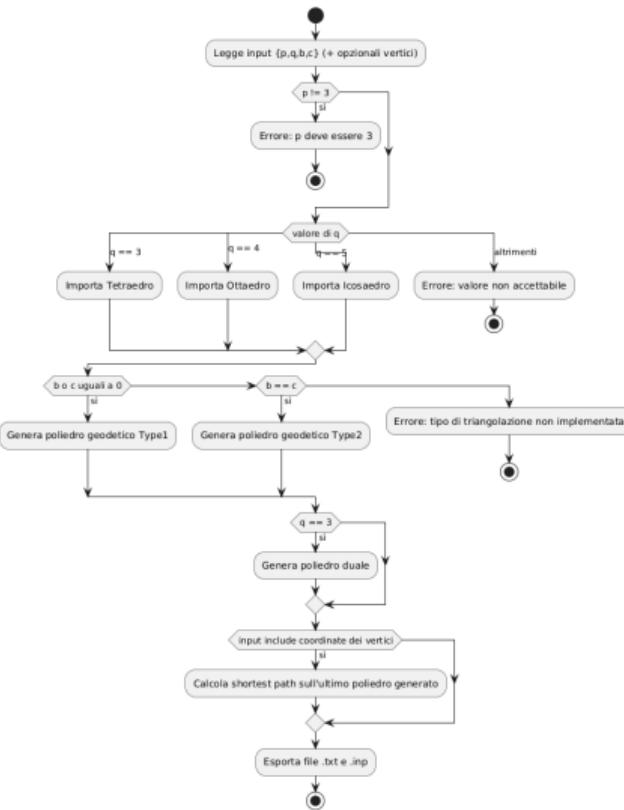


Namespace **TriangulationLibrary**

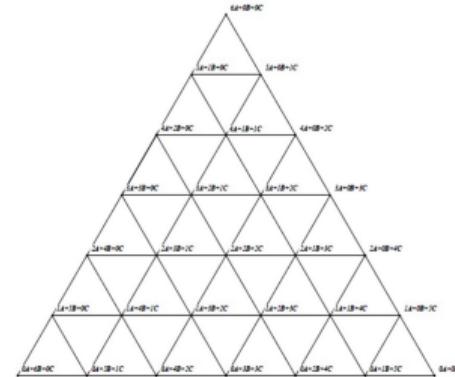
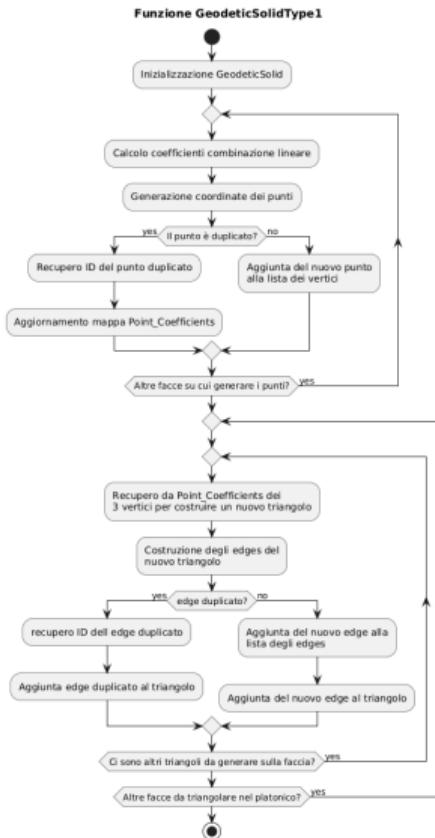
- struct **PolygonalMesh** → struttura dati
- namespace **FileManagement** → funzioni per l'importazione e l'esportazione dei dati
- namespace **Generation** → funzioni per la generazione dei geodetici e del cammino minimo
- namespace **InternalTools** → funzioni di supporto

Ogni funzionalità è stata opportunatamente testata con degli unit test usando la libreria Google Test.

Funzionamento del programma



Namespace Generation - GeodeticSolidType1

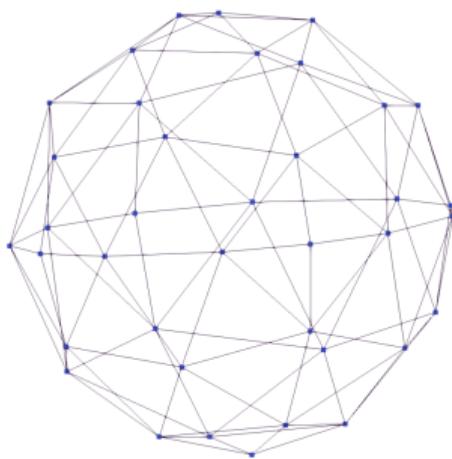


Problemi rilevanti incontrati

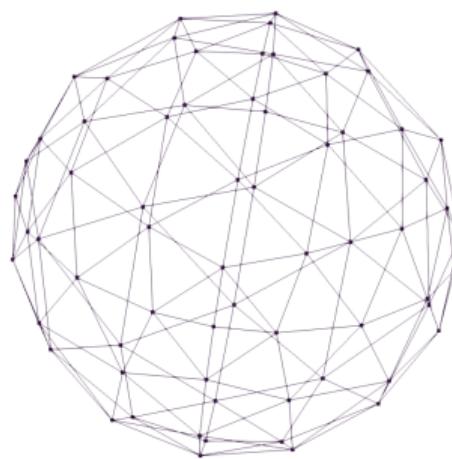
- Evitare generazione di vertici e spigoli duplicati → funzioni CheckDuplicatesVertex() e CheckDuplicatesEdge()
- Tenere traccia della posizione di un vertice su una faccia del poliedro → mappa con chiave $\{A, B, C, id_faccia_pp\}$
- Gestione indici matrici → inizializzare a dimensione del caso peggiore

Namespace Generation - GeodeticSolidType1

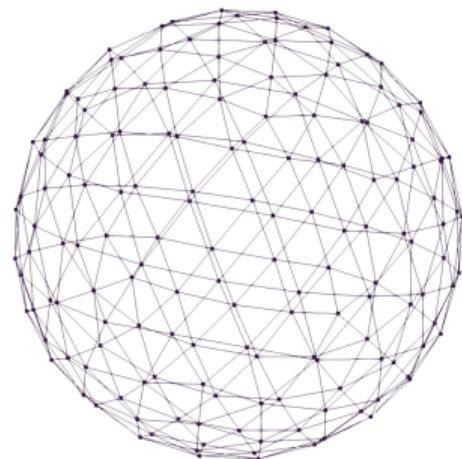
Alcuni esempi di solidi geodetici da triangolazione type1:



Solido di parametri $\{3, 4+\}_{(3,0)}$

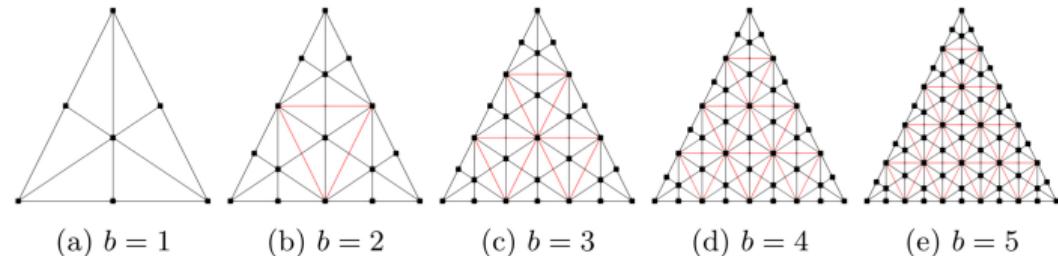
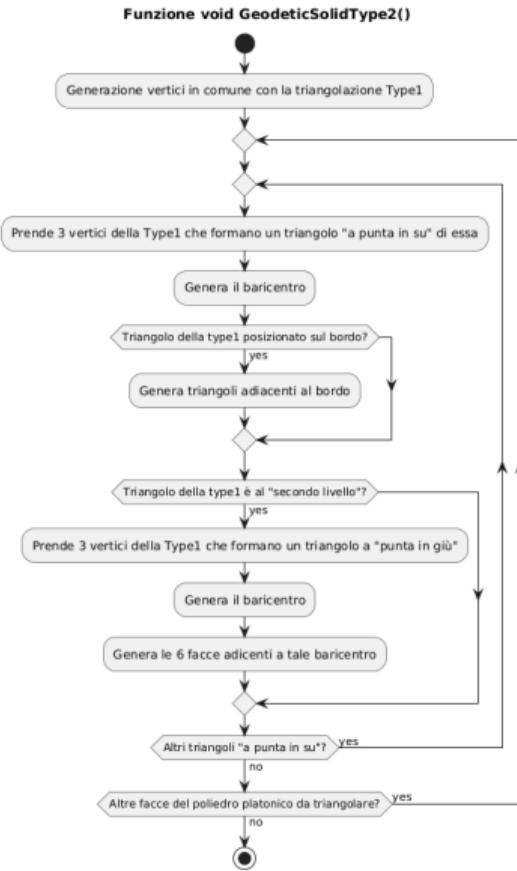


Solido di parametri $\{3, 4+\}_{(4,0)}$



Solido di parametri $\{3, 5+\}_{(4,0)}$

Namespace Generation - GeodeticSolidType2

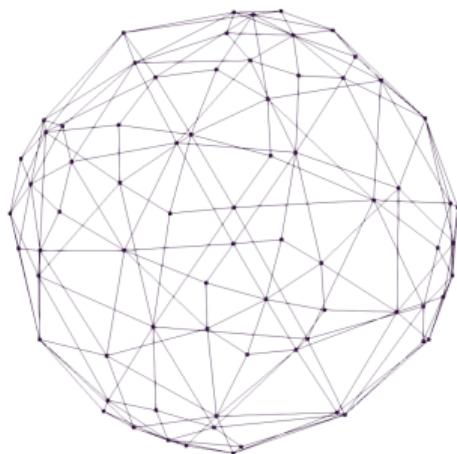


Problemi rilevanti incontrati

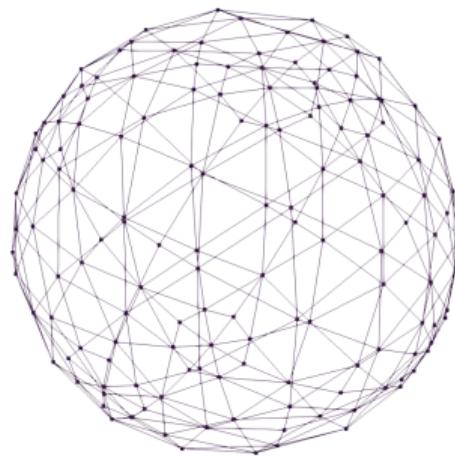
- Eventuali duplicati → stessa soluzione della type1
- Tenere traccia dei vertici della type1 → stessa mappa della type1
- Tenere traccia dei baricentri → mappa con chiave $\{i, j\}$ e nel valore mappato l'id baricentro associato
- Gestione indici matrici → inizializzare a dimensione del caso peggiore

Namespace Generation - GeodeticSolidType2

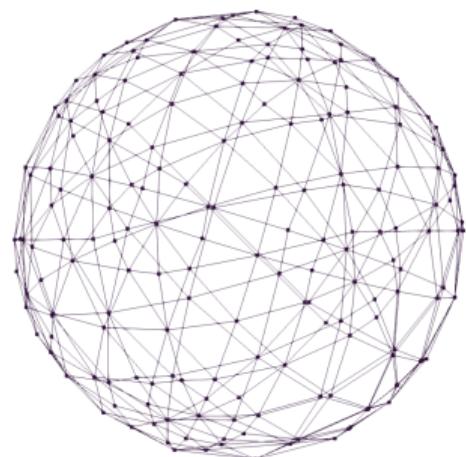
Alcuni esempi di solidi geodetici da triangolazione type2:



Solido di parametri $\{3, 4+\}_{(2,2)}$

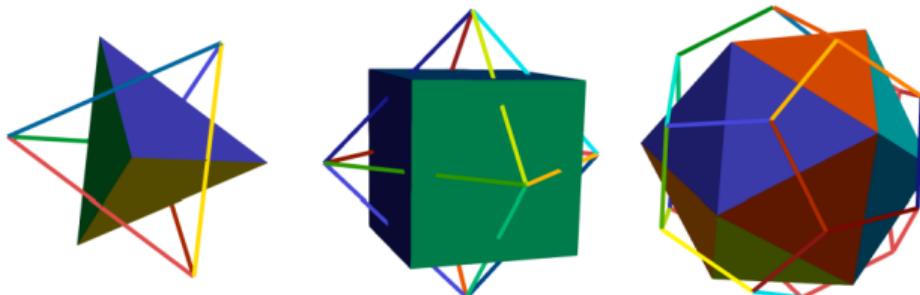


Solido di parametri $\{3, 4+\}_{(3,3)}$



Solido di parametri $\{3, 5+\}_{(2,2)}$

Namespace Generation - Dual



Problemi rilevanti incontrati

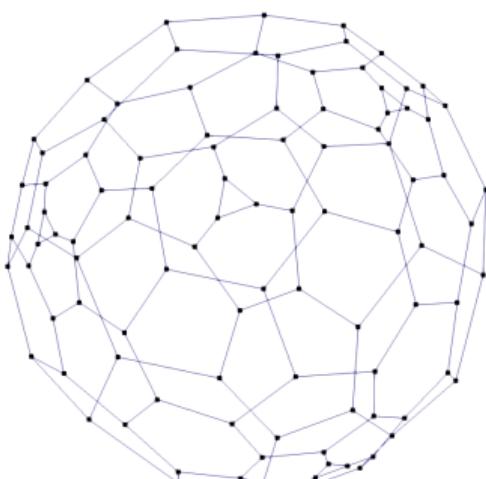
- Associare a ogni faccia di partenza il corrispettivo baricentro → mappa con ID faccia in chiave e ID baricentro in valore
- Il numero di lati della nuova faccia non è sempre 3 come nelle triangolazioni → occorre calcolarlo volta per volta
- Le facce con un vertice in comune devono essere ordinate in modo da essere adiacenti una all'altra → funzione OrderFaces()

Namespace Generation - Dual

Alcuni esempi di solidi duali nel caso $q = 3$:



Solido di parametri $\{3, 3+\}_{(0,3)}$

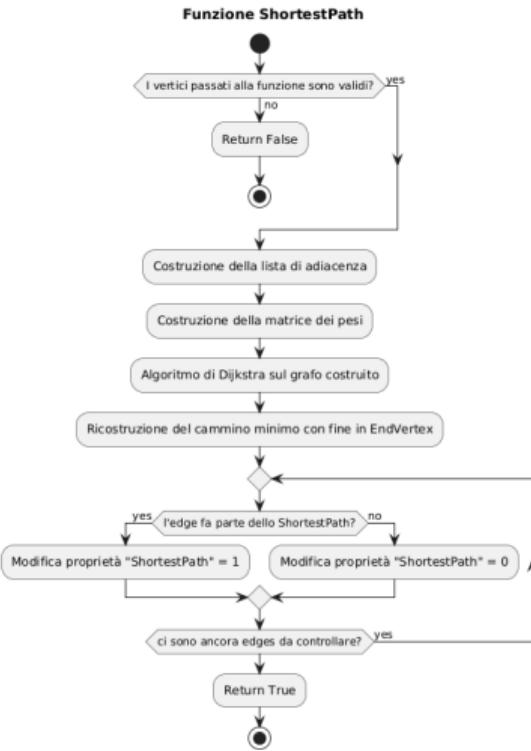


Solido di parametri $\{3, 3+\}_{(0,5)}$



Solido di parametri $\{3, 3+\}_{(2,2)}$

Namespace Generation - ShortestPath



Complessità dell'algoritmo

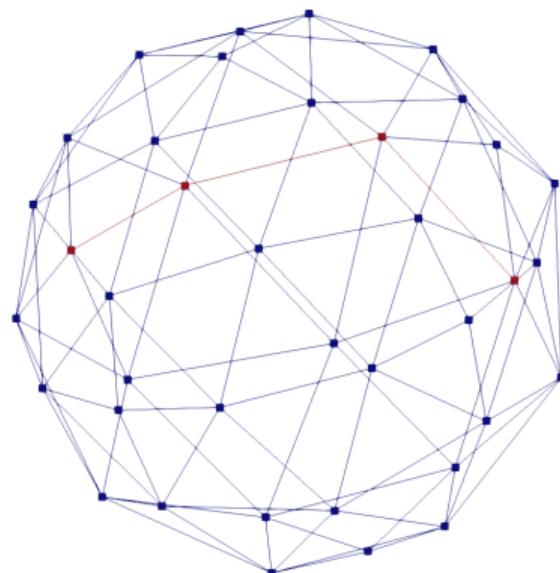
- Costruzione della lista di adiacenza: $\mathcal{O}(n \cdot e)$
- Costruzione della matrice dei pesi: $\mathcal{O}(n)$
- Ricerca cammino minimo: $\mathcal{O}((n + e) \log(n))$
- Ricostruzione del percorso: $\mathcal{O}(n)$
- Ricerca degli edges da colorare: $\mathcal{O}(n \cdot e)$

Problemi rilevanti incontrati

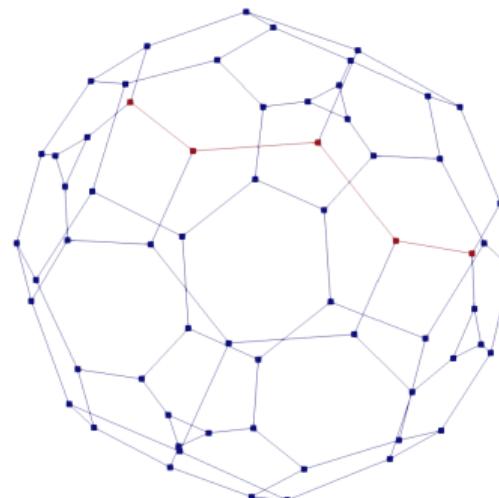
- Dopo la proiezione sulla sfera le lunghezze degli edges non sono tutte uguali → algoritmo di Dijkstra
- La priority queue della STL non supporta l'operazione "DecreaseKey" → aggiunta degli elementi con nuova priorità alla coda e controllo opportuno per evitare cicli inutili

Namespace Generation - ShortestPath

Esempi di generazione del cammino minimo:



Cammino minimo sul solido di parametri $\{3, 4+\}_{(3,0)}$ tra i vertici di id 5 e 19



Cammino minimo sul solido di parametri $\{3+, 3\}_{(4,0)}$ tra i vertici di id 5 e 15

```
appuser@THIS_IS_PCS_DOCKER:~/Data/ProgettoPCS2025/Debug$ ./Project 3 3 4 0 5 15
The platonic polyhedron corresponding to the Schläfli symbol {3, 3} is a tetrahedron
Generation of a generalized Goldberg polyhedron with Schläfli symbol {3+, 3}_{(4, 0)}
Shortest path between the vertices of id 5 and 15 found
Total length of the walk: 1.6442 - Number of edges between nodes: 4
```

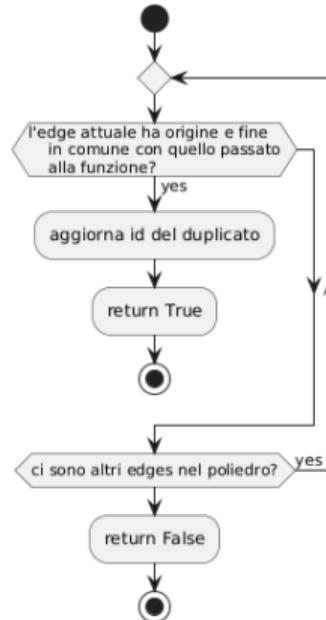
Esempio di output sul terminale relativo al caso sopra

Namespace InternalTools

Funzione CheckDuplicatesVertex



Funzione CheckDuplicatesEdge



Funzione OrderFaces

