

[Home](#) [Content](#)



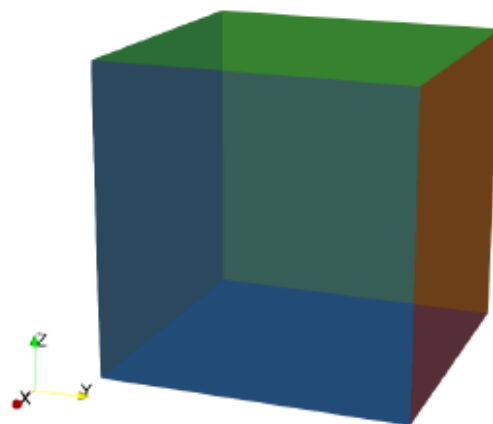
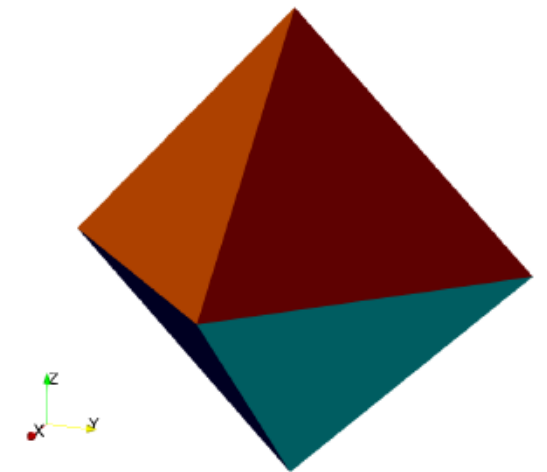
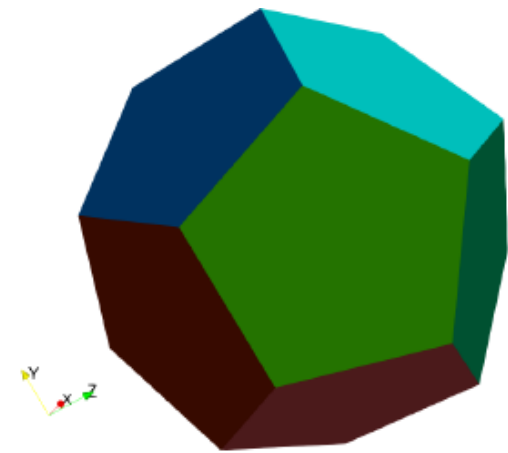
Politecnico
di Torino

SOLIDI PLATONICI

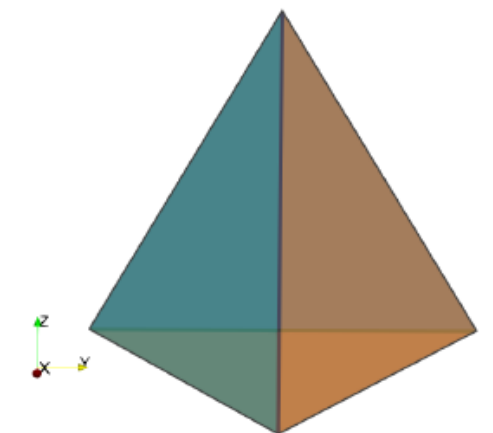
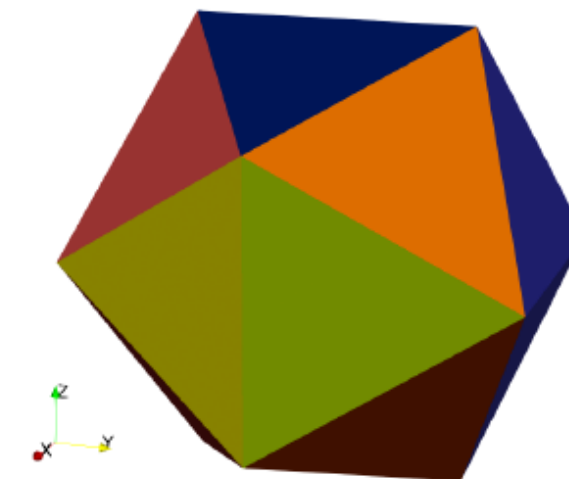
PRESENTAZIONE PROGETTO DI GRUPPO

NOEMI LAINÀ S311037

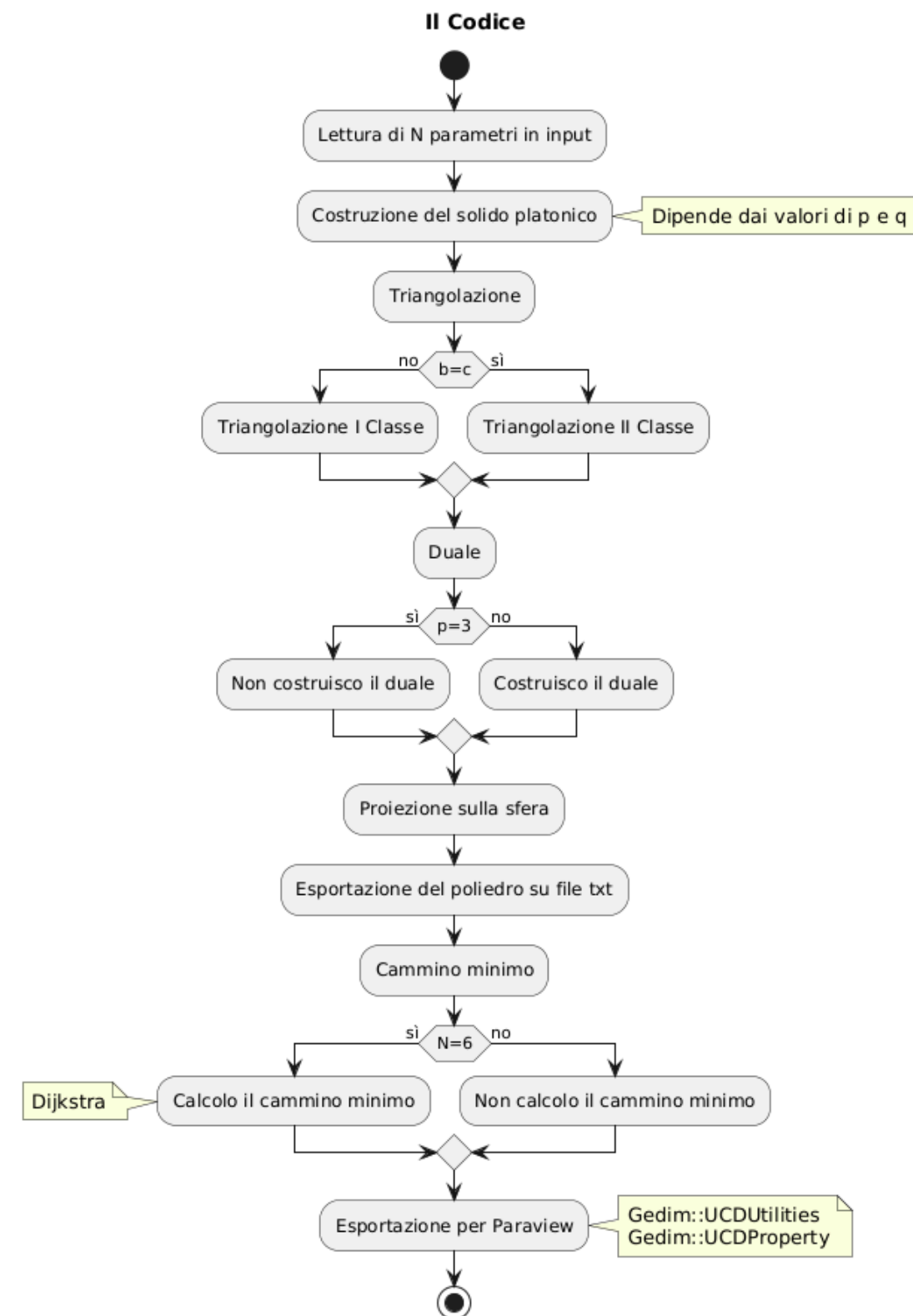
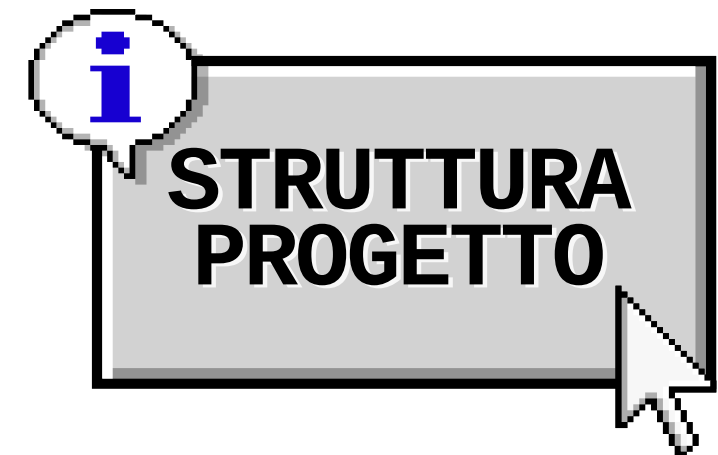
ULDERICO GUZZARDI S307530

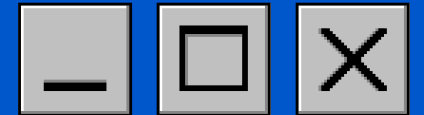


Start



[Home](#) [Content](#)





[Home](#) [Content](#)

PolygonalLibrary

Ⓢ PolygonalMesh

Cell0Ds

- unsigned int NumCell0Ds
- vector<unsigned int> Cell0DsId
- Eigen::MatrixXd Cell0DsCoordinates
- vector<unsigned int> Cell0DsShortPath

Cell1Ds

- unsigned int NumCell1Ds
- vector<unsigned int> Cell1DsId
- Eigen::MatrixXi Cell1DsExtrema
- vector<unsigned int> Cell1DsShortPath
- vector<unsigned int> Cell1DsEsistente

Cell2Ds

- unsigned int NumCell2Ds
- vector<unsigned int> Cell2DsId
- vector<unsigned int> Cell2DsNumVert
- vector<unsigned int> Cell2DsNumEdg
- vector<vector<unsigned int>> Cell2DsVertices
- vector<vector<unsigned int>> Cell2DsEdges

Cell3Ds

- unsigned int NumCell3Ds
- unsigned int Cell3DsId
- unsigned int Cell3DsNumVert
- unsigned int Cell3DsNumEdg
- unsigned int Cell3DsNumFaces
- vector<unsigned int> Cell3DsVertices
- vector<unsigned int> Cell3DsEdges
- vector<unsigned int> Cell3DsFaces

Utilizza:

- STL (vector)
- Eigen (MatrixXd, MatrixXi)

Rappresenta un poligono costituito da:

- Cell0D (vertici)
- Cell1D (lati)
- Cell2D (facce)

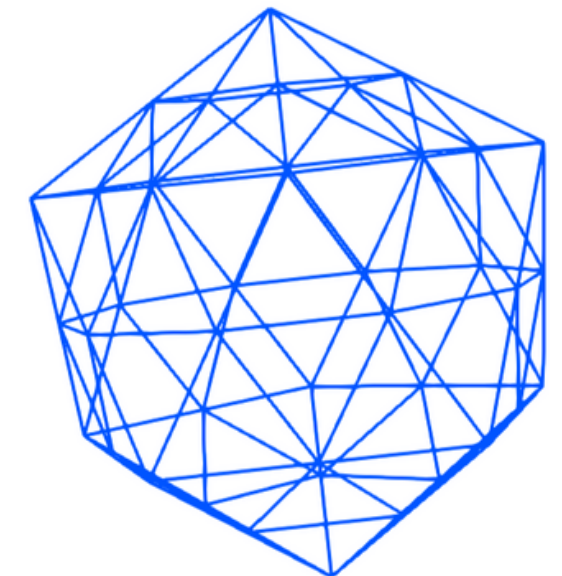
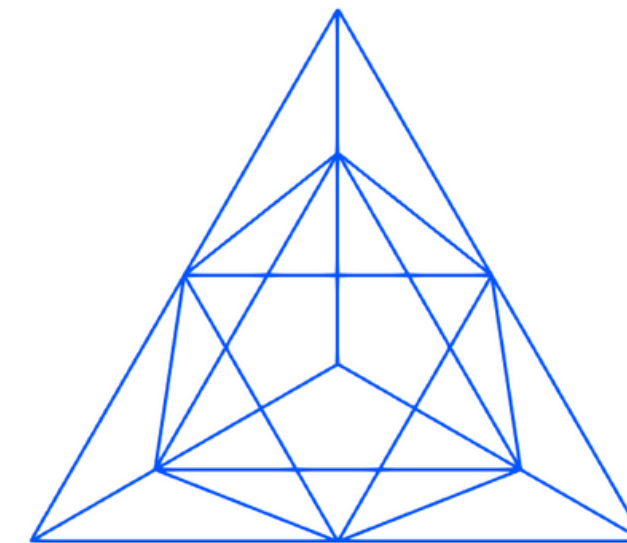
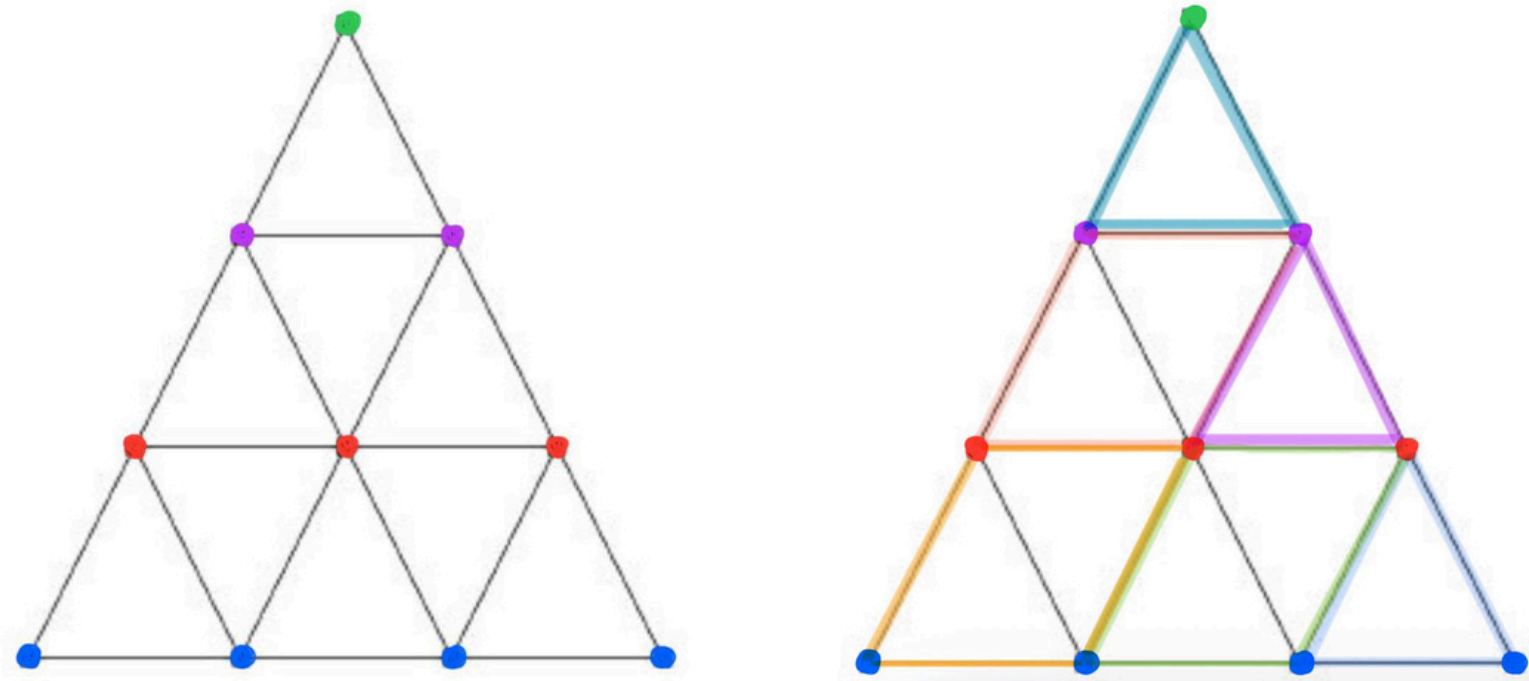
i

STRUCT



Home Content

- `vector<vector<unsigned int>> strati_vertici`



Esempio: memorizzazione per strati sui verti, e suddivisione per “parallelogrammi” per le facce triangolari (b=3).

Esempio: Tetraedro ed Icosaedro con triangolazione I classe



[Home](#) [Content](#)

3 "CATEGORIE" DI VERTICI:

1. Vertici della triangolazione di classe 1

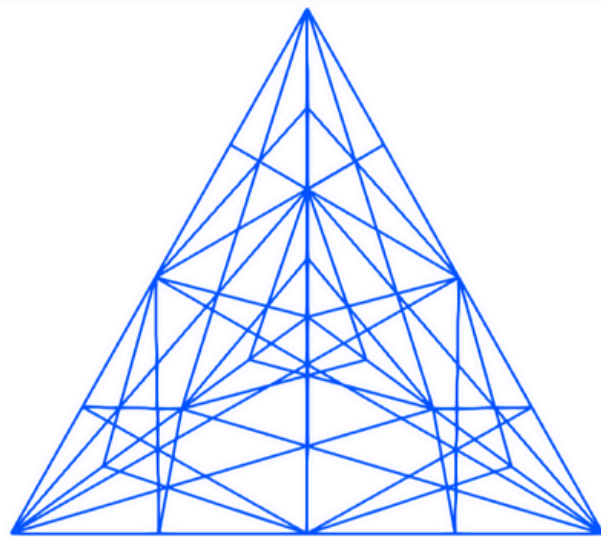
`vector<vector<unsigned int>> strati_vertici`

2. **Baricentri**

`vector<vector<unsigned int>> strati_baricentri`

3. Vertici intermedi

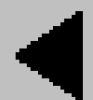
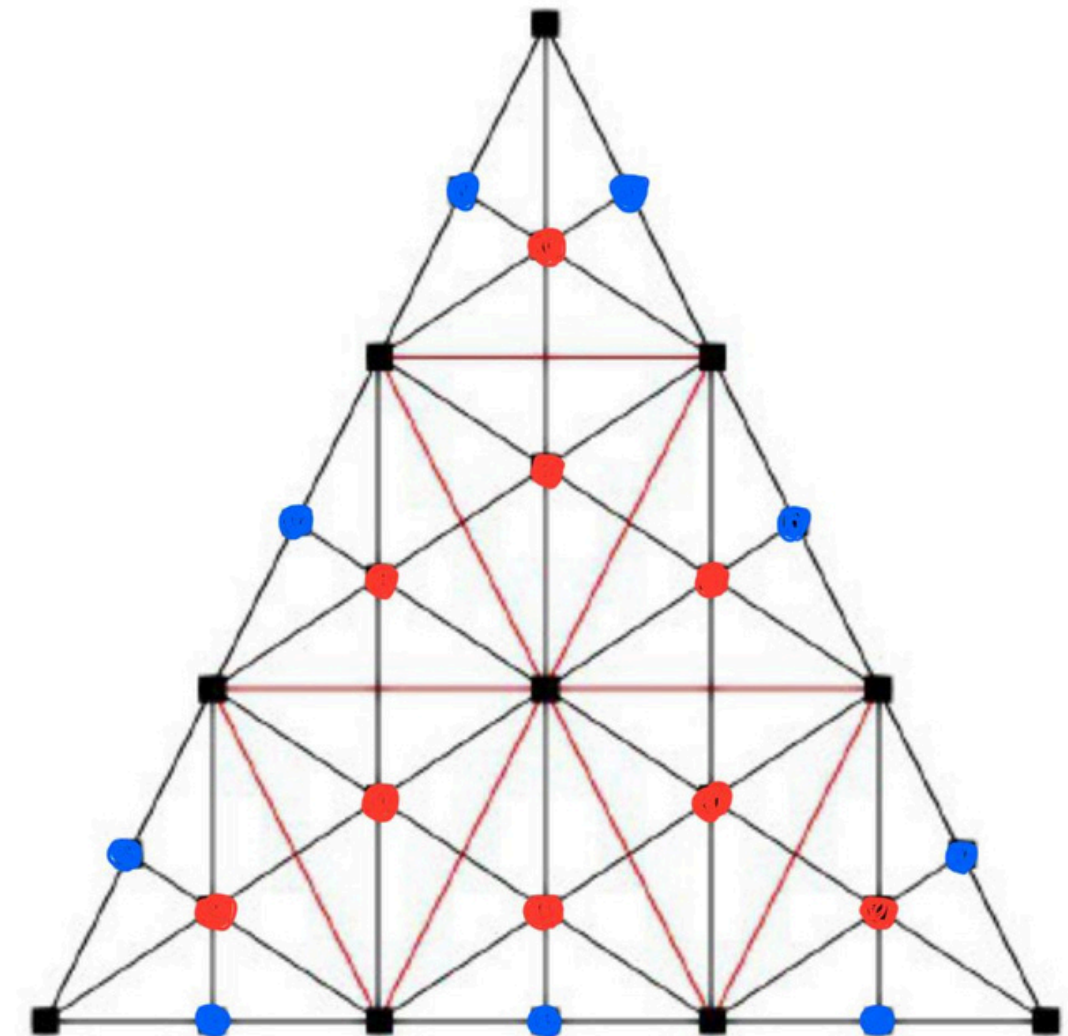
`vector<vector<unsigned int>> strati_intermedi`

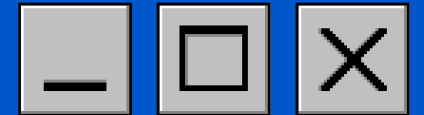


Esempio: tetraedro con triangolazione II classe



**TRIANGOLAZIONE
II CLASSE**





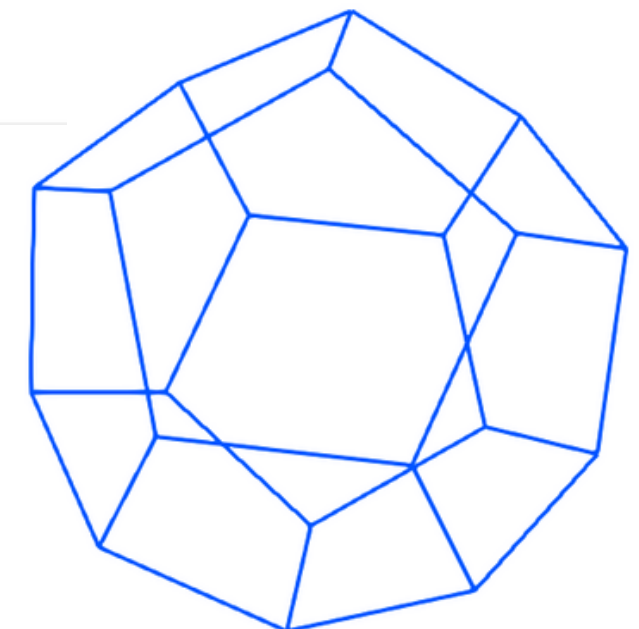
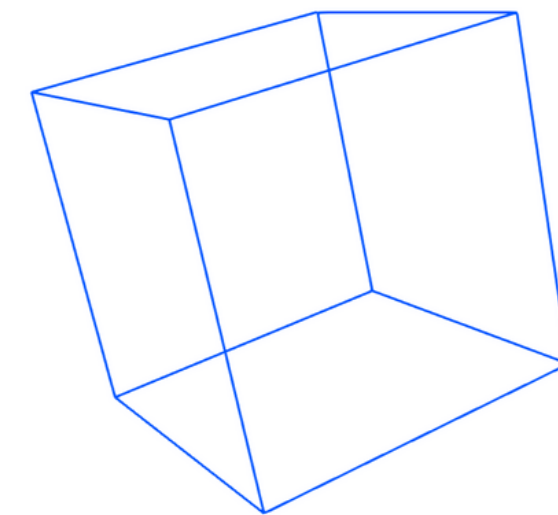
Home **Content**

- `map<vector<unsigned int>, vector<unsigned int>> lati_facce`
- `map<unsigned int, vector<unsigned int>> vertici_facce`
- `map<unsigned int, vector<unsigned int>> adiacenti`

<i>Poliedro iniziale</i>		<i>Duale</i>
vertici_poliedro	→	vertici_duale = facce_poliedro
num_lati	→	num_lati
facce_poliedro	→	facce_duale = vertici_poliedro

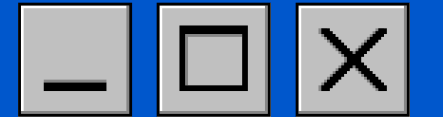


COSTRUZIONE DEL DUALE



Esempio: cubo e dodecaedro





Home Content

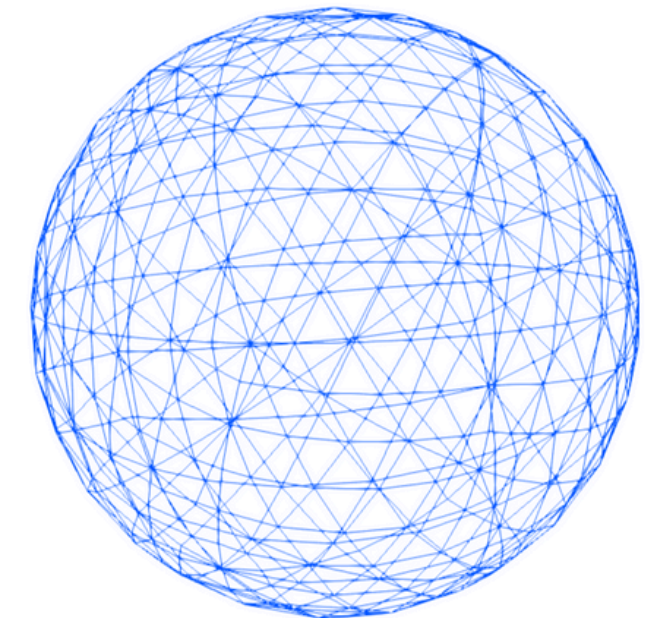
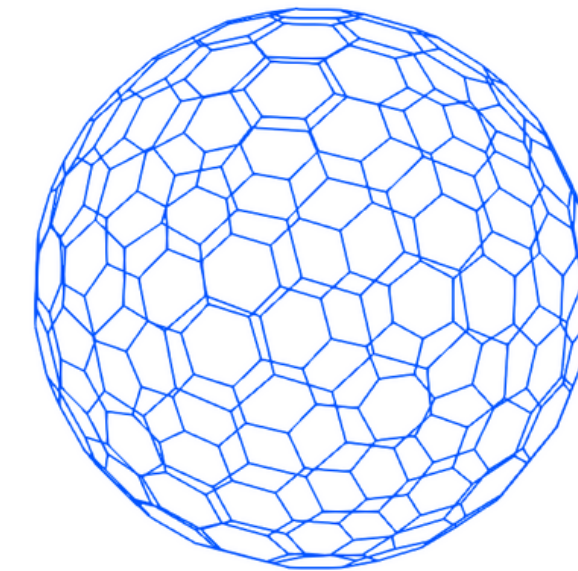


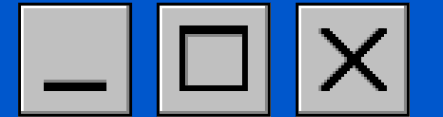
PROIEZIONE SULLA SFERA

- Eseguita dopo la triangolazione e/o dualizzazione

- `cell1Ds_properties[0].Label = "Lati Esistenti";`
`cell1Ds_properties[0].UnitLabel = "-";`
`cell1Ds_properties[0].NumComponents = 1;`

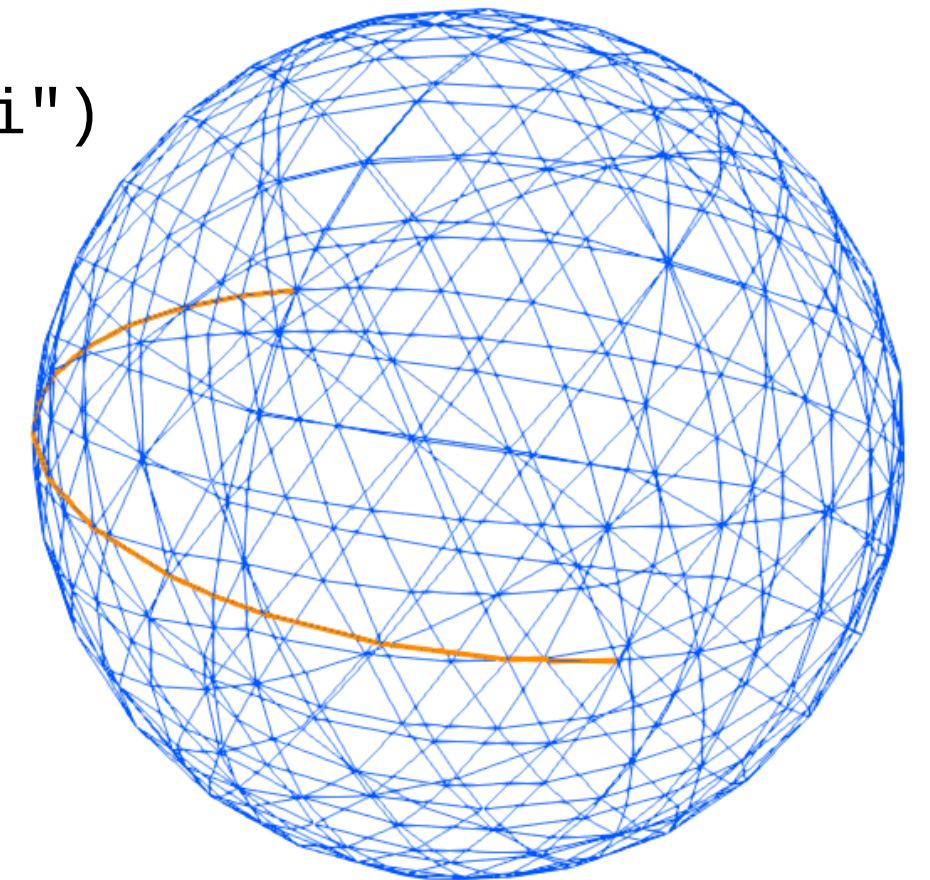
- `mesh.Cell0DsCoordinates(j,i)= mesh.Cell0DsCoordinates(j,i)/norma`
[j = 0, 1, 2 relative ad asse x, y, z]





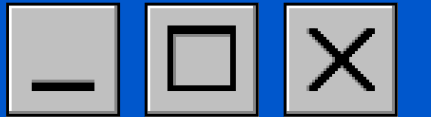
Home **Content**

- Algoritmo di Dijkstra
- `vector<vector<pair<unsigned int, double>>> LA(N)`
- `priority_queue<pair<double, unsigned int>, vector<pair<double, unsigned int>>, greater<pair<double, unsigned int>>> pq;`
- `cell0Ds_properties[0].Label = "ShortPathVertici"; ("ShortPathVertici")`
`cell0Ds_properties[0].UnitLabel = "-";`
`cell0Ds_properties[0].NumComponents = 1;`



Esempio: cammino minimo su
icosaedro proiettato con
triangolazione II classe





[Home](#) [Content](#)

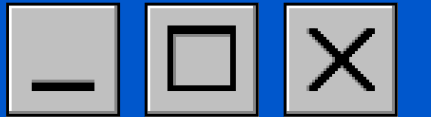


GOOGLE TEST



- Costruzione poliedro (5 Test)
- Controllo esistenza e appartenenza vertici/lati (3 Test)
- Triangolazione (5 Test)
- Duale (1 Test)
- Proiezione sulla sfera (1 Test)
- Cammino minimo (1 Test)





[Home](#) [Content](#)

GRAZIE PER
L'ATTENZIONE.

