

GEOMETRY

ALFONSO FASCÌ, DAVIDE FERRARESE, MATTIA PENATI

1. DEFINIZIONE DELL'INTERFACCIA

Il namespace **Geometry** contiene le classi relative alla manipolazione di oggetti geometrici primitivi (ad esempio segmenti, triangoli, quadrati, . . .), particolarmente utili nell'implementazione di formule di quadrature numerica, poiché contengono le funzioni per il passaggio di coordinate tra il dominio di riferimento $\hat{\Omega}$ e quello considerato Ω .

Tale classe dovrà contenere tre tipi di istruzioni:

- le trasformazioni delle coordinate dei punti dal riferimento $\hat{\Omega}$ a quello attuale Ω ,
- le trasformazioni inverse,
- le informazioni geometriche, come l'area del dominio e il determinante della matrice Jacobiana di $\mathbf{x} = \mathbf{x}(\hat{\mathbf{x}})$, fondamentali per l'implementazione delle formule di quadratura numerica.

L'unico dettaglio implementativo degno di nota è il costruttore, ideato per mettere in comunicazione la classe con la struttura dati della triangolazione. Infatti si è scelto di utilizzare la programmazione generica, al fine di rendere indipendente la nostra struttura, conservando al suo interno tutte le informazioni geometriche. Con l'utilizzo dei *template* si lascia la possibilità all'utente di specializzare il solo costruttore, e non l'intera struttura dati, in base ad una particolare scelta di implementazione per la triangolazione.

```
class Geometry {  
    private:  
        Matrix<double,2,2> A;  
        Matrix<double,2,2> invA;  
        Vector<double,2> b;  
  
    public  
        template <typename Face>
```

```

        Geometry (Face const & f);
};

template <◇
Geometry :: Geometry<MyOwnFaceType> (MyOwnFaceType const & f) {
    /* definizione di A, invA e b */
}

```

2. LA CLASSE **TRIANGLE**

Siccome nell'implementazione dei metodi ad elementi finiti avremo a che fare con delle triangolazioni di domini, vediamo come è stata implementata una classe di questo tipo per oggetti di forma triangolare, che chiameremo **Triangle**. Quindi, considerato il triangolo K di vertici $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^2$, si è deciso di identificarlo, tramite la trasformazione affine $\mathbf{x} = A\hat{\mathbf{x}} + \mathbf{b}$, con il dominio di riferimento \hat{K} , che ha vertici

$$\hat{\mathbf{x}}_0 = (0, 0), \quad \hat{\mathbf{x}}_1 = (1, 0), \quad \hat{\mathbf{x}}_2 = (0, 1).$$

Ovviamente la mappa dovrà soddisfare le condizioni $\mathbf{x}_i = A\hat{\mathbf{x}}_i + \mathbf{b}$ con $i = 0, 1, 2$, così da ottenere

$$A = (\mathbf{x}_1 - \mathbf{x}_0 | \mathbf{x}_2 - \mathbf{x}_0), \quad \mathbf{b} = \mathbf{x}_0.$$

Detto questo è immediato definire l'interfaccia per la classe **Triangle**, evitando di entrare nei dettagli dell'implementazione.

```

class Triangle {

    /* Trasformazioni dal riferimento all'attuale */
    double x (double const & x, double const & y) const;
    double y (double const & x, double const & y) const;

    /* Trasformazioni inverse */
    double invx (double const & x, double const & y) const;
    double invy (double const & x, double const & y) const;

    /* Informazioni geometriche */
    double area () const;

```

```
double det (double const & x, double const & y) const;  
};
```