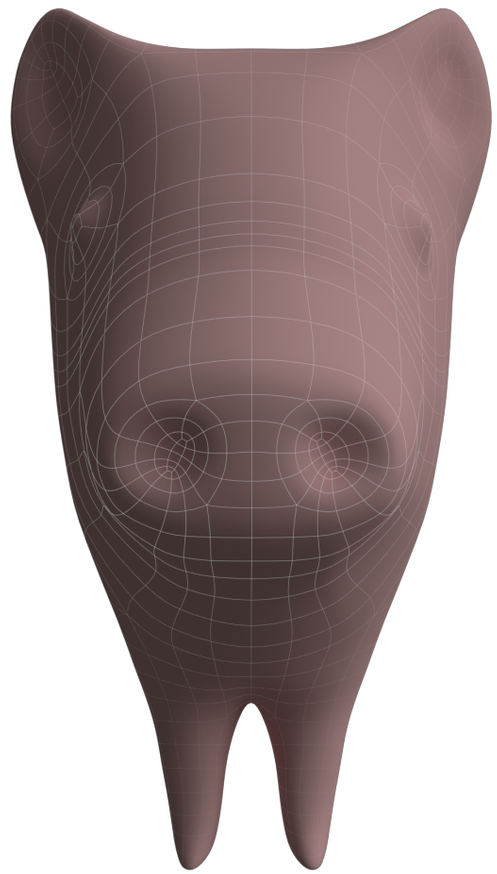


Geometria 3D per Programmatori

develer

giorgio marcias - giorgio@develer.com

Geometria 3D per Programmatori

- Introduzione
  - computer graphics 3D e geometria computazionale
- Modelli 3D
  - rappresentazioni matematiche di un oggetto tridimensionale
- Algoritmi e Strutture Dati
  - elementi costitutivi di un modello 3D
  - elaborare/modificare un modello 3D

3D Computer Graphics

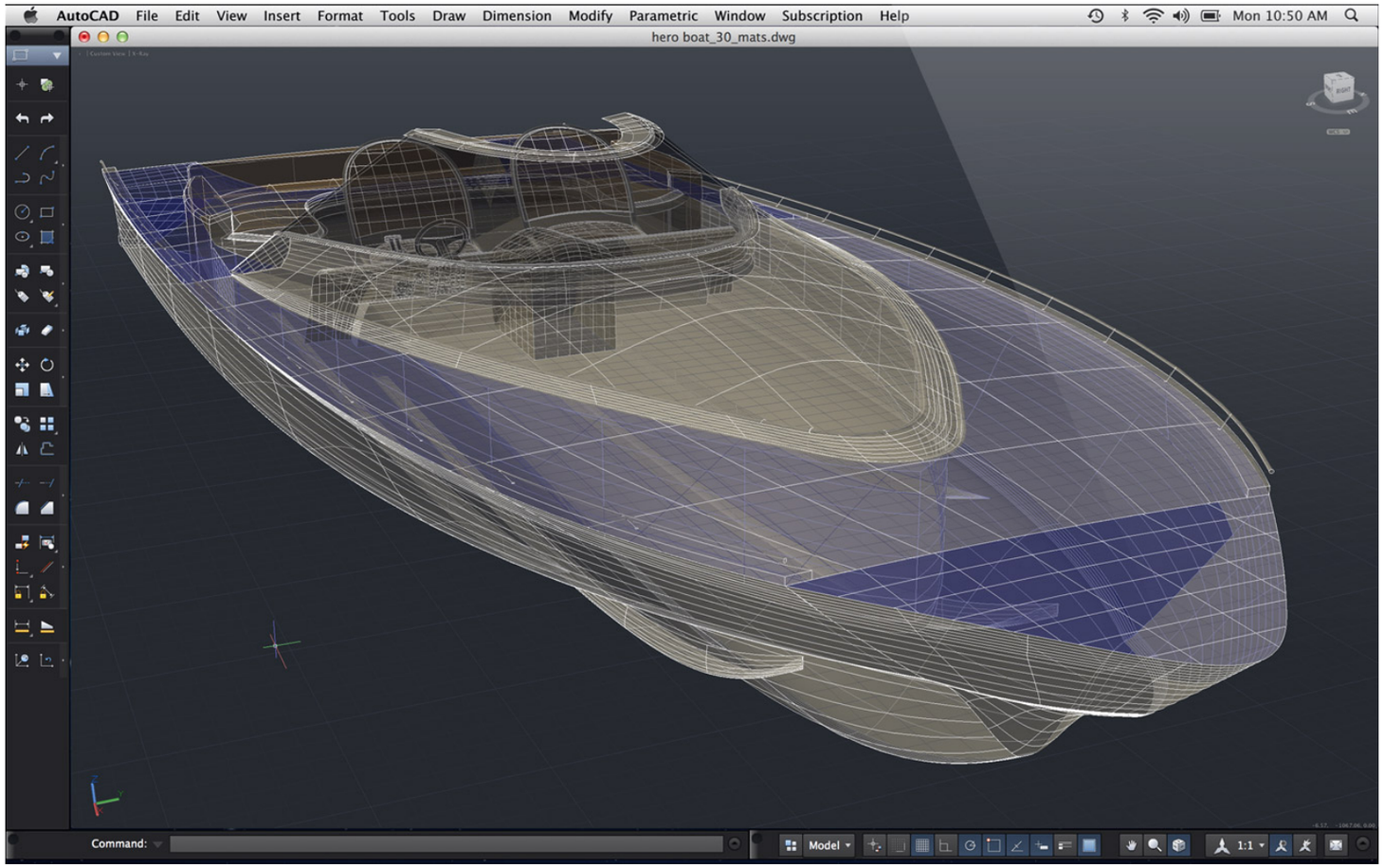
metodi e tecnologie per elaborare dati geometrici tridimensionali allo scopo di eseguire calcoli e produrre visualizzazioni (2D rendering, 3D printing, …)

1. modeling — acquisizione/definizione dati
2. animation — elaborazione/modifica dati
3. rendering — visualizzazione

3D Computer Graphics

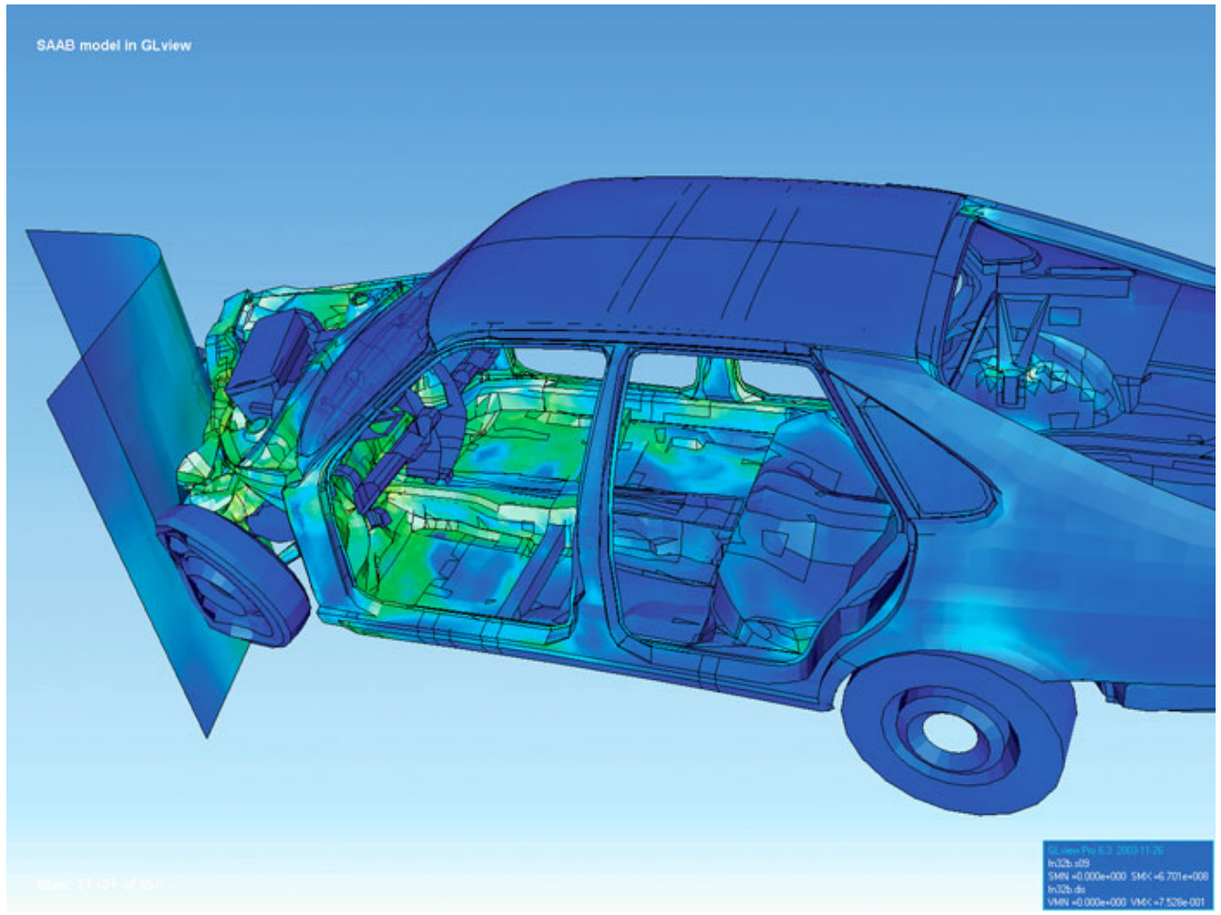
3D Computer Graphics

- progettazione industriale (CAD, …)



3D Computer Graphics

- progettazione industriale (CAD, …)
- simulazione fisica computerizzata



3D Computer Graphics

- progettazione industriale (CAD, …)
- simulazione fisica computerizzata
- videogiochi

3D Computer Graphics

- progettazione industriale (CAD, …)
- simulazione fisica computerizzata
- videogiochi
- animazione ed effetti visivi

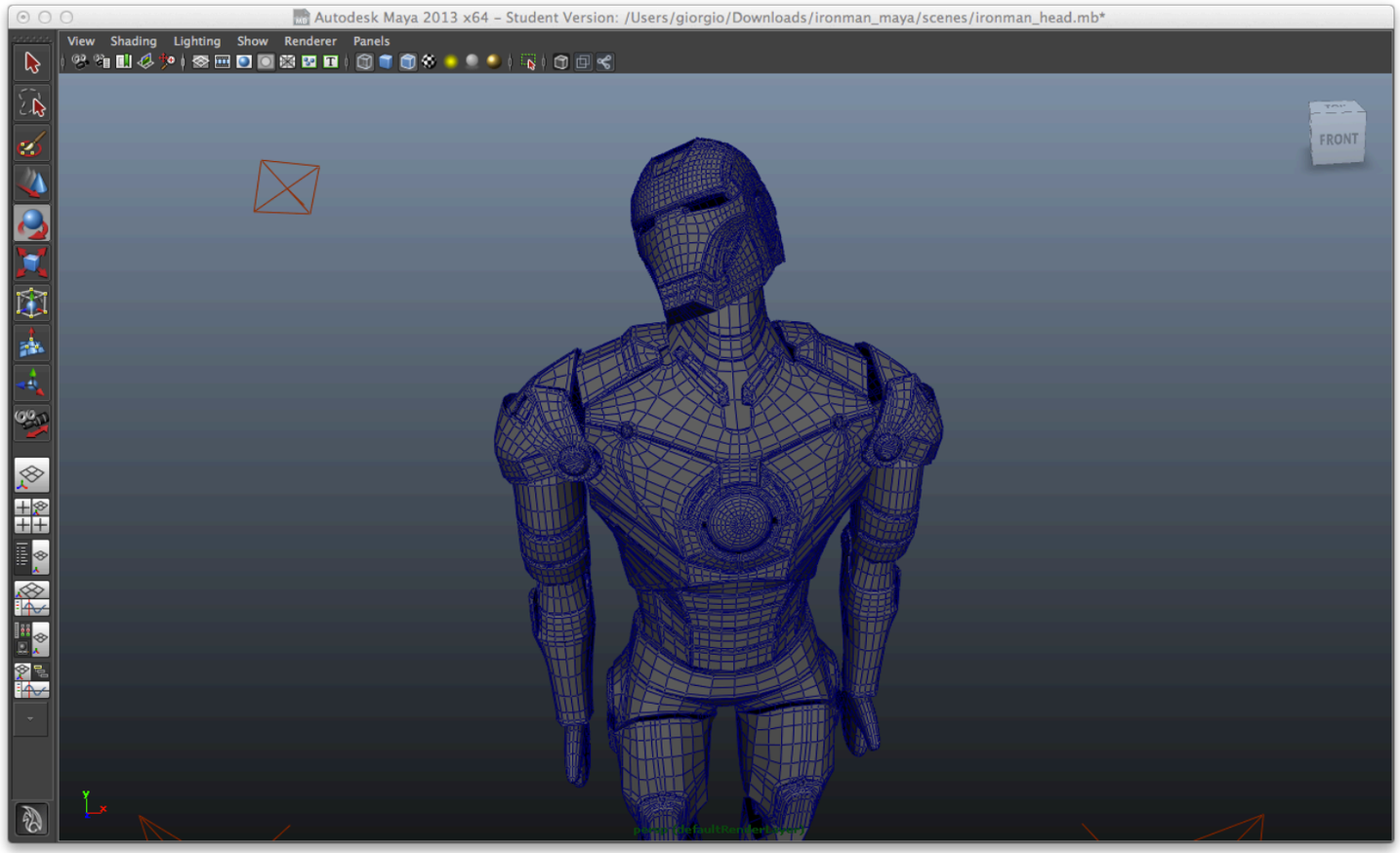
Modeling

il processo di definizione e sviluppo della rappresentazione matematica della geometria di un modello 3D

Modeling

il processo di definizione e sviluppo della rappresentazione matematica della geometria di un modello 3D

1. “a mano”
   tramite software interattivi (es. Maya, Blender, …)

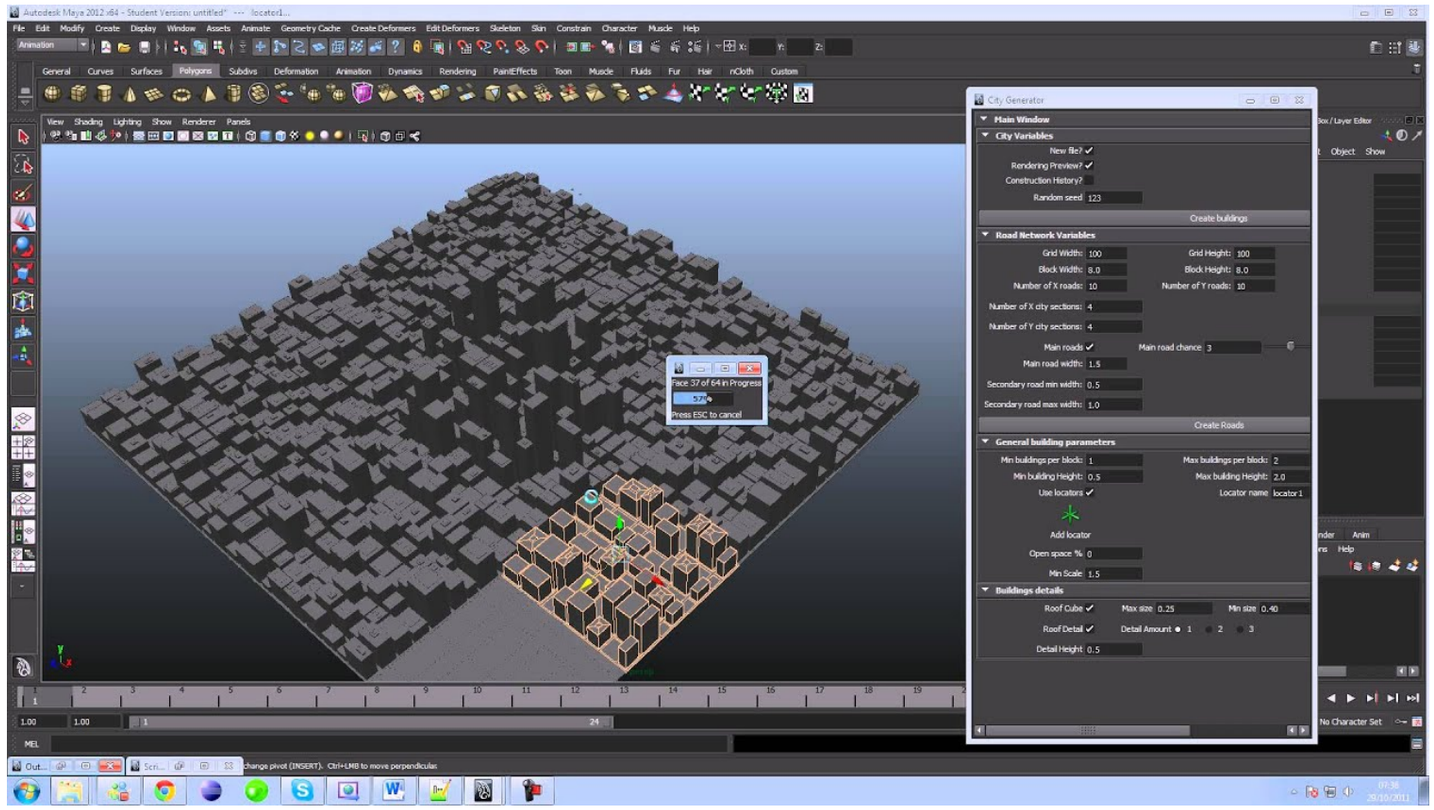


Modeling

il processo di definizione e sviluppo della rappresentazione matematica della geometria di un modello 3D

1. “a mano”
   tramite software interattivi (es. Maya, Blender, …)

2. procedurale
   definizione algoritmica



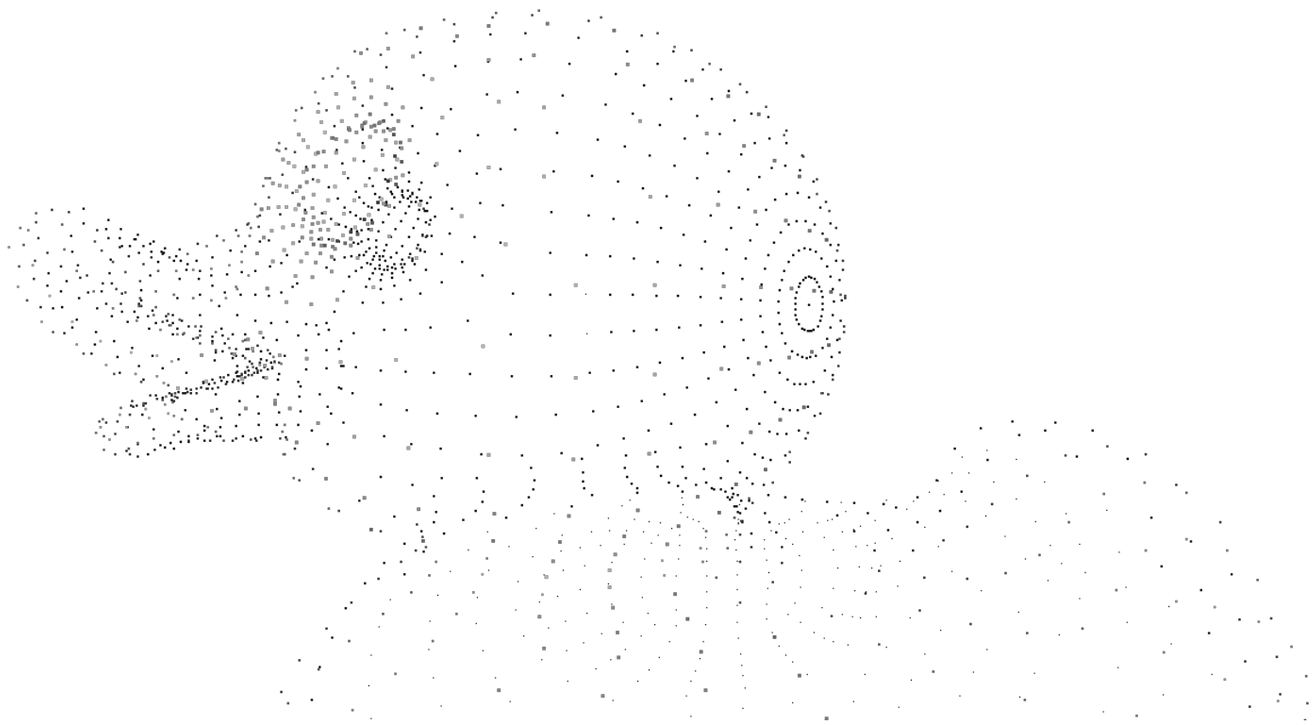
Modeling

il processo di definizione e sviluppo della rappresentazione matematica della geometria di un modello 3D

1. “a mano”
   tramite software interattivi (es. Maya, Blender, …)

2. procedurale
   definizione algoritmica

3. acquisizione
   via scanning



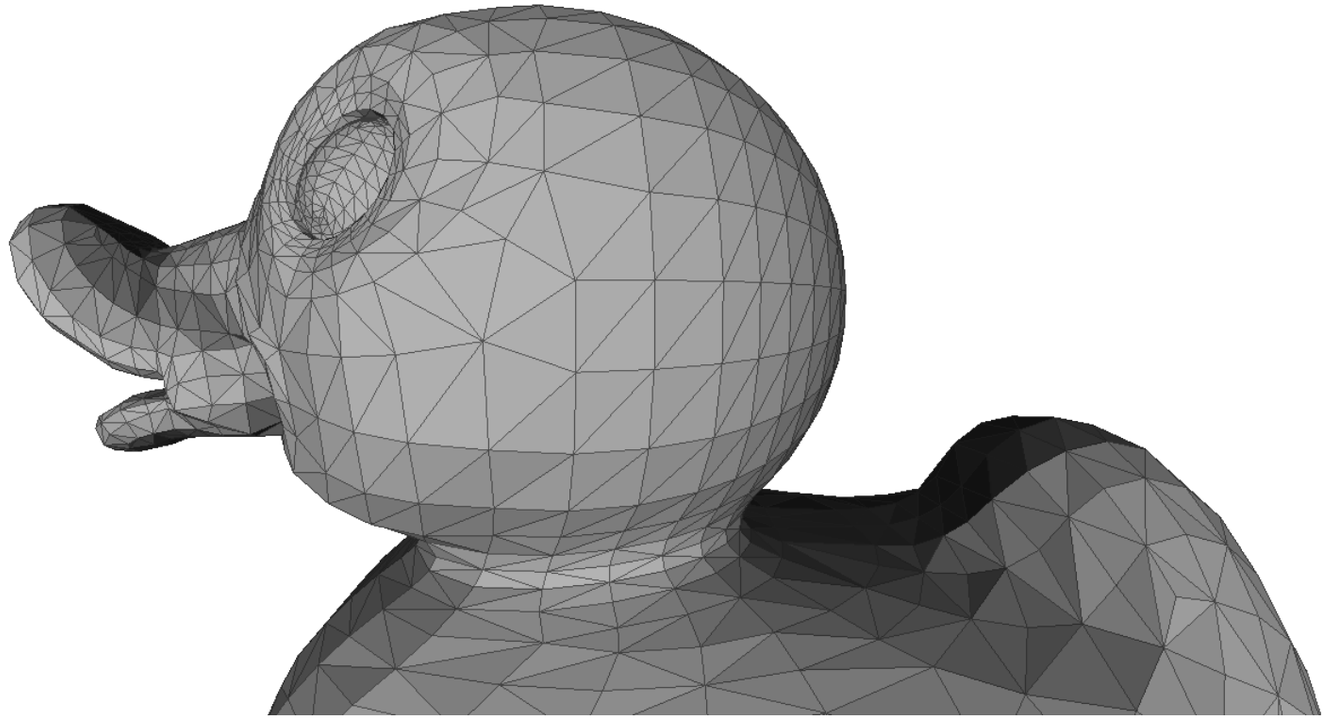
Modeling

il processo di definizione e sviluppo della rappresentazione matematica della geometria di un modello 3D

1. “a mano”
   tramite software interattivi (es. Maya, Blender, …)

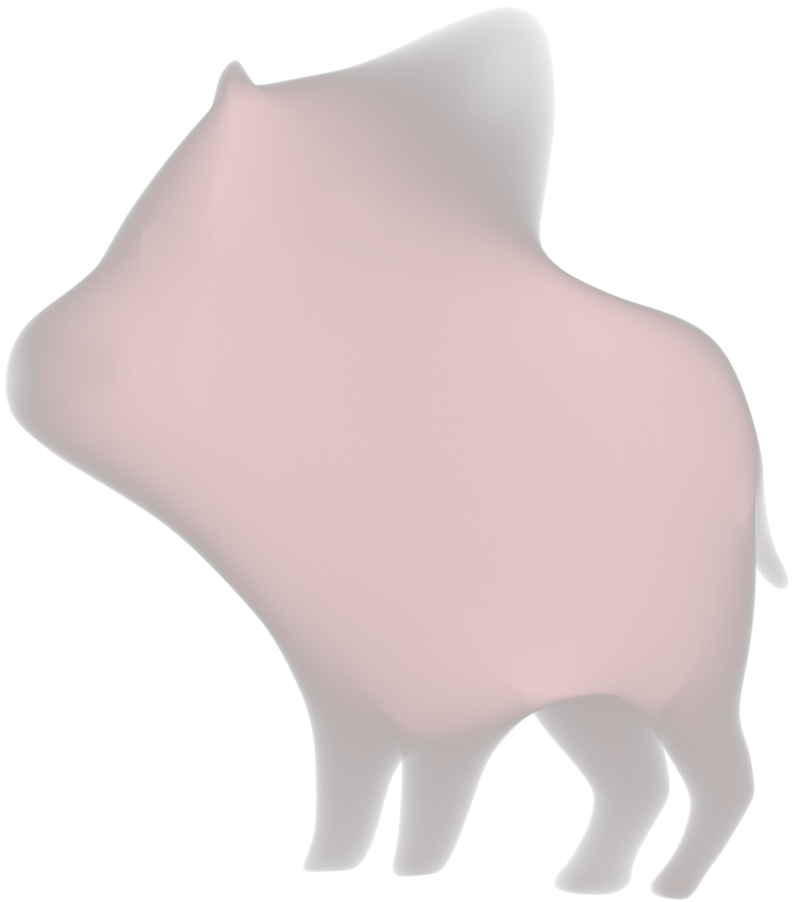
2. procedurale
   definizione algoritmica

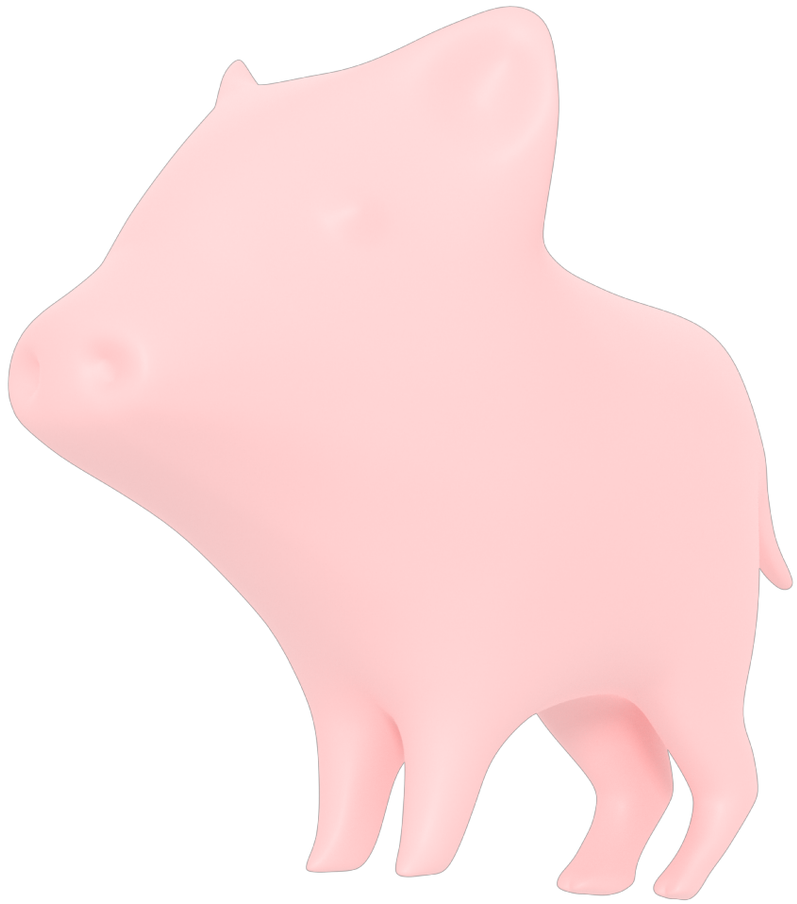
3. acquisizione
   via scanning



Il Modello 3D

come si rappresenta?
volume vs superficie



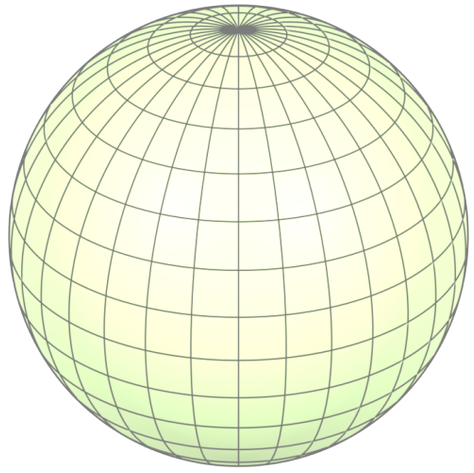


Superfici

parametriche *vs* implicite

$$f : \mathbf{D} \subset \mathbb{R}^2 \mapsto \mathbf{I} \subset \mathbb{R}^3 \qquad F(x, y, z) = 0$$

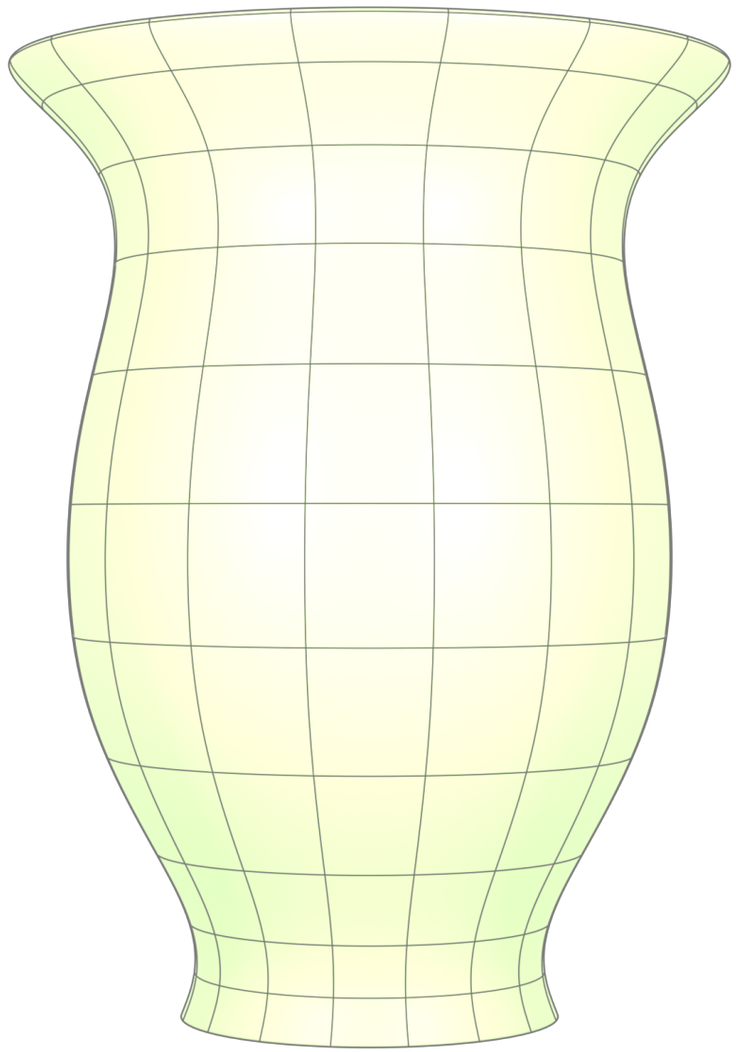
$$f(\alpha, \beta) = \begin{pmatrix} sin(\beta)cos(\alpha) \\ sin(\beta)sin(\alpha) \\ cos(\beta) \end{pmatrix} \qquad x^2 + y^2 + z^2 - 1 = 0$$



Superfici a tratti

- difficile rappresentare superfici complesse con singole funzioni matematiche

- campionamento

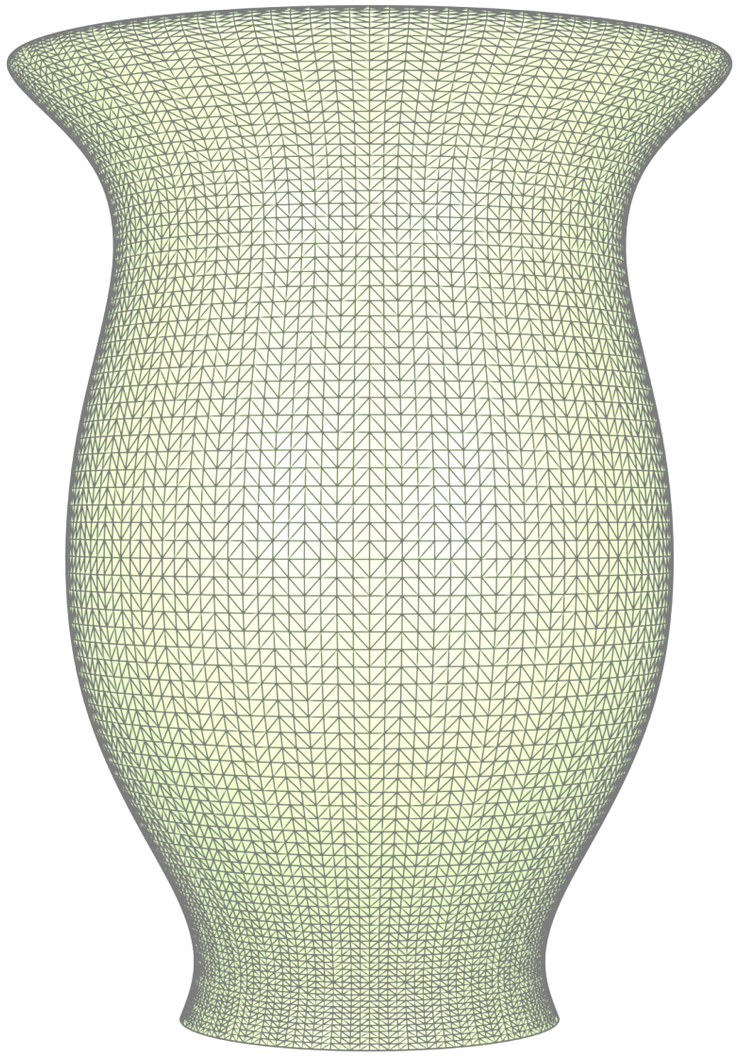


giorgio marcias - giorgio@develer.com

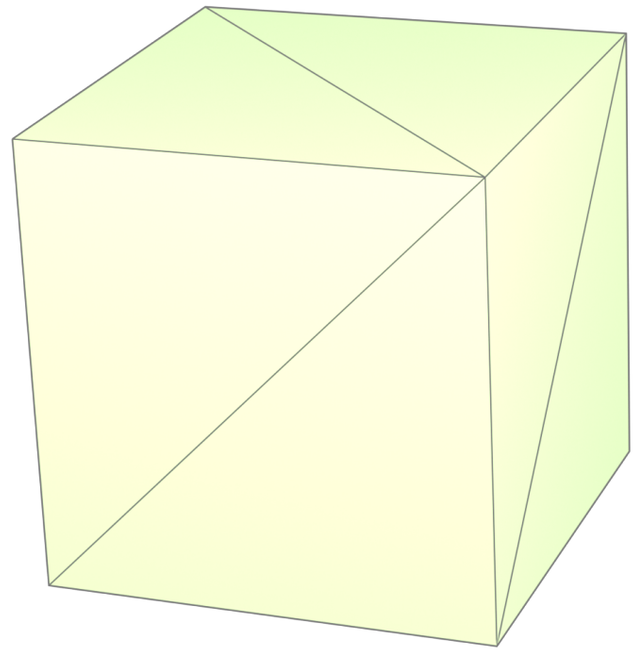
Superfici a tratti

- una superficie complessa è suddivisa in porzioni di superfici più semplici, unite ai bordi (mesh)

- es. porzioni di piani (mesh poligonali)
  - caso particolare: triangoli

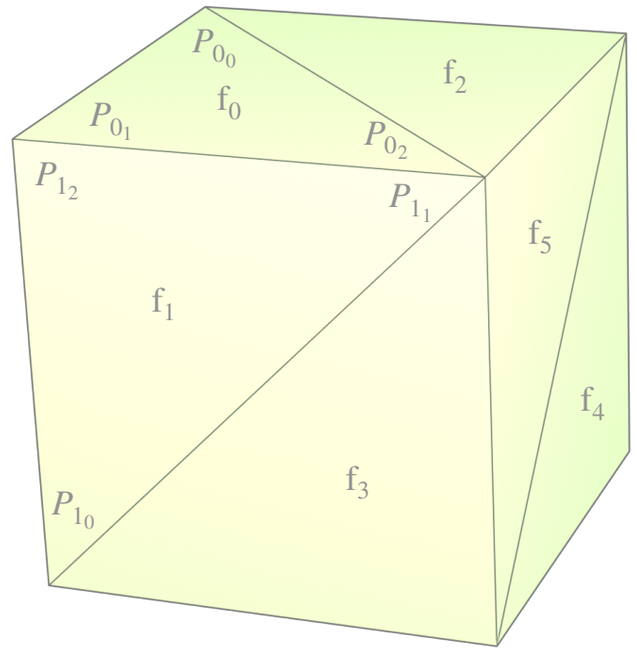


Mesh di triangoli: struttura dati semplice

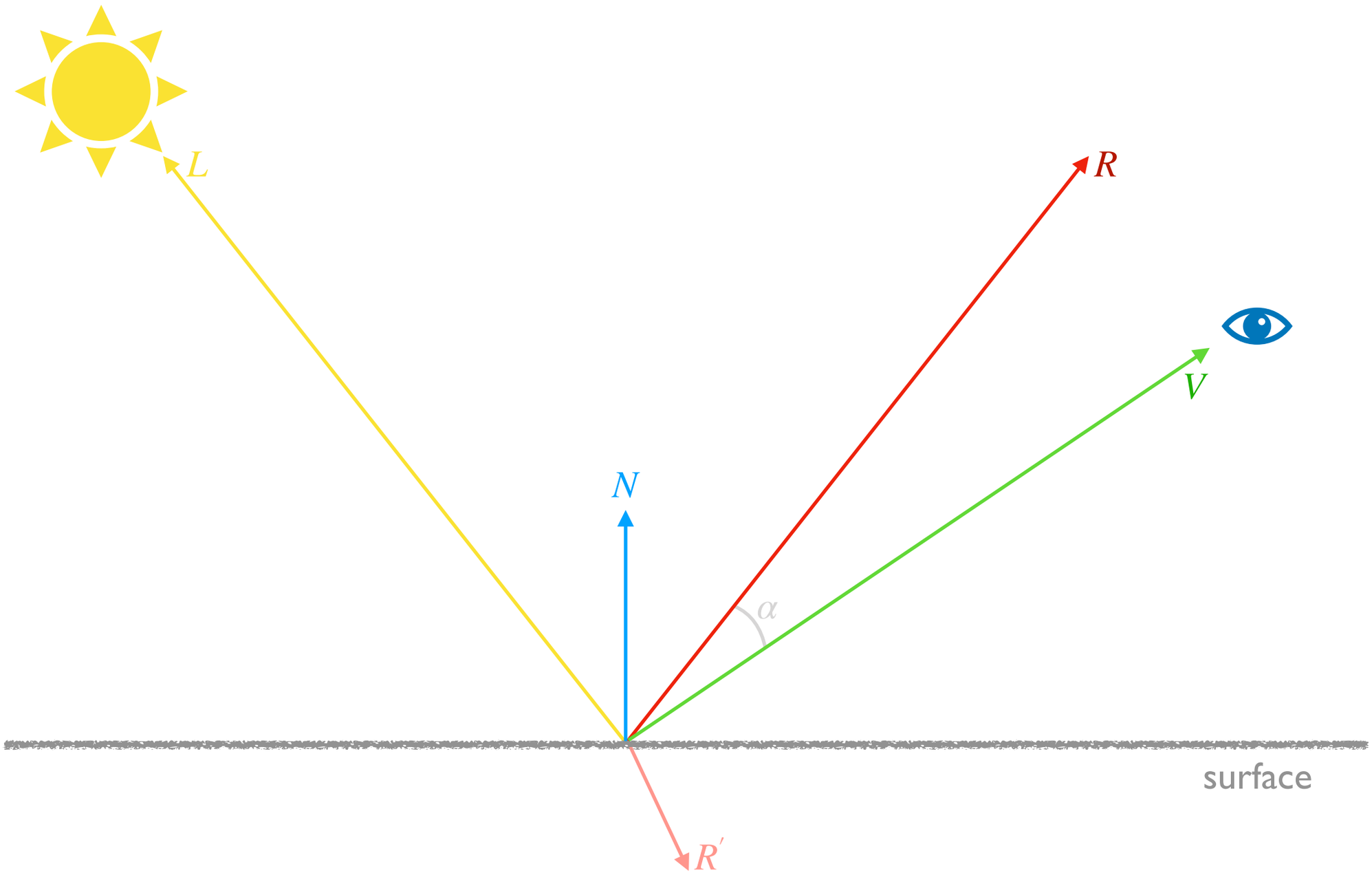


Mesh di triangoli: struttura dati semplice

face 0	$P_{0_0}\begin{pmatrix}x_{0_0}\\y_{0_0}\\z_{0_0}\end{pmatrix}$	$P_{0_1}\begin{pmatrix}x_{0_1}\\y_{0_1}\\z_{0_1}\end{pmatrix}$	$P_{0_2}\begin{pmatrix}x_{0_2}\\y_{0_2}\\z_{0_2}\end{pmatrix}$
face 1	$P_{1_0}\begin{pmatrix}x_{1_0}\\y_{1_0}\\z_{1_0}\end{pmatrix}$	$P_{1_1}\begin{pmatrix}x_{1_1}\\y_{1_1}\\z_{1_1}\end{pmatrix}$	$P_{1_2}\begin{pmatrix}x_{1_2}\\y_{1_2}\\z_{1_2}\end{pmatrix}$
face 2	$P_{2_0}\begin{pmatrix}x_{2_0}\\y_{2_0}\\z_{2_0}\end{pmatrix}$	$P_{2_1}\begin{pmatrix}x_{2_1}\\y_{2_1}\\z_{2_1}\end{pmatrix}$	$P_{2_2}\begin{pmatrix}x_{2_2}\\y_{2_2}\\z_{2_2}\end{pmatrix}$
face 3	$P_{3_0}\begin{pmatrix}x_{3_0}\\y_{3_0}\\z_{3_0}\end{pmatrix}$	$P_{3_1}\begin{pmatrix}x_{3_1}\\y_{3_1}\\z_{3_1}\end{pmatrix}$	$P_{3_2}\begin{pmatrix}x_{3_2}\\y_{3_2}\\z_{3_2}\end{pmatrix}$
...			



Illuminazione



Richiami di Geometria

- somma di vettori

$$\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_2 = \begin{pmatrix} x_1 + x_2 \\ y_1 + y_2 \\ z_1 + z_2 \end{pmatrix}$$

- prodotto tra vettore e scalare

$$\mathbf{v} = a\mathbf{v}_1 = \begin{pmatrix} a x_1 \\ a y_1 \\ a z_1 \end{pmatrix}$$

- norma di un vettore

$$l = \|\mathbf{v}_1\| = \sqrt{x_1^2 + y_1^2 + z_1^2}$$

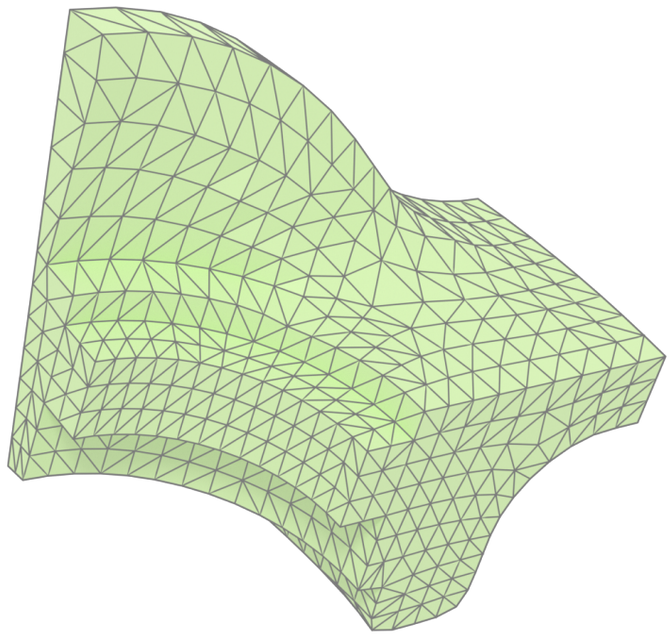
- prodotto scalare

$$\begin{aligned} a &= \mathbf{v}_1 \cdot \mathbf{v}_2 \\ &= x_1 * x_2 + y_1 * y_2 + z_1 * z_2 \\ &= \|\mathbf{v}_1\| \|\mathbf{v}_2\| \cos(\mathbf{v}_1 \angle \mathbf{v}_2) \end{aligned}$$

- prodotto vettoriale

$$\begin{aligned} \mathbf{v} &= \mathbf{v}_1 \times \mathbf{v}_2 \qquad \mathbf{v} \perp (\mathbf{v}_1, \mathbf{v}_2) \\ \|\mathbf{v}_1 \times \mathbf{v}_2\| &= \|\mathbf{v}_1\| \|\mathbf{v}_2\| \sin(\mathbf{v}_1 \angle \mathbf{v}_2) \end{aligned}$$

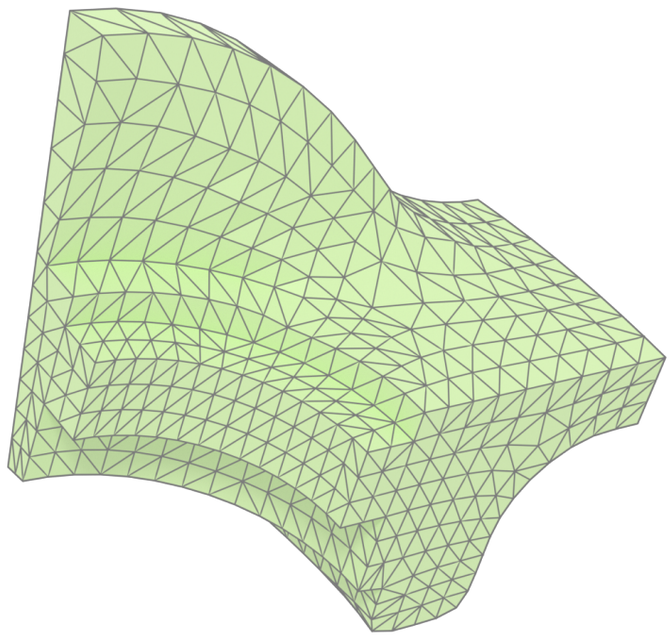
Superfici: shading



giorgio marcias - giorgio@develer.com

Superfici: shading

- flat shading



giorgio marcias - giorgio@develer.com

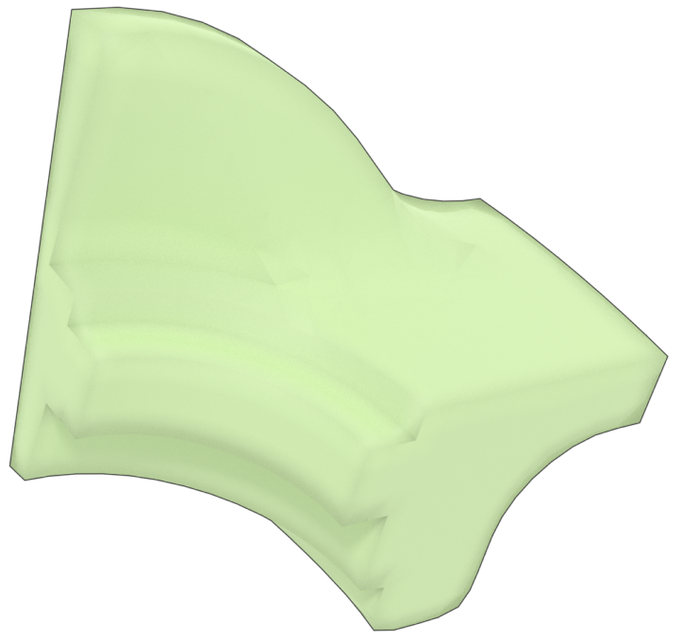
Superfici: shading

- flat shading

giorgio marcias - giorgio@develer.com

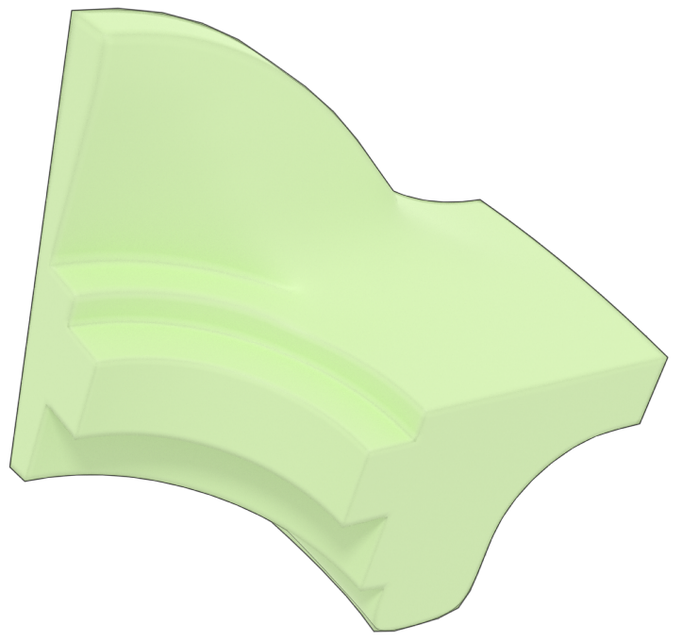
Superfici: shading

- flat shading
- smooth shading

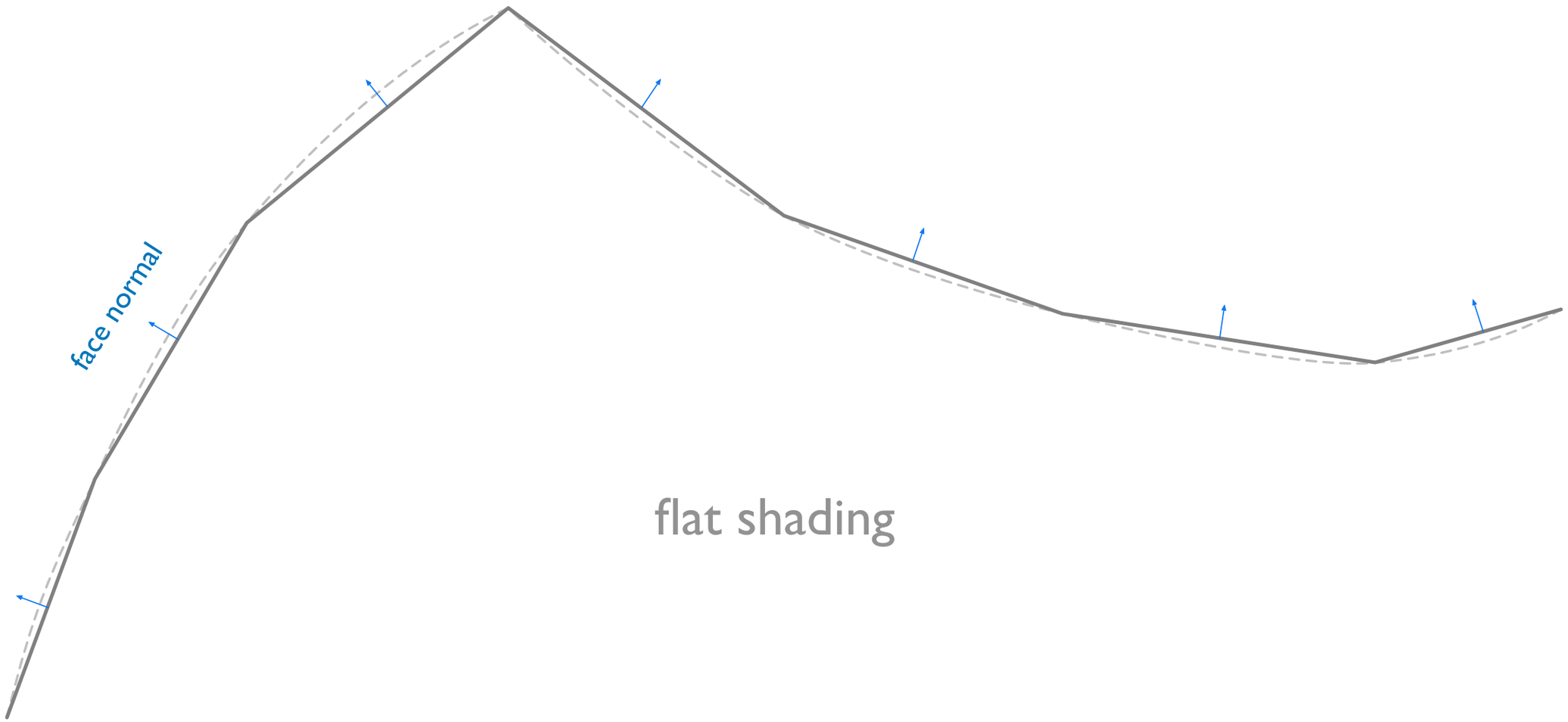


Superfici: shading

- flat shading
- smooth shading
- smooth + spigoli



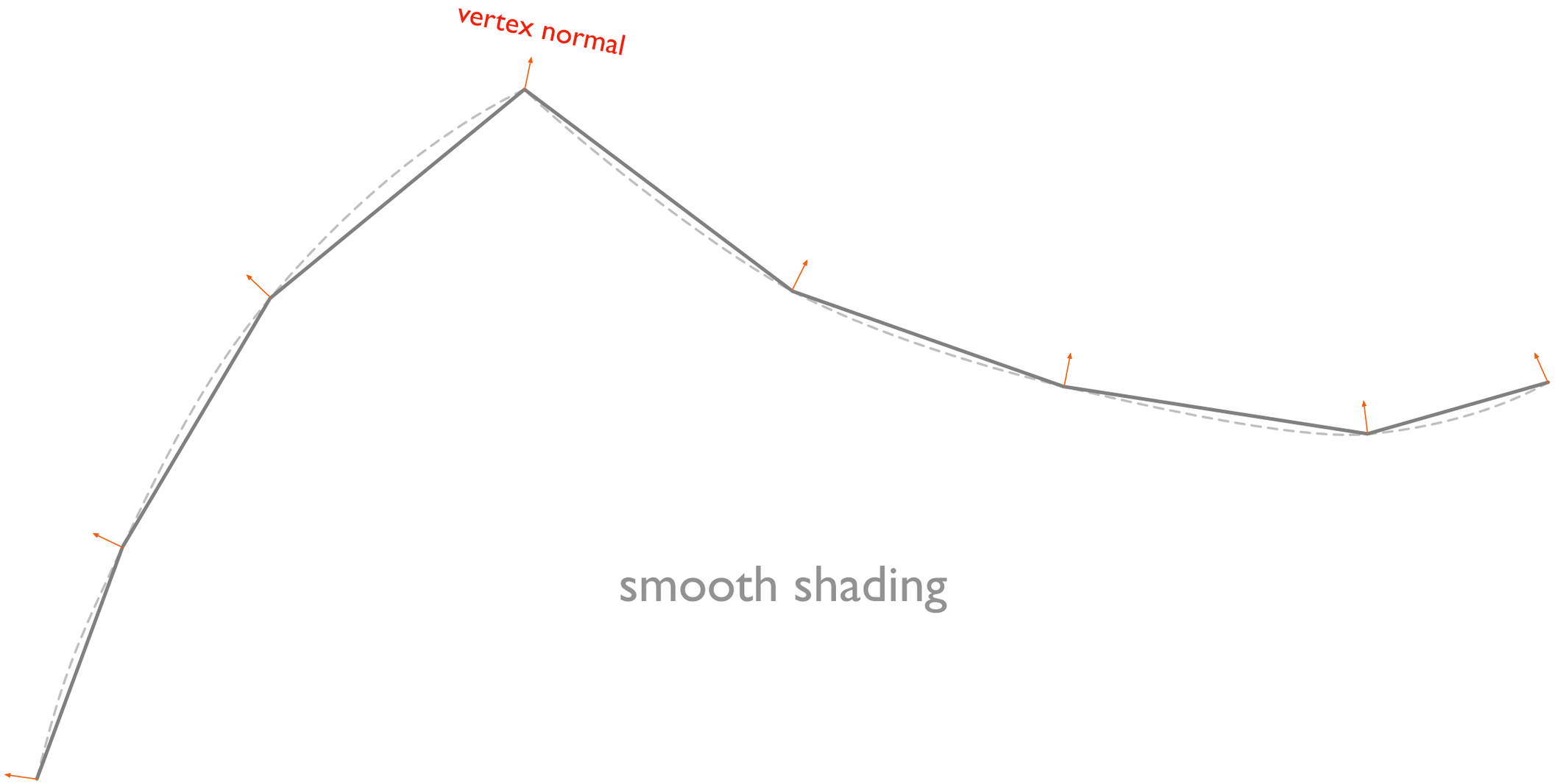
Surface: shading



flat shading

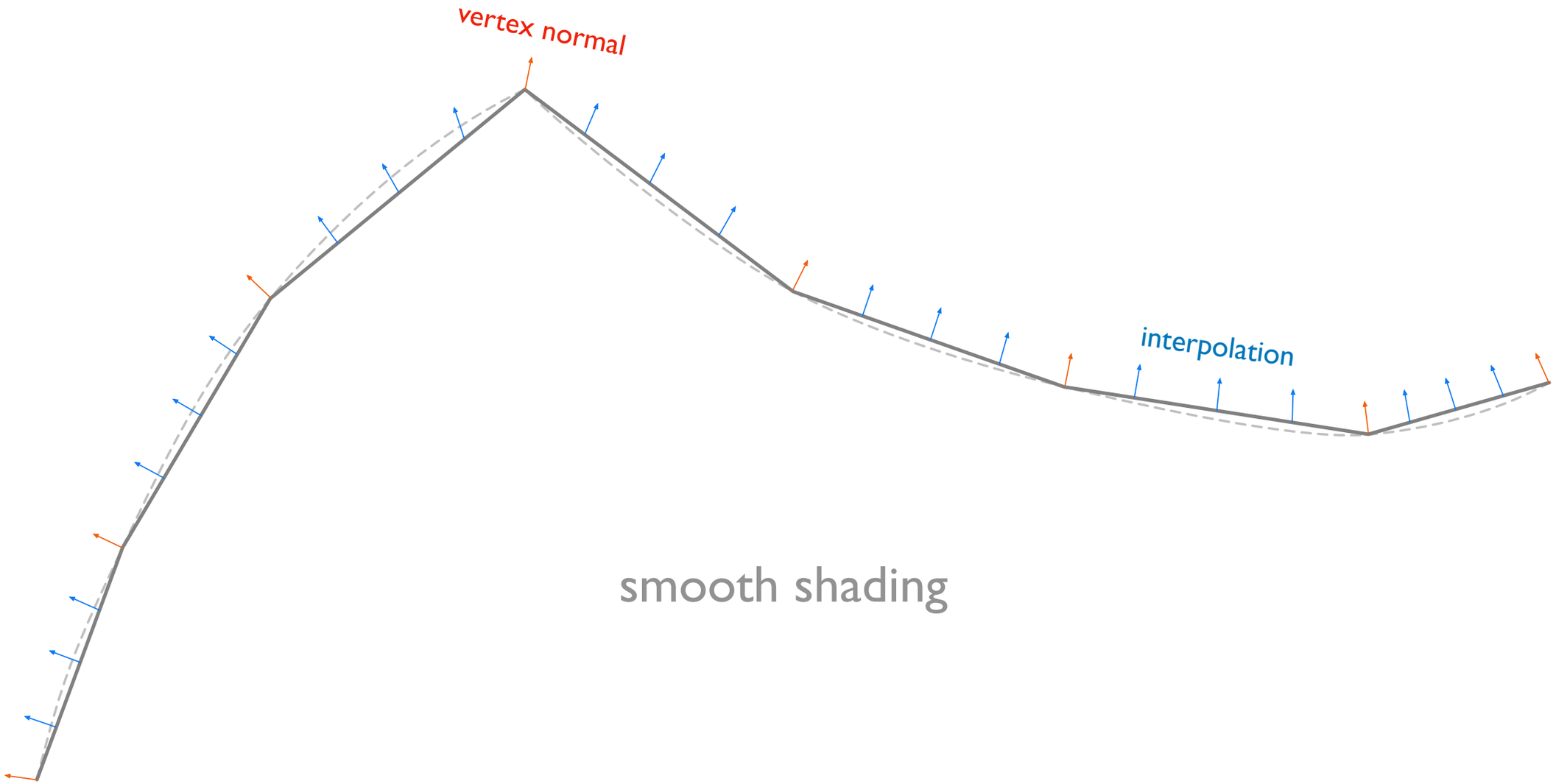
giorgio marcias - giorgio@develer.com

Surface: shading



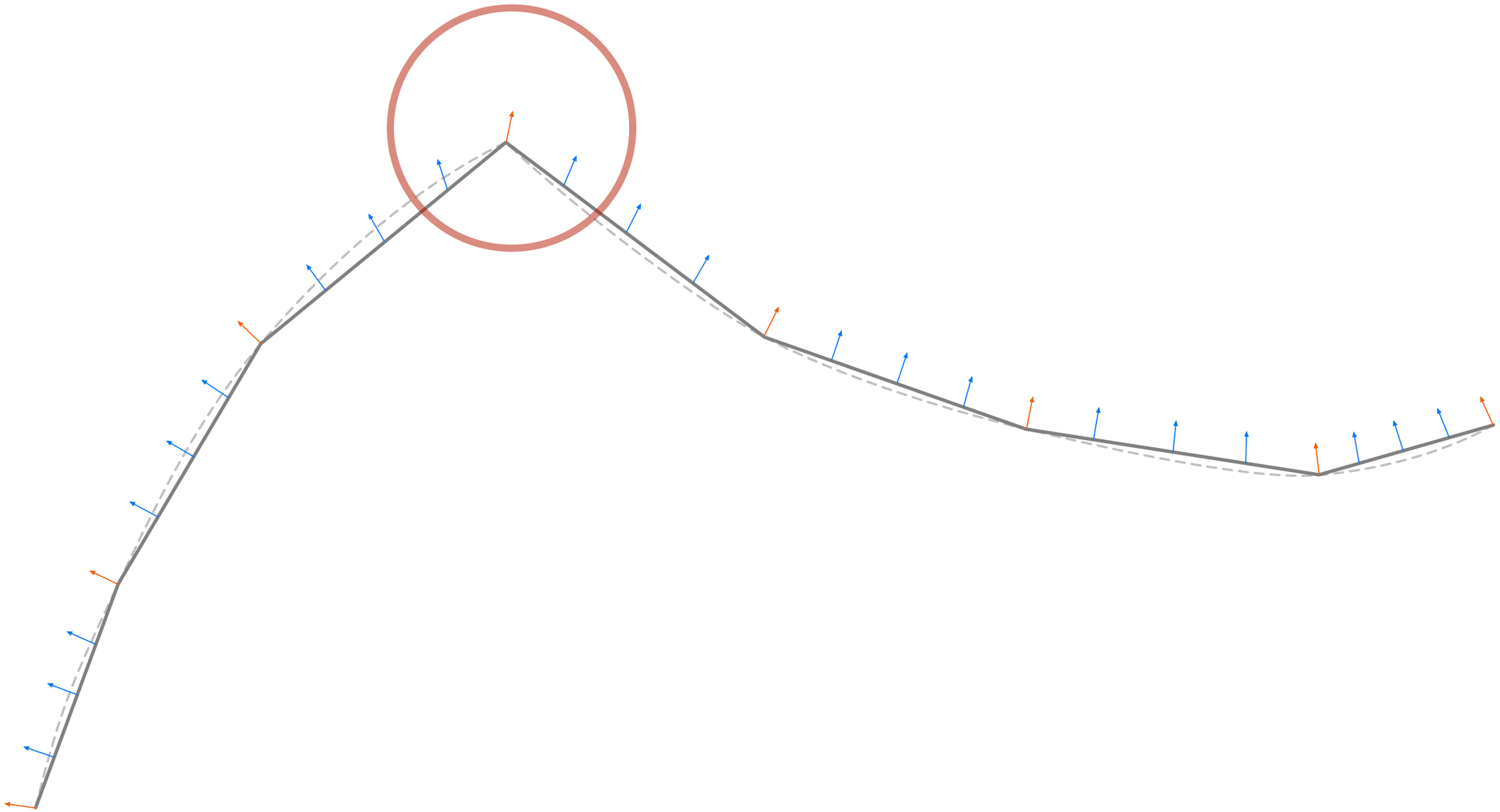
smooth shading

Surface: shading



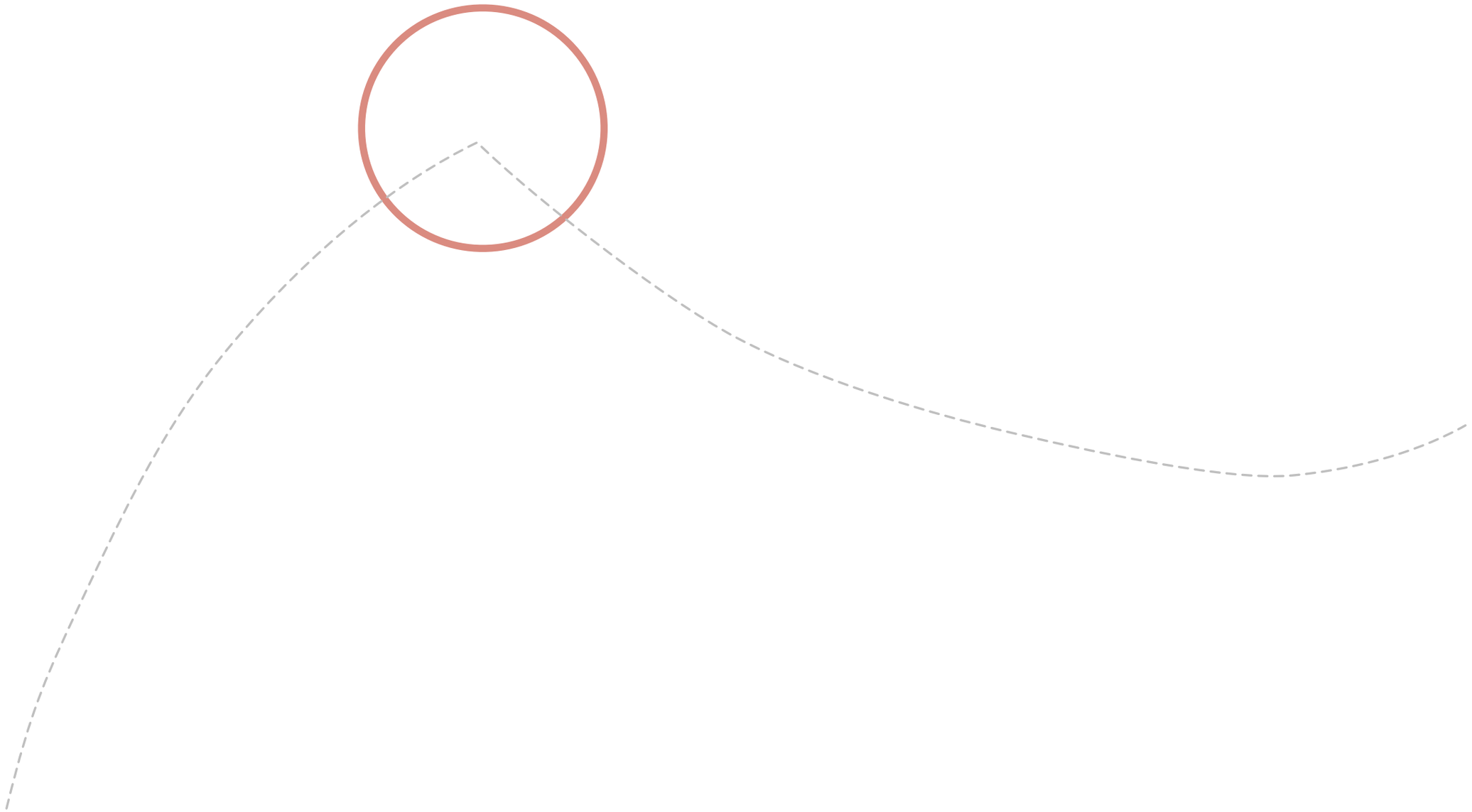
smooth shading

Surface: shading



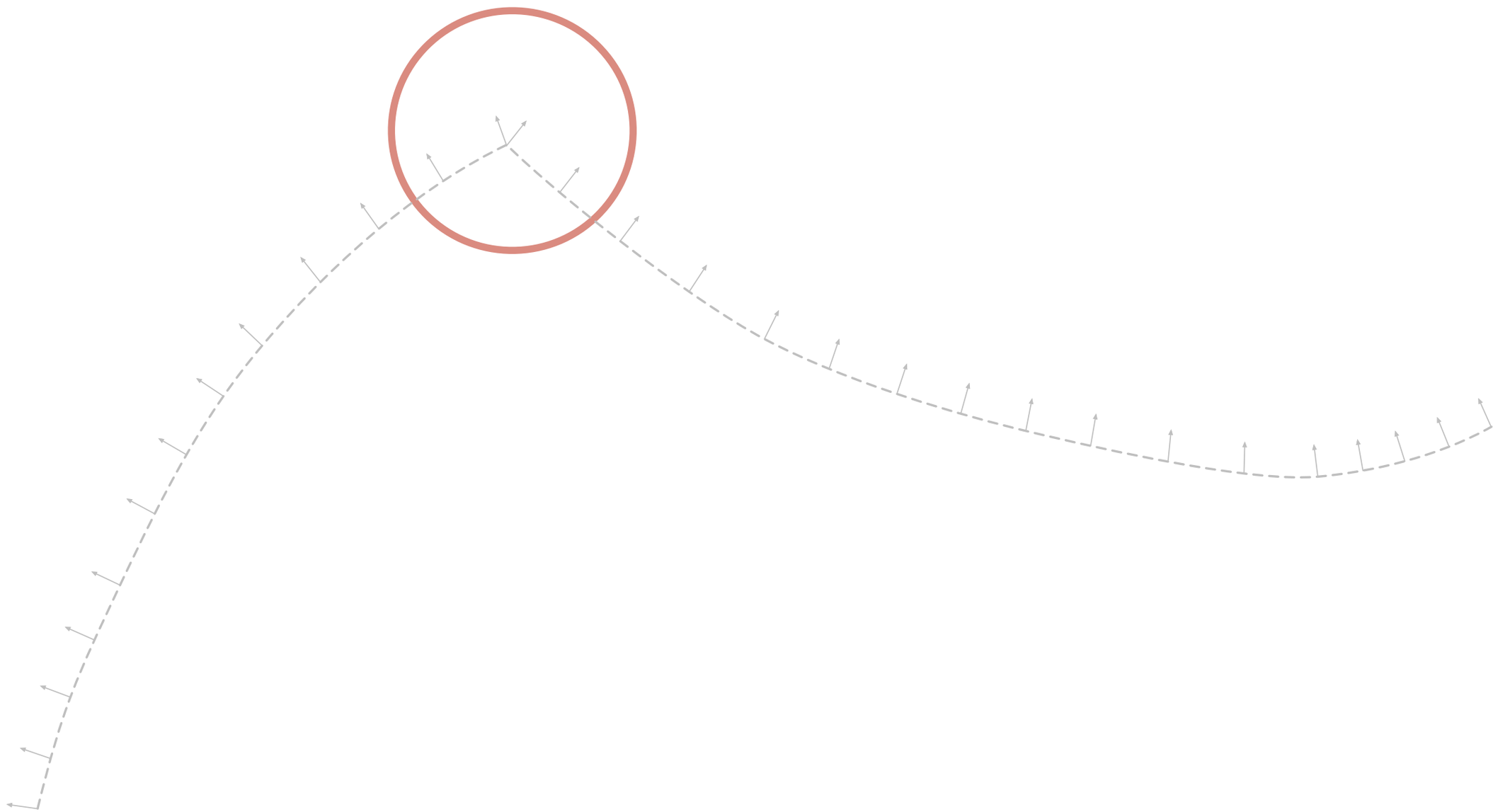
giorgio marcias - giorgio@develer.com

Surface: shading



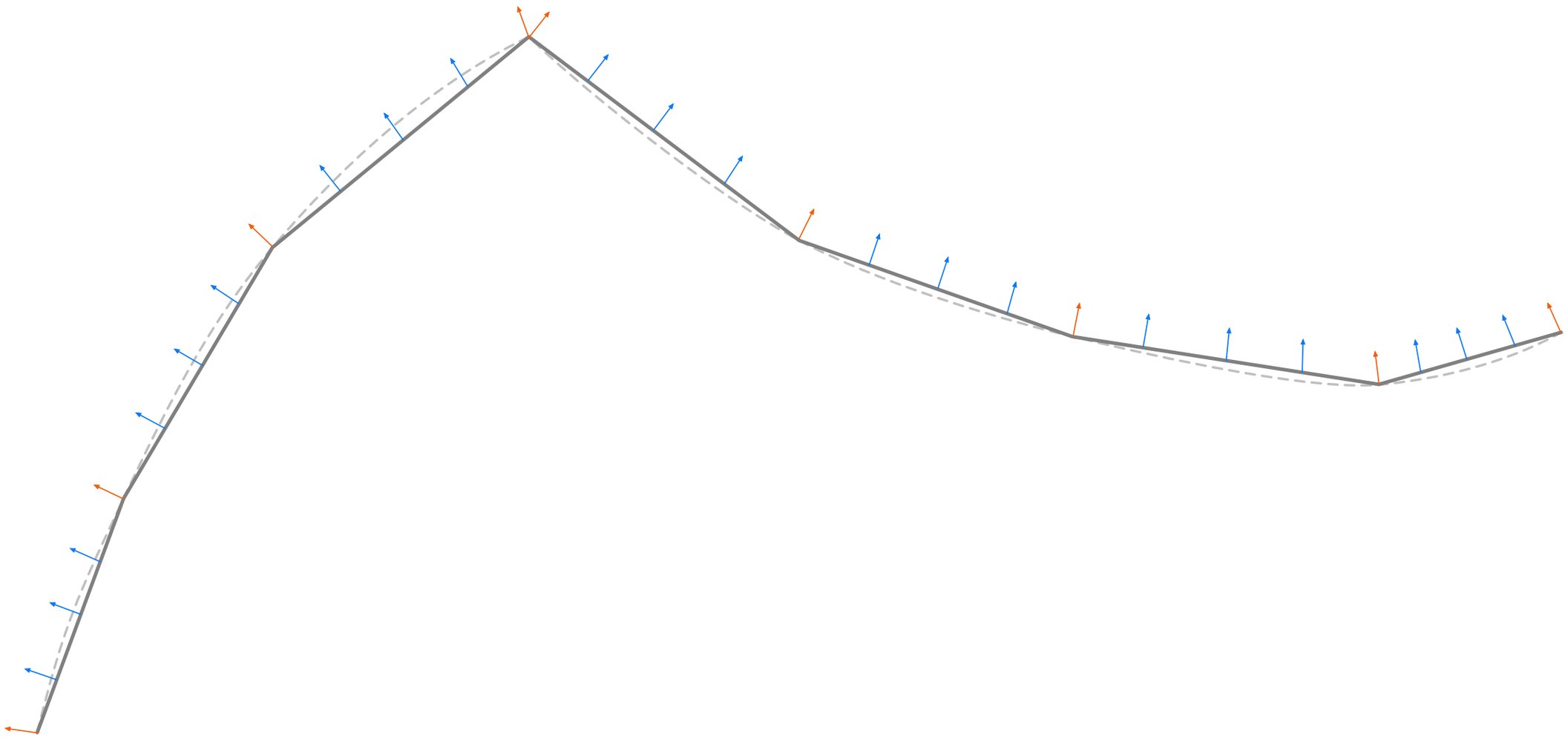
giorgio marcias - giorgio@develer.com

Surface: shading



giorgio marcias - giorgio@develer.com

Surface: shading



giorgio marcias - giorgio@develer.com

Codice e Librerie

- repo: https://github.com/giorgiomarcias/WS_geo_3D

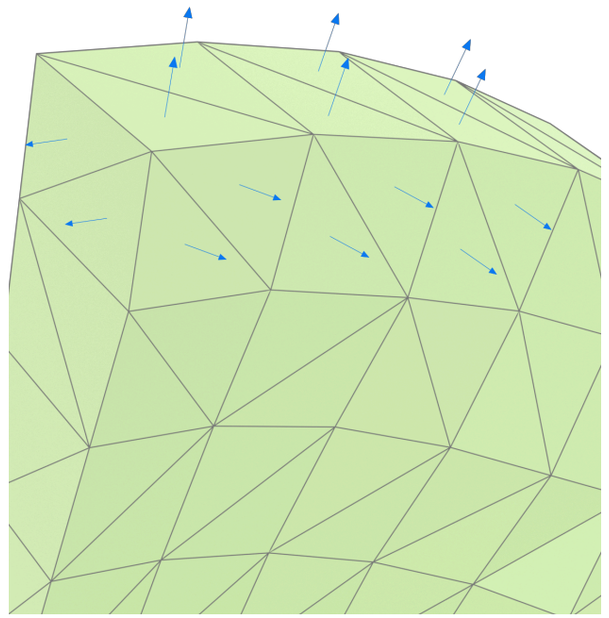
  - ```
    git clone --recurse-submodules https://github.com/giorgiomarcias/WS_geo_3D.git
    cd WS_geo_3D
    mkdir build
    cd build
    cmake ..
    make
    ```

- **libIGL** https://libigl.github.io
  viewer e geometry processing

- **eigenlib** http://eigen.tuxfamily.org
  algebra lineare e calcolo matriciale
    ```

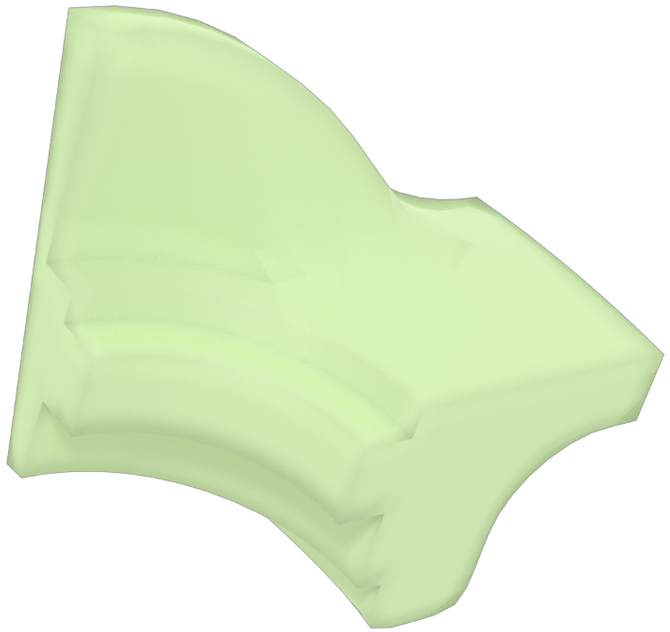
Esercizio 1

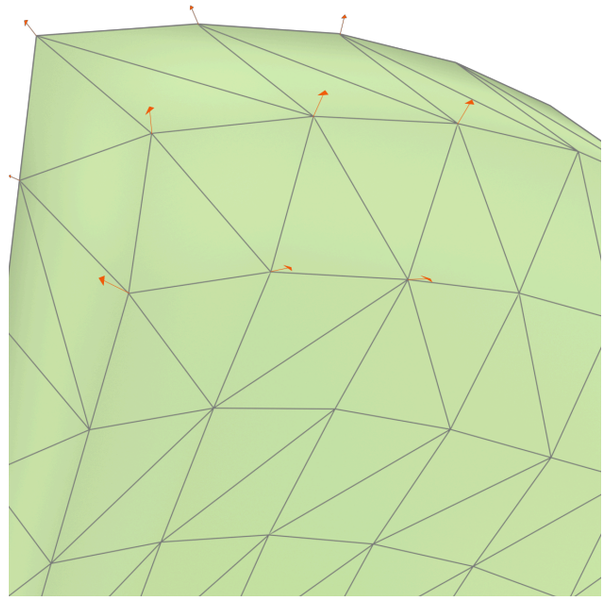
flat shading: calcolare le normali dei triangoli



Esercizio 2

smooth shading: calcolare le normali dei vertici





develer

giorgio marcias - giorgio@develer.com

Esercizio 3

smooth & sharp shading: calcolare le normali dei vertici e dei triangoli