# Geometric Modeling

2015

Tutorial: Assignment 2 (prac. & theo.)

TU Delft

# Exam

## Exam

- When: Friday, 26th June 2015 14:00-16:00h
- Where: EEMCS Building, Lecture Hall F

# Assignments

## Hand-in of the Practical Assignment 2

- Upload your one .zip file containing your program, source and documentation on Tuesday, 9th June to blackboard groups file exchange (or send it via email)
- Wednesday, 10th June:
  - Grading in personal interviews
  - 20 min slots
  - Group must *show up entirely*
    - Only for the 20min, not the whole time
  - Everybody is graded individually, based on:
    - The group's implementation
    - Personal knowledge about the implementation
    - Everybody must be able to *explain all of the code*

# Assignments

## Schedule: Hand-in Practical Assignment 2

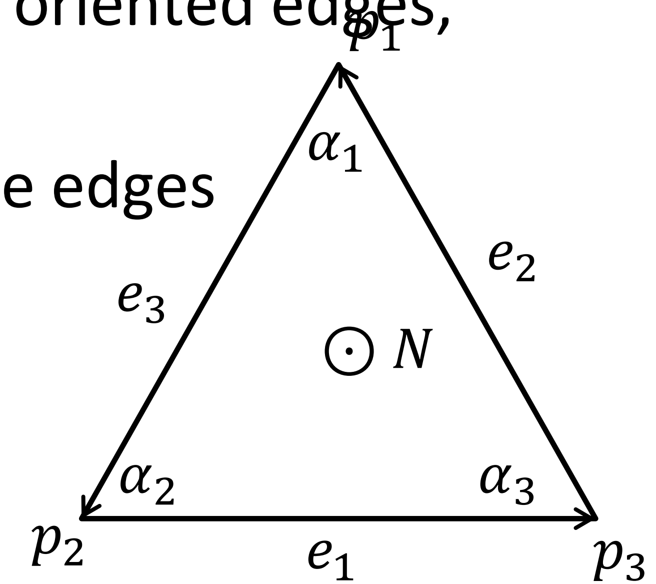
- You only need to be present for the interview slot of your group
- Location: **EEMCS – HB 11.270 (Klaus), HB 11.120 (Christopher)**
- Date: Wednesday, $10^{th}$ June

| Time | Klaus | Christopher |
|---|---|---|
| 15:45-16:05 | Group 3 | Group 1 |
| 16:10-16:30 | Group 6 | Group 5 |
| 16:35-16:55 | Group 7/8 | Group 9 |
| 17:00-17:20 | Group 10 | |
| 17:25-17:45 | Group 11 | |

# Hints: Task 1

## Reminder: Gradient matrix of a triangle

- Notation: $N$ normal of the triangle, $e_i$ oriented edges, $u_i$ function values at the vertices
- The orientations of the normal and the edges has to be consistent




$$\nabla u(p) = \frac{1}{2\text{area}(T)} \sum_{i=1}^{3} u_i \, N \times e_i$$

$$\frac{1}{2\text{area}(T)} \begin{pmatrix} (N \times e_1)_x & (N \times e_2)_x & (N \times e_3)_x \\ (N \times e_1)_y & (N \times e_2)_y & (N \times e_3)_y \\ (N \times e_1)_z & (N \times e_2)_z & (N \times e_3)_z \end{pmatrix}$$

# Hints: Task1

## Gradient matrix of a mesh

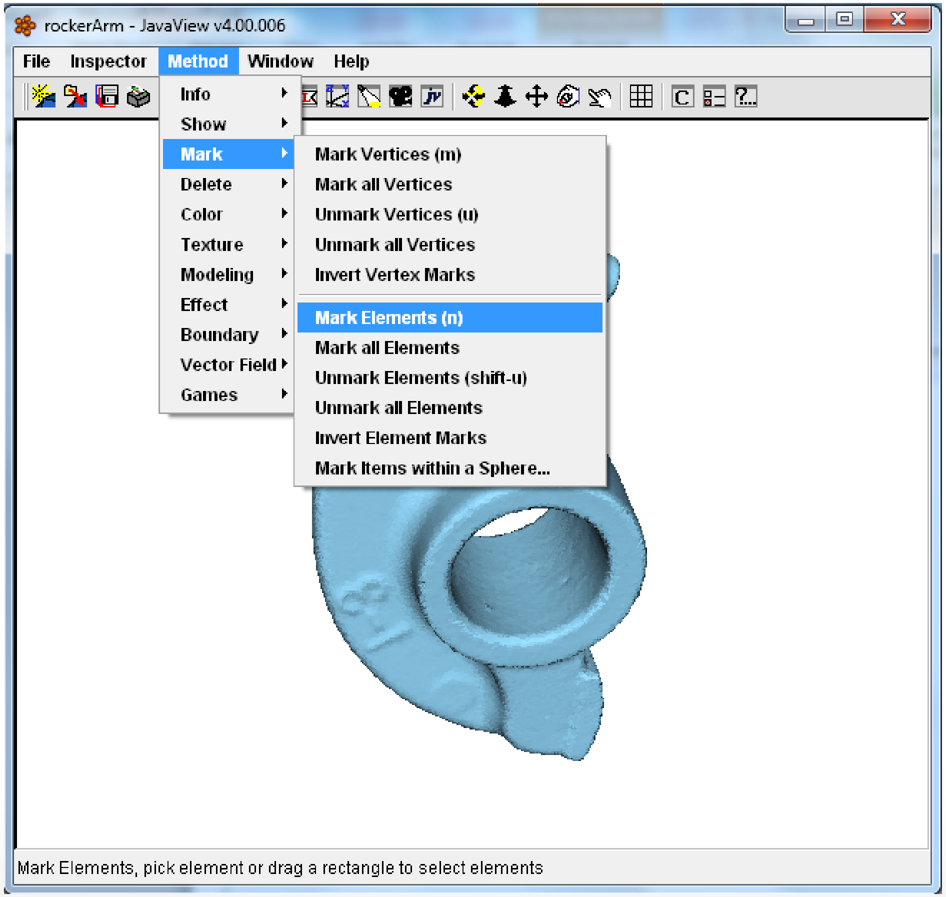
- All vertices have indices
- Choose an order for the gradients and their coordinates
  - For example: list x,y,z coordinates of all triangles in the order of the triangle indices
- For every triangle compute the 3x3 matrix and sort the entries to the corresponding locations in the "big" matrix

Test your matrix before using it for Task 2!

# Hints: Task 2

## Select triangles in JavaView

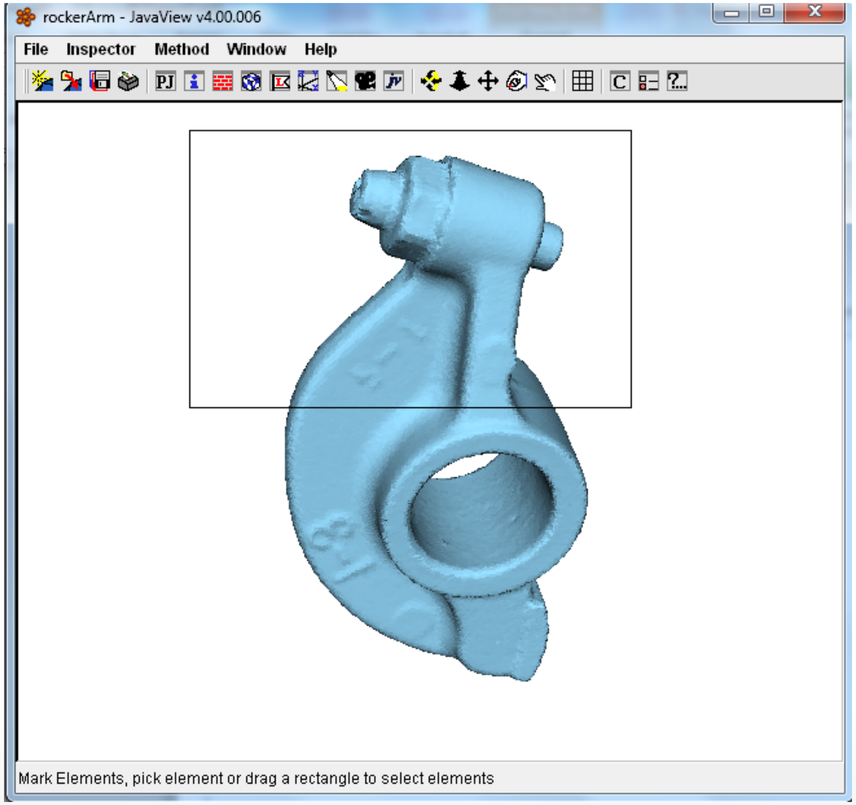
- Toggle on triangle selection mode (or keep `n` pressed)

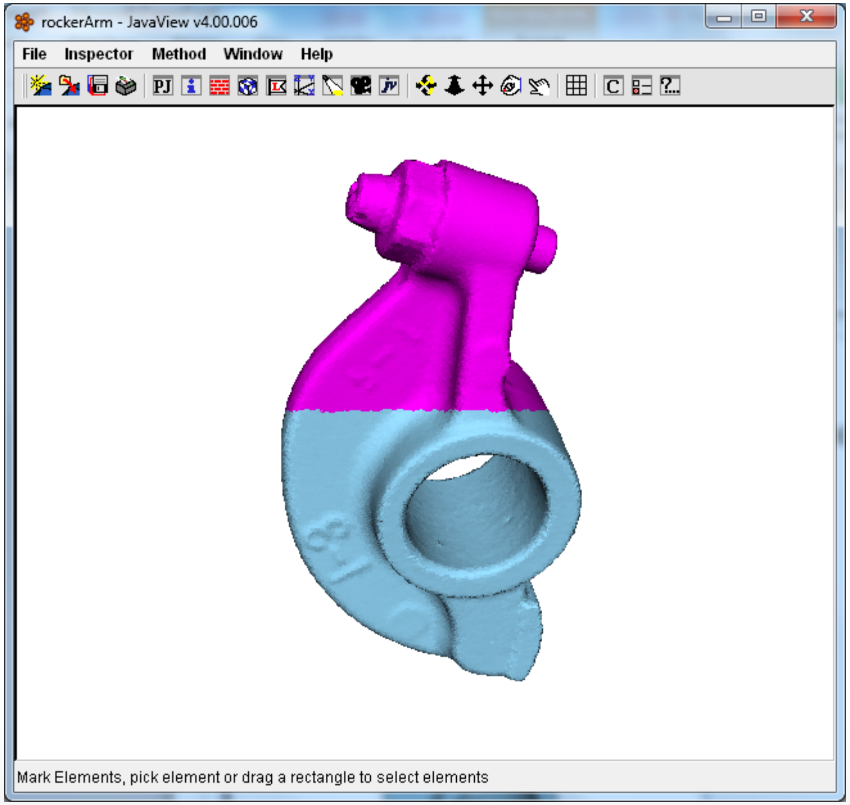


# Hints: Task 2

## Select triangles in JavaView

- Drag rectangle with the mouse and release mouse button



# Hints: Task 2

**Check if element 17 is selected**

```
PiVector element = geom.getElement(17);
Element.hasTag(PsObject.IS_SELECTED);
```

The last line returns a boolean, which has the value `true` if the element is selected

# Hints: Task 2

## User interface (one possibility)

- Text fields to specify 3x3 matrix
- Button compute deformed surface
- Button reset

- Interesting matrices include:
  - Multiples of the identity (scaling)
  - Diagonal matrices (with positive entries) anisotropic scaling in the directions of the globale coordinate axes
  - Rotations
  - ...

# Hints: Task 2

## Overview

- Set up gradient matrix (see Task 1)
- Compute gradients of embedding (vector listing the vertex coord.)
  - You get three gradients per triangle (one for each coordinate function (x,y,z-coordinates of vertices))
- Triangles are selected and a 3x3 matrix is specified by the user
- 3x3 matrix is applied to all three gradient vector of all selected triangles to get the modified gradients$\tilde{g}_x, \tilde{g}_y, \tilde{g}_z$ (3 per triangle)
- The new coordinate functions are computed by solving three linear systems (one for each coordinate)

$$G^T M_V G \tilde{x}_{(x,y,z,)} = G^T M_V \tilde{g}_{(x,y,z)}$$

- Display the deformed surface (which has $\tilde{x}_x, \tilde{x}_y, \tilde{x}_z$ as vertex coordinates)

# Hints: Task 2

## Concerning the linear system to be solved

- The matrix $S = G^T M_V G$ has a kernel (which consists of the constant functions)
- One way to determine these degrees of freedom: After reconstruction move the barycenter to the origin
- If you have troubles solving the linear system, add $\epsilon\, M$ to $S$ (where $\epsilon$ is a small positive real value and $M$ the mass matrix). Then the matrix is positive definite. If $\epsilon$ is small enough this only slightly affects the deformation.
  - What this does is that it adds a penalty for the $L^2$-norm of the embedding of deformed surface