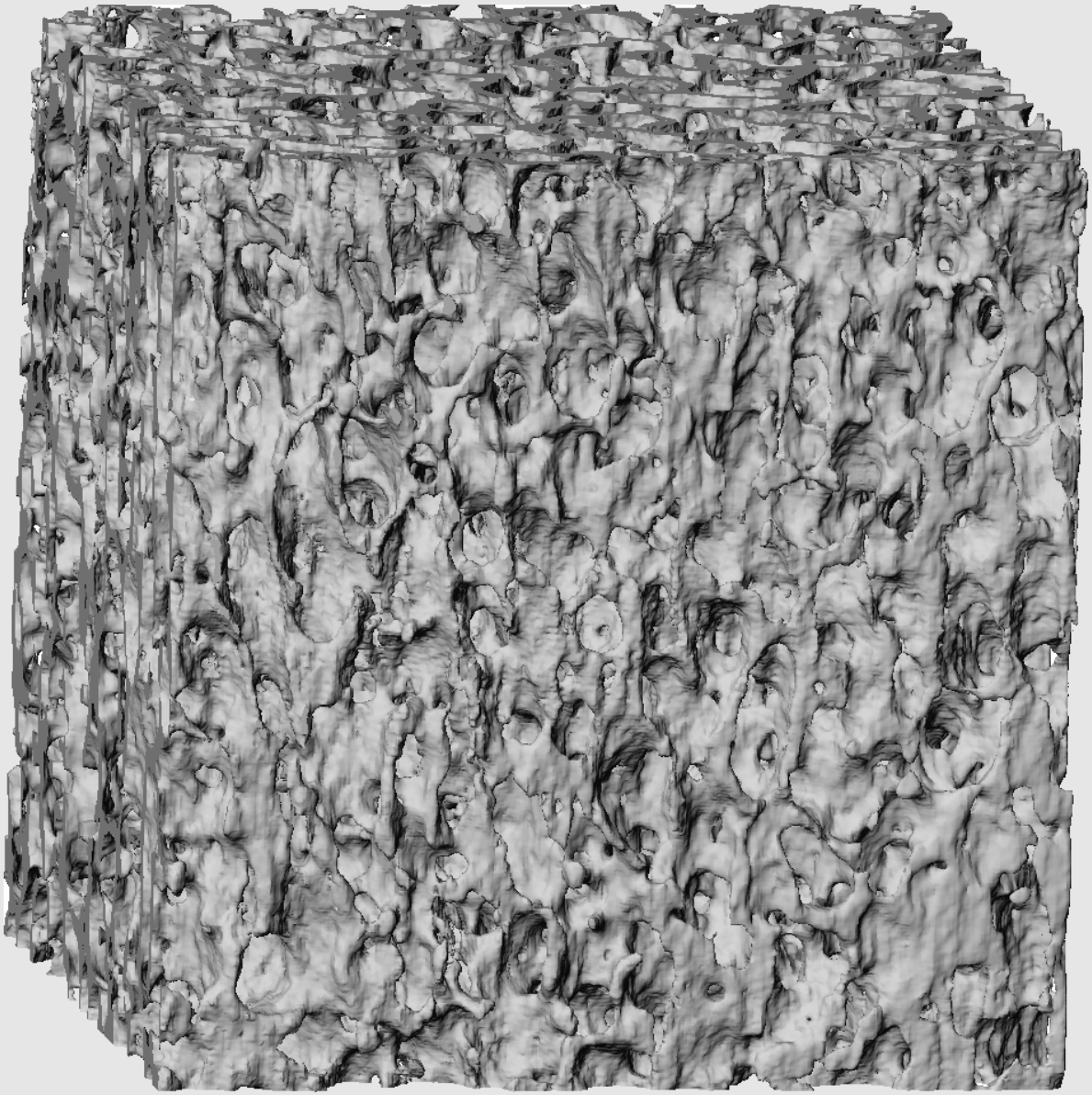
Composite Finite Elements for Efficient Computations on Complicated Domains

Lars Ole Schwen

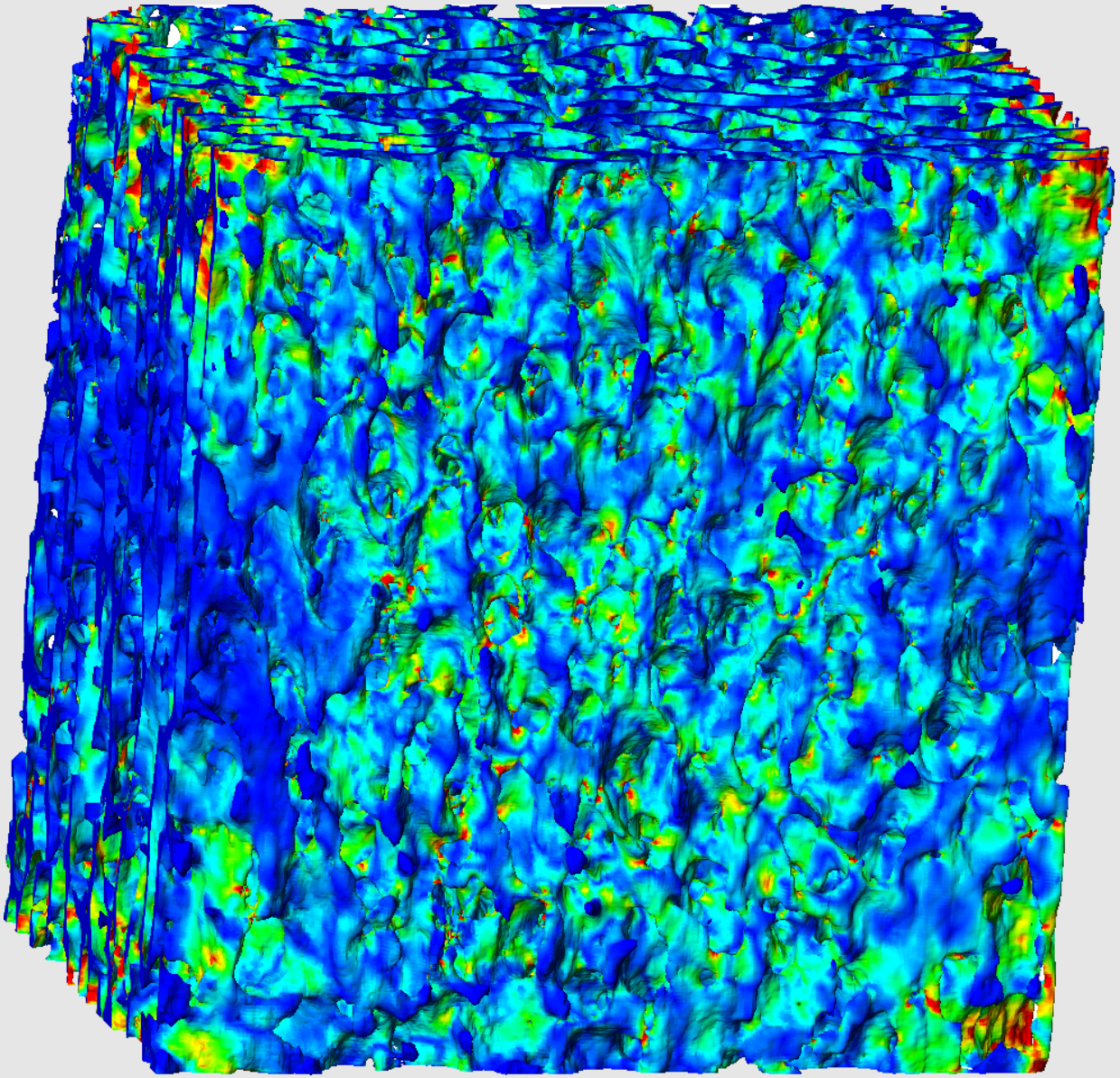
Institute for Numerical Simulation
Hausdorff Center for Mathematics
University of Bonn

DDK, Bonn, 2007–01–23

Goal: Complicated Microstructure



Goal: Complicated Microstructure, Elastically Deformed



Outline

Classical Finite Elements
- Linear Finite Elements in 1D

Finite Elements in 2D
- Classification of Grids
- Structured Grids
- Unstructured Grids
- Comparison

Composite Finite Elements
- Geometry Extraction from Images
- Construction of CFE Grids
- CFE Basis Functions
- Results (Elasticity)

Cooperations and Outlook

Classical Finite Elements
- Linear Finite Elements in 1D

Finite Elements in 2D
- Classification of Grids
- Structured Grids
- Unstructured Grids
- Comparison

Composite Finite Elements
- Geometry Extraction from Images
- Construction of CFE Grids
- CFE Basis Functions
- Results (Elasticity)

Cooperations and Outlook

Main Steps for a Finite Element Computation

1. obtain and preprocess image data: detect geometry
2.
3.
4.
5.

Main Steps for a Finite Element Computation

1. obtain and preprocess image data: detect geometry
2. generate a suitable mesh
3.
4.
5.

Main Steps for a Finite Element Computation

1. obtain and preprocess image data: detect geometry
2. generate a suitable mesh
3. with basis functions on mesh, set up system of equations
4.
5.

Main Steps for a Finite Element Computation

1. obtain and preprocess image data: detect geometry
2. generate a suitable mesh
3. with basis functions on mesh, set up system of equations
4. solve system of equations
5.

Main Steps for a Finite Element Computation

1. obtain and preprocess image data: detect geometry
2. generate a suitable mesh
3. with basis functions on mesh, set up system of equations
4. solve system of equations
5. output result: visualization or other output

Main Steps for a Finite Element Computation

1. obtain and preprocess image data: detect geometry
2. generate a suitable mesh
3. with basis functions on mesh, set up system of equations
4. solve system of equations
5. output result: visualization or other output

We'll focus on 2 to 4 ...

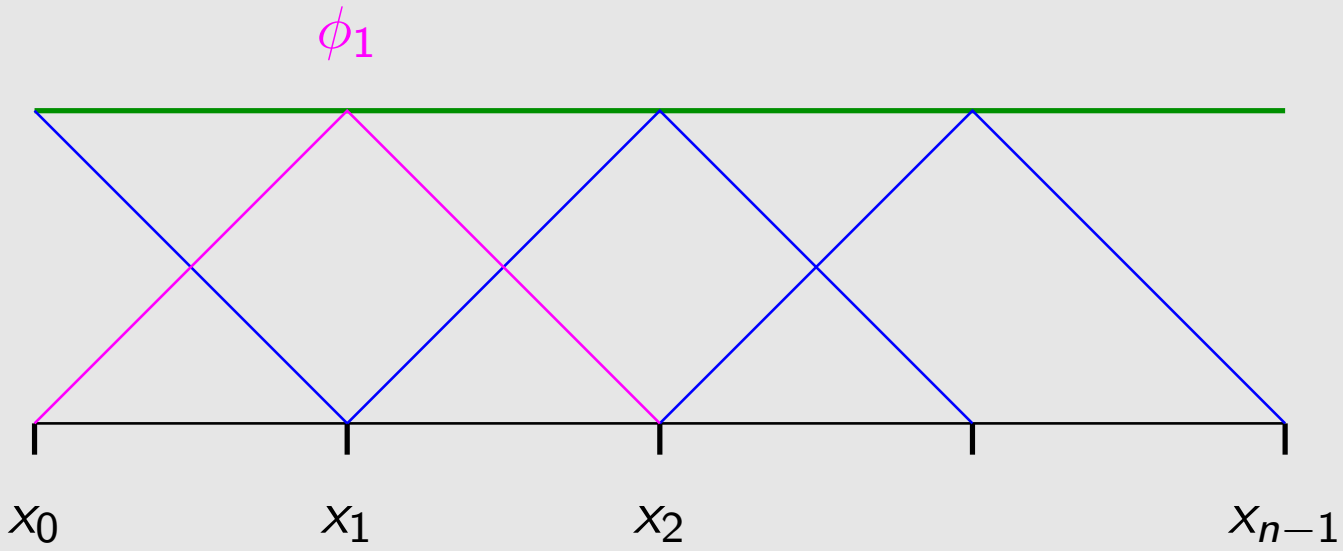
Classical Finite Elements
Linear Finite Elements in 1D

Finite Elements in 2D
Classification of Grids
Structured Grids
Unstructured Grids
Comparison

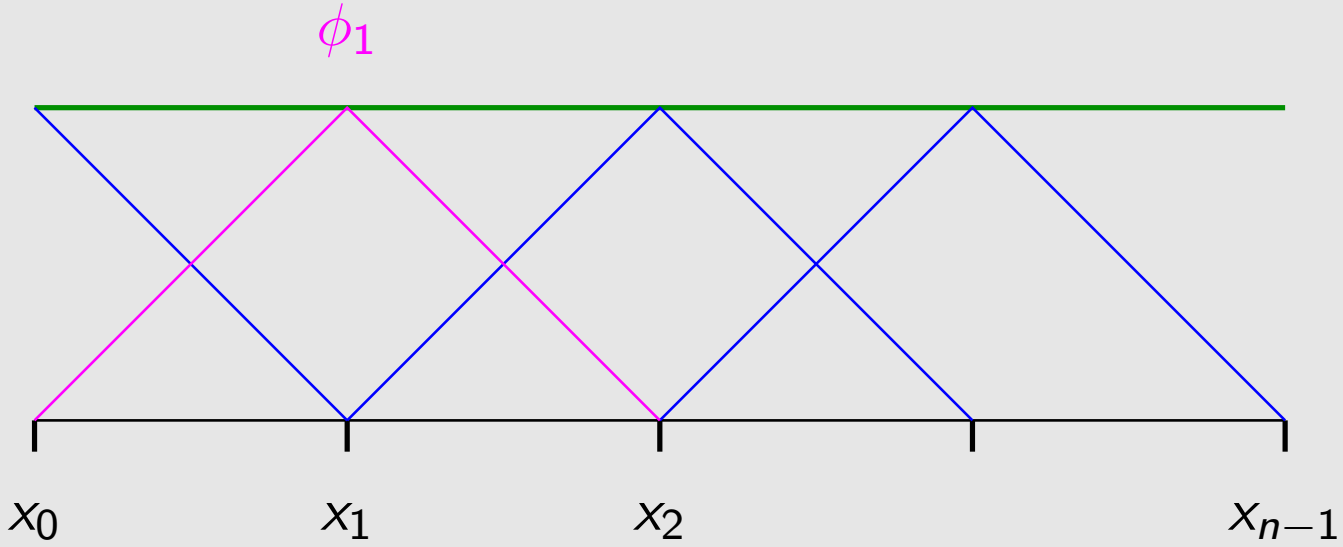
Composite Finite Elements
Geometry Extraction from Images
Construction of CFE Grids
CFE Basis Functions
Results (Elasticity)

Cooperations and Outlook

Discretization



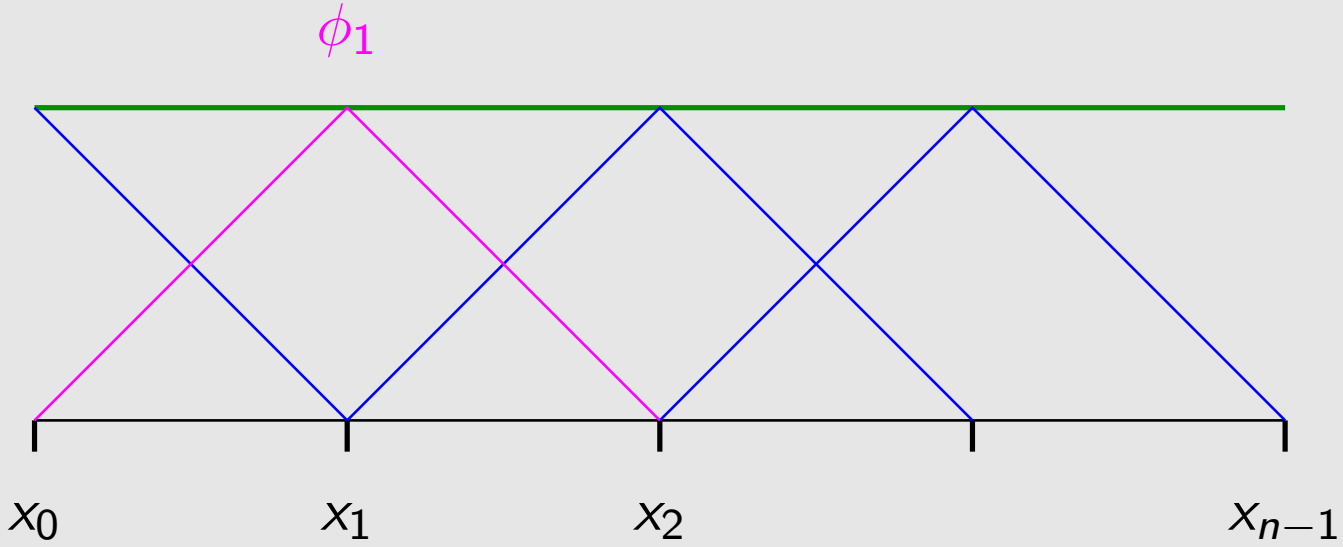
Discretization



Interpolation:

$$u_h(x) = \sum_{j=0}^{n-1} U_j \cdot \phi_j(x)$$

Discretization



Interpolation:

$$u_h(x) = \sum_{j=0}^{n-1} U_j \cdot \phi_j(x)$$

with

- $\phi_j(x_i) = \delta_{i,j}$
- ϕ_j piecewise linear
- $\sum_{j=0}^{n-1} \phi_j(x) = 1 \forall x \in \Omega$

Classical Finite Elements
- Linear Finite Elements in 1D

Finite Elements in 2D
- Classification of Grids
- Structured Grids
- Unstructured Grids
- Comparison

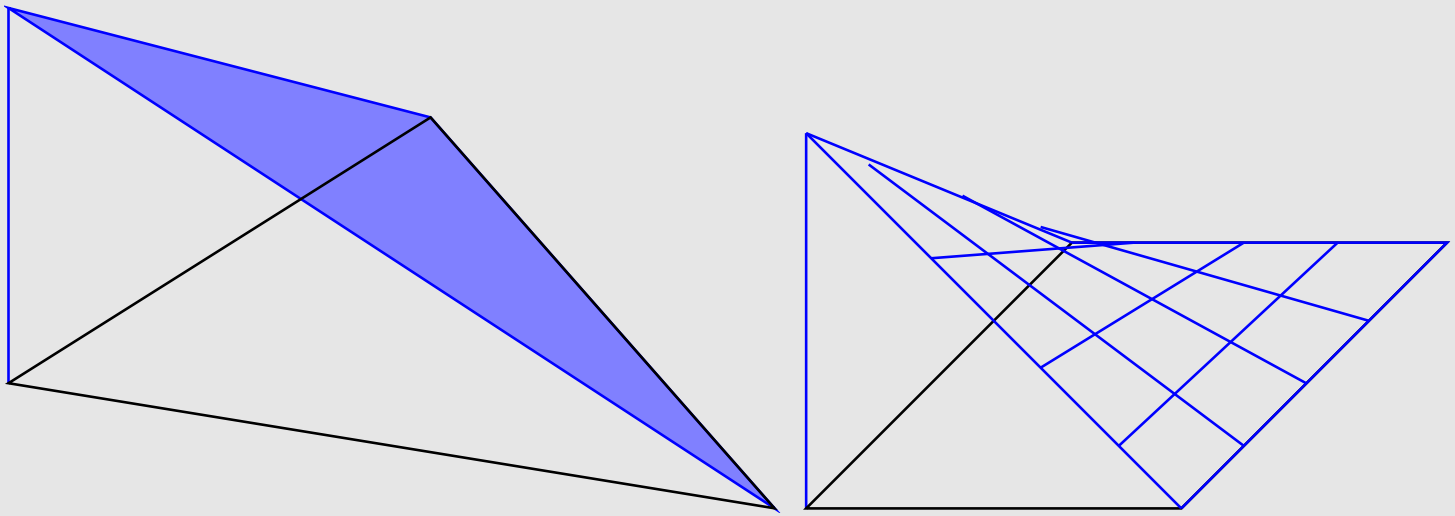
Composite Finite Elements
- Geometry Extraction from Images
- Construction of CFE Grids
- CFE Basis Functions
- Results (Elasticity)

Cooperations and Outlook

Generalization of “linear”

Two possibilities

- linear basis functions on triangles (tetrahedra, . . .)
- multilinear bf on rectangles (cuboids, . . .)



Classical Finite Elements
Linear Finite Elements in 1D

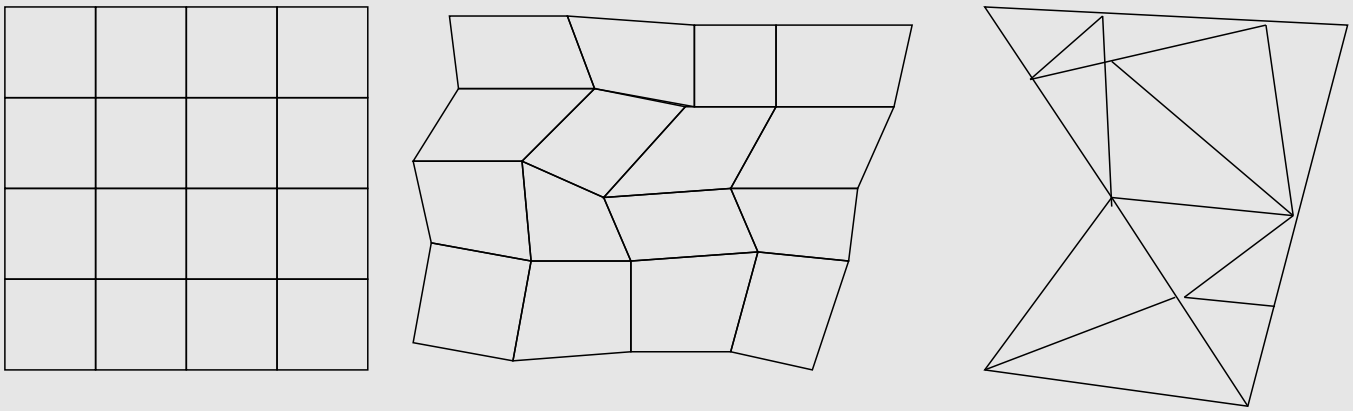
Finite Elements in 2D
Classification of Grids
Structured Grids
Unstructured Grids
Comparison

Composite Finite Elements
Geometry Extraction from Images
Construction of CFE Grids
CFE Basis Functions
Results (Elasticity)

Cooperations and Outlook

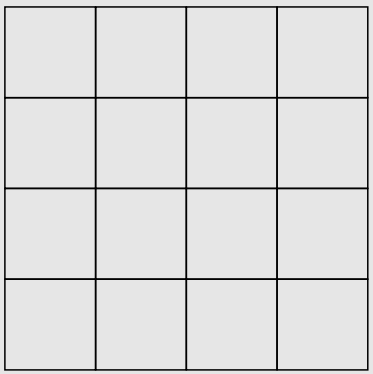
Classification of Grids

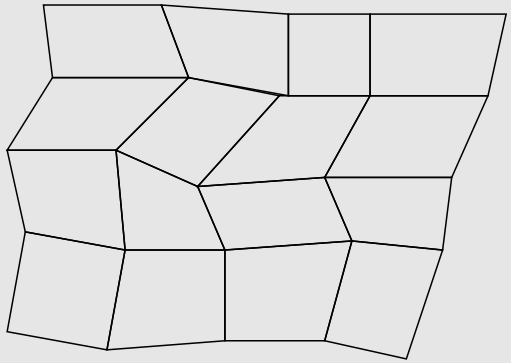
- uniform grids: equal grid cells

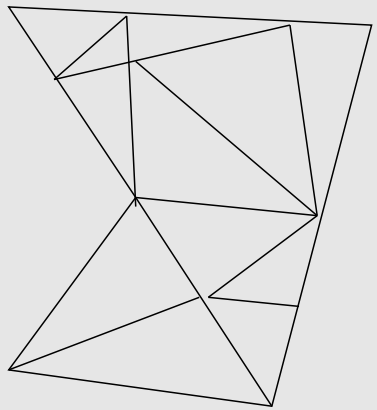


Classification of Grids

- uniform grids: equal grid cells
- structured grids: same topology at each point
-

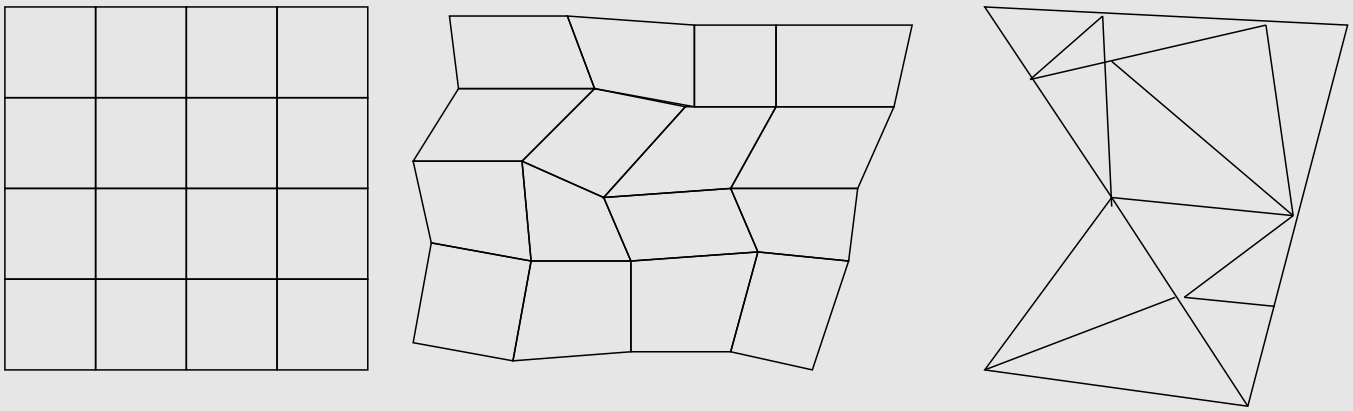






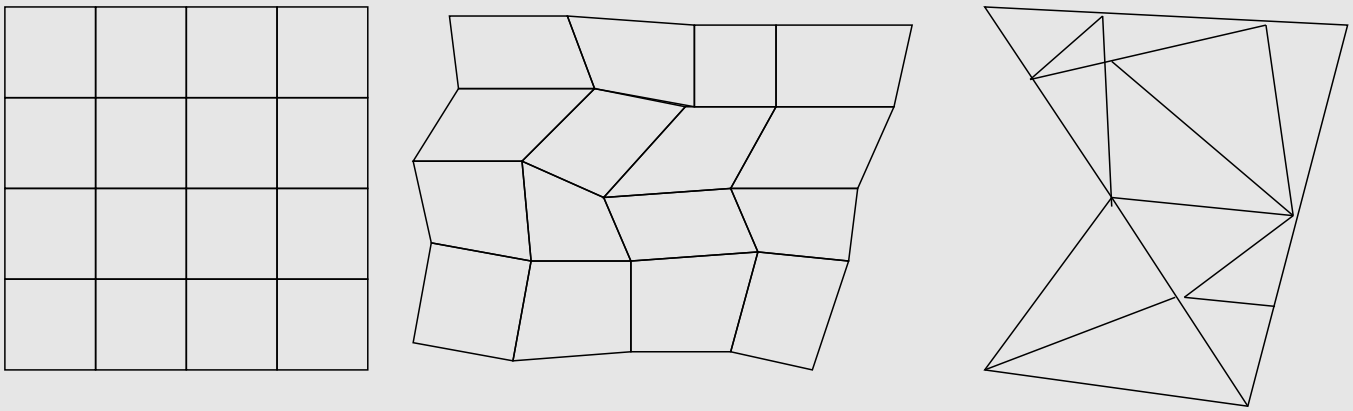
Classification of Grids

- uniform grids: equal grid cells
- structured grids: same topology at each point
- unstructured grids



Classification of Grids

- uniform grids: equal grid cells
- structured grids: same topology at each point
- unstructured grids



increasing geometric flexibility but increasing complexity

Characteristics of Grids

1. how to generate?
2. geometrically useful?
3. how to describe?
4. how to refine?
5. how to coarsen?
6. numerically “nice”? “nice” system of equations?

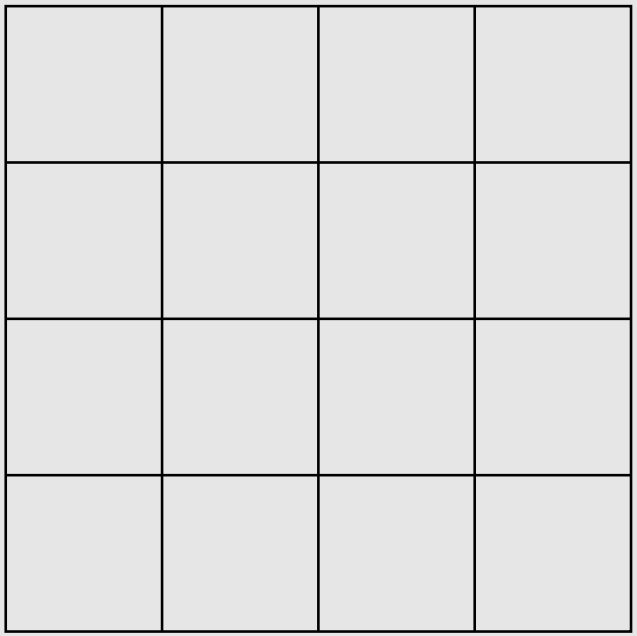
Classical Finite Elements
Linear Finite Elements in 1D

Finite Elements in 2D
Classification of Grids
Structured Grids
Unstructured Grids
Comparison

Composite Finite Elements
Geometry Extraction from Images
Construction of CFE Grids
CFE Basis Functions
Results (Elasticity)

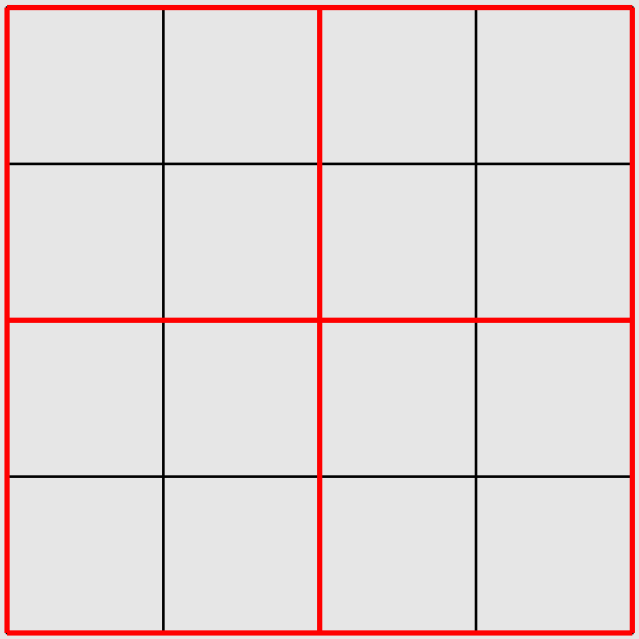
Cooperations and Outlook

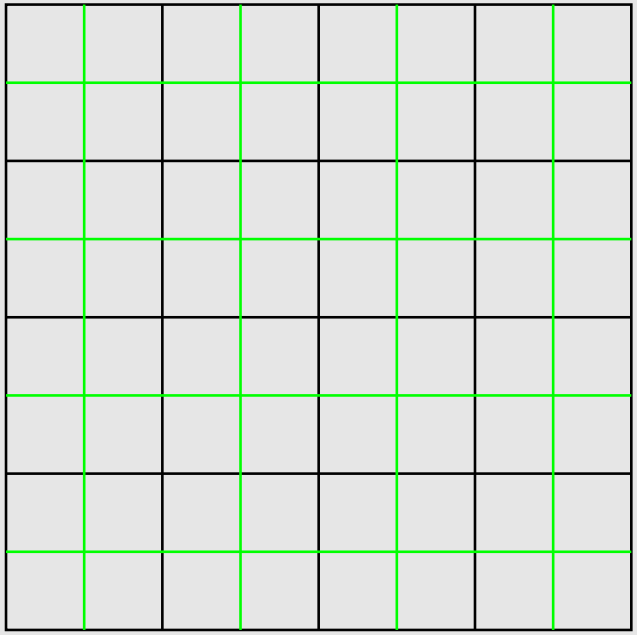
Structured Grids



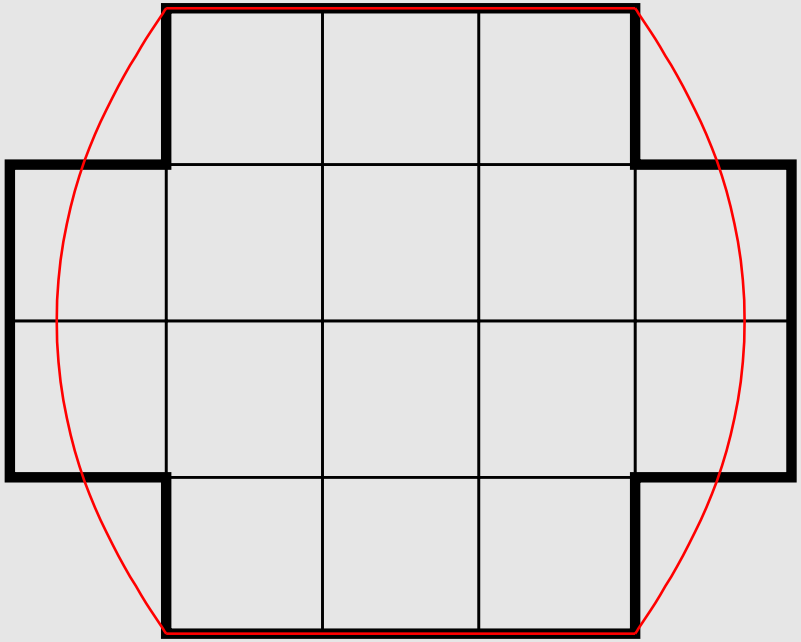
- easily generated and described implicitely
- sparse and structured system of equations

Structured Grids: Coarsening and Refinement





Structured Grids: Geometrically useful?



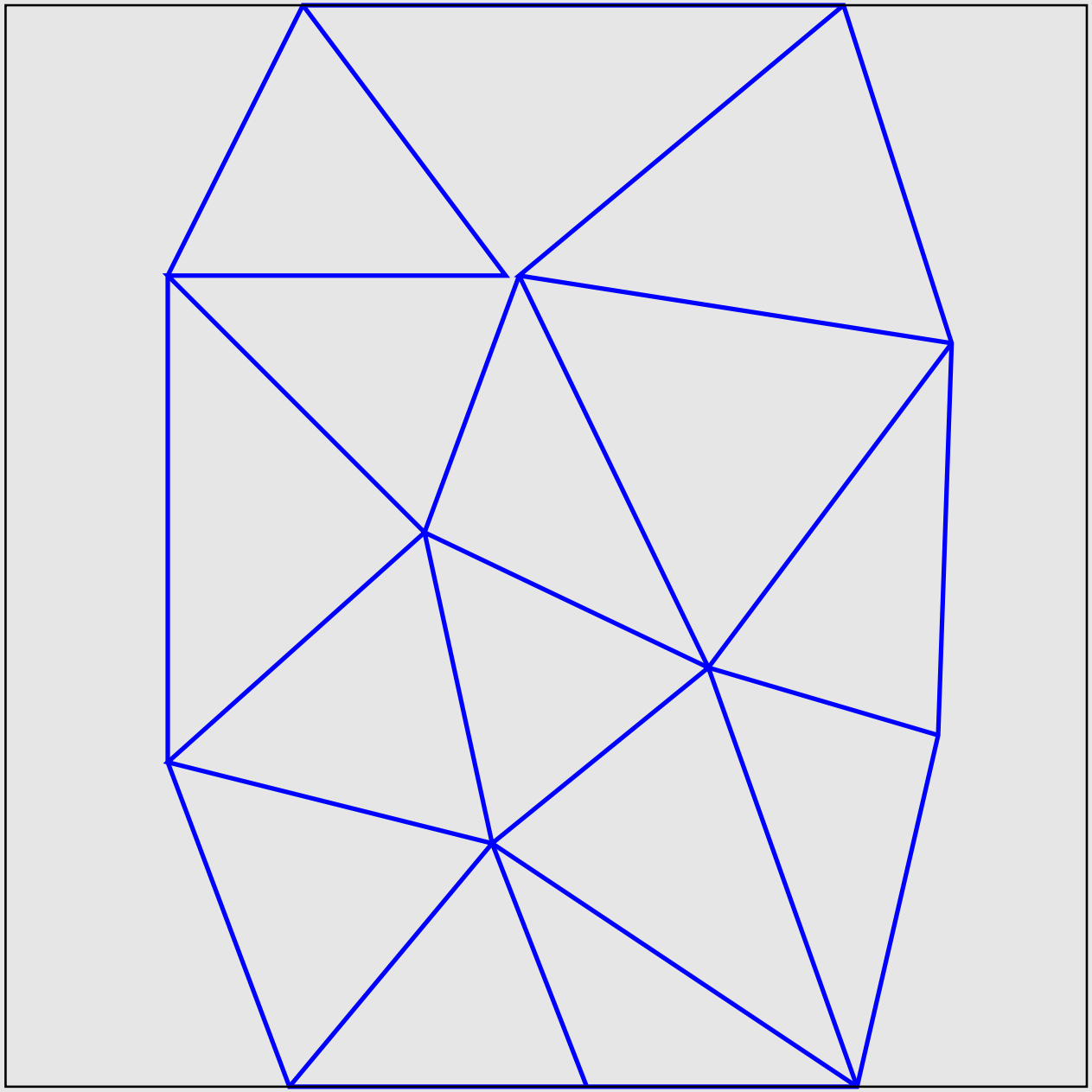
Classical Finite Elements
Linear Finite Elements in 1D

Finite Elements in 2D
Classification of Grids
Structured Grids
Unstructured Grids
Comparison

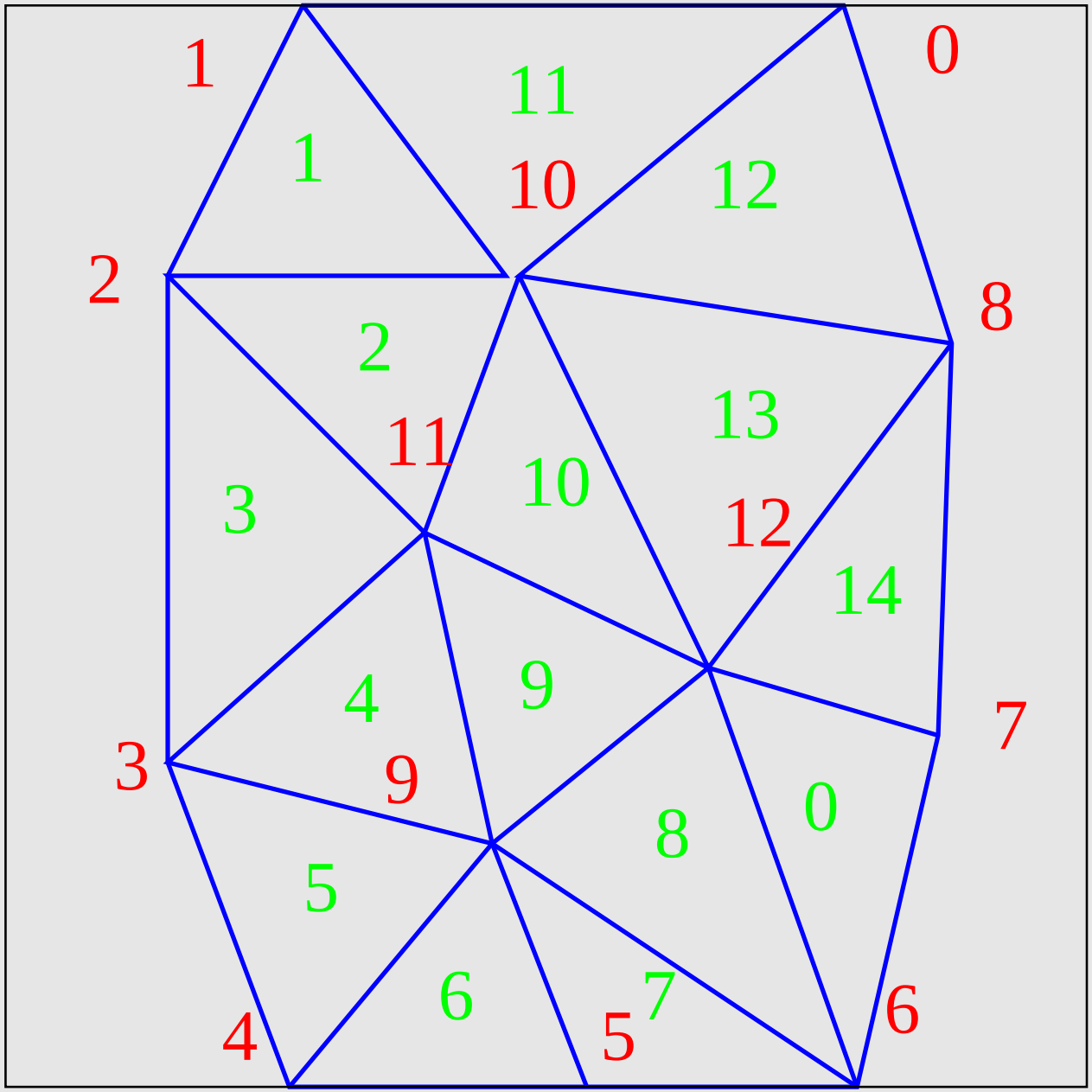
Composite Finite Elements
Geometry Extraction from Images
Construction of CFE Grids
CFE Basis Functions
Results (Elasticity)

Cooperations and Outlook

Unstructured Grids



Unstructured Grids: Storage



Unstructured Grids: Storage

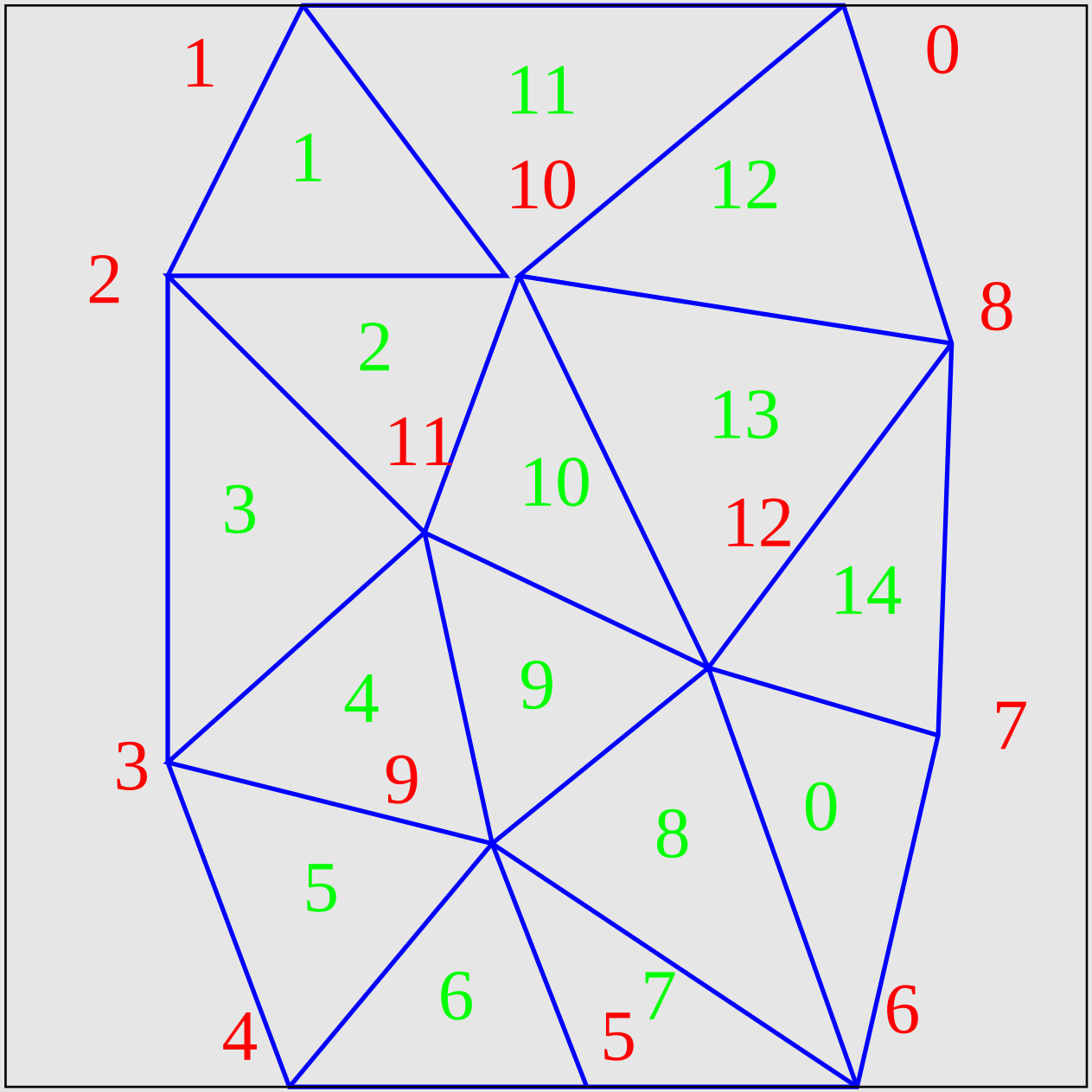
Node	x	y
0	0.8	1.0
1	0.3	1.0
2	0.2	0.75
3	0.2	0.25
4	0.3	0.0
5	0.5	0.0
6	0.8	0.0
7	0.85	0.4
8	0.9	0.65
9	0.45	0.2
10	0.5	0.7
11	0.4	0.55
12	0.65	0.4

Unstructured Grids: Storage

Node	x	y
0	0.8	1.0
1	0.3	1.0
2	0.2	0.75
3	0.2	0.25
4	0.3	0.0
5	0.5	0.0
6	0.8	0.0
7	0.85	0.4
8	0.9	0.65
9	0.45	0.2
10	0.5	0.7
11	0.4	0.55
12	0.65	0.4

Δ	P 0	P 1	P 2
0	6	7	12
1	1	2	10
2	2	11	10
3	2	3	11
4	3	9	11
5	3	4	9
6	9	4	5
7	5	6	9
8	12	9	6
9	11	9	12
10	11	12	0
11	10	0	1
12	0	10	8
13	12	8	10
14	12	7	8

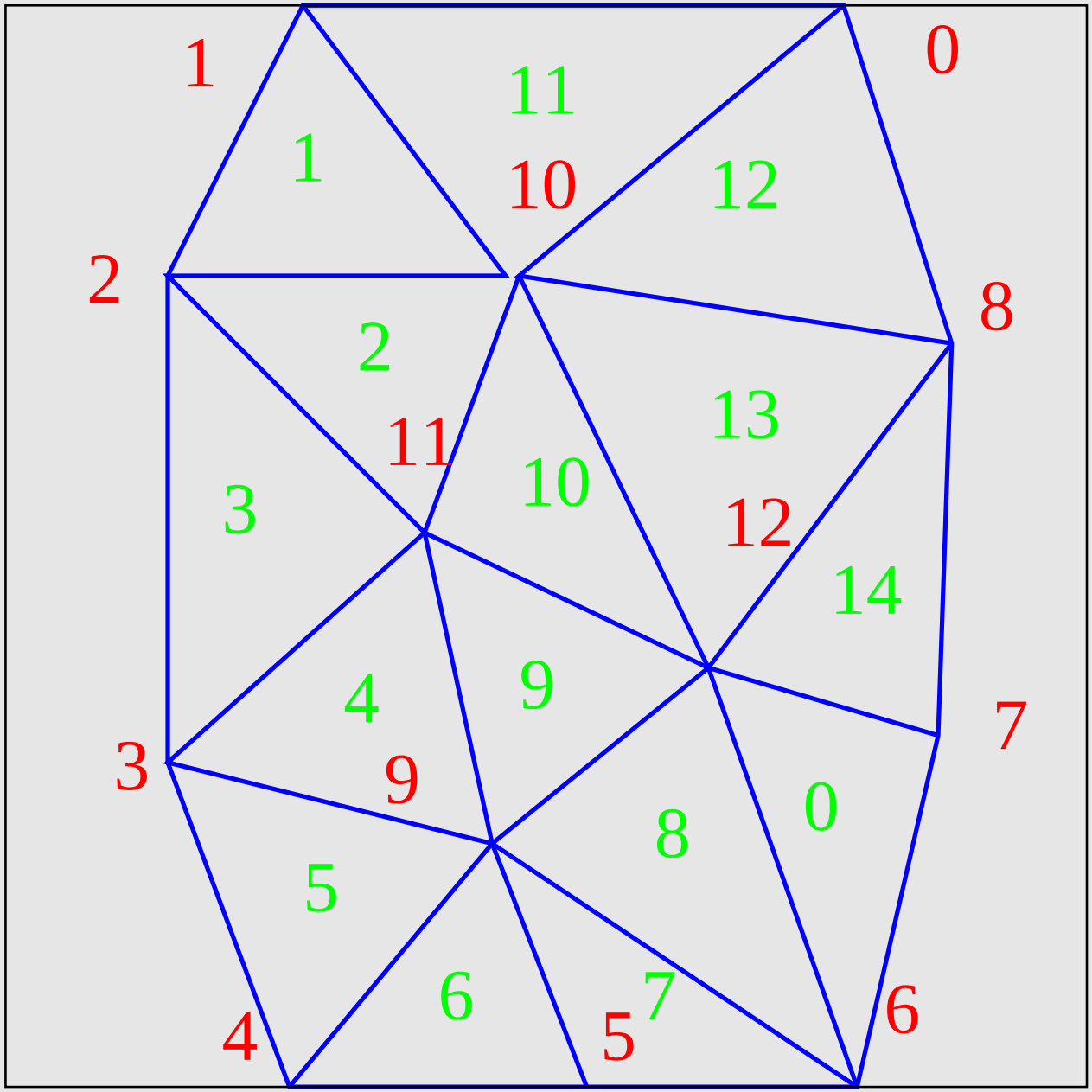
Implicite Information: Adjacent Triangles?



Implicite Information: Adjacent Triangles

Node	Δ 0	Δ 1	Δ 2	Δ 3	Δ 4	Δ 5
0	11	12				
1	11	1				
2	1	3	2			
3	3	4	5			
4	5	6				
5	7	6				
6	0	8	7			
7	14	0				
8	12	13	14			
9	4	5	6	7	8	9
10	1	2	10	13	12	11
11	2	3	4	9	10	
12	9	8	0	14	13	10

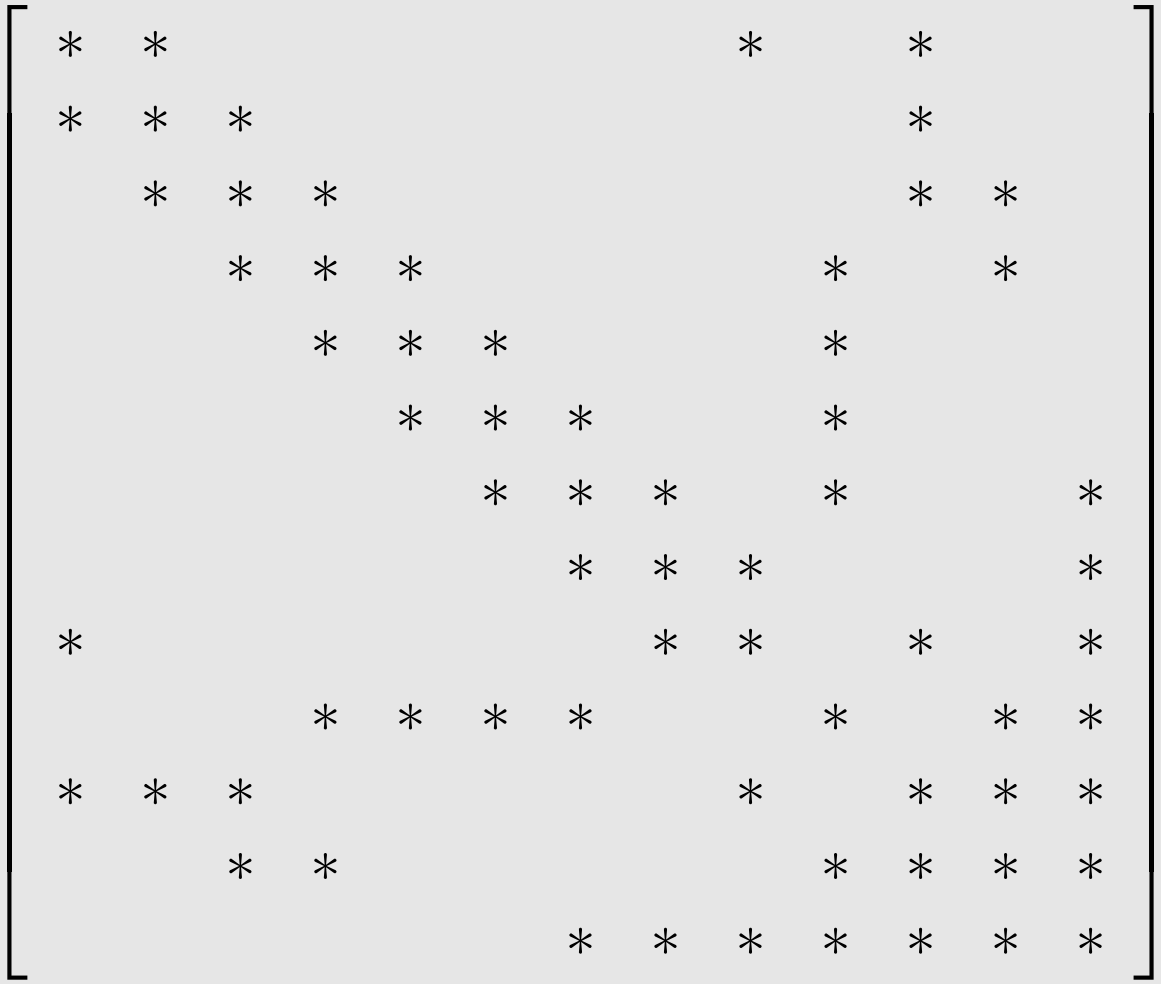
Implicite Information: Node Neighbors?



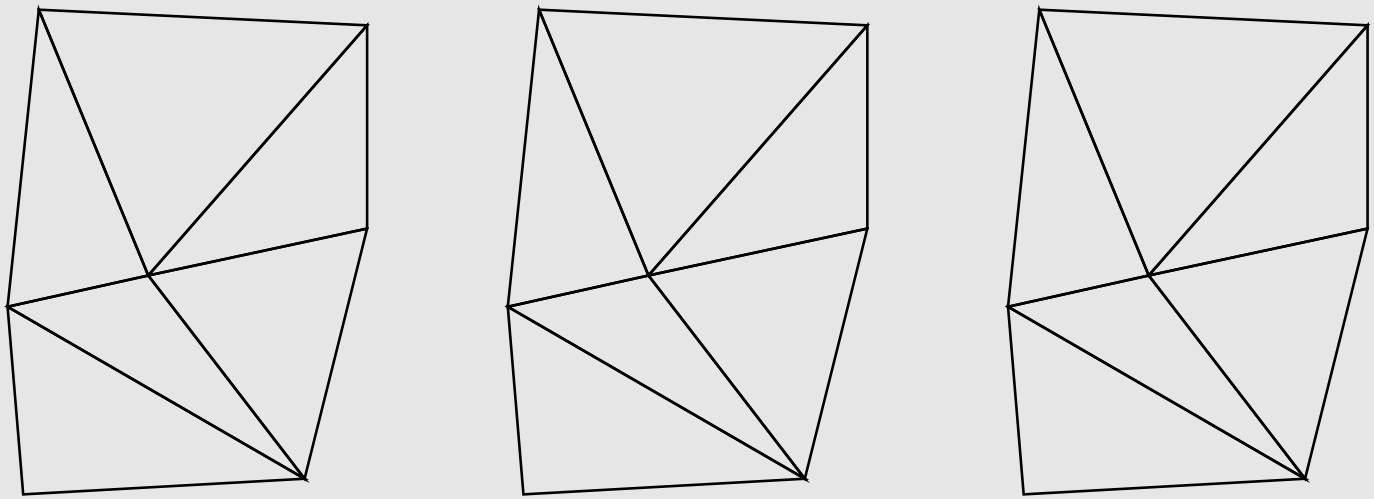
Implicite Information: Node Neighbors

Node = N 0	N 1	N 2	N 3	N 4	N 5	N 6
0	1	10	8			
1	2	10	0			
2	1	3	11	10		
3	4	9	11	2		
4	3	5	9			
5	4	6	9			
6	7	12	9	5		
7	6	8	12			
8	0	10	12	7		
9	4	5	6	12	11	3
10	1	2	11	12	8	0
11	2	3	9	12	10	
12	11	9	6	7	8	10

Matrix Structure



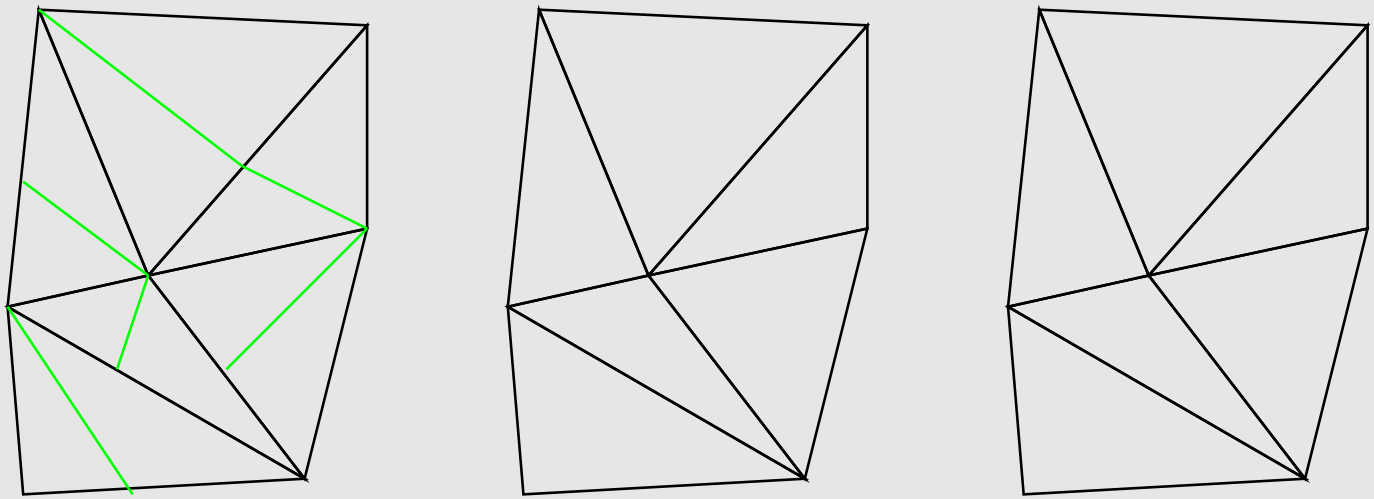
Unstructured Grids: Refinement



Three canonical methods for refinement:

- bisection
- trisection
- quadrisection

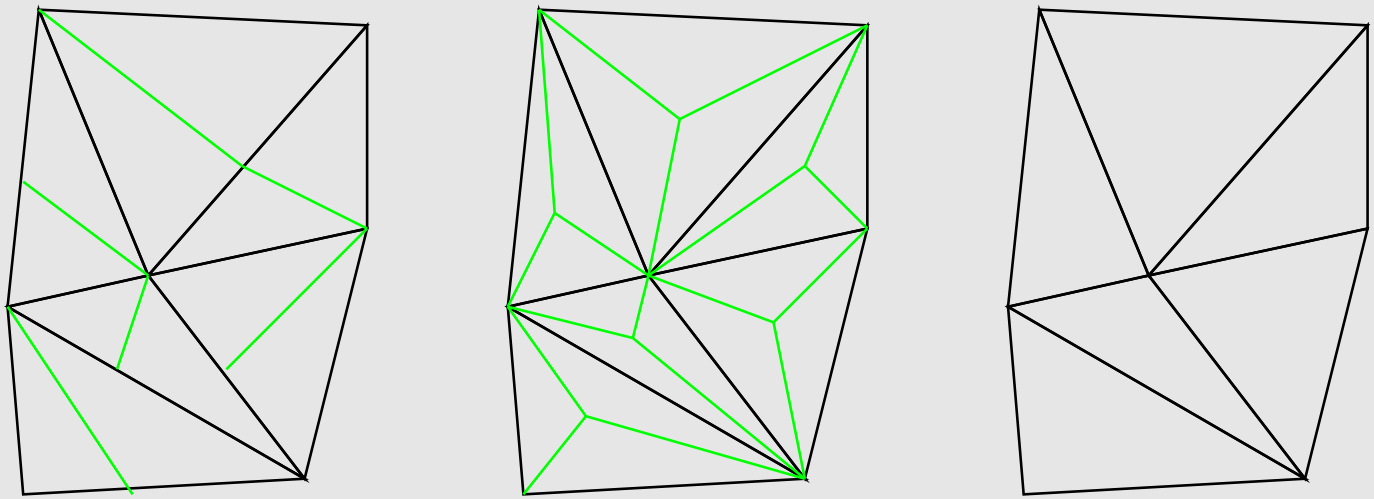
Unstructured Grids: Refinement



Three canonical methods for refinement:

- bisection: may get hanging nodes, leads to acute angles
- trisection
- quadrisection

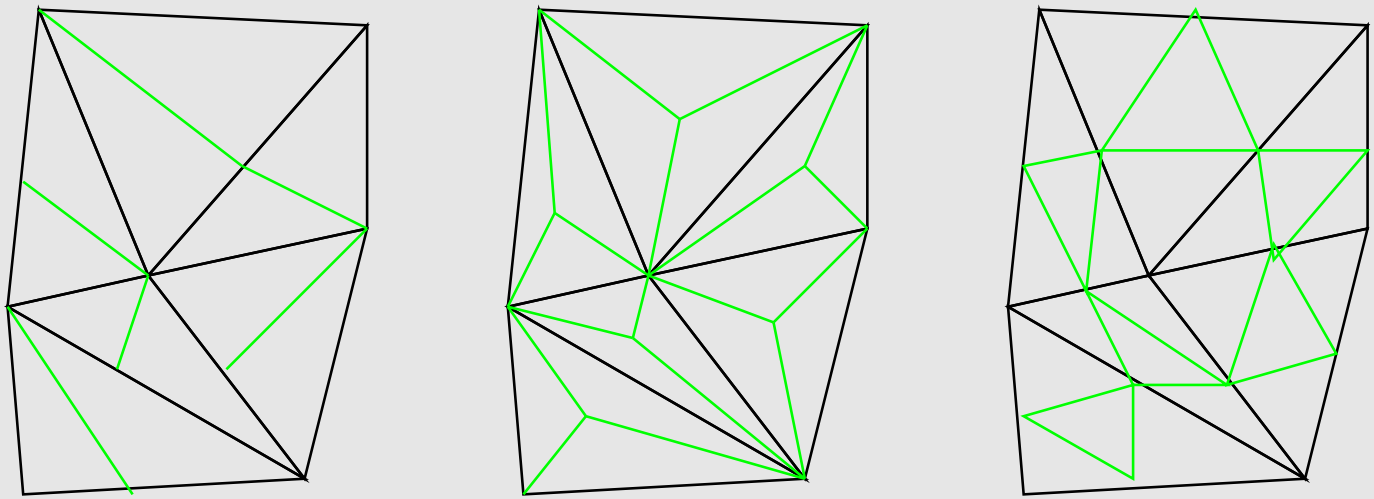
Unstructured Grids: Refinement



Three canonical methods for refinement:

- bisection: may get hanging nodes, leads to acute angles
- trisection using barycenter: leads to acute angles
- quadrisection

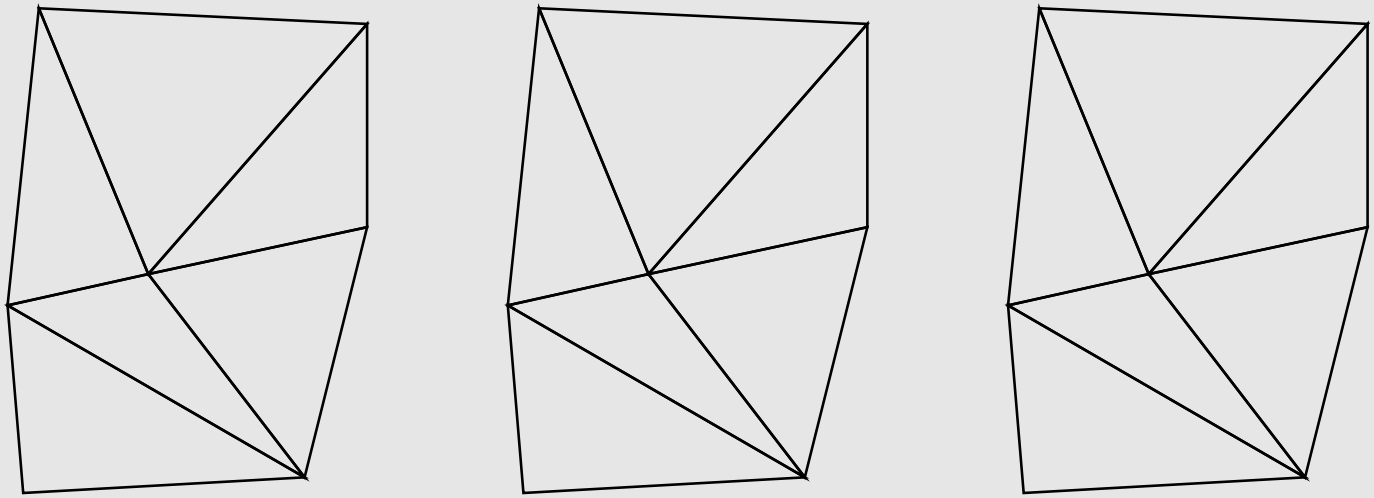
Unstructured Grids: Refinement



Three canonical methods for refinement:

- bisection: may get hanging nodes, leads to acute angles
- trisection using barycenter: leads to acute angles
- quadrisection using edge midpoints: get “similar” triangles

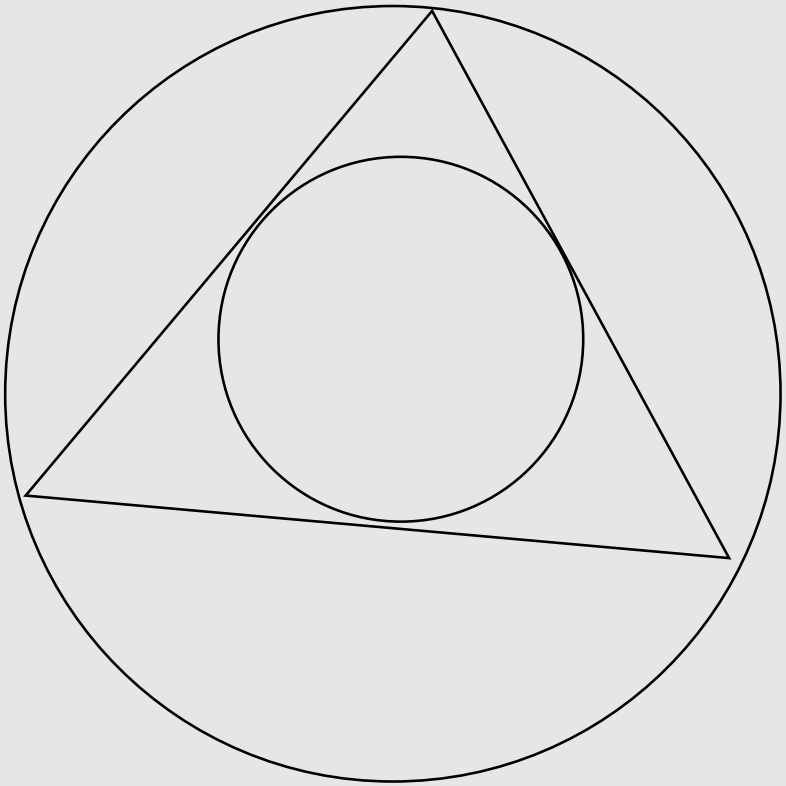
Unstructured Grids: Coarsening



- no canonical method for coarsening
- can remove nodes and generate new triangles

Badly Shaped Triangles

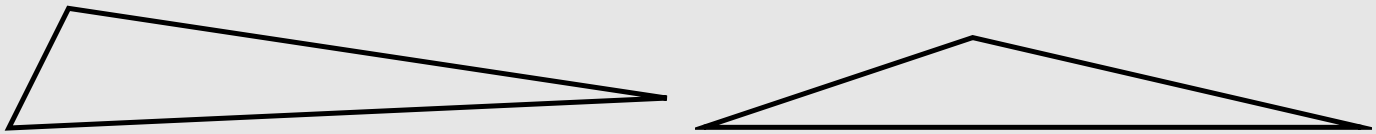
bad aspect ratio: small ratio of inner / outer radius



Badly Shaped Triangles

Classification (Cheng, Dey et al. 2000):

- dagger (1 short edge, ≥ 1 small angle)
- blade (0 short edges, 2 small angles)



Extension to 3D

- hexahedra (cubes) instead of squares

Extension to 3D

- hexahedra (cubes) instead of squares
- tetrahedra instead of triangles (4 vertices)
-
-

Extension to 3D

- hexahedra (cubes) instead of squares
- tetrahedra instead of triangles (4 vertices)
- refinement: midpoints of edges
-

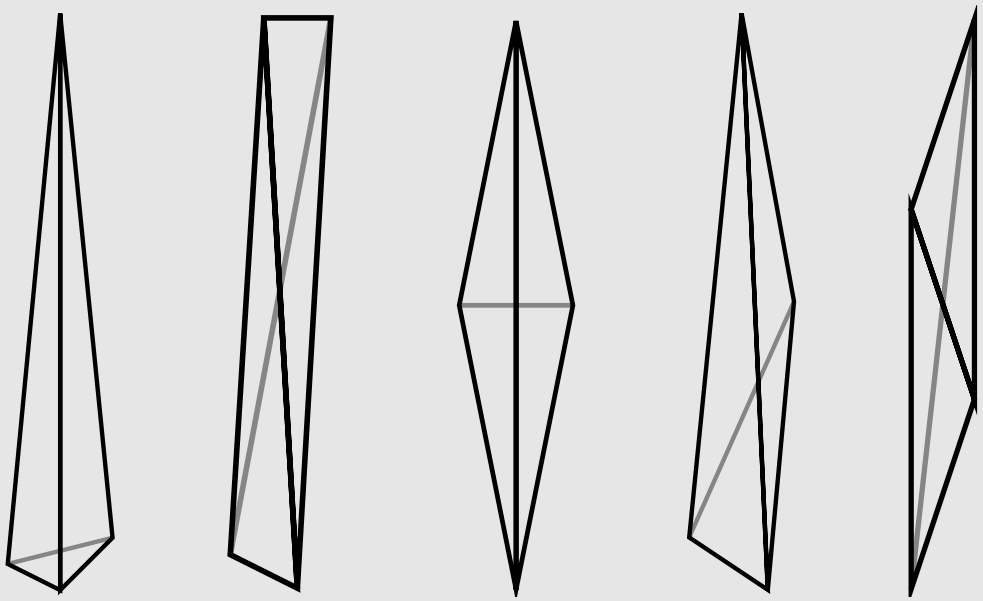
Extension to 3D

- hexahedra (cubes) instead of squares
- tetrahedra instead of triangles (4 vertices)
- refinement: midpoints of edges
- zoo of badly-shaped tetrahedra

Badly Shaped Tetrahedra

Tetrahedra with points close to line:

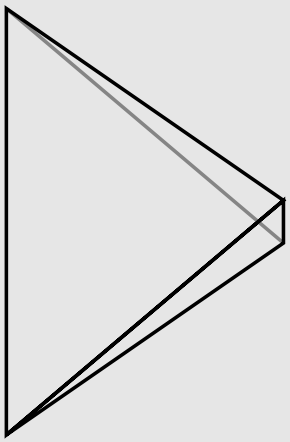
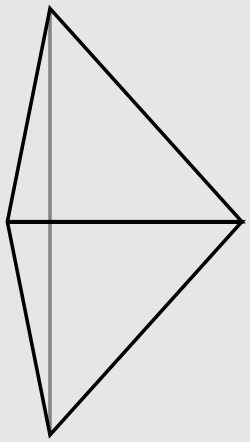
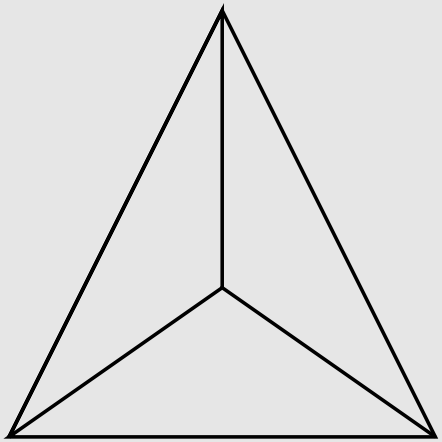
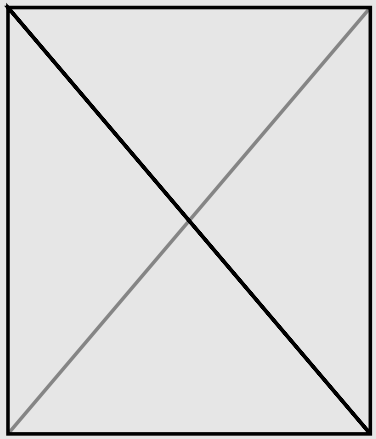
- spire: 3:1
- splinter: 2:2
- spear: 1:2:1
- spike: 2:1:1
- spindle: 1:1:1:1



Badly Shaped Tetrahedra

Tetrahedra with points close to plane:

- wedge: 1 long edge with small dihedral angle
- spade: 2
- cap: 3
- sliver: 4

Classical Finite Elements
Linear Finite Elements in 1D

Finite Elements in 2D
Classification of Grids
Structured Grids
Unstructured Grids
Comparison

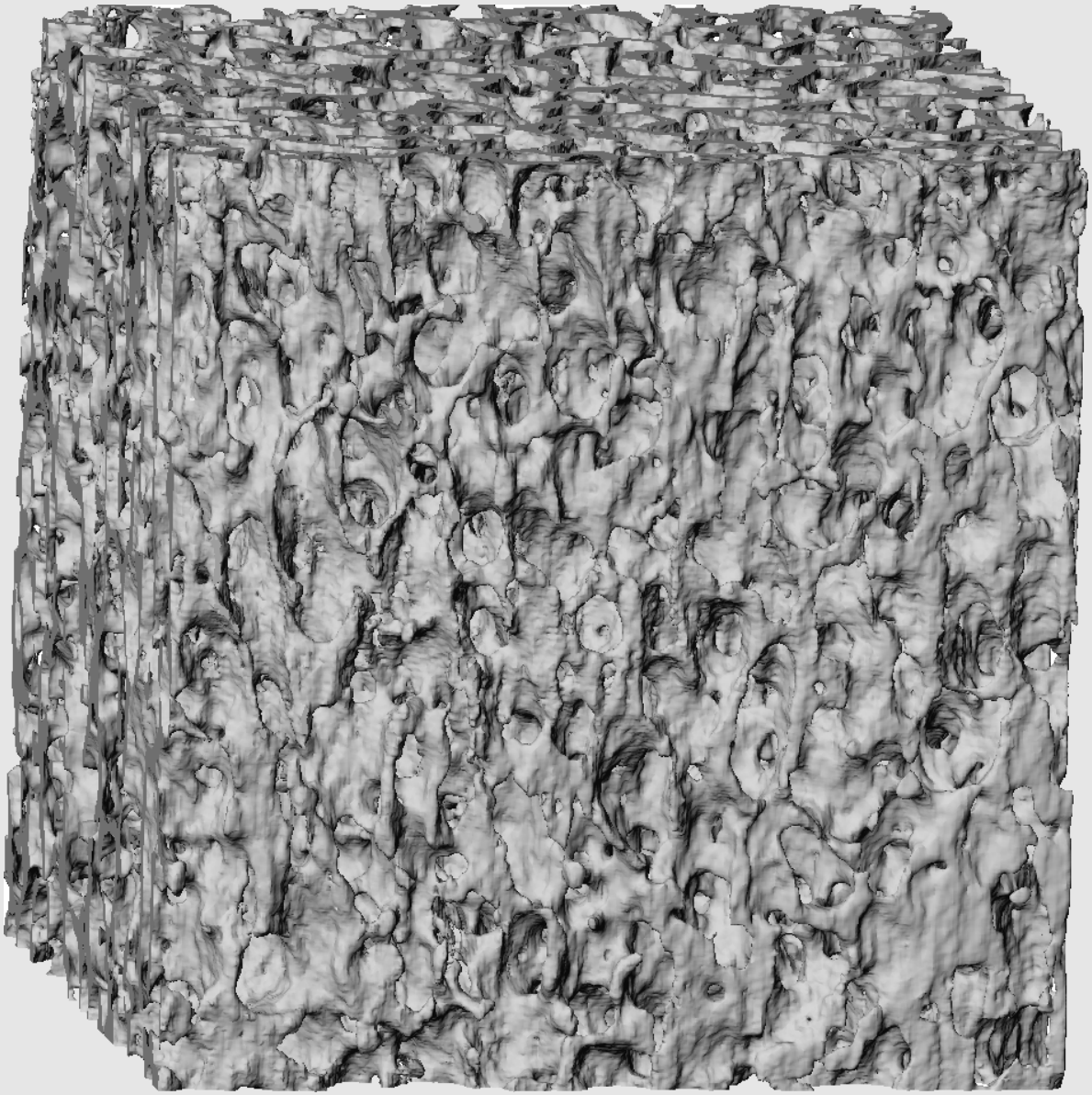
Composite Finite Elements
Geometry Extraction from Images
Construction of CFE Grids
CFE Basis Functions
Results (Elasticity)

Cooperations and Outlook

Structured vs. Unstructured Grids

	structured grids	**unstructured grids**
basis functions	multilinear	linear
generate	easy	difficult
geometry	not flexible	flexible
description	cheap: implicite	expensive: explicite
refinement	easy	easy
coarsening	easy	difficult
numerically nice	sparse and structured	sparse but unstructured

Goal: Complicated Microstructure



Goal: Complicated Microstructure

We have seen:

- structured grids cannot represent complicated geometries
- unstructured grids can – but they are expensive

Goal: Complicated Microstructure

We have seen:

- structured grids cannot represent complicated geometries
- unstructured grids can – but they are expensive

We have not seen:

- it is difficult to generate “good” unstructured grids

Classical Finite Elements
- Linear Finite Elements in 1D

Finite Elements in 2D
- Classification of Grids
- Structured Grids
- Unstructured Grids
- Comparison

Composite Finite Elements
- Geometry Extraction from Images
- Construction of CFE Grids
- CFE Basis Functions
- Results (Elasticity)

Cooperations and Outlook

Composite Finite Elements: The Idea

For complicated geometry:

- classically, use complicated mesh with simple basis functions
-
-

Composite Finite Elements: The Idea

For complicated geometry:

- classically, use complicated mesh with simple basis functions
- CFE: use simple mesh with complicated basis functions
-

Composite Finite Elements: The Idea

For complicated geometry:

- classically, use complicated mesh with simple basis functions
- CFE: use simple mesh with complicated basis functions
- both are immediately given by the image data (μCT scan)

Classical Finite Elements
Linear Finite Elements in 1D

Finite Elements in 2D
Classification of Grids
Structured Grids
Unstructured Grids
Comparison

Composite Finite Elements
Geometry Extraction from Images
Construction of CFE Grids
CFE Basis Functions
Results (Elasticity)

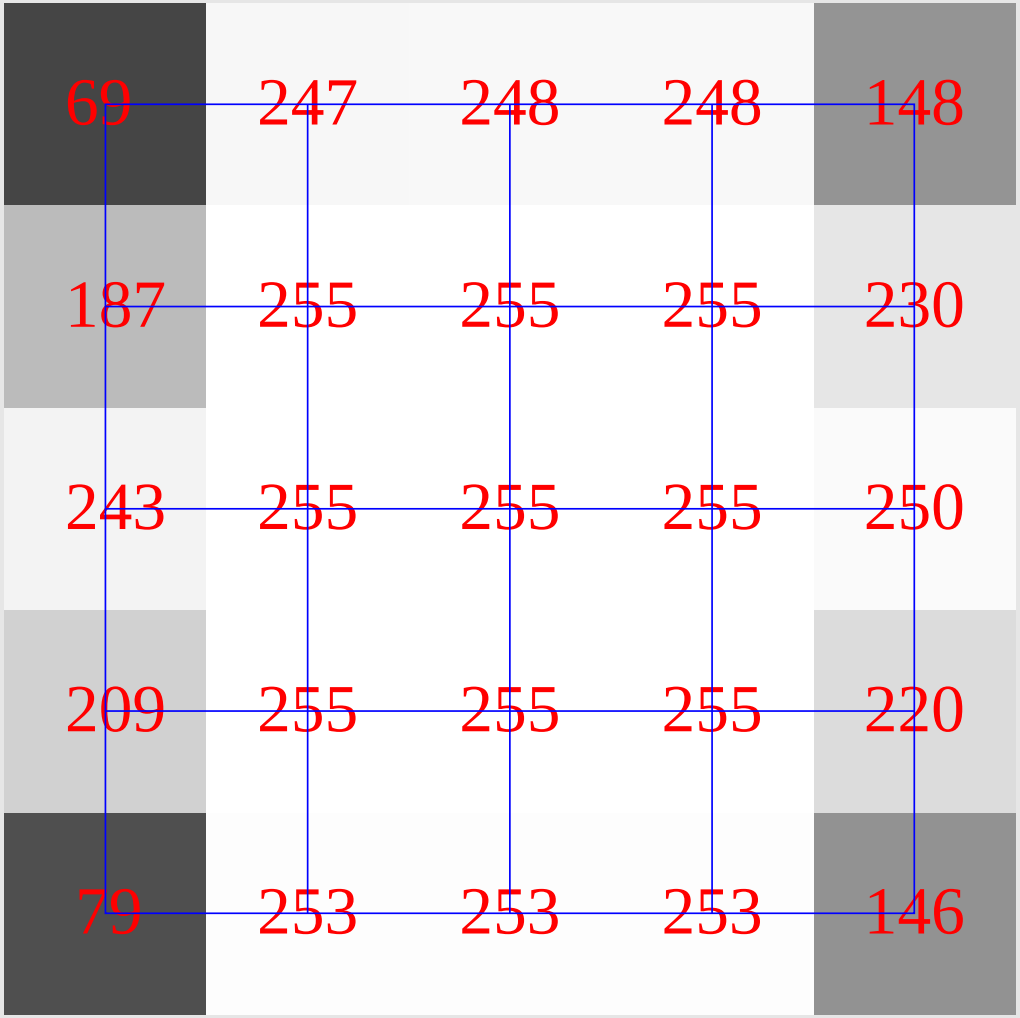
Cooperations and Outlook

Grey Images: CT-Scans

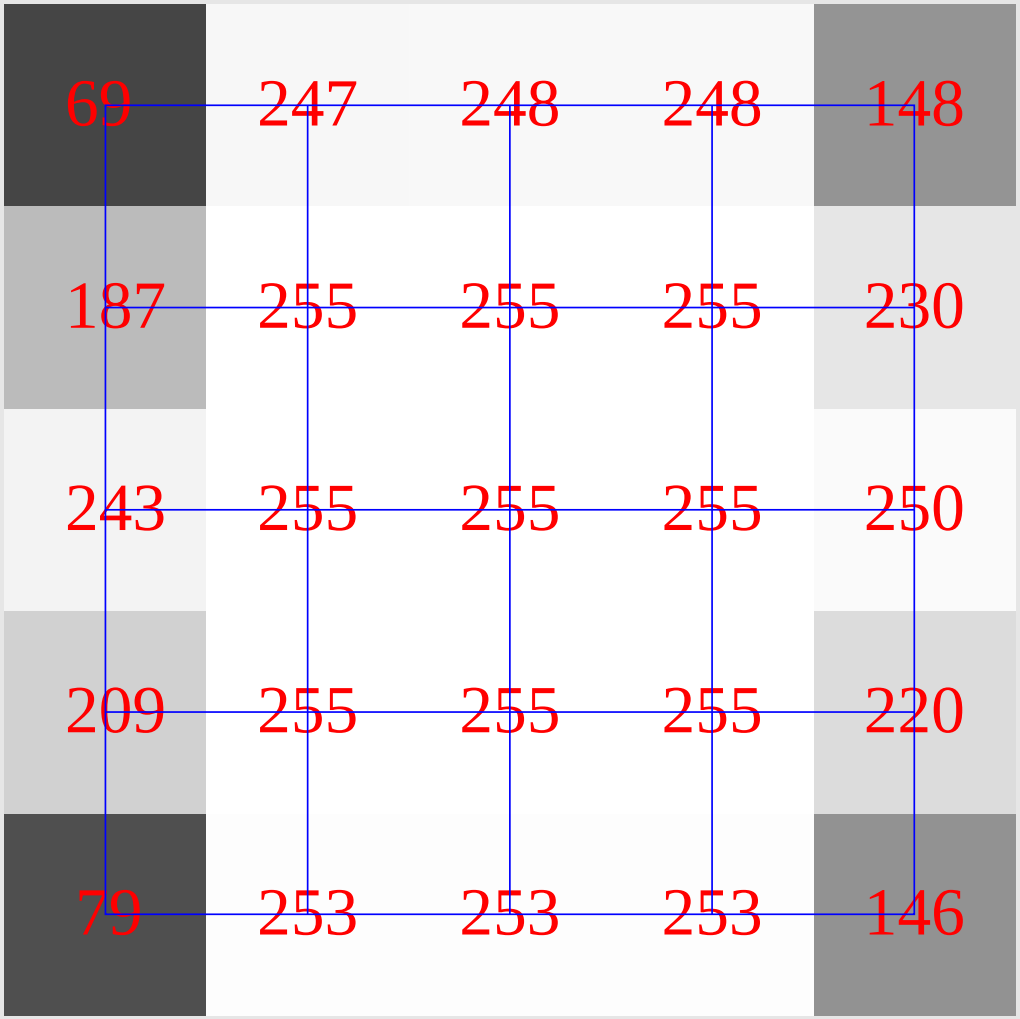
69	247	248	248	148
187	255	255	255	230
243	255	255	255	250
209	255	255	255	220
79	253	253	253	146

grey values represent X-ray attenuation

A Natural Grid for Grey Images



A Natural Grid for Grey Images



object boundary (240) can be recovered: keep these points

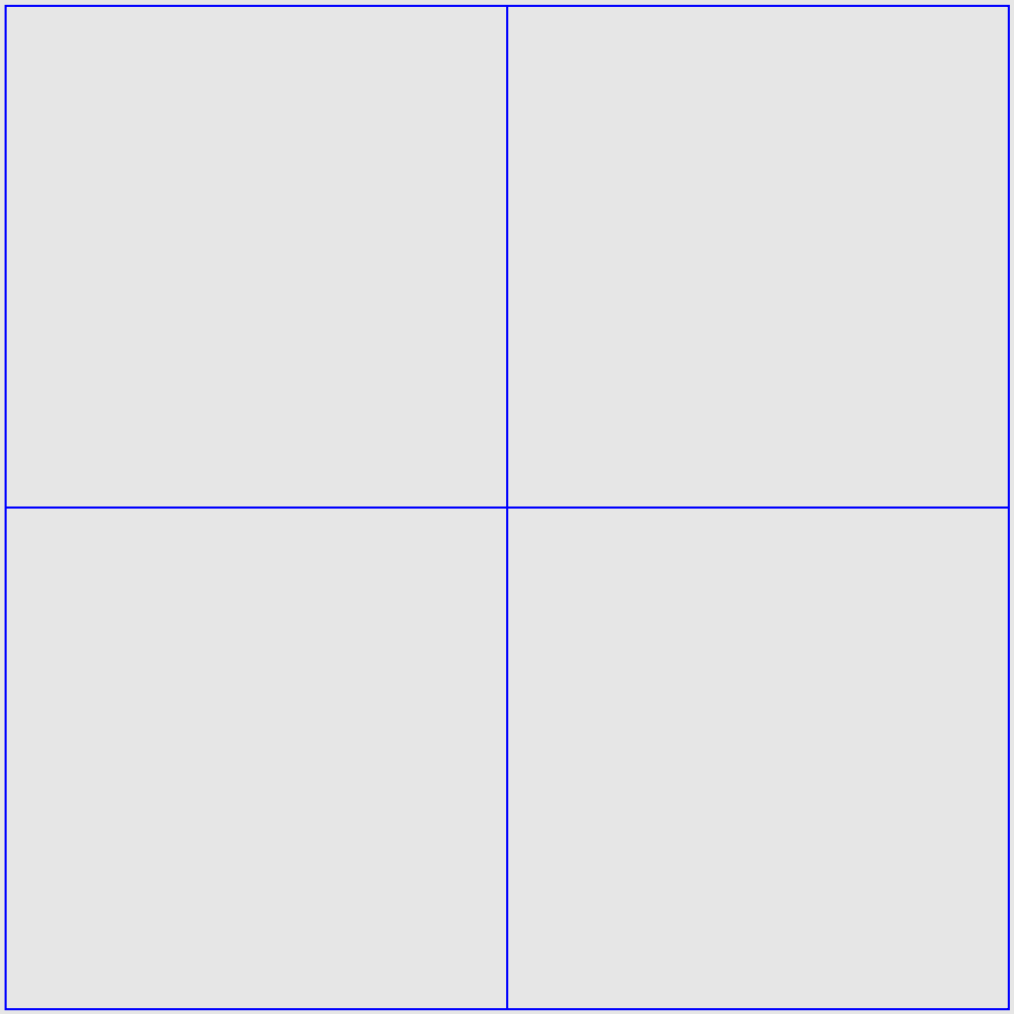
Classical Finite Elements
Linear Finite Elements in 1D

Finite Elements in 2D
Classification of Grids
Structured Grids
Unstructured Grids
Comparison

Composite Finite Elements
Geometry Extraction from Images
Construction of CFE Grids
CFE Basis Functions
Results (Elasticity)

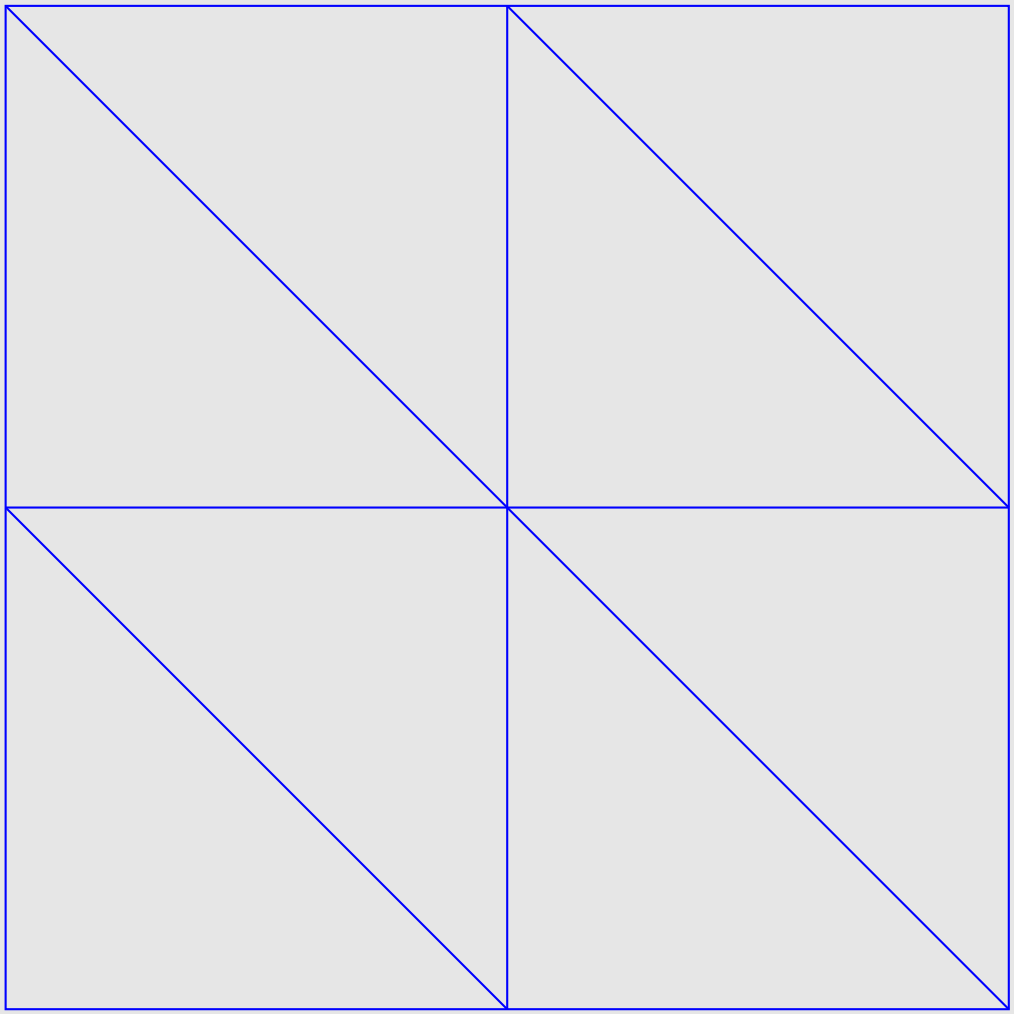
Cooperations and Outlook

Construction of CFE Grids



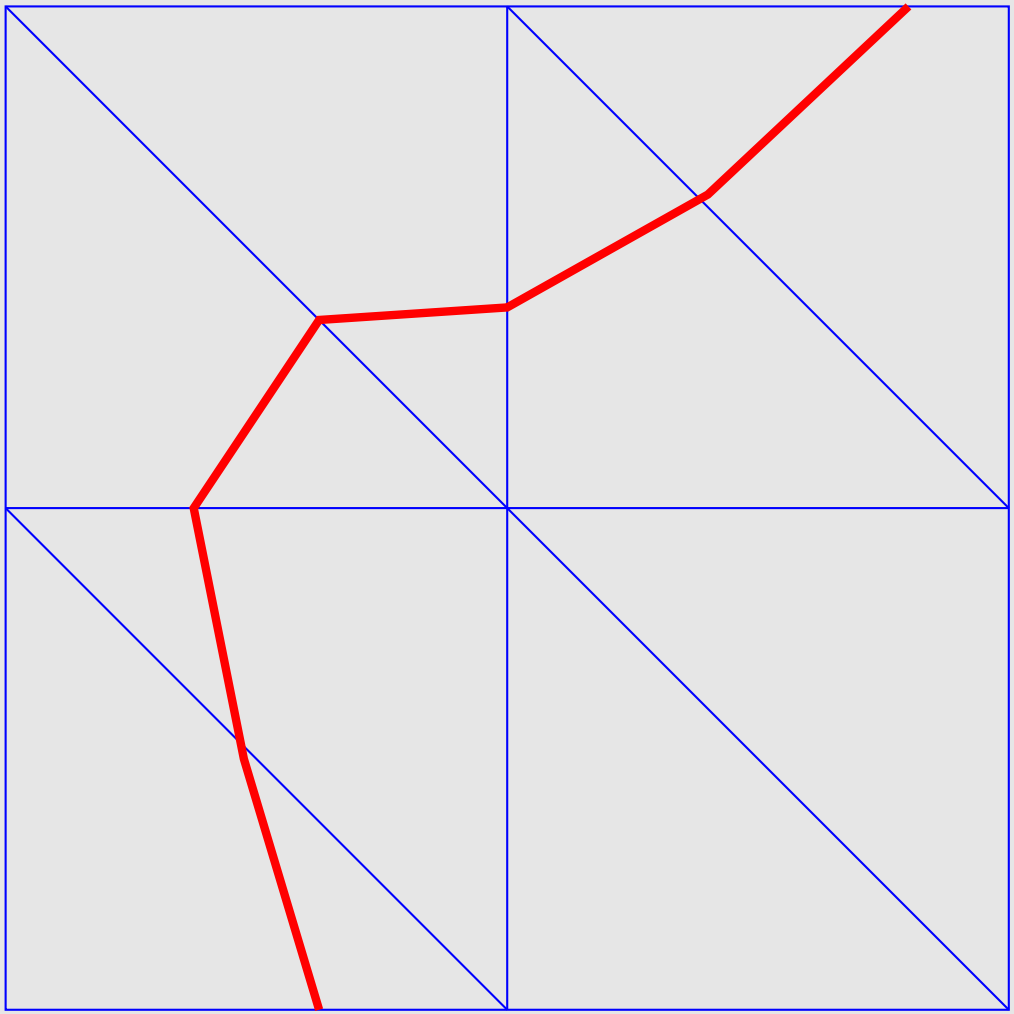
a grid consisting of squares

Construction of CFE Grids



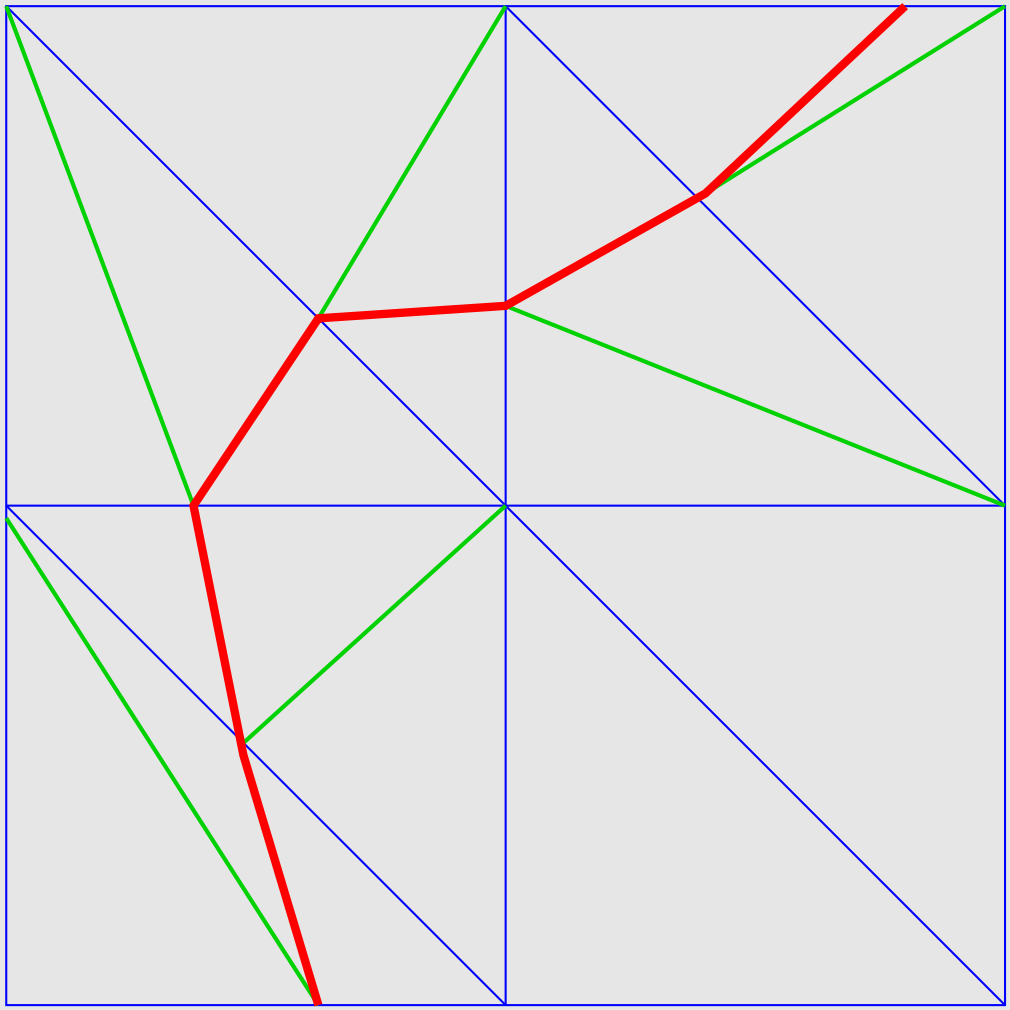
squares cut in triangles

Construction of CFE Grids

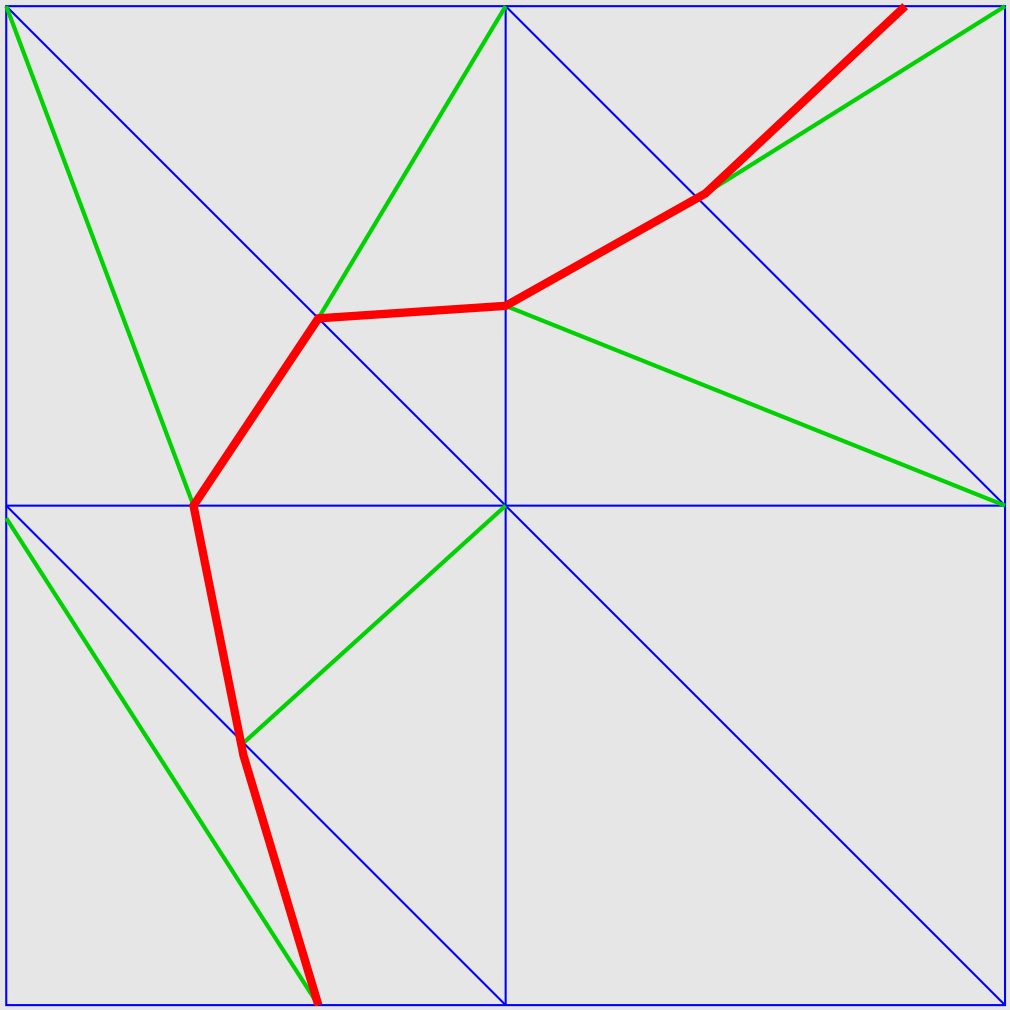


object boundary recovered from image

Construction of CFE Grids



Construction of CFE Grids



the “virtual grid” you don’t want to use

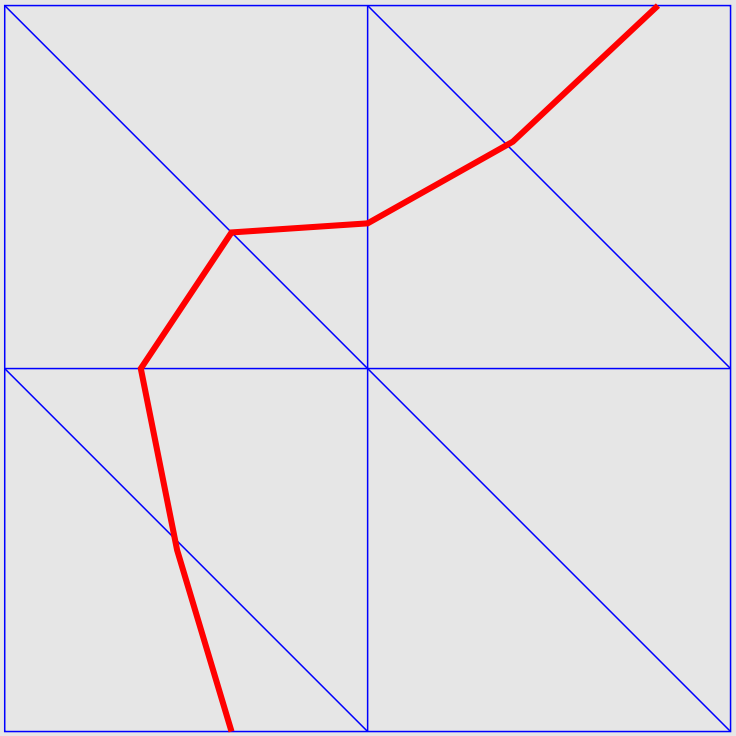
Classical Finite Elements
Linear Finite Elements in 1D

Finite Elements in 2D
Classification of Grids
Structured Grids
Unstructured Grids
Comparison

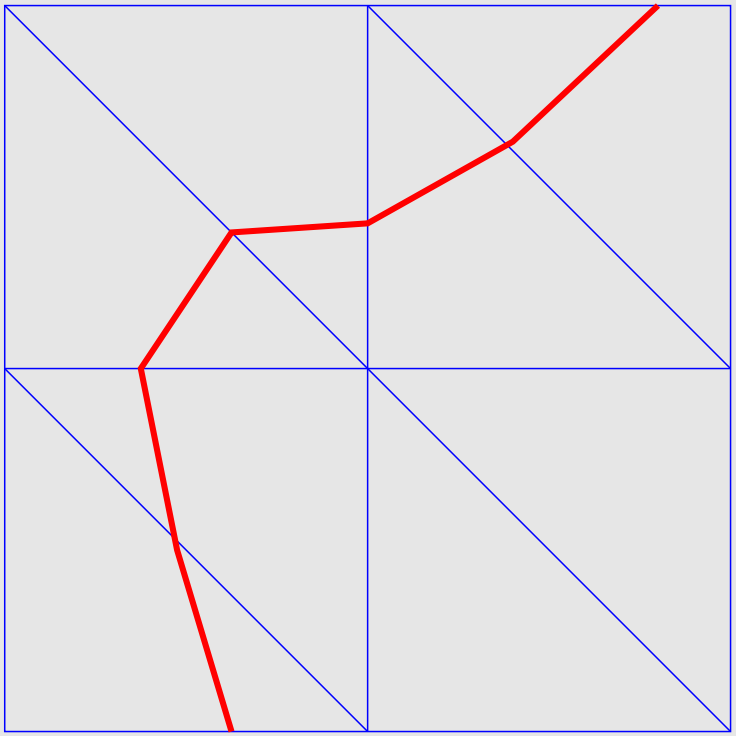
Composite Finite Elements
Geometry Extraction from Images
Construction of CFE Grids
CFE Basis Functions
Results (Elasticity)

Cooperations and Outlook

CFE Basis Functions

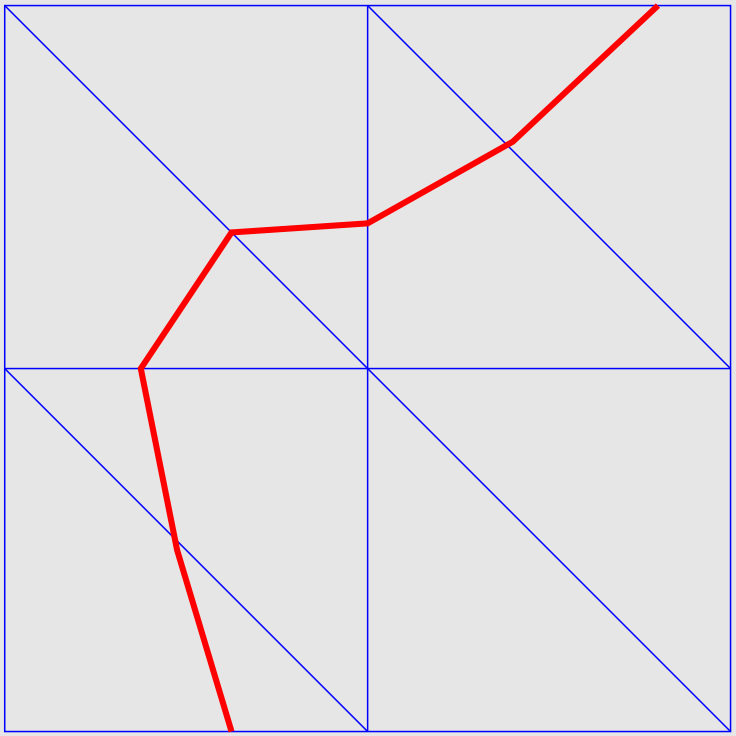


CFE Basis Functions



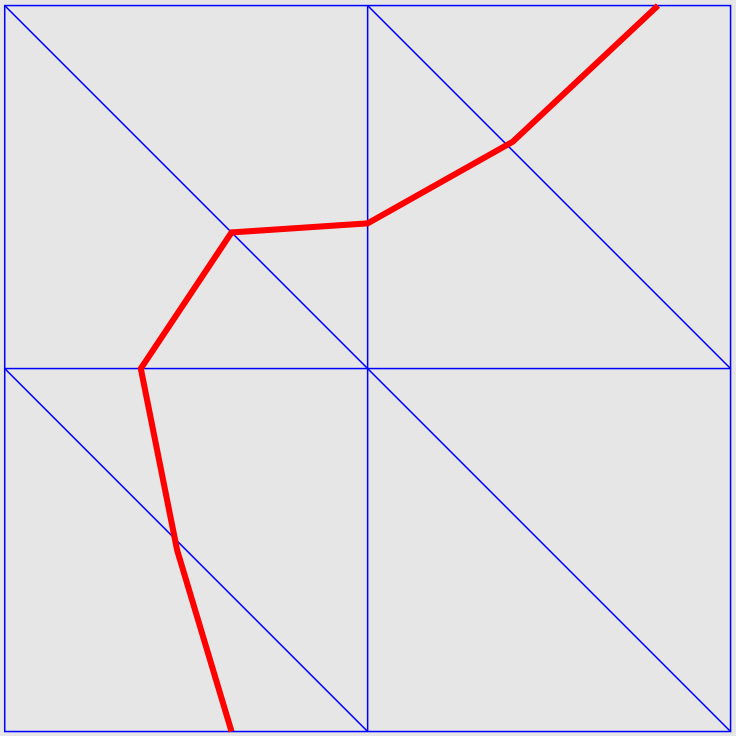
- regular grid points only as degrees of freedom: simple neighborhood relations

CFE Basis Functions



- regular grid points only as degrees of freedom: simple neighborhood relations
- cut off basis functions on regular grid at interface
-

CFE Basis Functions



- regular grid points only as degrees of freedom: simple neighborhood relations
- cut off basis functions on regular grid at interface
- virtual grid only needed internally

Composite Finite Elements: Advantages and Disadvantages

- meshing is easy, mesh is given by image data

Composite Finite Elements: Advantages and Disadvantages

- meshing is easy, mesh is given by image data
- structured grid
- hence structured sparse system of equations

Composite Finite Elements: Advantages and Disadvantages

- meshing is easy, mesh is given by image data
- structured grid
- hence structured sparse system of equations
- computing entries more difficult (b/c basis functions contain complexity of structure)
-
-

Composite Finite Elements: Advantages and Disadvantages

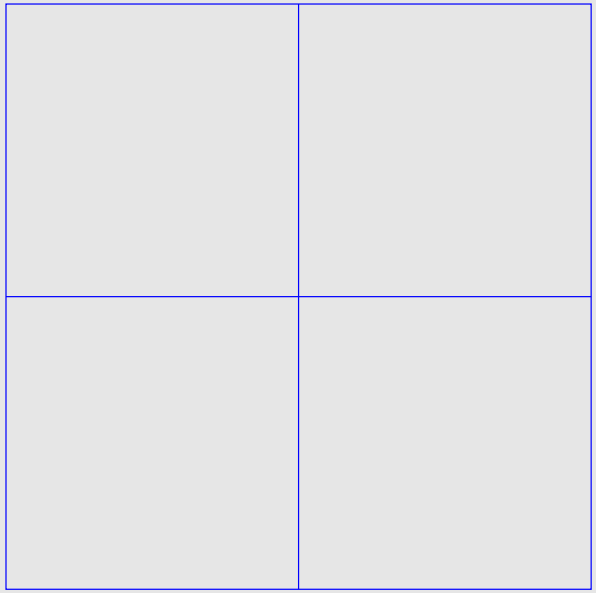
- meshing is easy, mesh is given by image data
- structured grid
- hence structured sparse system of equations
- computing entries more difficult (b/c basis functions contain complexity of structure)
- need more degrees of freedom than for classical unstructured grids
-

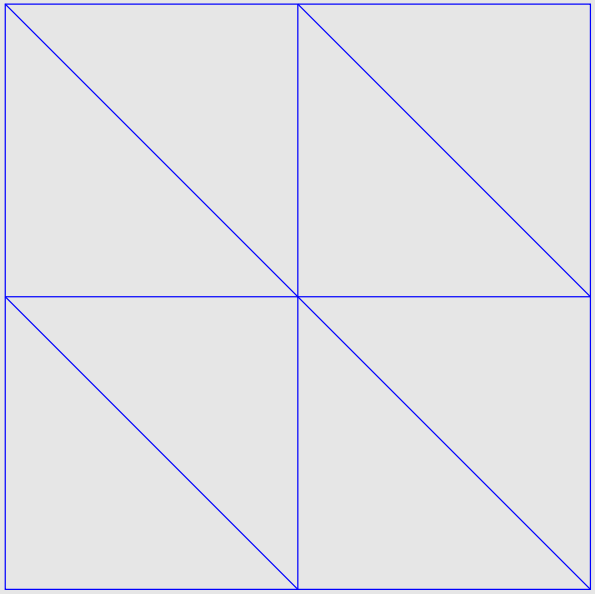
Composite Finite Elements: Advantages and Disadvantages

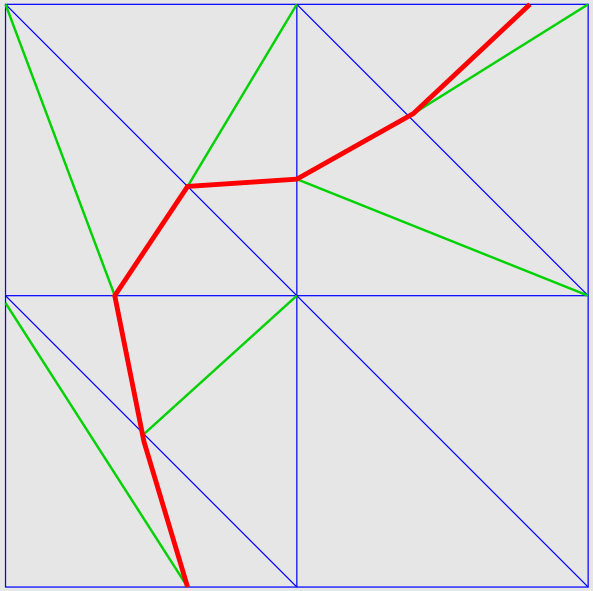
- meshing is easy, mesh is given by image data
- structured grid
- hence structured sparse system of equations
- computing entries more difficult (b/c basis functions contain complexity of structure)
- need more degrees of freedom than for classical unstructured grids
- multigrid methods can be used as solvers, hence computation is more efficient

Extension to 3D

69	247	248	248	148
187	255	255	255	230
243	255	255	255	250
209	255	255	255	220
79	253	253	253	146







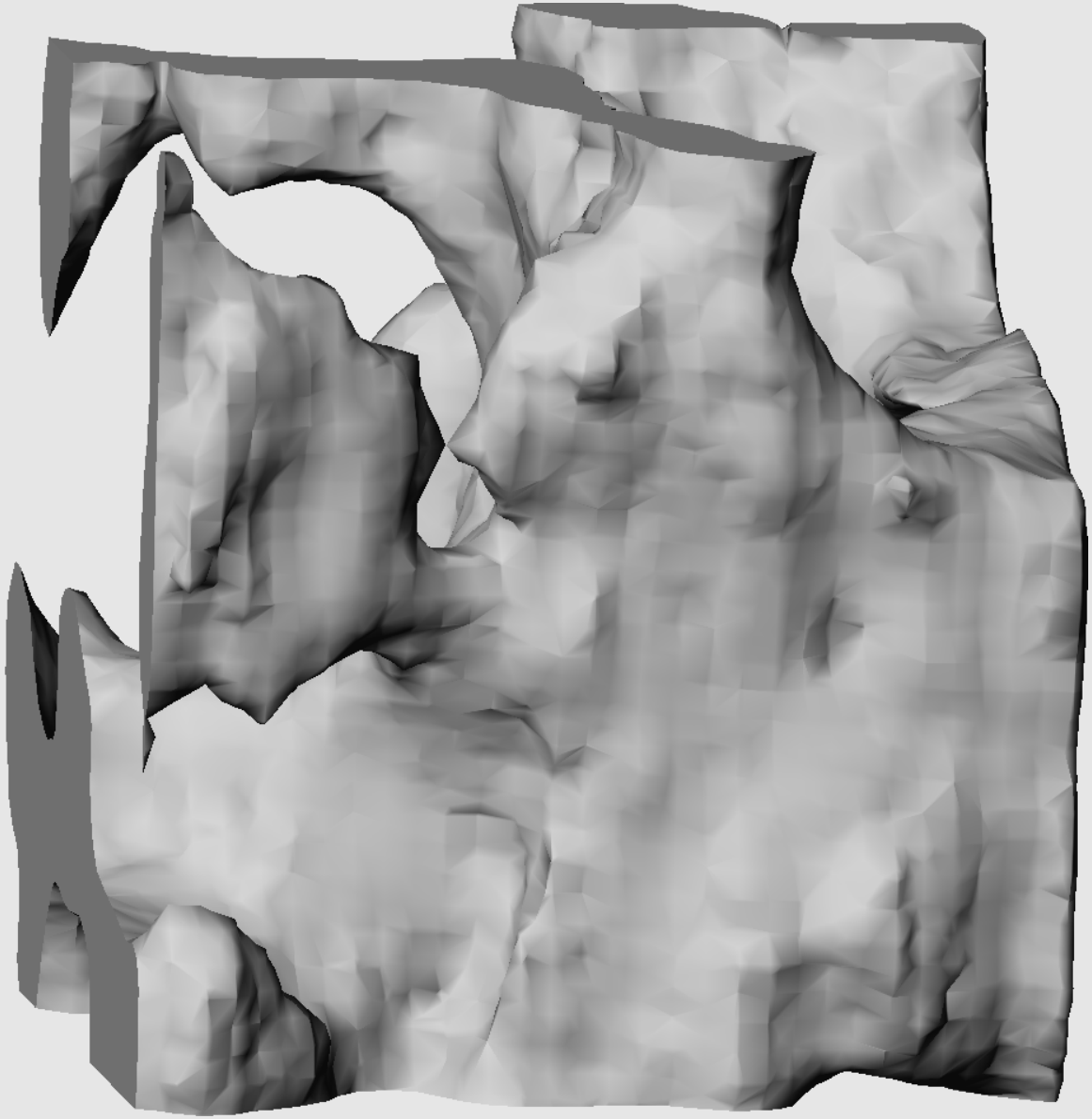
Classical Finite Elements
Linear Finite Elements in 1D

Finite Elements in 2D
Classification of Grids
Structured Grids
Unstructured Grids
Comparison

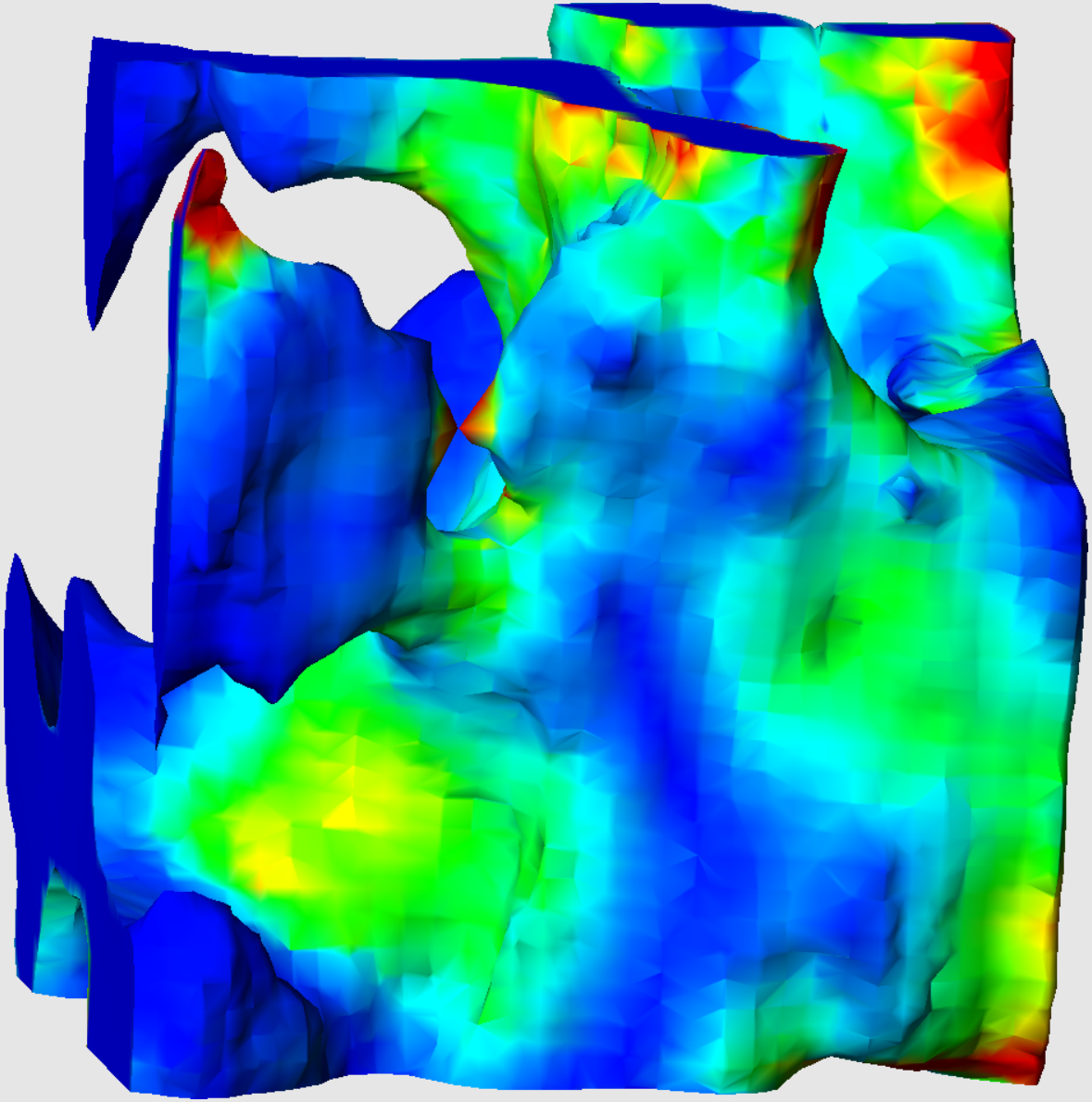
Composite Finite Elements
Geometry Extraction from Images
Construction of CFE Grids
CFE Basis Functions
Results (Elasticity)

Cooperations and Outlook

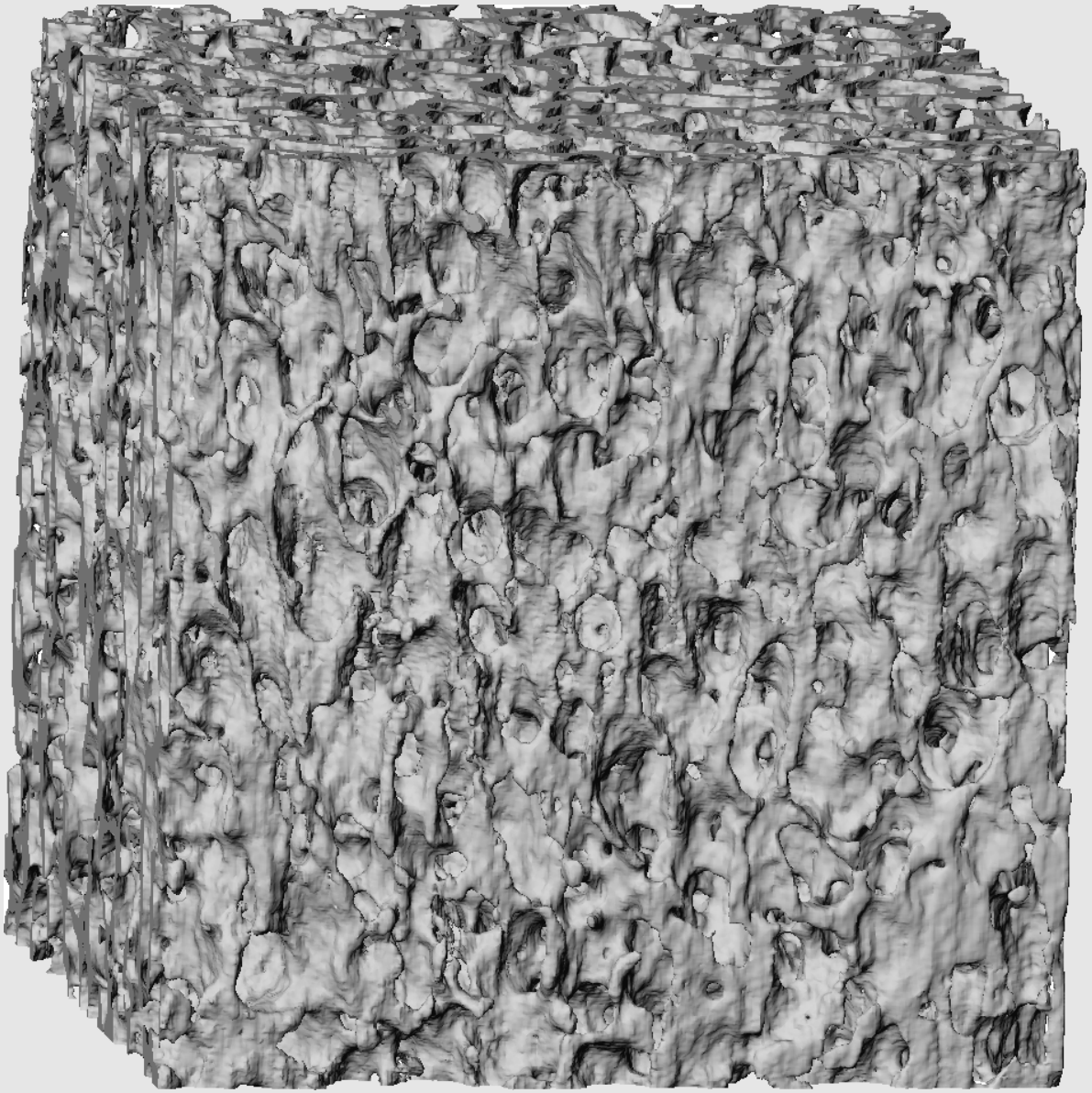
A Small Structure: Undeformed Geometry



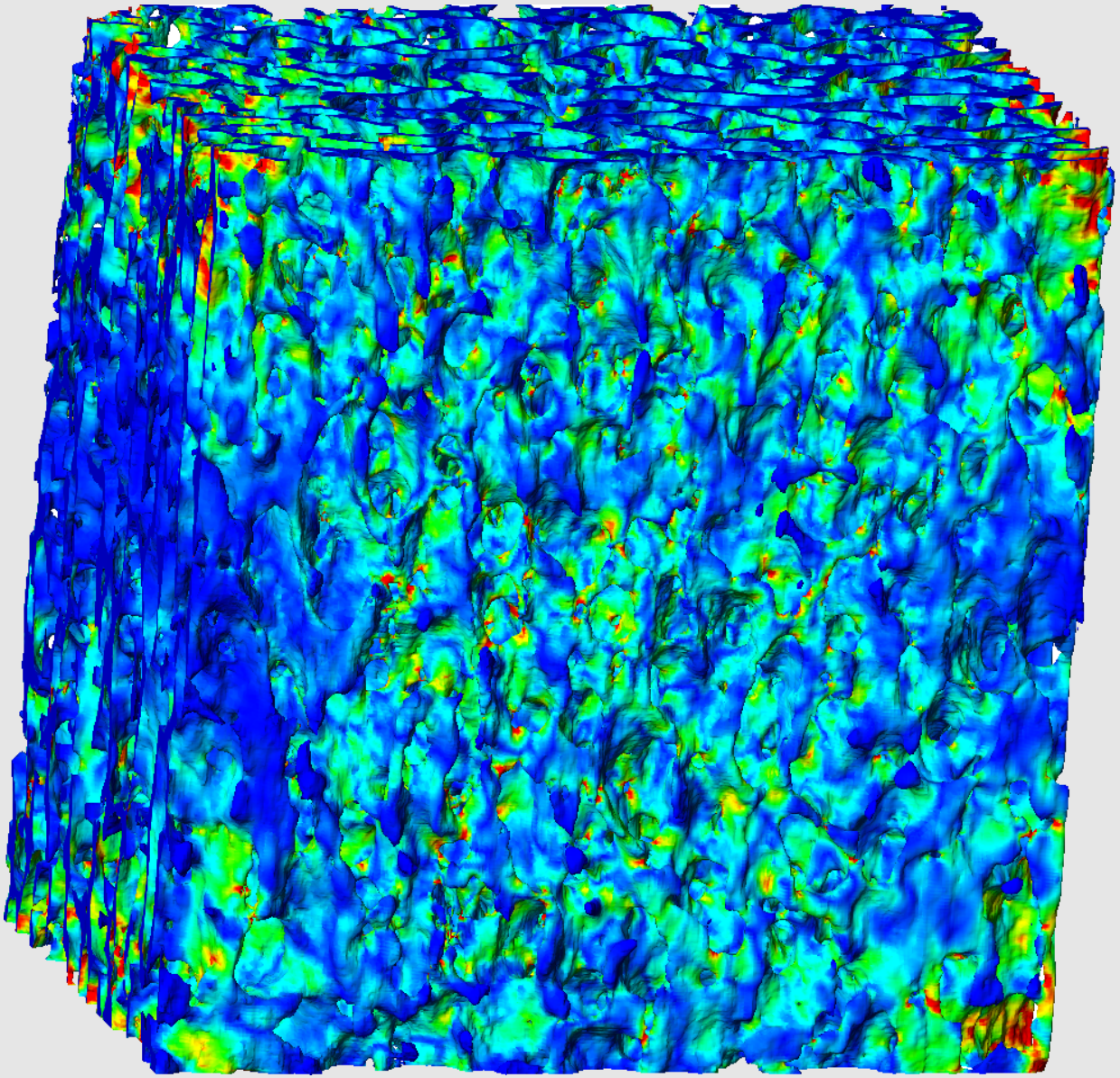
A Small Structure: Deformed Geometry



A Complicated Structure: Undeformed Geometry



A Complicated Structure: Deformed Geometry



Classical Finite Elements
- Linear Finite Elements in 1D

Finite Elements in 2D
- Classification of Grids
- Structured Grids
- Unstructured Grids
- Comparison

Composite Finite Elements
- Geometry Extraction from Images
- Construction of CFE Grids
- CFE Basis Functions
- Results (Elasticity)

Cooperations and Outlook

Cooperations

- CFE (theory) mainly developped by Stefan Sauter et. al. University of Zurich
- first implementation by Florian Liehr, Diplom thesis University of Duisburg
- CFE (implementation) in cooperation with Tobias Preusser, MeVis, University of Bremen
- experimental validation with Uwe Wolfram et. al., Institute for Orthopedic Research and Biomechaics, University of Ulm

Outlook

- composite materials: jumping coefficients

Outlook

- composite materials: jumping coefficients
- multigrid method: efficient solver

Outlook

- composite materials: jumping coefficients
- multigrid method: efficient solver
- elasticity: vector valued problem
-
-
-

Outlook

- composite materials: jumping coefficients
- multigrid method: efficient solver
- elasticity: vector valued problem
- computation of forces
-
-

Outlook

- composite materials: jumping coefficients
- multigrid method: efficient solver
- elasticity: vector valued problem
- computation of forces
- visualization
-

Outlook

- composite materials: jumping coefficients
- multigrid method: efficient solver
- elasticity: vector valued problem
- computation of forces
- visualization
- experimental validation

Mathematician at Work

Summary

Classical Finite Elements

- Linear Finite Elements in 1D

Finite Elements in 2D

- Classification of Grids
- Structured Grids
- Unstructured Grids
- Comparison

Composite Finite Elements

- Geometry Extraction from Images
- Construction of CFE Grids
- CFE Basis Functions
- Results (Elasticity)

Cooperations and Outlook

References

- Wolfgang Hackbusch, Stefan Sauter: Composite Finite Elements for the approximation of PDEs on domains with complicated micro-structures, Numerische Mathematik 75:447–472, 1997
- Florian Liehr: Ein effizienter Löser für elastische Mikrostrukturen, Diplomarbeit Universität Duisburg-Essen, 2004
- Tobias Preusser, Ole Schwen, Florian Liehr, Stefan Sauter, Martin Rumpf: An Approach to Image Based Computing, in preparation
- Siu-Wing Cheng, Tamal K. Dey, Herbert Edelsbrunner, Michael A. Facello,Shang-Hua Teng: Sliver Exudation, Journal of the ACM 47:883-904, 2000