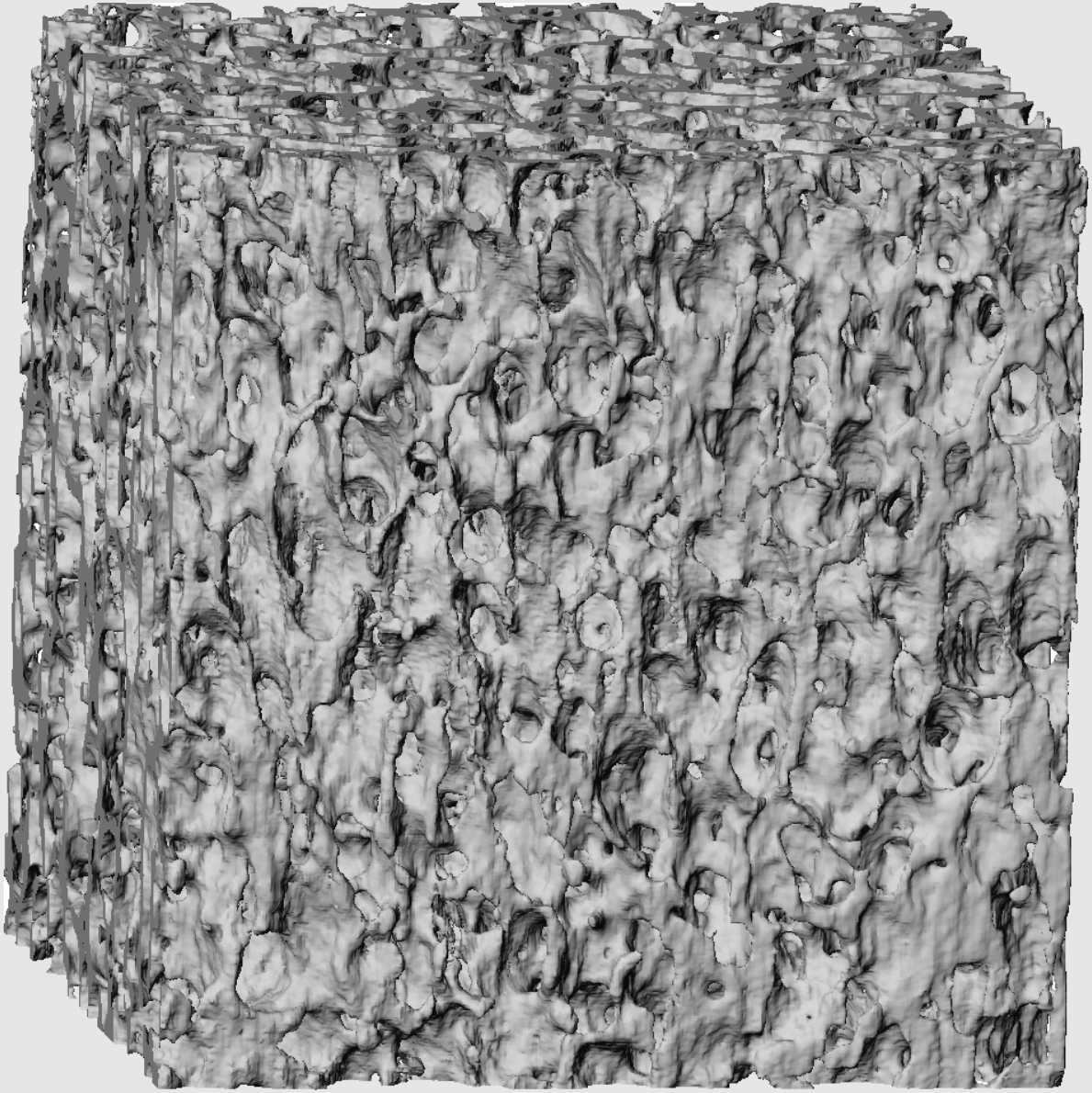
Composite Finite Elements for Complicated Microstructures

Lars Ole Schwen

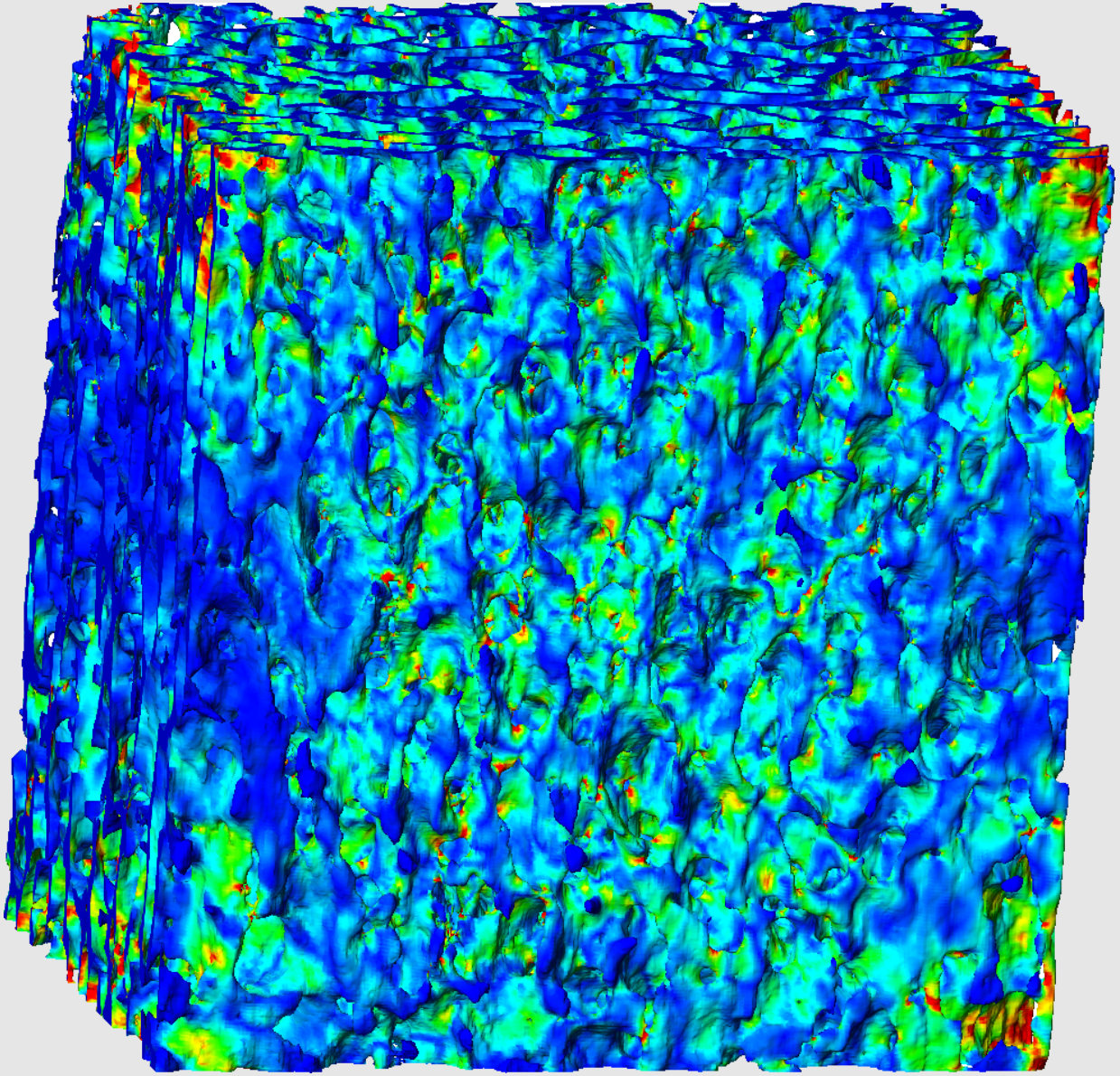
Institute for Numerical Simulation, University of Bonn

CdE-Seminar Siegburg, 2006-11-11

Goal: Complicated Microstructure



Goal: Complicated Microstructure, Elastically Deformed

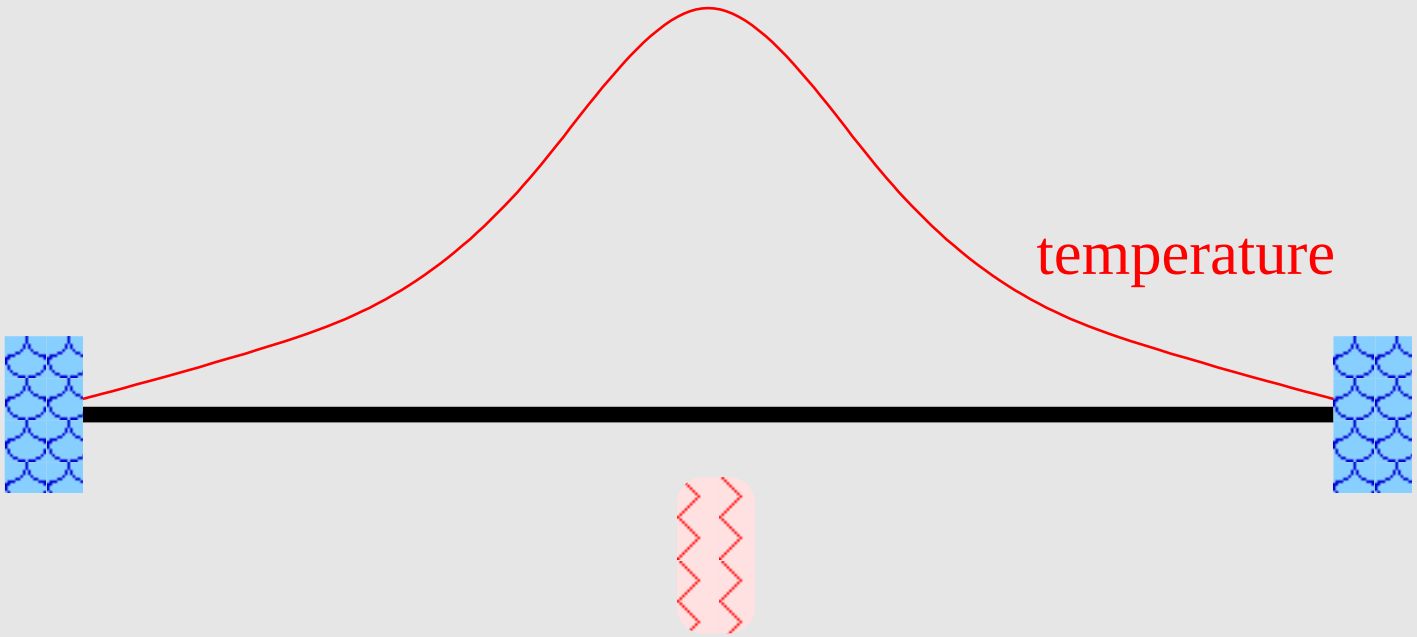


Outline

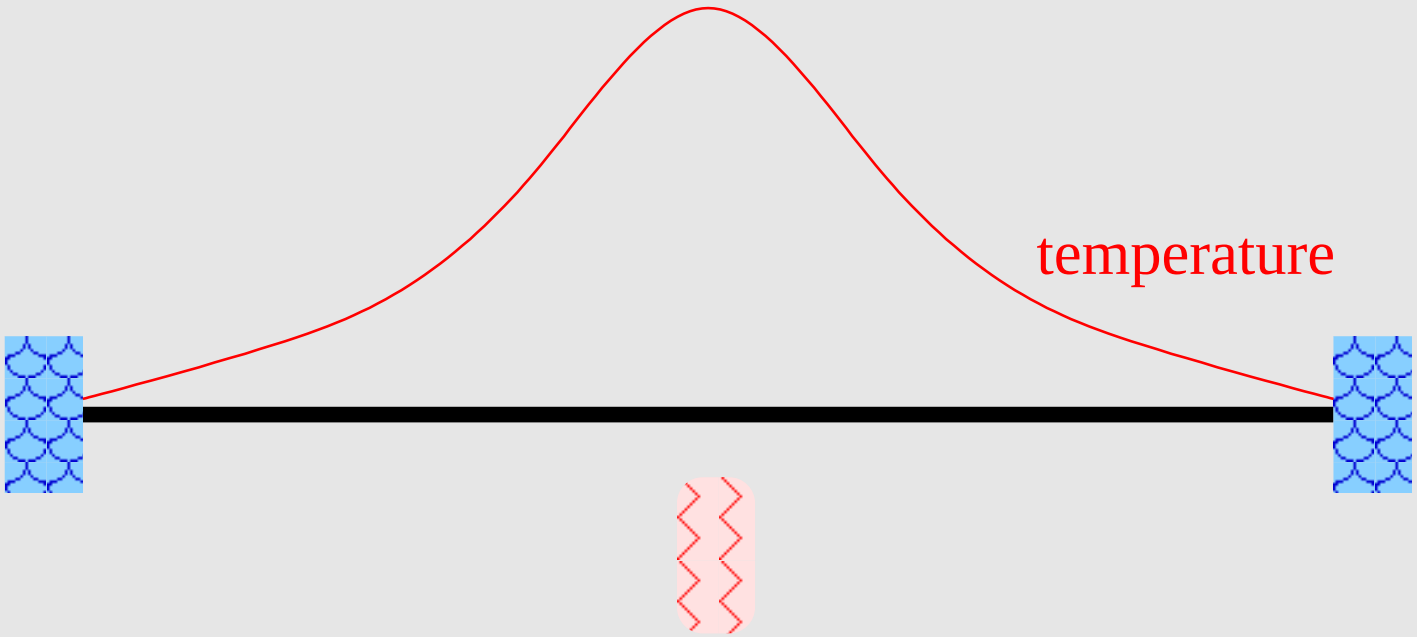
Finite Elements

- used to numerically solve partial differential equations
- powerful method
- efficient computations

Steady State Heat Conduction



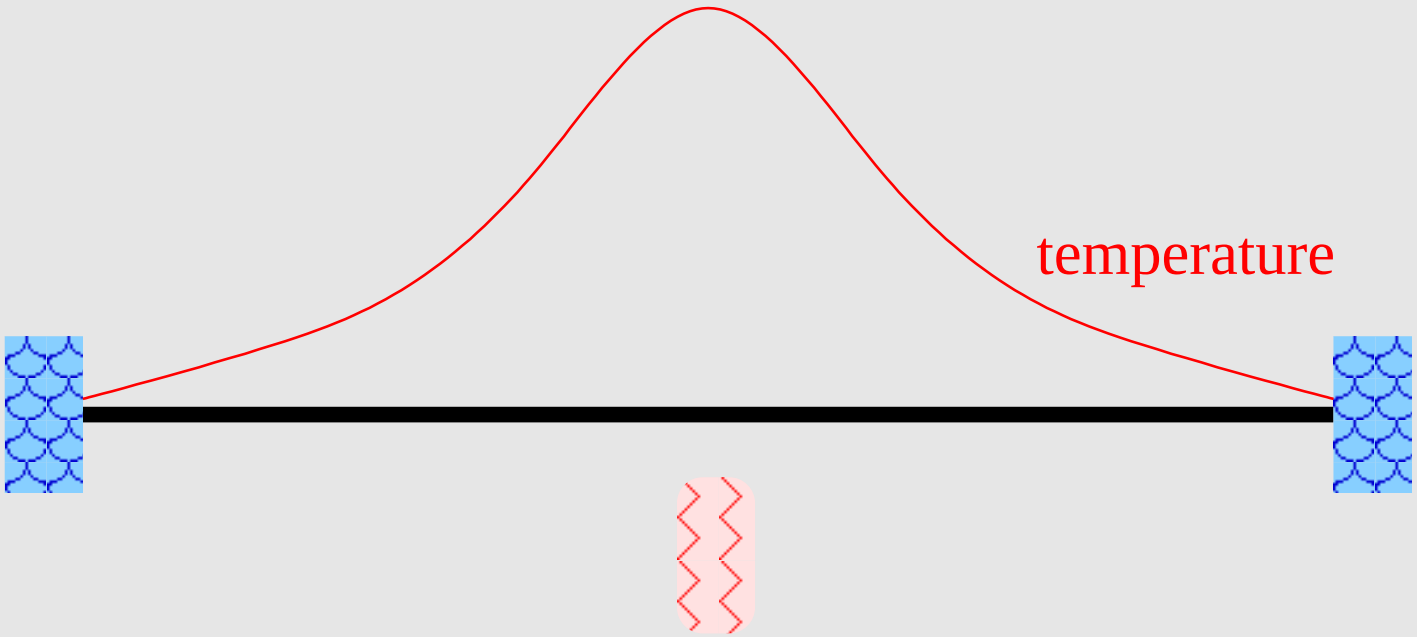
Steady State Heat Conduction



Find $u : \Omega = [0,1] \to \mathbb{R}$ so that

$$
\begin{aligned}
-u''(x) &= f(x) && \text{in } (0,1) \\
u(0) &= 0 \\
u(1) &= 0
\end{aligned}
$$

Steady State Heat Conduction



Find $u : \Omega = [0,1] \to \mathbb{R}$ so that

$$
\begin{aligned}
-u''(x) &= f(x) && \text{in } (0,1) \\
u(0) &= 0 \\
u(1) &= 0
\end{aligned}
$$

Do we need $u \in \mathcal{C}^2(\Omega)$?

Weak Equation

$$-u''(x) = f(x)$$

Weak Equation

$$
\begin{aligned}
& -u''(x) = f(x) \\
\iff \quad & \int_\Omega -u''\psi = \int_\Omega f\psi \quad \forall \psi \in H_0^1(\Omega)
\end{aligned}
$$

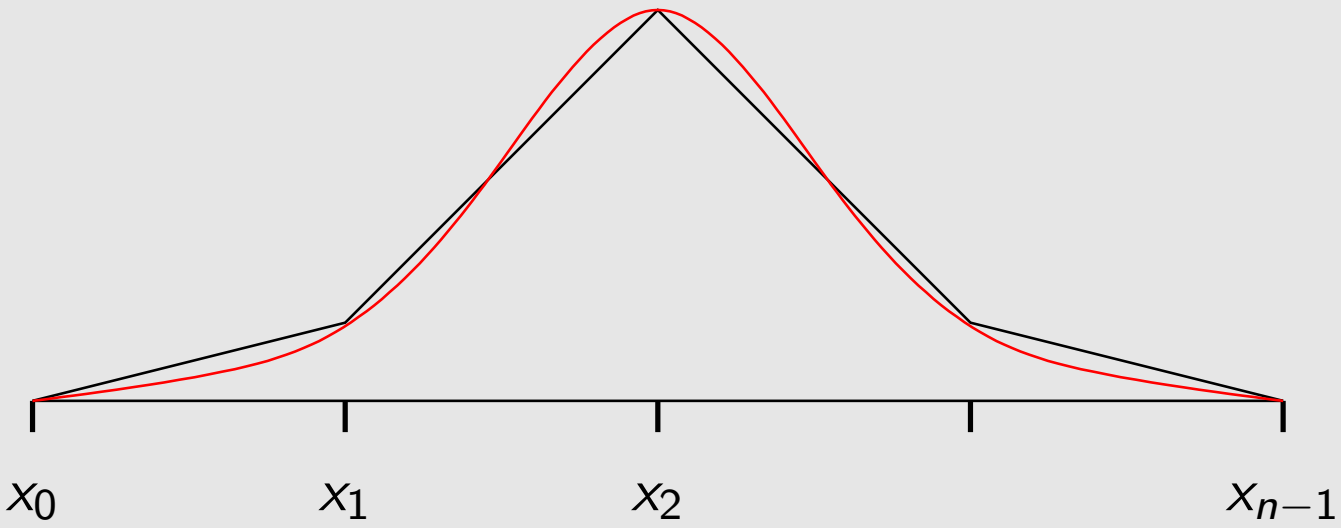
Weak Equation

$$
\begin{aligned}
& -u''(x) = f(x) \\
\iff \quad & \int_\Omega -u''\psi = \int_\Omega f\psi \quad \forall \psi \in H_0^1(\Omega) \\
\iff \quad & \int_\Omega u'\psi' + \left[u'(1)\psi(1) - u'(0)\psi(0)\right] = \int_\Omega f\psi \quad \forall \psi \in H_0^1(\Omega)
\end{aligned}
$$

Weak Equation

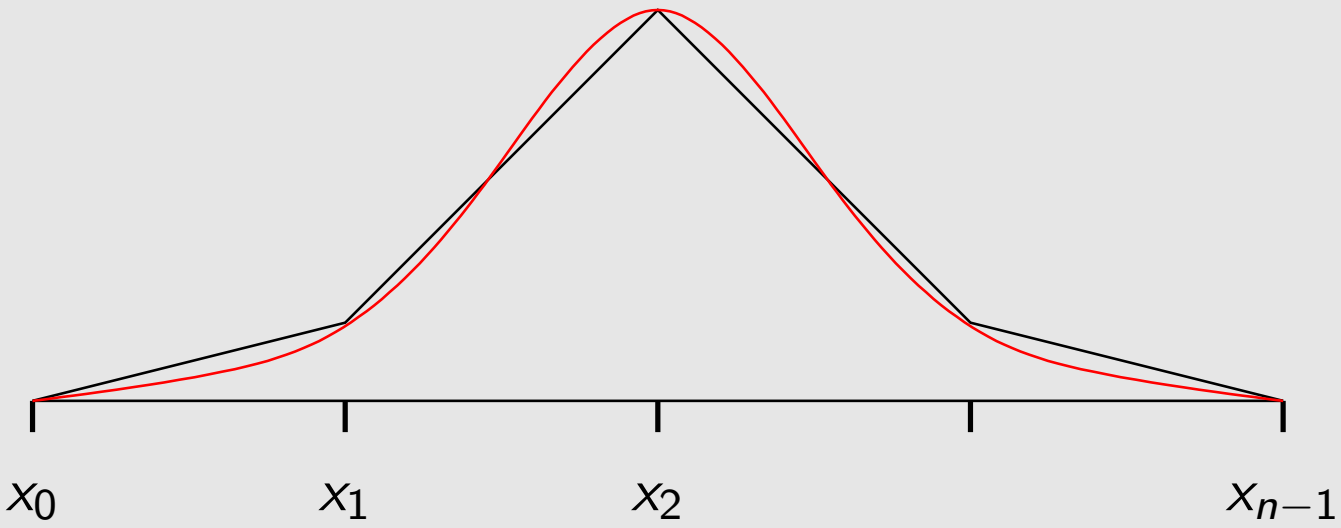
$$
\begin{aligned}
& -u''(x) = f(x) \\
\iff \quad & \int_\Omega -u''\psi = \int_\Omega f\psi \quad \forall \psi \in H_0^1(\Omega) \\
\iff \quad & \int_\Omega u'\psi' + \left[u'(1)\psi(1) - u'(0)\psi(0)\right] = \int_\Omega f\psi \quad \forall \psi \in H_0^1(\Omega) \\
\iff \quad & \int_\Omega u'\psi' = \int_\Omega f\psi \quad \forall \psi \in H_0^1(\Omega)
\end{aligned}
$$

Discretization



Want to approximate $u \approx u_h = \mathcal{I}[U]$

Discretization



Want to approximate $u \approx u_h = \mathcal{I}[U]$ with

- $U \in \mathbb{R}^n$,
- $U_i = u(x_i)$
- $\mathcal{I}$ piecewise linear interpolation

Discretization

Interpolation:

$$u_h(x) = \mathcal{I}[U](x) = \sum_{j=0}^{n-1} U_j \cdot \phi_j(x)$$

Discretization

Interpolation:

$$u_h(x) = \mathcal{I}[U](x) = \sum_{j=0}^{n-1} U_j \cdot \phi_j(x)$$

with

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

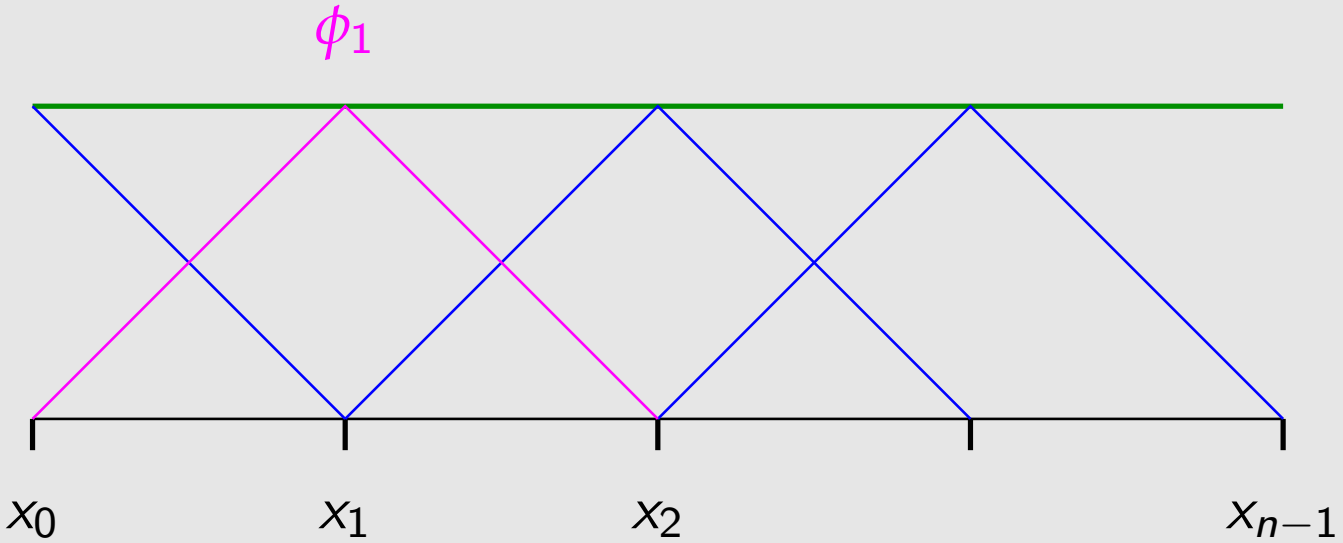
Discretization

Interpolation:

$$u_h(x) = \mathcal{I}[U](x) = \sum_{j=0}^{n-1} U_j \cdot \phi_j(x)$$

with

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



Discrete Equation

$$\int_\Omega u'\psi' = \int_\Omega f\psi \qquad\qquad \forall \psi \in H_0^1(\Omega)$$

Discretize u and f, take basis functions ψ of corresponding finite-dimensional subspace of $H_0^1(\Omega)$ only

Discrete Equation

$$\int_\Omega u'\psi' = \int_\Omega f\psi \qquad\qquad \forall \psi \in H_0^1(\Omega)$$

Discretize u and f, take basis functions ψ of corresponding finite-dimensional subspace of $H_0^1(\Omega)$ only

$$\int_\Omega \left(\sum_{j=0}^{n-1} U_j\phi_j\right)' \psi_i' \;=\; \int_\Omega \left(\sum_{j=0}^{n-1} F_j\phi_j\right)\psi_i \quad \forall i = 1,\ldots,n-2$$

Discrete Equation

$$\int_\Omega u'\psi' = \int_\Omega f\psi \qquad \forall \psi \in H_0^1(\Omega)$$

Discretize u and f, take basis functions ψ of corresponding finite-dimensional subspace of $H_0^1(\Omega)$ only

$$\begin{aligned}
\int_\Omega \left(\sum_{j=0}^{n-1} U_j\phi_j\right)' \psi_i' &= \int_\Omega \left(\sum_{j=0}^{n-1} F_j\phi_j\right) \psi_i \quad \forall i = 1,\ldots,n-2 \\
\sum_{j=0}^{n-1} U_j \int_\Omega \phi_j'\psi_i' &= \sum_{j=0}^{n-1} F_j \int_\Omega \phi_j\psi_i \quad \forall i = 1,\ldots,n-2
\end{aligned}$$

Discrete Equation

$$\int_\Omega u'\psi' = \int_\Omega f\psi \qquad \forall \psi \in H_0^1(\Omega)$$

Discretize u and f, take basis functions ψ of corresponding finite-dimensional subspace of $H_0^1(\Omega)$ only

$$\begin{aligned}
\int_\Omega \left(\sum_{j=0}^{n-1} U_j\phi_j\right)' \psi_i' &= \int_\Omega \left(\sum_{j=0}^{n-1} F_j\phi_j\right)\psi_i \quad \forall i = 1,\ldots,n-2 \\
\sum_{j=0}^{n-1} U_j \int_\Omega \phi_j'\psi_i' &= \sum_{j=0}^{n-1} F_j \int_\Omega \phi_j\psi_i \quad \forall i = 1,\ldots,n-2 \\
\sum_{j=0}^{n-1} U_j L_{ji} &= \sum_{j=0}^{n-1} F_j M_{ji} \quad \forall i = 1,\ldots,n-2
\end{aligned}$$

Discrete Equation

$$\int_\Omega u'\psi' = \int_\Omega f\psi \qquad \forall \psi \in H_0^1(\Omega)$$

Discretize u and f, take basis functions ψ of corresponding finite-dimensional subspace of $H_0^1(\Omega)$ only

$$\begin{aligned}
\int_\Omega \left(\sum_{j=0}^{n-1} U_j\phi_j\right)' \psi_i' &= \int_\Omega \left(\sum_{j=0}^{n-1} F_j\phi_j\right)\psi_i \quad \forall i = 1,\ldots,n-2 \\
\sum_{j=0}^{n-1} U_j \int_\Omega \phi_j'\psi_i' &= \sum_{j=0}^{n-1} F_j \int_\Omega \phi_j\psi_i \quad \forall i = 1,\ldots,n-2 \\
\sum_{j=0}^{n-1} U_j L_{ji} &= \sum_{j=0}^{n-1} F_j M_{ji} \quad \forall i = 1,\ldots,n-2
\end{aligned}$$

plus boundary conditions ($i = 0, i = n-1$):

$$L \cdot U = M \cdot F$$

Errors

- model, material parameters, . . . :

Errors

- model, material parameters, . . . : not our problem

Errors

- model, material parameters, . . . : not our problem
- weak problem

Errors

- model, material parameters, . . . : not our problem
- weak problem
- discretization

Errors

- model, material parameters, . . . : not our problem
- weak problem
- discretization
- inexact solution of system

Steps for a Finite Element Computation

1. geometry:
2. grid:
3. basis Functions:
4. system of Equations:
5. interpretation of Result:

Steps for a Finite Element Computation

1. geometry: rod
2. grid: equidistant points
3. basis Functions: hat functions
4. system of Equations: e. g. solved by CG
5. interpretation of Result: temperature

Steps for a Finite Element Computation

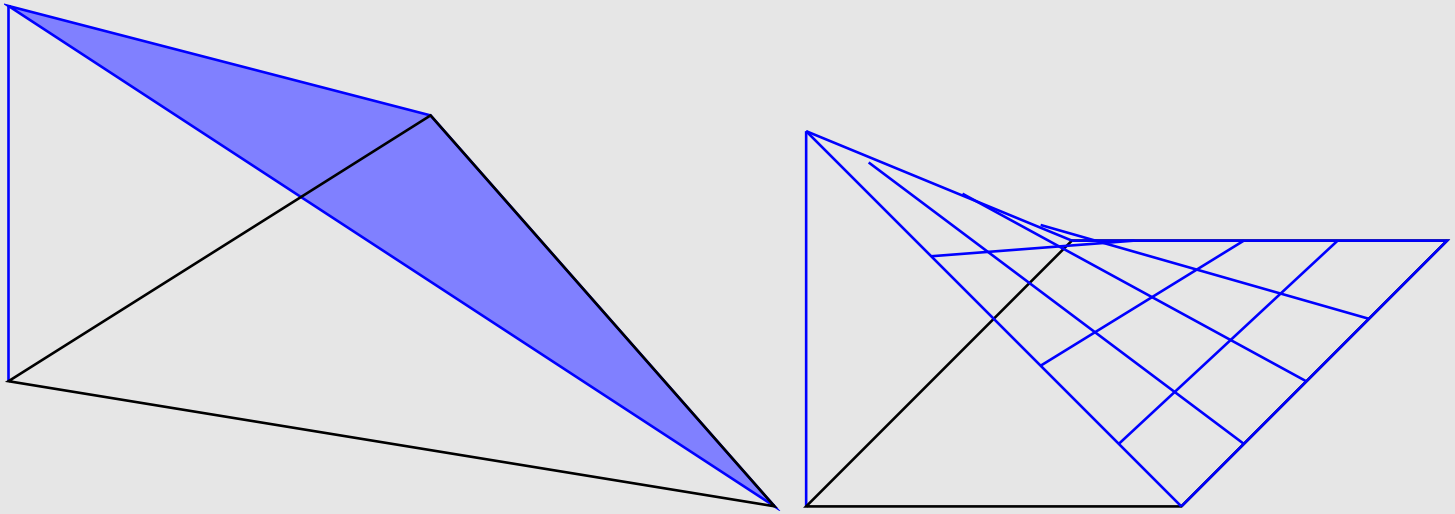
1. geometry: rod
2. grid: equidistant points
3. basis Functions: hat functions
4. system of Equations: e. g. solved by CG
5. interpretation of Result: temperature

We'll focus on 2 and 3.

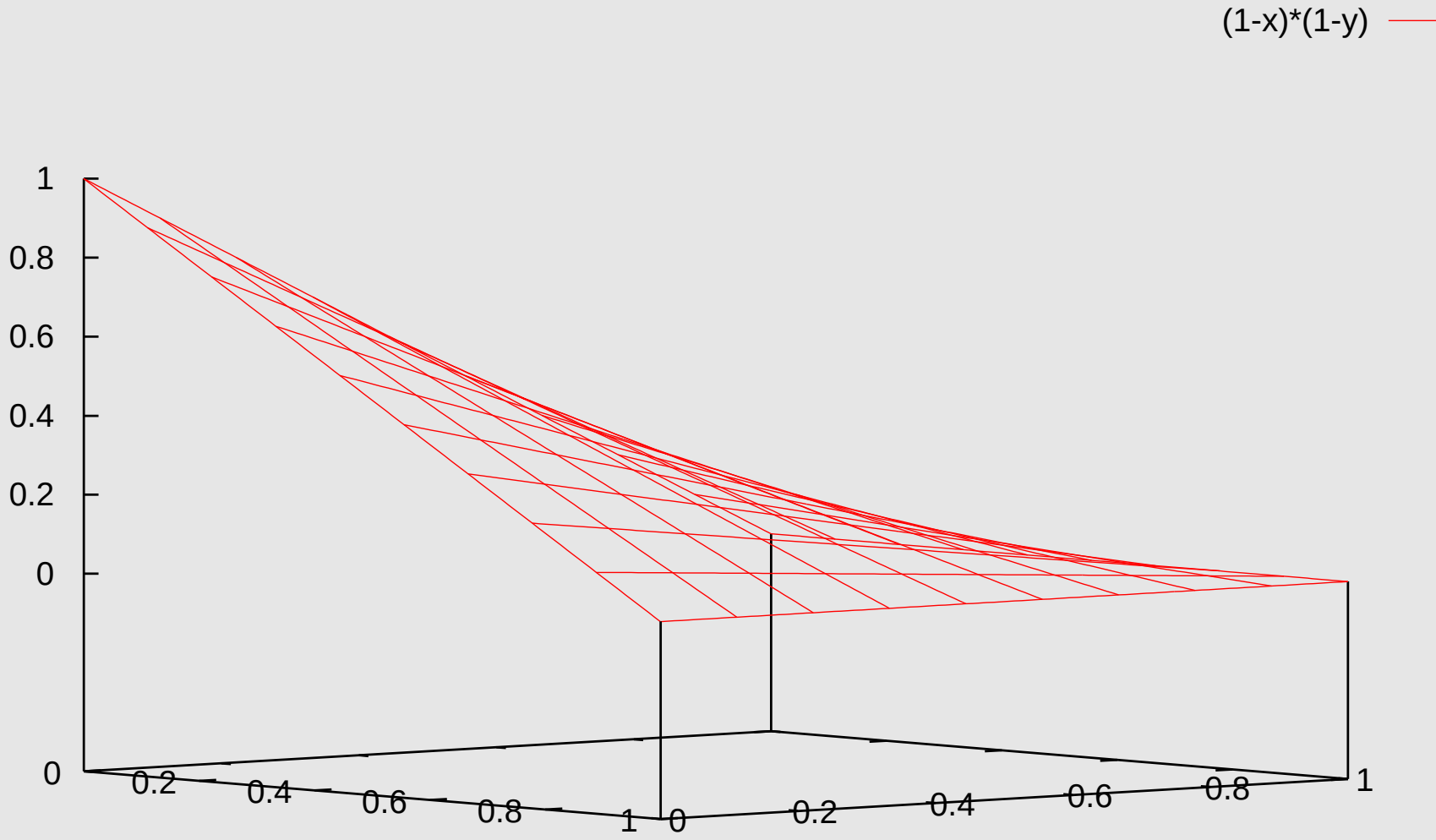
Generalization of “linear”

Two possibilities

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

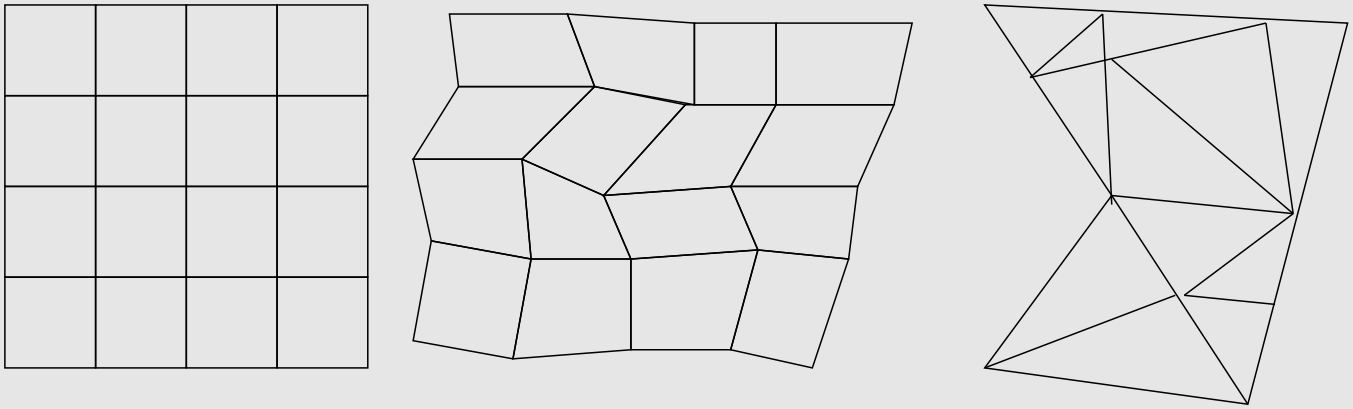


Bilinear Basis Function



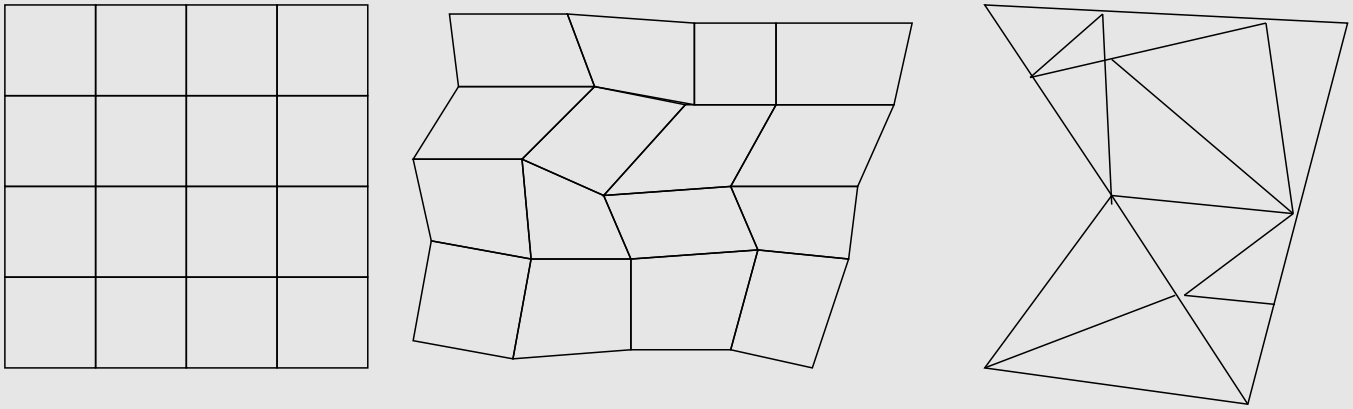
Classification of Grids

- uniform grids: equal grid cells



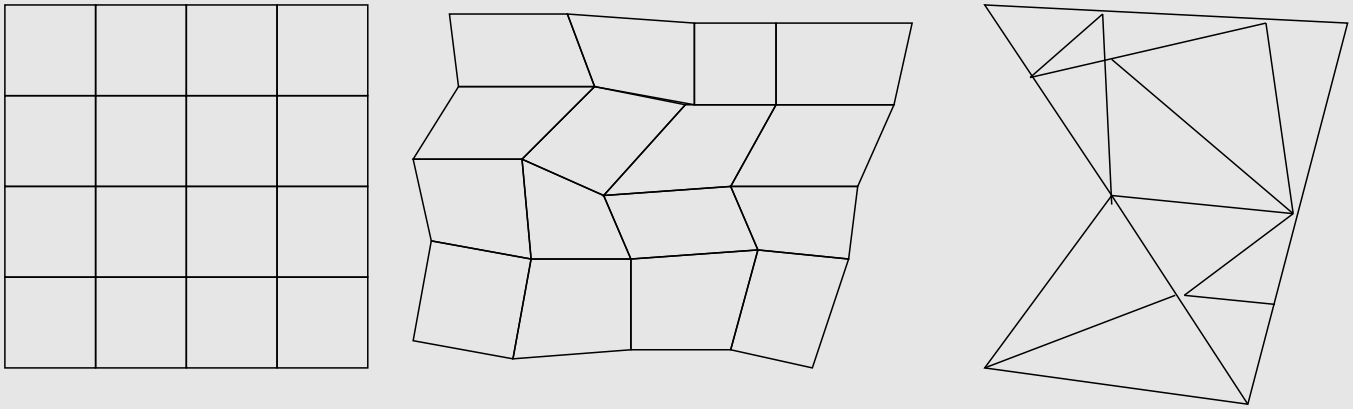
Classification of Grids

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



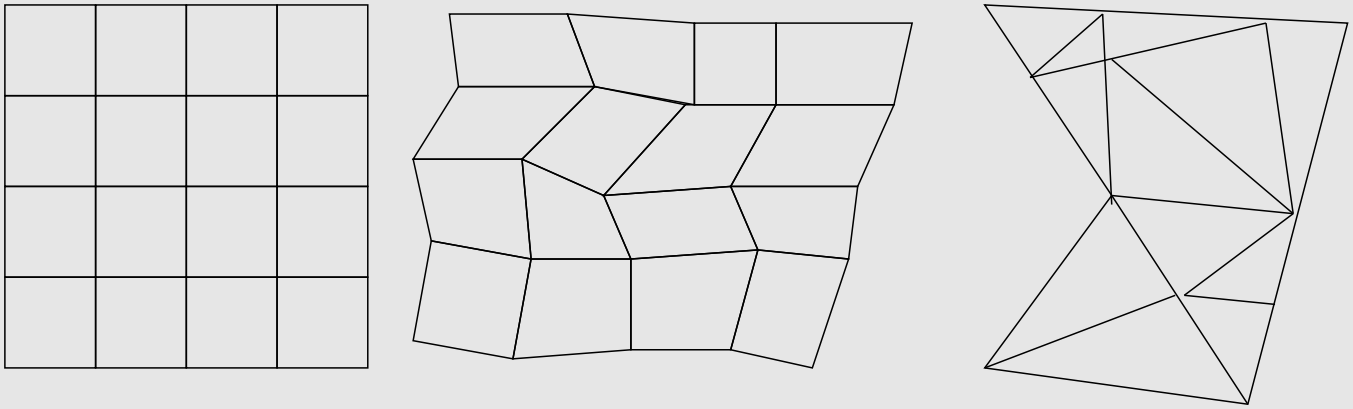
Classification of Grids

- 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

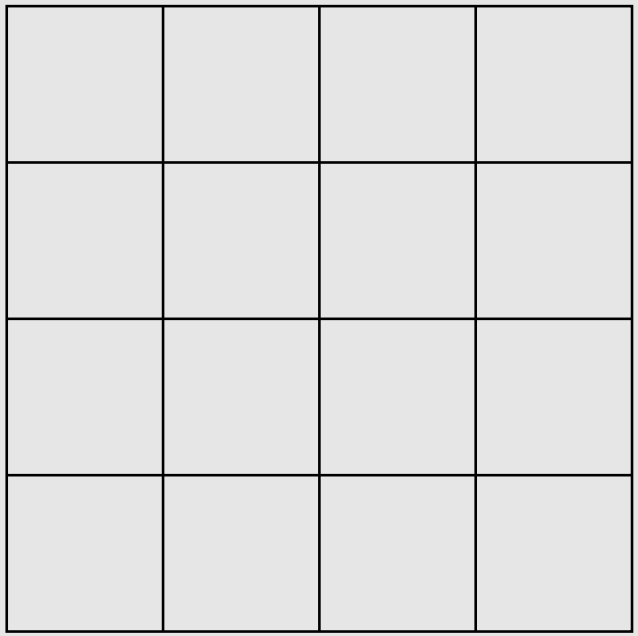


increasing geometric flexibility but increasing complexity

Characteristics of Grids

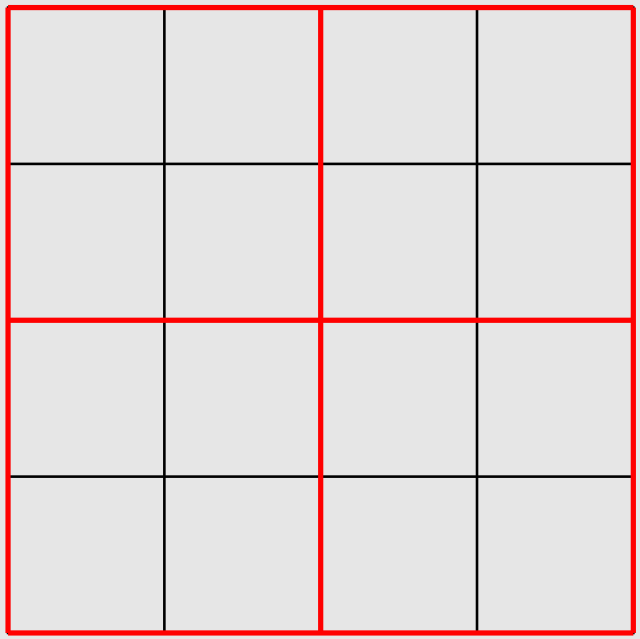
1. how to generate?
2. geometrically useful?
3. how to describe?
4. how to refine?
5. how to coarsen?
6. numerically "nice"? "nice" system of equations?

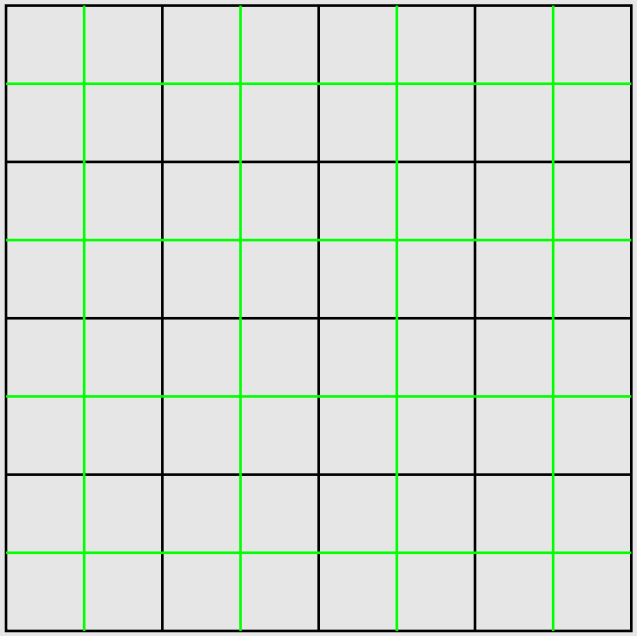
Structured Grids



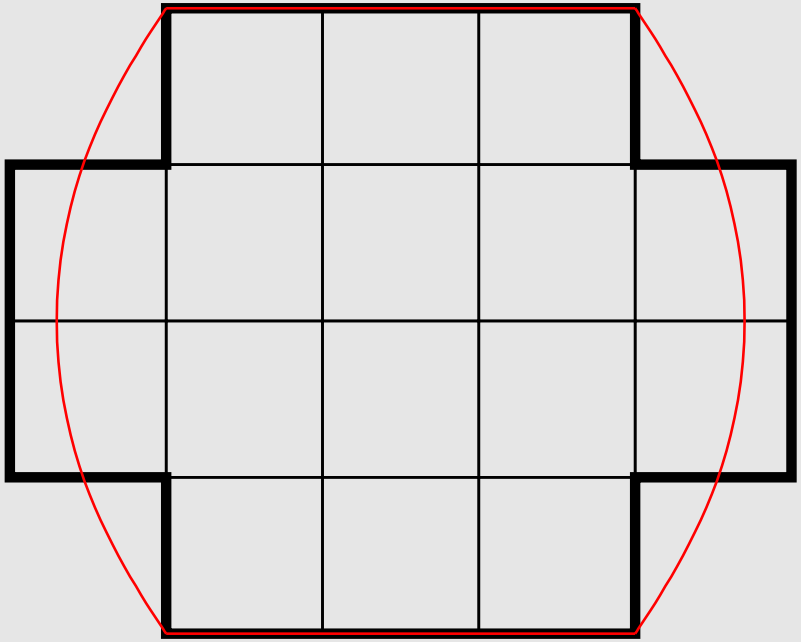
- easily generated and described implicitely
- sparse and structured system of equations

Structured Grids: Coarsening and Refinement

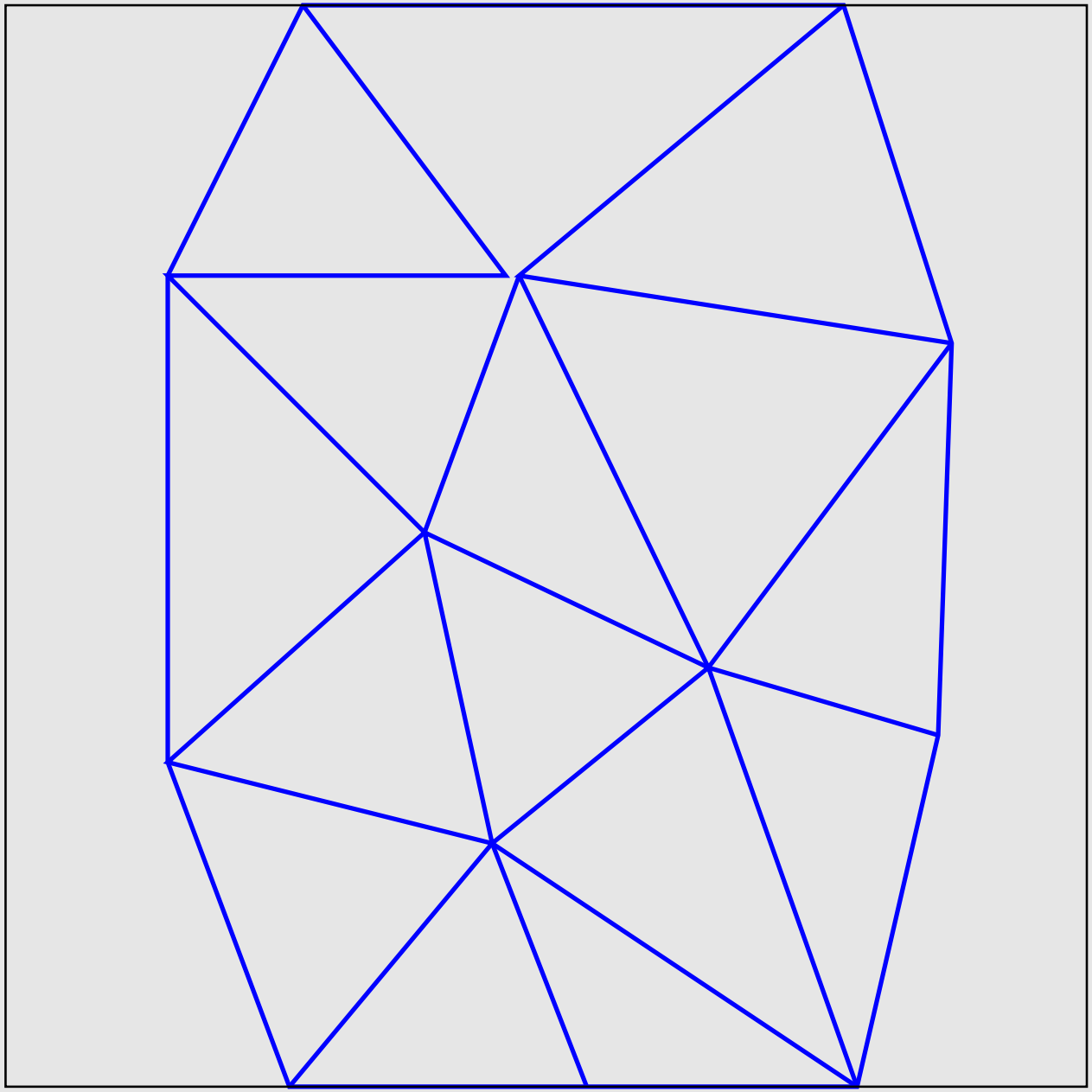




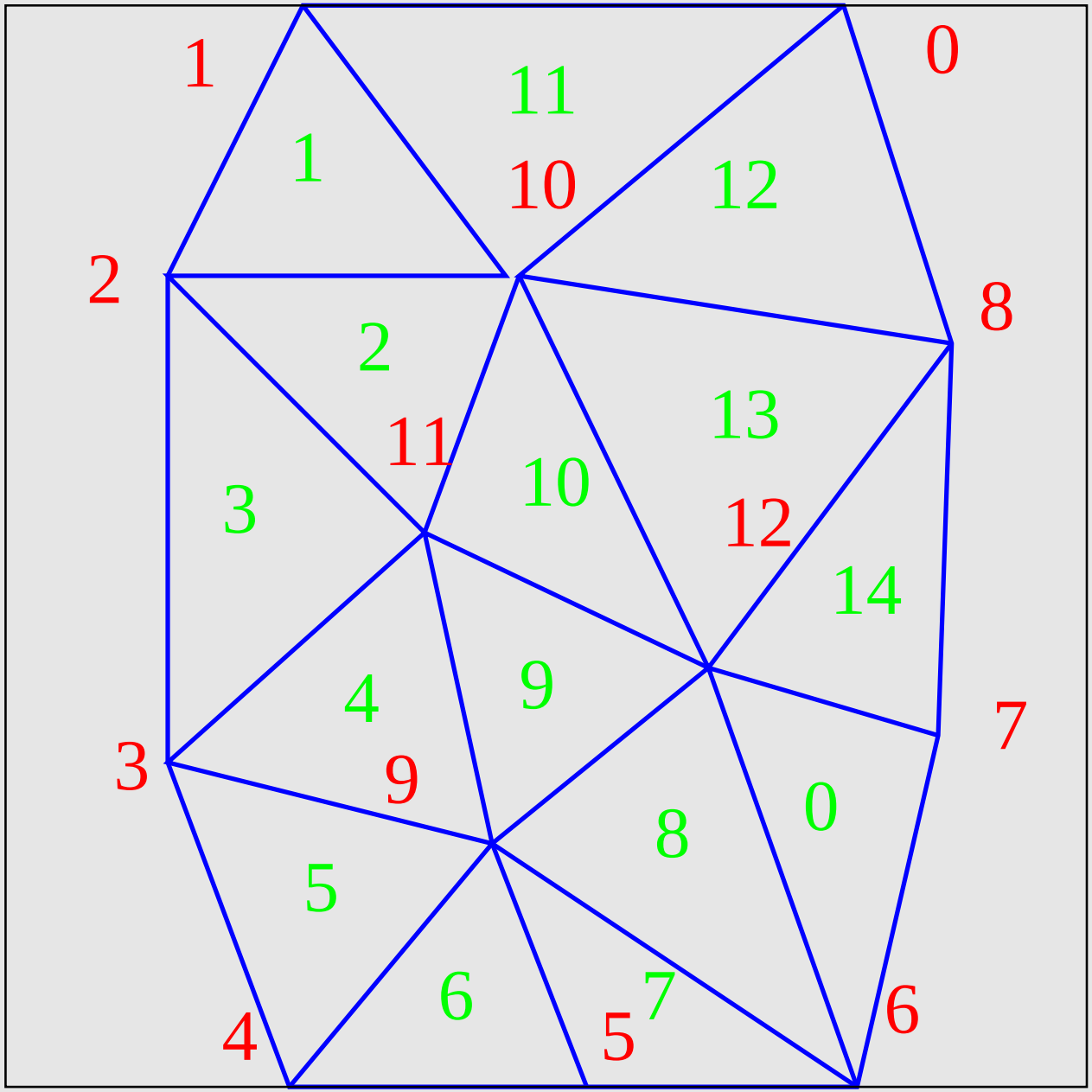
Structured Grids: Geometrically useful?



Unstructured Grids



Unstructured Grids: Storage



Unstructured Grids: Storage

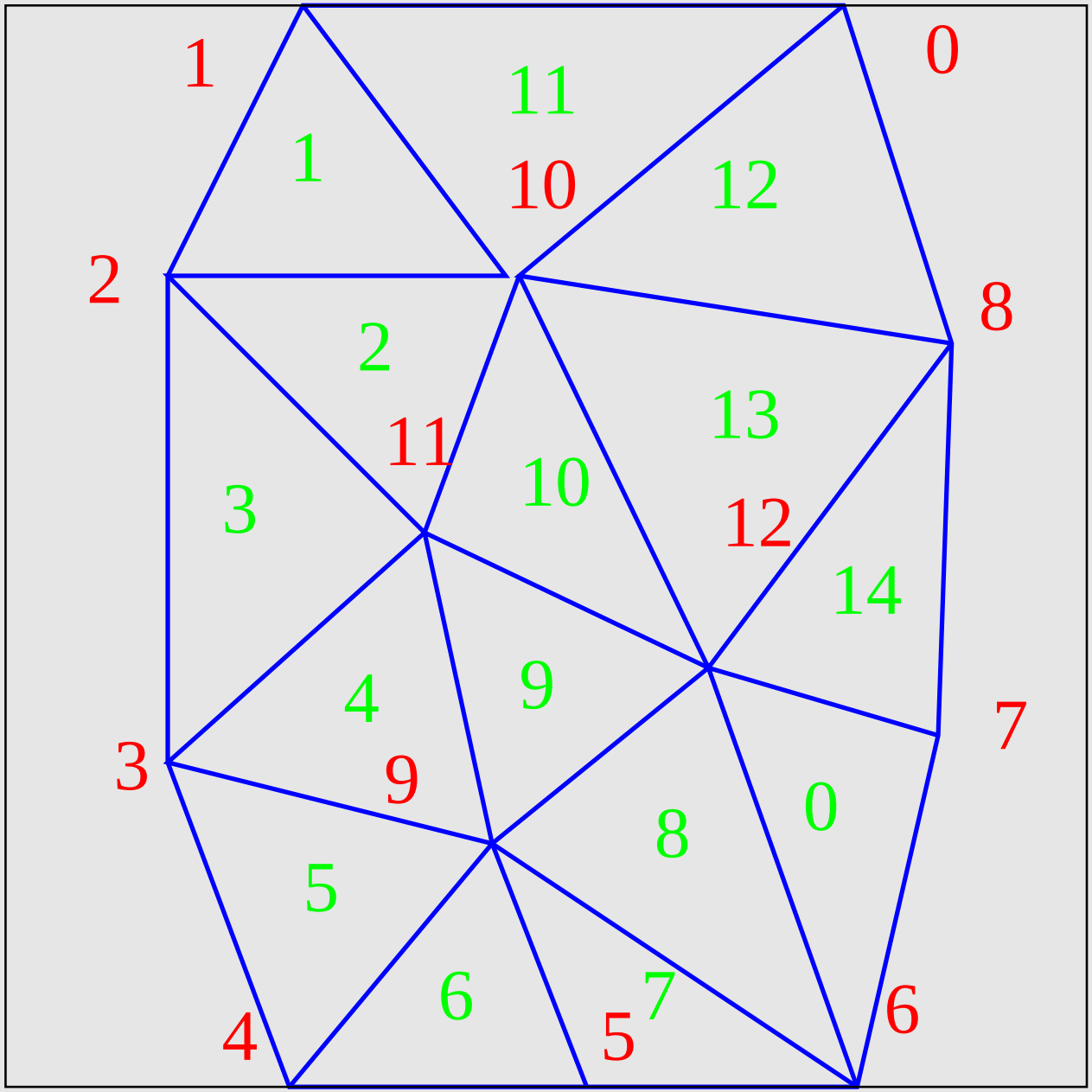
Node	x	y
0	0.8	1.0
1	0.3	1.0
2	0.2	0.75
3	0.2	0.25
4	0.3	0.0
5	0.5	0.0
6	0.8	0.0
7	0.85	0.4
8	0.9	0.65
9	0.45	0.2
10	0.5	0.7
11	0.4	0.55
12	0.65	0.4

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

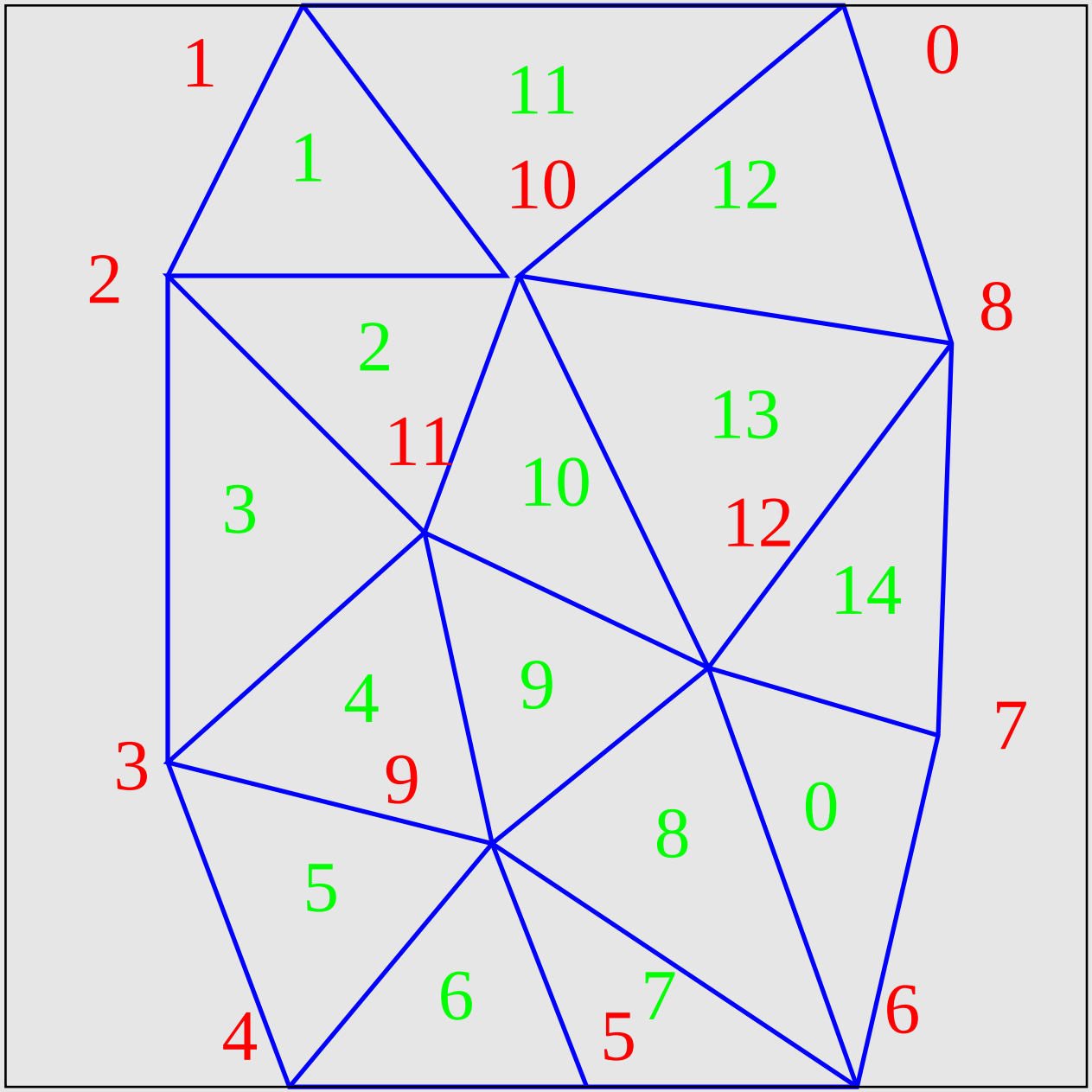
Implicite Information: Adjacent Triangles?



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?



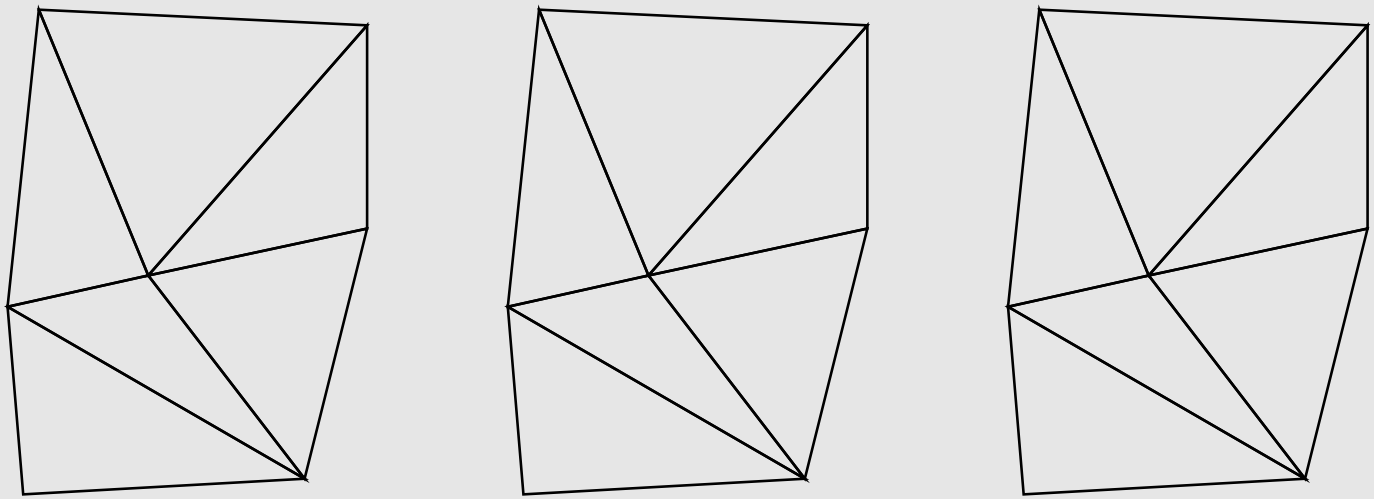
Implicite Information: Node Neighbors

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

$$
\begin{bmatrix}
* & * & & & & & & & * & & * & & \\
* & * & * & & & & & & & & * & & \\
 & * & * & * & & & & & & & * & * & \\
 & & * & * & * & & & & & * & & * & \\
 & & & * & * & * & & & & * & & & \\
 & & & & * & * & * & & & * & & & \\
 & & & & & * & * & * & & * & & & * \\
 & & & & & & * & * & * & & & & * \\
* & & & & & & & * & * & & * & & * \\
 & & & * & * & * & * & & & * & & * & * \\
* & * & * & & & & & & * & & * & * & * \\
 & & * & * & & & & & & * & * & * & * \\
 & & & & & & * & * & * & * & * & * & *
\end{bmatrix}
$$

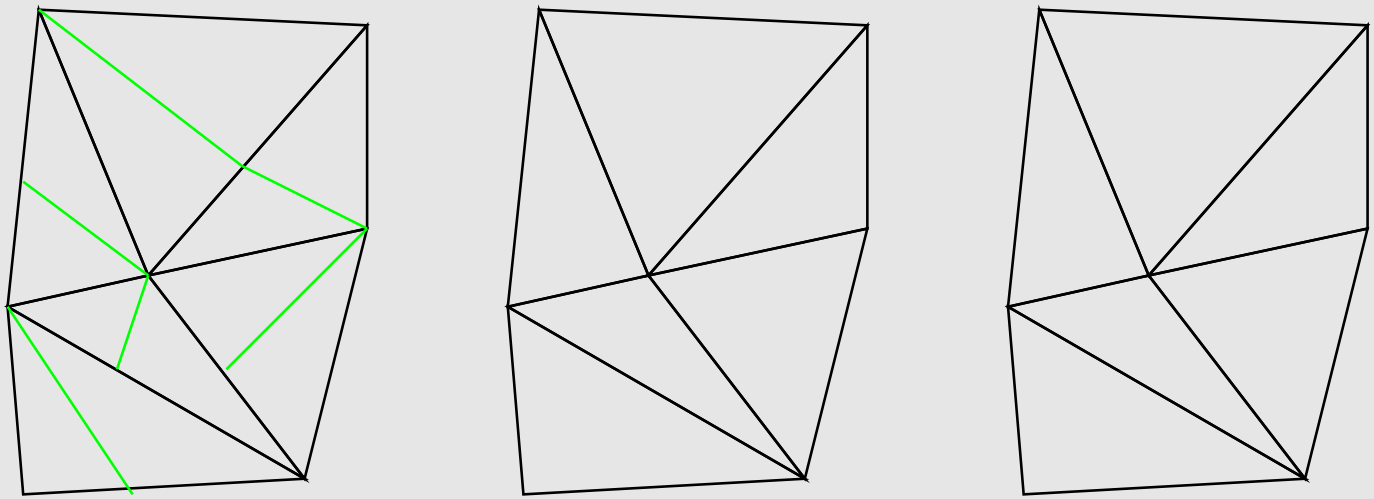
Unstructured Grids: Refinement



Three canonical methods for refinement:

- bisection
- trisection
- quadrisection

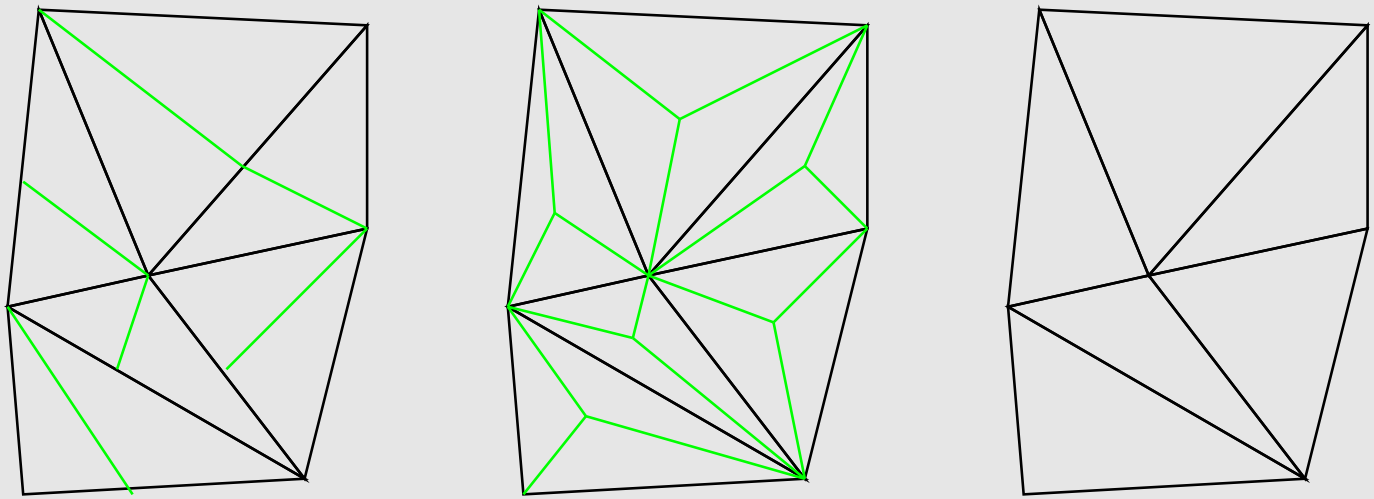
Unstructured Grids: Refinement



Three canonical methods for refinement:

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

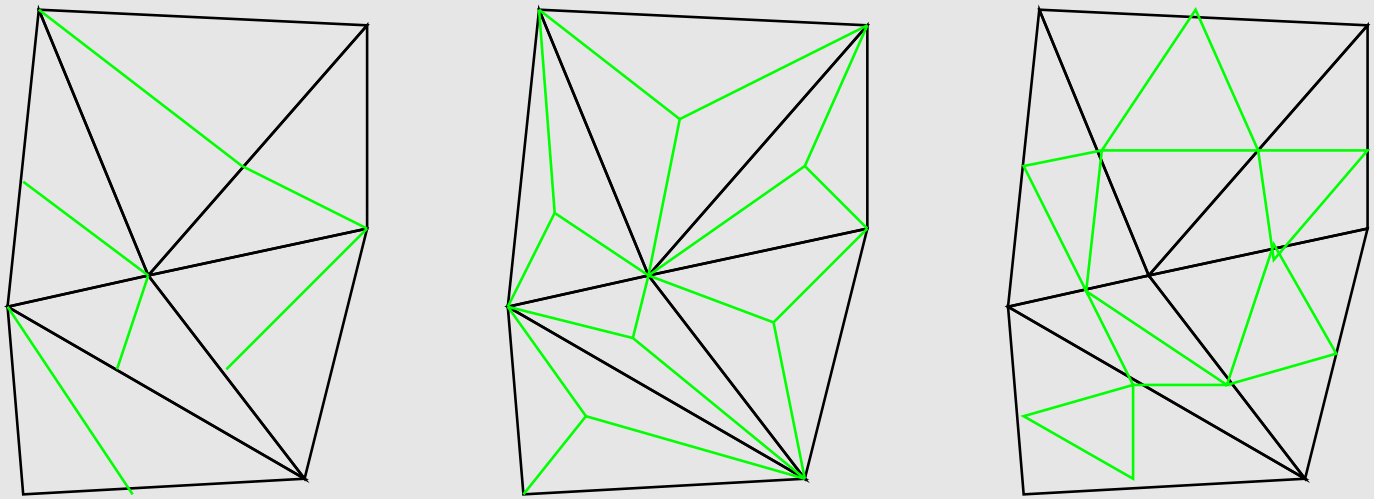
Unstructured Grids: Refinement



Three canonical methods for refinement:

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

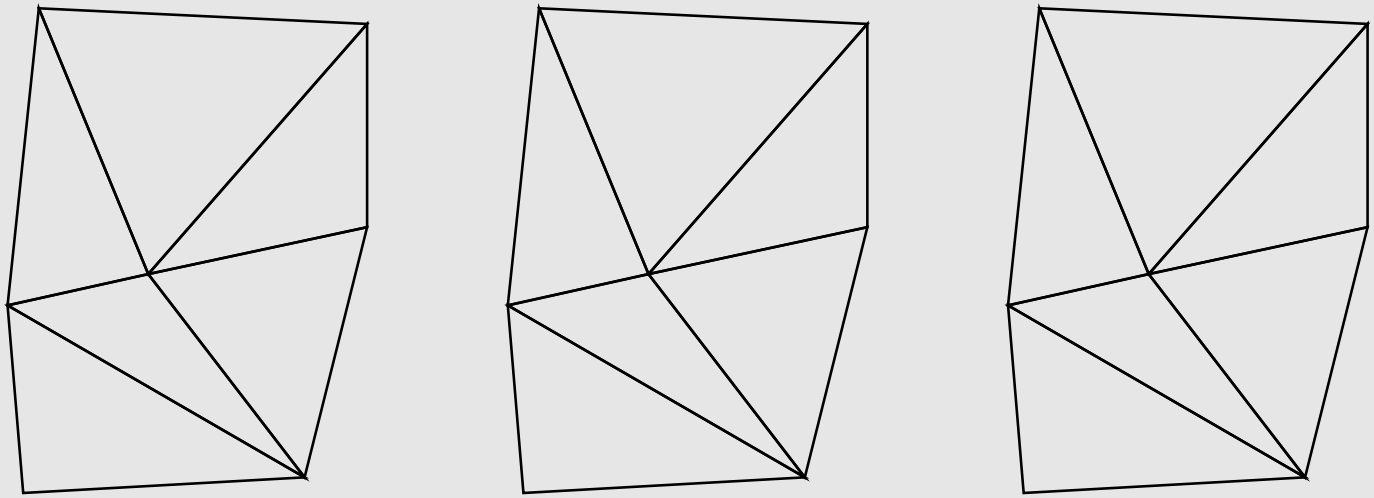
Unstructured Grids: Refinement



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

Quality Measures of Unstructured Grids

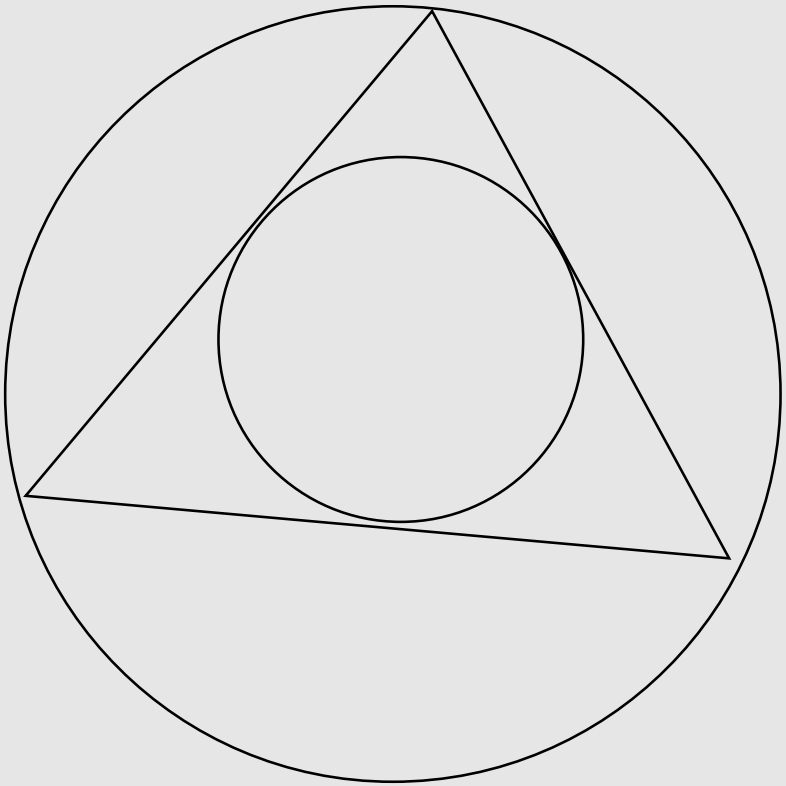
Different authors use different measures of non-degeneracy:

- angles bounded away from 0°
- angles bounded away from 90°

Quality Measures of Unstructured Grids

Different authors use different measures of non-degeneracy:

- angles bounded away from 0°
- angles bounded away from 90°
- bounded ratio longest / shortest edge
- bounded ratio outer / inner radius



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

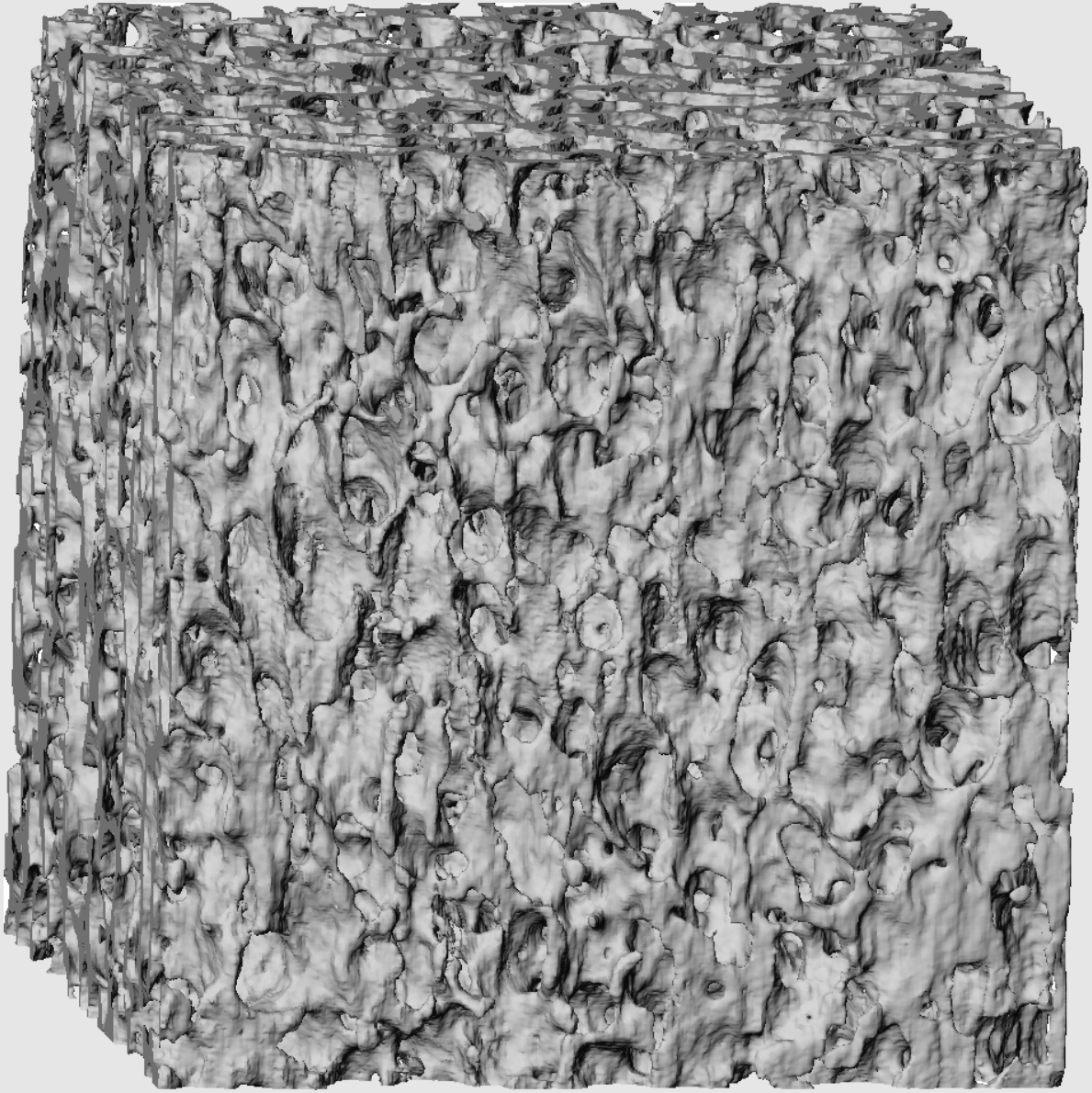
Extension to 3D

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

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

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:

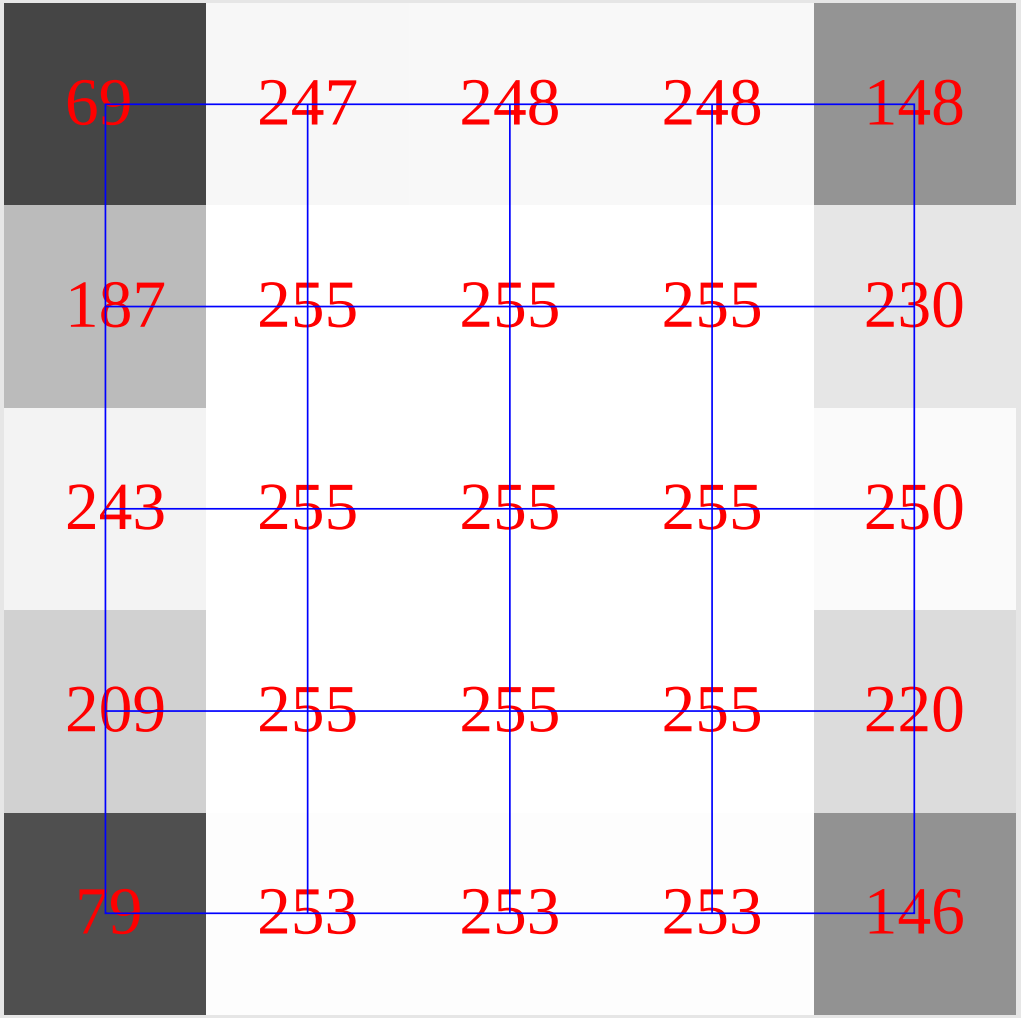
- classically, use complicated mesh with simple basis functions
- CFE: use simple mesh with complicated basis functions
- both are immediately given by the image data (μCT scan)

Grey Images: CT-Scans

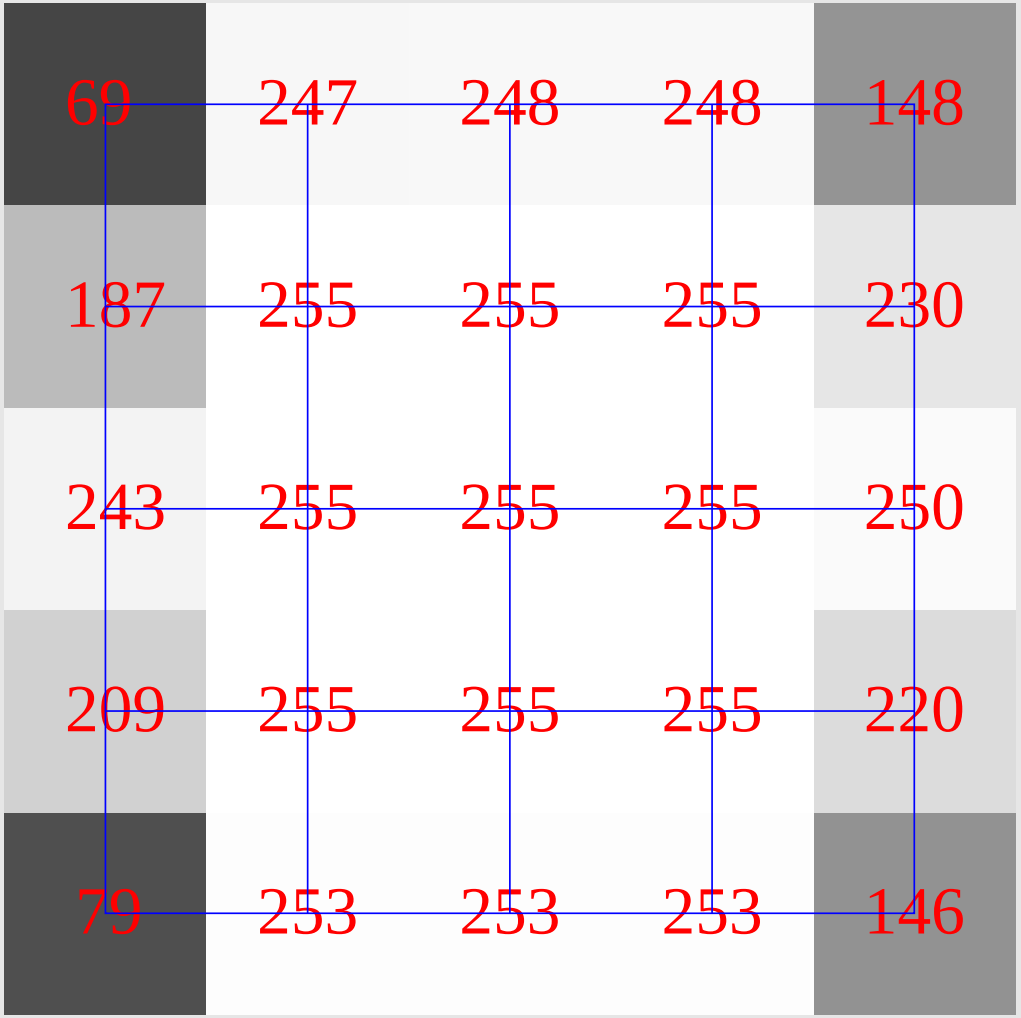
69	247	248	248	148
187	255	255	255	230
243	255	255	255	250
209	255	255	255	220
79	253	253	253	146

grey values represent X-ray attenuation

A Natural Grid for Grey Images

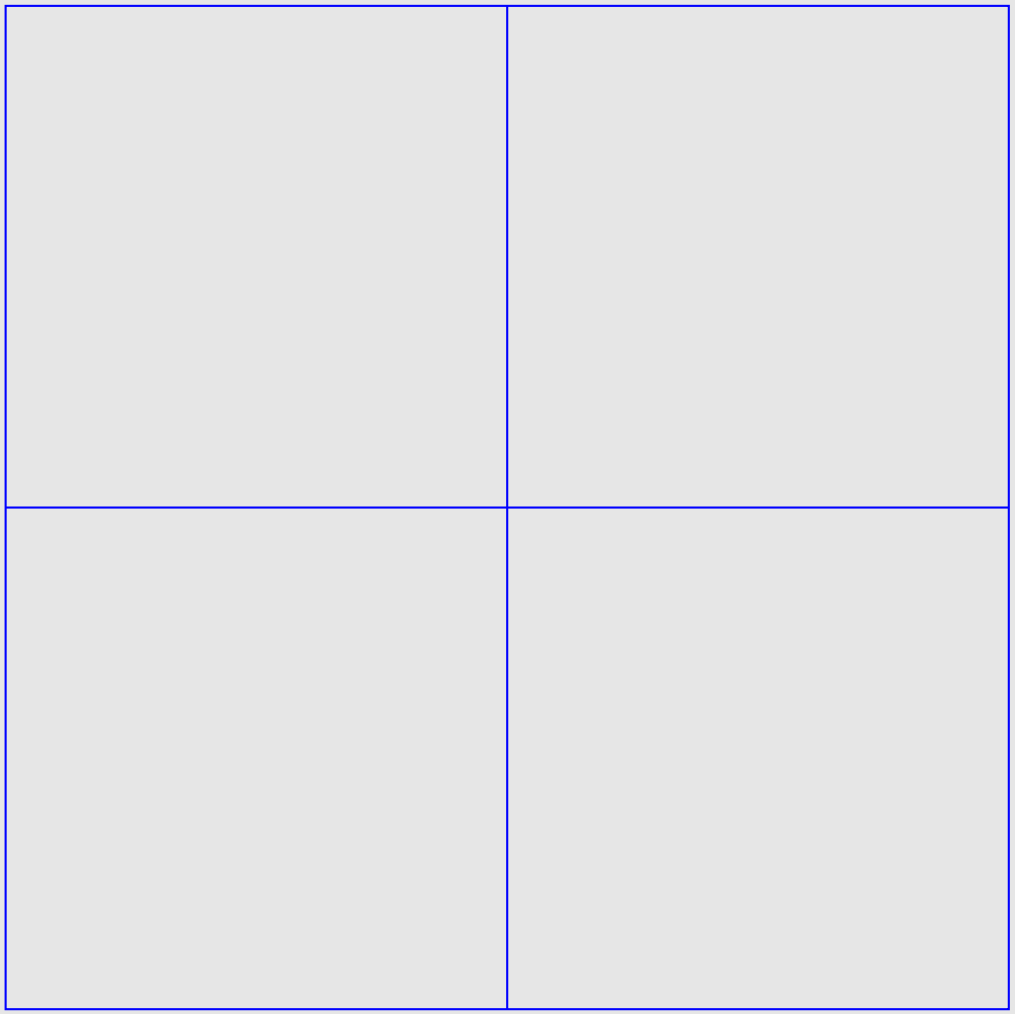


A Natural Grid for Grey Images



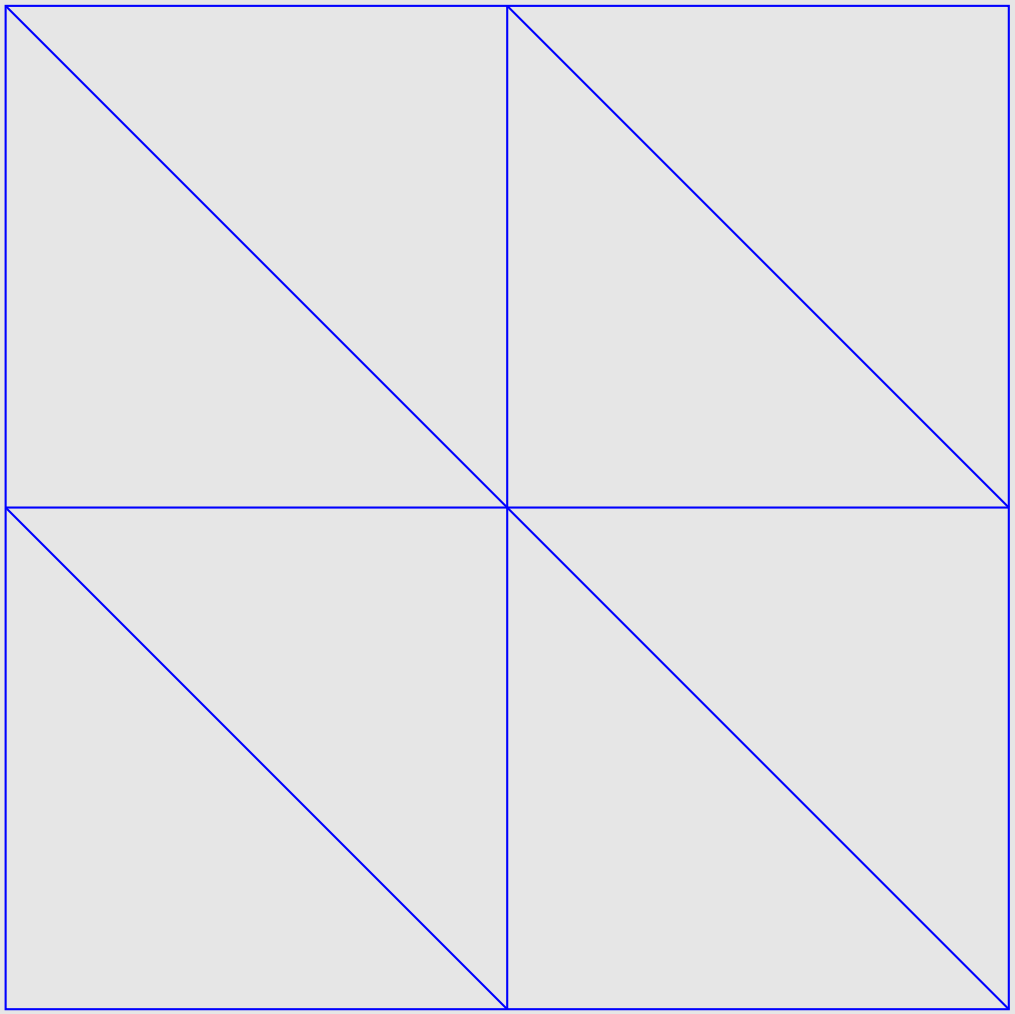
object boundary (240) can be recovered: keep these points

Construction of CFE Grids



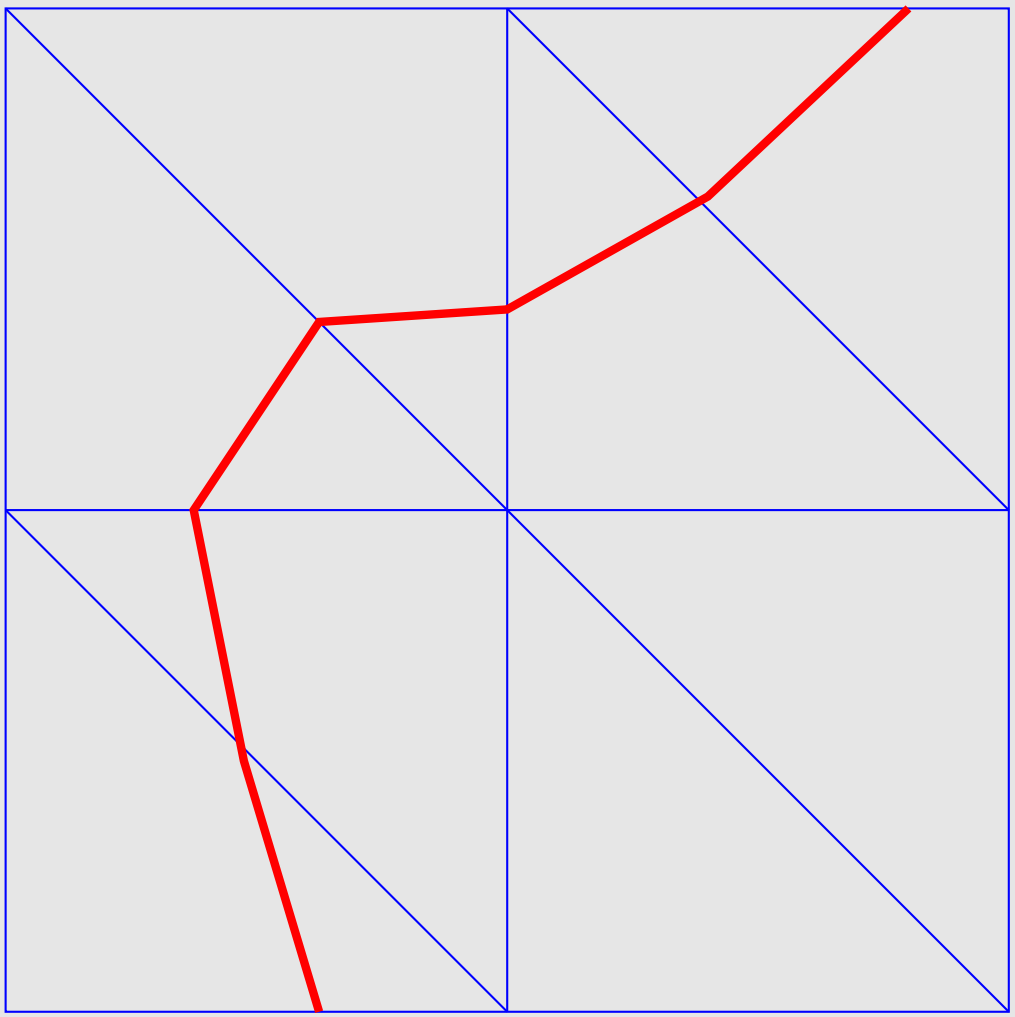
a grid consisting of squares

Construction of CFE Grids



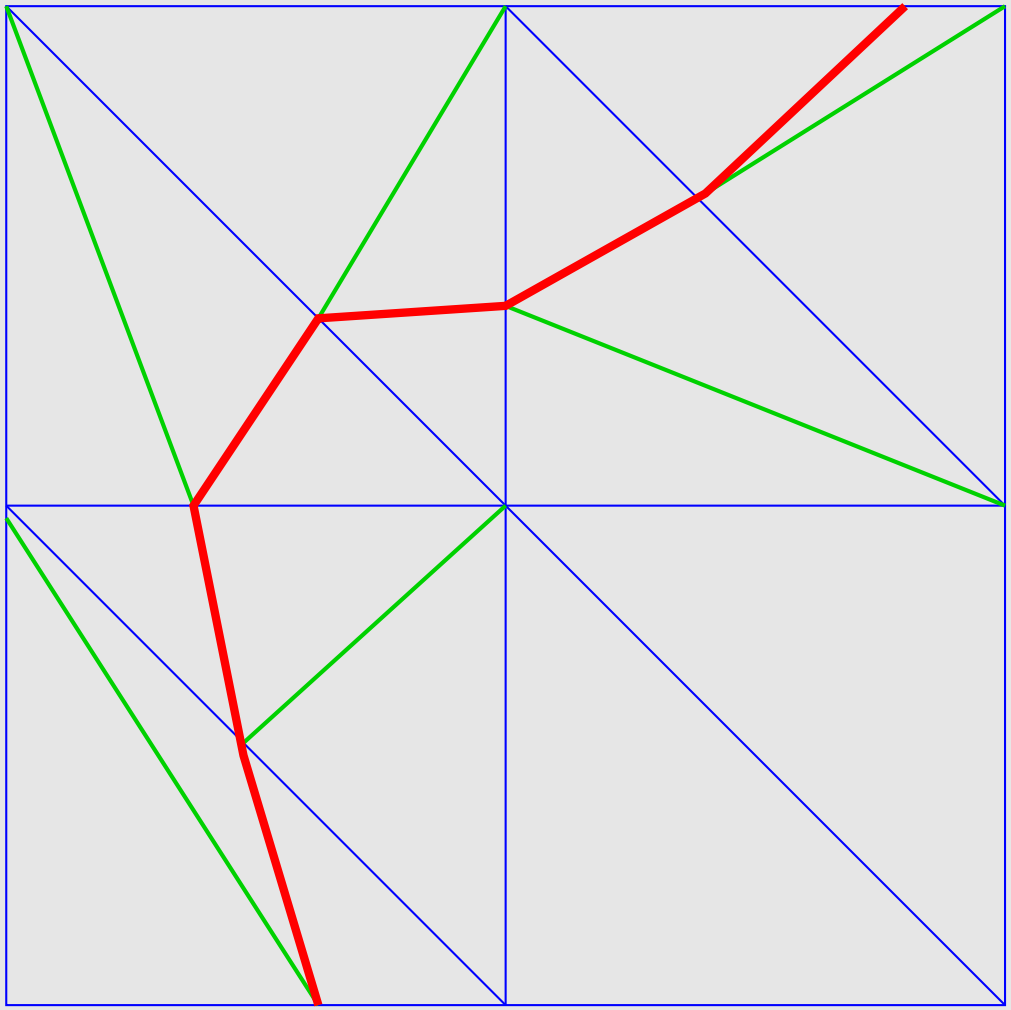
squares cut in triangles

Construction of CFE Grids

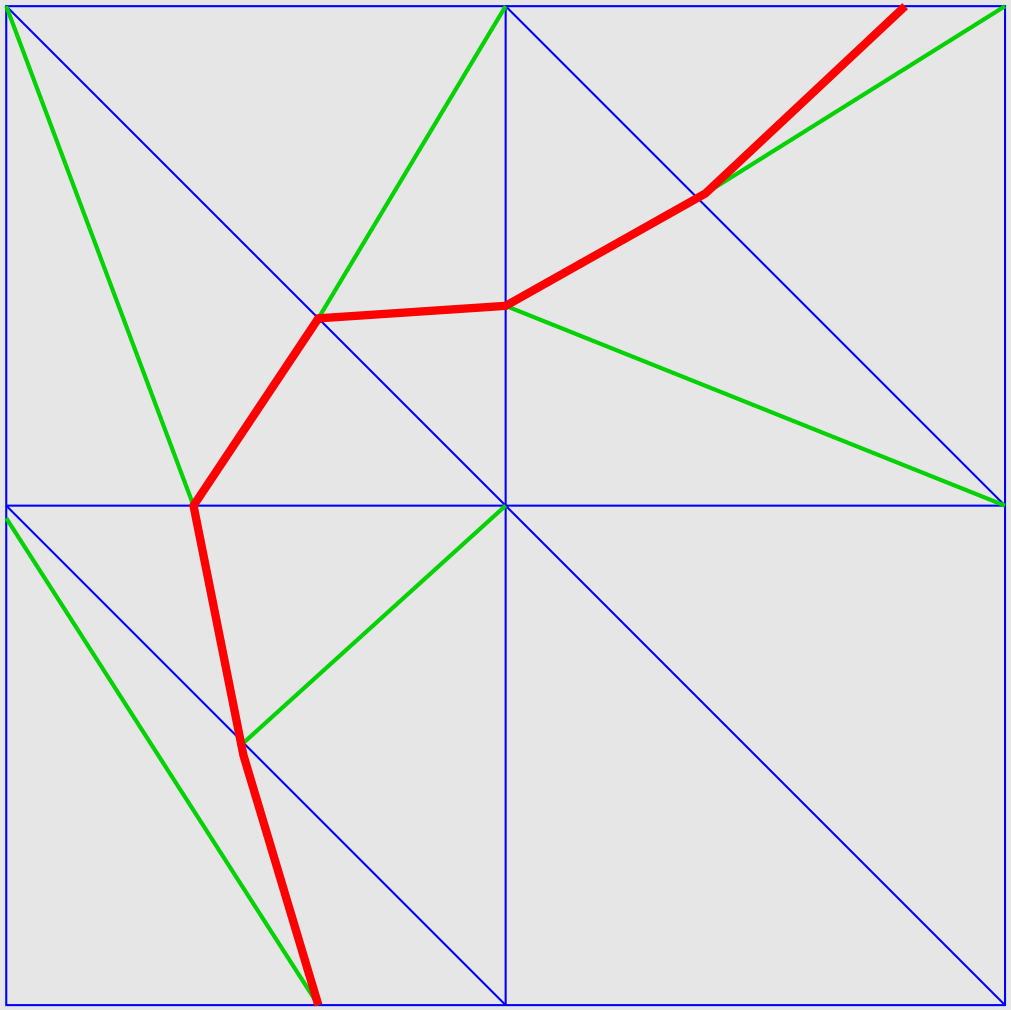


object boundary recovered from image

Construction of CFE Grids

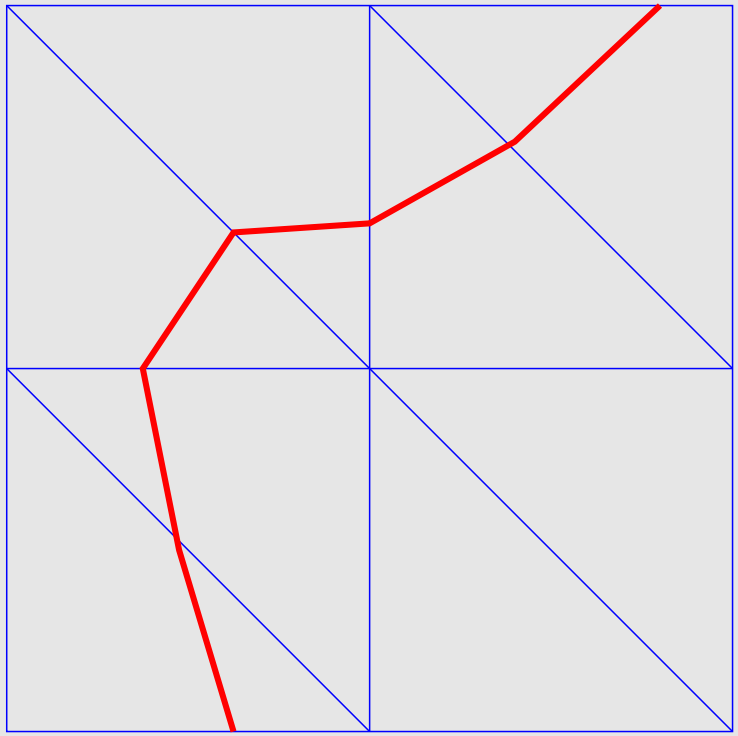


Construction of CFE Grids

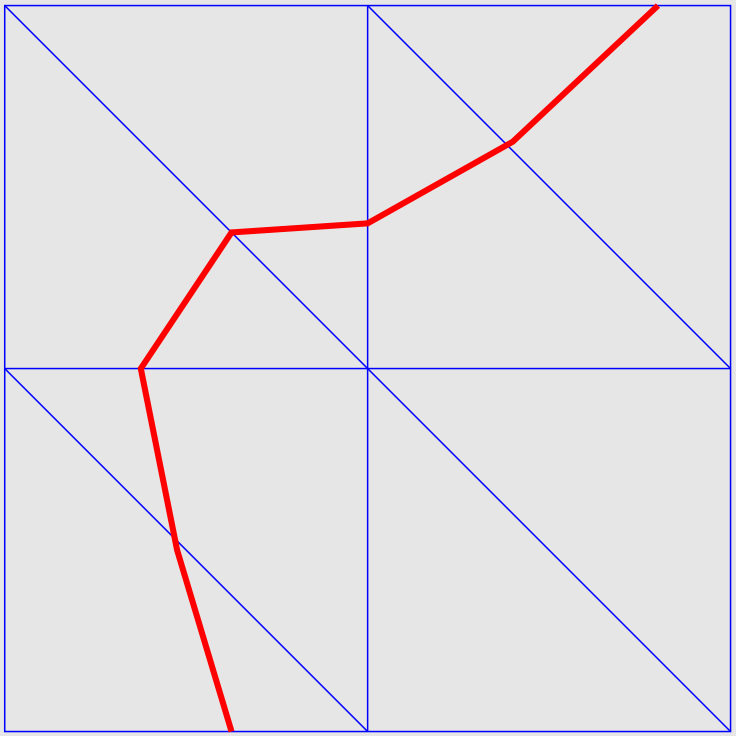


the "virtual grid" you don't want to use

CFE Basis Functions

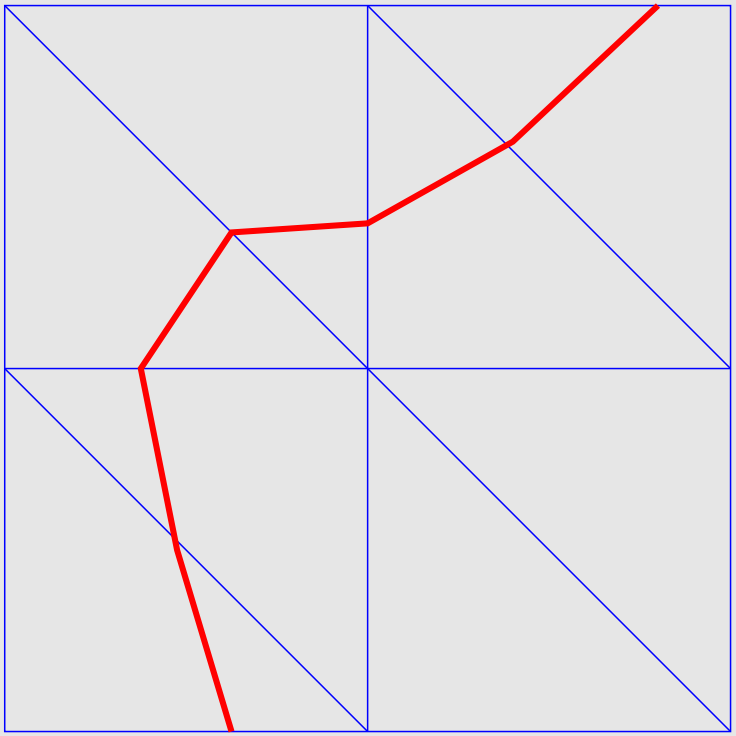


CFE Basis Functions



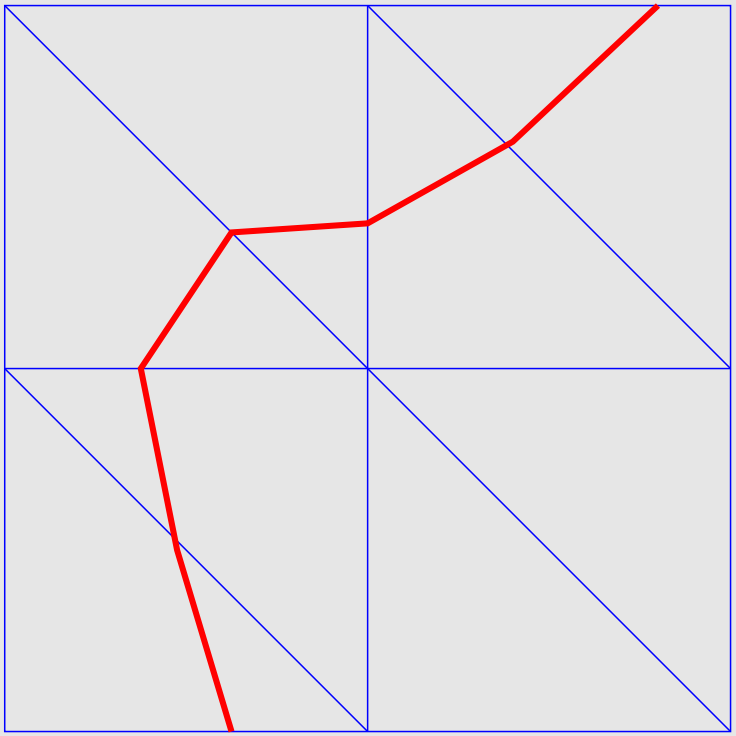
- regular grid points only as degrees of freedom: simple neighborhood relations

CFE Basis Functions



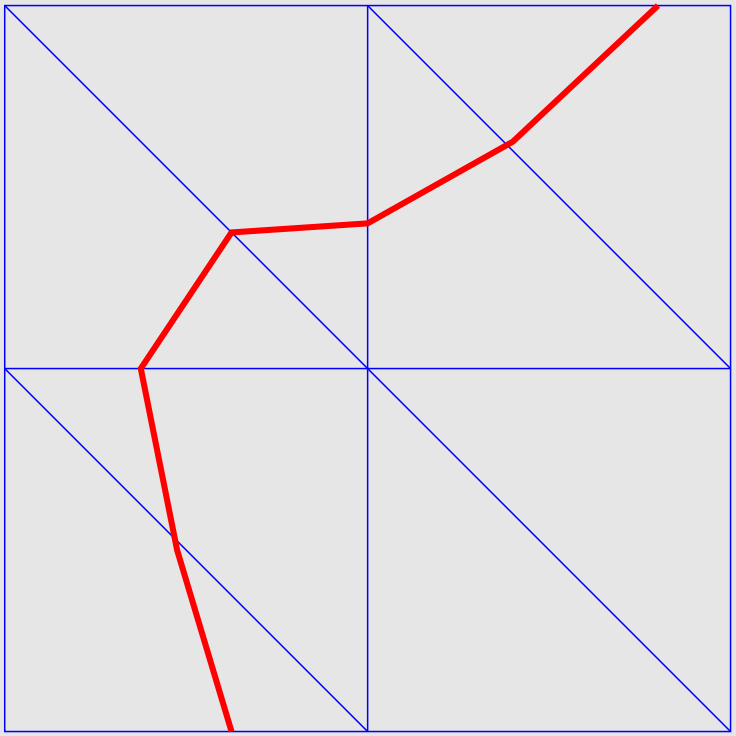
- regular grid points only as degrees of freedom: simple neighborhood relations
- cut off basis functions on regular grid at interface

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

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

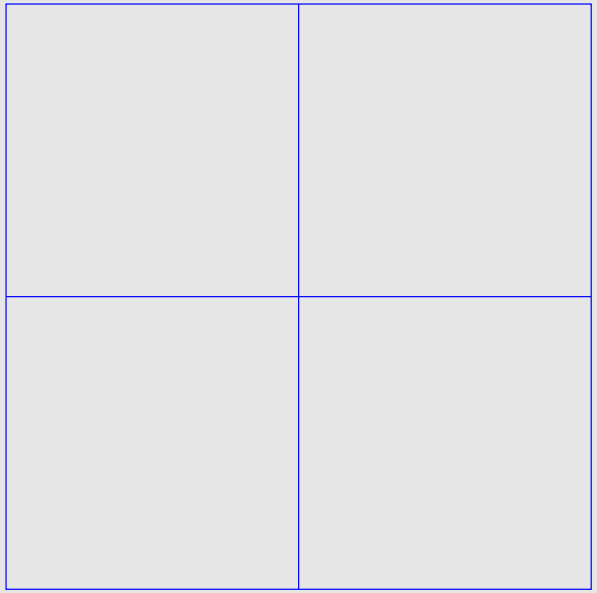
- meshing is easy, mesh is given by image data
- structured grid
- hence structured sparse system of equations
- computing entries more difficult (b/c basis functions contain complexity of structure)
- need more degrees of freedom than for classical unstructured grids
- multigrid methods can be used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

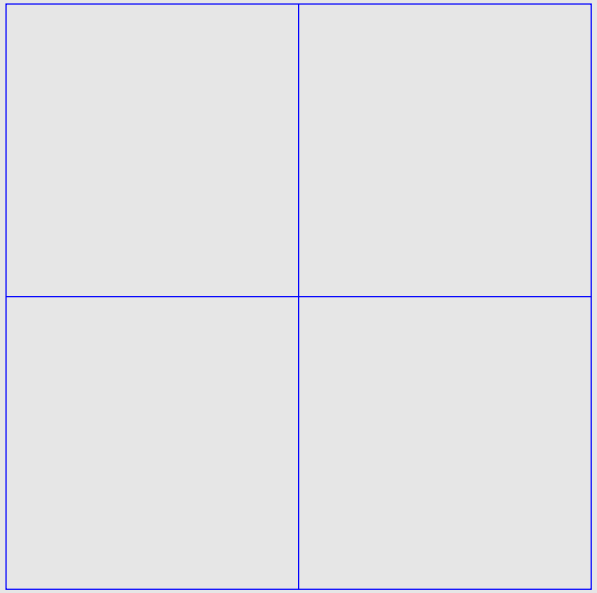
Extension to 3D

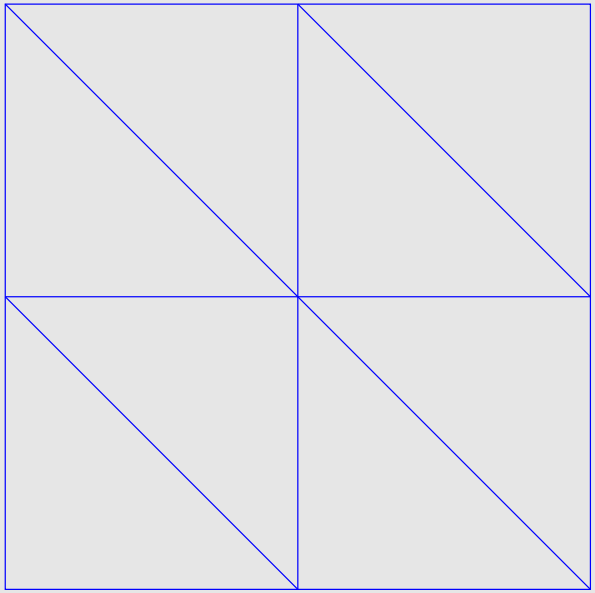
69	247	248	248	148
187	255	255	255	230
243	255	255	255	250
209	255	255	255	220
79	253	253	253	146



Extension to 3D

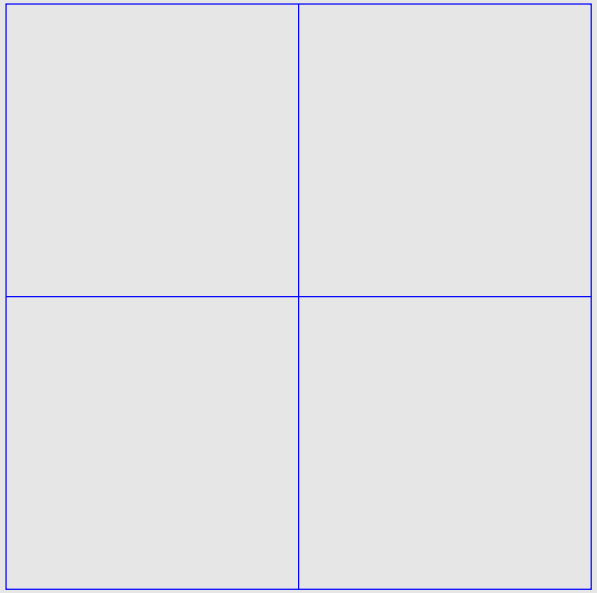
69	247	248	248	148
187	255	255	255	230
243	255	255	255	250
209	255	255	255	220
79	253	253	253	146

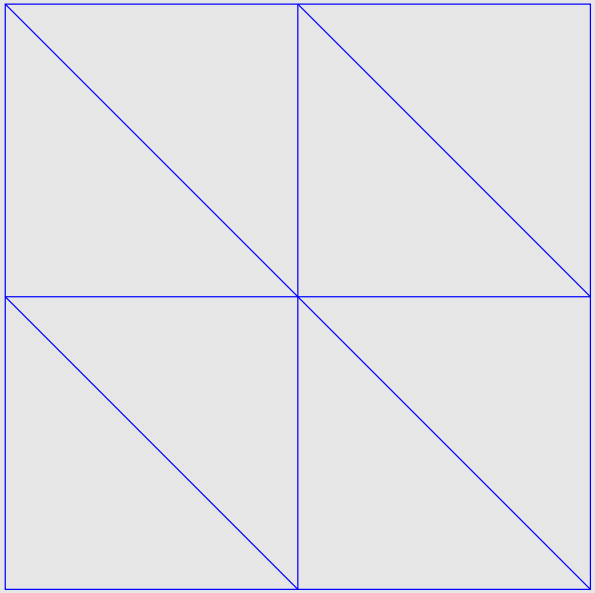


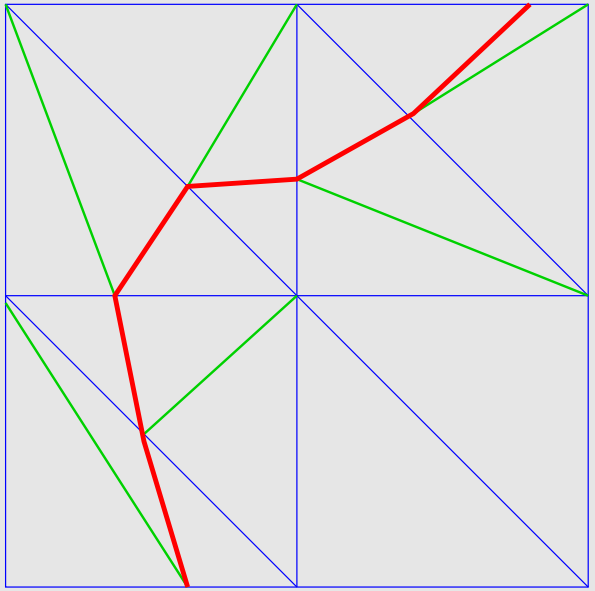


Extension to 3D

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







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

Experimental Validation

Experimental Validation

. . . und bei den Knochen vom Schwein nicht das »i« vergessen!

Summary

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
- Tobias Preusser, Markus Rech, Martin Rumpf, Stefan Sauter, Ole Schwen: Three-Dimensional Composite Finite Elements for Scalar Problems with Jumping Coefficients, in preparation