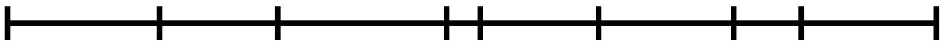
Block 4

wie machen das bloß die Profis? Ein (kleiner) Einblick in Deal II

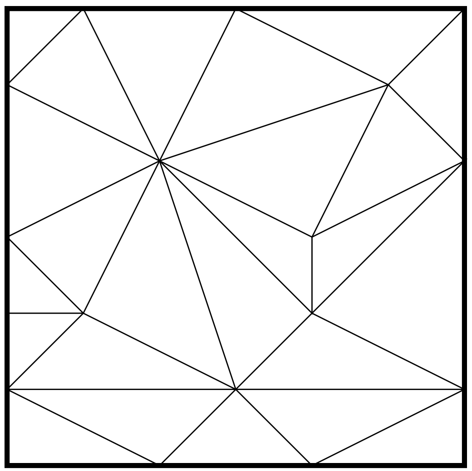
Sind Dreiecke immer dreieckig?

Fakt: In der FE-Community wird jede (geeignete) Zerlegung des Gebietes "Triangulierung" genannt

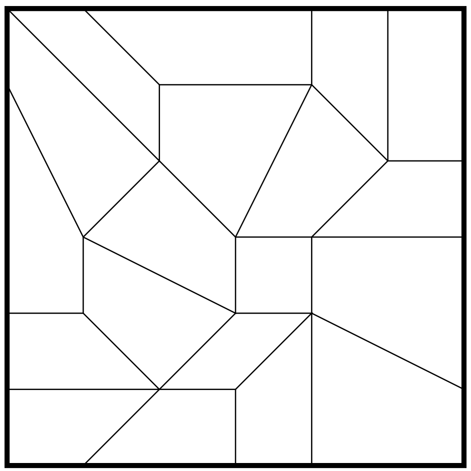
Triangulierung in 1D:



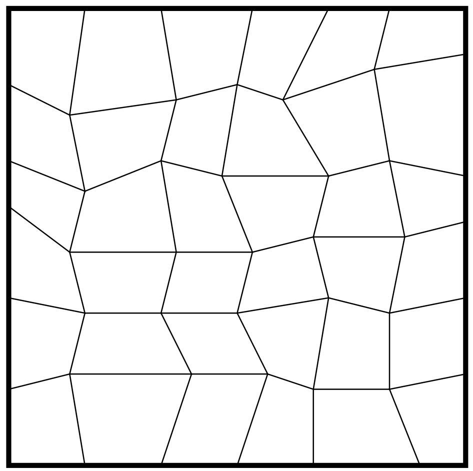
in 2D:



unstrukturiert

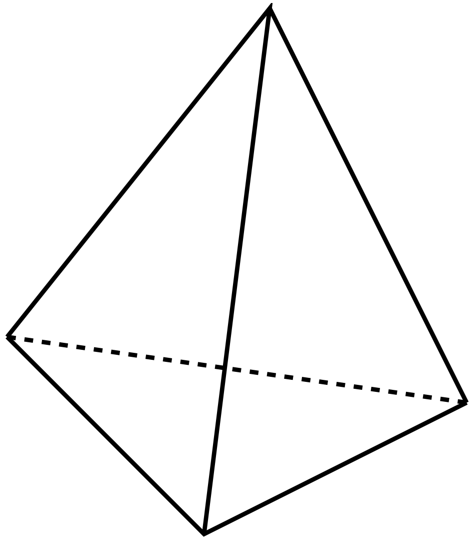


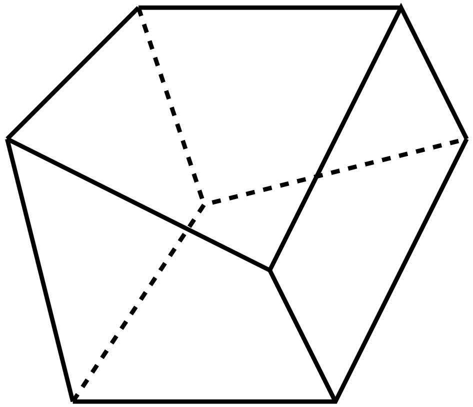
unstrukturiert



strukturiert

3D “Dreiecke” (Zellen):



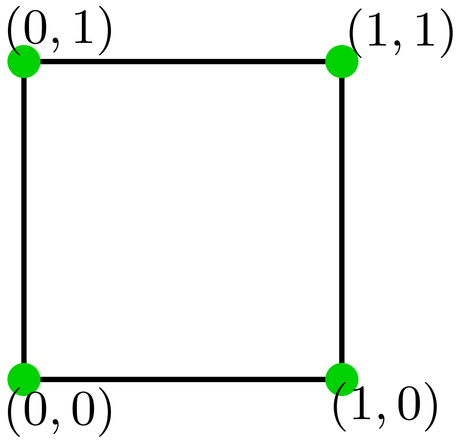


Fazit: sich nicht von Namen verwirren lassen!

heute: 4-eckige Triangulierungen

Bilineare Elemente

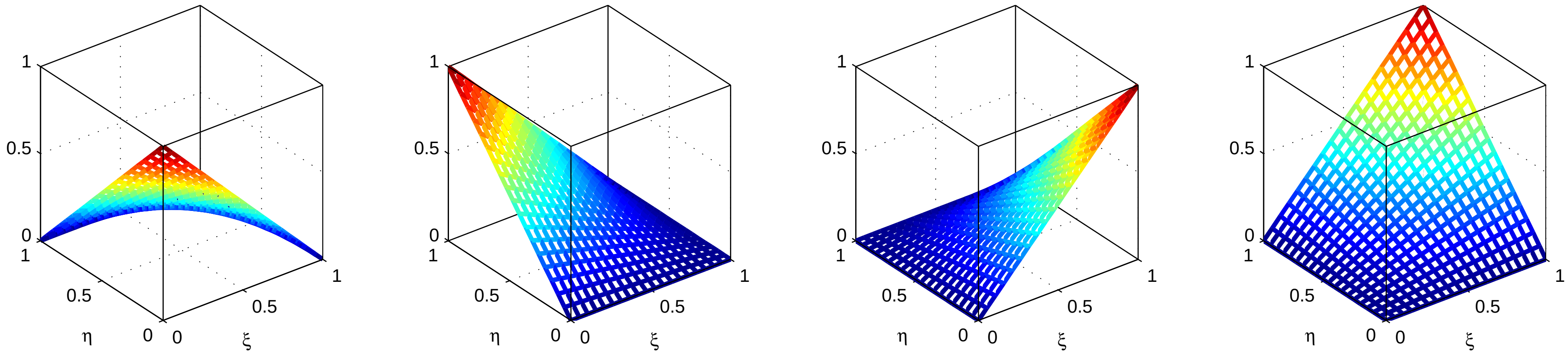
Referenzelement $\hat{\tau}$:



Bilineare Funktionen:

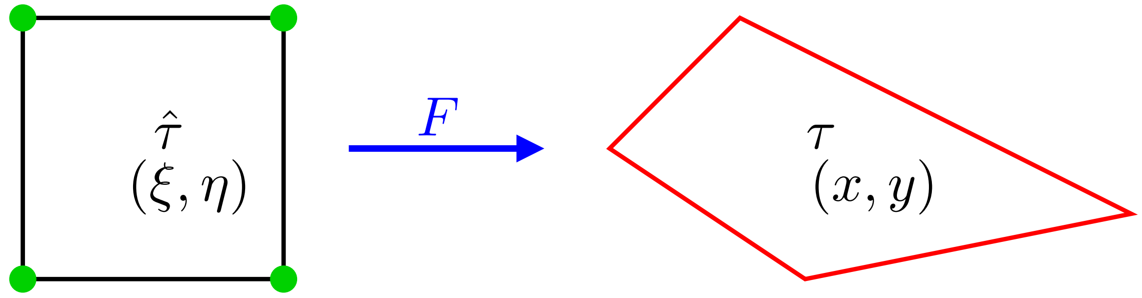
$$Q_1 = \mathrm{span}\,(1, \xi, \eta, \xi\eta)$$

Basisfunktionen auf dem Referenzelement:



$\hat{p}_1 = 1 - \xi - \eta + \xi\eta$ $\quad$ $\hat{p}_2 = \eta - \xi\eta$ $\quad$ $\hat{p}_3 = \xi - \xi\eta$ $\quad$ $\hat{p}_4 = \xi\eta$

Allgemeine Vierecke: erhalte die Basisfunktionen durch Transformation



wobei

$$F(\xi, \eta) = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_1 + a_2\xi + a_3\eta + a_4\xi\eta \\ b_1 + b_2\xi + b_3\eta + b_4\xi\eta \end{pmatrix} \quad , \quad a_i, b_i \in \mathbb{R}$$

Basiselemente auf dem Ursprungselement τ:

$$p_i(x, y) = \hat{p}_i(F^{-1}(x, y))$$

Sind die Äpfel besser oder die Birnen?

…Geschmacksache…

Genauso hier: ob man nun Dreiecke, Vierecke, strukturierte oder unstrukturierte Gitter benutzt hängt von vielen Faktoren ab. Zum Beispiel:

- Was für ein Gittergenerator ist vorhanden?
- Wie einfach ist die Adaptivität (lokale Gitterverfeinerung) zu erreichen? (so genannte “hängende Knoten”)
- Kompatibilität mit der restlichen Software
- usw

Deal II

deal.II is a C++ program library targeted at the computational solution of partial differential equations using adaptive finite elements. It uses state-of-the-art programming techniques to offer you a modern interface to the complex data structures and algorithms required.

The main aim of deal.II is to enable rapid development of modern finite element codes, using among other aspects adaptive meshes and a wide array of tools classes often used in finite element program. Writing such programs is a non-trivial task, and successful programs tend to become very large and complex. We believe that this is best done using a program library that frees the application programmer from aspects like grid handling and refinement, handling of degrees of freedom, input of meshes and output of results in graphics formats, and the like. Also, support for several space dimensions at once is included in a way such that programs can be written independent of the space dimension without unreasonable penalties on run-time and memory consumption.

[www.dealii.org]

Fragen für heute

Das Wichtigste für heute ist die Hauptstrukturen eines komplizierten FEM-Lösers nachzuvollziehen. Als Basis benutzen wir dafür Tutorial 3 aus der Deal-II Kollektion. Schauen Sie sich dieses an (Tutorials 1 und 2 so weit wie zum Verständnis notwendig)

1. Was ist das überhaupt?
   (a) Welche sind die wichtigsten Klassen/Funktionen?
   (b) Welche Funktionen führen welche Schritte in der allgemeinen FEM-Beschreibung (Blöcke 1 und 2) durch?
   (c) Vergleichen Sie die PDE-Toolbox und Deal. Welche Funktionen gehören zum Pre- und Postprozess?
2. Wie benutze ich das Ding?
   (a) Geben Sie das Ergebnis als eps-Datei aus
   (b) Lösen Sie das Problem mit $f = -1$
   (c) Lösen Sie das Problem auf 3mal, 7mal verfeinerten Gittern

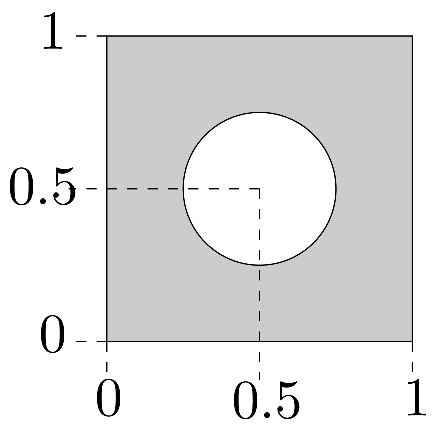
Für die weiteren Schritte nehmen Sie die Datei `laplace.tar`. (Author: M. Kaip.) Packen Sie diese im Verzeichnis `deal.II` aus (dabei entsteht ein Verzeichnis `LaplaceVariations`). Unter anderem enthält das Paket ein Werkzeug zum Visualisieren der Ergebnisse in Matlab.

Auf der Basis dieses Programms lösen Sie die Gleichung

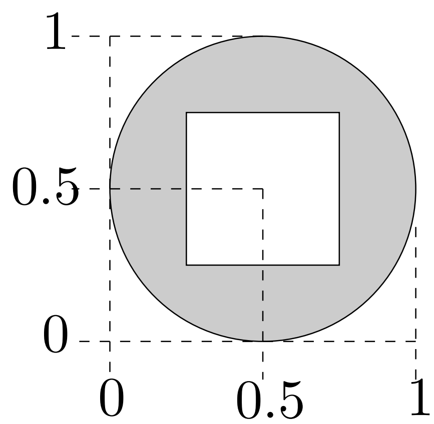
$$\Delta u = -9 \sin 3x + 6y \quad \text{in } \Omega$$

mit Dirichlet Randbedingungen $\qquad u = \sin(3x) + y^3 \quad \text{am } \partial\Omega$

mit dem Gebiet Ω:



(A)



(B)

Die exakte Lösung dieses Problems ist gegeben durch $u_{ex} = \sin(3x) + y^3$. Visualisieren Sie den numerischen Fehler (Matlab-Tool) und berechnen Sie $e := \|u - u_{ex}\|_\infty$. Wie verhält sich e bei der Gitterverfeinerung?

Vortragsthemen

Elemente höherer Ordnung

V1 Approximationseigenschaften (Cea's Lemma)

V2 Demonstration in 1D

- Konvergenzordnung
- Fehlerverhältnis, Vergleich mit P_1-Elemente
- Rechenkosten (Aufbau der Steifigkeitsmatrix)

V3 Demonstration in 2D (Deal, PDE-Toolbox erlaubt leider keine P_2-Elemente)

- Konstruktion der Basiselemente (bi-quadratisch, bi-kubisch)
- Konvergenzordnung: Einfluß der Basiselemente, Einfluß der numerischen Integration
- Laufzeitanalyse, Rechenkosten

Literatur:

— [4], Seiten 57-60, 69-78
— [2], 3.3, 3.4, 3.5.3
— Deal-II, Tutorial 6

Weitere Anwendungen

V1 Andere Gleichungen, schwache Formulierungen, Randbedingungen

- Stationäre elliptische Probleme: Helmholtz, Elastizität, Strömungsmechanik
- Stationäre Probleme als Baustein für zeitabhängige und nichtlineare Prozesse

V2 FEM für Gleichungssysteme

- Basisfunktionen, Aufbau der Steifigkeitsmatrix
- Elastizitätsgleichungen als Beispiel (Deal/PDE-Toolbox)

V3 Beispiele aus der Strömungsmechanik

- Instabile Elemente, Brezzi-Bedingung

V* Inhomogene Dirichlet Randbedingunen: Implementierung (Block 2)

Literatur:

— [5], Seiten 148⇒, 255–261, 262–272, 276–294
— [1], Seite 175, 3.2.2, 3.3.2, 3.4.1
— [6]
— Deal-II, Tutorial 8

Adaptivität

V1 Motivation für lokale Verfeinerungen.

- "Anwendungsgezwungene" Verfeinerungen
- Einfache Beispiele. Großer Fehler $\Rightarrow$ verfeinern/global verfeinern. Warum ist Schnittfehler (truncation error) nicht genug?

V2 A posteriori Fehlerschätzungen, Dualproblem

V3 Eigene Beispiele: Laufzeit und Genauigkeitsanalyse mit/ohne adaptiver Gitterverfeinerung

Literatur:

— [2], Seiten 129–146
— [3]
— [4], 8.7

Literatur

[1] R. Rannacher. *Numerische Mathematik 3 (Numerik von Problemen der Kontinuumsmechanik), Vorlesungsskriptum WS 2006/2007*, Heidelberg

[2] R. Rannacher. *Numerische Mathematik 3 (Numerik Partieller Differentialgleichungen), Vorlesungsskriptum SS 2006*, Heidelberg

[3] W. Bangerth and R. Rannacher. *Adaptive Finite Element Methods for Differential Equations*, Birkhäuser, 2003.

[4] AG Technomathematik. *Numerik partieller Differentialgleichungen I: Elliptische und parabolische Gleichungen, Vorlesungsskript*, Kaiserslautern.

[5] D. Braess. *Finite Elemente: Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie*, Springer, 2003

[6] W. Bangerth, *Assembling matrices in deal.II*, 2002.