

# Praktikum Wissenschaftliches Rechnen



# Allgemeines

Vorlesungsseite:

[wwwmath.uni-muenster.de/num/Vorlesungen/WissenschaftlichesRechnen\\_WS1213](http://wwwmath.uni-muenster.de/num/Vorlesungen/WissenschaftlichesRechnen_WS1213)

vorläufiger Termin:

Donnerstags 16:00 - 18:00 Uhr

Scheinkriterium:

1. regelmäßige Abgaben der Übungszettel
2. Anwesenheit

Begleitend zur Vorlesung: Numerik Partieller  
Differentialgleichungen I



## Über mich

- ▶ WS 2008/2009: erstes DUNE-Praktikum
- ▶ 07/2009 bis 03/2012: Wissenschaftliche Hilfskraft
- ▶ 03/2012: Diplomarbeit "Vereinheitlichter Rahmen zur Implementierung hybridisierter Diskretisierungsverfahren"
- ▶ seit 04/2012: Wissenschaftlicher Mitarbeiter: Simulation von Gefäßverkalkungen

## Beispiel: RT-Hybrid, order 1

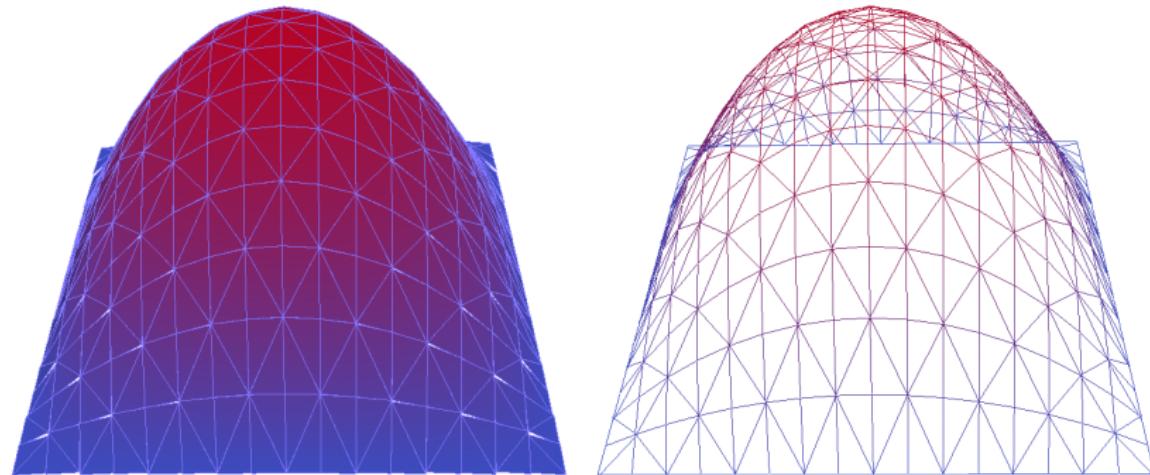


Figure: Druck/Lagrange Multiplizierer

## Beispiel: RT-Hybrid, Ordnung 5, 8 Elemente

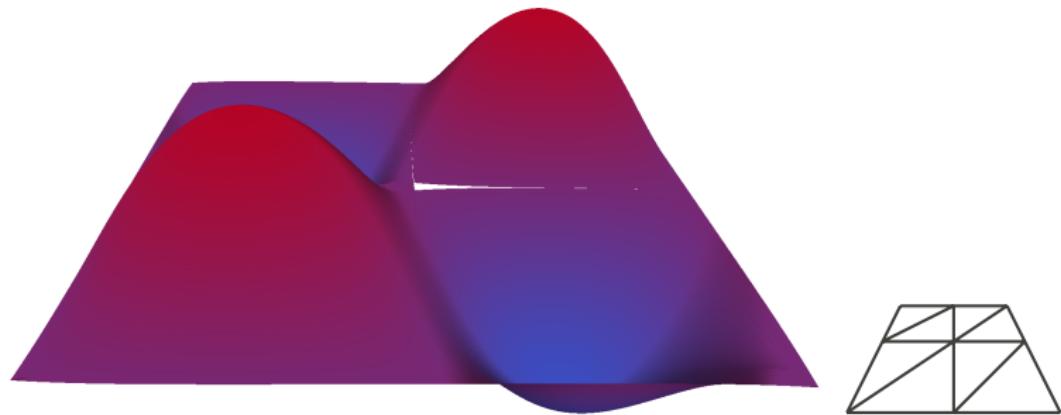


Figure: Druck/Gitter



## Ziele des Praktikums

1. sicherer Umgang beim Programmieren mit Linux
2. Erfahrung mit Programmierkonzepten
3. Erfahrung bei der Implementierung von numerischen Verfahren
4. Grundlagenkenntnisse der DUNE-Konzepte
5. erste Erfahrung mit DUNE

# Warum DUNE?



Distributed and Unified Numerics Environment

Homepage: [www.dune-project.org](http://www.dune-project.org)

DUNE = Distributed and Unified Numerics Environment

1. Modularität
2. Flexibilität
3. Trennung von Datenstrukturen und Algorithmen
4. Effizienz
5. Einbindung externer Pakete

# Inhalt des Praktikums

1. Entwicklung eines Gitters in 1D mit äquidistanter Verfeinerung
2. Diskrete Funktionen und Finite Elemente
3. Lineare Algebra: Sparse- und Dense-Matrizen, CG-Löser
4. Poisson-Problem mit Finite Elemente
5. Entwicklung eines adaptiven Gitters in 1D
6. Adaptives Poisson-Problem
7. Template-Meta-Programming
8. DUNE: dune-common, dune-geometry und dune-grid
9. DUNE: Poisson-Problem in dune-fem

## Coding-Styles

1. Einrückung(!), einheitlich 2 oder 4 Leerzeichen,
2. Kommentare,
3. Klassennamen groß, Camel Case, d.h. `MeineKlasse`,
4. Variablen klein, Camel Case, d.h. `meineVariable`,
5. Präprozessor-Konstanten komplett groß, d.h. `KONST`
6. Membervariablen mit Unterstrich, d.h. `var_`,
7. Typdefinitionen mit Type, d.h. `MeinType`,
8. maximal ein Befehl pro Zeile, keine überlangen Zeilen,
9. kein `using namespace`,
10. (einheitliche Whitespaces)



## Weitere Empfehlungen

### 1. const-Correctness