

Adaptive Brandsimulationen

Verifikation des Strömungslösers

17. Mai 2017 | Marc Fehling | Computersimulation für Fußgängerverkehr und Brandschutz

Einführung

- Marc Fehling
- geboren 1990
- M. Sc. Physik, RUB, 2015
- seit November 2015:
Doktorand am
Jülich Supercomputing Centre (JSC),
Abteilung Sicherheit & Verkehr



Inhaltsverzeichnis

Motivation

Modellierung von Bränden

Finite Elemente Methode (FEM)

Adaptive Gitterverfeinerung (AMR)

High-performance computing auf JURECA

Zusammenfassung & Ausblick

Motivation: Brandschutz in komplexer Architektur

- Moderne Architektur und Großprojekte (wie BER) oft nicht vollständig durch Bauverordnung erfasst.
- Individuelle Untersuchungen sind notwendig:
 - Modellexperimente, CFD Untersuchungen, ...



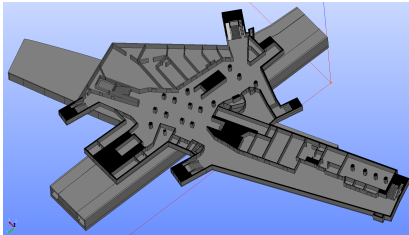
Abbildung: Innenbereich der BMW Welt, München.

Motivation: Projekt Orpheus



Optimierung der **R**auchableitung und **P**ersonenführung in U-Bahn**h**öfen: **E**xperimente **u**nd **S**imulationen

- Rauch- und Feuerausbreitung in komplexen U-Bahnstationen.
 - Multidimensionaler Parameterraum für Brandszenarien, inklusive Untergrundklima.



Modellierung von Bränden

Physikalische Vorgänge:

- **Fluiddynamik.**
 - Navier–Stokes Gleichungen.
 - Turbulenzmodell.
- **Strahlung.**
 - Discrete Transfer Radiation.
 - Discrete Ordinates.
 - Monte–Carlo.
- **Verbrennung**
 - Mixture fraction.
 - Finite rate kinetics.
- **Pyrolyse.**

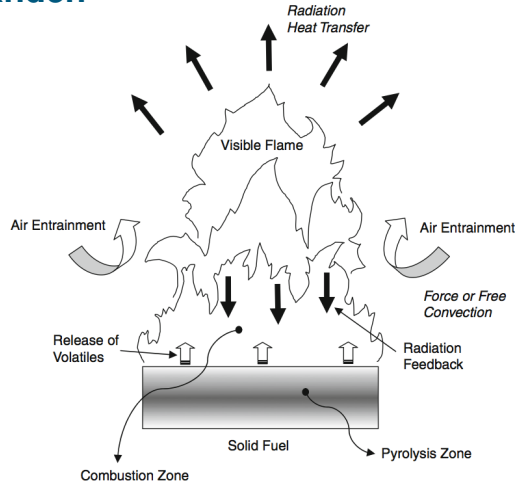


Abbildung: Verbrennung von festem Brennstoff in Luft. [1]

Modellierung: Gleichungen zur Rauchausbreitung

- Rauchausbreitung mit inkompressiblen Navier–Stokes (INS) Gleichungen:

$$\nabla \cdot \mathbf{u} = 0$$

$$\rho_0 [\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u}] + \nabla p - \nabla (2 \mu \epsilon_{ij}(\mathbf{u})) = \mathbf{f}(T)$$

$$\rho_0 [\partial_t T + (\mathbf{u} \cdot \nabla) T] - 2 \frac{\mu}{c_p} \epsilon_{ij}(\mathbf{u}) : \nabla \mathbf{u} - \nabla \cdot \left(\frac{\mu}{Pr} \nabla T \right) = \gamma$$

mit Verzerrungstensor $\epsilon_{ij}(\mathbf{u}) = \frac{1}{2} [\nabla \mathbf{u} + (\nabla \mathbf{u})^T]$.

- Turbulenzmodell: Smagorinsky–Lilly LES [2]:

$$\mu = \mu_{mol} + \mu_{turb} \quad \text{mit} \quad \mu_{turb} = \rho_0 (C_s h)^2 ||\epsilon_{ij}(\mathbf{u})||_2 \cdot$$

Numerische Methoden

Methoden zur Raumdiskretisierung für numerische Strömungsdynamik, sowie Software-Pakete, die diese verwenden:

- Finite Differenzen Methode (FDM).
 - [Open source](#): NIST Fire Dynamics Simulator (FDS), ...
- Finite Volumen Methode (FVM).
 - [Open source](#): FireFOAM (OpenFOAM), ...
- Finite Elemente Methode (FEM).
- Lattice–Boltzmann Methode (LBM).

FEM: Eigenschaften

Warum FEM?

FEM: Eigenschaften

Warum FEM?

- Unstrukturierte Gitter.
 - Genauere Repräsentation des Rechengebiets ohne Treppeneffekte.

FEM: Eigenschaften

Warum FEM?

- Unstrukturierte Gitter.
 - Genauere Repräsentation des Rechengebiets ohne Treppeneffekte.
- hp-Adaptivität.
 - Dynamische Auflösung des Gitters.
 - Adaptive Polynomgrade der Basisfunktionen.
 - Verbessere Genauigkeit dort, wo etwas passiert!

FEM: Eigenschaften

Warum FEM?

- Unstrukturierte Gitter.
 - Genauere Repräsentation des Rechengebiets ohne Treppeneffekte.
- hp-Adaptivität.
 - Dynamische Auflösung des Gitters.
 - Adaptive Polynomgrade der Basisfunktionen.
 - Verbessere Genauigkeit dort, wo etwas passiert!
- Unstetige Galerkin (DG) Methoden.
 - Erlaube Unstetigkeiten über Zellgrenzen.
 - Stetige Galerkin (CG) Methoden instabil für advektionsartige Probleme, erfordern besondere Stabilisierung.

JuFire



- Untersuchung der Anwendbarkeit der finiten Elemente Methode (FEM) für Brandsimulationen.
- Verwendung der open-source Bibliothek deal.II [3].
 - 'Baukasten' zur Erstellung von FEM Codes.



Differential **E**quations **A**nalysis **L**ibrary

JuFire: Eigenschaften

Implementiert:

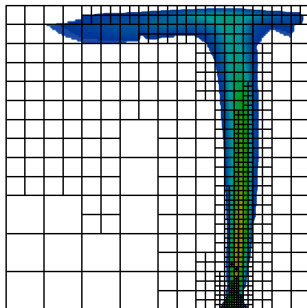
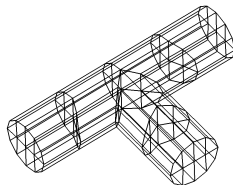
- Unstrukturierte Gitter.
- Stetige Galerkin Methoden.
- Adaptive Gitterverfeinerung.
- MPI Parallelisierung.
- Verwendung von CAD Modellen als Mannigfaltigkeiten.

Aktuelles Aufgabenfeld:

- Verifikation des Strömungslösers.

Ausblick:

- Unstetige Galerkin Methoden.
- p-Adaptivität.



Verifikation: Richardson Extrapolation

- Wie erhält man Konvergenzraten für zeit- **und** raum aufgelöste Probleme?
- Streiche überflüssigen Fehler mit Richardson Extrapolation. Wähle Verhältnis r für schrittweise Verfeinerungen.
- Beispiel für Ortskonvergenz:

$$\left. \begin{aligned} f_1 - f_0 &= c_t \Delta t^{p_t} + c_h \Delta h^{p_h} \\ f_2 - f_0 &= c_t \Delta t^{p_t} + c_h (r \Delta h)^{p_h} \\ f_3 - f_0 &= c_t \Delta t^{p_t} + c_h (r^2 \Delta h)^{p_h} \end{aligned} \right\} \ln \left(\frac{f_3 - f_2}{f_2 - f_1} \right) = p_h \ln(r)$$

Verifikation: Konvergenzraten

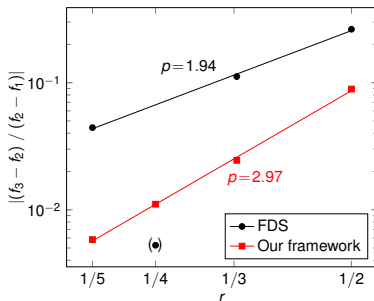


Abbildung: Ortskonvergenz.

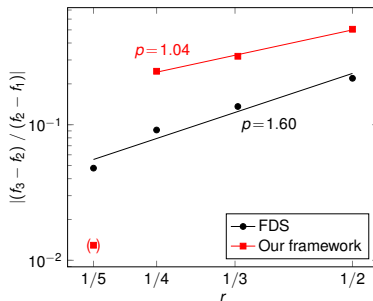


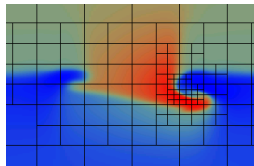
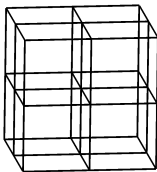
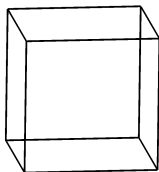
Abbildung: Zeitkonvergenz.

Adaptive Gitterverfeinerung (AMR)

- Lösung von Differentialgleichungen mit *FEM* erfordern:
 - Raumdiskretisierung in Zellen der Länge h .
 - Basisfunktionen mit Polynomgrad p .
- Aber: Je genauer die Lösung durch Wahl von h und p , desto länger die Rechenzeit.

Adaptive Gitterverfeinerung (AMR)

- Lösung von Differentialgleichungen mit *FEM* erfordern:
 - Raumdiskretisierung in Zellen der Länge h .
 - Basisfunktionen mit Polynomgrad p .
 - Aber: Je genauer die Lösung durch Wahl von h und p , desto länger die Rechenzeit.
- **Adaptive Verfeinerung** als „Kompromiss“.
- Lokale Anpassung der Parameter, wo nötig.
 - Dynamische Gitterauflösung zur Laufzeit.



J.Dreher, RUB

AMR: Algorithmus

Finde Entscheidungskriterium für
Verfeinerung/Vergröberung.

Unsere Wahl:

- Berechne $||\nabla u||$ auf jeder Zelle.
- Normiere Werte in Bezug auf alle Zellen.

AMR: Algorithmus

Finde Entscheidungskriterium für
Verfeinerung/Vergröberung.

Unsere Wahl:

- Berechne $||\nabla u||$ auf jeder Zelle.
- Normiere Werte in Bezug auf alle Zellen.
- Markiere obere/untere 30% zur Verfeinerung/
Vergröberung zu jedem Zeitschritt.
- Definiere oberste/unterste Verfeinerungsebenen.

AMR: Algorithmus

Finde Entscheidungskriterium für
Verfeinerung/Vergrößerung.

Unsere Wahl:

- Berechne $||\nabla u||$ auf jeder Zelle.
- Normiere Werte in Bezug auf alle Zellen.
- Markiere obere/untere 30% zur Verfeinerung/
Vergrößerung zu jedem Zeitschritt.
- Definiere oberste/unterste Verfeinerungsebenen.
- Zellen dürfen sich nicht mehr als eine Verfeinerungsebene unterscheiden.

AMR: Beispiel

- Demonstration anhand eines bewegenden Wirbels aus einer formerhaltenden Potentialströmung.

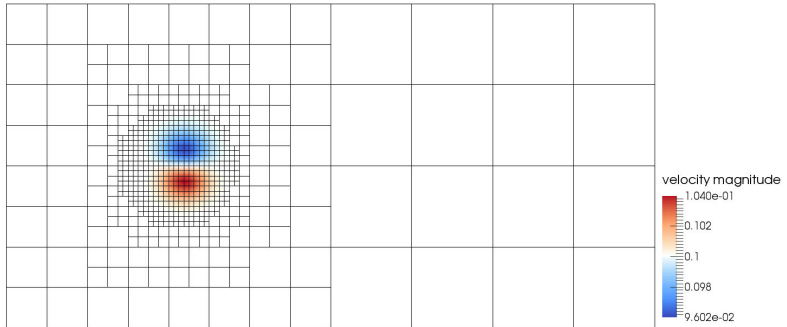


Abbildung: Video des Geschwindigkeitsbetrags eines bewegenden Wirbels, überlagert mit dem entsprechenden Gitter.

AMR: Vorteile

- Vergleich von Laufzeit und Genauigkeit zwischen gleichförmigen und adaptiv verfeinerten Gittern (Wirbelbeispiel).

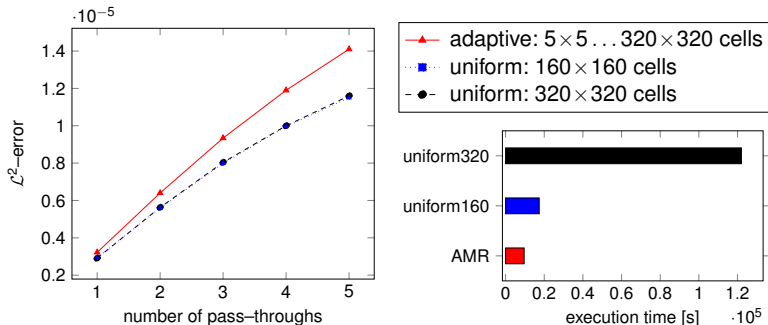


Abbildung: Globale $\mathcal{L}^2$ -Fehler zu jedem periodischen Durchlauf des Wirbels.

Abbildung: Laufzeit der seriellen Wirbelsimulation auf einem gewöhnlichen Desktop-PC.

AMR: Errors

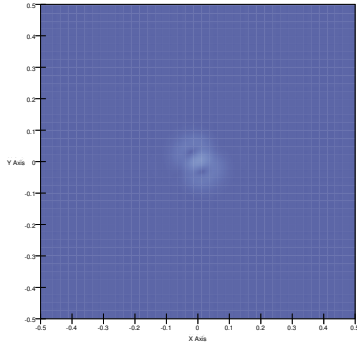


Abbildung: Statisches Gitter.

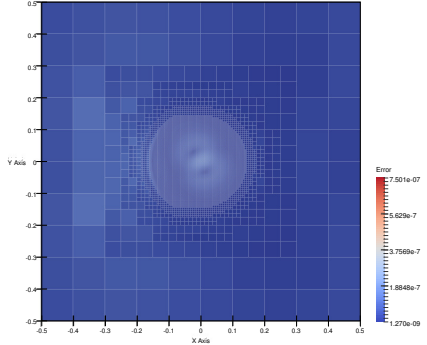


Abbildung: Adaptives Gitter.

1. Durchlauf.

AMR: Errors

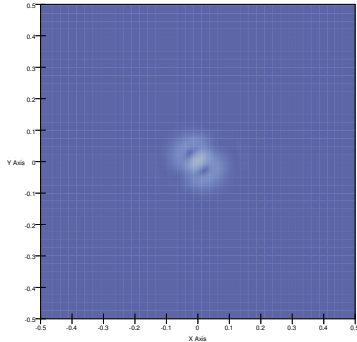


Abbildung: Statisches Gitter.

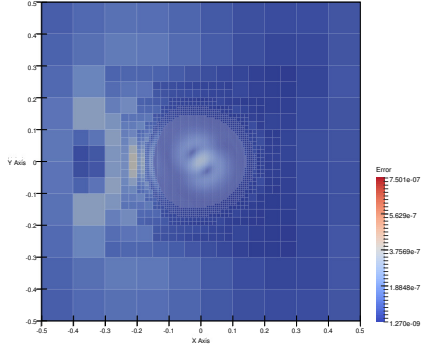


Abbildung: Adaptives Gitter.

2. Durchlauf.

AMR: Errors

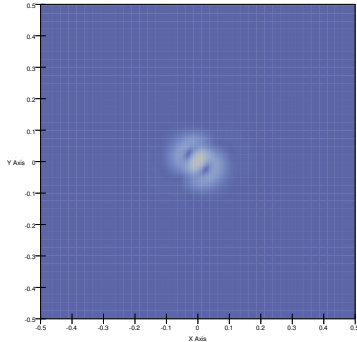


Abbildung: Statisches Gitter.

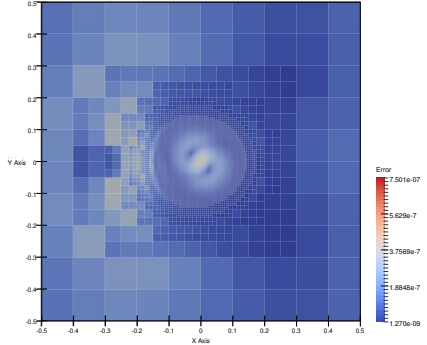


Abbildung: Adaptives Gitter.

3. Durchlauf.

AMR: Errors

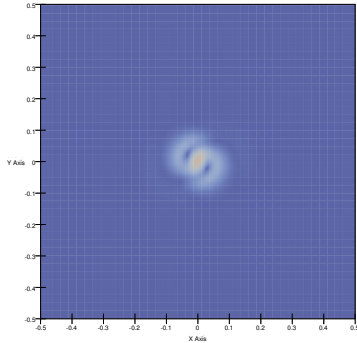


Abbildung: Statisches Gitter.

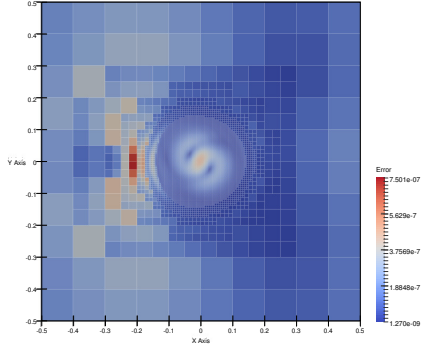


Abbildung: Adaptives Gitter.

4. Durchlauf.

AMR: Errors

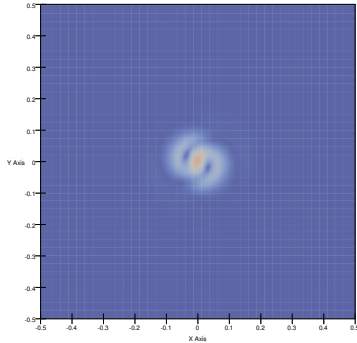


Abbildung: Statisches Gitter.

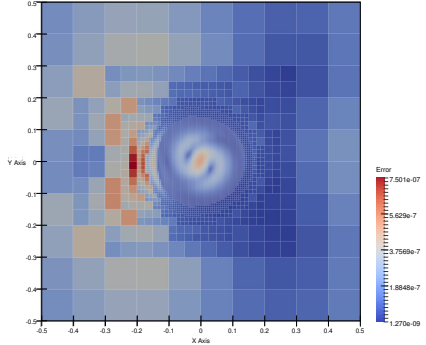


Abbildung: Adaptives Gitter.

5. Durchlauf.

HPC auf JURECA: Parallelisierung

- MPI Parallelisierung mit Trilinos und p4est durch deal.II Backends.

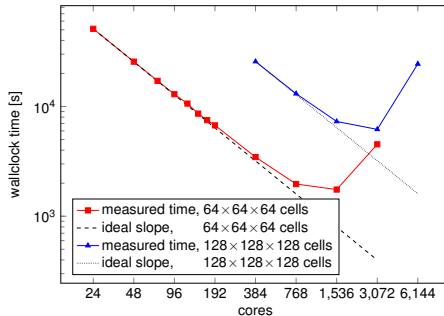


Abbildung: Starke Skalierung auf JURECA für festgelegte Probleme in 3D mit bis zu 2.097.152 Zellen.

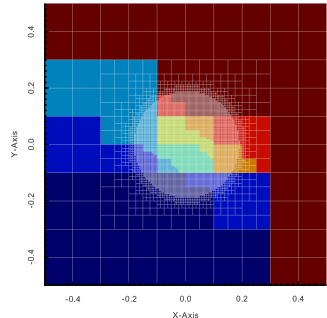


Abbildung: Beispielhafte Gebietszerlegung mit p4est für einen AMR Fall.

Zusammenfassung & Ausblick

Zusammenfassung:

- Adaptive Gitterverfeinerung läuft zufriedenstellend.
- Strömungslöser bestätigt erste Konvergenztest.

Ausblick:

- Implementation weiterer Modelle.
 - Auftrieb, Strahlung, Verbrennung, Pyrolyse, ...
- Erweiterung der Numerik.
 - DG Verfahren, p-Adaptivität, ...
- Vergleich mit anderer Simulationssoftware für Brände.
- Validierung anhand Experimenten.

Vielen Dank für Ihre Aufmerksamkeit!

Literaturverzeichnis I



Guan Heng Yeoh and Kwok Kit Yuen.

Computational Fluid Dynamics in Fire Engineering – Theory, Modelling and Practice.

Butterworth–Heinemann, 1st edition, 2009.



Joseph Smagorinsky.

General circulation experiments with the primitive equations.

Monthly Weather Review, 91(3):99–164, March 1963.

Literaturverzeichnis II



W. Bangerth, D. Davydov, T. Heister, L. Heltai,
G. Kanschat, M. Kronbichler, M. Maier, B. Turcksin, and
D. Wells.

The deal.II library, version 8.4.

Journal of Numerical Mathematics, 24, 2016.



Cedric Taylor and P. Hood.

A numerical solution of the Navier–Stokes equations
using the finite element technique.

Computers & Fluids, 1:73–100, 1973.

Literaturverzeichnis III



Bernardino Roig.

One-step Taylor Galerkin methods for convection-diffusion problems.

Journal of Computational and Applied Mathematics,
204:95–101, 2007.



N.D. Heavner and L.G. Rebholz.

Locally chosen grad-div stabilization parameters for finite element discretizations of incompressible flow problems.

SIAM Undergraduate Research Online (SIURO), 7.

Adaptive Brand- simulationen: Anhang Verifikation des Strömungslösers

17. Mai 2017 | Marc Fehling | Computersimulation für Fußgängerverkehr und Brandschutz

FEM: Formulierung

- Lösung Variationsgleichung aus der 'schwachen' Formulierung der Differentialgleichung mit Bilinearform $a(u, v)$:

$$\exists u \in V : \forall v \in V : a(u, v) = f(v)$$

- Wähle Teilraum V_h mit Basis w_i , aus dem die numerische Lösung $u_h = \sum u_i w_i \in V_h$ konstruiert wird:

$$a(u_h, w_j) = \sum a(w_i, w_j) u_i = f(w_j) \rightarrow AU = F$$

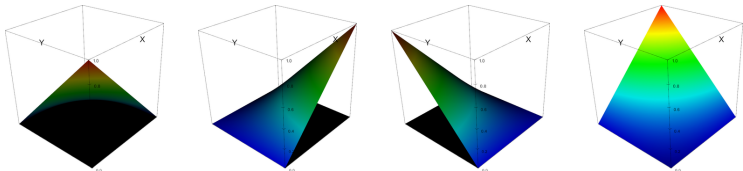


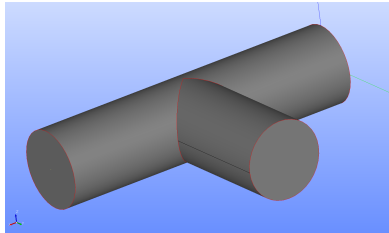
Abbildung: Q_1 Elemente in 2D (source: [deal.II](#))

JuFire: Algorithmus

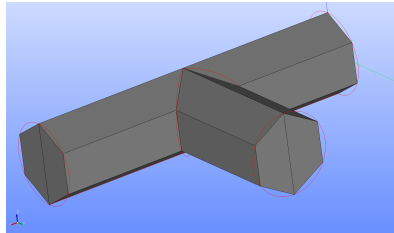
- Zeitentwicklung mit backward differentiation formula (BDF)
- Taylor–Hood Elemente [4]: $(\mathbf{u}, p) \in Q_{k+1}^{\text{dim}} \times Q_k$
- Entkopplung von $(\mathbf{u}, p)$ mit Chorin's Projektionsschema
 - Führt zur Poisson Gleichung für den Druck
- Stabilisierung der Impulsbilanz
 - Taylor–Galerkin Stabilisierung [5] für Taylor–Hood Elemente
 - Zusätzliche Diffusion in Strömungsrichtung
 - Grad–div Stabilisierung [6] zur Forderung von $\nabla \cdot \mathbf{u} = 0$
- Neumann Entwicklung für schnelle Matrizenkonstruktion
- Boussinesq Näherung für die Auftriebskraft

Unstrukturierte Gitter: Beispiel: T-Rohr

- Warum T-Rohr?
 - Zusammengesetzte Körper.
 - Gebogene Flächen (→ Zylindermantelfläche).
 - Unstetige Kanten (→ „Schweißnaht“).
- Entwicklung Verfahren zur Erstellung von zulässigen Startgittern.



CAD Modell.



Startgitter.

Iterationen globaler Verfeinerung:

