

Vorlesung

Einführung in die Praktische Informatik

Wintersemester 2019/2020

Universität Heidelberg

Organisatorisches

Vorlesung

- Dozent
Peter Bastian, INF 205, Raum 1/400, Sprechstunde: Do 11:00–12:00 Uhr
- Übungsleiter
Marian Piatkowski
email: `ipi-ws2019@dune-project.org`
- TutorInnen: Studierende höherer Semester
- Webseite zur Vorlesung
https://conan.iwr.uni-heidelberg.de/teaching/info1_ws2019/
Informationen, Unterlagen, Übungsblätter
- Skript
Basierend auf Sadlo (2015), Bastian (2003,2011,2014), Neuss (2006)
Ausgabe gedruckter Exemplare in der ersten Vorlesung

Übungen

- **Sinn der Übungsgruppen**
Vertiefen Stoff der Vorlesung
Jede Woche wird ein Blatt mit Aufgaben ausgegeben
Besprechen der Aufgaben in den Übungsgruppen
Fragen zur Vorlesung, Übungen und Programmieren stellen
- **Derzeit 20 Übungsgruppen**, Abgabe in Gruppen je 2–3 Teilnehmer
(keine Einzel-Abgaben!)
- **Anmeldung über MÜSLI**
<https://muesli.mathi.uni-heidelberg.de/lecture/view/1106>
Anmeldungszeitraum: **bis So., 20.10., 23:59 Uhr**
Einteilung **am Montag 21.10.19** (aktuelle Einteilung gilt nicht!)
System arbeitet mit Präferenzen und **nicht first come first served**
Ergebnis der Einteilung: In MÜSLI einloggen!

Ablauf der Übungen

- **Ausgabe der Übungsblätter**
Donnerstags, 16 Uhr auf der Webseite der Vorlesung
- **Abgabe der Lösungen**
Donnerstags, 14 Uhr ct (**vor** der Vorlesung)
Abgabe in Zettelkästen vor dem Dekanat, Mathematikon, 1. Stock
- **Erstes Blatt**
Ausgabe: 24.10.19 (**nächste Woche!**)
Abgabe: 31.10.19, 14 Uhr
- **Beginn der Übungsgruppen: Donnerstag, 24.10., 9:00 Uhr**
Kennenlernen, Fragen zu den Aufgaben und der Vorlesung, UNIX Einführung

Übungszeiten

Wann	Wo	Wer
Mo 9 - 11	SR 7	Ferdinand W
Mo 9 - 11	SR 8	Johannes
Mo 9 - 11	SR 9	Frederick
Mo 11 - 13	SR 10	Johannes
Di 9 - 11	SR 6	Nils
Di 9 - 11	SR 7	David
Di 11 - 13	SR Stat	David

Wann	Wo	Wer
Mi 9 - 11	SR 10	Nils
Mi 9 - 11	SR 7	Ferdinand W
Mi 11 - 13	SR 10	Ferdinand V
Mi 14 - 16	SR 10	Ferdinand V
Mi 14 - 16	SR 11	Julien
Mi 14 - 16	SR 6	Frederick
Do 9 - 11	SR 6	Julien
Do 11 - 13	SR 11	Patrick
Do 16 - 18	SR 10	Patrick
Fr 9 - 11	SR 10	Florian
Fr 11 - 13	SR 7	Florian
Fr 14 - 16	SR 7	Erik
Fr 16 - 18	SR 10	Erik

Leistungsnachweis für 8 LP

- **Erfolgreiche Übungsteilnahme**
mindestens **50% der Punkte** aus den Übungsaufgaben
Präsentation einer Lösung erwünscht
- **Bestehen der Klausur**
Donnerstag, 6. Februar 2020, 14-16 Uhr
Nachklausur vor Beginn Vorlesungszeit Sommersemester
- Wer bereits die erfolgreiche Teilnahme an den Übungen zu der Vorlesung vom letzten Jahr nachweisen kann und zur Prüfung angetreten ist, ist zur Klausur zugelassen.
- Im B.Sc.angewandte Informatik ist dieses Modul die Orientierungsprüfung

Unterschiedliche Vorkenntnisse

Angebote für Anfänger

- Nächste Woche in den Übungsgruppen:
„Grundlagen der Bedienung von UNIX-Systemen“
- Tutoren helfen bei Anfangsschwierigkeiten: Fragen!
- Auch ich freue mich über Fragen in der Vorlesung oder danach!
Es gibt keine dummen Fragen!

Programmierkurs

- Einführung in C++ **unabhängig** von der Vorlesung
- Pflichtveranstaltung für B.Sc. Informatik, daher Priorität für diese Gruppe

Praktisches Üben

- Programmieren ist wesentlicher (nicht alleiniger) Inhalt der Vorlesung
- Beim Programmieren gilt: **Übung macht den Meister** ! Programmieren ist eine Kunst. Eines der berühmtesten Bücher der Informatik von Donald E. Knuth heißt „**The Art of Computer Programming**“
Nutzen Sie alle gebotenen Möglichkeiten zum Üben!
- In der Vorlesung/Übung benutzen wir eine **UNIX-Programmierungsumgebung**. Sie sollten Zugang zu so einem System haben um die Übungen durchführen zu können. Geeignet sind LINUX, Mac oder ein Windows-System mit virtueller Maschine oder Linux Subsystem (W10)
- Informationen gibt es auf der Webseite . . . oder ihrem Tutor

Dozent

- **Peter Bastian**

1989 Diplom Informatik (Uni Erlangen-Nürnberg)

1994 Promotion Mathematik (Uni Heidelberg)

1999 Habilitation Scientific Computing (Uni Kiel)

2001 Professur (IWR, Uni Heidelberg)

2006 Professur (Informatik, Uni Stuttgart)

2001 Professur (IWR, Uni Heidelberg)

- **Arbeitsgebiete**

Numerik partieller Differentialgleichungen

Höchstleistungsrechnen

Anwendungen: Strömungsmechanik, poröse Medien

Einführung und Motivation

Was ist Informatik

Wissenschaft von der systematischen Verarbeitung von Information, besonders der automatischen Verarbeitung mit Hilfe von Digitalrechnern

Wikipedia, Duden Informatik

Computer science is no more about computers than astronomy is about telescopes, biology is about microscopes or chemistry is about beakers and test tubes.

Michael R. Fellows and Ian Parberry, Computing Research News, January 1993

Inhalt der Vorlesung

- Grundlegende Konzepte der Informatik kennenlernen
z. B. Formale Sprachen, Turingmaschine, Algorithmenbegriff, Komplexität, Abstraktion, Rekursion, Graphen, . . .
- Algorithmisches Denken schulen
Problem → Algorithmus → Programm
- Programmieren im Kleinen
Programmierstile (funktional, prozedural, objektorientiert, generisch),
Erlernen der Programmiersprache C++
Aber: Vorbereitung für Programmieren im Großen!
- Grundlegende Algorithmen und Datenstrukturen
Exhaustive search, Backtracking, numerische Verfahren, Suchen, Sortieren, . . .
Listen, Felder, Heaps, Stacks, Graphen, Bäume, . . .

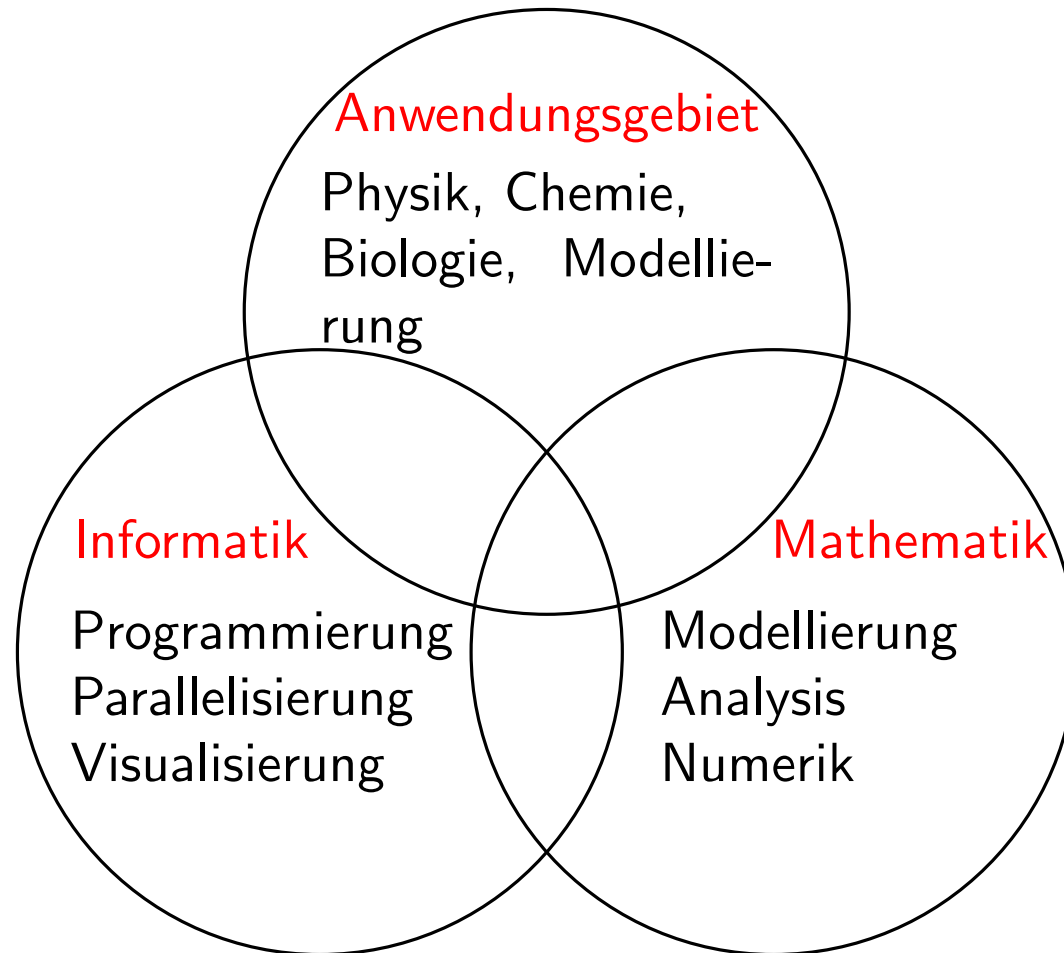
Informatik als Wissenschaft

- Wortschöpfung aus „Information“ und „Automatique“ erstmals benutzt von Philippe Dreyfus (1962, laut Wikipedia).
- Grundlagen waren
 - Theorie der Berechenbarkeit (Turing, Church, 1937)
 - Entwicklung elektromechanischer/elektronischer Rechenmaschinen (Z3, 1941, ENIAC, 1946)
 - entsprechende Anwendungen (Kryptographie, ballistische Berechnungen, Differentialgleichungen lösen)
- Erster deutscher Informatikstudiengang WS 1968/69 in Karlsruhe

Teilgebiete der Informatik

- **Theoretische Informatik**
Logik und Berechenbarkeit, Automatentheorie und formale Sprachen, Semantik, Komplexitätstheorie
- **Technische Informatik**
Elektrotechnische Grundlagen, Architektur von Rechenanlagen, Chipentwurf, Netzwerkkomponenten, Fehlertoleranz, . . .
- **Praktische und Angewandte Informatik**
Betriebssysteme, Softwareengineering, Datenbanken, Programmiersprachen, Visualisierung, Mensch-Maschine-Interaktion
Anwendungen, z. B. **Wissenschaftliches Rechnen**

Wissenschaftliches Rechnen



Ableitung

Vorbemerkung: Der nachfolgende Stoff dient der Motivation von Informatik UND Mathematik (hoffentlich nicht Abschreckung) und ist NICHT Inhalt der weiteren Vorlesung!

f eine Funktion in einer Variablen:

$$f : \mathbb{R} \rightarrow \mathbb{R}$$

Ableitung:

$$\frac{df}{dx}(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

Partielle Ableitung

f eine Funktion in mehreren Variablen (x, y, z, t) :

$$f : \mathbb{R}^3 \rightarrow \mathbb{R}$$

Partielle Ableitung bezüglich der Variablen x :

$$\frac{\partial f}{\partial x}(x, y, z) = \lim_{h \rightarrow 0} \frac{f(x + h, y, z) - f(x, y, z)}{h}$$

Differentialoperatoren

Gradient:

$$\nabla f(x, y, z) = \begin{pmatrix} \frac{\partial f}{\partial x}(x, y, z) \\ \frac{\partial f}{\partial y}(x, y, z) \\ \frac{\partial f}{\partial z}(x, y, z) \end{pmatrix}$$

Divergenz einer vektorw. Funktion $f(x, y, z) = (f_x(x, y, z), f_y(x, y, z), f_z(x, y, z))^T$:

$$\nabla \cdot \vec{f}(x, y, z) = \frac{\partial f_x}{\partial x}(x, y, z) + \frac{\partial f_y}{\partial y}(x, y, z) + \frac{\partial f_z}{\partial z}(x, y, z)$$

Laplace einer skalaren Funktion:

$$\Delta f(x, y, z) = \frac{\partial^2 f}{\partial x^2}(x, y, z) + \frac{\partial^2 f}{\partial y^2}(x, y, z) + \frac{\partial^2 f}{\partial z^2}(x, y, z)$$

Differentialgleichungen

Gewöhnliche Differentialgleichung:

$$\frac{df}{dt}(t) = g(t, f(t)) \quad t \in [a, b], \quad f(a) = g_0$$

Partielle Differentialgleichung:

$$\frac{\partial^2 f}{\partial x^2}(x, y, z) + \frac{\partial^2 f}{\partial y^2}(x, y, z) + \frac{\partial^2 f}{\partial z^2}(x, y, z) = g(x, y, z) \quad (x, y, z) \in \Omega \subset \mathbb{R}^3$$

$$f(x, y, z) = \gamma(x, y, z) \quad (x, y, z) \in \partial\Omega$$

Ableitungen wurden erfunden um Differentialgleichungen zu formulieren!

Sternentstehung (Strömungsmechanik)



Cone nebula from <http://www.spacetelescope.org/images/heic0206c/>

Euler-Gleichungen

Ein Modell zur Sternentstehung stellen die Euler-Gleichungen (Leonhard Euler, 1707-1783) der Gasdynamik mit Gravitation dar. Diese sind ein nichtlineares System partieller Differentialgleichungen:

$$\partial_t \rho + \nabla \cdot (\rho v) = 0 \quad (\text{Massenerhaltung})$$

$$\partial_t(\rho v) + \nabla \cdot (\rho v v^T + pI) = -\rho \nabla \Psi \quad (\text{Impulserhaltung})$$

$$\partial_t e + \nabla \cdot ((e + p)v) = -\rho \nabla \Psi \cdot v \quad (\text{Energieerhaltung})$$

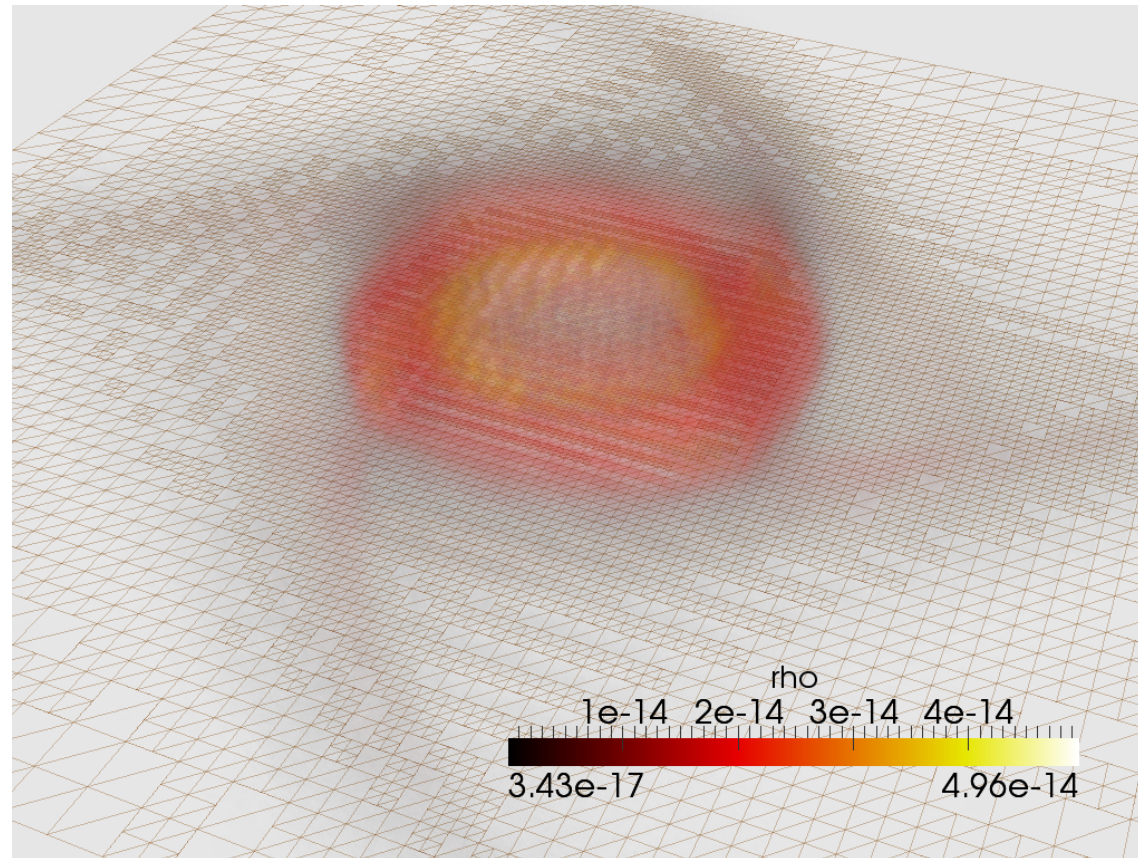
$$\Delta \Psi = 4\pi G \rho \quad (\text{Gravitationspotential})$$

Bessere Modelle beinhalten innere Reibung (Navier-Stokes Gleichungen), erweiterte Zustandsgleichung und Strahlungstransport.

Existenz und Regularität der inkompressiblen Navier-Stokes-Gleichungen ist eines der sieben Millennium Prize Problems.

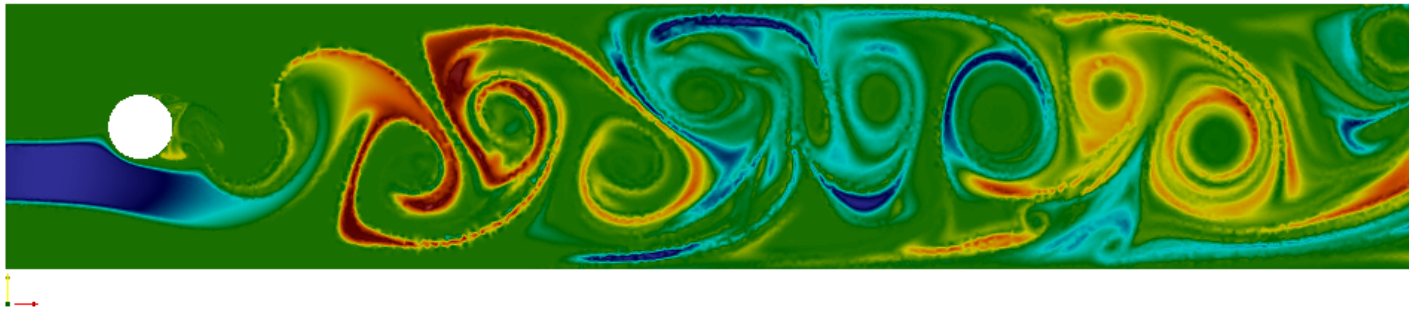
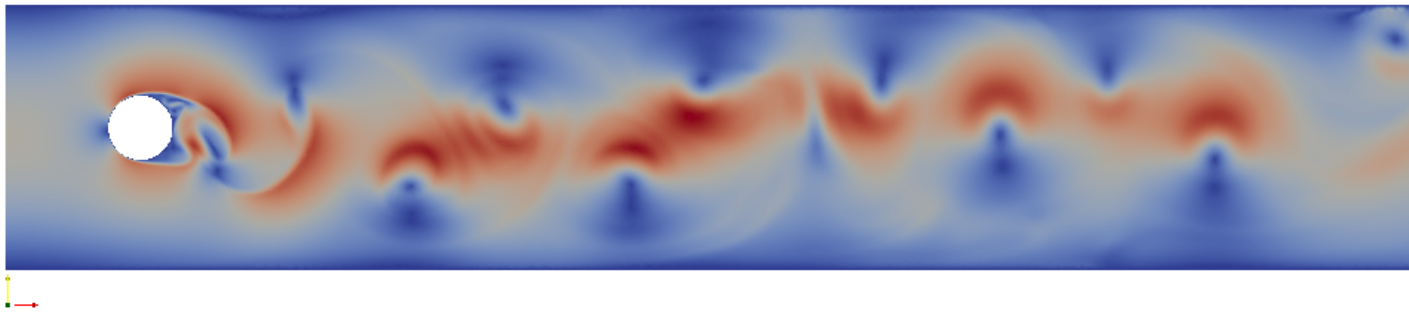
Numerische Simulation der Sternentstehung

durchgeführt von Marvin Tegeler in seiner Diplomarbeit (2011).

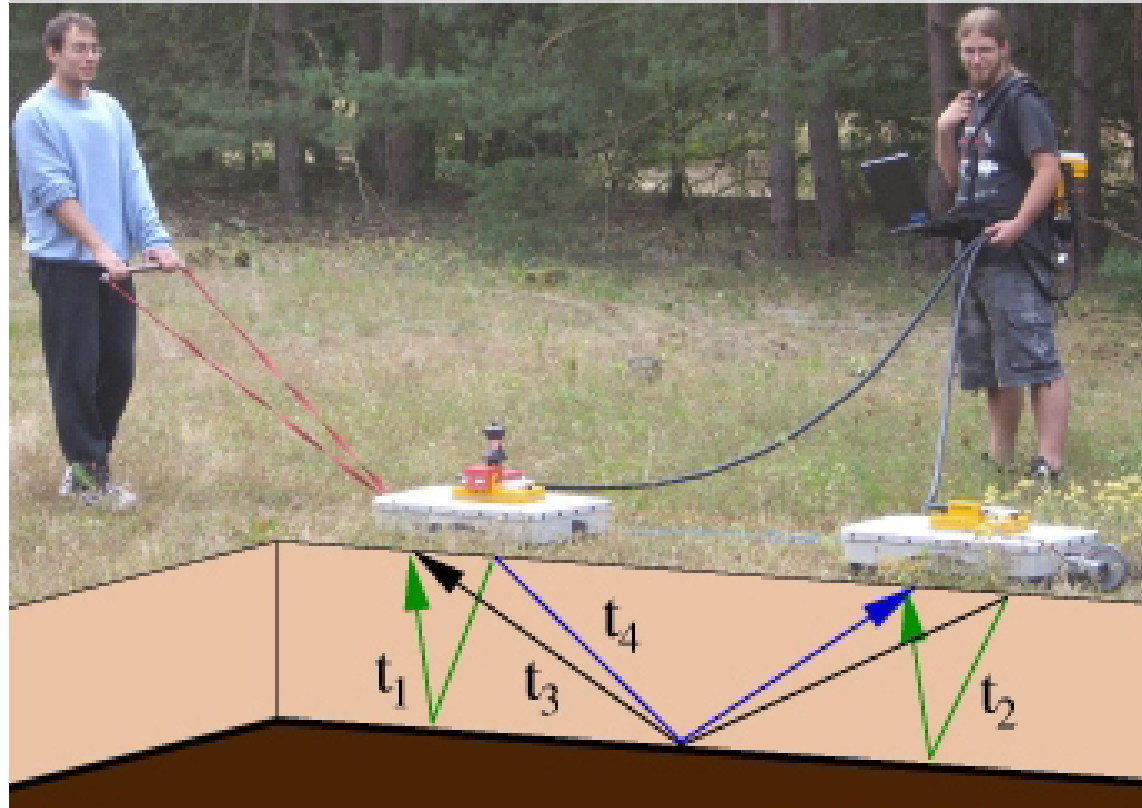


Von Karmansche Wirbelstraße @ Re 1500

Diplomarbeit Marian Piatkowski



Bodenradar



Bestimme Strukturen im Boden durch Reflexion von Radarwellen

(Makroskopische) Maxwell-Gleichungen

beschreiben die Ausbreitung elektromagnetischer Wellen und wurden von James Clerk Maxwell im Jahr 1861 angegeben.

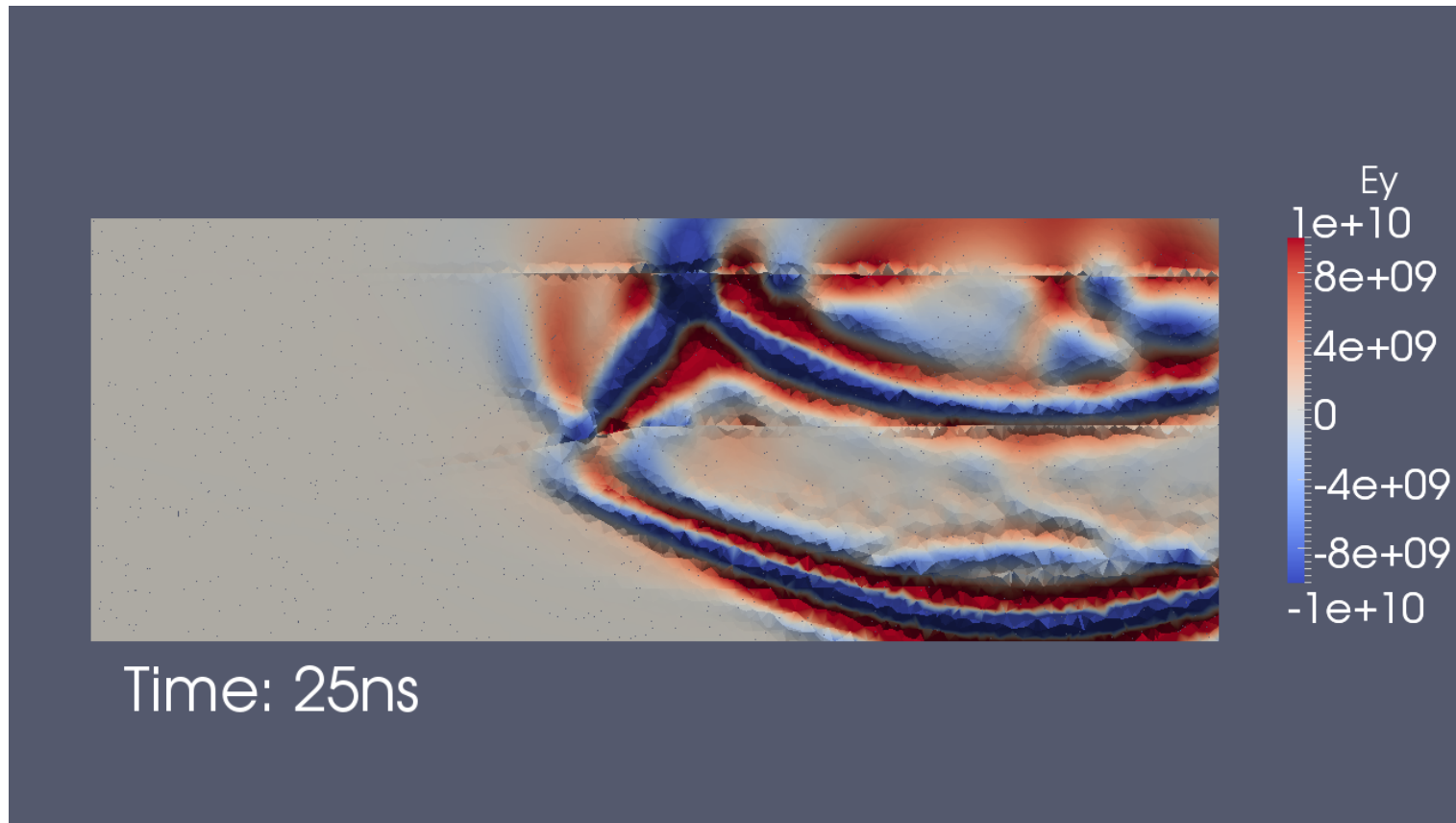
System linearer partieller Differentialgleichungen erster Ordnung:

$$\begin{aligned}\nabla \times E &= -\partial_t B && \text{(Faraday)} \\ \nabla \times H &= j + \partial_t D && \text{(Ampère)} \\ \nabla \cdot D &= \rho && \text{(Gauß)} \\ \nabla \cdot B &= 0 && \text{(Gauß für Magnetfeld)} \\ D &= \epsilon_0 E + P && \text{(elektrische Flussdichte)} \\ H &= \mu_0^{-1} B - M && \text{(magnetische Feldstärke)}\end{aligned}$$

plus Rand- und Anfangsbedingungen

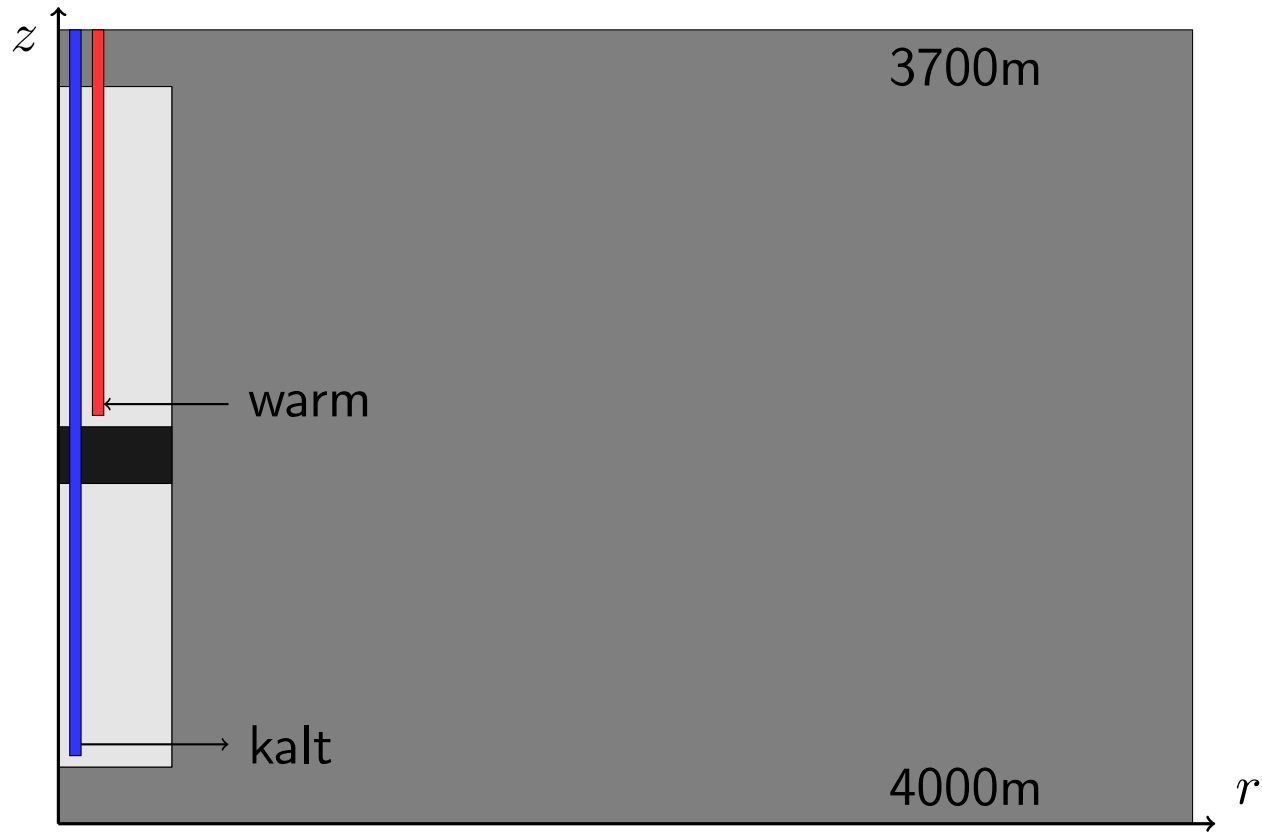
Simulation des Bodenradars

Jorrit Fahlke (IWR, 2011)



Eine Geothermieanlage

Einlochanlage in einem tiefen Aquifer (Zweidimensionaler Schnitt)



Gekoppelte Wasser- und Wärmeströmung

System nichtlinearer partieller DGL für Druck p und Temperatur T :

$$\partial_t(\phi\rho_w) + \nabla \cdot \{\rho_w u\} = f \quad (\text{Massenerhaltung})$$

$$u = \frac{k}{\mu}(\nabla p - \rho_w g) \quad (\text{Darcy-Gesetz})$$

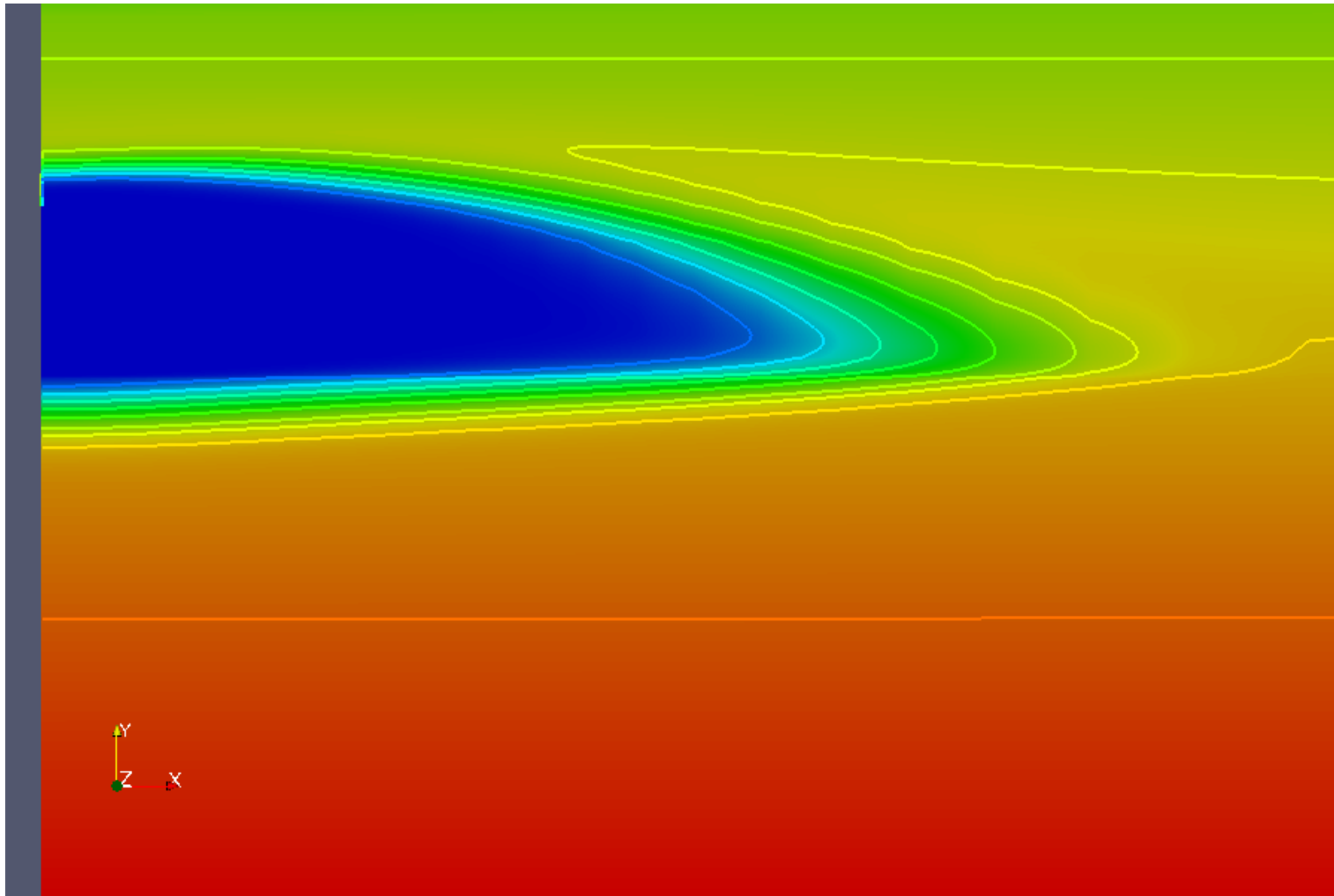
$$\partial_t(c_e\rho_e T) + \nabla \cdot q = g \quad (\text{Energieerhaltung})$$

$$q = c_w\rho_w u T - \lambda\nabla T \quad (\text{Wärmefluss})$$

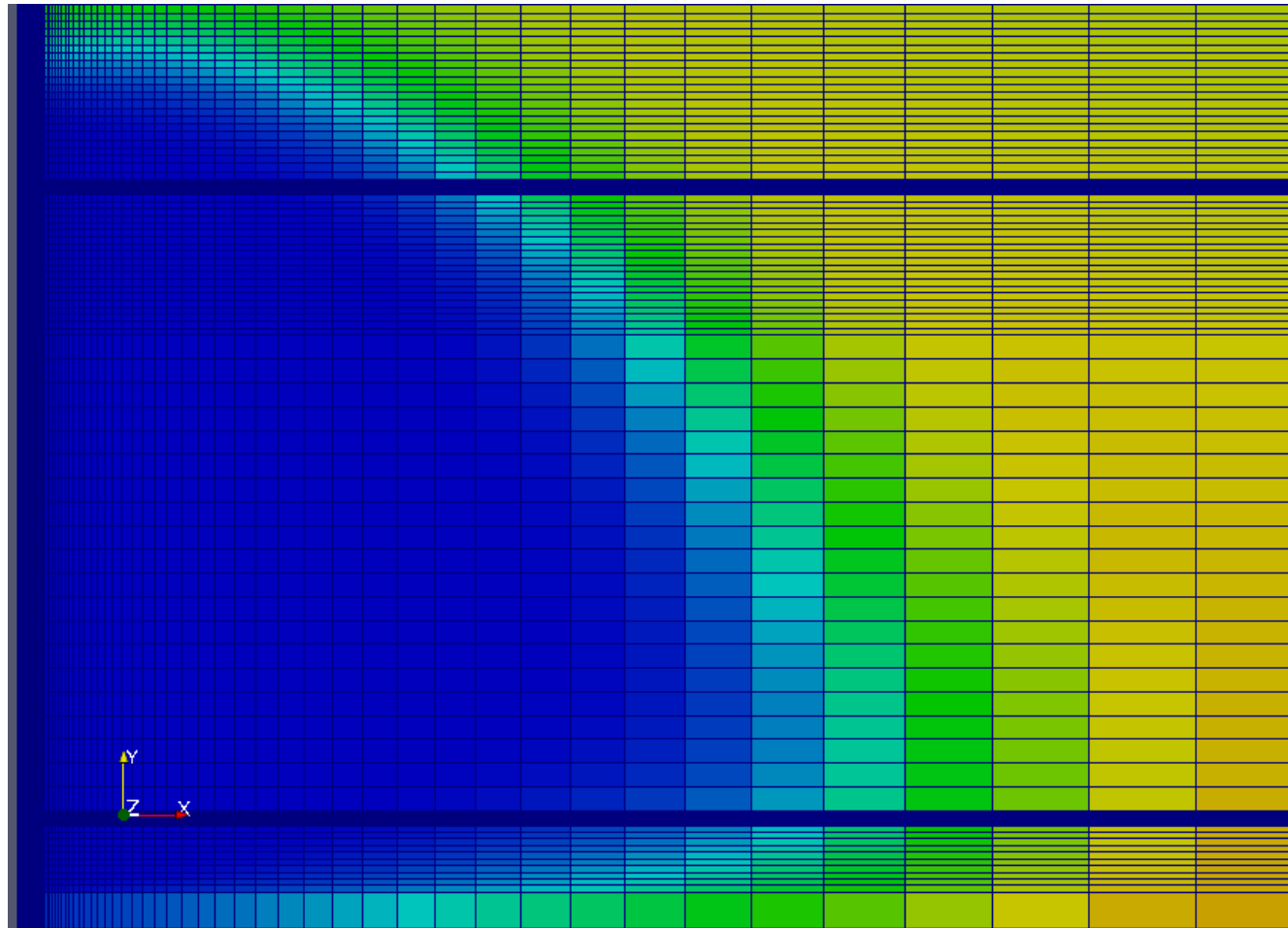
Nichtlinearität: $\rho_w(T)$, $\rho_e(T)$, $\mu(T)$

Permeabilität $k(x)$: 10^{-7} im kiesgefüllten Bohrloch, 10^{-16} im Verschluss

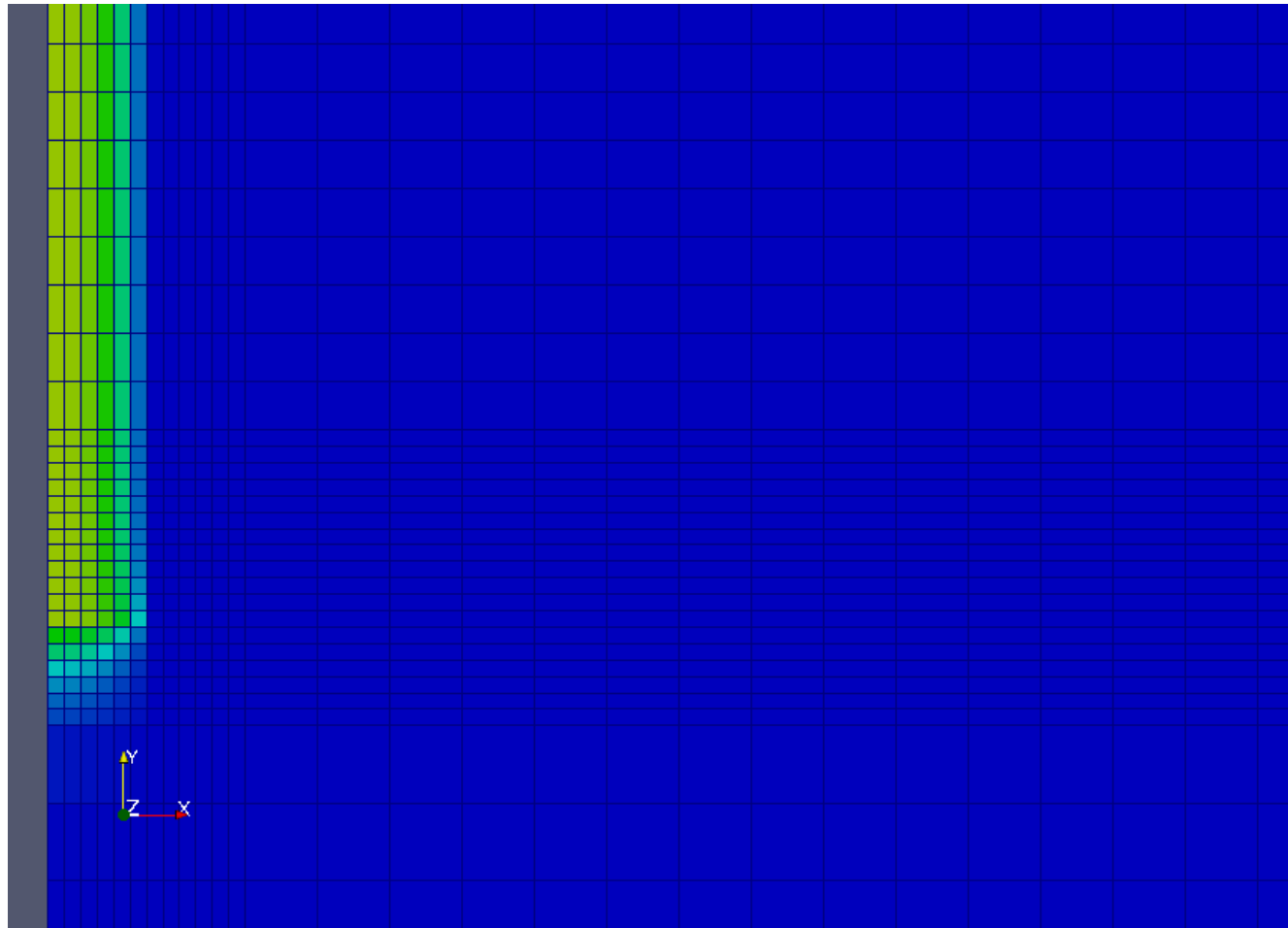
Raum-/Zeitskalen: $R=15$ km, $r_b=14$ cm, Sekunden (0.3 m/s im Bohrloch) bis Jahre



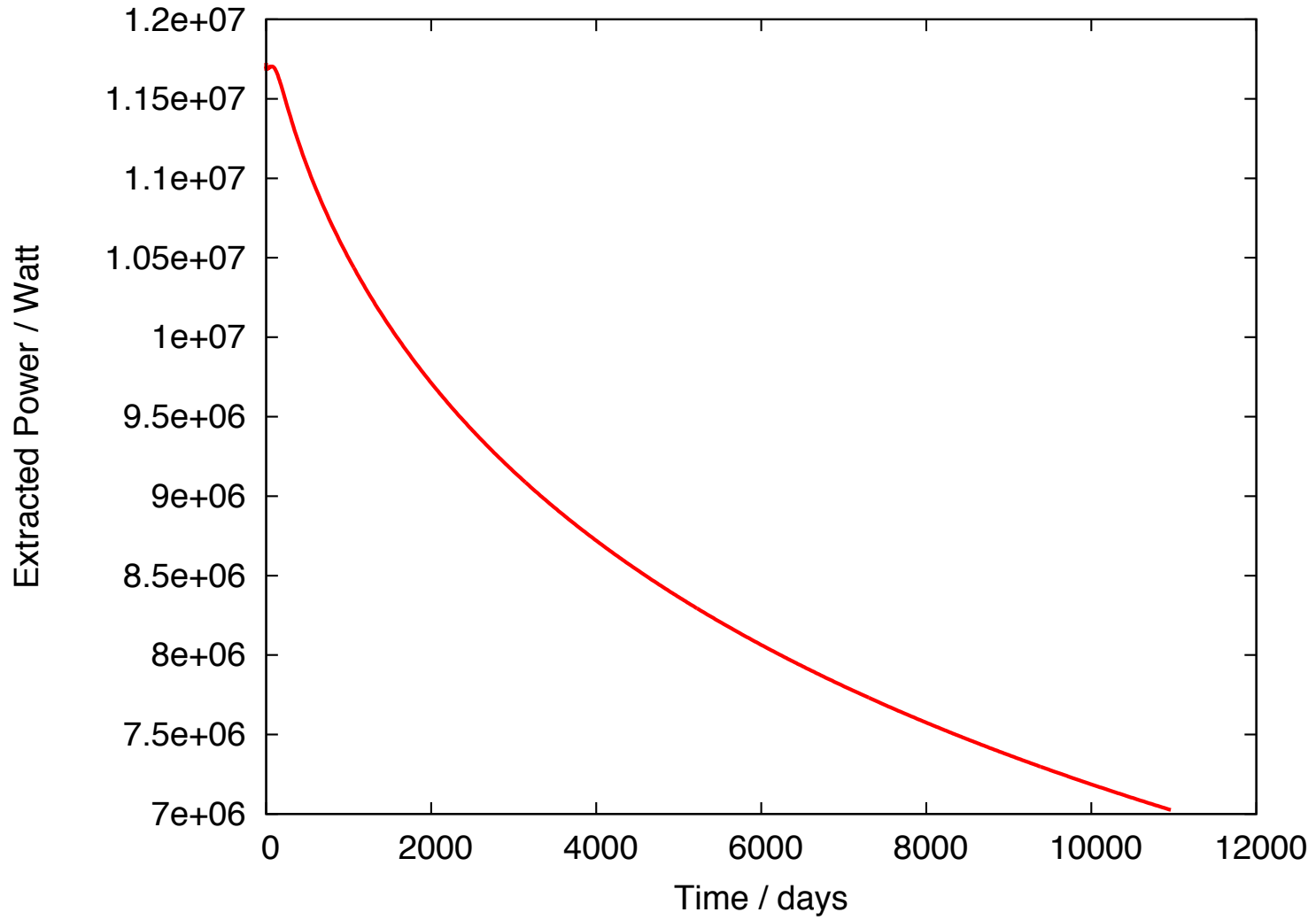
Temperaturverlauf nach 30 Jahren Betrieb



Detail am Einspeisebereich



Temperatur im Bohrloch



Entzugsleistung über 30 Jahre

Dichtegetriebene Strömung

in einem porösen Medium

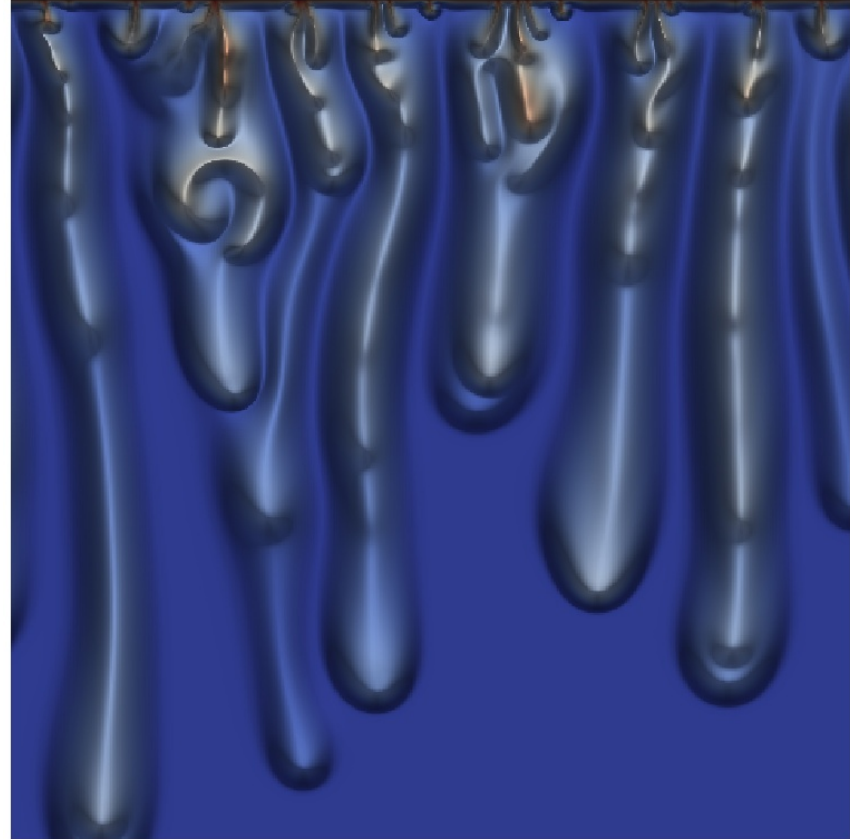
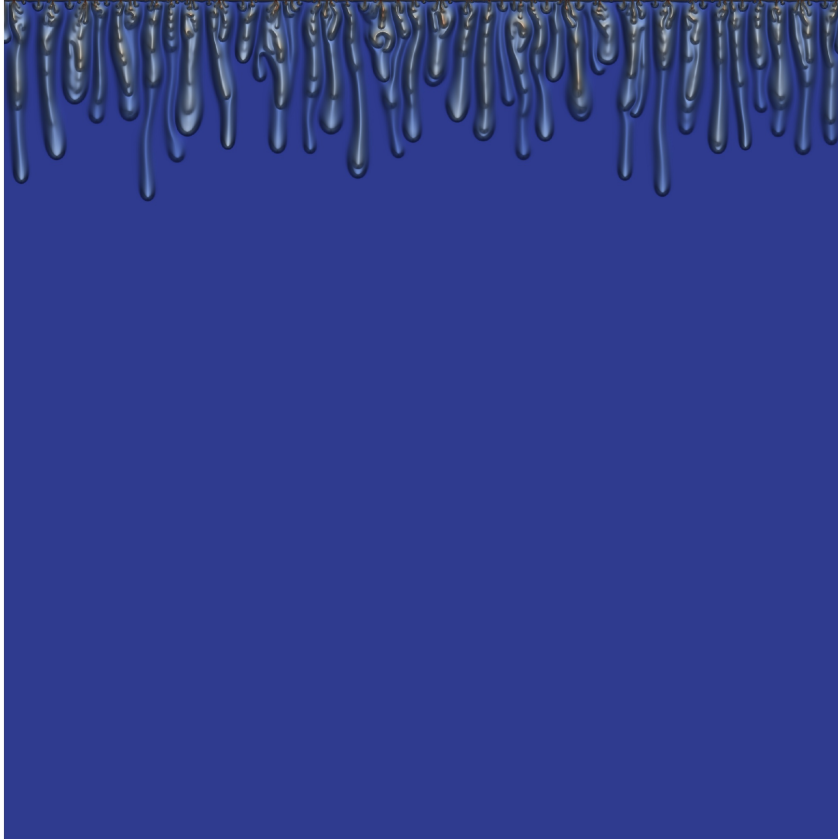
$$\nabla \cdot v = 0, \quad v = -(\nabla P - \omega_s \mathbf{1}_z)$$
$$\partial_t \omega_s + \nabla \cdot \left(v \omega_s - \frac{1}{Ra} \nabla \omega_s \right) = 0$$

Dichteres Fluid über weniger dichtem Fluid führt zu instabiler Strömung

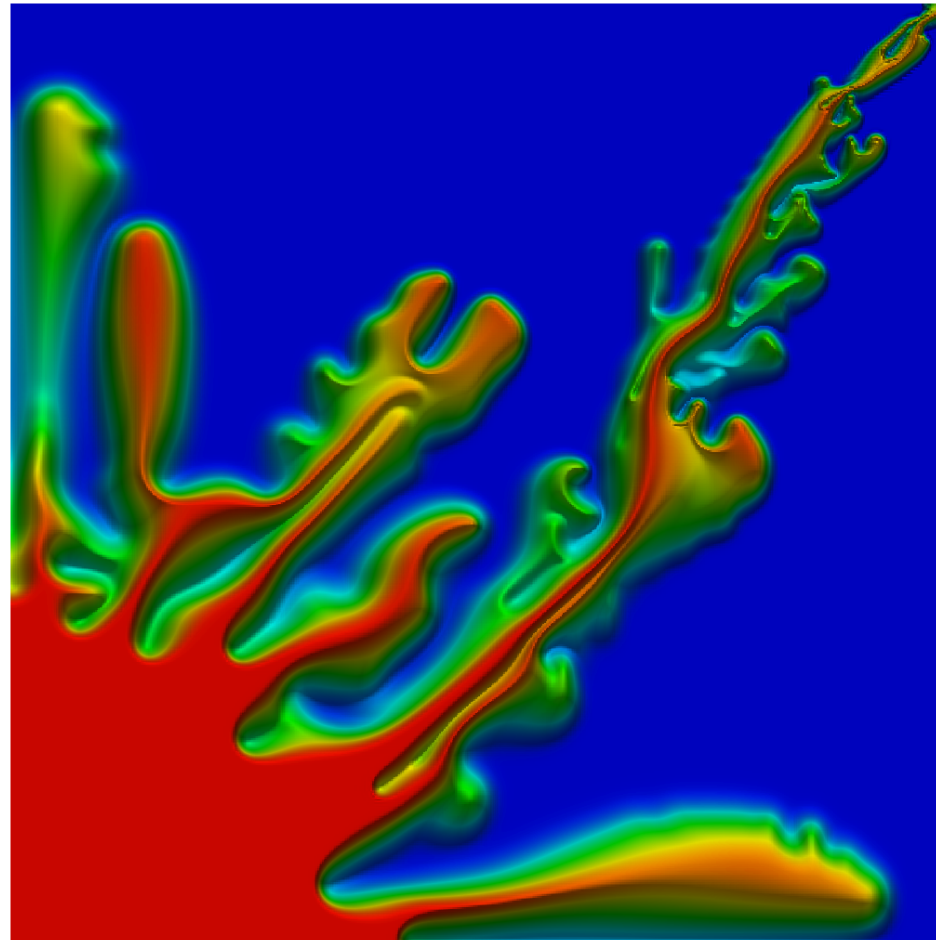
Erhöht die Durchmischung, z. B. bei der Sequestrierung von CO₂

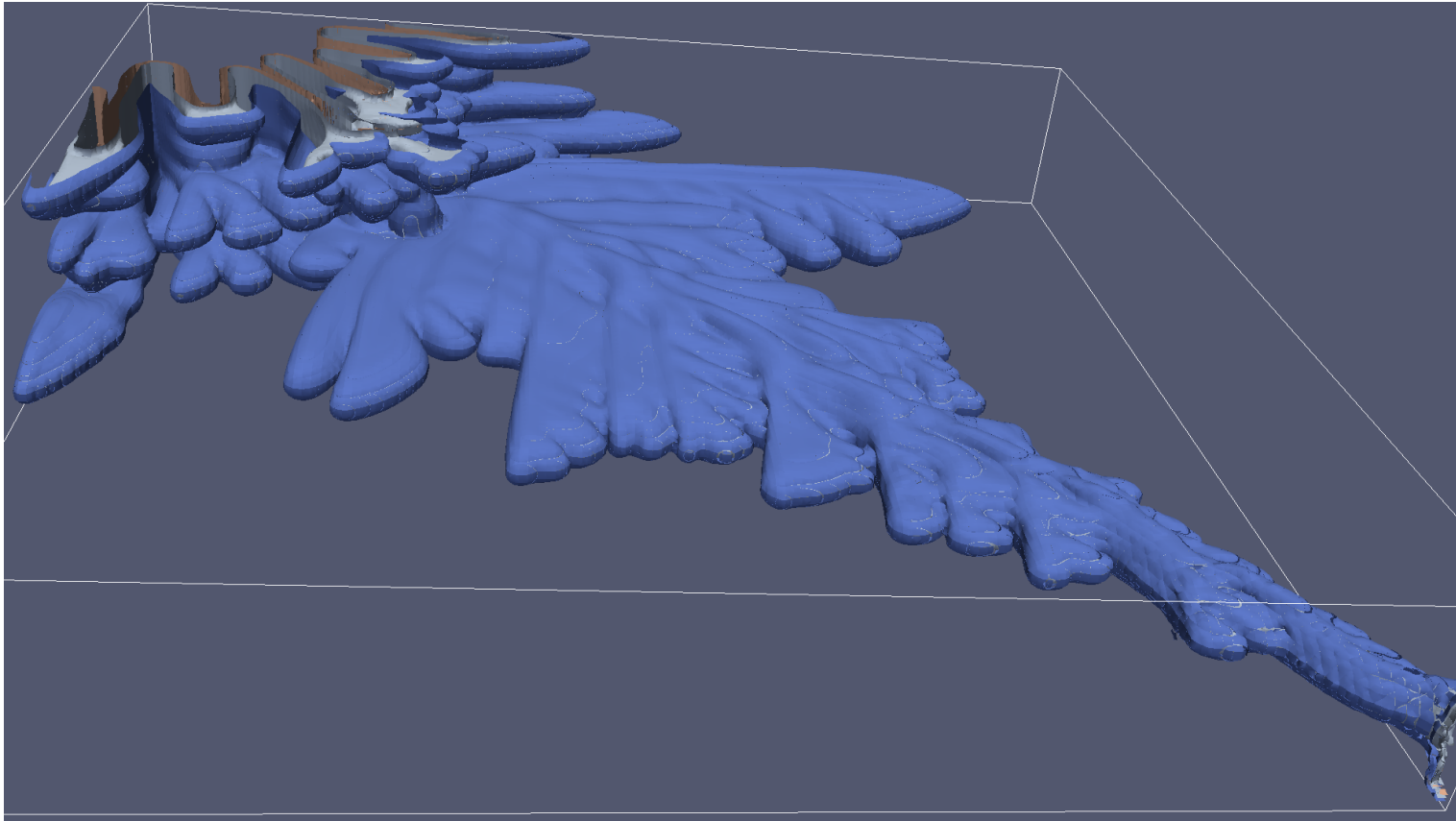
Wichtiger Effekt bei geophysikalischen Strömungen (dort: Navier-Stokes Gleichung)

Zweidimensionale Simulation:



Viscous Fingering





Alle Simulationen wurden mit dem Softwarerahmenwerk DUNE durchgeführt

<http://www.dune-project.org>

Viele weitere Anwendungen

- Wetter und Klima
- Ölreservoirsimulation, CO₂ Sequestrierung, Lagerung radioaktiver Abfälle
- Geophysikalische Strömungen im Erdinnern
- Tsunamisimulation
- Festigkeit von Materialien
- Brennstoffzellen
-

Numerische Lösung partieller Differentialgleichungen

Physik des 19. Jahrhunderts, aber Mathematik und Informatik des 20. Jahrhunderts!

Treibende Kraft bei der Entwicklung des Computers, insbesondere Höchstleistungsrechner:

An automatic computing system is a (usually highly composite) device, which can carry out instructions to perform calculations of a considerable order of complexity — e.g. to solve a non-linear partial differential equation in 2 or 3 independent variables numerically.

John von Neumann, First Draft of a Report on the EDVAC, 30. Juni 1945

Verbindet Informatik, Physik und Mathematik

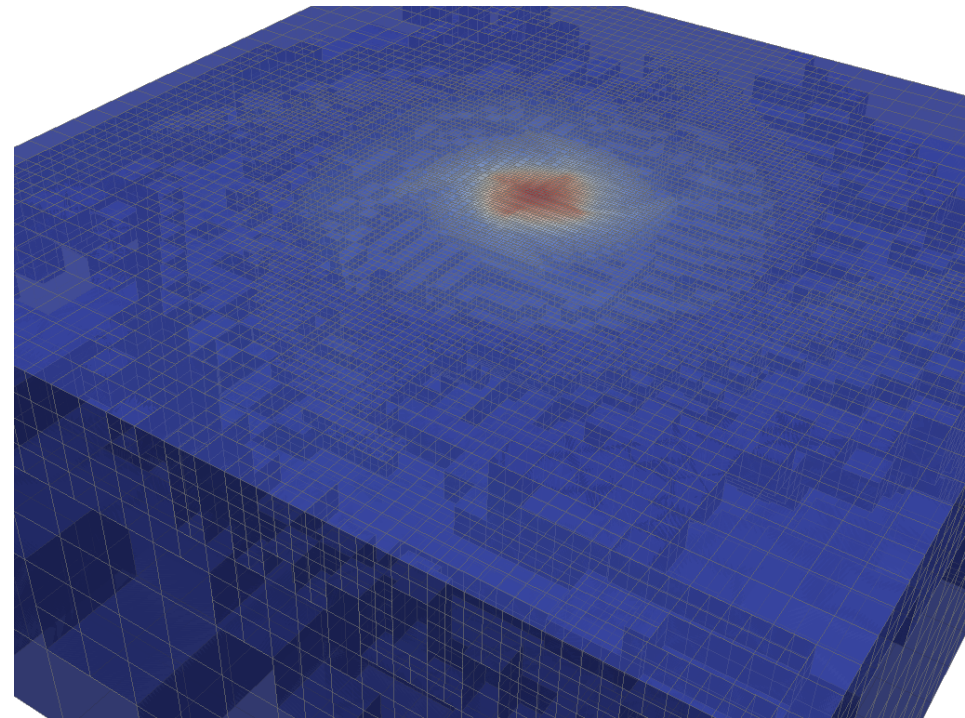
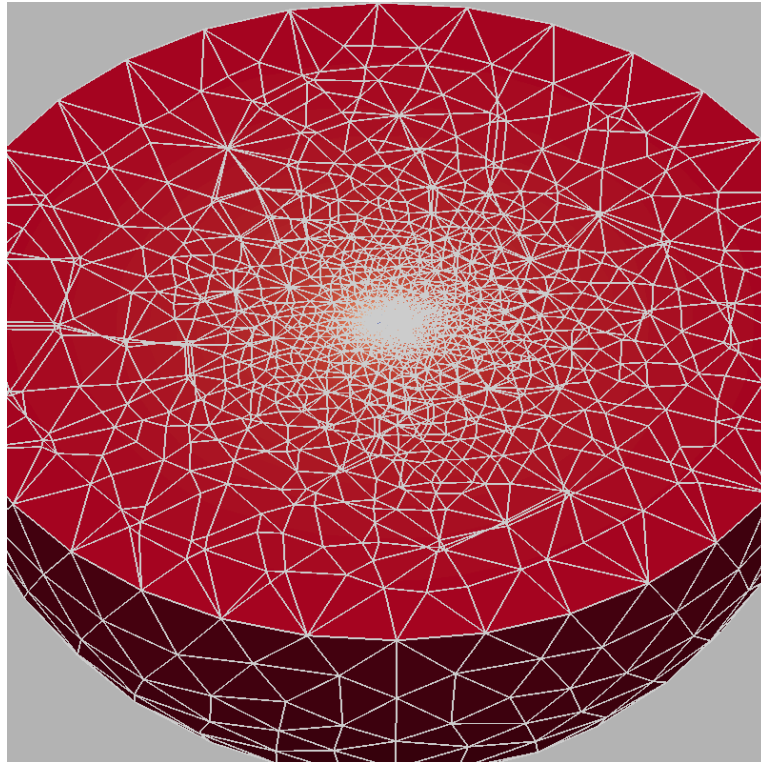
Etwa Jugene (294.912 Cores, Nummer 2 in Europa im Jahr 2011)



Ermöglicht z. B. die iterative Lösung von (bestimmten) linearen Gleichungssystemen mit 10^{11} Unbekannten in 4 Minuten.

#1 TOP500 Juni 2019: Summit mit 150 Petaflop = $0.15 \cdot 10^{18}$ Flop/s

Erfordert komplexe Algorithmen und Datenstrukturen, z. B. zur adaptiven Triangulierung:



Umfangreiches Softwareprojekt: <http://www.dune-project.org>

Literatur

H.-J. Appelrath, J. Ludewig: Skriptum Informatik – eine konventionelle Einführung. B. G. Teubner Verlag, 5. Auflage, 2000.

H. Abelson, G. J. Sussman mit J. Sussman: Struktur und Interpretation von Computerprogrammen, Springer Verlag, 1998.

B. Stroustrup: The C++ Programming Language, Addison-Wesley, 4. Auflage, 2013.

C++ Programming, freies WikiBook http://en.wikibooks.org/wiki/C%2B%2B_Programming

U. Schöning: Ideen der Informatik. Oldenburg Verlag, 2002.

D. R. Hofstatter: Gödel, Escher Bach: Ein Endloses Geflochtenes Band. dtv Taschenbuch, 11. Auflage, 2007.