Numerical Methods and Software

Introduction

Uwe Naumann

Informatik 12:
Software and Tools for Computational Engineering (STCE)

RWTH Aachen University

Contents

Cover Story

Motivation

Introduction

Admin

Summary and Next Steps

Outline

Cover Story

Motivation

Introduction

Admin

Summary and Next Steps

Cover Story

Modern Family

ⓒmy wife and me

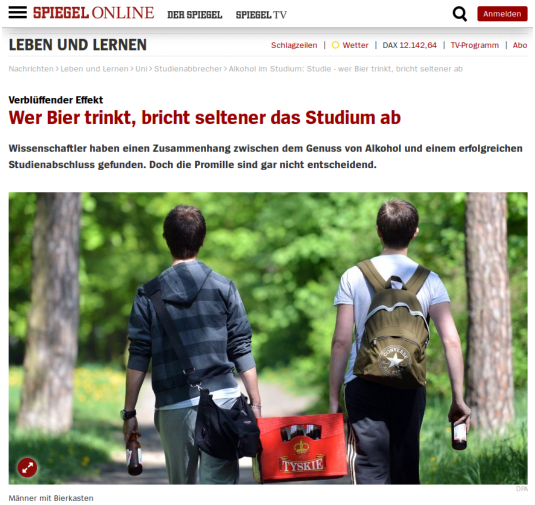
Cover Story

Gender Equality

©dex1.info: Dinge, die Männer an Frauen unattraktiv finden

Cover Story

Indeed ...



SPIEGEL ONLINE DER SPIEGEL SPIEGEL TV Anmelden

LEBEN UND LERNEN

Schlagzeilen | Wetter | DAX 12.142,64 | TV-Programm | Abo

Nachrichten > Leben und Lernen > Uni > Studienabbrecher > Alkohol im Studium: Studie - wer Bier trinkt, bricht seltener ab

Verblüffender Effekt

Wer Bier trinkt, bricht seltener das Studium ab

Wissenschaftler haben einen Zusammenhang zwischen dem Genuss von Alkohol und einem erfolgreichen Studienabschluss gefunden. Doch die Promille sind gar nicht entscheidend.

DPA

Männer mit Bierkasten

©Spiegel Online

Outline

Cover Story

Motivation

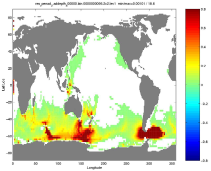
Introduction

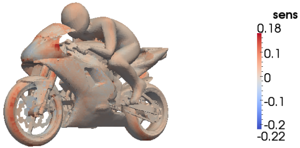
Admin

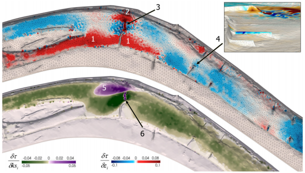
Summary and Next Steps

Motivation

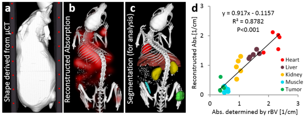
Numerical Simulation





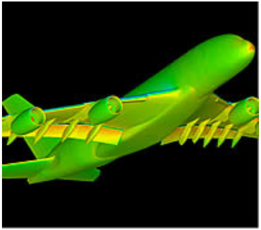


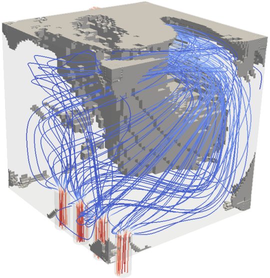
1
1
2
3
4
5
6
$\frac{\delta\tau}{\delta ks_i}$
-0,04
-0,02
0
0,02
0,04
-0,05
0,05
$\frac{\delta\tau}{\delta z_i}$
-0,08
-0,04
0
0,04
0,08
-0,1
0,1

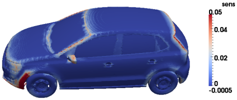


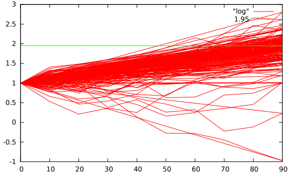
Motivation

Numerical Simulation









Motivation

Aiming for ...

- accuracy
- efficiency
- sustainability

... of numerical simulations.

This course provides basic understanding of selected numerical methods including corresponding software support.

Motivation

Accuracy of Numerical Simulation

```
#include<iostream>

float f(float x) { return x; }

int main() {
  float x=1; const float h=1e-7; // increase gradually ...
  std::cout << (f(x+h)-f(x))/h << std::endl;
  return 0;
}
```

Note: We expect the derivative (as well as it finite difference approximation) to be equal to one.

Remark: My code examples are written in simple C++. If you do not “speak” C++, then you will pick some of it up during this course as a nice side effect :-)

Motivation

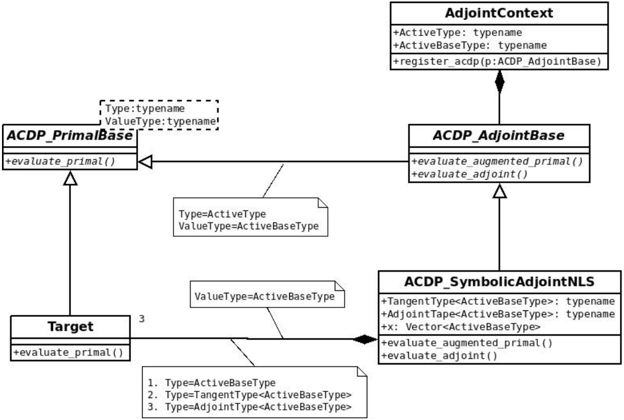
Efficiency of Numerical Simulation

```
#include ”Eigen/Dense”
#include <cassert>
#include <iostream>

int main(int c, char* v[]) {
  assert(c==2);
  int n=std::stoi(v[1]);
  using T=float;
  // using T=double;
  typedef Eigen::Matrix<T,Eigen::Dynamic,Eigen::Dynamic> MT;
  typedef Eigen::Matrix<T,Eigen::Dynamic,1> VT;
  MT A=MT::Random(n,n), B=MT::Random(n,n);
  VT C=VT::Random(n);
  MT D=A*B*C;
  // MT D=A*(B*C);
  std::cout.precision(7);
  // std::cout.precision(15);
  std::cout << D << std::endl;
  return 0;
}
```

Motivation

Sustainability of Numerical Simulation


AdjointContext
+ActiveType: typename
+ActiveBaseType: typename
+register_acdp(p:ACDP_AdjointBase)
Type:typename
ValueType:typename
ACDP_PrimalBase
+evaluate_primal()
ACDP_AdjointBase
+evaluate_augmented_primal()
+evaluate_adjoint()
Type=ActiveType
ValueType=ActiveBaseType
ACDP_SymbolicAdjointNLS
+TangentType<ActiveBaseType>: typename
+AdjointTape<ActiveBaseType>: typename
+x: Vector<ActiveBaseType>
+evaluate_augmented_primal()
+evaluate_adjoint()
ValueType=ActiveBaseType
Target
+evaluate_primal()
3
1. Type=ActiveBaseType
2. Type=TangentType<ActiveBaseType>
3. Type=AdjointType<ActiveBaseType>

Use your / aquire software engineering skills!

Outline

Cover Story

Motivation

Introduction

Admin

Summary and Next Steps

Reasons for not just using pen and paper ...

- mathematically unknown solution needs to be approximated [iteratively]
- mathematically known solution needs to be evaluated repeatedly over a short time period

Ability of computers to perform a very large number of elemental algorithmic steps within a short time is exploited.

Efficient, robust, flexible and sustainable numerical solutions require good knowledge of computer infrastructure (programming language, compiler, libraries, hardware and run time support)

Idea Behind This Course

- introduction to numerical computing for undergraduate CS students
- essential prerequistes for further specialization in diverse subareas of CS (machine learning, speech recognition, computer graphics and vision, high-performance scientific computing ...)
- focus on numerical algorithms and software based on essential calculus and linear algebra (no formal proofs).
- covering
  - 1D case ⇒ visualization
  - *n*D case illustrated by lower-D examples ⇒ visualization

Story in 1D ...

- essential calculus and linear algebra
- error analysis and computer arithmetic
- numerical / algorithmic differentiation
- root finding (linearization → Newton's method)
- unconstrained nonlinear convex optimization (steepest descent, Newton's method)
- linear regression (normal equation, Givens rotation, Householder reflection)
- nonlinear regression (linearization + linear regression)

... and generalization to *n*D case.

Introduction

Recall

- A-Level Math
- Analysis
- Linear Algebra
- Programming

STCE

Research, development and teaching @ STCE is driven by

- algorithmic differentiation
- numerical methods
- combinatorial scientific computing
- program analysis and transformation
- high-performance computing
- simulation software engineering

See www.stce.rwth-aachen.de for more.

Outline

Cover Story

Motivation

Introduction

Admin

Summary and Next Steps

Introduction

Admin

- main lecture (videos; Q&A via Zoom) covers general story including
  - theory
  - algorithms
  - sample prototype implementation
- software lecture / lab (videos and Zoom in second half of semester; t.b.a.) covers numerical software including
  - Eigen
  - GNU Scientific Library
  - NAG Library
- tutorial (Zoom starting in second week) supports main lecture by discussing both pen&paper as well as programming exercises → bonus credits
- learning material comprises videos, slides, code, reference solutions for tutorial exercises, pointers into literature

Bonus Credits

for Tutorial Exercises

You can earn up to two bonus credits for submission of reasonable solutions to

- 60% (single credit)
- 80% (two credits)

of the tutorial exercises.

A single credit improves your final grade (≤ 4.0) by one level (e.g. from 1.7 to 1.3).

Literature

... to complement material presented during lectures/tutorials:

- Heath: Scientific Computing, Second Edition, McGraw-Hill, 1997.
- Dahmen, Reusken: Numerik für Ingenieure und Naturwissenschaftler, 1. Auflage, Springer, 2008.
- Knorrenschildt: Numerische Mathematik, Carl Hanser Verlag, 6. Auflage, 2017.

Outline

Cover Story

Motivation

Introduction

Admin

Summary and Next Steps

Summary and Next Steps

- Numerical simulation is an essential element of mankind's toolbox for understanding the world.

- Computer scientists should to able to use numerical software to support, for example, machine learning, computer graphics, high-performance scientific computing.

- Computer scientists should drive the development of numerical software to ensure sustainability $\Rightarrow$ in-depth knowledge is required.

- You should attend this course ... ;-)