TU Berlin, Scientific Computing
Winter Semester 2021/2022

Slide lecture 1

Jürgen Fuhrmann

juergen.fuhrmann@wias-berlin.de

Me

- Name: Dr. Jürgen Fuhrmann (no, not Prof.)
- Affiliation: Weierstrass Institute for Applied Analysis and Stochastics (WIAS) Berlin;
  Deputy Head, *Numerical Mathematics and Scientific Computing*
- Email: **juergen.fuhrmann@wias-berlin.de**
- Course homepage:
  https://www.wias-berlin.de/people/fuhrmann/SciComp-WS2122/
- Course chat (zulip): https://tub-scicomp-ws-21-22.zulipchat.com/
- Experience/Field of work:
  - Numerical solution of partial differential equations (PDEs)
  - Development, investigation, implementation of finite volume discretizations for nonlinear systems of PDEs
  - Ph.D. on multigrid methods
  - Applications: electrochemistry, semiconductor physics, groundwater...
  - Software development:
    - WIAS code pdelib (http://pdelib.org)
    - Julia PDE solver package `VoronoiFVM.jl` (http://github.dom/j-fu/VoronoiFVM.jl)
    - Languages: C, C++, Python, Lua, Fortran, Julia
    - Visualization: OpenGL, VTK

Admin stuff

- Lectures will be recorded
- Videos + Slides + Julia notebooks will be available via https://www.wias-berlin.de/people/fuhrmann/SciComp-WS2122/
- Weekly material uploads by Wed night (hopefully)
- Official lecture time Fri 14-16 will be used for in-person Q&A. Will try to translate this via zoom, links will be provided in the chat or per email. Probably I will need to split the course into two halves meeting bi-weekly.
- I will use the email address used for enrolling for all communication, zulip invitations etc. Please keep me informed about any changes.
- All code examples and assignments will be in Julia, either as notebooks or as Julia files. Things should work on Linux, MacOSX, Windows
- Assignments and course projects will given to groups of three - I will set up a google doc for this purpose
- Examinations will be based on coding projects provided around mid of Dezember

Introduction

About computers and (scientific) computing

There was a time when “computers” were humans

Harvard Computers, circa 1890

By Harvard College Observatory - Public Domain

https://commons.wikimedia.org/w/index.php?curid=10392913

HARVARD COLLEGE OBSERVATORY.

CIRCULAR 173.

PERIODS OF 25 VARIABLE STARS IN THE SMALL MAGELLANIC CLOUD.

The following statement regarding the periods of 25 variable stars in the Small Magellanic Cloud has been prepared by Miss Leavitt.

A Catalogue of 1777 variable stars in the two Magellanic Clouds is given in H.A. 60, No. 4. The measurement and discussion of these objects present problems of unusual difficulty, on account of the large area covered by the two regions, the extremely crowded distribution of the stars contained in them, the faintness of the variables, and the shortness of their periods. As

It was about science – astronomy

Computations of course have been performed since ancient times. One can trace back the termin “computer” applied to humans at least until 1613.

The “Harvard computers” became very famous in this context. Incidently, they were mostly female. They predate the NASA “hidden figures” who enabled the first US spacefligths.

WEATHER PREDICTION

BY

NUMERICAL PROCESS

Second edition

BY

LEWIS F. RICHARDSON, B.A., F.R.Met.Soc., F.Inst.P.

FORMERLY SUPERINTENDENT OF ESKDALEMUIR OBSERVATORY

LECTURER ON PHYSICS AT WESTMINSTER TRAINING COLLEGE

L.F.Richardson 1922: 64000 human computers sit in rooms attached to a transparent cupola, they project their results which are combined by some main computers at the center

Does this scale ?

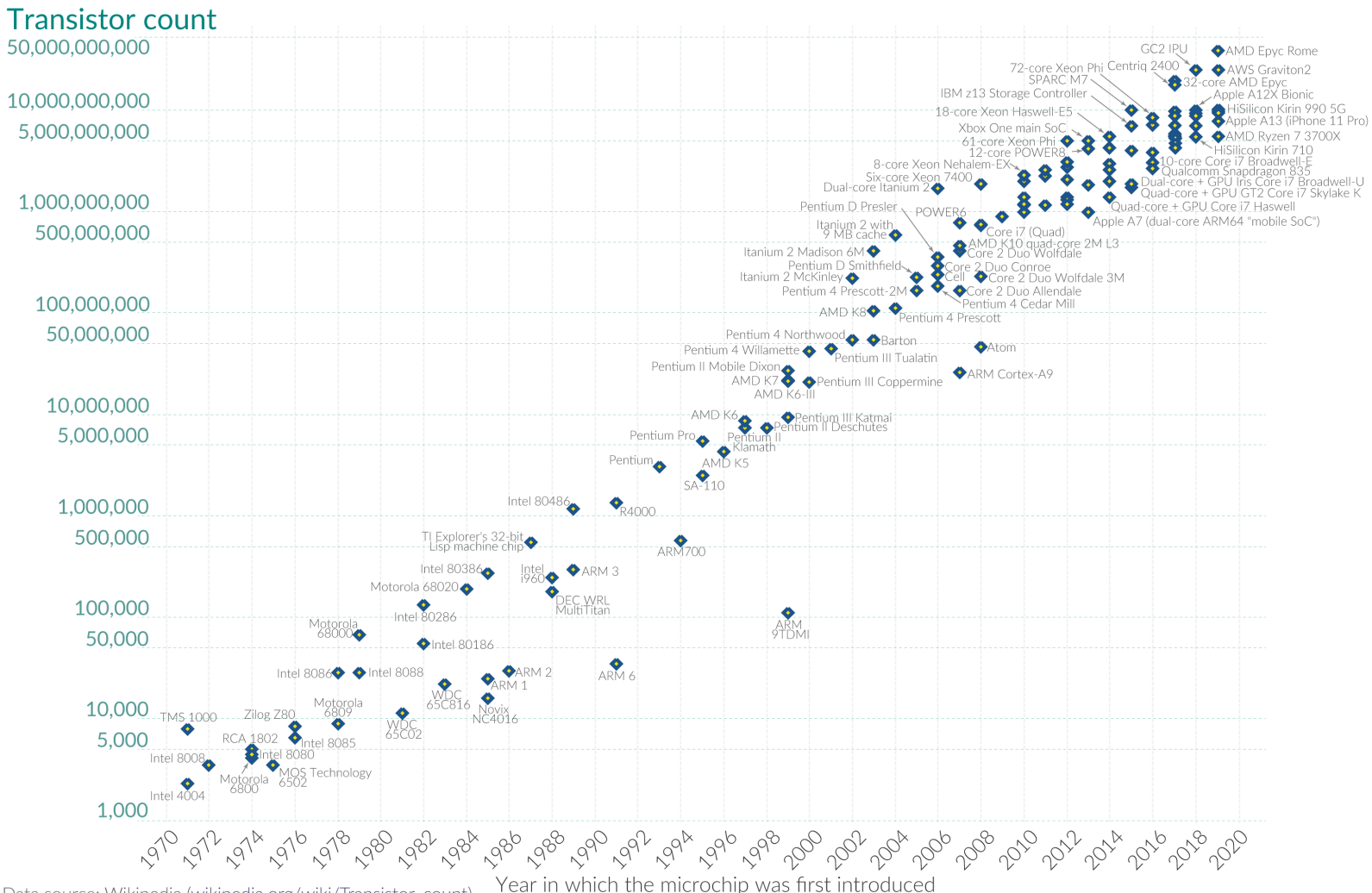
1986 Illustration of L.F. Richardson's vision by S. Conlin

Computing was taken over by machines

Moore's Law: The number of transistors on microchips doubles every two years

Our World in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years.
This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.



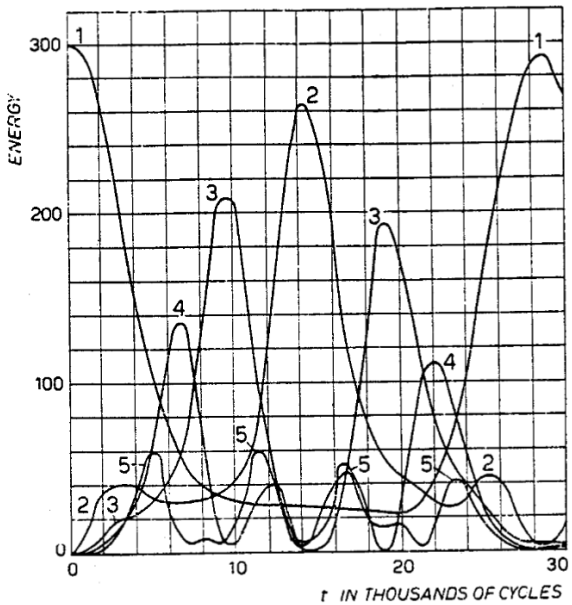
Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)
OurWorldinData.org – Research and data to make progress against the world's largest problems.

Computational engineering

- Starting points: artillery trajectories, nuclear weapons, rocket design, weather ...
- Now ubiquitous:
  - Structural engineering
  - Car industry
  - Oil recovery
  - Semiconductor design
  - ...
- Use of well established, verified, well supported commercial codes
  - Comsol
  - ANSYS
  - Eclipse
  - ...

As soon as computing machines became available . . .

. . . Scientists "misused" them to satisfy their curiosity



266.

STUDIES OF NON LINEAR PROBLEMS

E. FERMI, J. PASTA, and S. ULAM
Document LA-1940 (May 1955).

ABSTRACT.

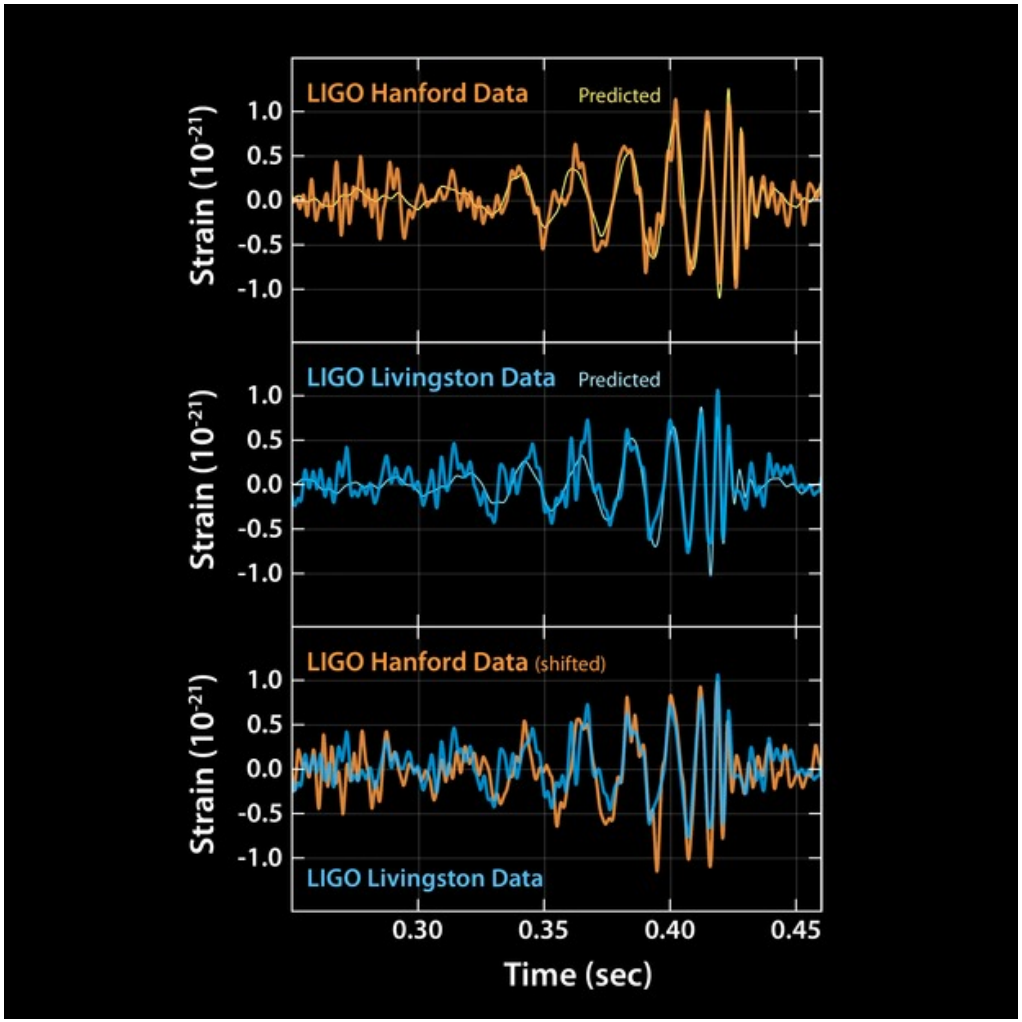
A one-dimensional dynamical system of 64 particles with forces between neighbors containing nonlinear terms has been studied on the Los Alamos computer MANIAC I. The nonlinear terms considered are quadratic, cubic, and broken linear types. The results are analyzed into Fourier components and plotted as a function of time.

". . . Fermi became interested in the development and potentialities of the electronic computing machines. He held many discussions [. . .] of the kind of future problems which could be studied through the use of such machines."

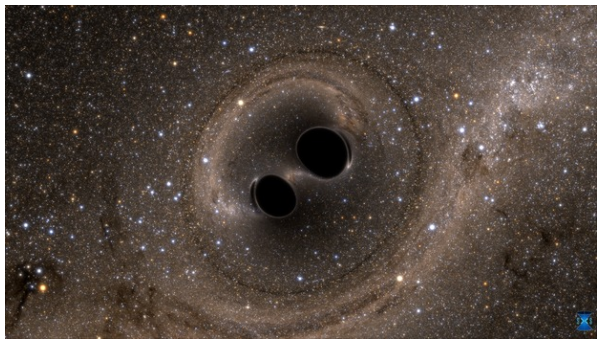
Fermi,Pasta and Ulam studied particle systems with *nonlinear* interactions

Calculations were done on the MANIAC-1 computer at Los Alamos

And they still do. . .



Caltech/MIT/LIGO Lab



SXS, the Simulating eXtreme Spacetimes (SXS) project (http://www.black-holes.org)

Verification of the detection of gravitational waves by numerical solution of Einstein's equations of general relativity using the "Spectral Einstein Code"

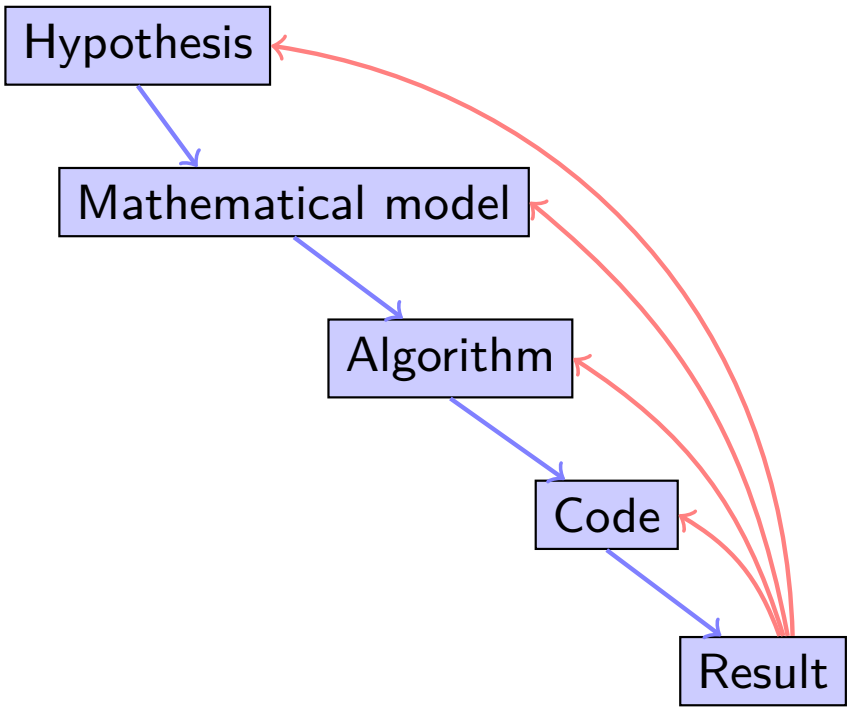
Computations significantly contributed to the 2017 Nobel prize in physics

Scientific computing

"The purpose of computing is insight, not numbers."
(https://en.wikiquote.org/wiki/Richard_Hamming)

- Frontiers of Scientific Computing
  - Insight into complicated phenomena not accessible by other methods
  - Improvement of models to better fit reality
  - Improvment of computational methods
  - Generate testable hypotheses to be verified by experiments
  - Support experimentation in other scientific fields
  - Exploration of new computing capabilities
  - Prediction, optimization of complex systems
- Good scientifc practice
  - Reproducibility
  - Sharing of ideas and knowledge
- Interdisciplinarity
  - Numerical Analysis
  - Computer science
  - Modeling in specific fields

General approach



- Possible (probable) involvement of different persons, institutions
- It is important to keep interdisciplinarity in mind

Scientific computing tools

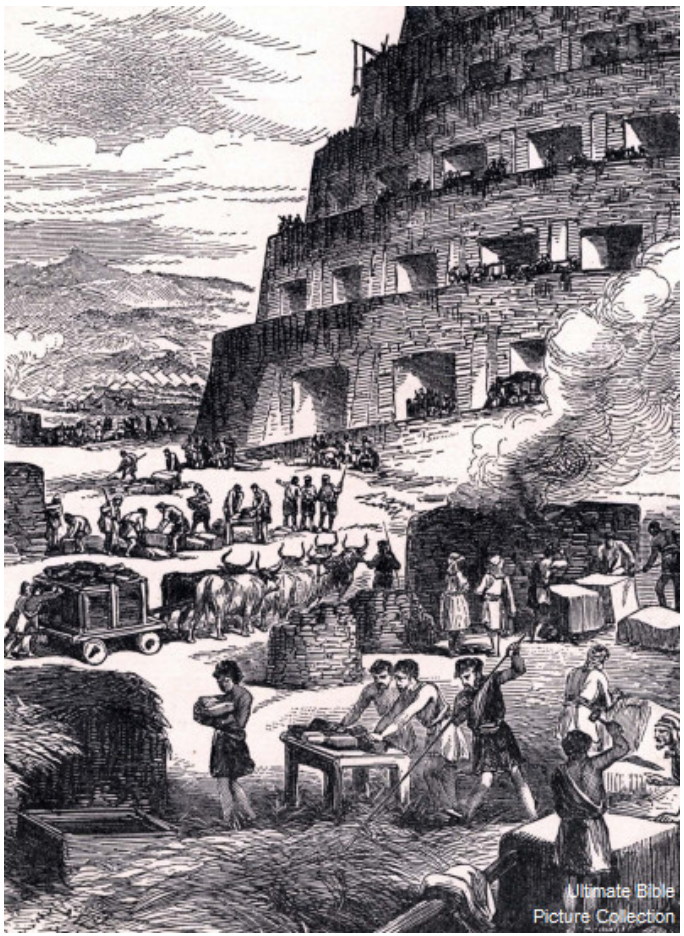
Many of them are Open Source

- General purpose environments
  - Matlab, Mathematica
  - COMSOL
  - Python + ecosystem
  - R + ecosystem
  - **Julia**
- "Classical" computer languages + compilers
  - Fortran, C, C++
- Established special purpose libraries
  - Linear algebra: **LAPACK**, **BLAS**, **UMFPACK**, Pardiso, SuperLU
  - Mesh generation: **triangle**, TetGen, NetGen, Gmsh
  - Eigenvalue problems: ARPACK
  - Visualization libraries: **VTK**
- Tools in the "background"
  - Build systems Make, CMake
  - Editors + IDEs: emacs, jedit, eclipse, atom, Visual Studio Code ...
  - Debuggers
  - Version control: Subversion (svn), **git**, Mercurial (hg) ...

Tools writtend in **bold face** will be used in this lecture

Confusio Linguarum

From Christian Mythology: trying to build the tower of Babel



"And the whole land was of one language and of one speech. ... And they said, Go to, let us build us a city and a tower whose top may reach unto heaven. ... And the Lord said, behold, the people is one, and they have all one language. ... Go to, let us go down, and there **confound their language that they may not understand one another's speech**. So the Lord scattered them abroad from thence upon the face of all the earth." (Daniel 1:1-7)

Same appears to be happening with computer languages ...

Intended aims and topics of this course

- Introduction to Julia as fresh approach to combine efficient computation with easy composability
- Focus on partial differential equation (PDE) solution
  - Solution of large linear systems of equations
  - Finite element and finite volume methods
  - Mesh generation
  - Nonlinear solvers
  - Automatic differentiation
  - Aspects of parallelization,Visualization
- Elements of Scientific Computing not covered:
  - Stochastic methods
  - Machine learning