Felix Koehler

f.<SURNAME>@tum.de   github.com/ceyron   linkedin.com/in/felixkoehler   x.com/felix_m_koehler

- My research focuses on machine learning (in particular deep learning) for physical simulations (like fluid and solid mechanics), adjoint methods and automatic differentiation, inverse problems as well as optimization algorithms
- My recent paper *APEBench* (a PDE surrogate benchmark suite in JAX) was accepted at NeurIPS 2024
- I own the largest YouTube channel (@MachineLearningSimulation) for in-depth tutorials in Scientific Machine Learning (>24k subs)

EXPERIENCE

PhD Student

Technical University of Munich, supervised by Nils Thuerey   September 2022 - Present (anticipated graduation: 12/26)

- Published at Neurips 2024: *APEBench: A Benchmark for Autoregressive Neural Emulators of PDEs* (Arxiv, Code, Project Page)
- Coordinated annual lecture in GamePhysics (applied numerical methods + simple mechanical modeling) for >200 students
- Supervised multiple students, e.g., Unrolled vs. Implicit Autodiff for Linear System Solves by Kanishk Bhatia
- Instructed a lecture on *Autodiff and Adjoints in Differentiable Physics* for our *Physics-Based DL* master course (Recording, Slides)

Research Student

Siemens Corporate Technology, supervised by Dirk Hartmann   March 2020 - April 2022

- Researched explainable & efficient explicit approximations to model predictive control strategies with patent filed (US20240176310)

Research Intern

Volkswagen   October 2018 - March 2019

- Extended Volkswagen's topology optimization suite with adjoint-based sensitivity analysis, resulting in up to 40% additional weight reduction for the investigated casted steering knuckles; results presented at 13th WCSMO (Paper)

Additional Positions

- Research Engineer @ TUM Hyperloop in electromagnetic simulations with ANSYS Maxwell and Multidisciplinary Optimization, where I developed surrogate models for electromagnetic levitation/propulsion and coordinated access to LRZ supercomputing resources
- Multiple Teaching Assistant positions: ODEs , PDEs , Vector Calculus , Linear Algebra , Thermodynamics

SKILLS

- Spoken languages: English (professional, IELTS 8.5/9.0), German (native), French & Swedish (elementary)
- Python : JAX , Tensorflow 2 , PyTorch , Matplotlib , Pandas , Seaborn , Scikit-Learn , SciPy , NumPy , FEniCS
- C++ : OpenMP , MPI , Eigen
- Other programing languages: Julia , bash , FORTRAN
- Tools/Platforms: Linux (Ubuntu & Arch), Git , Docker , OpenFoam , Paraview , ANSYS Maxwell , Siemens Amesim

EDUCATION

M.Sc. Computational Science & Engineering (CSE)

Technical University of Munich, **GPA: 1.2** *(German system: 1.0 (best) to 6.0 (worst))*, best 2%   October 2019 - April 2022

- Relevant course work: Numerical Analysis (2 semesters), Parallel Programming , High-Performance Computing , Parallel Numerics , Scientific Computing , ML for Graphs & Sequential Data , Uncertainty Quantification , Machine Learning , Probabilistic Machine Learning , Visualization , Nonlinear Finite Element Method , Computational Plasticity
- Honors: Scholarship by *Studienstiftung des Deutschen Volkes* (most prestigious national scholarship program in Germany)

B.Sc. Mechanical Engineering

Technical University of Braunschweig, **GPA: 1.1** *(German system: 1.0 (best) to 6.0 (worst))*   October 2015 - January 2019

PROJECTS & OPEN SOURCE

- Machine Learning & Simulation YouTube channel: I created more than 200 videos on AI for Science and adjacent topics (autodiff , optimization , probability , neural networks etc.). Examples: FNO , DeepONet , Adjoint ODEs , Variational Inference
- Exponax: I designed and maintain a JAX-based Fourier-spectral PDE solvers library based on ETDRK methods (Docs)
- PDEquinox: I designed and maintain a JAX/Equinox-based library for PDE neural operators like ConvNets , ResNets , MLPs , UNets , Dilated ResNets (Docs)
- Course on Scientific Computing in Python: I devised and taught a graduate-level course three times to 25 students each (Resources, Recording Day 1, Recording Day 2, Recording Day 3)
- I translated a Shallow Water Equations solver to Julia (*Tsunamis.jl* GitHub Repository, Report paper)
- I implemented Stable Fluids (for solving the Navier-Stokes Equations) in Julia (*StableFluids.jl* GitHub Repository)