# Research Landscape

Collection of links to browse

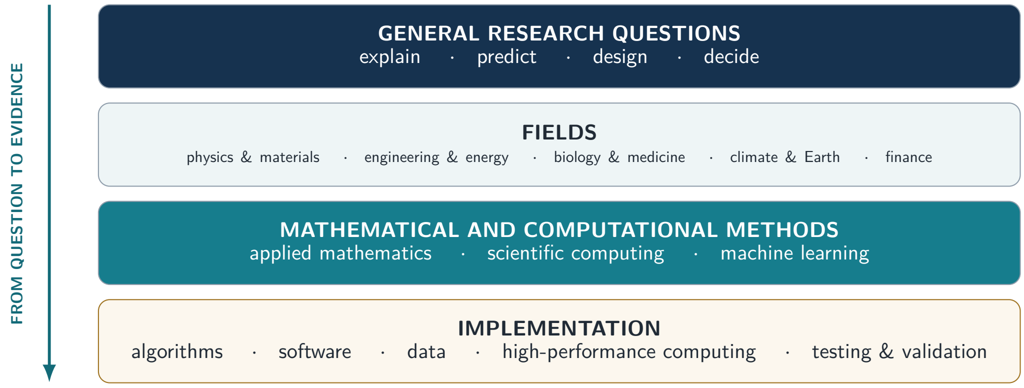
A research and career guide for undergraduates in
Applied mathematics, scientific computing, and AI

26 August 2026

First edition. Links and affiliations were checked against primary sources on 26 August 2026. Research organizations change; verify current openings and group structures before applying.

# Introduction

Research is often introduced through its visible subject: a new material, a changing climate, a biological system, or an engineered structure. Behind these subjects is a shared technical backend. Applied mathematics formulates models and precise questions; scientific computing turns them into reliable calculations; and machine learning extracts structure from data or augments models when direct simulation is too costly.




## Perspective

The three areas are distinct but strongest when combined. Mathematics supplies structure, assumptions, and guarantees; scientific computing brings scale, stability, and reproducibility; machine learning adds flexible inference, surrogates, and discovery from data. Together they support hybrid research in which equations constrain learned models, simulations create data, optimization connects parameters to observations, and uncertainty analysis tests conclusions. This foundation can lead toward computational science, numerical analysis, scientific machine learning, domain research, research software, or industrial R&D.

**Questions this guide will help you answer**

- Which research problems do I care enough about to study deeply?
- What mathematics turns those problems into models I can analyze?
- Which numerical methods, software, and computing systems make the models usable?
- Where can machine learning add value, and where is a classical method more reliable?
- Which researchers, laboratories, and companies work at these intersections?
- What project would demonstrate that I can formulate, implement, and validate an idea?
- Which next step—coursework, research, graduate study, or industry—fits my evidence and goals?

## How to use this guide

Use the guide as a funnel: begin broadly, then narrow your attention as your curiosity becomes more specific.

1. **Do not read it as a ranking.** The order of fields, researchers, laboratories, and companies does not indicate importance. Treat every entry as a possible starting point.
2. **Browse to see the big picture.** Scan the research areas, methods, people, career paths, and projects. Notice connections before trying to choose a specialization.
3. **When something interests you, dig deeper.** Follow its links, learn the required ideas, read an accessible paper or lecture, and reproduce one small result. Let evidence replace a first impression.
4. **Reach out to relevant people.** Contact researchers, students, engineers, or mentors whose work closely matches your interest. Ask a specific question and show what you have already read, derived, or built.
5. **Search for opportunities.** Look for research projects, internships, open-source work, summer programs, graduate study, and technical roles where you can apply the direction you discovered. Verify requirements and deadlines at the official source.

## About me

**Yerbol Palzhanov**

I am an Assistant Professor at SDU University and a researcher in applied mathematics. My interests include numerical analysis, finite element methods, scientific computing, and computational problems arising in fluid dynamics and biomolecular systems.

I've compiled my perspective into this guide to help undergraduates see how mathematical ideas, computational methods, research communities, and career opportunities connect. Learn more about my research, publications, and teaching at palzhanov.github.io.

**Continue exploring**

After exploring this guide, you may want to look more deeply into a topic or feel motivated to join one of these communities.

1. For deeper exploration, I plan to publish accessible overviews of research work on my YouTube channel. I may also create a separate document that explains foundational ideas in greater depth while remaining suitable for casual reading.

2. If you want to join these communities, begin by connecting with more people and building solid foundational knowledge. With sustained effort, these paths are entirely achievable. If I create a dedicated guide on getting started, I will link it here as well.

This guide is hosted on GitHub at github.com/palzhanov/research_landscape, where you can find its source files and future updates.

# Contents

# Part I

# Academia

# How computational research is organized

Academic fields overlap. Numerical analysis asks whether an algorithm is stable and convergent; scientific computing asks how to make it useful at scale; a domain scientist asks whether the model captures reality. Scientific machine learning adds learned components, but it does not remove these earlier questions. This part is therefore organized by research problem rather than university.

This chapter presents possible research directions and highlights prominent professors whose work provides a broad view of recent trends in each field. The selection is intended as a starting point for exploration rather than a complete survey.

# Chapter 1

# Numerical analysis and scientific computing

Numerical analysis studies how continuous mathematics becomes reliable finite computation. If $Ax = b$ is solved on a computer, one asks not only whether a routine returned a vector, but how conditioning, roundoff, discretization, and stopping criteria affect the answer. Scientific computing combines that analysis with data structures, software design, and hardware.

Core ideas include approximation and interpolation; consistency, stability, and convergence; conditioning and backward error; direct and iterative linear solvers; eigenvalue and singular-value problems; sparse matrices; preconditioning; and parallel algorithms. A fundamental distinction is

$$\underbrace{\|u - u_h\|}_{\text{total error}} \leq \underbrace{\|u - \Pi_h u\|}_{\text{approximation}} + \underbrace{\|\Pi_h u - u_h\|}_{\text{discrete/solver effects}} .$$

Typical research asks how to solve a billion-variable sparse system, design a robust preconditioner, certify an approximation, exploit GPUs without losing numerical reliability, or reduce a high-dimensional simulation while retaining its governing structure.

## 1.1 J. Nathan Kutz

*University of Washington. Boeing Professor in AI & Data-Driven Engineering; appointments in electrical and computer engineering and applied mathematics.*

Kutz connects scientific computing, dynamical systems, numerical linear algebra, reduced-order modeling, and machine learning. His work and textbooks are unusually effective bridges from matrices and differential equations to data-driven models. Undergraduates should begin with his applied linear algebra and data-driven science materials, then reproduce a DMD or SINDy example. **Links:** faculty page; research group, books, and course materials.

## 1.2 Yurii Nesterov

*The Chinese University of Hong Kong, Shenzhen. Professor in the School of Data Science; recipient of the 2026 Gauss Prize.*

Nesterov developed foundational complexity theory and accelerated first-order methods for convex optimization, ideas now central to large-scale scientific computing and machine learning. Compare gradient descent with Nesterov acceleration on a smooth convex problem, then study why their convergence rates differ. **Links:** faculty profile; 2026 Gauss Prize announcement.

## 1.3 Yousef Saad

*University of Minnesota. CSE Distinguished Professor and William Norris Land Grant Chair in Large-Scale Computing.*

Saad is a central figure in sparse matrix computation, Krylov-subspace methods, eigenvalue problems, preconditioning, and parallel numerical linear algebra. His books connect mathematical analysis to algorithms that remain practical for very large systems, while his software provides concrete implementations to inspect and test. **Links:** faculty profile; books, software, and research materials.

## 1.4 James Demmel

*University of California, Berkeley. Professor Emeritus and Professor in the Graduate School in mathematics and computer science.*

Demmel's work joins numerical accuracy with high-performance implementation. His contributions to LAPACK and ScaLAPACK, communication-avoiding algorithms, and error analysis show why matrix algorithms must be designed together with floating-point arithmetic, memory hierarchies, and parallel hardware. **Links:** Wikipedia; personal and research page; Matrix Computations course materials.

## 1.5 Jack Dongarra

*University of Tennessee, Knoxville. University Distinguished Professor Emeritus; recipient of the 2021 ACM A. M. Turing Award.*

Dongarra helped create the numerical-software infrastructure underlying modern scientific computing, including major contributions to LINPACK, BLAS, LAPACK, ScaLAPACK, and performance benchmarks. His career illustrates how algorithms, portable software standards, and changing computer architectures shape one another. **Links:** faculty profile; Innovative Computing Laboratory.

## 1.6 Nick Higham (1961–2024)

*University of Manchester. A major reference for numerical linear algebra, matrix functions, stability, and floating-point computation.*

Higham's books and expository writing remain excellent training in the habit of asking what a computed answer means. Start with his explanations of condition numbers and backward stability, then test textbook algorithms on deliberately ill-conditioned matrices. **Links:** archive and articles; software.

**Search terms:** *numerical stability, backward error, Krylov subspace, preconditioning, sparse direct solver, randomized linear algebra, mixed precision.*

Chapter 2

# Finite-element methods and numerical PDEs

Partial differential equations (PDEs) describe fields that vary in space and time. For the Poisson problem

$$-\nabla \cdot (k\nabla u) = f \quad \text{in } \Omega,$$

finite-element analysis multiplies by a test function $v$, integrates, and uses integration by parts. With suitable boundary conditions the weak form is: find $u \in V$ such that

$$\int_\Omega k\nabla u \cdot \nabla v \, dx = \int_\Omega f v \, dx \quad \text{for every } v \in V.$$

Approximating $V$ by basis functions on a mesh yields a sparse system. The field matters because weak formulations and finite elements handle complex geometry, variable materials, and coupled physics while supporting rigorous error analysis. Core concepts are Sobolev spaces, Galerkin orthogonality, approximation estimates, mesh quality, adaptivity, nonlinear iteration, and scalable solvers.

PDE → weak form → mesh → finite elements → sparse system → solver

Typical research problems include constructing stable elements, proving convergence, estimating error after a computation, adapting a mesh, representing moving interfaces, and preconditioning the resulting algebraic systems. Common tools include FEniCSx, deal.II, MFEM, Firedrake, PETSc, Gmsh, C++, Python, and MPI.

## 2.1 Maxim Olshanskii

*University of Houston. Professor of mathematics; finite elements, surface and evolving-domain PDEs, fluid dynamics, model reduction, and numerical linear algebra.*

Olshanskii works across analysis and computation, including Navier–Stokes solvers, cardiovascular mathematics, and geometrically unfitted methods. **TraceFEM** can be understood as follows: embed a curved surface in a simple three-dimensional background mesh, use ordinary bulk finite-element functions, and take their traces on the surface. The mesh need not align with the surface, which is attractive when the surface moves or changes shape; analysis and stabilization ensure the resulting system remains meaningful. **Links:** personal page and publications; publication list.

## 2.2 Wolfgang Bangerth

*Colorado State University. Finite elements, adaptivity, inverse problems, and large-scale scientific software.*

Bangerth is a principal force behind `deal.II`. Students can learn how mathematical abstractions become well-tested C++ interfaces and parallel adaptive computations by following its tutorials. His work is particularly useful for understanding that research software architecture is part of computational mathematics rather than an afterthought. **Links:** faculty page; deal.II.

## 2.3 Jesse Chan

*Rice University. High-order finite-element and discontinuous Galerkin methods, stable wave and fluid discretizations, and GPU computing.*

Chan develops provably stable high-order methods for PDEs together with efficient implementations on many-core and GPU architectures. His work is a strong entry point for students who want to connect approximation theory, conservation and stability, and performance engineering. Begin with a one-dimensional discontinuous Galerkin method, then study how numerical fluxes and polynomial order affect stability and accuracy. **Links:** personal research page; Rice profile.

## 2.4 Béatrice Rivière

*Rice University. Noah Harding Chair and Professor; discontinuous Galerkin methods, porous-media flow, fluid mechanics, and iterative solvers.*

Rivière is known for the theory and implementation of discontinuous Galerkin methods and for numerical models of multiphase flow in porous and deformable media. Her work shows how analysis, high-order discretization, and applications in energy and biomedicine reinforce one another. Her book on discontinuous Galerkin methods is a useful advanced entry after a first FEM course. **Links:** Rice profile; COMP-M group.

## 2.5 Douglas N. Arnold

*University of Minnesota. McKnight Presidential Professor of Mathematics; mixed finite elements, numerical PDEs, mechanics, and finite element exterior calculus.*

Arnold helped develop finite element exterior calculus, a framework that connects geometry, topology, differential complexes, and stable discretization. His work explains why compatible finite-element spaces preserve essential structure in problems such as electromagnetics, elasticity, and incompressible flow. The periodic table of finite elements and his FEEC materials provide especially useful maps of the field. **Links:** personal page and publications; finite element exterior calculus materials.

## 2.6 Patrick E. Farrell

*University of Oxford. Professor of numerical analysis; adaptive PDE discretization, nonlinear solvers, bifurcation, and PDE-constrained optimization.*

Farrell develops numerical methods and software for difficult nonlinear PDE problems, including techniques that compute multiple solution branches rather than only the state reached from one initial guess. His research connects finite-element discretization with solver design, automatic differentiation, optimization, and reproducible scientific software, offering a view of current computational PDE research. **Links:** Oxford profile, publications, and awards; finite-element lecture materials.

## 2.7 L. Ridgway Scott

*University of Chicago. Finite-element analysis, scientific parallel computing, automated modeling, and the FEniCS project.*

Scott has made foundational contributions to finite-element approximation and has written influential texts on FEM and scientific parallel computing. His work on automated modeling with FEniCS is especially relevant to students learning how mathematical formulations can be translated into reusable software while preserving analytical meaning. **Links:** personal page, books, and resources.

**Search terms:** *weak formulation, Galerkin method, Sobolev space, a posteriori error, adaptive mesh, unfitted FEM, TraceFEM, preconditioning.*

## Chapter 3

# Computational fluid dynamics

Computational fluid dynamics (CFD) turns conservation laws into numerical predictions of fluid motion. For incompressible Newtonian flow,

$$\rho(\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u}) - \mu \Delta \mathbf{u} + \nabla p = \mathbf{f}, \qquad \nabla \cdot \mathbf{u} = 0.$$

The pressure constraint, nonlinearity, boundary layers, turbulence, and broad range of scales make the field difficult. The research chain is

$$\text{Navier–Stokes} \to \text{discretization} \to \text{linear/nonlinear solvers} \to \text{verification} \to \text{simulation}.$$

Major directions include incompressible flow, turbulence modeling, stable mixed methods, finite volumes, high-order spectral elements, fluid–structure interaction, multiphase flow, reduced-order modeling, adaptive meshes, and high-performance implementations. Common tools range from FEniCSx and OpenFOAM to PETSc, MPI, visualization systems, and domain-specific research codes.

### 3.1 Annalisa Quaini

*University of Houston. CFD, cardiovascular flow, finite-volume and finite-element methods, multiphysics, and data-driven ROMs.*

Quaini connects high-fidelity simulation with faster reduced models. A useful entry question is how POD modes and learned corrections behave when the flow regime changes. **Links:** profile; research projects; publications.

### 3.2 Parviz Moin

*Stanford University. Franklin P. and Caroline M. Johnson Professor; turbulence, large-eddy simulation, flow physics, and high-performance CFD.*

Moin pioneered direct and large-eddy simulation of turbulent flows and helped establish computational fluid mechanics as a predictive scientific discipline. His work is a strong guide to the interaction among physical modeling, numerical resolution, and the computing required to represent turbulent scales. **Links:** Stanford profile; Center for Turbulence Research.

## 3.3 Petros Koumoutsakos

*Harvard University. Herbert S. Winokur, Jr. Professor of Computing in Science and Engineering; multiscale flow simulation, particle methods, active matter, and scientific machine learning.*

Koumoutsakos develops computational methods that connect fluid mechanics with biological systems, optimization, and machine learning. His research is especially useful for seeing how new algorithms are tested on difficult multiscale phenomena rather than only on standard benchmark flows. **Links:** Harvard profile and research.

## 3.4 Tim Colonius

*California Institute of Technology. Frank and Ora Lee Marble Professor; unsteady flows, aeroacoustics, cavitation, flow instability, and reduced-order modeling.*

Colonius studies how coherent structures, instability, sound generation, and complex interfaces can be represented and predicted computationally. His group connects high-fidelity simulation with analysis and reduced models, making it a useful bridge from classical CFD to modern data-driven dynamics. **Links:** Caltech profile; research group.

## 3.5 Sanjiva Lele

*Stanford University. Professor of Aeronautics and Astronautics and of Mechanical Engineering; compressible turbulence, aeroacoustics, multiphysics, and high-fidelity simulation.*

Lele develops high-order numerical methods and physical models for turbulent and compressible flows, including problems involving shocks, combustion, sound, and high-speed vehicles. His work illustrates how discretization quality and physical fidelity must advance together in demanding CFD applications. **Links:** Stanford profile and publications.

**Search terms:** *Navier–Stokes, Reynolds number, turbulence closure, pressure–velocity coupling, inf–sup condition, fluid–structure interaction, multiphase flow, high-order method, flow control.*

Chapter 4

# Scientific machine learning

Scientific machine learning (SciML) combines scientific structure, numerical methods, and learning. The goal is not merely to fit data, but to make useful predictions about systems governed by geometry, conservation laws, differential equations, and uncertainty. A useful taxonomy is more precise than calling every regression model "physics informed."

**PINNs** place differential-equation residuals, data mismatch, and boundary conditions in a training objective. They are flexible, especially for inverse problems, but optimization can be difficult and a conventional solver is often the correct baseline.

**Neural operators** learn maps between functions, such as coefficient or initial-condition fields and PDE solution fields. DeepONet uses branch and trunk networks; the Fourier Neural Operator performs learned transformations in spectral space.

**Surrogates** approximate an expensive input–output map in a specified parameter range. Their speed is meaningful only with held-out accuracy, physical checks, and out-of-distribution tests.

**Differentiable simulation** exposes derivatives through numerical steps, enabling parameter estimation, design, and control.

**Hybrid models** keep trusted physics and learn only a closure, correction, constitutive relation, or unresolved component.

The standard workflow is

$$\text{physics} \to \text{PDE} \to \text{simulation} \to \text{data} \to \text{learned surrogate} \to \text{verification}.$$

Core foundations are PDEs, approximation, numerical linear algebra, probability, optimization, automatic differentiation, and scientific validation. Common tools include PyTorch, JAX, Julia SciML, DeepXDE, NeuralOperator, FEniCSx, and conventional domain solvers.

## 4.1 George Em Karniadakis

*Brown University. Professor of applied mathematics and leader of the CRUNCH group.*

Karniadakis works on high-order methods, stochastic systems, fluid mechanics, computational biology, uncertainty, PINNs, and operator learning. The CRUNCH group identifies itself as the home of the original PINN and DeepONet work. For an undergraduate, the right entry is not to copy a PINN notebook: first solve a small PDE conventionally, then compare accuracy, computational cost, boundary error, and sensitivity to training choices. **Links:** CRUNCH group; Brown profile; DeepXDE.

## 4.2 Rebecca Willett

*University of Chicago. Mathematical foundations of machine learning, inverse problems, signal processing, and scientific ML.*

Willett studies trustworthy learning in data-limited scientific settings, including climate, imaging, materials, astronomy, and data assimilation. Her work is a useful entry to the statistical side of SciML: physical structure can reduce data requirements, but uncertainty, identifiability, and reproducibility remain central. **Links:** research site; university profile.

## 4.3 Anima Anandkumar

*California Institute of Technology. Bren Professor of Computing and Mathematical Sciences; neural operators, tensor methods, and AI for scientific simulation.*

Anandkumar develops learning methods for physical systems, including neural operators that approximate mappings between functions rather than predictions at one fixed discretization. Her work offers an entry to large-scale AI-for-science research while keeping generalization, physical structure, and computational performance in view. **Links:** Caltech profile; neural-operator research.

## 4.4 Paris Perdikaris

*University of Pennsylvania. Associate Professor; physics-informed learning, operator learning, generative models, uncertainty quantification, and physical simulation.*

Perdikaris studies how physical priors and observations can be combined in learned models for forward and inverse problems. His work is useful for comparing PINNs, neural operators, and scientific foundation models while asking where uncertainty and limited data constrain their reliability. **Links:** Penn Applied Mathematics and Computational Science profile.

## 4.5 Rose Yu

*University of California San Diego. Professor of Computer Science and Engineering; spatiotemporal machine learning, dynamical systems, and physics-guided AI.*

Yu develops structure-aware learning methods for large spatiotemporal systems, with applications including climate, transportation, and physical forecasting. Her research is a good route into the question of how symmetries, dynamics, and domain knowledge improve prediction beyond generic sequence models. **Links:** UC San Diego profile.

**Search terms:** *physics-informed neural network, DeepONet, Fourier neural operator, differentiable physics, hybrid model, surrogate, inverse problem, scientific validation.*

# Chapter 5

# Reduced-order modeling and data-driven dynamics

High-fidelity simulations may contain millions of unknowns even when their dominant behavior occupies a much smaller set of coordinates. Reduced-order modeling (ROM) seeks compact models that retain the important input–output or dynamical behavior. Data-driven dynamics seeks governing structure directly from observations.

Suppose simulation snapshots form $X = [x_1, \ldots, x_m]$. A truncated SVD

$$X \approx U_r \Sigma_r V_r^\top$$

defines a Proper Orthogonal Decomposition (POD) basis $U_r$. Projecting the governing equations onto that basis can replace millions of unknowns with $r$ coordinates. DMD fits approximately $X' \approx AX$ and studies spectral modes of the time-advance map. SINDy fits $\dot{X} \approx \Theta(X)\Xi$ with a sparse coefficient matrix $\Xi$. Koopman methods seek a linear evolution rule in a lifted observable space.

Applications include aerodynamic design, flow control, climate and ocean models, biological dynamics, and real-time digital twins. A reduced model is not automatically reliable: inspect truncation energy, conservation, long-horizon stability, parameter coverage, and speedup including offline data generation. Common tools are NumPy/SciPy, PyDMD, PySINDy, FEniCSx, MATLAB, and domain simulation codes.

## 5.1 Steven L. Brunton

*University of Washington. DMD, SINDy, Koopman methods, reduced-order control, and fluid dynamics.*

Brunton is a central entry point for connecting dynamical-systems theory with practical algorithms and code. Follow his work to understand what each decomposition assumes and how sampling, noise, transients, and control inputs change the result. His *Eigensteve* channel is especially useful because it develops intuition visually while preserving derivations and code connections. **Links:** research and teaching; Eigensteve; UW profile.

## 5.2 Karen Willcox

*University of Texas at Austin. Projection-based ROM, operator inference, multifidelity methods, and predictive digital twins.*

Willcox shows how a reduced model becomes useful in design and decisions rather than remaining only a compression exercise. Her work emphasizes limited expensive data, uncertainty, and reliable updating. A predictive twin combines models and data, updates its state, represents uncertainty, and serves a decision. **Links:** profile; research group.

## 5.3 Anthony T. Patera

*Massachusetts Institute of Technology. Ford Foundation Professor of Engineering; certified reduced-basis methods, parametrized PDEs, and computational mechanics.*

Patera helped establish reduced-basis methods with rigorous a posteriori error estimates and efficient offline–online computation. His work shows how a reduced model can deliver both speed and a quantitative statement about reliability over a parameter domain. **Links:** MIT profile; research group.

## 5.4 Jan S. Hesthaven

*EPFL. Professor Emeritus; high-order numerical methods, certified reduced-basis methods, uncertainty quantification, and scientific machine learning.*

Hesthaven connects high-order PDE discretization with projection-based reduction and newer nonintrusive learned models. His work provides a broad view of how approximation theory, stability, certification, and data-driven methods interact in reduced modeling. **Links:** EPFL profile; MCSS laboratory.

## 5.5 Gianluigi Rozza

*SISSA. Professor of Numerical Analysis; reduced-order modeling, parametrized PDEs, CFD, shape optimization, and optimal control.*

Rozza develops reduced models for complex geometries and parameter-dependent problems in engineering and medicine. His research is a useful entry to the full ROM pipeline: geometry and discretization, basis construction, hyper-reduction, bifurcation or control, and reusable open-source software. **Links:** personal page; SISSA mathLab.

**Search terms:** *POD, SVD, DMD, Koopman operator, SINDy, system identification, Galerkin projection, operator inference, hyper-reduction, digital twin.*

Chapter 6

# Optimization, machine learning, and mathematical AI

Optimization asks for $x$ minimizing an objective $f(x)$ under constraints. Convexity can make local and global optima coincide; nonconvex learning problems do not offer that guarantee. Core tools include gradients and Hessians, Lagrange multipliers and duality, proximal methods, stochastic gradients, constrained optimization, automatic differentiation, and large-scale distributed methods.

The field matters because robust machine learning, reinforcement learning, inverse problems, optimal control, experimental design, and PDE-constrained engineering all require optimization. Typical research asks how to exploit structure, scale an algorithm, handle noisy gradients, certify a result, or optimize while satisfying physical or safety constraints. Common tools include Python, NumPy/SciPy, CVXPY, PyTorch, JAX, Julia/JuMP, and specialized large-scale solvers.

## 6.1 Stephen Boyd

*Stanford University. Convex optimization, control, distributed algorithms, and accessible mathematical education.*

Boyd's freely available convex optimization text and lectures are an excellent structured entry. Students should learn to recognize convex problems, derive optimality conditions, and use a modeling package without confusing a solver call with mathematical understanding. **Links:** homepage and courses; CVXPY.

## 6.2 Arkadi Nemirovski

*Georgia Institute of Technology. John P. Hunter, Jr. Chair and Professor; convex optimization, complexity theory, robust optimization, and large-scale algorithms.*

Nemirovski has shaped modern continuous optimization through interior-point methods, first-order methods, and algorithms for robust and nonsmooth problems. His work is an entry to a central research question: what accuracy can an algorithm guarantee with a limited number of expensive computations? **Links:** Georgia Tech profile.

## 6.3 Lieven Vandenberghe

*University of California, Los Angeles. Professor of Electrical and Computer Engineering; convex, conic, and semidefinite optimization with applications in control and signal processing.*

Vandenberghe develops optimization methods that expose and exploit mathematical structure. His work and teaching show how duality, sparsity, and cone formulations turn seemingly different engineering problems into models that can be analyzed and solved systematically. **Links:** UCLA profile; homepage and teaching materials.

## 6.4 Benjamin Recht

*University of California, Berkeley. Professor of Electrical Engineering and Computer Sciences; optimization, machine learning, control, and empirical evaluation.*

Recht studies the mathematical and experimental foundations of learning and control, from optimization algorithms to the behavior of benchmarks and large models. His research encourages students to combine proofs and algorithms with careful tests of whether a claimed improvement survives realistic evaluation. **Links:** Berkeley profile; research essays.

## 6.5 Jorge Nocedal

*Northwestern University. Walter P. Murphy Professor; large-scale nonlinear and stochastic optimization, machine learning, and numerical software.*

Nocedal is a leading developer of quasi-Newton and limited-memory methods, including algorithms used throughout scientific computing and machine learning. His work is a strong route from derivations of search directions and line searches to robust software for genuinely large problems. **Links:** Northwestern profile; homepage and software.

**Search terms:** *convex duality, stochastic gradient, proximal algorithm, constrained optimization, nonconvex optimization, robust ML, reinforcement learning, PDE-constrained optimization.*

Chapter 7

# Uncertainty quantification and inverse problems

A forward model predicts observations $y = \mathcal{G}(\theta) + \eta$ from parameters $\theta$. An inverse problem infers $\theta$ from $y$; it may be non-unique, unstable, or expensive. In Bayesian form,

$$\pi(\theta \mid y) \propto \pi(y \mid \theta)\pi(\theta),$$

so the answer is a posterior distribution rather than one unexplained best fit. Uncertainty quantification (UQ) studies how uncertain inputs, model discrepancy, numerical error, and noisy data affect predictions. Data assimilation updates a dynamical state as observations arrive; experimental design asks what to measure next.

Applications include inferring underground properties from geophysical signals, estimating material or tissue parameters, medical imaging, calibrating climate models, tracking aerospace states, and updating digital twins. Core concepts are probability, Bayesian inference, regularization, sampling, surrogate models, sensitivity analysis, and identifiability. Common tools include probabilistic programming systems, SciPy, PETSc-based forward solvers, MCMC packages, and domain simulation software.

## 7.1 Youssef Marzouk

*MIT. Breene M. Kerr Professor in Aeronautics and Astronautics; Bayesian computation, inverse problems, data assimilation, and experimental design.*

Marzouk develops transport methods, UQ, data assimilation, experimental design, and machine learning for complex physical systems. His path includes Sandia National Laboratories, making his work a useful bridge between mathematical statistics, simulation, and mission-driven research. Undergraduates can start with a low-dimensional Bayesian parameter-estimation problem before attempting high-dimensional PDE inversion. **Links:** MIT profile; UQ group.

## 7.2 Andrew M. Stuart

*California Institute of Technology. Bren Professor of Computing and Mathematical Sciences; Bayesian inverse problems, data assimilation, and stochastic modeling.*

Stuart develops the mathematical foundations of inferring uncertain functions and parameters from incomplete, noisy data. His work is essential for understanding when a posterior distribution is well-defined and how algorithms connect finite computations to infinite-dimensional inverse problems. **Links:** Caltech profile and publications.

## 7.3 Omar Ghattas

*The University of Texas at Austin. Ernest and Virginia Cockrell Chair; large-scale inverse problems, uncertainty quantification, optimization, and high-performance computing.*

Ghattas develops scalable methods for Bayesian inversion and optimization constrained by complex PDEs. His work shows how adjoints, low-rank Hessian structure, randomized algorithms, and parallel computing make uncertainty analysis possible even when each forward simulation is expensive. **Links:** Oden Institute profile.

## 7.4 Roger Ghanem

*University of Southern California. Tryon Chair and Professor; stochastic mechanics, uncertainty propagation, and probabilistic computational models.*

Ghanem is a foundational figure in polynomial-chaos and stochastic finite-element methods. His research connects probabilistic representations with mechanics and simulation, while emphasizing the difficult distinction among uncertain inputs, incomplete models, and numerical approximation. **Links:** USC profile.

## 7.5 Raúl Tempone

*King Abdullah University of Science and Technology. Professor of Applied Mathematics and Computational Science; stochastic numerics, Bayesian inverse problems, and multilevel methods.*

Tempone develops adaptive and hierarchical algorithms for propagating and estimating uncertainty efficiently. His work is a useful entry to multilevel Monte Carlo and related methods, where many inexpensive coarse simulations are combined with fewer costly accurate ones. **Links:** KAUST profile.

**Search terms:** *Bayesian inverse problem, posterior, MCMC, transport map, regularization, identifiability, data assimilation, sensitivity analysis, optimal experimental design.*

## Chapter 8

# Computational mathematics in bioengineering

Living systems create mathematical problems at several scales: reaction–diffusion models for signaling, electrophysiology for excitable tissue, continuum mechanics for growth and deformation, CFD for blood flow, fluid–structure interaction for valves and vessels, and inverse problems for patient-specific parameters. Biological models must confront variability, uncertain measurements, changing geometry, and incomplete mechanisms.

### 8.1 Ellen Kuhl

*Stanford University. Walter B. Reinhold Professor in the School of Engineering and director of Stanford Bio-X; living-matter physics and computational biomechanics.*

Kuhl develops theoretical and computational models of living systems, including heart and brain mechanics, growth, remodeling, and personalized simulation. Her work is a strong example of finite elements becoming a translational research tool: continuum mechanics and constitutive laws lead to code, validation, and questions relevant to medicine. She is also a founding member of the Living Heart Project, which connects academic biomechanics and industrial simulation. **Links:** Computational Biomechanics Group; Stanford profile.

### 8.2 Natalia Trayanova

*Johns Hopkins University. Murray B. Sachs Professor of Biomedical Engineering; computational cardiology, electrophysiology, personalized heart models, and ML in cardiology.*

Trayanova develops image-based computational models of diseased hearts to study arrhythmias, risk, and patient-specific treatment. Her work shows the full translational chain: biophysics and PDE models, medical images and parameterization, high-performance simulation, clinical collaboration, and prospective evaluation. It is a valuable counterexample to treating a "digital twin" as only a visualization. **Links:** Johns Hopkins profile; Computational Cardiology Laboratory.

### 8.3 Alison Marsden

*Stanford University. Douglass M. and Nola Leishman Professor; cardiovascular biomechanics, patient-specific simulation, optimization, and medical-device design.*

Marsden develops numerical methods and open software for cardiovascular blood-flow simulation, congenital-heart surgery, and treatment planning. Her work shows how geometry from medical images, CFD and fluid–structure interaction, uncertainty, and clinical questions come together in a patient-specific computational pipeline. **Links:** Stanford profile; Cardiovascular Biomechanics Computation Lab.

## 8.4 Alfio Quarteroni

*Politecnico di Milano and EPFL. Professor Emeritus at both institutions; numerical PDEs, multiscale cardiovascular modeling, and scientific computing.*

Quarteroni develops mathematical and computational models spanning blood flow, cardiac electrophysiology, mechanics, and their interactions. His work illustrates how numerical analysis and multiphysics coupling can build organ-scale simulations without losing sight of stability, approximation, and validation. **Links:** EPFL profile; Politecnico di Milano profile.

## 8.5 Boyce E. Griffith

*University of North Carolina at Chapel Hill. Professor of Mathematics and Biomedical Engineering; biological fluid–structure interaction, cardiac simulation, and adaptive scientific software.*

Griffith develops immersed-boundary and adaptive methods for problems in which fluids interact with flexible biological structures, including heart valves and cardiac tissue. His work is a practical route into the combined challenges of multiphysics modeling, parallel algorithms, verification, and reusable open-source research software. **Links:** UNC research page; IBAMR software.

Connections across this guide are direct. FEniCSx can express elasticity and reaction–diffusion weak forms; PETSc can solve the resulting sparse nonlinear systems; CFD describes circulation; UQ reports credible ranges rather than a single patient prediction; SciML can accelerate repeated evaluation or learn missing constitutive behavior. None of these makes biological validation optional.

**Search terms:** *computational biomechanics, growth and remodeling, cardiac mechanics, hemodynamics, reaction–diffusion, electrophysiology, fluid–structure interaction, patient-specific modeling, cardiac digital twin.*

Chapter 9

# Research software every computational mathematician should know

Software is part of the scientific argument. Learn a small stack deeply enough to inspect assumptions, test convergence, and reproduce results; do not collect library names.

## 9.1 Finite-element ecosystems

### 9.1.1 FEniCS / FEniCSx

FEniCS automates key parts of finite-element computation: one writes variational forms close to their mathematical notation, defines meshes and function spaces, and assembles solvers from Python or C++. **FEniCSx** is the modern generation, built around UFL forms, Basix elements, DOLFINx, MPI, and PETSc. It is an excellent first research framework for Poisson and heat equations, elasticity, Navier–Stokes, reaction–diffusion, biomechanics, and inverse problems.

Begin with Poisson; derive the weak form yourself; perform a mesh-refinement study; then change boundary conditions or coefficients. More ambitious work can couple fields, use adjoints for inversion, or generate high-fidelity data for a surrogate. **Links:** project; DOLFINx documentation; community tutorial.

### 9.1.2 deal.II, MFEM, and Firedrake

`deal.II` is a mature C++ library emphasizing adaptive finite elements, extensibility, and parallel high-performance computing; its numbered tutorials are exemplary research-software education. `MFEM`, developed with strong national-laboratory use, targets scalable finite elements, high-order methods, GPU execution, and modular discretizations. `Firedrake` provides automated FEM through Python and symbolic variational forms, with solver composition through PETSc.

**Links:** deal.II; MFEM; Firedrake.

## 9.2 Solvers and scalable computation

`PETSc` is not primarily an FEM library. It supplies distributed vectors and matrices, Krylov solvers, preconditioners, nonlinear solvers, time integrators, and optimization interfaces. FEM frameworks often assemble a discrete system and delegate its solution to PETSc. Learning to inspect residual histories, tolerances, block structure, and preconditioners is more valuable than selecting a solver by name.

Six important numerical libraries are:

**PETSc** Distributed sparse linear and nonlinear solvers, preconditioners, time integration, and optimization for large PDE-based simulations.

**Eigen** Header-only C++ templates for dense and sparse linear algebra, decompositions, and small-to-medium single-node computations.

**Trilinos** A modular HPC ecosystem for distributed linear algebra, solvers, preconditioning, optimization, discretization, and performance portability.

**FFTW3** Fast one- and multidimensional Fourier transforms for real or complex data, with threaded and MPI execution options.

**SUNDIALS** Robust ODE, DAE, nonlinear-system, and sensitivity solvers for time-dependent scientific models.

**SuiteSparse** Sparse direct solvers and matrix algorithms, including sparse Cholesky and LU factorization, ordering, and graph-based methods.

**Links:** PETSc; Eigen; Trilinos; FFTW3; SUNDIALS; SuiteSparse.

The surrounding stack has layers:

| Layer | Tools | What to learn |
|---|---|---|
| Core numerical work | Python, NumPy, SciPy | arrays, vectorization, sparse matrices, tests |
| Learning / autodiff | PyTorch, JAX, Julia SciML | optimization, automatic differentiation, ODE/PDE workflows |
| Performance | C++, profiling, CUDA | memory hierarchy, kernels, mixed precision |
| Distributed HPC | Linux, MPI, PETSc | domain decomposition, collective communication, scaling |
| Reproducibility | Git, environments, containers | versioned code, dependencies, automated checks |

**Links:** NumPy; SciPy; PyTorch; JAX; Julia SciML; CUDA; MPI standard.

# Part II

# Industry

## Chapter 10

# Industrial research ecosystems

Industrial research is neither one category nor simply product development. A fundamental lab may publish openly and pursue general methods; a scientific-software team may turn decades of numerical analysis into robust solvers; an engineering-simulation group may be judged by validation, certification, and customer workflows. Job titles include research scientist, applied scientist, numerical software engineer, research engineer, computational scientist, simulation engineer, optimization scientist, and HPC engineer. Read actual job descriptions: the same title can mean different work.

## Chapter 11

# Google, DeepMind, Meta, and NVIDIA

## 11.1 Google Research

Google Research conducts both fundamental and applied work, with the aim of turning research into scientific, societal, and product impact. Its output includes papers, datasets, open-source software, models, and systems deployed at very large scale. The organization groups its research into three broad themes:

**Applied AI and sciences** Earth AI, health AI, science AI, sustainability, and crisis resilience. Relevant mathematical work includes physical and statistical modeling, spatiotemporal prediction, uncertainty, optimization, computational biology, and evaluation against scientific baselines.

**Foundational ML and algorithms** Algorithms and theory, information retrieval, machine intelligence, perception, and natural-language processing. This work draws on optimization, probability, statistics, linear algebra, geometry, learning theory, and efficient numerical methods.

**People, systems, and quantum AI** Human–computer interaction, networking, quantum AI, responsible AI, software engineering, and software systems. Applied mathematicians contribute through algorithms, performance modeling, experimental design, numerical reliability, and large-scale computation.

Google Research is distributed across specialized teams such as algorithms and optimization, applied science, climate and sustainability, health, large-scale optimization, learning theory, operations research, and system performance. A project may move between open-ended research, publication, reusable research software, and integration into a Google product. Students should therefore inspect individual teams and recent publications rather than treating Google Research as one homogeneous laboratory.

Typical routes include research scientist, research engineer, software engineer, and internship or student-program roles. A strong portfolio should demonstrate depth in one area, careful empirical evaluation, and the ability to implement an idea at meaningful scale. For scientific work, domain knowledge and validation are as important as model architecture.

**Links:** overview; research areas; teams; publications; careers and locations.

**Monitor:** scientific machine learning, weather and climate, computational biology, algorithms, research scientist, research engineer.

## 11.2 Google DeepMind

DeepMind is distinct but related: its research ecosystem includes reinforcement learning, reasoning, world models, weather, mathematics, biology, and AI for scientific discovery. AlphaFold is the clearest example of AI, geometry, statistics, biology, and scientific validation meeting in one program. Weather work similarly joins physical fields, probabilistic forecasts, numerical benchmarks, and large-scale learning. Students should study both model architecture and evaluation: a scientific claim requires comparison with strong domain baselines.

Potential backgrounds include machine learning, statistics, optimization, computational biology, physics, mathematics, and high-performance systems. **Links:** research; science; AlphaFold.

## 11.3 Meta Fundamental AI Research (FAIR)

FAIR conducts fundamental AI research with strong traditions in open papers and code. Its themes have included self-supervised representation learning, computer vision, language, reasoning, optimization, robotics, and world models. DINO illustrates self-supervised visual representation learning; V-JEPA studies predictive architectures in representation space. These are research directions, not generic product labels.

Mathematics contributes through optimization, probability, representation geometry, robustness, efficient training, and rigorous experiments. Typical roles distinguish research scientists, who often define and publish research programs, from research engineers, who combine experiments, systems, and implementation—though real teams blur the boundary.

**Links:** Meta AI research; V-JEPA; DINOv3.

**Monitor:** representation learning, self-supervised learning, world models, multimodal reasoning, optimization, robustness, research engineer.

## 11.4 NVIDIA Research

NVIDIA is not only a hardware company. Its research connects numerical algorithms, GPU computing, computer graphics, robotics, climate, physical simulation, and learned engineering models.

### 11.4.1 Algorithms and numerical methods

GPU-aware work depends on numerical linear algebra, FFTs, sparse algorithms, iterative solvers, mixed precision, optimization, parallel complexity, and memory movement. A mathematically sound method can still be unsuitable for a GPU if it exposes too little parallelism or moves too much data. Conversely, speed is irrelevant if an unstable implementation corrupts the result.

### 11.4.2 AI-aided engineering

NVIDIA's AI-Aided Engineering research explicitly studies neural operators, geometry, boundary and initial conditions, uncertainty, fluid dynamics, materials, optimization, inverse design, and digital twins. Learned operators can provide fast feedback in a design loop, but training coverage and physical validation delimit where that speed is trustworthy.

### 11.4.3 Physics and simulation

Research includes forward and inverse simulation of fluids, deformable bodies, cloth, muscles, granular materials, and other physical systems; differentiability and neural representations connect simulation to estimation and design. Climate work such as spherical Fourier neural operators illustrates the need to respect domain geometry rather than apply flat-grid machinery blindly.

**Why NVIDIA attracts applied mathematicians**

Numerical mathematics + parallel algorithms + GPU computing + scientific AI.

Strong preparation can come from applied mathematics, numerical analysis, computational physics or mechanics, optimization, graphics, or HPC. Build evidence through a verified solver and careful CPU/GPU performance study, not only a GPU tutorial.

**Links:** algorithms and numerical methods; AI-Aided Engineering; high-fidelity physics; PhysicsNeMo.

**Monitor:** numerical algorithms, CUDA, scientific ML, neural operators, differentiable simulation, climate simulation, inverse design, research engineer.

## Chapter 12

# Scientific software and simulation companies

## 12.1 MathWorks

MathWorks is a mathematical-software company whose MATLAB and Simulink ecosystems encode numerical linear algebra, ODE and DAE integration, optimization, control, signal processing, symbolic computation, visualization, parallel computing, and machine learning. Research-oriented engineering can mean designing an algorithm, translating it into reliable library code, selecting tolerances, preserving backward compatibility, constructing adversarial tests, writing documentation, and profiling on varied hardware.

Possible roles include numerical software engineer, optimization developer, automatic-differentiation engineer, code-generation engineer, and application specialist. A strong candidate can derive an algorithm and also make it predictable for users who do not know its internals. Students should inspect documentation examples as specifications and compare results with independent implementations.

**Links:** software careers; numerical computing; parallel computing.

**Monitor:** numerical algorithms, solver development, optimization, control, automatic differentiation, scientific visualization, code generation.

## 12.2 Dassault Systèmes / SIMULIA

The SIMULIA portfolio includes Abaqus for finite-element analysis and tools spanning structural mechanics, nonlinear materials, contact, durability, CFD, electromagnetics, optimization, multiphysics, and virtual testing. Applications range across aerospace, automotive, biomedical devices, manufacturing, materials, and energy. The wider Dassault ecosystem frames integrated models as virtual twins.

Industrial FEM differs from a focused academic code. Research may propose a new element or prove convergence on a model problem; industrial software must handle complex CAD geometry, units, material libraries, contact, restarts, noisy meshes, licensing, huge model files, and long-lived user workflows. Verification, validation, documentation, performance, and regression testing become first-class requirements. Modern directions include simulation surrogates, AI-assisted design exploration, and acceleration, but they inherit the need for physical validity.

Students who enjoy FEniCS, FEM, or computational mechanics can prepare by learning continuum mechanics, nonlinear solution methods, meshing, and software engineering. Reproduce a benchmark in both an open research code and a commercial package if access permits; explain differences rather than treating one as truth.

**Links:** Abaqus; SIMULIA; virtual twins.

**Monitor:** Abaqus, nonlinear FEM, computational mechanics, multiphysics, solver developer, simulation methods, virtual twin, surrogate modeling.

## 12.3 Microsoft Research / AI for Science

Microsoft Research's AI for Science work joins machine learning with computational chemistry, molecular simulation, materials, biology, physics, and scientific emulators. The core opportunity is not to paste AI onto a discipline, but to represent scientific objects and constraints, build useful training or simulation data, quantify uncertainty, and connect predictions to experiments.

**Links:** AI for Science; science research.

**Monitor:** AI for science, molecular simulation, computational chemistry, scientific foundation model, emulator, materials discovery, research software engineer.

## 12.4 US national laboratories

Argonne, Lawrence Livermore, Los Alamos, Oak Ridge, and Sandia are important destinations between academia and industry. They work on PDE solvers, numerical linear algebra, HPC, climate, materials, optimization, UQ, and scientific ML, often at scales or mission horizons unavailable elsewhere. Many widely used packages, including PETSc and MFEM, have deep laboratory roots. Citizenship or clearance conditions apply to some roles, but not all research opportunities; always read eligibility rules.

**Links:** Argonne Mathematics and Computer Science; LLNL Computing; ORNL Computing; Sandia Computing Research; Los Alamos simulation and computation.

## Chapter 13

# Frontier AI research organizations

## 13.1 OpenAI

From a research perspective, OpenAI works on frontier models, reasoning, reinforcement learning, multimodal systems, code, agents, evaluation, and safe deployment. Mathematical relevance includes probability, optimization, algorithms, formal reasoning, statistics, experimental design, and high-performance systems. Scientific-computing experience is valuable when models must interact with code, simulations, data, and verifiers.

AI as a scientific collaborator is an emerging workflow: it may help search literature, propose constructions, write or debug code, analyze outputs, and connect tools. It does not remove responsibility for proof checking, numerical verification, provenance, uncertainty, or experimental confirmation. The scientifically interesting research questions include how to evaluate open-ended reasoning, couple models to formal or computational verifiers, recognize uncertainty, and design agents that preserve an auditable trail.

**Links:** research; science; careers.

**Monitor:** mathematical reasoning, reinforcement learning, agents, code generation, evals, formal methods, AI for science, research engineer.

## 13.2 Anthropic

Anthropic's official research structure emphasizes alignment, interpretability, economic research, societal impacts, and frontier red teaming, alongside work on model capabilities and reasoning. Mechanistic interpretability asks how internal computations produce behavior. In mathematical language, one seeks structured descriptions of a high-dimensional nonlinear system: sparse features, dictionaries, circuits, causal interventions, and representations.

The group's scaling-monosemanticity work applies dictionary learning through sparse autoencoders to model activations. The analogy to sparse coding is useful, but features should not be treated as a complete explanation: representation discovery, circuit-level use, causal validation, and scalability are separate problems. Research and engineering are tightly linked because experiments require distributed data pipelines and large-scale model instrumentation.

Anthropic also publishes science-oriented work and collaborations. Treat claims of autonomous discovery cautiously: inspect the task, human contribution, tools, evaluation, and reproducibility. Relevant backgrounds include ML, mathematics, statistics, neuroscience, biology, software systems, security, and policy-aware empirical research.

**Links:** research; scaling monosemanticity overview; science program; careers.

**Monitor:** mechanistic interpretability, sparse autoencoder, dictionary learning, alignment science, model evaluations, reasoning, AI safety, research engineer.

# Chapter 14

# Researchers from Kazakhstan

The Qazaq International Science and Technology Association (QIST) is the largest global professional association of Qazaq scientists and technology specialists. Founded in 2014 and formally registered in 2024, it connects researchers, professors, doctoral students, and industry experts across more than 30 countries. QIST supports professional networking, research collaboration, fellowships, mentorship, industry partnerships, and engagement with science and technology policy.

**Links:** QIST website; about QIST.

The people below are some of the researchers and professionals known to me whose outstanding profiles I find inspiring. They do not necessarily work in applied mathematics, scientific computing, or machine learning; they are included because their education, research, and career paths may inspire you. This is a personal selection, not a ranking or an exhaustive directory.

## 14.1 Batyr Ilyas

Batyr Ilyas is a postdoctoral researcher at UC Berkeley and Lawrence Berkeley National Laboratory who studies quantum materials through ultrafast and scanning-probe spectroscopy.

**Links:** Google Scholar; LinkedIn.

## 14.2 Nurlybek Mursaliyev

Nurlybek Mursaliyev is the founder and CEO of Biodock, where he applies artificial intelligence to microscopy and biological image analysis for research and drug development.

**Links:** Google Scholar; LinkedIn.

## 14.3 Maulen Uteuliyev

Maulen Uteuliyev is a drug-discovery scientist at UT MD Anderson Cancer Center whose work draws on synthetic and medicinal chemistry to advance preclinical oncology programs.

**Links:** Google Scholar; LinkedIn.

## 14.4 Zhambyl Shaikhanov

Zhambyl Shaikhanov is an assistant professor of electrical and computer engineering at the University of Maryland whose research spans next-generation wireless networking, security, and sensing.

**Links:** Google Scholar; LinkedIn.

## 14.5 Arman Zharmagambetov

Arman Zharmagambetov is a research scientist at Meta working at the intersection of machine learning, optimization, and the security and robustness of AI systems.

**Links:** Google Scholar; LinkedIn.

## 14.6 Aisulu Aitbekova

Aisulu Aitbekova is an assistant professor of chemical engineering at Carnegie Mellon University who develops light- and heat-driven catalytic processes for sustainable fuels and chemicals.

**Links:** Google Scholar; LinkedIn.

## 14.7 Ayan Zhumekenov

Ayan Zhumekenov is a materials chemist and research fellow at Nanyang Technological University whose work focuses on functional materials, halide perovskites, and optoelectronic applications.

**Links:** Google Scholar; LinkedIn.

## 14.8 Aida Zhumabekova

Aida Zhumabekova is a software engineer at Intrinsic who develops intelligent robotics and automation systems for advanced manufacturing.

**Links:** Google Scholar; LinkedIn.

Chapter 15

# Research resources and links

## 15.1 Learn the landscape

**SIAM** SIAM News, journals, conferences, and student chapters reveal how applied mathematics is organized.

**arXiv** Browse Numerical Analysis, Machine Learning, and relevant physics or engineering categories. Use alerts selectively.

**Google Scholar / Semantic Scholar** Follow authors and citation trails, but judge sources by content and venue rather than ranking.

**Eigensteve** Brunton's video courses cover linear algebra, dynamics, control, and data-driven modeling.

**Computational Science Stack Exchange** scicomp.stackexchange.com is useful when a question is precise and reproducible.

## 15.2 Foundational reading

- Trefethen and Bau, *Numerical Linear Algebra*; Higham, *Accuracy and Stability of Numerical Algorithms*.
- LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations*; Brenner and Scott, *The Mathematical Theory of Finite Element Methods*.
- Brunton and Kutz, *Data-Driven Science and Engineering*; Kutz, *Data-Driven Modeling & Scientific Computation*.
- Boyd and Vandenberghe, *Convex Optimization*.
- Bishop, *Pattern Recognition and Machine Learning*; Murphy, *Probabilistic Machine Learning*.
- Quarteroni, Manzoni, and Negri, *Reduced Basis Methods for Partial Differential Equations*.

Do not attempt the list linearly. Choose the chapter needed for a project, solve exercises, and record what the theory explains in your experiments.

## 15.3 Five major conferences

These recurring conferences provide broad views of the communities most relevant to this guide. They are not a ranking; the best venue depends on the research question.

**SIAM CSE** The SIAM Conference on Computational Science and Engineering covers modeling, simulation, numerical methods, scientific software, HPC, uncertainty quantification, and scientific machine learning. Conference page.

**SIAM Annual Meeting** A broad meeting across applied mathematics, computational science, data science, PDEs, optimization, biology, climate, and industrial applications. SIAM conferences.

**NeurIPS** A major interdisciplinary meeting for machine learning, optimization, probabilistic methods, representation learning, and AI systems, with tutorials and specialized workshops. Conference page.

**ICML** A leading machine-learning conference spanning theory, algorithms, empirical methods, and applications. Conference page.

**SC** The International Conference for High Performance Computing, Networking, Storage, and Analysis; especially relevant to parallel algorithms, scientific applications, and research software. Conference page.

## 15.4 A paper-reading protocol

On the first pass, identify the question, claim, evidence, and assumptions. On the second, reconstruct notation and the key derivation. On the third, inspect experiment design and failure cases. Then reproduce the smallest result. Keep a bibliography file and notes containing exact page/equation references; one carefully read paper per week is already a substantial pace.

**Link-quality rule**

Prefer a researcher's page, official university or company page, project documentation, and the original paper. A summary can help navigation but should not be the foundation for an affiliation or technical claim.