

MATHEMATICS:

KEY ENABLING TECHNOLOGY FOR SCIENTIFIC MACHINE LEARNING



MATHEMATICS: KEY ENABLING TECHNOLOGY FOR SCIENTIFIC MACHINE LEARNING

CONTENT

3 -	EXECUTIVE SUMMARY
5-	MATHEMATICS: KEY ENABLING TECHNOLOGY FOR SCIENTIFIC MACHINE LEARNING
19-	THE NATIONAL INITIATIVE 'AI AND MATHEMATICS' (AIM) THE IMPORTANCE OF MATHEMATICS FOR DUTCH AI RESEARCH
23	MSODE: MODELLING, SIMULATION AND OPTIMIZATION IN A DATA-RICH ENVIRONMENT WIL SCHILDERS
31-	RANDOM PROCESSES IN MACHINE LEARNING JORIS BIERKENS
36-	MACHINE LEARNING FOR MULTISCALE SIMULATIONS IN CLIMATE DAAN CROMMELIN AND WOUTER EDELING, CENTRUM WISKUNDE & INFORMATICA, AMSTERDAM
39-	FRUITFULLY COMBINING LINEAR ALGEBRA, STATISTICS, OPTIMIZATION, AND DATA SCIENCE GIULIA FERRANDI, PERFECT GIDISU, SIBYLLE HESS, AND MICHIEL HOCHSTENBACH
44-	A TALE OF TWO CONFERENCES, OR: THE ONGOING DANCE BETWEEN MATH AND MACHINE LEARNING PETER GRÜNWALD
47	UNCERTAINTY QUANTIFICATION FOR PHYSICS-INFORMED DEEP LEARNING MENGWU GUO AND CHRISTOPH BRUNE
52	GPU ACCELERATION OF THE SEVEN-LEAGUE SCHEME FOR LARGE TIME STEP SIMULATIONS OF STOCHASTIC DIFFERENTIAL EQUATIONS SHUAIQIANG LIU, GRAZIANA COLONNA, LECH GRZELAK AND CORNELIS OOSTERLEE
57-	PHYSICS-INFORMED MACHINE LEARNING EMBEDDED INTO ISOGEOMETRIC ANALYSIS MATTHIAS MÖLLER, DEEPESH TOSHNIWAL, AND FRANK VAN RUITEN
60-	GEOMETRIC DEEP LEARNING FOR PRECISION MEDICINE JELMER WOLTERINK AND JULIAN SUK
63-	GEOMETRIC LEARNING FOR IMAGE ANALYSIS REMCO DUITS
78-	ABOUT THIS BOOKLET

80

EXECUTIVE SUMMARY

MATHEMATICS: KEY ENABLING TECHNOLOGY FOR SCIENTIFIC MACHINE LEARNING

Artificial Intelligence (AI) will strongly determine our future prosperity and well-being. Due to its generic nature, AI will have an impact on all sciences and business sectors, our private lives and society as a whole. AI is pre-eminently a multidisciplinary technology that connects scientists from a wide variety of research areas, from behavioural science and ethics to mathematics and computer science.

Without downplaying the importance of that variety, it is apparent that mathematics can and should play an active role. All the more so as, alongside the successes of AI, also critical voices are increasingly heard. As Robert Dijkgraaf observed in NRC in May 2019: "Artificial intelligence is in its adolescent phase, characterised by trial and error, self-aggrandisement, credulity and lack of systematic understanding." Mathematics can contribute to the much-needed systematic understanding of AI, for example, greatly improving reliability and robustness of AI algorithms, understanding the operation and sensitivity of networks, reducing the need for abundant data sets, or incorporating physical properties into neural networks needed for superfast and accurate simulations in the context of digital twinning. Dutch mathematicians absolutely recognise the potential of artificial intelligence, machine learning and (deep) neural networks for future developments in science, technology and industry. At the same time, a sound mathematical treatment is essential for all aspects of artificial intelligence, including imaging, speech recognition, analysis of texts or autonomous driving, implying it is essential to involve mathematicians in all these areas. In this booklet, we highlight the role of mathematics as a key enabling technology within the emerging field of scientific machine learning. We present the national initiative ''AI and Mathematics'' (AIM), and showcase the work of several Dutch mathematicians.



MATHEMATICS: KEY ENABLING TECHNOLOGY FOR SCIENTIFIC MACHINE LEARNING

Computational scientific discovery is at an interesting juncture. While traditionally we have models of lots of different scientific phenomena, such as the famous Maxwell and Navier-Stokes equations, and abundant data being generated from experiments - our computational capabilities appear unable to keep up. Often, problems are too large for realistic simulation. Besides, problems are multiscale and very stiff. Solving such problems requires tedious work on suitable algorithms as well as getting code to run on GPUs and supercomputers. The next step forward is a combination of scientific computing and machine learning, combining mathematical models with data based reasoning, presented as a unified set of abstractions and a high performance implementation. This new area of research is referred to as scientific machine learning.

Scientific machine learning has been taking the academic world by storm as an interesting blend of traditional scientific modeling with machine learning methodologies like deep learning. While traditional deep learning methodologies have had difficulties with scientific issues like stiffness, interpretability, and enforcing physical constraints, this blend with numerical analysis and differential equations has evolved into a field of research with new methods, architectures, and algorithms which overcome these problems while adding the data-driven automatic learning features of modern deep learning. Many successes have already been found, with tools like physicsinformed neural networks, universal differential equations, deep backward stochastic differential equation solvers for high dimensional partial differential equations, and neural surrogates showcasing how deep learning can greatly improve scientific modeling practice.

Mathematics will be essential in addressing the challenges that we encounter in the rapidly evolving field of scientific machine learning. Below, we will discuss these challenges in more detail, and indicate how mathematics can provide solutions.



Scientific machine learning

Scientific machine learning is a core component of artificial intelligence and a computational technology that can be trained, with scientific data, to augment or automate human skills. Scientific machine learning has the potential to transform science and engineering research. Breakthroughs and major progress will be enabled by harnessing investments in massive data from scientific user facilities, mathware and software for predictive models and algorithms as well as high-performance computing platforms, besides a diverse contingent of researchers in many disciplines. The crosscutting nature of machine learning and artificial intelligence provides a strong incentive for formulating a prioritized research agenda to maximize the capabilities and scientific benefits; mathematics is a key technology that is indispensable in achieving progress in this important area.

In recent years, the interest in machine learning based approaches for science and engineering has increased rapidly. This growing enthusiasm stems from the combined development and use of efficient data analysis algorithms; massive amounts of data available from scientific instruments, scientific computations, and other sources; advances in high-performance computing; and the successes reported by industry, academia, and research communities. A conventional notion of machine learning involves training an algorithm to automatically find patterns, signals, or structure that may be hidden within massive data sets whose exact nature is unknown and therefore cannot be programmed explicitly. The algorithm's predictive capability is a learned skill⁽¹⁾. We seek to improve upon and harness the analytical and predictive power of machine learning to maximize its impact on science and engineering applications. This relatively new field is named scientific machine learning, with mathematics as its key technology.

There are many potential benefits from engaging machine learning in computational science. Machine learning methods are particularly useful for discovering correlations in high-dimen-

D AI AND MATHEMATICS

Samuel in 1959, is that machine learning is the field of study that gives computers the ability to learn without being explicitly programmed. A more recent and detailed definition is due to Jordan: "learning" is the process of transforming information into expertise or knowledge; "machine learning" is automated learning. The input information to machine learning includes training data and, in the case of scientific machine learning, a priori knowledge or information (such as physical conservation principles). The output information is "knowledge" or "expertise" used to make predictions or perform some activity.

So what is machine learning? A simple definition, due to Arthur

Scientific machine learning poses a rich set of challenges that span a broad set of potential use cases and, in contrast to many existing applications of machine learning, a deeper consideration of the structure of the problem at hand is critical. Indeed, simulation supports a tremendously broad range of activities, some of which support scientific goals of discovery and understanding and others that underpin decisions through prediction, optimization, and uncertainty quantification. The nature of the potential impact of scientific machine learning varies across these activities, just as the need for new scientific machine learning methodologies varies. We draw an analogy with the past decades of advancements in optimization: these advancements have been driven by an explicit and clear realization of the diverse structure of different optimization problem classes (linear programs, integer programs, mixed-integer programs, partial differential equation constrained optimization, etc.) and the need for structure-exploiting techniques. Similar investments are required to advance domain-specific, structure-exploiting scientific

machine learning.

Predicting outside the range of training data may, however, lead to 1 erroneous results

sional data and, thus, can be useful in analyzing computational results. Machine learning methods also can construct surrogates for complex forward models; e.g., with neural network (NN), Gaussian process (GP), and related methods. Machine learning can also assist in dimension reduction for high-dimensional data; e.g., by learning/discovering low-dimensional manifolds underlying the data, based on mathematical principles. Such dimension-reduction methods can be employed for understanding the dynamical structure behind the data. Machine learning based dimension reduction also can help define effective distance measures between data sets, thereby providing paths toward effective likelihoods for complex model calibration and parameter estimation from observational data. Already, machine learning methods have been used in various science applications, including analyzing turbulent flow computations, subsurface flow modeling, solid mechanics modeling of diverse materials, geophysics, and combustion modeling.

Still, despite these successes, numerous challenges remain in the path toward routinely adopting machine learning in scientific applications. In many of the most successful machine learning examples, such as image recognition, system developers know the "ground truth" sufficiently well to check the results, often even while training the models. Almost by definition, the most interesting scientific applications of scientific machine learning are those, such as materials discovery or high-energy physics, where the answers are unknown beforehand or the results of an automated system are not easily verified. Instead, familiar questions from scientific computing are, for example:

- How reliably will a given algorithm work; e.g., for what type and quantity of data do we expect results?
- How robust is a certain solution; e.g., how might slightly different data or the addition of noise change the results?
- How rigorously have the assumptions and underlying theories been defined and validated? For classical tech-

niques, such as analyzing partial differential equation based models, these questions lead to familiar concepts, including well-posedness, stability, numerical approximation, and uncertainty quantification.

These questions have led to a long history of relevant research and to reliable and robust outcomes from partial differential equation based models used in many different application areas. On the other hand, equivalent concepts for scientific machine learning based models are not well established, and the lack of precise definitions and clearly expressed assumptions often leads to the failure of machine learning based methods. Finally, the ultimate goal of analysis is for a scientist to gain new insights, adding a human dimension to the scientific learning problem. This process requires both integrating the existing body of human knowledge into the scientific machine learning approach and providing help for users to understand how a given approach works.

Independent of these fundamental (and somewhat abstract) differences between machine learning as seen in the media and scientific data analysis, significant practical and technical distinctions also exist. Scientific machine learning has the potential to significantly advance diverse scientific areas and will transform the way science is done. It is likely that before long, various experiments and simulations will no longer be primarily limited by what data they can collect but by how well they are able to extract insights from the data they have. However, to take full advantage of the combination of massive data collections and scientific machine learning for scientific discovery, we must understand the current state of the art, where it may not meet the demand of various scientific applications, and what the key open research directions are to address the shortcomings. Mathematics is the key technology that is ready to take on the challenges and contribute to addressing these shortcomings.

Scientific machine learning methods are unlikely to ever replace established domain models based on physical mechanisms and scientific knowledge; however, there is a significant opportunity for scientific machine learning to complement traditional domain models. Domain knowledge includes physical principles, symmetries, constraints, expert feedback, computational simulations, uncertainties, etc. Research should focus on integrating such domain knowledge with scientific machine learning methods. Such integration is expected to improve accuracy, interpretability, and defensibility of scientific machine learning models while simultaneously reducing data requirements and accelerating scientific machine learning model training. Progress will require new mathematical methods to learn improved model features that are constrained by domain knowledge, including fusion of multimodal and heterogeneous data sources to extract features.

Incorporating scientific domain knowledge in the machine learning process is a task unique to scientific machine learning. Awareness of domain knowledge can enhance domain-agnostic data in terms of accuracy, interpretability, and defensibility of scientific machine learning models. Furthermore, incorporating scientific domain knowledge has the potential to dramatically reduce data requirements, as well as to accelerate training and prediction.

Domain knowledge is found in many forms, such as physical principles, constraints, symmetries, conservation laws, and other knowledge gained from theoretical or computational studies. Scientific domain knowledge can be expressed in many forms, including physical models (e.g., ab initio or first-principles physics), physical constraints (e.g., symmetries, invariances, conservation laws, asymptotic limits), computational simulations, uncertainties, correlations in space and time, and structural forms (e.g., discrete, graph-like, non-smooth data). For such domain knowledge, both theoretical foundations and computational infrastructure exist (e.g., solvers and simulations) that can benefit scientific machine learning.

Domain knowledge has been proven to help supervised and unsupervised machine learning, as well as in generating synthetic data (e.g., with constrained generative adversarial networks) and reinforcement learning. Although scientific data may satisfy (e.g., modulo various types of errors and noise) underlying laws of physics, directly leveraging such domain knowledge can allow the learning process to focus on modeling more challenging and computationally impractical phenomena with less labeled data. Domain knowledge can be incorporated for various objectives, including improved interpretability and robustness, as well as in a multitude of ways; e.g., both machine learning enhanced modeling and simulation as well as intelligent automation and decision support tasks.

Research is needed regarding methods that incorporate domain knowledge into feature selection. Progress will require new mathematical methods to learn improved model features that are constrained by domain knowledge, including fusion of multimodal and heterogeneous data sources to extract features. The central question is "which knowledge should be leveraged in scientific machine learning, and how should this knowledge be included?"

One research avenue involves incorporation of domain knowledge through imposition of constraints that cannot be violated. These hard constraints could be enforced during training, replacing what typically is an unconstrained optimization problem with a constrained one. A similar avenue for incorporating domain knowledge involves modifying the objective function (soft constraints) used in training. A particular challenge is the need to incorporate uncertain or incomplete domain-specific knowledge, as well as multiple physics and data sources that have different time or space fidelity and/or are multiscale. These challenges are similar to those evident in the mathematical areas of reduced order modeling, multifidelity optimization, and uncertainty quantification, but with an acute need for techniques (e.g., quantifiable error metrics and objectives) that facilitate the automation required in a machine learning pipeline.

Another research avenue incorporates domain knowledge through the chosen form (e.g., basis employed) of the machine learning model. For example, incorporating algebraic invariances, such as symmetries and scaling in kernel approaches, have been shown to improve generalization performance. Similarly, convolutional neural networks can incorporate knowledge in vision processing domains through convolutional filters that exploit locality, whereas the recurrent nature of recurrent neural networks can incorporate knowledge for speech recognition. An opportunity exists for principled selection of model forms across broad scientific domains, as well as for understanding the associated computational burden and effect on accuracy.

Interpretable scientific machine learning

Traditionally, physical understanding has been the bedrock of modeling. A user's confidence in a model's predictions is directly linked to the conviction that the model accounts for her/his domain knowledge; e.g., the right variables, parameters, and physical laws. In general, a tension exists between the need for increased complexity in machine learning models to improve results and the need for users to interpret the models and derive new insights and conclusions. This challenge has been widely recognized. However, scientific machine learning applications have unique challenges and opportunities to use existing domain knowledge to increase machine learning model interpretability. Progress will require developing new exploration and visualization approaches to interpret complex models using domain knowledge, as well as new mathematical metrics to quantify model differences.

One of the challenges in applying machine learning is the inherent complexity of many of its techniques. The canonical examples for machine learning complexity are deep learning based approaches. Deep learning promises unprecedented advances in dealing with a range of data types but relies on millions of degrees of freedom, connected in complex arrangements and trained through hand-tuned optimizations. In machine learning, significant expertise exists for selecting architectures, tuning optimization procedures, etc. However, current understanding is limited regarding how and/or why these techniques work and why they can be predictive. As a result, machine learning approaches may provide excellent classification performance while struggling to afford insight into the solution. Because novel insights are essential to science and engineering, the interpretability of machine learning methods must be improved, and mathematics can provide the missing links here.

Typically, one of the first steps in any data-processing pipeline is an initial exploration of data set structure. For example, questions regarding how smooth a given regression function should be necessitates some understanding of the geometry of both spaces, as well as the mapping between them. In general, understanding data characteristics has the potential to contribute significantly to the entire scientific machine learning pipeline-from understanding the distribution of the input data, to analyzing the model fitting through its path on the optimization landscape, to interpreting the output. In traditional scientific data analysis, this step often involves common visualization approaches; e.g., rendering slices through threedimensional data, computing level sets, etc. Visualization provides a guick and intuitive way to understand data smoothness, the range of values, or if the data should be scaled. However, scientific machine learning data often are high-dimensional and/or complex, which complicates visualization. There is a need for methods that provide (human) users with scientific machine learning insights into data characteristics beyond traditional statistical indicators or other integrated measures.

Research is needed to develop quantitative measures or replace them with interpretable qualitative characteristics. These may include new types of descriptors to convey the geometry, topology, or general information content of high-dimensional data. These challenges are intimately related to areas in mathematical research such as applied differential geometry.

Once a machine learning model has been selected, it is important to interpret the process by which it is fit/optimized to the data. That is, can the relationship between the model input, operation, and output be rationalized or explained? Such interpretability allows users to understand the model's results along with their robustness and sensitivity. Traditionally, physical understanding has been the bedrock of modeling. A user's confidence in model predictions is linked directly to the conviction that the model accounts for the right variables (e.g., temperature, pressure, or material density), parameters (e.g., inflows or reaction rates), and physical laws (e.g., heat-mass balance or energy). Very simple models are readily interpretable. However, once the problem dimensions extend beyond a few and the model's complexity increases slightly, then model understanding, particularly for stakeholders, becomes significantly more difficult. This problem is greatly exacerbated for nonlinear models. Models, such as deep neural networks, have sufficient nonlinearity and complexity to complicate routine interpretation. Although the performance of these complex models is impressive, their lack of interpretability makes them insufficient for high-regret and safety-critical systems. For verification and for developing trust in the model, intuitive model introspection and interactive exploration of the solution space are vital for convincingly conveying results to stakeholders.

Given this objective of decomposing the model and decision process into interpretable human- meaningful and humanmanageable steps, research is needed to provide a decision process decomposition for complex scientific machine learning models. In doing so, the human-meaningful steps may include connecting abstract representations to known laws (e.g., of physics) or interpretable concepts. Research is needed to provide model exploration and interpretation capabilities, as well as to enable the trade-off between model interpretability, flexibility, and accuracy for use in model selection. To take its place as a scientific methodology and be accepted for common use in domain sciences and high-regret applications, scientific machine learning methods must be robust and reliable. While machine learning methods are much used, the integration of protocols for verification, validation, and reproducibility are in their infancy. The credibility of research based on scientific machine learning requires that outcomes come from a process that is not sensitive to perturbations in training data, modeling choice, and/or computational errors. Progress will require research for showing that scientific machine learning methods and implementations are well-posed, stable, and robust. Mathematics possesses the tools to address this challenge.

Machine learning already has made a significant impact in a variety of high-regret applications, including clinical diagnoses, security, metal fatigue in aviation, and environmental contamination. Such applications exemplify the importance of robustness and rigor in machine learning to minimize the risks associated with its use. These applications also point out the need for interpretability in scientific machine learning predictions. The potential negative impacts of misused machine learning has led to calls for policies to "anticipate, monitor, and retrospectively review operations" for managing algorithms in high-impact applications. and to initiate research that explores ways to manage algorithm behavior.

To gain acceptance as a legitimate scientific methodology, scientific machine learning must achieve the same level of scientific rigor expected of established methods deployed in science and applied mathematics. Basic requirements include validation and limits on inputs and context implicit in such validations, as well as verification of the basic algorithms to ensure they are capable of delivering known prototypical solutions exactly (cf., the use of manufactured solutions to test numerical algorithms). In essence, these properties encapsulate a requirement for the scientific methodology to be reproducible and for the basic techniques to be well-posed and stable.

With its emphasis on well-defined analytic processes for stability and error analysis, applied mathematics can provide a mechanism for developing scientific machine learning methods with robustness. Such robustness will address issues related to sensitivity to training set size, choice of data in training and test sets, numerical instability in learning algorithms, scalability, and parallelization with complicated and heterogeneous hardware. While the need to investigate these issues has been expressed since the early days of machine learning and artificial intelligence, more research is still needed. For instance, to set scientifically based rather than heuristic guidelines on acceptable classifiers that process experimental data from beamlines or predict failure in major components. It also is extremely unlikely that existing applied mathematics methodology will extend automatically to scientific machine learning. For example, classical linear algebra approaches are designed to optimize computations with sparsity structures arising from discretisations of PDEs. However, scientific machine learning needs are likely to be quite different. Similarly, classical von Neumann stability analysis of numerical approximations is unlikely to be adequate for establishing stability of complex learning schemes. Thus, complementary research in applied mathematics methodology is necessary.

There is a basic need to establish a solid mathematical foundation for studying properties of the underlying implicit model, the algorithms used to analyze the models, and the sensitivities of outcomes to training data. For effective use in advancing and testing scientific hypotheses, scientific machine learning must be insensitive to the effects of intrinsic perturbations—in data and the model—that are not symptomatic of the underlying system. There are substantial links between the concept of scientific machine learning "stability" and the broader concept of "well-posedness". These foundations should formulate concepts and definitions that pave the way for a deeper understanding of scientific machine learning.

A mathematical framework provides the option to develop models and algorithms that are insensitive to the effect of perturbations not intrinsic to the underlying system. Appropriate regularization enables useful information to be gleaned from an inverse problem in a clear and reproducible fashion. There is a challenge to identify appropriate techniques that can play a similar role in scientific machine learning.

The outcome of a machine learning process is either a decision (classification) or a prediction. For reliable and credible use of scientific machine learning, we need the ability to rigorously guantify machine learning performance in these outcomes. Performance measurement implies an assessment of quality, as well as a cost measure of computations and/or data preparation and management. Traditional measures of acceptable quality based on statistical cross-validation-type approaches often are heuristic. Measures of prediction quality such as a priori and a posteriori error estimates for numerical approximations of partial differential equations will be transformative in allowing the development of optimal and reliable machine learning algorithms for different uses. Such error estimates also will enable scientific machine learning processes that allow iterative model improvement. Research establishing quantitative estimates of prediction quality, including effective confidence bounds, will greatly enhance the usefulness of scientific machine learning to decision makers and users. Finally, research is needed on algorithms that have proven convergence rates with weak dependence on bad data, especially in situations with a large amount of data of unproven quality or minimal availability of human expertise.

Data-intensive scientific machine learning

Scientific machine learning in large-scale complex models and data faces a range of challenges, including high-dimensional, noisy, and uncertain input data, as well as limited information about model validity. Incorporating statistics, uncertainty quantification, and probabilistic modeling into scientific machine learning will provide a framework for managing some of these challenges. In particular, these approaches can address ill-conditioning, non-uniqueness, and over-fitting and allow for requisite uncertainty quantification in machine learning predictions. In addition, statistical and probabilistic methods can help uncover structure in data to improve scientific insight. At the same time, applying these methods in scientific machine learning is challenged by large data volume and complexity, as well as the high-dimensional structure of probabilistic scientific machine learning models. Progress requires developing improved methods for statistical learning in high-dimensional scientific machine learning systems with noisy and complex data, for identifying structure in complex high-dimensional data, and for efficient sampling in high-dimensional parametric and model spaces.

The central role of data in scientific machine learning suggests an associated fundamental role for statistics and uncertainty quantification methods. Data used to train machine learning models are often noisy, uncertain, incomplete, sparse, and only partly informative. Similarly, machine learning models themselves are subject to uncertainty in their general form, internal structure, and associated parameters. Statistics provide an array of methods to address these data and model complexities and uncertainties in scientific machine learning. Ultimately, this enhances the ability to use extreme-scale computations and experimental data for scientific discovery in physical systems highly relevant to the science and engineering community.

Statistics and uncertainty quantification methods can add significant robustness to scientific machine learning fitting/regression methods. Often, statistical methods are useful for dealing with the problem of over-fitting with deep learning and convolutional neural networks given small amounts of data. In many places where multiple minima exist and the solution space is large, a high degree of ill-conditioning is present, considering the many nearly equivalent feasible solutions. Reformulating the issue as a statistical inverse problem can add significant conditioning, changing the question from determining the best solution to finding the set of solutions with significant probability. Further, this reformulation provides a solution with quantified uncertainty estimates in its parameters/weights and potentially its structure. For example, incorporating probabilistic modeling in machine learning targets feasible estimation of uncertainty in machine learning predictions. A robust probabilistic/statistical analysis of noise, errors, and uncertainties in machine learning in high-dimensional systems enables reliable discovery of correlations and causal structures in large scale systems of interest. Similarly, Bayesian integration is useful for general high-dimensional function applications with relevance in machine learning. Bayesian modeling also is finding utility in Bayesian generative adversarial networks. Additionally, there are potential advantages for regression and scientific machine learning with nonparametric models, such as Gaussian processes, versus pre-determined neural network structures in deep learning.

Despite the stated benefits, some challenges often increase upon setting the machine learning problem in a probabilistic context. These include increased dimensionality from probabilistic modeling, the need to deal with high computational costs in uncertainty quantification sampling strategies, and the need for adequate error modeling in data and models. The non-parametric identification of structure in high dimensional data is a significant challenge. High dimensionality can necessitate large numbers of samples to allow for reliable identification of underlying structure in high-dimensional computational and experimental data. This is particularly true when the system does not admit a sufficiently low-dimensional underlying structure. Likewise, uncertainty quantification in machine learning in high dimensions presents a number of challenges. Uncertainty quantification for scientific machine learning involves the formulation of probabilistic machine learning models, inference of machine learning model structure and (hyper)-parameters with quantified uncertainty, and forward propagation of uncertainty to machine learning model predictions. Both the statistical inverse problem and the forward uncertainty quantification problem can become quite expensive in high-dimensional complex machine learning models.

Advances are necessary to improve methods for analysis and discovery of structure in scientific data and physical systems models. Knowledge of underlying structure affords a means for interpreting the features of data and models. Structure includes a number of underlying properties of data and models. For example, correlation and causal structure in data and models are crucial ingredients. Similarly, structure includes the specification of low-dimensional manifolds and geometry underlying high-dimensional data and complex dynamical systems. Structure is a crucial component of some machine learning methods (e.g., Gaussian process models), and the accuracy of these methods depends on how well one can learn the structure from data or physics-based models. While there is a large body of work in this area, there are significant gaps in the state of the art when dealing with both high dimensionality and complexity in data and models.

Despite recent advances in methods for discovering low-dimensional structures in data, much work is needed in applying these methods to large-scale physical systems. Continued research toward improved methods for discovering sparsity and low-rank structure in machine learning models will be useful for dealing with the high-dimensionality challenge. Additionally, extracting models from data is an important area of work toward improved machine learning. Similar advances are needed to improve methods for discovering structure underlying data, including learning underlying geometry beyond principal component analysis and building on work targeting discovery and representation of diffusion manifolds. Knowledge of data structure is necessary for effective sampling on manifolds, potentially allowing for embedding physical constraints in scientific machine learning system training. There also are opportunities for advances in the definition and utilization of geodesics and correlation metrics for providing measures of distance that are useful for model comparison, calibration, selection, and validation.



Machine learning enhanced modelling and simulation

Simulation codes available in many domains of science and engineering model complex physical phenomena, often with dramatic variations in scale and behavior even within a single simulation. For performance, robustness, and fidelity, human expertise typically is integral in the simulation process to obtain quality solutions. The growing trend is for the models, discretisations, and numerical solvers at the heart of application codes to be more adaptive, usually through the use of simple theoretical controls and/or heuristics. There still are tremendous gains to be realized through the judicious use of scientific machine learning algorithms to better adapt aspects of the numerical models and their interactions with increasingly complex computer hardware. Similarly, traditional numerical algorithms are at the core of scientific machine learning algorithms, so scientific machine *learning can be made more efficient, robust, and scalable by* leveraging the extensive knowledge of the scientific computing community. Catalyzing the interaction and interplay of scientific computing and machine learning algorithms has the potential to improve the throughput of both, but progress will require developing new methods to quantify trade-offs and optimally manage the interplay between traditional and machine learning models and implementations.

Scientific computing traditionally has been dominated by complex, resource-intensive numerical simulations. However, the rise of data-driven scientific machine learning models and algorithms provides exciting new opportunities. Traditional scientific computing forward simulations often are referred to as "inner-loop" modeling (cf. "outer-loop" problems, such as sensitivity analysis and optimization). The combination of traditional scientific computing knowledge with machine learning based adaptivity and acceleration has the potential to increase the performance and throughput of inner-loop modeling. Conversely, to address truly "big data" data using high-performance computing resources, machine learning algorithms must be scalable and efficient. The scientific computing community has decades of expertise involving numerical algorithms, especially dealing with the challenges of parallel computing, that can benefit the inner loop of training in machine learning. Therefore, an opportunity exists to advance machine learning, particularly at scale, by entraining more involvement from the computational mathematics community.

The overarching objective is to use machine learning to improve the performance and throughput of numerical simulations through machine learning enabled adaptivity. Success will result in a reduced need for human intervention and specialized expert knowledge to produce forward simulations efficiently and accurately. The ultimate benefit will be the ability to produce intelligent simulation capabilities that automatically provide solutions robustly with guaranteed accuracy and in the least amount of time—all within the user's prescribed constraints.

Generically, the "inner loop" refers to the computations within each time step of a simulation and/or within each iteration in an iterative solver. In the typical inner loop of a forward simulation, one or more numerical algorithms are used to advance or converge the solution of the discrete model. Typically, there are many choices for the numerical analyst: discretisations, linear solvers, nonlinear solvers, eigensolvers, interpolations, preconditioners, different models, etc. Even for a specific solver, for instance, there are additional choices, such as relaxation parameters, smoothing operators, and/or the number of Krylov subspace vectors. Expert knowledge is the primary tool by which these choices are made today, but the choices often are made and fixed at the start of the simulation. They do not change with the evolving behavior of the numerical solution. Furthermore, as demonstrated by adaptive step size control in ordinary differential equation integrators, adaptive mesh refinement, and shock-capturing schemes, there can be great advantages to adapting the solution process to the solution, even as it is being computed. Adaptivity can present complex choices between method order and discretization or matching solution processes. Much of this adaptivity is driven by heuristics and simplified analysis, which still require a great deal of human experience and evaluation. There is a need for research to realize the opportunity that scientific machine learning presents in enabling even greater adaptivity, which will help address the growing complexity of applications and architectures; increase performance in terms of fidelity, robustness, and/or time to solution; and increase throughput by reducing the need for human guidance or intervention in the simulation workflow.

The possibilities for advances in machine learning enhanced modeling and simulation are at least as numerous as the multitude of algorithms used in scientific computing. Such advances will benefit from research into adaptive numerical algorithms and adaptive implementation of numerical algorithms.

The scientific computing community has decades of experience in scalable numerical algorithms that can benefit machine learning. Scientific machine learning using high performance computing resources will require machine learning algorithms that are scalable and efficient. The inner loop of the scientific machine learning training process involves mathematical optimization algorithms and the linear algebra solvers used within these optimization techniques. A key consideration is that the numerical problems presented by machine learning will, in general, have different structures than the traditional partial differential equation based problems, which affords an opportunity to develop novel solver techniques specifically designed for machine learning. The performance of an optimization algorithm used in training is a core design consideration in machine learning. As learning models (and their training) have become more diverse and complex, algorithms have been adopted from diverse areas, such as convex optimization, non-smooth optimization, robust optimization, semi-definite programming, stochastic optimization, derivative-free optimization, and global optimization. A majority of the machine learning frameworks deployed today use some variant of the stochastic gradient descent method for this purpose. Many of these algorithmic variants have been rigorously analyzed and differ based on their parallelism and access to training data, including cases where very few passes through the data are allowed or where the data are distributed in such a way that synchronous access is infeasible. Significant research has been performed by the machine learning and optimization communities, leading to improved mini- and multi-batch as well as asynchronous algorithms. At the same time, accelerated and momentum-based techniques have been studied to reduce the number of iterations required by an optimization algorithm. Advances in non-convex optimization methods also have benefited machine learning in terms of the ability to incorporate loss functions and training objectives with favorable learning and regression properties. Examples include methods for variance reduction and globalization techniques, such as trust-region methods. Depending on problem size and solution requirements, first-order, second-order, and secant methods have been successfully employed in machine learning.

Despite ongoing research, much work remains to develop numerical algorithms that improve the speed and efficiency of machine learning training, particularly at large scale. A closer collaboration between the numerical algorithms and machine learning communities could prove to be a fruitful research area that advances the capabilities of machine learning with high performance computing. There is a research need for scalable and efficient machine learning on high performance computing resources by leveraging the expertise of the scientific computing community to develop appropriate scalable solver algorithms. Success in this research will lead to faster, more robust training of learned models, particularly at large scale, that will allow training over larger data sets and enable in situ training in scientific applications.

The scientific computing community has a well-established history of developing advanced, scalable solvers. As such, there is an obvious opportunity to leverage the existing knowledge from high-performance scientific computing to address scientific machine learning method costs and achieve scalability. The close collaboration on optimization techniques for machine learning is merely the beginning. Other techniques from computational mathematics; e.g., multilevel solvers, multifidelity solvers, floating-point compression, and domain decomposition techniques, also may contribute to improved machine learning training performance. Finally, an opportunity exists for the codesign of new and adaptation of existing scientific machine learning algorithms for different computer architectures.







THE NATIONAL INITIATIVE 'AI AND MATHEMATICS' (AIM)

THE IMPORTANCE OF MATHEMATICS FOR DUTCH AI RESEARCH

Al has changed our lives in the past 10 years, and it will change our lives much more in the years to come - Al is seen as the most important technology for decades to come. Al is a pre-eminently multidisciplinary technology, connecting scientists from a wide variety of research fields, from behavioural science and ethics to mathematics and computer science. Without wanting to undermine the importance of this variety, we want to emphasise the contribution of mathematics to Dutch AI research. With a clear vision of where mathematics can contribute to AI, we want to facilitate mathematicians to engage in a dialogue with research partners from other fields of science.

Mathematical AI in the Netherlands is very successful. Several themes in which mathematics plays a major role are mentioned in recent AI research agendas and investigated in large research programmes (responsibility, transparancy, generalizability, explainability). We fully endorse the importance of these themes and want to contribute to consortia that address these research challenges. Besides research in multidisciplinary consortia, fundamental mathematical research remains important in AI, as a basis for new developments in the longer term. In the early days of AI, the emphasis was on theoretical computer science and logic, but now many more types of mathematics are fundamental: mathematical statistics, game theory, graph theory and dynamical systems, among others. Often the greatest challenges

and opportunities lie at the intersection of different areas, and these intersections are often virgin territory in mathematics as well. Excellent mathematical research can and must play a major role in the Dutch AI research of the future.

We illustrate this with three important roles of mathematics in AI: as a basis for the design of AI methods, as a basis for analysis and understanding of AI methods and directly in the application of AI methods.

Mathematics and the design of AI methods

Today, successful AI is often thought of as deep learning, learning from large amounts of data using huge neural networks, with successful applications in speech and image recognition. Deep learning arose from brilliant intuitions and grew through intensive trial-and-error engineering and the use of vast amounts of data; mathematics played a modest role until recently. This may have obscured the fact that mathematical concepts and methods form the basis of just about all other successful methods and innovations in AI: support vector machines and kernel methods (state-of-the-art methods for machine learning when the amount of available data is limited), boosting, Bayesian learning, causal learning and reasoning, graphical models (crucial for explainable AI) are all equally successful and mathematical in nature. Very recent developments in the important area of machine learning and privacy also lean heavily on mathematics, where the concept of differential privacy was introduced as early as 2011.

Indispensable mathematical topics at the heart of all these methods are probability, statistics, optimisation and approximation theory and logic. But very different forms of mathematics also play a role: for example, we see mathematics in the form of differential geometry and PDEs recurring in the design of new, advanced machine learning methods. Directed graphs are indispensable in research on reasoning, argumentation and uncertainty. The study of logic, game theory and dynamical systems is useful for the design of knowledge, social procedures and interaction.

Geometric deep learning

This field focuses on applying geometric structure to learning systems in order to improve or enable new data structures. For example, groups and representation theory can be used to exploit symmetries in data and problems that promote generalisation and reduce the search space for useful functions (deep networks).

Bekkers, E. J. (2019). B-Spline CNNs on Lie groups. International Conference on Learning Representations.

Mathematics and the analysis of AI methods

Major applications of deep learning include speech recognition (Siri, Alexa, Google Assistant), machine translation and image recognition - with AI in medical image recognition sometimes outperforming human experts. But there is something strange going on: major AI methods produce 'black box' algorithms that cannot be explained in domain-relevant concepts. They are based on extremely large neural networks that are found through a labour-intensive process: all kinds of networks and parameter values are tried out, and in applications where very large amounts of data are available, one eventually gets something that works very well. But the system has no answer to the obvious and important question "why"? There are paradoxical examples where you change some pixels in a picture of, say, a dog just a little bit (a human cannot see the difference), but the computer suddenly thinks it is a cat - and we do not understand why the computer changes its mind. Without a fundamental understanding of these issues, the applicability of AI remains limited - in applications where, for example, little data is available, where an estimation of reliability is essential ('uncertainty quantification') or where data is heavily biased, truly successful AI applications are often still lacking. Mathematics is an excellent tool for studying and understanding how AI methods work. Especially with methods from analysis, statistics and logic, it is possible to gain insight into fundamental possibilities and limitations, theoretical performance guarantees, optimality and uncertainty quantification, explainability, accountability and social behaviour. It is precisely through the interaction between different forms of mathematics and their application that progress can be achieved.

For several years now, mathematical research into "why does deep learning work so well? has been gaining momentum. There is also increasing insight into the fundamental limitations, for example for explainable algorithms and responsible behaviour of AI systems. With its strong tradition in analysis, mathematical statistics and logic, The Netherlands can play a major role here.

Another important question: why do successful AI systems sometimes use such enormous amounts of energy compared to the human brain. Fundamental theoretical questions include under what circumstances algorithms are efficient or not, and whether a continuous flow of data requires alternatives to digital computing (e.g. neuromorphic computing).

Bayesian networks for explainability

Bayesian networks (BNs) use directed graphs to describe probability distributions compactly and intuitively. The NWO Forensic Science research programme, for example, investigated how and to what extent BNs can be designed and explained for the analysis of murder cases. An important question for AI is how BNs can be used to learn comprehensible causal structures from data. This is a field in which essential contributions are made from many different disciplines, and in which the underlying mathematics is the common denominator that creates synergy and helps to avoid speech confusion.

Mooij, J.M., Magliacane, S., Claassen, T. (2020). Joint Causal Inference from Multiple Contexts. Journal of Machine Learning Research 21(99):1-108

Why does Deep Learning work?

Using techniques from statistics and approximation theory, it is possible to better understand in which situations Deep Learning works and what the optimal architectures of deep networks are. It turns out that, despite the huge number of parameters, the 'effective complexity' of neural networks is often limited, which explains why there is not too much 'overfitting'.

Schmidt-Hieber (2020). Nonparametric regression using deep neural networks with ReLU activation function. Annals of Statistics, to appear.

Mathematics and the application of AI methods

In applied mathematics, there is a long tradition of transferring mathematical methods to concrete applications. AI methods are increasingly used in this context, in a variety of applications. They often combine ideas and methods from several areas. One example is causal reasoning, often based on Bayesian networks. The theory was developed in mathematical AI for the correct modelling of reasoning with uncertainty and later proved applicable in many other areas. For example, causal reasoning is indispensable for efficient mathematical models for genetic data and is intensively used in research into new medical treatments.

AIM for the best

In the coming years, a lot of investments will be made in the development and application of AI in the Netherlands. Mathematical AI research can play a major role in many applications, and is essential for laying a solid foundation for the future of AI.

The AIM network

The AIM network unites Dutch research groups that conduct AI research in which mathematics plays a major role. Participating organisations with their representatives:

- Peter Grünwald, (CWI)
- Leo van Iersel (TU Delft)
- Etienne de Klerk (Universiteit Tilburg)
- Johannes Schmidt-Hieber (Universiteit Leiden)
- Christoph Brune (Universiteit Twente)
- Eric Cator (Radboud Universiteit Nijmegen)
- Sjoerd Dirksen (Universiteit Utrecht)
- Joris Mooij (Universiteit van Amsterdam)
- Wil Schilders (TU Eindhoven)
- Bart Verheij (Rijksuniversiteit Groningen)
- Harry van Zanten (VU Amsterdam)
- Steven Kelk (Maastricht University)
- Ilker Birbil (Erasmus University Rotterdam)





MSODE: MODELLING, SIMULATION AND OPTIMIZATION IN A DATA-RICH ENVIRONMENT

WIL SCHILDERS

Abstract Modelling, Simulation & Optimization (MSO) remain the cornerstone for the development of most products in the fields of industry, health, energy or even finance. Although High Performance Computing, Data Analytics and Artificial Intelligence offer new opportunities, their impact on innovation and the improvement of products and services could remain partial without a massive effort on the axes of modelling, simulation and complex systems optimization. Major opportunities, in particular, the establishment of digital twins, rely on the connections at the interface between fields of expertise, domains, businesses and across the complete lifecycle of products and systems. At the same time, methods have outpaced computational power in terms of capability over the past decades. The MSODE initiative outlined in this article is guided by the certainty that a high level approach on Modelling, Simulation & Optimization, enriched by data analytics and intensive computing, is a considerable economic asset.

Vision

On the one hand, the future development of industry and society exhibits strongly increasing complexity and at the same time ever-shorter innovation cycles. On the other hand, digitisation and the internet of things have led to an explosion of data and information. Without novel computational tools and paradigms we will not be able to manage these challenges. There is a clear need to strengthen European competitive advantage in industrial innovations and to start a new initiative to meet the associated societal challenges ahead of us.

For almost all domains of science and engineering and in almost all industrial sectors, model-based approaches are well established. A multitude of commercial and open source software for modelling, simulation and optimization (MSO) based on mathematical models ("mathware") is available. At the same time increasingly large amounts of process and product data are available and strong artificial intelligence solutions have been developed to exploit these. All this is fostered by computers becoming more and more powerful.



These developments lead to the vision that in the near future holistic approaches can be achieved that combine all these developments. A complete industrial product or process in its whole life cycle can be accompanied by a virtual representation, often called digital twin that allows design optimization, process control, lifecycle management, predictive maintenance, risk analysis and many other features. Digital twins are so important to business today, that they were named one of Gartner's Top 10 Strategic Technology Trends for 2017 [1], as well as in subsequent editions. They are becoming a business imperative, covering the entire lifecycle of an asset or process and forming the foundation for connected products and services. Companies that fail to respond will be left behind.

"Mathematics is the language of digital twins!"

To establish this vision or to even come close to it, several new developments that involve different scientific communities have to take place and many obstacles have to be removed. A core need are novel mathematical technologies, to describe, to structure, to integrate and to interpret across disciplines. Mathematics is the language of digital twins!

History

NASA was the first to dabble with pairing technology – the precursor to today's digital twin – as far back as the early days of space exploration. How do you operate, maintain, or repair systems when you aren't within physical proximity to them? That was the challenge NASA's research department had to face when developing systems that would travel beyond the ability to see or monitor physically.

Figure 1. Concept of a digital twin (picture taken from [2])



Michael Grieves at the University of Michigan first wrote of the concept using the digital twin terminology in 2002 [2]. The digital twin serves as a bridge between the physical and digital world. The components are connected to a cloud-based system or a dedicated hardware that uses sensors to gather data about real-time status and working condition. This input is analysed against business and other contextual data. Lessons are learned and opportunities are uncovered within the virtual environment that can be applied to the physical world.

Digital twins are powerful masterminds to drive innovation and performance. It is predicted that companies who invest in digital twin technology will see a 30 percent improvement in cycle times of critical processes.

State-of-the-art

Although first successes are reported [3] and many claims are made [4], neither the classical MSO approaches based on mathematical models and their software implementations, nor the constantly improving techniques for data analysis and machine learning will be enough to achieve this visionary goal [5,6]. Even the rapid improvements in modern computing hardware and especially algorithms/software are not sufficient to achieve this. Currently, due to the high manual human effort, only major companies with large R&D departments can afford to build digital twins, but it would be desirable that companies on all scales can profit from the development. New generations of mathematical paradigms are required to convey today's highly fragmented approaches in the various disciplines.

Figure 2. High level schematic of the MSODE initiative and its connection to other initiatives



Technology challenges

Currently, models, methods, as well as software implementations and data sets are of highly different fidelity requiring many manual interactions. Figure 2 summarizes these interactions in a schematic way (including the interactions with existing initiatives). To meet the future challenges, it is necessary to develop novel MSO paradigms allowing a systematic MSO based approach to build highly automated modularized networks of model hierarchies (from very high fidelity physics based models to very coarse, surrogate, or even purely data based models), and that can deal with multi-physics and multi-scale systems. Key will be a convergence of artificial intelligence methods and first principle approaches.

typically used in MSO by laying down novel mathematical principles as the core language of digital twins.

Furthermore, the model hierarchies should

- be able to (automatically) evolve with the availability of new information, data, or even changes in the process,
- allow adaptive models and solutions with seamless choice of accuracy and speed,
- allow real-time and interactive simulation and optimisation,
- be made robust towards inaccuracies in the data and the model,
- be able to quantify the uncertainties and risks that come with the determined solutions,
- lead to the convergence of artificial intelligence and physics-based models,
- exploitation new computing architectures, e.g. combined cloud - edge solutions,
- be flexible for new user interaction concepts,
- allow the use of advanced black box solvers MSO software packages.

In the following section, we will go into more detail concerning one of these bullet points, namely the convergence of artificial intelligence and physics-based models, as it leads to great opportunities and many mathematical challenges.

Combining physics-based and data-based models

There is a multitude of opportunities emerging from the combination of physics-based modelling/simulation and data-based machine learning techniques. This is the way forward. On the one hand, we must not ignore the vast amount of fundamental knowledge, built up in many centuries, and only rely on data. On the other hand, we should exploit the increasing availability of data (both measured and computed) and computational resources to analyse these. Hence, we should combine the best of both worlds, make use of the advantages of available techniques in both fields, and thereby creating a breakthrough in the field of modelling and simulation for many scientific and engineering disciplines.

The U.S.A. is a frontrunner in this emerging area. The Department of Energy (DOE) is always very keen on new developments and is one of the biggest sponsors of scientific research. The national laboratories also play a key role in the development of new and emerging technologies. In January 2018 a workshop was held, under the auspices of DOE, to identify basic research needs for the field of scientific machine learning. This information then was used to examine the opportunities, barriers, and potential for high scientific impact through fundamental advances in its mathematical, statistical, and computational research foundations. The workshop report [7], published early 2019, concludes that scientific machine learning and artificial intelligence will have broad use and transformative effects across all fields of science.

The report was followed by a sequence of town hall meetings organised by the Argonne, Oak Ridge, and Berkeley National Laboratories with more than 1000 scientists and engineers participating. The goal of the town hall series was to examine scientific opportunities in the areas of artificial intelligence, Big Data, and high-performance computing (HPC) in the next decade, and to capture the big ideas, grand challenges, and next steps to realising these opportunities. The terminology "AI for Science" was used to broadly represent the next generation of methods and scientific opportunities in computing, including the development and application of AI methods (e.g., machine learning, deep learning, statistical methods, data analytics, automated control, and related areas) to build models from data and to use these models alone or in conjunction with simulation and scalable computing to advance scientific research. An extensive report [8] with conclusions for many scientific and engineering disciplines, ranging from chemistry, materials and nanoscience, biology and life sciences, nuclear and high energy physics to engineering and manufacturing was published at the end of 2019. The main overall conclusion is that "AI will not magically address all opportunities and challenges discussed in the report. Much work will be required within all science disciplines, across science infrastructure, and in the theory, methods, software, and hardware that underpin AI methods. Bringing AI to any specific domain—whether it is nuclear physics or biology and life sciences will demand significant effort to incorporate domain knowledge into AI systems, quantify uncertainty, accuracy, and appropriately integrate these new mechanisms into state-of-the-art computational and laboratory systems."

Both reports agree to a large extent with our point of view: **exploiting prior information and knowledge to construct combinations of physics-based models and data-based learning machines**. One of the fore-runners in this field is George Karniadakis of Brown University [9,10,11], who is strongly promoting the field of, what he calls, physics-informed neural networks (PINNs), cf. Figure 3 below being an example of how to combine neural networks with the physics-informed model structure (PDE).

The general aim of Karniadakis' work is to set the foundations for a new paradigm in modelling and computation that enriches deep learning with the longstanding developments in mathematical physics, or vice versa depending on the angle of view. Methods developed utilise a fully connected neural network (NN) to map a space-time domain to the unknown solution of an initial- and boundary-value problem. The NN is inserted into the governing partial differential equation (PDE) or variational principle and symbolically differentiated. This yields another NN,

Figure 3: Physics-informed neural networks (origin: George Karniadakis)



with modified activation functions but identical parameters, a so-called physics-informed NN. No labelled data is required for training. Rather, a combined loss function is minimised. One component is associated with the initial and boundary conditions, the other either with the residual norm or the variational functional of the PDE. The latter component enforces the structure of the physics equation.

Concepts similar to that proposed by Karniadakis are being developed in several places, as more and more researchers are realising that the integration of machine learning and more general artificial intelligence technologies with physical modelling based on first principles will impact scientific computing in science and engineering in fundamental ways. One such development is to embed physics simulation into deep learning. This work is motivated by control engineering, in particular by the development of intelligent reinforcement learning agents. "*The end result is that we can embed an entire physical simulation environment as a layer in a deep neural network, enabling agents to both learn the parameters of the environments to match observed behaviour and improve control performance via traditional gradient based learning.*" [12]. The main ingredient is an adjoint-based solver, which allows efficient backpropagation of gradients and avoids their tedious computation by finite differences. Then, deep convolutional neural networks can be integrated seamlessly with physics-based models in machine learning platforms such as PyTorch and TensorFlow. Another development is in combining field inversion and machine learning (FIML [13]). This method stems from computational fluid dynamics (CFD).

For turbulent flows one may either solve Navier-Stokes equations by direct numerical simulation (DNS) or large eddy simulation (LES). This approach is accurate but numerically expensive, since it involves a range of space and time scales. On the other hand, one may use the Reynolds averaged Navier-Stokes (RANS) method, where turbulence effects are accounted for by phenomenological models rather than first principles. This method is much more efficient but less accurate. With the help of FIML, both approaches can be combined.

"The future needs Computational Science and Engineering, blending data driven and physicsbased perspectives"

Karen Willcox, director Oden Institute for Computational Engineering and Sciences

The foregoing clearly demonstrates that we are at a tipping point within scientific and engineering research: first principle-based models will need to be combined with data-based models. Such hybrid modelling combines first principle-based models with data-based models into a joint architecture and has the potential to improve the Pareto trade-off between simulation accuracy and simulation cost significantly, bringing scientific computing in science and engineering to the next level. Awareness of domain knowledge can enhance domain-agnostic data in terms of accuracy, interpretability, and robustness of models. Furthermore, incorporating scientific domain knowledge has the potential to dramatically reduce data requirements, as well as to accelerate training and prediction. Domain knowledge is found in many forms, such as physical principles, constraints, symmetries, conservation laws, and other knowledge gained from theoretical or computational studies.

Scientific domain knowledge can be expressed in many forms, including physical models (e.g., ab initio or first-principles physics), physical constraints (e.g., symmetries, invariances, conservation laws, asymptotic limits), computational simulations, uncertainties, correlations in space and time, and structural forms (e.g., discrete, graph-like, non-smooth data). Looking into more detail, we notice that a large class of models can be decomposed into conservation laws and constitutive laws. The conservation laws are of topological nature and can therefore be discretised easily, leaving little room for data-driven techniques. The situation is different for the constitutive relations, which are of metric nature, and encode phenomenological (material) properties.

Except for simple media (local, linear) there are many potential complications (non-local, hysteretic, non-linear, multi-scale, multi-physics, etc.). Here, data-driven models can be useful, provided that the models fulfil certain admissibility criteria, which can often be expressed in terms of invariance with respect to symmetry groups.

It is also clear that the developments of combining real intelligence with artificial intelligence towards hybrid modelling are still in their infancy. Only in the last few years, researchers have started to fully acknowledge the potential of combining first principlebased models with neural networks or other artificial intelligence techniques. The number of publications in the field is growing exponentially (see, for example, Gartner's assessment [1]). In the next sections, we will detail the research directions we aim to pursue, and which we expect to provide significant, substantial and ground-breaking contributions to this exciting emerging field of research within science and engineering. Key high-level challenges, still open and hence to be addressed, are hybrid modelling methodologies that:

- are applicable across a wide range of scientific domains,
- can deal with large-scale and networked systems,
- preserve fundamental system properties, such as stability, structure, dissipativity, etc.,
- strike a superior balance between accuracy and complexity of the resulting models,
- guarantee robustness of model validity, while limiting the amount of measured data needed.

Conclusion

We strongly believe that mathematicians should focus their attention on creating novel synergies between physics-based and data-driven approaches, in order to develop a next-generation modelling framework for physical processes and engineering systems. To this end, a detailed mathematical theory needs to be developed to support such synergetic marriage leading to models within science and engineering with superior model accuracy, computational efficiency, robustness to uncertainties and explainability. This field will be full of very nice challenges, and it is hoped that many mathematicians will take up these challenges. In our project UNRAVEL (''Unraveling neural networks with structure-preserving computing'' [14]), the challenge is taken up with the intention to transfer our knowledge about so-called mimetic methods (mimicking the behaviour of the problem to be solved).

Together with industry, this will also lead to true digital twins, needed to bring Europe and its industry forward. This is also expressed in the booklet displayed below, which was made by the European Service Network of Mathematics for Industry and Innovation (EU-MATHS-IN), together with its Industrial Core Committee that has Siemens, Bosch, Shell, Michelin and several other European companies on board.



References

- Gartner: Top 10 Strategic Technology Trends for 2018 https://www.gartner.com/smarterwithgartner/gartner-top-10-strategictechnology-trends-for-2018/
- [2] http://innovate.fit.edu/plm/documents/doc_mgr/912/1411.0_Digital_ Twin_White_Paper_Dr_Grieves.pdf
- [3] GE: https://www.ge.com/reports/scientists-built-a-digital-twinof-a-carbattery-to-make-it-last-longer/
- [4] ANSYS: http://www.ansys.com/products/releasehighlights/digital-twin
- [5] ORACLE: Digital Twins for IoT Applications http://www.oracle.com/us/solutions/internetofthings/digitaltwins-foriot-apps-wp-3491953.pdf
- [6] Engineering.com: Mission Impossible: The Digital Twin: https://www.engineering.com/DesignSoftware/DesignSoftwareArticles/ ArticleID/15919/Mission-Impossible-The-Digital-Twin.aspx
- [7] N. Baker at al, "Basic research needs for Scientific Machine Learning Core technologies for Artificial Intelligence", Prepared for Department of Energy Advanced Scientific Computing Research (2019)
- [8] R. Stevens, V. Taylor, J. Nichols, A.B. Maccabe, K. Yelick, and D. Brown, "Al for Science", Department of Energy, Office of Science (2020)
- [9] M. Raissi, P. Perdikaris, and G.E. Karniadakis, "Inferring solutions of differential equations using noisy multi-fidelity data", J. Comp. Phys., 335, 736–746 (2017)
- [10] M. Raissi, P. Perdikaris, and G.E. Karniadakis, "Machine learning of linear differential equations using Gaussian processes", J. Comp. Phys., 348, 683–693 (2017)
- [11] M.Raissi, P.Perdikaris, and G.E.Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations", J. Comp. Phys., vol. 378, pp. 686-707 (2019)
- [12] F. de Avila Belbute-Peres, K. Smith, K. Allen, J. Tenenbaum, and J.Z. Kolter, "End-to-end differentiable physics for learning and control", In: Advances in Neural Information Processing Systems, pp. 7178–7189 (2018)
- [13] E.J. Parish and K. Duraisamy, "A paradigm for data-driven predictive modeling using field inversion and machine learning", J. Comp. Phys. 305, 758–774 (2016).
- [14] https://nwo-unravel.nl/

RANDOM PROCESSES IN MACHINE LEARNING

JORIS BIERKENS

What is a random process?

Imagine you have a single coin and some time to spend. What you might do is start flipping the coin, counting the number of heads and subtracting the number of tails. Whenever head comes up, you add one to the number you are keeping in mind, and when tail comes up, you subtract one.

Let us give a name to the number we find after n throws: We will call this number Sn. In mathematical terminology, the sequence of numbers S_0 , S_1 , S_2 , ... is called a *random walk*. It is an example of what mathematicians call a *random process* or *stochastic process*¹.

The field of probability theory concerns itself with analyzing such random processes. For example if you would repeat this experiment for a very long time (and rescale the time and coordinate axes), the process behaves increasingly as a Brownian motion $X_{(t)}$.

A Brownian motion is a random process that evolves according to a particular probabilistic law, continuously in time and space. It was named after the botanist Robert Brown, who first described the phenomenon in 1827, while looking through a microscope at pollen of the plant Clarkia Pulchella immersed in water.





(b) Brownian motion X(t)

Fig. 1.2: A random walk and its limiting behaviour (a Brownian motion) in two dimensions



(a) Two-dimensional random walk



(b) Two-dimensional Brownian motion

We can also throw two coins, and keep track of two positions at the same time. This would give us two sequence of numbers. If we plot these in a two-dimensional plot, this could look like Figure 1.2(a), with as a limiting process a two-dimensional Brownian motion $X_{(t)} = (X_{1(t)}, X_{2(t)})$ (here $X_{(t)}$ denotes the first coordinate of the process at time t). And nothing stops us from using as many coins as we like, thus constructing a random walk in any dimension.

The important observation is that a random walk, or its continuous counterpart, a Brownian motion, describes a random exploration of a multidimensional space. This makes these processes so useful in machine learning as I will describe in the subsequent sections.

Optimization in machine learning

Suppose we are training a machine to carry out a certain task. A well known example is training a machine to recognize hand-written numbers; see Figure 1.3. In this case the machine learning task would be, for a given input of pixel values of a handwritten number, to find the corresponding numeric value $0, 1, \ldots, 9$.

Often we assume that we can give a *risk* score *R* to the performance of the machine: if the machine produces all of its outputs correctly, the value of *R* will be small, if it provides wrong outputs, the value of *R* is large. We will not go into the details here on how such a risk score is exactly defined or computed.

Furthermore we should imagine that we can tune the machine in great detail: within the machine there are all kinds of adjustments we can make that specify how it carries out its task. One knob x_1 may affect the importance of the first pixel, another one x_2 the importance of the second pixel. There may be a set



(c) Within the machine there are various settings we may adjust, to affect its performance.

(a) An extract from the famous MNIST database that is commonly used for training image processing systems.

(b) A machine processing its input to produce a desired out- put. For example, the machine might have as its input the pixel values of a handwrit- ten number, and produce as its output one of the numbers 0, 1, 2,..., 9.

ting which determines the importance of the average value of the pixels, etc. We want to find the best possible adjustment, i.e. find a value for the settings that minimize the risk. This means that we can think of the risk as a function

$$(x_1, x_2, \ldots, x_d) \rightarrow R(x_1, x_2, \ldots, x_d)$$

and we can think of machine learning as an *optimization problem*: that is to find the optimal value (x_1^*, \ldots, x_d^*) of all the settings, so that the risk $R(x_1^*, \ldots, x_d^*)$ is the minimal possible risk, corresponding to the best possible performance of the machine. We call (x_1^*, \ldots, x_d^*) the global minimum.

Stochastic algorithms

As explained above, in machine learning or statistics we often wish to minimize a *risk function* R that describes how far away a typical prediction generated by a 'machine' is from an actual observation. The inputs (x_1, \ldots, x_d) are then interpreted as parameters of the machine which we can tune to make the best possible predictions, and $R(x_1, ..., x_d)$ is the risk corresponding to this choice of parameters. Suppose for simplicity that all parameters $x_1, ..., x_d$ can assume values between 1 and 10 (just like the knobs displayed in Figure 1.3(c).

A straightforward way to find the best possible settings would be to *discretize* the knobs, and consider for each knob e.g. just the possible values 1, 2, ..., 10. We could then simply try *all* possible values for *each* x_i , evaluate the corresponding risk $R(x_1, \ldots, x_d)$, and choose the optimal value x_1, \ldots, x_d . This means that we have to process

$$\underbrace{10 \times 10 \times \dots \times 10 = 10^d}_{d} \qquad \text{possible settings}$$

For larger values of *d*, very soon this becomes a serious computational challenge or even impossible to process completely. As an example, in machine learning problems for self driving

cars, there are often thousands of parameters to control. Already for d = 50 it would take the best supercomputer in the world² longer than the life time of the universe to perform this computation.

So we have to be more clever and randomness will be an important ingredient. Suppose we let the coordinates x_1, \ldots, x_d evolve as a random walk. It is known from probability theory that eventually the random walk will visit all possible (10^d) discrete settings; unfortunately this takes extremely long.

A better idea is to guide the random using the risk function, by iterating the following procedure, starting from a given position (x_1, \ldots, x_d) . We introduce a parameter β called inverse *temperature*³, where you can think for now of $\beta = 1$.

- 1. Randomly increase or decrease one coordinate x_i by ± 1 (or any other stepsize).
- If the risk R(x₁, ..., x_d) becomes smaller due to the proposed change of the coordinate, always accept the proposed change: the proposed coordinates become the new coordinates.
- 3. If the risk $R(x_1, \ldots, x_d)$ becomes larger due to the proposed change of the coordinate, *accept* the proposed change with probability $\exp(-\beta[R_{(x_{new})} R_{(x_{old})}])$ (a number smaller than one); otherwise *reject* the proposed change and leave (x_1, \ldots, x_d) as it was before.

If you think about this algorithm, it will favor transitions to coordinates that have a smaller risk, which is something we like to see. But it also allows (with a certain probability) transitions to states that seem disadvantageous. Why would we want to do that? The main reason for allowing transitions to coordinates with larger risk is that this allows to escape *local minima*. Fig. 1.4: A risk function with a global minimum and a local minimum. A small probability to move in directions of a larger risk, allows to escape a local minimum and eventually reach the global minimum that we are aiming for.



The algorithm described above is called the Metropolis (-Hastings) algorithm [2]. It is an example of a Markov Chain Monte Carlo (MCMC) algorithm. I have introduced it here as an optimization algorithm, but you can also think of it as an exploration algorithm. In fact, what the algorithm really does, is that it explores the target probability distribution $p(x_1, \ldots, x_d) = exp(-\beta R(x_1, \ldots, x_d))/K$ (where K is a normalization constant)⁴. This means that the trajectory spends, in the long run, an amount of time at locations (x_1, \ldots, x_d) proportional to the value of $exp(-\beta R(x_1, \ldots, x_d))$. You also see that by adapting β , you can choose how strong the influence of the risk function is: for $\beta = 0$, the probability distribution is flat, whereas for a large value of β the probability distribution has (almost) just a single peak at the global minimum of R. Algorithms that gradually increase β in an MCMC algorithm are known as simulated annealing methods: they gradually diminish the amount of exploration, and concentrate more and more around the global minimum.
Fig. 1.5: The Metropolis algorithm applied to a 2D spherical distribution. The figures are plots of the trajectory of the Markov chain along with a contour plot of the target distribution. We see an example of very small step size and very coarse step size (both resulting in slow exploration), as well as an optimally tuned stepsize with efficient exploration.



Research in Markov Chain Monte Carlo algorithms

You may think that we are done now that we have learned about the Metropolis-Hastings algorithm, but to this day there are many important problems remaining. One of these is that the efficiency of the exploration crucially depends on the type of proposal we are making. For example, if we propose very small steps, then we are quite likely to accept our proposals⁵, but the exploration will be slow. On the other hand, if we propose very big steps, then we may well reject most of the proposals because the acceptance probability becomes too small. We want to propose moves that have the optimal size: Remarkably it can be shown theoretically that in this case an average acceptance probability of 0.234 is ideal! See Figure 1.5 for an example.

This is just an example of the type of questions you run into. There are many ways to design Markov chains which have the correct target distribution, and we wish to find the best one. In my own research I am using a recently discovered type of Markov processes: so called piecewise deterministic Markov processes [1]. These travel for a random amount of time in a fixed direction, until changing direction, see Figure 1.6 for an example. Fig. 1.6: An example of a piecewise deterministic Markov process is the Zig-Zag process which moves in directions within the discrete set 1, +1 d, where d is the dimension. In this illustration d = 2.



References

- 1 From the Greek 'στοχαστικός' ('stokhastikos'): to guess
- 2 The European LUMI computer announced in 2020, operating at 5×10¹⁷ FLOPS. Here FLOPS denotes 'Floating point Operations Per Second' and is a measure of the computational strength of a computer.
- 3 Terminology arising from the field of statistical physics.
- 4 Here $exp(a) = e^a$.
- 5 Since the value of R will not change very much for a new proposal and therefore the acceptance probability will be close to one.
- J. Bierkens, P. Fearnhead, and G. O. Roberts. The Zig-Zag Process and Super- Efficient Sampling for Bayesian Analysis of Big Data. Annals of Statistics, 47(3):1288–1320, 2019.
- [2] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. The Journal of Chemical Physics, 21(6):1087, 1953.

MACHINE LEARNING FOR MULTISCALE SIMULATIONS IN CLIMATE

DAAN CROMMELIN AND WOUTER EDELING, CENTRUM WISKUNDE & INFORMATICA, AMSTERDAM

Abstract Modeling and simulation of multiscale systems such as the climate system is highly challenging, as straightforward simulation including all system details is computationally infeasible. Machine learning of so-called parameterizations that represent effects of small-scale system features has emerged as a promising way forward. We briefly describe several open mathematical problems that need to be resolved to develop this approach into a mature and robust methodology.

Multiscale systems: a challenge for simulation

Multiscale modeling and simulation is an outstanding challenge in many areas of science and engineering. The necessity to include many small-scale details in computations, in order to get the large-scale behaviour right, forms a major obstacle for studying phenomena and system behavior in fluid flow, materials science, molecular dynamics, and more. An example of a multiscale system that is especially urgent and relevant in the current era is the climate system. Among climate scientists, it is well-known that the fine details of e.g. ocean flow, cloud formation and precipitation microphysics have significant impact on the global patterns and long-term variations of the climate system. Including all these small-scale details in computer simulations is computationally very expensive. Even with the computational resources that are nowadays available, such simulations are often impossible. It will be many decades before we may be able to run climate models in which every individual cloud is accurately resolved, over climate-change timescales.

Because of these computational limitations, instead of simulating all details, their effects on large-scale system features are included via simplified representations, usually referred to as "parameterizations" (or "closures"). Constructing such parameterizations is difficult, as in many instances it is unknown how to derive them systematically from first principles.

Learning of parameterizations

Learning parameterizations from data has emerged as a promising alternative, with a diverse set of methodologies ranging from data-inferred Markov chains [4] to deep neural networks [8]. The training data needed for learning can come from highly detailed model simulations over a small spatial domain or a limited time interval (to make these detailed simulations feasible), or from observations. We note that although the focus in this chapter is on climate science applications, the potential of data-driven methods for parameterization is recognized and explored in other domains as well, see e.g.[7]. The purpose of a parameterization learned from data is to capture the feedback from small-scale details on large-scale system behaviour. Thus, it does not model the entire climate system, but only this feedback. The parameterization must be combined with a physics-based model for large-scale system features to get a model of the entire system. In the context of climate modeling and fluid dynamics, the largescale model includes the partial differential equation (PDE) that describes fluid flow (Navier-Stokes equation). Typically, this PDE is solved on a spatial grid, and at each grid node the PDE-based equations are augmented by parameterizations to represent various small-scale feedbacks (as illustrated in Figure 2.1). With parameterizations obtained by machine learning (ML), the result is a hybrid PDE-ML model.

A hybrid PDE-ML model

With both the physics-based PDE and the training data given, setting up such a hybrid PDE-ML model may seem straightforward: train the ML-based parameterization using the available data, take an existing numerical solver for the PDE, and couple the two. However, it involves several unresolved mathematical challenges. One is that the parameterization, although often applied gridpoint-wise, is in fact representing a spatio-temporal field. Capturing the correlations (and even memory) in time and space of this field is difficult, especially when taking into consideration that there is a two-way coupling between the PDE and the parameterization.

When training the parameterization in an off-line manner, not taking into account the coupling to the PDE at the next stage, there is no guarantee that the coupled PDE-ML system will behave like the "true" system (the hypothetical computational model in which all details are included in the computations) even if the off-line training result is accurate. Even worse, the coupled PDE-ML system can be numerically unstable, see e.g. the discussion on neural-network based parameterization and stability problems in [2]. Getting a better handle on these stability and accuracy issues is an open problem, one that has to be resolved to be able to build robust methods.

Fig. 2.1: In a climate model, the equations that describe atmospheric flow are typically solved on a spatial grid that spans the earth. Detailed features, such as individual clouds, are too small to be resolved on this grid. Instead, their feedback on large-scale features is represented by parameterizations in each grid box. These parameterizations can be learned from data, resulting in (e.g.) a neural network that is evaluated at each grid box and at each model time step.



Model uncertainty

The use of parameterizations entails loss of information of many small-scale system features, as only their effect on largerscale features is represented. This leads, inevitably, to model uncertainty. An important research line in weather and climate modeling has been to develop probabilistic methods to account for this uncertainty, building stochastic rather than deterministic parameterizations (see e.g. [1, 6]). In e.g. [4] a data-based method to this end was proposed. However, how to deploy modern deep learning techniques for purposes of stochastic parameterization has hardly been investigated yet. Initial explorations in this direction can be found in e.g. [3, 5]. Putting the methods proposed in these studies to effective use in complex, high-dimensional models (in climate science or elsewhere) has not been done yet. On the theoretical side, the mathematical underpinning of these methods needs to be further developed.

Conclusion

Parameterization of small-scale features is highly relevant for simulating the climate system and other multiscale systems. Learning parameterizations from data is an approach that holds great promise, however several mathematical challenges need to be overcome to develop it into a mature and robust methodology. These include dealing with spatio-temporal structures, resolving issues of stability and accuracy, and accounting for model uncertainty. Combining research forces from mathematics and AI will be crucial to make major steps forward on this urgent scientific problem.

References

- J Berner et al. Stochastic parameterization: Toward a new view of weather and climate models. Bull. Amer. Meteor. Soc., 98(3):565–588, 2017.
- [2] ND Brenowitz et al. Interpreting and stabilizing machine-learning parametriza- tions of convection. J. Atmos. Sci., 77(12):4357–4375, 2020.
- [3] D Crommelin and W Edeling. Resampling with neural networks for stochastic parameterization in multiscale systems. Physica D, 422:132894, 2021.
- [4] D Crommelin and E Vanden-Eijnden. Subgrid-scale parameterization with conditional Markov chains. J. Atmos. Sci., 65(8):2661–2675, 2008.
- [5] DJ Gagne et al. Machine learning for stochastic parameterization: Generative adversarial networks in the Lorenz'96 model. J. Adv. Model. Earth Syst., 12(3):e2019MS001896, 2020.
- [6] TN Palmer et al. Representing model uncertainty in weather and climate pre- diction. Annu. Rev. Earth Planet. Sci., 33:163–193, 2005.
- [7] S Pan and K Duraisamy. Data-driven discovery of closure models. SIAM J. Appl. Dyn. Syst., 17(4):2381–2413, 2018.
- [8] S Rasp et al. Deep learning to represent subgrid processes in climate models. PNAS, 115(39):9684–9689, 2018.



FRUITFULLY COMBINING LINEAR ALGEBRA, STATISTICS, OPTIMIZATION, AND DATA SCIENCE

GIULIA FERRANDI, PERFECT GIDISU, SIBYLLE HESS, AND MICHIEL HOCHSTENBACH

Abstract We give a very brief impression of some current hot and intriguing problems in the very active intersection of the fields of large-scale scientific computing, statistics, data science, linear algebra, and numerical optimization (cf., e.g., [10]). There are countless connections between these fields, which result in elegant and powerful mathematical techniques, and interpretable and useful results.

Feature extraction

Large data matrices arise in many fields and applications, such as data mining, medical and financial applications, (social) networks, bio-informatics, AI, etc. Important tasks include finding useful patterns, features, or trends from the data, or reducing the data dimension. Key techniques to attain useful results include low-rank matrix approximations, matrix decompositions, and suitable projections.

Suppose $A \in \mathbb{R}^{m \times n}$ is a (data) matrix, usually of large dimensions, with the data points as columns (or rows). Often one is interested in extracting low-dimensional, relevant and interpretable information from the data. Since in practice data often tends to be (approximately) of low rank, the first classical problem is to consider, for a modest $k \ll min(m, n)$,

$$\min_{\operatorname{rank}(\mathbf{A}_k) = k} || \mathbf{A} - \mathbf{A}_k ||,$$
(3.1)

where the norm usually is the Frobenius norm. The wellknown solution to this problem is the *truncated singular value decomposition* (TSVD), involving left and right singular vectors. To be precise, if $A = U\Sigma V^T$ is the SVD, then $A_k = U_k \Sigma_k V_k^T$, where the diagonal matrix Σ_k contains the *k* largest singular values, and the columns of U_k and V_k are the corresponding left and right singular vectors. In statistics, the V_k -vectors (of centered data) are called *principal components*, which are the linear combination of features which maximize the variance.

However, the solution to (3.1) usually has no clear interpretation in terms of the original data points. Moreover, the singular vectors generally do not preserve properties of the data as sparsity and positiveness. To address this, several alternatives have been proposed. An *interpolative decomposition* (ID; see, e.g., [3]) is of the form $A \approx CZ$, where $C \in \mathbb{R}^{m \times n}$ is a representative selection of the columns (either data points or variables), and Z is often constructed to minimize || A - CZ ||. Since the selection of C tends to be an (NP) hard problem, several approaches based on greedy selection or the SVD have been proposed. An example of the appreciable difference can be seen in Figure 3.1 (left). Similarly, a row-oriented ID decomposition is of the form $A \approx ZR$, where R is a relevant subset of the rows. A *CUR decomposition* $A \approx CMR$ is a two-sided variant of the ID decomposition, which contains subsets of both columns and rows, and where M usually minimizes || A - CMR || (see, e.g., [9]). An ID or CUR decomposition has a clear interpretation in terms of variables and/or data points; a drawback is that the approximation of the original data tends to be of much lower quality compared to the SVD; see also some small examples in Section 3.4.

Fig. 3.1: Left: difference between an SVD (red orthogonal axes, corresponding to main features but not to actual data point) and an ID or CUR decomposition (green axes, actual data points) for a set of data points (blue). Right: best one-dimensional projection maximally separating the two groups.



In many situations, our problems have additional structure. For instance, if the set of variables naturally splits into two subsets X and Y, it may be relevant to extract linear combinations of features that have maximal correlation. This problem is referred to as *canonical correlation analysis* in statistics; its solution is given by the *restricted singular value decomposition* in linear algebra. A corresponding relevant optimization problem in terms of the covariance matrices Σ_{XX} , Σ_{XY} , Σ_{YY} is to maximize $||U^T \Sigma_{XY} V||$, under the conditions $U^T \Sigma_{XX} U = I_k$, and $V^T \Sigma_{YY} V = I_k$. Some current developments focus on exploiting low-dimensional and other structures of the data by advanced mathematical techniques.

Dimension reduction

As dimension reduction is often an indispensable first step in AI, this is a very popular research area. Linear dimension reduction is closely connected to feature extraction, but here the emphasis is on choosing a suitable $m \times k$ orthogonal basis U for a low-dimensional space; the associated orthogonally projected data points are then $U^T a_j$, where a_j are the columns of A. This means we are solving (cf. (3.1))

$$\min_{U \in \mathbb{R}^{m \times k}, Z \in \mathbb{R}^{n \times k}} ||A - UZ^T||, \quad s.t. \ U^T U = I_k,$$

with an ("essentially unique") solution in terms of the TSVD: $U = U_k$ and $Z = A^T U_k = V_k \Sigma_k$.

There are also various nonlinear dimension reduction techniques (see, e.g., [5]), some of which have a statistical motivation. For example, *locally linear embedding* first approximates $A \approx AW$, where W is an $n \times n$ weighting matrix, where we allow only some of the elements to be nonzero; typically ℓ per column, the number of included nearest neighbors. With affine combination requirements $\sum_i w_{ij} = 1$ for all j, we then are interested in preserving the modest quantity || A - AW || to a projected version $|| U^T (A - AW) ||$, where U is of size $m \times k$, subject to the constraints that $Y := U^T A$ is centered with unit covariance. In fact, the solution may again be derived from a TSVD.

Some of the current research efforts study the problem of maximizing the projection of one quantity (represented by *A*), while minimizing the projection of another (represented by the matrix *B* of the same size), for instance in the form of a trace ratio maximization problem (see, e.g., [8]):

$$\min_{U \in \mathbb{R}^{m \times k}} \frac{\operatorname{tr}(U^T A U)}{\operatorname{tr}(U^T B U)} \qquad s.t. \ U^T U = I_k, \tag{3.2}$$

where *tr* denotes the trace, and *U* corresponds to the desired projection. This is not directly related to a generalized eigenvalue problem involving the pair (*A*, *B*); the latter is in fact a simplification. Problem (3.2) is often solved iteratively using eigenvectors of $A - \rho B$ for several ρ -values. Several approaches may be viewed as extensions of *Fisher's linear discriminant analysis*. Figure 3.1 (right) shows a simple example, using a between scatter matrix for *A* and a within-scatter matrix for *B*.

Clustering

Clustering, the task of collecting unclassified data points into appropriate groups, is also a very difficult task. One popular class of methods is *spectral clustering*, based on linear algebra techniques [11]. For more challenging data sets we may need statistical modeling and techniques (see, e.g., [7]). These may lead to a complicated nonlinear optimization problem for a loglikelihood function. These problems arising in clustering tend to have a huge number of local optima. Both for statistical flavored techniques as *expectation maximization* (see, e.g., [6]) and for optimization oriented methods as *spectral projected gradient* methods (see, e.g., [1]), one of the goals is to select promising starting points, to avoid local minima (*global optimization*), and to interpret and compare candidate solutions.

Matrix decompositions

While the SVD may be the best-known matrix decomposition, another useful factorization for is nonnegative input matrices A is the nonnegative matrix factorization (NMF) $A \approx XY^T$, where both factors $X \in \mathbb{R}^{m \times k}$ and $Y \in \mathbb{R}^{n \times k}$ have nonnegative elements (see, e.g., [2]). This decomposition is used in many fields, including image analysis and matrix completion (related to the "Netflix problem", which means guessing missing user ratings). Results of an NMF are sometimes easier to interpret, for instance in the context of images, where pixel values are nonnegative. While the SVD is easy to compute (polynomial costs of low degree), it is known that the computation of NMF is NP hard.

As a very simple example, consider the 4× 4 data matrix *A* with its best rank-1 approximation (via the SVD):

A =	$\begin{bmatrix} 9\\10\\2\\10 \end{bmatrix}$	7 1 3 6	$ \begin{array}{c} 10 \\ 10 \\ 2 \\ 10 \end{array} $		\approx	$\begin{bmatrix} 0.63 \\ 0.51 \\ 0.26 \\ 0.52 \end{bmatrix}$	• [28.2] •	$\begin{bmatrix} 0.59 \\ 0.31 \\ 0.61 \\ 0.43 \end{bmatrix}$	\approx	$\begin{bmatrix} 10.5 \\ 8.5 \\ 4.2 \\ 8.6 \end{bmatrix}$	$5.6 \\ 4.5 \\ 2.3 \\ 4.6 $	$ \begin{array}{r} 10.9 \\ 8.8 \\ 4.4 \\ 8.9 \end{array} $	7.8 6.3 3.1 6.4	
	L10	6	10	2]		[0.52]		L0.43]		L 8.6	4.6	8.9	6.4	

Since the factors $X = U_1$ and $Y = \Sigma_1 V_1$ are nonnegative, this is also the optimal NMF of rank 1. For k = 2, the best rank-2 approximation $U_2 \Sigma_2 V_2^T$ (or $X = U_2$ and $Y = \Sigma_2 V_2$) is approximately

						T					
[0.63]	0.33 J			۲0.59	-0.37]	1	F 9.4	6.0	9.9	10.3]	
0.51	-0.29	[28.2	1	0.31	0.15	~	9.4	4.1	9.6	4.1	
0.26	0.75	•	9.0	0.61	-0.33	\sim	1.8	3.3	2.2	8.9	•
0.52	-0.49	L	1	0.43	0.86		10.2	3.9	10.4	2.6	

This involves second columns of U_2 and V_2 which necessarily contain negative elements, since the U-columns need to be orthogonal, and similarly the V-columns. We have that $||A - A_2|| \approx 4.2$. An NMF with the columns of U scaled to have unit norm gives an approximation

$A \approx$	$\begin{bmatrix} 0.76 \\ 0.20 \\ 0.40 \\ 0.48 \end{bmatrix}$	$\left[\begin{array}{c} 0.15 \\ 0.96 \\ 0.15 \\ 0.17 \end{array} \right]$	$\begin{bmatrix} 10.80 \\ 10.03 \\ 11.32 \\ 12.20 \end{bmatrix}$	8.09 0.00 8.20 1.52	$T \approx$	ſ	$9.3 \\ 10.0 \\ 5.5 \\ 6.5$	$7.6 \\ 2.0 \\ 4.0 \\ 4.8$	$9.8 \\ 10.2 \\ 5.7 \\ 6.8$	9.4 4.0 5.1 6.1	
	[0.48]	0.17]	L12.20	1.52		L	6.5	4.8	6.8	0.1 J	

with necessarily a (often much) larger error \approx 9.3. Numerous approaches to compute an NMF are currently developed, such as based on alternating minimization, multiplicative updates, and projected gradients. When we restrict an NMF so that all *n* rows of the *Y*-factor have unit 1-norm, it provides an approximate *convex* combination of the data points as sum of the *k* columns of *X*. This means that the columns of *X* have a tendency to extreme data points. There are also close links between NMFs and clustering (cf., e.g., [4]). If we impose an extra constraint that *Y* should be binary (only containing zeros and ones), with all rows summing up to *I*, then this corresponds to ("hard") clustering with the columns of *X* as the cluster centers. There are also similar connections with soft ("uncertain") clustering methods.

A rank-2 CUR approximation, which should be built up from the columns and rows, is:

$\begin{bmatrix} 9\\10\\2 \end{bmatrix}$	$\begin{bmatrix} 10 \\ 5 \\ 9 \end{bmatrix}$	$\begin{bmatrix} -0.03 \\ 0.12 \end{bmatrix}$	$\begin{bmatrix} 0.12 \\ 0.12 \end{bmatrix}$.	$\begin{bmatrix} 2\\ 3\\ 2 \end{bmatrix}$	$\begin{bmatrix} 2 & 10 \\ 3 & 6 \\ 2 & 10 \end{bmatrix}^T \approx$	\approx	$\begin{bmatrix} 0.98 \\ 6.76 \\ 6.07 \end{bmatrix}$	2.39 4.69 -1.72	$0.98 \\ 6.76 \\ 6.07$	8.81 4.39 7.05
$\begin{bmatrix} 10\\2\\10 \end{bmatrix}$	$\begin{bmatrix} 9\\2 \end{bmatrix}$. 0.12	-0.12].	$\begin{bmatrix} 2\\9 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}$	\approx	$\begin{bmatrix} -6.07\\ 9.54 \end{bmatrix}$	-1.72 5.69	-6.07 9.54	7.95 1.73

with a much larger error \approx 18.9. (Note that this approximation contains negative elements.)

Many techniques have both close theoretical and practical connections with matrix eigenvalue problems, as already mentioned in Section 3.3. The SVD of a data matrix A has relations with the eigenvalue decomposition of A^TA , of AA^T , and of $\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}$. The latter augmented matrix is also useful for biclustering problems. Other relevant matrix factorizations include the rankrevealing QR decomposition (used for instance for a greedy selection of data points in an ID decomposition) and the generalized SVD.

Outlook

The intersection of linear algebra, statistics, data science, and optimization is an elegant, beautiful, blossoming, and fruitful area. It is of huge importance for AI and in engineering disciplines, and may be essential for reliable dimension reduction, efficient methods, and interpretable results. Moreover, there are beautiful connections with several other math fields such as graphs, complex networks, high-order tensors, inverse problems and regularization, and probability theory.

There is a clear need for mathematical innovations, for several reasons. First, there is a continuous growth in the size of the data sets, which requires large-scale techniques in scientific computing, linear and nonlinear algebra, and numerical optimization. Second, it is widely recognized that one of the frequent bottlenecks of results of AI is the lack of interpretability of the results, and/or a clear understanding of how the results are obtained. Several techniques, of which a CUR decomposition and an NMF are examples, may be crucial to enhance understanding. Third, it is understood that it is of great benefit to exploit any (mathematical) structure of the data, such as intrinsic low-dimensionality. Fourth, it is crucial to understand and quantify the uncertainty in the computed results; this aspect may sometimes not be completely understood by users of the obtained results. This quantification often requires a combination of scientific computing, inverse problems, and statistics.

Finally, there currently is an increasing need and trend connecting the fields of linear algebra, statistics, and numerical optimization. This synergy provides very interesting results, and may also be necessary to solve the challenging and demanding questions.

On the one hand, several of the described problems are NP hard, which leads to challenging tasks with many alternative solution techniques. One the other hand, some of the necessary ingredients (such as approximating a TSVD) are among the easiest problems in numerical linear algebra, which ensures that a large number of very elegant mathematical methods are also practically feasible. For instance, while a "full" SVD may

be too expensive for large-scale data, mathematically elegant and efficient methods exist to approximate partial information for this case. Additionally, for truly huge data sets or streaming data, randomized approaches are being developed that access only part of the data (see, e.g., [3]).

There is a wide range of fascinating and tough open problems. Especially the intersection area of the mentioned mathematical fields may be expected to lead to breakthroughs, and will remain fresh and green for decades to come.

Acknowledgement This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 812912.



References

- E. G. Birgin, J. M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. SIAM J. Optim., 10(4):1196– 1211, 2000.
- [2] N. Gillis. The why and how of nonnegative matrix factorization. In Regularization, Optimization, Kernels, and Support Vector Machines, pages 257–291, 2014.
- [3] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.
- S. C. Hess. A Mathematical Theory of Making Hard Decisions Model Selection and Robustness of Matrix Factorization with Binary Constraints. Ph.D. Thesis, Universität Dordtmund, 2019.
- [5] E. Kokiopoulou, J. Chen, and Y. Saad. Trace optimization and eigenproblems in dimension reduction methods. *Numer. Linear Algebra Appl.*, 18(3):565–602, 2011.
- [6] T. K Moon. The expectation-maximization algorithm. *IEEE Signal Process.* M., 13(6):47–60, 1996.
- [7] M. E. J. Newman and E. A. Leicht. Mixture models and exploratory analysis in networks. Proc. National Academy of Sciences, 104(23):9564–9569, 2007.
- [8] T. T. Ngo, M. Bellal , and Y. Saad. The trace ratio optimization problem. SIAM Review, 54(3):545–569, 2012.
- [9] D. C. Sorensen and M. Embree. A DEIM induced CUR factorization. SIAM J. Sci. Computing, 38(3):A1454–A1482, 2016.
- [10] G. Strang. Linear Algebra and Learning from Data. Wellesley-Cambridge Press, 2019.
- [11] U. Von Luxburg. A tutorial on spectral clustering. Statistics and computing, 17(4):395–416, 2007.

A TALE OF TWO CONFERENCES, OR: THE ONGOING DANCE BETWEEN MATH AND MACHINE LEARNING

PETER GRÜNWALD

Abstract We review some key historical developments in machine learning by relating them to two of its most prestigious conferences: the huge NeurIPS and the much much smaller COLT. The story highlights the crucial role of mathematics for machine learning (and hence for AI in general), in the past as well in the future.

Waves and Fashions in AI

The terminology Artificial Intelligence was coined at a legendary workshop that took place in Dartmouth, in 1956. Here is a famous quote from the meeting: 'every aspect of learning, or any other feature of intelligence, can be so precisely described that a machine can be made to simulate it...': in the early days, it was believed that automated learning would play an essential role in AI. Such automated learning methods included, but were certainly not limited to, what we now call artificial neural networks. In the 1960s and 1970s however, the field moved more and more away from learning, focusing on so-called 'logical' approaches to AI, relying on hard-coded knowledge about the world. Especially after the publication of Minsky and Papert's 1969 book Perceptrons, the neural network approach, and with it other forms of automated learning, came to be seen as a dead end. Something, however, began to stir again in the 1980s, when more and more young researchers became interested in machine learning (ML), as it then became to be called.

Not surprisingly, these researchers started to organize: 1987 saw the first edition of *NeurIPS*, the world's currently largest and most influential conference in machine learning. NeurIPS, until 2019 known as NIPS, stands (and stood) for *Neural Information Processing Systems*, and in the early years it was part conference on artificial neural networks and other biologically inspired cognitive models ('how does the brain work'), part a conference on using such models to make better machine learning systems, and part conference on machine learning per se.

Some of the new machine learning enthousiasts were theoretical computer scientists with a strong mathematical inkling. In 1988, just one year after the first NIPS, they organized the first annual COLT workshop. COLT¹ stood for *Computational* (i.e., machine) *Learning Theory* — coincidentally the acronym was coined by Dutch computer scientist and learning theory pioneer Paul Vitányi, who later became my Ph.D. thesis supervisor. *Theory* here really means *mathematical* theory: most papers at COLT read like pure mathematics papers, full of headers with *Theorem*, *Lemma* and *Proof*, interspersed with formula after formula.

In the 1990s, COLT — even though it remained small and not very well-known developed into a highly prestigious annual conference on its own. Two of the main developments in machine learning in the 1990s can be traced back to COLT: *boosting* and *support*

vector machines (SVMs). Many of the ground-breaking papers on boosting were originally published at the COLT conference, where its inventors, Freund and Schapire, are central figures. While most original work on SVMs was not directly published at COLT, it was developed in the same style — mathematical derivations — and in close intellectual proximity: SVMs are due to Vladimir Vapnik, whose groundbreaking work on machine learning theory was rediscovered by learning theorists in the mid 1980s. His most well-known book, with over 90000 citations, is *the Nature of Statistical Learning Theory* — and is once again essentially a mathematics book: theorems, proofs, formulas.

Waves and Fashions in ML

Arguably, SVMs and boosting were, alongside with the – also quite mathematical Bayesian networks, the leading ML technology until around 2012, when deep neural networks came along and revolutionized the field. Ironically, the mathematical SVMboosting-Bayes nets methods had themselves replaced the earlier wave of neural networks in the 1990s — which was already a second wave, having in turn replaced the no-learning approaches of the 1970s. These second-generation neural nets were quite popular in the 1980s, but they did not work so well on the small data sets and slow computers of that age. Because of their mathematical intractability, it was not so clear how to improve their performance, and they were therefore, starting from the mid 1990s, superseded by methods such as SVMs and boosting, which were much more amenable to mathematical analysis. Although, as we've seen, the N in NIPS stood for Neural, I was told back in 2010 that "the last paper at NIPS with 'neural' in the title probably appeared around 1996." All this changed again around 2012 when the new 'deep' generation of neural nets came along – partly based on better algorithms, silently developed in the late 1990s, partly because of incomparably larger data sets and computational power, they now started to

shine and perform extremely well in several applications, mostly involving images and language.

Nowadays, deep nets may be said to dominate the field. Much of the progress in deep nets has been achieved by flashes of insight combined with trial-and-error rather than mathematical analysis — see Fig. 4.1. Still, leading researchers in AI and Machine Learning are highly aware of the importance of mathematics to

Fig. 4.1: Title Suppressed Due to Excessive Length Machine Learning Without Math (reprinted from the inescapable xkcd.com series). I would be the last to deny that trial-and-error approaches combined with brute force computing and brilliant intuitions have led to tremendous advances in machine learning. But the grand historical picture rather suggests that there are really alternating waves of progress some trial-and-error-based, others fueled by mathematical analysis.



the field — deep nets are still quite ill-understood (nobody really knows why they work so well) and it is expected that, once we have a better mathematical understanding, new roads will open up. I myself have been involved in trying to better match mathematical error guarantees to neural network outputs, and I am very happy to see other work on the mathematics of and with deep nets taking off in the Netherlands, for example in (but not limited to!) Eindhoven and Twente. Taking a grand historical perspective and admittedly ignoring much of messy reality one discerns alternating waves of trial-and-error research (1980s), much more first-principles-oriented mathematics research (1990s, 2000s) and then trial-and-error research again, and it then may only be natural to expect that the next breakthrough will be...mathematical in nature.

The Importance of Being Mathematical

In this respect, it is also worthwhile noting that many of the program chairs of the huge NeurIPS conference (13000 participants in 2019) regularly publish at the tiny COLT conference (about 250 participants in 2019): a quick count shows that over the last 15 years, an average of 30% of Neurips chairs have close links to COLT! It seems that the mathematicians inside machine learning are punching far above their weight, but quite successfully so.

Why am I telling you all this? Part of the reason, I admit it, is advertisement: Since 2019 I have been the chair of the Association for Computational Learning, the organization running the Annual COLT Conference. But another part is a Worry. A Worry that the role of Mathematics in Machine Learning is under-appreciated in the Netherlands. It is a well-known and eternal rant but I am going to repeat it anyway: here at home, the immediate question is more often than not *What can I use it for and what is the deployment time*? or even *How does your research support the BV Nederland*? — an attitude that Vincent Icke has called *De Dictatuur van* het Nut. Mathematical research is curiosity-driven. It often has fantastic practical repercussions, but these come later - and are usually not clearly anticipated when the research itself is done. I sometimes feel like I have to defend myself doing such research, even getting questions like the somewhat anti-scientific 'why would you care understanding how it works when we simply see that it works so well?'. Ironically, while the common perception about the USA is that it is all and only about the money, all the top universities there have mathematically oriented machine learning theory groups — at every single COLT you'll see papers from Princeton, Yale, Columbia, Stanford, Berkeley, MIT and the like — and, witness the strong link to COLT I mentioned above the US-dominated NeurIPS conference, with all its highly applicable research, still has an agreeably soft spot for theory. I would love to see more of that attitude in the Netherlands. But I shouldn't rant too much — so let me end on a positive note: despite my complaints, good mathematical ML in the Netherlands, has somehow managed to exist and keep existing since the late 1980s - and recently, it has found some very active proponents, as is proved by the book you are now reading. Mathematicians are now really doing their best to raise awareness. I hope they succeed.

Note

I In 2018, NIPS was renamed to NeurIPS because of its association with the word nipples and its meaning in Japanese. Given the perhaps even stronger connotations, one wonders when COLT will change its name...

UNCERTAINTY QUANTIFICATION FOR PHYSICS-INFORMED DEEP LEARNING

MENGWU GUO AND CHRISTOPH BRUNE

Abstract The development of physics-informed deep learning is radically changing computational science and engineering, allowing for an effective integration ofphysics-based and datadriven modeling. Deep learning provides a powerful tool forthe discovery of governing dynamics underneath data and enables nonlinear modelreduction. A Bayesian viewpoint of deep learning is discussed in this chapter towards the quantification of modeling uncertainties in physics-informed deep learning.

Introduction

In computational science and engineering, data-driven modeling benefits from up-to- date techniques of machine learning and stands out for its capabilities in large-scaleproblems that are deemed challenging for traditional simulation tools. Especially inthe areas of engineering and health, the growing need for probabilistic design optimization, risk assessment, model updating, and sensor data integration, all over thelifecycle of real-world assets, requires multi-query and real-time analysis in whichparametric PDEs are usually used to model the variation in the physical states. In such a context, the high-fidelity, detailed simulations are known to be too computationally intensive to enable realtime decision support, and the computational cost of evaluations at large amounts of parameter locations is unaffordable. To reduce the high cost of detailed parametric simulations, datadriven surrogate models havebeen developed to approximate the maps from characteristic parameters or featuresto output quantities of interest, often using algorithmic methods of deep learning. In the typical scenario of purely data-driven modeling, large datasets are often requiredso that the deep learning models can be properly trained. However, it is often the case that the size of the training dataset provided by simulations or experiments cannot meet the needs of model training. In this case, physics information has to beincorporated into the deep learning models to guarantee a good generalization. Sucha modeling strategy is often referred to as physics-informed deep learning [1, 2]. Generally speaking, these models benefit from deep learning's power in data analytics, reduce the expensive cost of physics-based simulations, overcome the difficulties stemming from limited training data, and enables a synthesis of both data-driven and physicsinformed natures.

On the other hands, the fast, accurate solution of large-scale forward and inverseproblems governed by nonlinear, time-dependent, parametric, and possibly highdimensional PDEs promotes the urgent need for nonlinear model order reduction that discovers low-rank structures and their governing equations underlying the physics. There is no doubt that deep learning can provide effective algorithmic tools for suchcomputational tasks. In fact, the synthetic integration of physics-based modeling and datadriven manifold learning forms an important part of the emerging field ofphysics-informed deep learning.

However, there exist many types of modeling uncertainties in physics-informeddeep learning, including the inherent uncertainties in the system parameters, data noise, sampling errors, discretization and truncation errors from the physics-based models, and those stemming from the assumptions made for data-driven discovery, as well as many factors due to the blackbox nature of deep learning. Therefore, toproperly address the explainability, reliability and robustness of such systems, the uncertainty quantification of physics-informed deep learning becomes an importanttopic to be promptly investigated.

Learning reduced models from data

For a mathematical model governed by parametric, time-dependent, nonlinear PDEs, a detailed spatial discretization often leads to a high-dimensional dynamical systemdescribed by ODEs. When solving such a system, there will be a high demand for computational resource, both CPU time and memory. Model order reduction aims to reduce a large-scale system to a relatively small dimensionality, achieving a substantially reduced computational cost without significantly compromising the accuracy.Based on a collection of snapshot solutions over the time-parameter domain, a typical strategy of reduced order modeling is to extract a predominant, low-dimensional subspace of the solutions manifold and project the full-order system onto such a reduced space. Seeking to find a low-dimensional structure with underlying patternfrom high-dimensional data, model order reduction essentially shares the spirits of machine learning. Indeed, the construction of reduced spaces often employs the unsupervised learning methods of dimensionality reduction, and the interpolation of parametric variation can benefit from the algorithmic methods of supervised learning [3].

Different from the conventional model reduction that projects the known governing equations onto the reduced space, the data-driven versions of reduced order modeling learn the lowdimensional dynamical systems from data. Aligned perfectly with the scope of physics-informed machine learning, deep neural networks provide a handy tool for such data-driven discovery of the physics, dynamics, or features of the reduced systems [4, 5, 6]. However, the reduction of original systems, the noise-corrupted training data, and the assumptions made for the discovery of reduced-order physics, as well as many other factors, all lead to modeling uncertainties and computational errors, which makes uncertainty quantification especially critical to physics-informed deep learning.

A Bayesian viewpoint of deep learning

It is well known that Bayesian neural networks provide a probabilistic framework for the quantification of modeling uncertainties in deep learning. In fact, Bayesian neural networks have close ties with Gaussian processes, the latter of which follows a kernel method and thus can be investigated in reproducing kernel Hilbert spaces (RKHS). As shown in the conceptual diagram in Fig. 5.1, we briefly note such connections between Bayesian deep learning and Gaussian processes, and discuss the relevant theoretical background via kernel ridge regression, RKHS, and Barron spaces.

Bayesian neural network as Gaussian process

It can be shown that, under certain assumptions, the prior of a multi-layer Bayesianneural network is equivalent to a Gaussian process [7, 8, 9], which provides a concise but rigorous viewpoint for the uncertainty quantification of deep learning. Definedrecursively through the neural network layers, the equivalent Gaussian process, denoted by $\mathcal{GP}(h_{NN}(x), k_{NN}(x, x'))$ and corrupted by an independent noise $\mathcal{N}(0, \sigma_{e}^{2})$, can be used for regression.

Conditioning on the input-output training data (*X*, *y*), the noise-free posterior output $y^*(x)|X$, *y* follows a new Gaussian process, i.e., $y^*(x)|X$, $y \sim \mathcal{GP}(h^*_{NN}(x), k^*_{NN}(x, x'))$ whose mean and covariance functions are

 $\begin{aligned} &h_{\rm NN}^*(x) = h_{\rm NN}(x) + k_{\rm NN}(X, x)^{\rm T} [K + \sigma_{\rm e}^2 I_M]^{-1}(y - h_{\rm NN}(X)) \text{ ; and} \\ &k_{\rm NN}^*(x, x') = k_{\rm NN}(x, x') - k_{\rm NN}(X, x)^{\rm T} [K + \sigma_{\rm e}^2 I_M]^{-1} k_{\rm NN}(X, x') . \end{aligned}$

in which $K = k_{\text{NN}}(X, X)$. In addition, the hyperparameters, including the noise σ_{ϵ}^2 and those for the Bayesian neural net, can be determined by maximizing the marginal likelihood.





Reproducing kernel map reconstruction and kernel ridge regression

An RKHS [10] $\mathcal{H}_{k_{\text{NN}}}(\Omega)$ is induced by the kernel function k_{NN} and provides a completion of the following function space of reproducing kernel map reconstruction:

$$\mathcal{C}_{k_{\mathrm{NN}}} = \left\{ f(\boldsymbol{x}) = \sum_{m=1}^{M} \beta_m k_{\mathrm{NN}}(\boldsymbol{x}^{(m)}, \boldsymbol{x}) \middle| M \in \mathbb{N}^+, \boldsymbol{x}^{(m)} \in \Omega, \beta_m \in \mathbb{R} \right\}.$$

Comparing the posterior mean $h_{NN}^*(x)$ with the prior mean $h_{NN}(x)$, the correction term, $\Delta_{NN}(x) = h_{NN}^*(x) - h_{NN}(x) = k_{NN}(X, x)^T \beta$, is evidently a reproducing kernel map reconstruction, and thus $\Delta_{NN} \in \mathcal{H}_{k_{NN}}$. Here

$$\begin{split} \boldsymbol{\beta} &= [\mathbf{K} + \sigma_{\epsilon}^{2} \mathbf{I}_{M}]^{-1} (\mathbf{y} - h_{\text{NN}}(\mathbf{X})) \\ &= \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^{M}} \left\{ \|\mathbf{y} - h_{\text{NN}}(\mathbf{X}) - \boldsymbol{\Delta}_{\text{NN}}(\mathbf{X})\|_{2}^{2} + \sigma_{\epsilon}^{2} \|\boldsymbol{\Delta}_{\text{NN}}\|_{\mathcal{H}_{k_{\text{NN}}}}^{2} \right\} \,, \end{split}$$

in which $||\Delta_{NN}||^2_{\mathcal{H}_{K_{NN}}} = \beta^T K \beta$, i.e., the combination coefficients of the kernel map reconstruction are determined through a least squares problem regularized by the squared RKHS norm, often referred to as a kernel ridge regression [11]. In thisway, connections have been built among Bayesian neural network, Gaussian process, RKHS, and the corresponding regularization defining a kernel ridge regression.

Reproducing kernel Hilbert space vs. Barron space

When considering the case of a two-layer neural network, the induced kernel functioncan be simplified as

$$k_{\mathrm{NN}}(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}_{(\boldsymbol{w}, b) \sim \pi}[\phi(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b) \cdot \phi(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}' + b)] := k_{\pi}(\boldsymbol{x}, \boldsymbol{x}'),$$

when (w,b) follows any distribution $\pi.$ We define for a fixed $\pi \in P(\mathbb{S}^{d_{\mathrm{in}}})$ that

$$\mathcal{H}_{\pi}(\Omega) = \left\{ f(\boldsymbol{x}) = \int_{\mathbb{S}^{d_{\text{in}}}} \alpha(\boldsymbol{w}, b) \phi(\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x} + b) \, \mathrm{d}\pi(\boldsymbol{w}, b) \Big| \|f\|_{\mathcal{H}_{\pi}} < \infty \right\},$$

with
$$||f||_{\mathcal{H}_{\pi}}^{2} := \mathbb{E}_{(\boldsymbol{w},b)\sim\pi}[|\alpha(\boldsymbol{w},b)|^{2}],$$

where $\mathbb{S}^{d_{\text{in}}} := \{(\boldsymbol{w}, b) | \| \{ \boldsymbol{w}^{\text{T}}, b \}^{\text{T}} \|_{1} = 1 \}$, and $P(\mathbb{S}^{d_{\text{in}}})$ denotes the collection of all probability measures on $(\mathbb{S}^{d_{\text{in}}}, \mathcal{F}), \mathcal{F}$ being the Borel σ -algebra on $\mathbb{S}^{d_{\text{in}}}$. It has been shown that $\mathcal{H}_{\pi} = \mathcal{H}_{k_{\pi}}$ [12]. Naturally connected with such an RKHS is the Barron space [13] defined as

$$\begin{split} \mathcal{B}_2(\varOmega) &= \left\{ f(\boldsymbol{x}) = \int_{\mathbb{S}^{d_{\mathrm{in}}}} \alpha(\boldsymbol{w}, b) \phi(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x} + b) \, \mathrm{d}\pi(\boldsymbol{w}, b) \Big| \pi \in P(\mathbb{S}^{d_{\mathrm{in}}}), \\ \|f\|_{\mathcal{B}_2} &< \infty \right\}, \ \text{with} \ \|f\|_{\mathcal{B}_2}^2 := \inf_{\pi} \mathbb{E}_{(\boldsymbol{w}, b) \sim \pi}[|\alpha(\boldsymbol{w}, b)|^2]. \end{split}$$

Thus we have $\mathcal{B}_2(\Omega) = \bigcup_{\pi \in P(\mathbb{S}^{d_{\mathrm{in}}})} \mathcal{H}_{\pi}(\Omega) = \bigcup_{\pi \in P(\mathbb{S}^{d_{\mathrm{in}}})} \mathcal{H}_{k_{\pi}}(\Omega)$, i.e., the Barron space \mathcal{B}_2 is the union of a class of RKHS $\mathcal{H}_{k_{\pi}}$ that are defined by the neural-network induced kernels k_{π} through two-layer neural networks. It would be a promising research direction to explore the theoretical foundation of the aforementioned Bayesian viewpoint in Barron spaces.

Outlook

Physics-informed deep learning, especially deep-learning-based nonlinear reducedmodeling, will find many large-scale, complex applications in lots of emerging areasin engineering and health, such as advanced manufacturing, robotics, and the digitaltwinning of industrial and medical assets. All these applications will thus motivate the advancement of robust, scalable physics-informed deep learning in a multiphysics,multiscale, multifidelity context. A careful comparison with classical and datadriven regularization [14] for inverse problems may lead to valuable insights. Basedon the preliminary discussions on the connections among Bayesian neural networks,Gaussian processes, reproducing kernel Hilbert spaces and Barron spaces, further mathematical theory has yet to be developed and will lay the foundation of uncertaintyquantification for physics-informed deep learning. **Acknowledgment**: M. Guo is supported by *Sectorplan Bèta* under the focus area *Mathematics of Computational Science*. C. Brune acknowledges support by the EUHorizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie grant agreement No. 777826 (NoMADS).

References

- N. Baker, F. Alexander, T. Bremer, A. Hagberg, Y. Kevrekidis, H. Najm, M. Parashar, A. Patra, J. Sethian, S. Wild, K. Willcox, and S. Lee. Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence. USDOE Office of Science, Washington, DC, 2019.
- [2] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang. and L. Yang. Physics-informed machine learning. Nature Reviews Physics, 3(6), pages 422- 440, 2021.
- [3] M. Guo. and J. S. Hesthaven. Data-driven reduced order modeling for timedependent problems. Computer Methods in Applied Mechanics and Engineering, 345, pages 75–99, 2019.
- [4] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton. Data-driven discovery of coordinates and governing equations. Proceedings of the National Academy of Sciences, 116(45), pages 22445–22451, 2019.
- [5] B. Peherstorfer, and K. Willcox. Data-driven operator inference for nonintrusive projection-based model reduction. Computer Methods in Applied Mechanics and Engineering, 306: pages 196-215, 2016.
- [6] M. Kalia, S. L. Brunton, H. G. Me er, C. Brune, and J. N. Kutz. Learning normal form autoencoders for data-driven discovery of universal, parameterdependent governing equations. arXiv: 2106.05102, 2021.
- [7] M. Guo. A brief note on understanding neural networks as Gaussian processes. arXiv: 2107.11892, 2021.
- [8] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein. Deep neural networks as Gaussian processes. arXiv: 1711.00165, 2017.
- [9] R. M. Neal. Bayesian Learning for Neural Networks. Springer Science & Business Media, 2012.
- [10] C. K. Williams and C. E. Rasmussen. Gaussian Processes for Machine Learning. MIT Press, Cambridge, 2006.
- [11] P. L. Bartlett, A. Montanari, and A. Rakhlin. Deep learning: a statistical viewpoint. arXiv: 2103.09177, 2021.
- [12] A. Rahimi and B. Recht. Uniform approximation of functions with random bases. In 2008 46th Annual Allerton Conference on Communication, Control, and Computing, pages 555–561. IEEE, 2008.
- [13] W. E, C. Ma, and L. Wu. The Barron space and the flow-induced function spaces for neural network models. Constructive Approximation, pages 1– 38, 2021.
- [14] S. Arridge, P. Maass, O. Öktem, and C. B. Schönlieb. Solving inverse problems using data-driven models. Acta Numerica, 28, pages 1–174, 2020.



GPU ACCELERATION OF THE SEVEN-LEAGUE SCHEME FOR LARGE TIME STEP SIMULATIONS OF STOCHASTIC DIFFERENTIAL EQUATIONS

SHUAIQIANG LIU, GRAZIANA COLONNA, LECH GRZELAK AND CORNELIS OOSTERLEE

Abstract Monte Carlo simulation is widely used to numerically solve stochastic differential equations. Although the method is flexible and easy to implement, it may be slow to converge. Moreover, an inaccurate solution will result when using large time steps. The Seven League scheme [1], a deep learning-based numerical method, has been proposed to address these issues. This paper generalizes the scheme regarding parallel computing, particularly on Graphics Processing Units (GPUs), improving the computational speed.

Introduction

In this paper, we will develop a highly accurate numerical discretization scheme for stochastic differential equations (SDEs), which is based on taking possibly large discrete time steps. We "learn" to take large time steps [1] by using an artificial neural network (ANN), in the context of supervised machine learning, with the help of stochastic collocation polynomials (SCMC, see [2]).

In many forms and flavors, the deep learning paradigm [3] receives much attention in science and engineering nowadays. The Physics-Informed Neural Networks (PiNN) [4], for example, combining physical and mathematical insights in an unsupervised learning fashion, starts to enter the field of solving ordinary (ODEs) and partial differential equations (PDEs). The corresponding computational costs are nontrivial, however, simply because the underlying equations need to be learned from scratch, and this costs time. Supervised learning, based on labels, on the other hand, is a classical form of machine learning, which is often more efficient as there is an offline stage, in which the inputoutput labelled relations are being learned, followed by a highly efficient online stage, where the learned manifold of solutions is evaluated for new input values. In our work, supervised learning is employed, which, however, strongly relies on a sophisticated numerical stochastic collocation technique, to achieve a challenging task within numerical analysis. In the present paper, we want to show that by parallelization the method's efficiency can be greatly enhanced, thus computing times of both the online and the offline stages are significantly reduced, on a graphics processing unit (GPU).

The basic idea of the 7L scheme is to learn a small number of (conditional) stochastic collocation points, and the learned neural network function is employed to forecast the unknown collocation points for the next time step. Then, by means of the stochastic collocation Monte Carlo sampler (SCMC) [2], the random paths are generated. Interestingly, the strong convergence error of the 7L scheme is independent of the size of the simulation time step. In other words, different from classical numerical schemes, the accuracy of the numerical solution does not decrease when solving SDEs with large time steps due to the learning stage.

Parallel computing on GPUs is prevalent nowadays, especially for numerical simulations that require intensive computational resources. Regarding stochastic differential equations, parallel implementations of classical Monte Carlo simulations on GPUs have been well studied, for example, [5, 6]. In this paper, we extend the 7L scheme by parallel computing to further improve its computational speed. There is inherent parallelism in the 7L scheme, as a neural network consists of a large number of artificial neurons that can work in parallel. Moreover, required interpolation functions are independent of each other. Thus, they can be easily distributed over different processing units.

Methodology, the 7L Scheme

Suppose a real-valued random variable Y(t) is defined on the probability space $(\Omega, \Sigma, \mathbb{P})$ with filtration $\mathcal{F}_{t \in [0,T]}$, sample space Ω , σ -algebra Σ and probability measure \mathbb{P} . For the time evolution of Y(t), consider the generic scalar Itô SDE,

$$dY(t) = a(t, Y(t), \theta)dt + b(t, Y(t), \theta)dW(t), \ 0 \le t \le T,$$
(6.1)

with the drift term $a(t, Y(t), \theta)$, the diffusion term $b(t, Y(t), \theta)$, model parameters θ , Wiener process W(t), and given initial value $Y_0 := Y(t = \theta)$. The solution of (6.1) is unique when the drift and diffusion terms meet some regularity conditions.

The basic discretization for each Monte Carlo path, is the Euler-Maruyama scheme [7], which reads,

$$\hat{Y}_{i+1} | \hat{Y}_i = \hat{Y}_i + a(t_i, \hat{Y}_i, \theta) \Delta t + b(t_i, \hat{Y}_i, \theta) \sqrt{\Delta t} \hat{X}_{i+1},$$
(6.2)

where $\hat{Y}_{i+1} := \hat{Y}(t_{i+1})$ is a realization (i.e., a number) from random variable $\tilde{Y}(t_{i+1})$, which represents the numerical approximation to exact solution $Y(t_{i+1})$ at time point t_{i+1} , and a real-

ization \hat{X}_{i+1} is drawn from the random variable X, which here follows the standard normal distribution $\mathcal{N}(0, 1)$. The Euler-Maruyama scheme will be used to generate the training data set. However, training will be based on tiny time steps (for accuracy reasons).

Similarly, the 7L scheme reads $\hat{Y}_{i+1}|\hat{Y}_i = g_m(\hat{X}_{i+1})$, where $g_m(\cdot)$ stands for a mapping function, transforming a standard normal distribution to the target distribution at time t_{i+1} . The function $g_m(\cdot)$ can be obtained through an interpolation technique, based on *m* pairs of collocation points (x_j, y_j) , where $j = 0, \ldots, m-1, x_j$ are obtained from the standard norm distribution *X* (here Gauss-Hermite quadrature points), and y_j are stochastic collocation points at time t_{i+1} , conditional on the previous realization \hat{Y}_i . In the context of Markov processes, the function of computing y_j can be written as follows,

$$y_j(t_{i+1}) \mid \hat{Y}_i = H_j \left(\hat{Y}_i, \ t_{i+1} - t_i, \ \theta \right).$$
(6.3)

So, a neural network can be trained to approximate the function $H_j(\cdot)$. The two key components of the scheme refer to the function $H_j(\cdot)$ and the interpolation function $g_m(\cdot)$, both of which will be parallelized in Section 6.2.

Parallelization

Parallelization is carried out by the parallel implementation of the appearing interpolation functions and the neural network involved in the algorithm.

Note that the parallelization of the 7L scheme is focused on the online stage, because the training stage is done offline and once. A variant, the 7L-CDC scheme, see the paper [1], with more interpolations in Step 3 above, can be parallelized similarly. In this work, we use the barycentric version [8] of Lagrange interpolation on GPUs and CPUs to fairly compare their speed performance.

Please refer to the original paper [1] for more interpolation techniques.

Algorithm I: A parallel algorithm of the 7L scheme

- 1. **Offline stage**: Train the ANNs to predict the stochastic collocation points. At this stage, we choose different θ values, simulate corresponding Monte Carlo paths, with small constant time increments $\Delta \tau = \tau_{i+1} \tau_i$ in $[0, \tau_{\max}]$, generate the corresponding collocation points $\hat{y_j} | \hat{Y_i} \approx y_j | \hat{Y_i}$, and learn the relation between inputs and outputs to obtain $\hat{H_k} \approx H_k$.
- 2. **Online stage**: Partition time interval

[0,*T*], $t_i = i T/N$, i = 0, ..., N, with equidistant "large" time step $\Delta t = t_{i+1} - t_i$, and output *N* sample paths.

3. Run the ANNs to compute *m* collocation points at time t_{i+1} for each path,

$$\hat{y}_{j}(t_{i+1})|\hat{Y}_{i} = \hat{H}_{j}(\hat{Y}_{i}, t_{i+1} - t_{i}, \theta), j = 1, 2, \dots, m,$$
(6.4)

and form a vector

$$\hat{y}_{i+1} = (\hat{y}_1(t_{i+1}) | \hat{Y}_i, \, \hat{y}_2(t_{i+1}) | \hat{Y}_i, \, \dots, \, \hat{y}_m(t_{i+1}) | \hat{Y}_i).$$

This step is parallelized by running the ANNs in a batch model on the GPU.

- Divide N sample paths into N_T groups, and allocate a group of N/N_T paths to a certain thread on GPUs.
- 5. For each of $\frac{N}{N_T}$ paths in a thread, compute interpolation function $g_m(\cdot)$, based on *m* pairs of (x_j, \hat{y}_j) .
- 6. Sample from *X* and obtain a sample $\hat{Y}_{i+1} | \hat{Y}_i = g_m(\hat{X}_{i+1})$.
- 7. Collect all paths \hat{Y}_{i+1} from N_T threads to form a compete set at time t_{i+1} .
- 8. Return to Step 3 by $t_{i+1} \rightarrow t_i$, iterate until terminal time *T*.

Numerical Results

In this section, we evaluate the computational performance of the parallized 7L scheme. Here we take the Ornstein-Uhlenbeck

(OU) process as an example. The OU process is a mean reverting process, defined as follows,

$$dY(t) = -\lambda(Y(t) - \overline{Y})dt + \sigma dW(t), \qquad 0 \le t \le T,$$
(6.5)

with \overline{Y} the long term mean of Y(t), λ the speed of mean reversion, and σ the volatility. The initial value is Y_0 , and the model parameters are $\theta := \{\overline{Y}, \sigma, \lambda\}$. Its analytical solution is given by,

$$Y(t) \stackrel{d}{=} Y_0 e^{-\lambda t} + \overline{Y}(1 - e^{-\lambda t}) + \sigma \sqrt{\frac{1 - e^{-2\lambda t}}{2\lambda}} X, \tag{6.6}$$

with $t_0 = 0$, $X \sim \mathcal{N}(0, 1)$. Equation (6.6) is used to compute the reference value to the path-wise error and the strong convergence.

In the training phase, the Euler-Maruyama scheme is used to discretize the OU dynamics and generate the data set (here five stochastic collocation points to learn within the ANN). After the training, the 7L scheme with the obtained ANNs is used to solve the OU process, as shown in **Algorithm I**.

The ANN hyperparameters are set as follows here. We use 4 hidden layers, 50 neurons per layers, a Softplus activation function, a Glorot initialization, the Adam optimizer, a batch size of 1024, and a learning rate of 10⁻³.

The parallized 7L scheme is evaluated on the GPU and CPU as follows,

- GPU Type: GeForce MX150, Graphic Cores: 384, Graphics clock: 1468 MHz, Memory speed: 6.01 GHz, Memory bandwidth: 48.06 Gb/s.
- CPU Type: Intel Core i7-8550U, Cores: 4, Maximum speed: 4.0 GHz, Base clock speed: 1.80 GHz.

The parallelization is done in CUDA (Compute Unified Device Architecture), the platform created by Nvidia. The threads, which are the basic operational units in CUDA, are computing processes that run in parallel. The number of threads N_T used for the parallelization is 256. Therefore, the number of paths N_B per thread is proportional to the total number of simulated paths N_P , $N_B = \frac{N_P}{N_T}$. The speedup ratio is defined by the running time of the parallelized code (on the GPU) divided by the running time of the original code (on the CPU). The reported time is obtained by running the corresponding code 100 times and taking the averaged execution time.

Table 6.1: Computational time (seconds) of the 7L and 7L-CDC schemes.

Number	th	e 7L scher	ne	the	the 7L-CDC scheme				
of paths	Sequential time	Parallel time	Speedup	Sequential time	Parallel time	Speedup			
1,000	1.555	0.268	5.8	1.296	0.062	20.9			
50,000	70.108	6.844	10.2	64.731	2.828	22.9			
100,000	134.745	14.623	9.2	132.198	5.886	22.5			
200,000	282.456	21.545	13.1	251.684	11.527	21.8			

As shown in Table 6.1, the speedup ratio appears to converge to 10 when the number of the sample paths increases. However, this ratio may fluctuate due to the unstable performance of GPUs and CPUs. The major acceleration comes from the parallelization of the interpolation process (here based on five collocation points), as the ANN running times on the GPU and CPU have a small difference in this test. The 7L-CDC scheme employs a global interpolation technique which is based on the marginal collocation points to compute the conditional collocation points for each random path, instead of using the ANNs for each path as the 7L scheme does in Step 3 of Algorithm I. The 7L-CDC scheme only requires the ANNs to compute a small number of marginal collocation points (here five marginal collocation points) along with the probability distribution. The 7L-CDC scheme can be used as a faster variant of the 7L scheme, as long as the global interpolation technique is computationally cheaper than the evaluation of the ANN. As shown in Table 6.1, the speedup ratio of the parallelized 7L-CDC scheme converges to 22. There are two interpolation processes (one for the five conditional collocation points and another for five marginal collocation points) in the 7L-CDC scheme, which explains why the speedup ratio is as twice as that of the 7L scheme.

The speedup ratio is also affected by other factors, for example, the number of threads and the configuration of the used GPU. The original paper [1] has proved that the numerical error does not grow when the simulation time step size increases. We find that the above property holds when the 7L scheme is implemented on GPUs in a parallel way.

Summarizing, a neural networks-based numerical solver for stochastic differential equations, the 7L scheme, has been parallelized to accurately carry out large time step simulations, with a further computational acceleration by a factor of 10 or even 20, on the used GPU.

Acknowledgment

This project is part of the ABC-EU-XVA project and has received funding from the European Unions Horizon 2020 research and innovation programme under the Marie Skldowska–Curie grant agreement No 813261.

References

- S. Liu, L. A. Grzelak, and C. W. Oosterlee. The Seven-League Scheme: Deep learning for large time step Monte Carlo simulations of stochastic differential equations. *arXiv*, page arXiv:2009.03202, September 2020.
- [2] L. A. Grzelak, J. Witteveen, M. Suarez-Taboada, and C. W. Oosterlee. The stochastic collocation Monte Carlo sampler: highly efficient sampling from expensive distributions. *Quant. Finance*, 19(2):339–356, 2019.
- [3] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.
- [4] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comp. Physics, 378:686–707, 2019.
- [5] A. M. Ferreiro, J. A. García, L. Souto, and C. Vázquez. A new calibration of the Heston stochastic local volatility model and its parallel implementa-

tion on GPUs. Math. Computers in Simulation, 177:467-486, 2020.

- [6] Á. Leitao and C. W. Oosterlee. GPU acceleration of stochastic grid bundling method for early-exercise options. *Int. J. Comp. Math.*, 92:2433– 2454, 2015.
- [7] E. Platen. An introduction to numerical methods for stochastic differential equations. *Acta Numerica*, 8:197–246, 1999.
- [8] J-P Berrut and L.N. Trefethen. Barycentric Lagrange Interpolation. SIAM Review, 46(3):501–517, 2004.



PHYSICS-INFORMED MACHINE LEARNING EMBEDDED INTO ISOGEOMETRIC ANALYSIS

MATTHIAS MÖLLER, DEEPESH TOSHNIWAL, AND FRANK VAN RUITEN

Abstract Physics-informed neural networks (PINNs) [1] are emerging technologiesthat strive to revolutionize the field of computeraided analysis of scientific and engineering problems by directly exploiting physical laws to drive network optimization. In this short note we propose a novel approach to embed the PINN paradigm into the framework of Isogeometric Analysis [2]. In contrast to classical PINNs which predict point-wise solution values to (initial-)boundary-value problems directly, ourIGA-PINNs learn solutions in terms of their expansion coefficients relative to a given B-Spline basis. This approach is furthermore used to encode the geometry and otherproblem parameters such as boundary conditions and feed them into the network asinputs, which allows the user to analyze different problem configurations effectively.

Physics-Informed Neural Networks

Since their introduction in the seminal paper [1], PINNs have seen multiple methodological advancements and applications to a wide range of problems as summarized in the review paper [3]. The general working principle of PINNs is as follows: Consider a two-dimensional boundary-value problem (BVP) of the form

 $\mathcal{L}u(x, y) = f(x, y) \qquad (x, y) \in \Omega, \tag{7.1}$

$$\mathcal{B}u(x, y) = g(x, y) \qquad (x, y) \in \Gamma, \tag{7.2}$$

where \mathcal{L} and \mathcal{B} represent differential operators defined in the domain Ω and its boundary $\Gamma := \partial \Omega$, respectively. Furthermore, let \mathcal{N} denote a feed-forward artificial neural network with multiple hidden layers. Feeding x and y as the network's primary inputs, the aim is to predict a solution $\hat{u}(x, y) = \mathcal{N}(x, y)$ such that it satisfies the above BVP. This is achieved by including the residual of (7.1) and the deviation of the predicted solution from the boundary conditions (7.2) into the loss function, i.e.

$$\ell(u) := \ell_{\mathcal{L}}(u) + \ell_{\mathcal{B}}(u), \quad \ell_{\mathcal{L}}(u) := \frac{\alpha}{N_{\Omega}} \sum_{k=1}^{N_{\Omega}} |\mathcal{L}\hat{u}(x_{k}, y_{k}) - f(x_{k}, y_{k})|^{2},$$

$$\ell_{\mathcal{B}}(u) := \frac{\beta}{N_{\Gamma}} \sum_{k=1}^{N_{\Gamma}} |\mathcal{B}\hat{u}(x_{k}, y_{k}) - g(x_{k}, y_{k})|^{2},$$
(7.3)

where N_{Ω} and N_{Γ} denote the number of training points (x_k, y_k) in the interior of Ω and on its boundary $\partial \Omega$, respectively. Other loss functions (e.g., integral quantities involving the solution) can be devised following this same residual-based principle. The parameters α and β can be used to give individual weights to the different components of the loss function. Non-scalar parameters like f and g and the shape of the domain Ω are often hardcoded into the network N. The notation $\mathcal{L}\hat{u}(x_k, y_k)$ implies that the differential operator is applied to the network, i.e. $\mathcal{LN}(x_k, y_k)$, which is technically achieved with the aid of algorithmic differentiation. Despite some impressive results reported in the literature and the ease of implementing PINNs, they lack a robust mechanism to let the user or an outer optimization algorithm easily vary the problem configuration. For instance, optimizing an airfoil's shape with respect to key aerodynamic indicators like the lift-to-drag ratio; or optimizing a geometry's parameterization as per (problemspecific) quality metrics.

Isogeometric Collocation Methods

Finite Element Analysis (FEA) and its extension to B-Splines and NURBS, Isogeometric Analysis (IGA) [2], are established approaches to analyse BVPs through detailed numerical simulation. In both cases, the approximate solution $u^h \approx u$ is represented in terms of an a-priori chosen basis expansion of the form

$$u^{h}(x, y) = \sum_{j=1}^{N_{b}} B_{j}(x, y) u_{j}, \qquad (7.4)$$

where B_j are the B-Spline basis functions and u_j the corresponding basis coefficients. While variational formulations are widely used, collocation methods [4] haverecently gained some popularity, especially in IGA. This is because of the higher continuity of B-Spline basis functions that allows to substitute (7.4) into the strongform of (7.1) and apply the differential operator to each basis function individually. Let $(x_i, y_i) \in \overline{\Omega}$ be collocation points such that the system matrix is invertible then the vector of solution coefficients can be computed from the system of equations

$$\sum_{j=1}^{N_b} \mathcal{L}B_j(x_i, y_i) u_j = f(x_i, y_i) \quad \forall i = 1, \dots, N_b.$$
(7.5)

The imposition of Dirichlet boundary conditions ($\mathcal{B} = id$) is straightforward, whereas other boundary conditions that involve derivatives of the solution require a more sophisticated approach [5] to suppress the generation of spurious oscillations when the mesh is non-uniform and the solutions exhibits only reduced regularity.

IGA-PINNs

We suggest a novel physics-informed machine learning approach that combines isogeometric collocation methods with PINNs. Following the IGA paradigm, let us approximate f, g and Ω in the form (7.4), for simplicity with the same B-Spline basis. The resulting coefficients f_j , g_j and $x_j = (x_j, y_j)$ serve as additional inputs to the PINN and will allow us to explore different problem configurations *after* the training phase has completed. Furthermore, the solution coefficients u_j from (7.5) become the network's outputs and the loss function (7.3) is modified as follows

$$\ell_{\mathcal{L}}(u) := \frac{\alpha}{N_{\Omega}} \sum_{k=1}^{N_{\Omega}} |\sum_{j=1}^{N_{h}} \mathcal{L}B_{j}(x_{k}, y_{k})\hat{u}_{j} - B_{j}(x_{k}, y_{k})f_{i}|^{2},$$

$$\ell_{\mathcal{B}}(u) := \frac{\beta}{N_{\Gamma}} \sum_{k=1}^{N_{\Gamma}} |\sum_{j=1}^{N_{h}} \mathcal{B}B_{j}(x_{k}, y_{k})\hat{u}_{j} - B_{j}(x_{k}, y_{k})g_{j}|^{2}.$$
(7.6)

Note that the domain Ω^h is defined by the following 'push-forward' map

$$\begin{pmatrix} x \\ y \end{pmatrix} = \mathbf{F}(\xi, \eta) := \sum_{j=1}^{N_b} \hat{B_j}(\xi, \eta) \begin{pmatrix} x_j \\ y_j \end{pmatrix}, \quad \begin{pmatrix} \xi \\ \eta \end{pmatrix} \in \hat{\Omega} := [0, 1]^2, \quad (7.7)$$

where $\hat{B}_j(\xi, \eta) := \hat{B}_{j1}(\xi) \otimes \hat{B}_{j2}(\eta)$ are bivariate tensor-product B-Splines defined on the reference domain $\hat{\Omega}$ The basis functions in (7.3) are then evaluated with the aid of the inverse 'pull-back' map, that is, $B_j(x_k, y_k) = \hat{B}_j \circ \mathbf{F}^{-1}(x_k, y_k)$. This parametric approach makes choosing the spatial sampling points *inside* Ω^h and *exactly on* Γ^h straightforward even for complex geometries as it suffices to select $0 \le \xi_k$, $\eta_k \le 1$ and map them into the physical domain with the push forward (7.7).

The schematics of our IGA-PINN is depicted in Fig. 7.1. Algorithmic differentiation is required to differentiate the loss function with respect to the network's weights *w* and biases *b*, i.e. $\frac{\partial \ell}{\partial(w,b)}$, to update the latter during the offline training phase, whereas the

Fig. 7.1: Schematic of isogeometric physics-informed neural networks (IGA-PINNs). A family of different problem configurations is encoded by varying the B-Spline coefficients of the geometry, source term, and the boundary conditions. Batch training takes place over the primary inputs, $\xi = (\xi, \eta)$.



Fig. 7.2: IGA-PINN solutions to a 2D Poisson problem on a half-quarter annulus with homogeneous (left) and non-homogeneous (right) Dirichlet boundary conditions.



evaluation of the loss function (7.6) uses the predictions \hat{u}_j 'as is' and adopts explicit expressions for the derivatives of the B-Spline basis functions.

Obviously, the resulting solution field (7.4) stays within the space spanned by the B-Spline basis functions by design and can be evaluated for all admissible coordinates (*x*, *y*). Fig. 7.2 shows the predicted solution for a two-dimensional Poisson problem ($\mathcal{L} = -\Delta$), where $f \equiv I$ and Ω (half-quarter annulus) have been kept fixed and the network has been trained for many different values for g_k with \mathcal{B} = id (Dirichlet boundary conditions), among them homogeneous Dirichlet values; cf. Fig. 7.2 (left). The solution on the right corresponds to boundary coefficients that the network has not seen during training. This linear elliptic problem can also be solved with simpler approaches (e.g., IGA collocation) but it serves as a demonstrator for IGA-PINNs. Forthcoming work will discuss nonlinear extensions (e.g., nonlinear PDEs, geometric parameterization changes), hyperparameter tuning, approximation errors and the computational efficiency.

References

- M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378:686–707, 2019.
- [2] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. Computer Methods in Applied Mechanics and Engineering, 194(39):4135–4195, 2005.
- [3] George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. Nature Reviews Physics, 3(6):422–440, May 2021.
- [4] Alessandro Reali and Thomas J. R. Hughes. An Introduction to Isogeometric Collocation Methods, pages 173–204. Springer Vienna, Vienna, 2015.
- [5] L. De Lorenzis, J.A. Evans, T.J.R. Hughes, and A. Reali. Isogeometric collocation: Neumann boundary conditions and contact. Computer Methods in Applied Mechanics and Engineering, 284:21–54, 2015. Isogeometric Analysis Special Issue.

GEOMETRIC DEEP LEARNING FOR PRECISION MEDICINE

JELMER WOLTERINK AND JULIAN SUK

Abstract Artificial intelligence (AI) is transforming scientific disciplines at a highpace. Convolutional neural networks are the workhorse of many AI methods but arelimited to data in Euclidean space, such as images. Instead, AI on graphs and triangular meshes requires the development of graph neural networks. We describe graphneural networks and their application to computational fluid dynamics surrogate modeling for personalized assessment of cardiovascular disease.

Introduction

Over the past few years, artificial intelligence (AI) techniques have taken many scientific fields by storm. These advances have to a large extent been driven by convolutional neural networks (CNNs), which consist of convolutional layers that apply trainable discrete convolution filters to – typically – images. For an image, convolution can be defined as

$$(k(\cdot) * f)_p := \sum_{q \in N(p)} f_q \cdot k(q), p \in I$$

where *p* is a pixel in the image, f_p is the pixel's intensity value, and $k(\cdot)$ is a filter kernel with trainable weights. Weights k(q) in the filter kernel indicate the contribution of each pixel in a local neighborhood N(p) with fixed size, where *q* indexes the neighborhood pixels. Through weight sharing, the same filter kernel is applied

at each pixel in the image. To find suitable values for $k(\cdot)$, CNNs are trained by minimizing a loss function through stochastic gradient descent or one of its variants. CNNs are extremely valuable for the analysis of data in Euclidean space, such as 1D time signals, 2D images, or 3D medical image volumes. Convolution layers can be stacked and combined with nonlinearities for tasks like image classification, localization, and segmentation, without the need for handcrafted features.

Despite their successes, CNNs are not readily applicable to data in non-Euclidean domains such as graphs. Two main assumptions of CNNs do not hold for graphs. First, the number of vertex neighbors can vary significantly within a graph. Hence, kernels as in Eq. 8.1 with a fixed number of neighbors cannot be used. Second, there is no canonical ordering of neighboring nodes and thus, kernel weights have no ordering. Nevertheless, leveraging advances in deep learning to graphs has many potential applications, not in the least in precision medicine. A promising development is that of *geometric deep learning*, in which graph neural networks (GNNs) are developed to specifically operate on graphs [1].

Graph neural networks

A GNN operating on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can be described as performing spatial convolution, where a trained filter is applied to each vertex $p \in \mathcal{V}.\, {\rm Let} f_p \in \mathbb{R}^{\mathcal{C}}$ be a feature vector. Then new features for p can be obtained as

$$((K^1, K^2(\cdot, \cdot)) * f)_p := f_p \cdot K^1 + \sum_{q \in N(p)} \rho(p, q) f_q \cdot K^2(p, q), p \in \mathcal{V}$$
(8.1)

where K^1 , $K^2 \in \mathbb{R}^{C \times \overline{C}}$ are trainable kernel matrices with c input features and \overline{c} output features, and $\rho(p, q) \in \mathbb{R}^{C \times C}$ determines the contribution of each neighbor q to the new features of p. Depending on the choice of $K^2(p, q)$ and $\rho(p, q)$, different GNN layers can be retrieved. These range from *isotropic* kernels that do not distinguish between different neighbor nodes by taking $K^2(p, q) = \frac{1}{|N(p)|} \overline{K}^2$ and $\rho(p, q) = id$, to *anisotropic* kernels that consider the individual neighbors of the graph vertex. For example, attention mechanisms $\rho(p,q) = \sigma(w \cdot (f_q - f_p))$ id with trainable weights $w \in \mathbb{R}^C$, and "softmax" $\sigma(\cdot)$. Like in CNNs, weight in GNNs are iteratively optmized. A major difference with CNNs operating on fixed grids is that, in inductive learning, the graph structure on which the GNN operates can vary from sample to sample.

Medical applications

Our group performs research on geometric deep learning techniques and their application to medical problems. Applications of geometric deep learning can be found in graphs and (discretized) manifolds extracted from medical images. For example, GNNs can be used to directly obtain a watertight triangular mesh segmentation of the coronary artery wall from computed tomography (CT) images [5]. Moreover, blood vessels in the human body can be considered graphs, with vessel segments as edges and branching points as vertices. GNNs provide a means to perform nodeclassification in such graphs for automatic labeling of blood vessel segments [3]. We have recently found that GNNs can also play a role in the quantification of blood flow in patients suffering from cardiovascular diseases, such as the magnitude and direction of wall shear stress (WSS) in the case of stenoses and aneurysms [4].





WSS cannot be measured directly but is typically estimated using computational fluid dynamics (CFD). This requires the extraction of an artery model from e.g.CT images, the meshing of this model, and iterative solution of the Navier-Stokes equations within the mesh. Especially this last step can be very time-consuming.

We found that instead of running CFD simulations for each new blood vessel, CFD simulations obtained in a training set can be used to train a GNN that directlypredicts 3D WSS vectors on the triangular mesh describing the vessel wall. By choosing input features on the graph vertices that locally describe the mesh geometry,and anisotropically defining $K^2(p, q)$ and $\rho(p, q)$ in Eq. 8.1[2], a GNN can be made equivariant with respect to the rotation group *SO*(3). This means that rotating the input mesh in 3D rotates the output vectors accordingly. Moreover, this GNN is translation-invariant, i.e., translating the mesh does not affect the output vectors. These are desirable properties for blood flow quantification.

Fig. 8.1 shows GNN predictions in two unseen synthetic triangular meshes representing a single vessel and a bifurcation. These examples show that the GNN is ableto estimate directional WSS vectors that closely resemble those obtained by CFD. Interestingly, while the structure of the data is globally different – we consider both individual vessels and bifurcating vessels – at the local level they consist of vertices and edges. Hence, a single GNN architecture can be applied to both these structures.

Outlook

The estimation of hemodynamic parameters in GNNs has potential applications in arange of cardiovascular diseases. For example, we collaborate closely with vascularsurgeons to further develop these models for patients suffering from abdominalaortic aneurysms (AAAs). AAAs are extremely lethal if left untreated, and rapidand accurate estimation of hemodynamic parameters based on GNNs could lead topatient-specific decision making, instead of currently used one-size-fits-all protocols. To achieve true value in precision medicine requires technology that generalizes well to new and unseen patients. Training data is often sparse in medical applicationsand, therefore, the efficient use of training data is key. Here, this is obtained by incorporating the right symmetries in our models. In follow-up work, we aim to explicitlyincorporate physics, in this case, the Navier-Stokes equations, during training formore efficient use of sparse training data. Our work shows how the combination ofcomputational science and artificial intelligence could have a real-world impact on precision medicine.

Acknowledgement This work is funded in part by the 4TU Precision Medicine program supported by High Tech for a Sustainable Future, a framework commissioned by the four Universities of Technology of the Netherlands. We thank our collaborators at the Qualcomm UvA (QUVA) lab.

References

- Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.
- [2] Pim de Haan, MauriceWeiler, Taco Cohen, and Max Welling. Gauge equivariant mesh CNNs: anisotropic convolutions on geometric graphs. In International Conference on Learning Representations, 2021.
- [3] Nils Hampe, JelmerMWolterink, Carlos Collet, Nils Planken, and Ivana Išgum. Graph attention networks for segment labeling in coronary artery trees. In *Medical Imaging 2021: Image Processing*, volume 11596, page 1159611, 2021.
- [4] Julian Suk, Pim de Haan, Phillip Lippe, Christoph Brune, and Jelmer M Wolterink. Mesh convolutional neural networks for wall shear stress estimation in 3D artery models. In *MICCAI STACOM 2021*, In press.
- [5] Jelmer M. Wolterink, Tim Leiner, and Ivana Išgum. Graph convolutional networks for coronary artery segmentation in cardiac CT angiography. In *Graph Learning in Medical Imaging*, 2019.

GEOMETRIC LEARNING FOR IMAGE ANALYSIS

REMCO DUITS

Abstract We develop a new geometric deep learning framework for convolutionalneural networks (CNNs) with a firm mathematical foundation, relying on partial differential equations (PDEs), differential geometry, Lie group analysis, and probability theory. It impacts mathematics by providing new fundamental theorems on homogeneous spaces. It also impacts medical image analysis, solving major challenges intracking and enhancement of complex vasculature.

Current geometric algorithms allow for geometric understanding of image analysis, but often fail at complex line structures (crossings, bifurcations), requiring costly user-interaction. Deep learning algorithms via CNNs perform superbly on specific datasets, but require massive annotations for training, lack geometric modelinterpretability, and fail to hard-code necessary equivariances.

I aim to bridge geometrical data-processing and machine learning based dataprocessing. To cope with complex structures in tracking and enhancement we lift image data to higher dimensional homogeneous spaces. To reduce manual input weuse geometric PDE-based data processing and training. To reduce highly redundantannotations we develop new equivariant CNNs arising from operator splitting of geometric PDEs. To underpin our algorithms I establish new theories on homogeneous spaces, and validate our algorithms in automatic enhancement and tracking of complex line structures in medical images, where reduction of user-input is crucial. We tackle this by applying our geometric learning algorithms on (multi-orientation) imagerepresentations on homogeneous spaces. It generalizes fundamental results on analysis, geometry, and probability theory to homogeneous spaces, and produces new geometric PDE-based CNNs and powerful image analysis algorithms that overcomecostly user-interactions and annotations.

Overall aim and key objectives

Worldwide, much research is ongoing to improve image analysis techniques to targetindustrial and medical image analysis. Stateof-the-art algorithms have a limited application scope, require costly user-interaction, while automatic analysis often fails at complex structures due to destructive interference of multiple target patternspresent at the same location (e.g. the crossing blood vessels in Fig. 9.1).

Current geometrical algorithms [14, 26, 40, 62, 77] in image analysis for imagedenoising, tracking or segmentation are mathematically well-founded by PDEs andODEs, but are limited in their ability to cope with anatomically complex structures(crossings/bifurcations), and require costly and subjective user-interaction. E.g. thewrong exit in Fig. 9.1(left) requires the user to note and correct this, cf. Fig. 9.1(right). To tackle such problems, the domain of images is extended to higher dimensional spaces, cf. Fig. 9.1(middle), so-called 'homogeneous spaces' as we explain later.

Current deep learning via convolutional neural networks (CNNs) [39, 52, 53, 54, 55, 68] performs superbly on specific tasks like vessel segmentations on specific datasets. Unfortunately, it requires massive annotated datasets for training, optimizes inefficiently over huge parameter spaces, does not guarantee equivariance (symmetries w.r.t. group actions, e.g. roto-translations), and lacks geometric model interpretability. This calls for a geometric embedding that allows for interpretability of the results, complexity reduction, and built-in equivariance.

Recently, inclusion of equivariance into the design of CNNs on homogeneous spaces has had significant success [6, 13, 21, 22, 23, 25, 48, 49, 71, 79], but suffers from limitations (on sampling, activations, convolutions, max-pooling, separability, holonomicity). Moreover, all of these works lack a fundamental geometric PDE description of network dynamics. In order to tackle above problems in geometric and deeplearning methods we investigate a new geometric, equivariant, PDE-based learning framework for automatic tracking and enhancement of complex vasculature in medical images that:

- combines new geometrical path optimization (for automatic tracking of com-plex line structures), with new crossing-preserving PDE flows (for image enhancement), and machine learning on homogeneous spaces,
- generates a new mathematical PDE-framework for geometrical deep learning where geometrically meaningful equivariant PDE-coeÿcients are trained in new equivariant CNNs on homogeneous spaces.

This integrates the established machine learning framework of equivariant CNNs with state-of-the-art equivariant geometric PDEbased image processing frameworks [3, 5, 29, 30, 31, 33, 34, 36, 44, 46, 56, 65, 81] that allows us to deal with complex structures in noisy images in a data-adaptive and automatic way, thereby reducing the need for costly user interaction.

Fig. 9.1: Current tracking algorithms on images often fail at locations where multiple lines meet (left). We avoid such collisions by 'lifting' the image to the homogeneous space of positions and orientations (where no crossings occur) and applying automated geodesic tracking, enhancement and learning for analysis (middle) prior to back-projection (right).



We close the gap between deep learning based image processing (that needs geometric model interpretability) and geometric PDE-based image processing (that needs automation).

A key aspect of my approach is to expand the domain of images from \mathbb{R}^d to homogenous spaces. A homogeneous space is a manifold on which a symmetry group *G* acts transitively. For example the homogeneous space of positions and orientations $\mathbb{M}_d := G/H$, with roto-translation group G = SE(d) := $\mathbb{R}^d \rtimes SO(d)$ and a subgroup *H* containing rotations about a reference axis. Setwise, one has $\mathbb{M}_d = \mathbb{R}^d \times S^{d-1}$, where S^{d-1} denotes the sphere of dimension d - 1, [28, 35]. It is the domain of an 'orientation score' [33] that encodes all orientations per position in the image. Such a score allows for improved image analysis algorithms, involving geometrypreserving image denoising and geodesic tracking via PDEs on \mathbb{M}_d (Fig. 9.1). It has had impact in image analysis [3, 5, 30, 31, 34, 44, 46, 56, 65, 81], mathematics [12, 28, 29, 33, 35, 36, 64, 74] and numerics [24, 42, 66, 82], and boosted multi-orientation image processing, see e.g. [9, 20, 58, 61, 62, 71, 80], However, it requires automation to overcome geometric hand-crafting.

We aim for new geometric control and probability theory, and roto-translation equivariant PDE-based deep learning via PDE-CNNs on \mathbb{M}_d . See Fig. 9.2 for the goal of automatic vessel tracking and denoising, and see Fig. 9.3 for automatic vessel segmentation via the visualized PDE-based CNNs. The proposed PDE-based deep learning is equivariant w.r.t. roto-translations and beyond PDE-CNNs [2, 67] on \mathbb{R}^d . More specifically, we will tackle:

Fig. 9.2: Top: instead of direct image processing, we process via an orientation score, obtained by convolving the image with rotated wavelets [3, 31, 46]. 2nd row: vessel-tracking in a 2D image via orientation scores [36]. 3rd row: crossing-preserving diffusion via the orientation score of a 3D image [46]. To overcome the 'hand-crafting' in previous works [15, 16, 36] and to include more long-range interactions, I propose to replace it by geometric deep learning via PDE-Based CNNs.



Challenge I. New Foundations for Geometric Learning on Homogeneous Spaces:

- a Set up axiomatic foundations for PDE-CNNs building on convolution network equivariances [6,22,28], and on PDE-based image processing [1, 28, 32, 70, 73].
- b Construct PDE-CNNs on homogeneous spaces, with advanced variants incl. mean curvature and TV flows [15, 19, 74] and erosion. Here network-weights have a clear geometric, probabilistic interpretation.
- c Generalize Central Limit Theorem on \mathbb{M}_d beyond [8] to account for infinite variance, nonholonomicity. Limits are α -stable Lévy processes [28] with Bellman processes [1] describing the dynamics of (PDE-)CNNs.
- d Develop a Cramér transform on \mathbb{M}_d , with tractable local

approximations. It sets an isomorphism between parts of PDE-CNNs, and gives analytic kernels for fast, accurate implementations [72].

Challenge II. Equivariant, PDE-based Processing and Machine Learning for Image Analysis:

- a Create and implement algorithms for equivariant PDE-CNNs (via convolutions [72], stencils [42], or B-splines [4]), and publish them as open source extensions to the PyTorch machine learning framework.
- b Analyze and improve Total Variation & Mean Curvature PDE flows and construct edge-enhancing filtering PDE flows on \mathbb{M}_d and integrate them in PDE-CNNs, by generalizing median filtering [43, 69] to \mathbb{M}_d .

Fig. 9.3: Illustration of PDE-based CNNs for vessel segmentation. Here convection and diffusion flows (for transport and regularization) will be implemented by linear group convolution, dilation (for max-pooling) and erosion (for data-sharpening) by morphological group convolutions [72]. The PDEs allow for a systematic geometric design of neurons in the network. The novelty is the embedding of PDE-based CNN ('PDE layers' augmented in the bottom part) into the orientation score paradigm [3, 27, 33]. It will automate data-driven orientation score processing and construction, and allows for a higher level of pattern-formation in contour perception [22, 52, 68, 76].



c Create theory, algorithms for data-driven improvements of our widely applicable [38] Cartan connections on M_d, and include it in: 1) our PDE-CNNs, 2) our geometric control/tracking [29, 36].

Challenge III. Automatic geometric vessel tree tracking and analysis, and crossingpreserving connectivity measures, in medical image analysis applications in collaboration with healthtech and clinical partners:

- a Apply our PDE-CNNs to improve vessel tree segmentation [72], and detection of diseases/landmarks [6, 51].
- b Automate vascular tree tracking with graph representation splitting at bifurcations building on [5, 7, 16, 36].
- c Improve geodesic tracking in \mathbb{M}_d via optimized, trained Cartan connections
- d Develop structural connectivity measures between anatomical regions via crossingpreserving geodesic wavefronts [17, 36, 59] in M₃ and include stabilization and local data-adaptation.

The objectives for our geometric learning framework

The overall objective is to build a well-understood PDE-based geometric learning framework on the space of positions and orientations that:

- deals with complex line-structures and complex geometries in images,
- accounts for local and global interactions between patterns in contour perception,
- expands and improves existing equivariant convolutional neural networks where we include probabilistic and geometric model interpretability,
- provides valuable tools for automatic vessel tracking and connectivity,
- bridges the gap between modern, effective heuristic machine learning methods, and state-of-the-art geometrical methods in image processing.

The 3 specific objectives of the project are:

- 1. Set up PDE-based geometric learning theory and algorithms for roto-translation equivariant PDE-based CNNs on \mathbb{M}_d and tackle medical imaging challenges (denoising, vessel segmentation).
- 2. Understand the effective CNN operators on \mathbb{M}_d and analyze their limiting behavior via isomorphisms between sub-parts.
- 3. Set up new theory and algorithms for optimal paths on \mathbb{M}_d that include new geometric and statistical data adaption, and tackle medical imaging challenges. Besides automatic tracking (Fig. 9.2) and segmentation (Fig. 9.3) of vessel trees in Xray and optical images, we also consider quantification of structural connectivity of the brain in DW-MRI [63].

In tackling these objectives via the earlier formulated challenges, we follow the flowchart depicted in Fig. 9.4.

Design Principles and Mathematical Background

In this subsection we provide mathematical background on 3 design principles that describe the overall architecture depicted in Fig. 9.3. The 3 principles are:

- A Lift images from \mathbb{R}^d to \mathbb{M}_d , see Figs. 9.1, 9.2 and 9.5.
- B Develop new PDE-based processing on the lifted data, see Fig. 9.2.
- C Develop new PDE-based CNNs for geometric learning on lifted data, see Fig. 9.3.

Principle A: New Dimensions: Lift images from \mathbb{R}^d to \mathbb{M}_d

In order to disentangle all local orientations in an image we lift the data from position space \mathbb{R}^d to the homogeneous space \mathbb{M}_d of positions and orientations, This means that we extend the domain of images. See Fig. 9.2, where we 'lift' the data from \mathbb{R}^d towards \mathbb{M}_d via invertible orientation scores. Such a lifting should not a priori temper data-evidence. This requires a special design such that stable reconstruction is possible (preferably by an intuitive integration over all angles, per position).



An orientation score $\mathcal{W}_{\psi}f$ reveals how an image is decomposed out of local orientations. It is obtained by probing image $f \in \mathbb{L}_2(\mathbb{R}^d)$ by a family of groupcoherent wavelets:

$$\mathcal{W}_{\psi}f(\mathbf{x},\mathbf{n}) = \int_{\mathbb{R}^d} \overline{\psi(\mathbf{R}_{\mathbf{n}}^{-1}(\mathbf{y}-\mathbf{x}))} f(\mathbf{y}) \, \mathrm{d}\mathbf{y},$$

for all $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{n} \in S^{d-1}$, $d \in \{2, 3\}$ (9.1)

for all roto-translations $g = (\mathbf{x}, \mathbf{R}_n) \in SE(d) = \mathbb{R}^d \rtimes SO(d)$, where \mathbf{R}_n is any rotation that maps a fixed reference orientation $\mathbf{a} \in S^{d-1}$ onto \mathbf{n} . See Fig. 9.5

For d = 3, wavelet ψ is axially symmetric around **a** so thereby the choice of **R**_{**n**} is irrelevant. Thereby, an orientation score extends the image domain to the Lie group quotient $\mathbb{M}_d := G/H$ with roto-translation Lie group G = SE(d) and subgroup $H = \{g \in SE(d) \mid g.(\mathbf{0}, \mathbf{a}) = (\mathbf{0}, \mathbf{a})\}$ and Lie group action:

$$(\mathbf{x}, \mathbf{R}).(\mathbf{y}, \mathbf{n}) := (\mathbf{x} + \mathbf{R}\mathbf{y}, \mathbf{R}\mathbf{n}), \tag{9.2}$$

for all $(\mathbf{y}, \mathbf{n}) \in \mathbb{R}^{d \times S^{d-1}}$ and all roto-translations $(\mathbf{x}, \mathbf{R}) \in SE(d)$. The invertibility of transform \mathcal{W}_{ψ} requires the use of proper [27, 46] wavelets ψ , which have equal 'mass' over the domain of each unitary irreducible representation of *SE*(*d*), cf. [33, App.A]. These domains are spheres in the Fourier domain.

The question arises how to train proper wavelets ψ . I.e. how do the training data tell us to optimally distribute the mass of proper wavelets over all spheres in the Fourier domain up to a (Nyquist) radius?

We tackle this by equivariant CNNs in (**I-b**) on data with ground truth vessel segmentations and where we include an invertibility constraint [27, 33] in the 1st layer of equivariant CNNs [13, 22, 48, 71]. Then the invertibility ensures that our subsequent denoising PDE evolutions and associated machine learning networks, will start at time t = 0 from the original image without a priori data damage. Fig. 9.5: We first lift the original image to an invertible orientation score (9.1) and enlarge the image domain to the space of positions and orientations \mathbb{M}_d . In this lifting procedure we will now train these proper wavelets ψ that preserve data-evidence before applying PDE-based deep-learning on the orientation score.



Principle B. Develop new geometric PDE-based processing on the lifted data.

For geometrical flows in the lifted data that allow for stable crossing-preserving enhancement and tracking of vasculature in medical images we rely on equivariant geometric PDEs on orientation scores. Such PDEs on orientation scores also underlie models of contour perception in human vision [11, 20, 61]. We categorize into:

B1 enhancement via left-invariant PDE's on orientation scores,

- **B2** tracking of blood vessels/fibers in orientation scores via optimal control problems. See Figure 9.3.
- **B1**: For enhancement of orientation scores $\mathcal{W}_{\psi}f$ we apply an equivariant PDE

convection fractional diffusion dilation/erosion

PDE: $\frac{\partial W}{\partial t} = -\mathbf{c} \cdot \nabla_{\mathcal{G}_1} W + |\Delta_{\mathcal{G}_2}|^{\alpha} W \pm ||\nabla_{\mathcal{G}_3} W||^{2\alpha}, \text{ on } \mathbb{M}_d \times \mathbb{R}^+, \\ W|_{t=0} = \mathcal{W}_{\psi} f \qquad \text{ on } \mathbb{M}_d, \quad (9.3)$

where the orientation score (9.1) serves as an initial condition, and where the gradient $\nabla \mathcal{G}_1 W$ is relative to a left-invariant metric tensor \mathcal{G}_1 on \mathbb{M}_d , and where $\Delta_{\mathcal{G}}$ is a leftinvariant Laplacian for diffusion indexed by a highly anisotropic (or sub-Riemannianwith infinite anisotropy; see Fig. 9.6) left-invariant metric tensor \mathcal{G}_2 . For the dilation(+ case) and erosion (- case) part we use another metric tensor \mathcal{G}_3 , see Remark 9.1.1.The convection (via transport vector c) takes care of equivariant transport, the fractional diffusion for equivariant regularization, and the dilation for equivariant geodesic front propagation, and the erosion for datasharpening.

Parameter $\alpha \in [\frac{1}{2}, 1]$ denotes an α -scale space parameter [32] for α -stable Lévy processes [41] that are the limiting distributions in the central limit theorem (CLT) describing the limits of iterative convolution processes, while dropping the constraint of finite 2nd order moments. This is well-known on \mathbb{R}^d [41] and applies also to \mathbb{M}_d .

Remark 9.1.1 Geometric parameter reduction. The metric tensors must induce equivariant image processing and must be well-defined on \mathbb{M}_d . This allows for geometric parameter reduction [73, cor.2.7]. E.g. If data-adapation is omitted one must use [73, Prop.2.8] the default metric tensors:

$$(\mathcal{G}_m)_{\mathbf{p}} \left(\dot{\mathbf{p}}, \, \dot{\mathbf{p}} \right) = \xi^2 |\dot{\mathbf{x}} \cdot \mathbf{n}|^2 + \| \, \dot{\mathbf{n}} \, \|^2 + \xi^2 \epsilon_m^{-2} \, \| \, \dot{\mathbf{x}} \wedge \mathbf{n} \, \|^2, \tag{9.4}$$

for all $\mathbf{p} = (\mathbf{x}, \mathbf{n}) \in \mathbb{M}_d$, $\dot{\mathbf{p}} = (\dot{\mathbf{x}}, \dot{\mathbf{n}}) \in T_{\mathbf{p}}(\mathbb{M}_d)$, for $m \in \{1, 2, 3\}$. We set the following anisotropy principles:

For m = 1 (convection) we set $\epsilon_1 = \xi = 1$.

For m = 2 (diffusion) and m = 3 (dilation) we set $0 < \epsilon_2 \ll 1$ and investigate the infinite anisotropy case $\epsilon_2 \downarrow 0$, in order to diffuse/dilate only along lines.

For m = 3 (erosion), i.e. the - case in (9.3), we set

 $\xi \gg 1$ and $\xi/\epsilon_3 \ll 1$, in order o sharpen the lifted data on \mathbb{M}_d spatially only orthogonal to the lines.

Furthermore, data-driven locally adaptive frames (LAFs) (II-c) can be included forbetter data-adaptation and results [38]. In Principle C we will train the metric tensors and investigate if the data indeed supports these geometric expectations.

B2: For the tracking we apply geodesic wavefront propagation from a compact connected source set $S \subset M_d$ by efficient [36, 58] computation of the distance map. This distance map is the viscosity solution of eikonal PDE system:

$$\forall \mathbf{p} \in \mathbb{M}_d \setminus \mathcal{S} : \mathcal{F}^*(\mathbf{p}, dW(\mathbf{p})) = 1, \text{ and } \forall \mathbf{p}_0 \in \mathcal{S} : W(\mathbf{p}_0) = 0$$
 (9.5)

defined on \mathbb{M}_d with dual $\mathcal{F}^*(\mathbf{p}, \cdot)$ of a Finsler function $\mathcal{F}(\mathbf{p}, \cdot)$ with $\mathbf{p} = (\mathbf{x}, \mathbf{n})$ that defines the (quasi)-distance:

$$d_{\mathcal{F}}(\mathbf{p}_{S}, \mathbf{p}_{E}) = \min_{\substack{\gamma \in \operatorname{Lip}([0, 1], \mathbb{M}_{d}), \\ \gamma(0) = \mathbf{p}_{S}, \gamma(1) = \mathbf{p}_{E}}} \int_{0}^{1} \mathcal{F}(\gamma(t), \dot{\gamma}(t)) dt.$$
(9.6)

Back-tracking of globally optimal geodesics $\gamma_{min}(\cdot)$ in (9.6) is done by an intrinsic gradient descent on the distance map, cf. [36, Thm.4]. In medical image processingthe goal is that γ_{min} follows blood vessels (III-b,c), or fibers (III-d), cf. [36, 63].

The primary objective (II-c) is to geometrically optimize and to train Finsler function \mathcal{F} in order to outperform existing geometric control approaches on \mathbb{M}_d , such that they 1. can account for high curvatures, 2. do not suffer from bias towards sampled orienta-

tions, 3. deal with corners and bifurcations, and 4. include statisticaldata-adaptation. We tackle item 1 and 2 by including dataadaptivity via LAFs andCartan connections [38]. We tackle item 3 by my asymmetric Finsler geometry [36]. We tackle item 4 via (I). The proposed form for the Finsler function \mathcal{F} is as follows.

Given the orientation score $U: \mathbb{M}_d \to \mathbb{C}$ we compute symmetric, positive definite, matrix field $\mathbf{p} \mapsto \mathbf{A}(\mathbf{p})$ encoding local directions/curvatures in U, (cf.[74]) and set:

$$\mathcal{F}(\mathbf{p}, \mathbf{\dot{p}}) = \mathcal{M}(\mathbf{p}) \sqrt{\nu \frac{\mathcal{G}_{1|\mathbf{p}}(\mathbf{\dot{p}}, \mathbf{A}(\mathbf{p}) \ \mathbf{\dot{p}})}{|\lambda_{min}(\mathbf{p})|} + (1-\nu) \mathcal{G}_{2|\mathbf{p}}(\mathbf{\dot{p}}, \mathbf{\dot{p}}) + \text{RMP}(\mathbf{p}, \mathbf{\dot{p}})}} (9.7)$$

for all $\dot{\mathbf{p}} := (\dot{\mathbf{x}}, \dot{\mathbf{n}})$ tangent vectors at base point $\mathbf{p} = (\mathbf{x}, \mathbf{n}) \in \mathbb{M}_d$, with $v \in [0, 1]$.

This Finsler function will yield optimal curvature adaptation of tracking of minimizing geodesics with in-place rotations above bifurcations/corners, with statistical data-adaptation in mobility $\mathcal{M}(\mathbf{p})$. The term RMP (\mathbf{p} , $\dot{\mathbf{p}}$) stands for 'reverse motion punishment' which forbids backward motions for explicit formulas see [36].

Fig. 9.6: Left: A car can only move in its current orientation or change its current orientation.

I.e. when the lifted path $\gamma(t) = (x(t), y(t), \theta(t))$ is considered, the tangent $\dot{\gamma}(t)$ is restricted to the span of left-invariant vector fields $\cos \theta(t)\partial_x + \sin \theta(t)\partial_y$ and ∂_θ , of which the green plane on the right is an example. Right: the meaning of shortest path between points in an image is determined by distance (9.6). The path optimization is formulated on $\mathbb{M}_2 \equiv \mathbb{R}^2 \times S^1$ as shown on the right. The cost for moving through the orange parts is lower than elsewhere, and is induced by the orientation score [36]. Our PDE-based CNNs [73] allow us to train this cost.


Fig. 9.7: Left: Conventional wavefront propagation in the image domain, leak at crossings, which can result in wrong exits (like in Fig. 9.1. In contrast wavefront propagation in orientation scores do not suffer from this problem. In green we show opaque geodesic wavefronts (growing spheres) prior to a steepest descent yielding optimal geodesics. In orange we show the mobility \mathcal{M} in (9.7). Middle: the symmetric case where the 'Reeds-Shepp' car (cf. Fig. 9.6) can use a reverse gear. Right: the asymmetric case without reverse gear.



Principle C. Geometric Learning on the lifted data via PDE-based CNNs

We want to achieve model interpretability of deep-learning models via equivariant CNNs. Equivariant CNNs on Lie group quotients [6, 13, 21, 22, 23, 25, 48, 49, 71, 79] have been effective yielding state-of-the-art results, but depite their group structure, they are still a 'black box' if it comes to model interpretability. We propose PDE-based equivariant CNNs on \mathbb{M}_d that we explain next.

Usual CNNs on \mathbb{Z}^d [52] iterate 1) possibly off-centered convolution kernels, 2) max-pooling, 3) regularization, 4) smoothed rectifying units (ReLU's). We see them as sampled operator splittings of a PDE evolution combining convection, *fractional* morphology and diffusion on \mathbb{R}^d , with convection-vectors for training center off-sets.

Next we replace \mathbb{R}^d by \mathbb{M}_d and obtain equivariant PDE-based CNNs on \mathbb{M}_d , where network variables ('weights') have a clear geometric and probabilistic interpretation, as they are PDE coefficients of the equivariant PDE systems (9.3). This allows for a geometric design and reduction of networks. For a geometric visualization of my viewpoint see Fig. 6 and 7 in our work [73].

We consider two types of our geometric learning approaches in \mathbb{M}_d :

C1 geometric PDE-based deep convolutional neural networks (CCNs) (I). They naturally arise as PDE operator splitting of our key PDEs, recall (9.3).

C2 training the geometric parameters in the Finsler functions (9.7) for (vessel) tracking (V).

C1. The depth of the PDE-G-CNN network is created by $\underline{\Phi}_l$ which maps $U_l \in (\mathbb{L}_2 (\mathbb{M}_d))^{N_l}$ to $U_{l+1} \in (\mathbb{L}_2 (\mathbb{M}_d))^{N_{l+1}}$

$$U_{l+1} = \underline{\Phi}_l(U_l), \text{ with } U_{l+1}^{k'} \coloneqq \sum_{k=1}^{N_t} \omega_{k'k}^l \cdot \underline{\Phi}_t^k(U_l^k), \qquad (9.8)$$
and $\underline{\Phi}_t^k(U) \coloneqq \chi(W(\cdot, t))$

for 'depth' parameter $l = 0, \ldots, L-1$, and 'width' parameter $k' = 1, \ldots, N_{l+1}$, and with total width $N_l \in \mathbb{N}$, and with weights $\omega_{k'k}^l$ s.t. $\sum_{k=1}^{N_l} \omega_{k'k}^l = 1$ to encode off-set interactions. They allow for 'geons' (geometric units describing shapes) in neural networks and psycho-physics [10]. Here χ denotes an activation function ('ReLu'). For intuitive visualizations of PDE-G-CNNs see [74, Figs.6,7] and [37, Fig.2].

The key idea is to insert W(g, t) that is the solution of equivariant PDE system (9.3), where in both PDE systems we set the variable U as initial condition, while setting multiple off-set variables $\mathbf{c} = \mathbf{c}^k$ for $k = 1, ..., N_l$. Only at the first layer of the forward iteration network we set the orientation score as initial condition.

In practice, we take advantage of nowadays very efficient GPUprocessing, and apply an operator splitting of the PDE-evolution. Preliminary investigations already yield much better results in applications [51, 73], revealing a great potential.

We train as follows: Let $\{f^i\}_{i=1}^N \subset \mathbb{L}_2(\mathbb{R}^d)$ be a training dataset consisting of images, with $\{V^i\}_{i=1}^N$ the ground-truth lifted segmented blood vessel segmentation measures. We compute the network inputs U_0^i from the orientation score of image f^i . We compute the outputs and minimize a quadratic/cross-entropy loss between training data and network outputs on the GPU.

This yields optimal geometric network parameters:

$$\boldsymbol{\Omega}^* = \{ (\mathbf{c}_l^k)_{i=1}^{N_l}, \ \boldsymbol{\omega}^l, \ [\mathcal{G}_2], \ [\mathcal{G}_3] \mid l = 0, \dots, \ L-1 \}$$
(9.9)

with advection vectors \mathbf{c}_{l}^{k} , with weight matrices $\omega^{l} = [\omega_{kk'}^{l}]$ hat store the weigths (9.8) and that account for the 'geons' mentioned above. Finally, $[\mathcal{G}_{m}]$ stores the left- invariant matrix coefficients of metric tensor \mathcal{G}_{m} , cf. (9.4).

The above provides us with optimized mobility measures \mathcal{M} for tracking (Fig. 9.6). Altogether, we gain geometric network interpretability of CNNs with impact:

 practical benefits for new inclusion of morphological PDEs and convolutions. Erosions in our PDE evolution (9.3) can keep the geometric information flows through the network sharp and effective, requiring less (ad-hoc) ReLu activiations. Dilations in (9.3) allow bridging of locally interrupted contours where required (by training). Together, dilations and erosions, allow us to eliminate the ad-hoc ReLU's (complicating interpretation) altogether, see [73, Prop.5.15].

 new impact on cortical models for line and contour perception.
 The first stages of human cortical vision are generally modeled by an operator design similar to ours in Fig. 9.2. Orientation selective cells in the primary visual cortex (discovered by Nobel prize laureates Hubel and Wiesel [45]) together encode all orientations per position (akin to an orientation score, (9.1) & Fig.9.2) and their interaction is modeled [11, 20, 29, 31, 60, 61] by sub-Riemannian geometric control, and contour perception via equivariant PDEs on \mathbb{M}_2 . We now include deeper layers to model 'geons' [10, 50] and long range pattern interaction [52].

- Is the geometry of human vision indeed the one for artifical vision?. Do sub-Riemmanian diffusion priors outperform Riemannian diffusion priors? In other words, is the geometry underlying the first stages of human cortical vision [11, 60, 61] also the optimal one in applications (III) of PDE-based CNNs?
- new detection of landmarks/abnormalities/diseases. The PDE-based CNNs on M_d can also be used for training spatial localization (detection) of abnormalities and landmarks in vasculature (e.g. main feeding vessels, micro-bleeds, optic disks [3], cancer [51], in the applications in Section 2b). Then the training is a multiclass classification, and we finish the network applying a Boltzman (or 'soft-max') distribution while minimizing a cross-entropy loss [6, 51].

C2. Training of geometric parameters in fundamental tools such as Finsler functions (9.7) in tracking (II-c). Here re-enforcement learning for dynamic evolution of \mathcal{F}^U allow for more stable tracking results. Moreover, in (II-c) we train the mobility \mathcal{M} of the Finsler function (9.7) via the PDE-based CNNs in (I).

The PDE Operator splitting for PDE-based CNNs

The operator splitting of the equivariant PDE evolution (9.3) boils down to iteratively activating one of the terms in their generator. Firstly, the convection PDE part in Figs. 9.2, 9.6 is solved by a transport along a characteristic curve, cf. [73, Prop.5.1]. Secondly, the solution of the fractional diffusion PDE case on homogeneous space $\mathbb{M}_d = G/H$, recall (9.3), is given by a linear group convolution with a kernel:

$$W(\mathbf{p}, t) = (K_t^{\alpha} * U)(\mathbf{p}) = \int_G K_t^{\alpha} (g^{-1} \cdot \mathbf{p}) \ U(g \cdot \mathbf{p}_0) \ dg$$
(9.10)

where \mathbf{p}_0 denotes the origin in the homogeneous space using the action (9.2) of theLie group on the homogeneous space, and in the integration above we use the usualHaar measure on G = SE(d). For my exact solutions of the kernels see [28, 64]. For adequate, tangible kernel approximations see [73], which we now further improve.Thirdly, the dilation/erosion parts are solved by morphological convolution

$$W(\mathbf{p}, t) = (k_t^{\alpha} \Box U)(\mathbf{p}) = \pm \inf_{g \in G} \{k_t^{\alpha} (g^{-1} \cdot \mathbf{p}) \pm U(g \cdot \mathbf{p}_0)\}.$$
(9.11)

Here the kernels k_t^{α} are positive and the –cases solve dilation PDE, and the + cases solves the erosion PDE, recall (9.3). Via the approximative Cramér transform [37] I derived tangible analytic solutions that are adequate [72, 73]. Fourthly, we caninclude mean-curvature flows [74] by median filtering (II-b).

To see how this applies to automatic vessel segmentation (III), see Fig. 9.3.

Knowledge utilisation

We aim for new interpretable, equivariant, geometric deep learning methods via convolutional networks that are based on PDEs on Lie groups and probability theory. We will solve theoretical challenges in probability theory (I) and establish stable differential geometric tracking methods (II,III). The methods also apply to many other fields (robotics [18], radar [57], machine learning [22], visual perception[20, 61]), but we focus on medical image analysis.

Successful medical imaging applications require computer aided diagnosis (CAD) tools, which are robust, efficient and relevant.

CAD tools for vessel tracking lack these criteria, hampering their effective clinical usage. Current vessel tracking toolsoften fail at complex structures (crossings, bifurcations), cf. Fig. 9.1. This typically results in a high level of costly user interactions. We will tackle this *generically* viaour geometric learning approach on \mathbb{M}_d , following the workflow in Fig. 9.4.

We test our automatic vessel/fiber tracking, vessel segmentation algorithms, and disease detection algorithms, in collaboration with health-tech partners on:

X-ray images (2D & 2D+time, 3D) of complex vasculature, pertaining to abdominal aortic arteries, coronary arteries, artery vein malformations (AVM), Fig. 9.8(A), or tumours. Automatic tracking, segmentation and analysis of vascular trees (e.g. coronary arteries) is crucial for treatment planning. To utilize our algorithms and knowledge in clinical practice we collaborate with industrialpartner Philips, expanding our collaboration [46].

Optical images (2D & 3D) of the eye containing vascular trees (III), provide *the* non-invasive way to diagnose diseases (glaucoma, Altzheimer's disease and diabetes [3]). The goal is to automatically track and segment full vascular trees, and to analyse them to find biomarkers (e.g. by tortuosity [78]) for earlydiagnosis, cf. Fig. 9.8(B). We test on public benchmark data, aiming to improve [3, 6], like in [73]. We deal with cusp-problems [3] and bifurcations [36].

Diffusion-Weighted MR images (5D) of the brain. We quantify structural connectivity in brain white matter via crossing-preserving, wavefront propagation, prior to geodesic tracking (III).

Histopathology Images (2D) of breast and skin. We aim to improve automaticprognostication of breast cancer patients [51] by using our (PDE-)G-CNNs (II).

Fig. 9.8: A: Top images: X-ray of vasculature near the abdominal aorta requires vessel enhancement and tracking for guidewire navigation d uring stenting. Bottom image: 3D X-ray of a brain with AVM. B: Top images: Optical retinal images where vascular tree analysis allows early diagnosis of diseases. Bottom images: 3D-OCT (Optical Coherence Tomography, [47]) requires 3D vessel enhancement and detection.



Fig. 9.9: C: tractography result for the cortical spinal tract for motorics in brain white matter in DW-MRI relevant for surgery planning requires manual inputs to overcome all crossing fiber structures. D: automatic prognostication of breast cancer [51] requires more descriptive morphological grouping of neighboring cell-boundaries.



Regarding geometric deep-learning via PDE-G-CNNs the work [73] shows that PDE-G-CNNs [37, 73] improve G-CNNs [6, 22] that in turn improve standard CNNs in many applications [6, 51]. In fact PDE-G-CNNs yield better classification performances together with huge network reductions and thanks to careful PyTorch coding by B. M. N. Smets without a loss of inference and training time. We applied tangible analytic approximations [73] for PDEs on \mathbb{M}_d that are sufficiently close to new exact PDEsolutions that I derived [28, 33].

Regarding geometric image processing via data-driven Cartan connections we tackled (2D, 3D and 5D) image processing tasks in [38], and recently we achieved new practical and theoretical results by including optimized locally-adaptive frames. Regarding geodesic vessel tracking along sub-Riemannian geodesics, see [3, 36], with comparisons to our new exact sub-Riemannian geodesics [35].

Regarding knowledge utilization (KU) with the Medical Image Analysis group (Dep. BME) at TU/e see [51, 81]. Regarding KU with health tech partner Philips see for example [6] (Philips Impact Award). Regarding KU in neuro-imaging see [74]. For KU on vessel tracking in retinal imaging (with UM) see [5, 37, 73, 75].

Acknowledgement We gratefully acknowledge the Dutch Foundation of Science NWO for financial support via the Talent Programme VICI 2020 Exact Sciences (VI.C 202–031).

References

- M. Akian, J. Quadrat, and M. Viot. Bellman processes. Lecture Notes inControl and Information Science, 199:302–311, 1994.
- [2] T. Alt, J. Weickert, and P. Peter. Translating diffusion wavelets and regular-isation into residual networks. CoRR, pages 1–12, 2020.
- [3] E.J. Bekkers. Retinal Image Analysis using Sub-Riemannian Geometry in SE(2). PhD thesis, Eindhoven University of Technology, 2017.
- [4] E.J. Bekkers. B-spline {cnn}s on Lie groups. In International Conference on Learning Representations, 2020.
- [5] E.J. Bekkers, R. Duits, T. Berendschot, and B. ter Haar Romeny. A multi- orientation analysis approach to retinal vessel tracking. JMIV, 49(3):583–610,2014.
- [6] E.J. Bekkers, M.W. Lafarge, M. Veta, K.A.J. Eppenhof, J. Pluim, and R. Duits. Roto-translation covariant convolutional networks for medical image analysis. In Frangi et al., editor, *MICCAI 2018*, pages 440–448, Cham, 2018. Springer.
- [7] Erik J. Bekkers, Da Chen, and Jorg M. Portegies. Nilpotent Approximations of Sub-Riemannian Distances for Fast Perceptual Grouping of Blood Vesselsin 2D and 3D. JMIV, 60(6):882–899, 2018.
- [8] Y. Benoist and J.-F. Quint. Central limit theorem for linear groups. Ann. Probab., 44(2):1308–1340, 03 2016.
- [9] M. Berlamio, L. Calatroni, V. Franceschi, B. Franceschiello, and D. Prandi. Acortical-inspired model for orientation-dependent contrast perception: A link with wilson-cowan equations. LNCS, 11603:472–484, 2019.
- [10] I. Biederman. Geon Theory as an Account of Shape Recognition in Mind, Brain, and Machine. BMVA Press, 1993.
- [11] U. Boscain, R. Chertovskih, J.-P. Gauthier, D. Prandi, and A. Remizov. Cortical-inspired image reconstruction via sub-Riemannian geometry and hypoellipticdiffusion. *ESAIM:ProcS*, pages 37–53, 2018.
- [12] U. Boscain, R. Duits, F. Rossi, and Y.L. Sachkov. Curve cuspless reconstruction via sub-Riemannian geometry. ESAIM: COCV, 20:784–770, 2014.
- [13] J. Bruna and S. Mallat. Invariant scattering convolution networks. IEEE-PAMI, 35(8):1872– 1886, 2013.
- [14] M. Burger, A. Sawatzky, and G. Steidl. First Order Algorithms in VariationalImage Processing, pages 345–407. Springer, Cham, 2016.
- [15] A. Chambolle and T. Pock. Total roto-translational variation. Numer Math, 142(3):611– 666, 2019.
- [16] D. Chen. New Minimal Paths Models for Tubular Structure Extraction and Image Segmentation. PhD thesis, Université Paris Dauphine, 2016.
- [17] D Chen and L.D. Cohen. Fast asymmetric fronts propagation for image seg- mentation. JMIV, 60:766–783, 2018.
- [18] Gregory S. Chirikjian and Alexander B. Kyatkin. Engineering Applications of Noncommutative Harmonic Analysis: With Emphasis on Rotation and MotionGroups. CRC Press, September 2000.
- [19] G. Citti, B Franceschiello, G. Sanguinetti, and A. Sarti. Sub-Riemannian meancurvature flow for image processing. SIIMS, 9(1):212–237, 2016.
- [20] G. Citti and A. Sarti. A cortical based model of perceptional completion in theroto-translation space. JMIV, 24(3):307–326, 2006.
- [21] T. Cohen, M. Geiger, and M. Weiler. A general theory of equivariant cnns onhomogeneous spaces. CoRR, abs/1811.02017, 2018.
- [22] T.S. Cohen and M. Welling. Group equivariant convolutional networks. Proc. of the 33rd Int. Conf. on Machine Learning, 48:1–12, 2016.
- [23] T.S. Cohen and M. Welling. Steerable CNNs. International Conference on Learning Representations, pages 1–14, 2017.
- [24] E.J. Creusen, R. Duits, A. Vilanova, and L.M.J. Florack. Numerical schemes for linear and non-linear enhancement of DW-MRI. NM-TMA, 6(1):138–168,2013.
- [25] S Dieleman, J. De Fauw, and K. Kavukcuoglu. Exploiting cyclic symmetry inconvolutional neural networks. ICML, 2016.
- [26] A. Dubrovina-Karni, G. Rosman, and R. Kimmel. Multi-region active contours with a single level set function. *IEEE PAMI*, 37(8):1585–1601, 2015.
- [27] R. Duits. Perceptual Organization in Image Analysis. PhD thesis, EindhovenUniversity of Technology, 2005. http://yp.wtb.tue.nl/pdfs/5474.pdf.
- [28] R. Duits, E. J. Bekkers, and A. Mashtakov. Fourier transform on the homoge-neous space of 3d positions and orientations for exact solutions to linear PDEs. *Entropy: Special Issue: Joseph Fourier 250th Birthday*, 21(1):1–38, 2019.
- [29] R. Duits, U. Boscain, F. Rossi, and Y. Sachkov. Association Fields via Cuspless Sub-Riemannian Geodesics in SE(2). JMIV, 49(2):384–417, 2014.

- [30] R. Duits, E. Creusen, A. Ghosh, and T. Dela Haije. Morphological and linearscale spaces for fiber enhancement in DW-MRI. JMIV, 46(3):326–368, 2013.
- [31] R. Duits, M. Felsberg, G. Granlund, and B. M. ter Haar Romeny. Image analysis and reconstruction using a wavelet transform constructed from a reducible representation of the Euclidean motion group. *IJCV*, 79(1):79–102, 2007.
- [32] R. Duits, L. M. J. Florack, J. de Graaf, and B. M. ter Haar Romeny. On the axioms of scale space theory. JMIV, 20:267–298, 2004.
- [33] R. Duits and E. M. Franken. Left invariant parabolic evolution equations on SE(2) and contour enhancement via invertible orientation scores. Quarterly of Applied mathematics, AMS, 68:255–331, June 2010.
- [34] R. Duits, H. Fuehr, B.J. Janssen, L.M.J. Florack, and H.A.C. van Assen. Evolu- tion equations on Gabor transforms and their applications. ACHA, 35(3):483–526, 2013.
- [35] R. Duits, A. Ghosh, T.C.J. Dela Ha e, and A. Mashtakov. On sub-Riemanniangeodesics in SE(3) whose spatial projections do not have cusps. Journal of Dynamical Control Systems, 22(4):771–805, 2016.
- [36] R. Duits, S.P.L. Meesters, J-M. Mirebeau, and J. M. Portegies. Optimal Pathsfor Variants of the 2d and 3d Reeds-Shepp Car with Applications in image analysis. JMIV, 60(6):816–848, July 2018.
- [37] R. Duits, B. Smets, E. Bekkers, and J.M. Portegies. Equivariant deep learningvia morphological and linear scale space pdes on the space of positions and orientations. In A. Elmoataz, J. Fadili, Y. Quéau, J. Rabin, and L. Simon, editors, SSVM in Computer Vision, pages 27–39, Cham, 2021. Springer.
- [38] R. Duits, B. M. N. Smets, A. J. Wemmenhove, J. W. Portegies, and E. J. Bekkers. Recent Geometric Flows in Multi-orientation Image Processing viaa Cartan Connection, pages 1– 60. Springer, Cham, 2021.
- [39] M. Elad. Deep, Deep Trouble. SIAM-NEWS, page 12, May 2017.
- [40] J. Fadili, G. Kutyniok, G. Peyré, G. Plonka-Hoch, and G. Steidl. Guest editorial: Mathematics and image analysis. JMIV, 52(3):315–316, Jul 2015.
- [41] W. Feller. An Introduction to Probability Theory and its Applications, vol- ume II. Wiley, New York, 1971.
- [42] E. M. Franken. Enhancement of Crossing Elongated Structures in Images. PhD thesis, TU/e, 2008.
- [43] F. Guichard and J.M. Morel. Partial differential equations and image iterativefiltering. In I.S. Duff and G.A. Watson, editors, *The State of the Art in Numerical Analysis.*, volume 63, pages 252–562. Clarendon Press, 1997.
- [44] J. Hannink, R Duits, and E.J. Bekkers. Crossing-preserving multi-scale vessel-ness. In Golland et al., editor, *MICCAI*, volume 8674, pages 603–610, 2014.
- [45] D. H. Hubel and T. N. Wiesel. Receptive fields of single neurons in the cat's striate cortex. Journal of Physiology, 148:574–591, 1959.
- [46] M. H. J. Janssen, A. J. E. M Janssen, J. Olivan Bescos, and R. Duits. Design and processing of invertible orientation scores of 3d images. JMIV, 60(9):1427–1458, 2018.
- [47] J. Khadamy, K. Abri Aghdam, and K. G. Falavarjani. An update on optical coherence tomography angiography in diabetic retinopathy. J Ophthalmic VisRes, 13:487–497, 2018.
- [48] R. Kondor. Group Theoretical Methods in Machine Learning. PhD thesis, Columbia University, 2008.
- [49] R. Kondor, Z. Lin, and T. Shubshendu. Clebsch-gordan nets: a fully fourier space spherical convolutional neural network. *NeurIPS*, pages 1–12, 2018.
- [50] J. Kubilius, S. Bracci, and H.P. Op de Beeck. Deep neural networks as a com-putational model for human shape sensitivity. *PLOS Computational Biology*, 12(4):1–26, 04 2016.
- [51] M. W. Lafarge, E. J. Bekkers, J. Pluim, R. Duits, and M. Veta. Roto-translation equivariant convolutional networks: Application to histopathology image anal-ysis. *MEDIA*, 68:101849, 2021. See also a demo of clinical integration of ourmethod at www.twitter.com/mitkoveta/status/1234577353961082881.
- [52] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. Nature Research, 521(7553):436–444, 2015.
- [53] Q. Li, B. Feng, L. Xie, P. Liang, H. Zhang, and T. Wang. A cross-modality learn-ing approach for vessel segmentation in retinal images. *IEEE-TMI*, 35(1):109–118, 2016.
- [54] P. Liskowski and K. Krawiec. Segmenting retinal blood vessels with deep neural networks. *IEEE-TMI*, 35(11):2369–2380, 2016.
- [55] G. Litjens, B.E. Bejnodri, A.A.A. Setio, F. Ciompi, M. Ghafoorian, J.A.W.M.van der Laak, B. van Ginneken, and C.I. Sánchez. A survey on deep learning inmedical image analysis. *Medical Image Analysis*, 42:60–88, December 2017.
- [56] S.P.L Meesters, P. Ossenblok, L. Wagner, O. Sch ns, P. Boon, L.M.J. Florack, Vilanova, and R. Duits. Stability metrics for optic radiation tractography: Towards damage prediction after resective surgery. *Journal of Neuroscience Methods*, 2017. doi: 10.1016/j.jneumeth.2017.05.029.

- [57] J.-M. Mirebeau and J. Dreo. Automatic differentiation of non-holonomic fastmarching for computing most threatening trajectories under sensors surveil- lance. In Frank Nielsen and Frédéric Barbaresco, editors, *Geometric Science of Information*, pages 791–800. Springer, 2017.
- [58] J.M. Mirebeau. Fast-Marching Methods for Curvature Penalized Shortest Paths. JMIV, 60(6):784–815, 2018.
- [59] M. Pechaud, M. Descoteaux, and R. Keriven. Brain Connectivity UsingGeodesics in HARDI, pages 482–489. Springer, 2009.
- [60] J. Petitot. The neurogeometry of pinwheels as a sub-Riemannian contactstructure. Journal of Physiology - Paris, 97:265–309, 2003.
- [61] J. Petitot. Elements of Neurogeometry. LN in Morphogenesis. Springer, 2017.
- [62] G. Peyré, M. Péchaud, R. Keriven, and L. D. Cohen. Geodesic methods in computer vision and graphics. *Found. Trends. Comput. Graph. Vis.*, 5(3):197–397, 2010.
- [63] J. Portegies, S. Meesters, P. Ossenblok, A. Fuster, L. Florack, and R. Duits. Brain Connectivity Measures via Direct Sub-Finslerian Front Propagationon the 5D Sphere Bundle of Positions and Directions. In E. Bonet-Carne, F. Grussu, L. Ning, F. Sepehrband, and C. M. W. Tax, editors, *ComputationalDiffusion MRI*, pages 309–321, Cham, 2019. Springer.
- [64] J. M. Portegies and R. Duits. New exact and numerical solutions of the (convection-) diffusion kernels on SE(3). DGA, 53:182–219, 2017.
- [65] J. M. Portegies, R. H. J. Fick, G. R. Sanguinetti, S. P. L. Meesters, G. Girard, and R. Duits. Improving Fiber Alignment in HARDI by Combining Contextual PDE Flow with Constrained Spherical Deconvolution. *PLoS ONE*, 10(10), 2015. doi: 10.1371/journal.pone.0138122.
- [66] J.M. Portegies. PDEs on the Lie Group SE(3) and their Applications in Diffusion-Weighted MRI. PhD thesis, Department of Mathematics and Com- puter Science, TU/e, February 2018.
- [67] L. Ruthotto and E. Haber. Deep neural network motivated by partial differential equations. JMIV, 62:352–364, 2020.
- [68] S. Sabour, N. Frosst, and G.E. Hinton. Dynamic routing between capsules. Advances in Neural Information Processing Systems, pages 3857–3867, 2017.
- [69] G. Sapiro. Geometric Partial Differntial Equations and Image Analysis. CUP,2001.
- [70] M. Schmidt and J. Weickert. Morphological counterparts of linear shift- invariant scalespaces. JMIV, 56(2):352–366, 2016.
- [71] L. Siffre. Rigid-Motion Scattering for Image Classification. PhD thesis, EcolePolyechnique, Paris, 2014. Supervisor: Stephane Mallat.
- [72] B.M.N. Smets. Geometric image denoising and machine learning. Mas- ter's thesis, IAM, CASA-TU/e, 2019. url: www.win.tue.nl/~rduits/ reportBartSmets.pdf.
- [73] B.M.N. Smets, J.W. Portegies, E.J. Bekkers, and R. Duits. PDE-based group equivariant convolutional neural networks. Technical report, ArXiV, 2020. www.arxiv.org/abs/2001.09046 (version 4).
- [74] B.M.N. Smets, J.W. Portegies, E. St-Onge, and R. Duits. Total variation and mean curvature PDEs on the Homogeneous Space of Positions and Orienta- tions. JMIV, 2020.
- [75] B. M. et. al. ter Haar Romeny. Brain-inspired algorithms for retinal image analysis. Machine Vision and Applications, 27(8):1117–1135, Nov 2016.
- [76] S. Ullman. High-Level Vision: Object Recognition and Visual Cognition. MITPress, 1996.
- [77] J. Weickert, S. Grewenig, C. Schroers, and A. Bruhn. Cyclic schemes for PDE-based image analysis. CV, 118(3):275–299, Jul 2016.
- [78] Danielle L Weiler, Carla B Engelke, Anna LO Moore, and Wendy W Harri-son. Arteriole tortuosity associated with diabetic retinopathy and cholesterol. Optometry & Vision Science, 92(3):384–391, 2015.
- [79] M. Weiler, M. Geiger, M. Welling, W. Boomsma, and T. Cohen. 3d steerableCNNs: Learning rotationally equivariant features in volumetric data. *NeurIPS*, pages 1–12, 2018.
- [80] S.G. Zadeh, S. Didas, M.W.M. Wintergerst, and T. Schultz. Multi-scale anisotropic fourthorder diffusion improves ridge and valley localization. JMIV, 59(2):257–269, 2018.
- [81] J. Zhang, B. Dashtbozorg, E.J. Bekkers, J. Pluim, R. Duits, and B.M. terHaar Romeny. Robust retinal vessel segmentation via locally adaptive deriva-tive frames in orientation scores. *IEEE-TMI*, 35(12):2631–2644, 2016.
- [82] J. Zhang, R. Duits, B.M. ter Haar Romeny, and G.R. Sanguinetti. Numerical approaches for left-invariant diffusions on SE(2), their comparisons to exact solutions, and applications in retinal imaging. NM-TMA, 9(1):1–50, 2016.



ABOUT THIS BOOKLET

This booklet was made possible by a financial contribution from the mathematics cluster NDNS+ and the Dutch Research Council (NWO). Dutch mathematics is organised into four mathematics clusters:

- NDNS+: Nonlinear Dynamics of Natural Systems (analysis, scientific computing, dynamical systems) - www.ndns.nl
- DIAMANT: Discrete, Interactive and Algorithmic Mathematics, Algebra and Number Theory www.diamant.science.uu.nl
- GQT: Geometry and Quantum Theory www.gqt.nl
- STAR: Stochastics Theory and Applied Research www.eurandom.tue.nl/STAR

All mathematics clusters have relations to topics and themes in artificial intelligence, as indicated in the figure below.



For the introductory chapter of this booklet, use has been made of several reports, listed below, that have been published by the Department of Energy (Advanced Scientific Computing Research) in the USA. We would like to express our thanks to Nathan Baker (Pacific Northwest National Laboratory, chair for report 1) and Steven Lee (DOE ASCR Applied Mathematics Lead).

 Basic Research Needs for Scientific Machine Learning: Core Technologies for AI (January 2019)

Foundational research areas and 3 major use cases are identified across the DOE.

Brochure - 4 pages: www.osti.gov/biblio/1484362 Full report - 109 pages: www.osti.gov/biblio/1478744

National AI Research & Development Strategic Plan: 2019 Update

Full report - 50 pages: www.nitrd.gov/pubs/National-Al-RD-Strategy-2019.pdf

AI for Science report (February 2020)

Over 1,000 scientists participated in a series of four "Al for Science" town halls organized by DOE National Laboratories from July - October 2019. The goal of the town hall series was to examine scientific opportunities in the areas of AI, Big Data, and high-performance computing, and to capture the big ideas, grand challenges, and next steps to realizing these opportunities.

Full report - 224 pages: www.osti.gov/biblio/1604756

 Basic Energy Sciences Roundtable on Producing and Managing Large Scientific Data with Artificial Intelligence and Machine Learning (2020)

Report from a Basic Energy Sciences roundtable in October 2019 to identify specific Priority Research Opportunities (PROs) and coordinated, long-term AI/ML research efforts that will drive major advances in neutron, photon, and nanoscale sciences.

Full report - 61 pages: www.osti.gov/biblio/1630823

 Opportunities and Challenges from AI and ML for the Advancement of Science, Technology, and the Office of Science Missions (September 2020)

An ASCR Advisory Subcommittee report that identifies strategies that ASCR can use, in coordination with the other Office of Science programs, to address the challenges and deliver on the opportunities. Full report - 63 pages: www.osti.gov/biblio/1734848

Autonomous Discovery in Science and Engineering (August 2021)

A "Welcome to the Autonomous World of Science" at DOE national laboratories, universities, and scientific research centers. Website has material for: Keynote talks, Breakout sessions, Software tutorials, Lightning talks Workshop website: https://autonomousdiscovery.lbl.gov/ Full report - 155 pages: https://doi.org/10.2172/1818491

The booklet was presented in the context of the workshop *Computational Science and Machine Learning* held in the Lorentz Center in Leiden, The Netherlands (www.lorentzcenter.nl/), a workshop center that hosts international scientific meetings. We thank the organisers of the workshop, and the staff of the Lorentz Center, for their assistance.

AUTHORS

Joris Bierkens Delft Institute of Applied Mathematics, joris.bierkens@tudelft.nl

Christoph Brune Professor and Chair of Applied Analysis, Department of Applied Mathematics, University of Twente, c.brune@utwente.nl

Graziana Colonna Departamento de Matemáticas, University of A Coruña, e-mail: graziana.colonna@udc.es

Daan Crommelin Centrum Wiskunde & Informatica, Amsterdam, and Korteweg-de Vries Institute for Mathematics, University of Amsterdam, Daan.Crommelin@cwi.nl

Remco Duits Department of Mathematics and Computer Science, TU Eindhoven

Wouter Edeling Centrum Wiskunde & Informatica, Amsterdam, Wouter.Edeling@cwi.nl

Giulia Ferrandi Department of Mathematics and Computer Science, TU Eindhoven, g.ferrandi@tue.nl

Perfect Y. Gidisu Department of Mathematics and Computer Science, TU Eindhoven, p.gidisu@tue.nl

Lech A. Grzelak Rabobank, Utrecht, the Netherlands, Lech.Grzelak@rabobank.com

Peter Grünwald CWI, Amsterdam, the Netherlands, and Leiden University, Mathematical Institute, peter.grunwald@cwi.nl

Mengwu Guo

Assistant Professor in Mathematical Theory of Uncertainty Quantification, Department of Applied Mathematics, University of Twente, m.guo@utwente.nl

Sibylle C. Hess Department of Mathematics and Computer Science, TU Eindhoven, s.c.hess@tue.nl

Michiel E. Hochstenbach Department of Mathematics and Computer Science, TU Eindhoven, m.e.hochstenbach@tue.nl

Shuaiqiang Liu Centrum Wiskunde & Informatica (CWI), Amsterdam, shuaiqiang.liu@cwi.nl

Matthias Möller Delft Institute of Applied Mathematics, Delft University of Technology, m.moller@tudelft.nl

Cornelis W. Oosterlee Mathematical Institute, Utrecht University, c.w.oosterlee@uu.nl

Frank van Ruiten Delft Institute of Applied Mathematics, Delft University of Technology

Wil Schilders, Eindhoven University of Technology, Past president of ECMI (2010-2011) and EU-MATHS-IN (2016-2020), President Elect ICIAM

Julian Suk Department of Applied Mathematics, University of Twente, j.m.suk@utwente.nl

Deepesh Toshniwal Delft Institute of Applied Mathematics, Delft University of Technology, d.toshniwal@tudelft.nl

Jelmer M. Wolterink Department of Applied Mathematics, University of Twente, j.m.wolterink@utwente.nl

COLOPHON

Editorial

Wil Schilders (TU Eindhoven & Platform Wiskunde Nederland) Special thanks to Harshit Bansal (TU/e), Peter Grunwald (CWI, chair of AIM initiative) and Christoph Brune (UT)

Design

WAT Ontwerpers, Utrecht www.watontwerpers.nl

Photo iStock Photo

Printing Books4You, 's-Hertogenbosch www.prijsprinter.nl

Contact

For further information contact Wil Schilders, w.h.a.schilders@tue.nl or contact the lead of the AIM initiative Peter Grunwald, peter.grunwald@cwi.nl











Scientific machine learning is a core component of artificial intelligence (AI) that has the potential to transform science and engineering research. Breakthroughs and major progress will be enabled by harnessing investments in massive data from scientific user facilities, software for predictive models and algorithms, high-performance computing platforms, and the workforce of researchers. The crosscutting nature of machine learning and artificial intelligence provides a strong incentive for formulating a prioritized research agenda to maximize the capabilities and scientific benefits. Mathematics is needed for the much-needed systematic understanding of AI, for example, greatly improving reliability and robustness of AI algorithms, understanding the operation and sensitivity of networks, reducing the need for abundant data sets, or incorporating physical properties into neural networks needed for superfast and accurate simulations in the context of digital twinning.