Automatic design of PDE solvers with neural-guided search

Main Supervisor: Ben Moseley
Co-Supervisor: Dinesh Parthasarathy

Keywords: automatic algorithm design, PDE solvers, multigrid methods, genetic programming, large language models, reinforcement learning, scientific machine learning

Background

Partial differential equations (PDEs) underpin simulations across science and engineering, from weather forecasting and climate modelling to fluid dynamics and digital twins. When these simulations are run on computers, they **generate large linear systems** which must be solved repeatedly, and **solving them often dominates the overall runtime** of the simulation. The performance of a linear solver depends strongly on the underlying problem, its discretisation, and the target hardware. Designing efficient, application-specific solvers therefore requires deep expertise in numerical linear algebra and high-performance computing, which most application scientists do not have.

Recent work [1,2] has shown that this **design process can be automated**. Using grammar-guided genetic programming, a context-free grammar defines the space of all valid multigrid solver algorithms, and an **evolutionary algorithm** searches this space for the most efficient solver for a given problem (Figure 1). This approach has discovered novel solvers with arbitrary cycling strategies that outperform widely used, expert-designed methods such as standard V- and W-cycles.

However, such algorithm search strategies remain computationally expensive. They often start from a random population of largely sub-optimal solvers, rely on generic, handcrafted mutation operators that explore the design space inefficiently, and must evaluate every candidate solver on the target hardware over many generations. Moreover, valuable solver information generated during each search is discarded rather than reused.

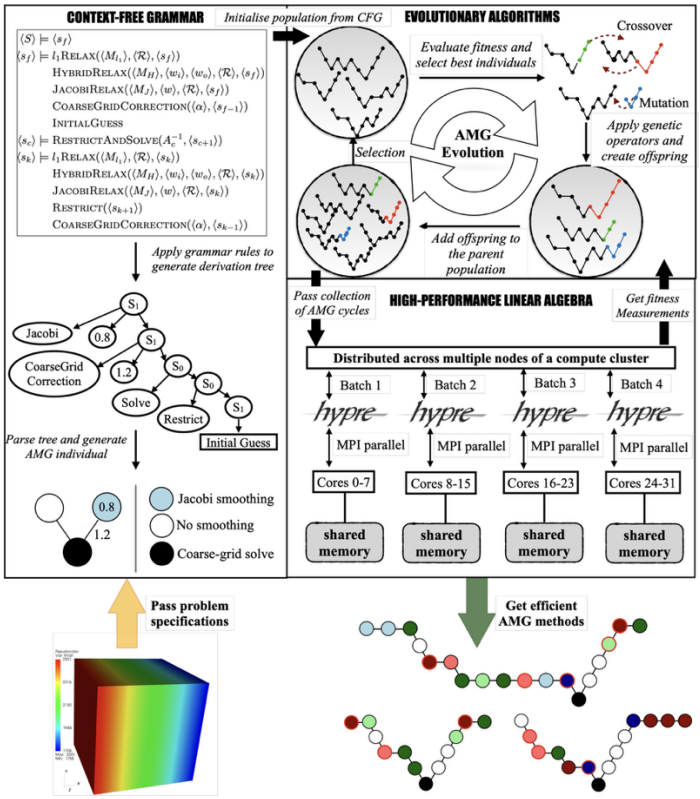


Figure 1 (reproduced from [1]): Automatic design of algebraic multigrid (AMG) solvers using grammar-guided evolution. Given an input problem, the approach combines a context-free grammar, an evolutionary algorithm and high-performance solvers to generate flexible AMG cycles. Nodes in the resulting cycles are colour-coded by smoothing operation. In this project, neural networks will learn from previous searches to guide this process.

Project

This project asks whether neural networks can learn how to search the algorithmic design space of PDE solvers more effectively. Rather than generating complete solver programs from scratch, we will train models on the rich data generated by previous grammar-guided searches, so that they **learn which changes to a solver are likely to improve its performance** for a given PDE problem. These learned search strategies will be integrated into the evolutionary framework, combining the exploration capabilities of genetic programming with data-driven guidance to accelerate the search and discover more efficient solver algorithms.

A key aspect of the project will be investigating how best to design and train these models, including how solvers and PDE problems should be represented and which machine learning architectures are most effective. We will also investigate how the search can **continually improve itself** as new data is generated, inspired by self-improving AI systems such as AlphaGo [3] and LLM-guided evolutionary systems such as FunSearch and AlphaEvolve [4,5], and whether learned search strategies transfer across problems and hardware.

The methods will be benchmarked on problems from the SuiteSparse Matrix Collection [6] and compared against existing grammar-guided evolutionary methods [1,2]. A major goal is to apply the approach to real-world simulations, such as large-scale earthquake simulation, realistic fluid simulations, and climate modelling, working with domain-specialist teams across Imperial.

Key research questions are: Can neural networks learn effective search strategies over algorithmic design spaces? How much can learned guidance accelerate the discovery of efficient solvers compared with standard evolutionary search? Can the search improve itself over time? Do learned search strategies generalise to unseen PDE problems and hardware? Can AI-guided search discover fundamentally new solver algorithms?

Impact

Automatically designed, high-performance PDE solvers will have a transformative impact on scientific computing. They would allow non-specialists to deploy efficient solvers tailored to their problem and hardware, reducing the runtime, cost and energy use of large-scale simulations in fields such as climate modelling, engineering design, and geophysics. More broadly, the project will establish new AI-driven approaches for automatically discovering scientific algorithms.

Supervisory team

Dr. **Ben Moseley** is an Assistant Professor in AI (Schmidt AI in Science Fellow) at the Department of Earth Science and Engineering and a Fellow at the Imperial I-X Centre. He heads the Scalable Scientific Machine Learning Lab and is an expert in scientific machine learning, physics-informed neural networks, neural differential equations, hybrid modelling, learned inverse algorithms, high-performance computing, geophysics, and planetary data science.

Dr. **Dinesh Parthasarathy** is a researcher at Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU). His research applies AI techniques to design efficient numerical methods for scientific simulations, with a focus on grammar-guided evolutionary algorithms for automatically designing multigrid solvers.

We are open to identifying further co-supervisors or project advisors as necessary. We actively encourage collaboration with industry and other research groups.

Research group

The student will be part of the **Scalable Scientific Machine Learning Lab** headed by Dr. Ben Moseley. The lab accelerates scientific research by designing scientific machine learning algorithms and applying them to impactful problems across science. See our lab website for more information.

We are a highly **cross-disciplinarity** team – we train our members across machine learning, applied mathematics, high-performance computing, and in domain-specific areas including geophysics, climate science, and planetary science. We **collaborate** with other groups at the Department of Earth Science and Engineering, I-X (Imperial's AI initiative), other Imperial departments, and with external universities and industry partners. Lab members are encouraged to present and publish at high-impact conferences and journals.

Student profile

We are looking for someone who is motivated to complete a PhD at the intersection of scientific machine learning, automatic algorithm design, and high-performance computing. This is an interdisciplinary project, and we do not expect candidates to arrive with expertise in all areas; instead, we are looking for someone with a strong technical foundation, enthusiasm for interdisciplinary research, and an ability to approach complex problems with creativity and curiosity.

Essential qualifications / experience:

- Good Master's degree in a relevant field (e.g. applied mathematics, physics, computer science, engineering, or related areas). Motivated candidates with an excellent bachelor's degree and a relevant research portfolio are encouraged to apply
- Completed courses in machine learning and/or applied mathematics
- Coding proficiency in e.g. Python/ C++/ Julia/ Fortran

Desirable qualifications / experience:

- Experience in numerical linear algebra and/or numerical modelling (e.g. multigrid, Krylov methods, finite element methods)
- Experience with evolutionary algorithms, genetic programming, or optimisation
- Familiarity with deep learning architectures, large language models, and/or reinforcement learning
- Proficiency with Python machine learning frameworks (PyTorch, JAX (with Equinox))
- Experience in scientific, HPC, GPU, and/or parallel computing
- Relevant publications and/or industry experience are a plus

Funding

This project is not currently funded through a research grant and is eligible for College and/or Departmental scholarship funding. For more details on scholarship funding and deadlines see here.

Apply

If you are interested, please start by sending us:

- CV (including education, and any research experience).
- Brief motivation letter (200 – 400 words) where you should highlight how your experience enables you to pursue the project, and your own vision for the project (can be in the email body).
- Any additional materials that support your application (optional).

For more details on our lab's PhD application process see here.
For more details on the Imperial PhD application process see here.

Contact

Dr. Ben Moseley: b.moseley@imperial.ac.uk

References

[1] Parthasarathy et al., (2026). Automated Grammar-based Algebraic Multigrid Design With Evolutionary Algorithms. ArXiv. arxiv.org/abs/2603.17641
[2] Parthasarathy et al., (2025). Towards Automated Algebraic Multigrid Preconditioner Design Using Genetic Programming for Large-Scale Laser Beam Welding Simulations. PASC '25: Proceedings of the Platform for Advanced Scientific Computing Conference. doi.org/10.1145/3732775.3733589
[3] Silver et al., (2016). Mastering the game of Go with deep neural networks and tree search. Nature.
[4] Romera-Paredes et al., (2024). Mathematical discoveries from program search with large language models. Nature.
[5] Novikov et al., (2025). AlphaEvolve: A coding agent for scientific and algorithmic discovery. ArXiv.
[6] SuiteSparse Matrix Collection, https://sparse.tamu.edu/about