Development of a parallel Multigrid Method in Composyx

November 5, 2025

- **Contract type:** Internship
- **Level of qualifications required:** Master 2
- **Fonction:** Intern

1 Context

The solution of large, sparse linear systems is a fundamental topic in numerical linear algebra and continues to drive active research in academia and industry. Developing efficient solvers for such systems requires particular attention, especially to preconditioning strategies. Achieving scalable performance on modern architectures is a key objective, and multigrid methods have emerged as one of the most effective approaches toward this goal.

Multigrid methods rely on a hierarchy of increasingly finer grids, each typically obtained by uniformly refining the previous one. The coarsest grid is chosen to make the residual problem inexpensive to solve, while the finest grid provides the desired level of accuracy for the overall solution.

composyx is a C++ library for numerical linear algebra designed with composability in mind. It enables users to express a wide range of numerical algorithms through a high-level interface and to deploy them seamlessly from laptops to supercomputers.

2 Assignment

The objective of this internship is to develop a parallel version of a multigrid solver within the composyx library.

To achieve this, we focus on a simple elliptic problem defined on a one-dimensional geometry. The goal is to leverage the existing data structures of composyx, such as `PartMatrix`, `PartDenseMatrix`, or suitable extensions of these, to partition the grids or layers within the multigrid hierarchy.

Composyx will provide high-level building blocks (e.g., matrix–vector multiplication) together with bindings to highly optimized kernels targeting specific hardware architectures (CPU, GPU, . . .). Using these components will make it possible to develop multigrid solvers that are portable across diverse computing platforms. Performance and scalability of the resulting code will be assessed on the plafrim platform, leveraging distributed multicore and GPU nodes.

If times permits, the next step will be to investigate higher-level abstractions of the numerical algorithm, with the goal of proposing a unified framework that encompasses both geometric (GMG) and algebraic (AMG) multigrid methods.

3 Skills

- Knowledge of numerical linear algebra and high-performance computing (MPI, OpenMP, runtime systems, etc.);
- Knowledge and experience in software development;
- Proficiency in modern C++ and generic programming;
- Software architecture and programming paradigms, software engineering, best practices, and software development tools: version control (git, GitLab), CI/CD (GitLab CI), package building (CMake), . . . ;
- Ability to write and present in both French and English;
- Ability to propose and implement reference implementations, prototypes, and demonstrators: autonomy, creativity, proactive monitoring, and responsiveness to needs;
- Ability to understand scientific contexts and requirements, and translate them into technological implementations.

4 Contacts

- Inria team: concace

- Supervisors:
  - Emmanuel Agullo / emmanuel.agullo@inria.fr
  - Gilles Marait / gilles.marait@inria.fr
  - Clement Guillet / clement.guillet@inria.fr