

## Speeding up the implementation of grid-based PDE approximations in Julia

*Santiago Badia and Francesc Verdugo*

We present Gridap, a novel computational framework for the grid-based approximation of PDEs in the Julia programming language. The main motivation behind this library is to provide an easy-to-use framework for the development of complex PDE solvers in a flexible style, close to interpreted languages like Python, without sacrificing the performance of compiled languages.

State-of-the art scientific software libraries, e.g., deal.ii or FEniCS, are implemented in C/C++ or FORTRAN and provide high-level Python interfaces. Application experts, whose needs are already covered by the PDE library can readily use these high-level interfaces to easily enjoy advanced simulation tools. However, such approach is not satisfactory when users need functionality not provided by the library. Extending the code usually requires to modify the C/C++ or FORTRAN back-end, which has a steep learning curve and poor productivity. This is known as the two programming language problem.

Julia is a recent programming language that has being designed having computationally intensive numerical algorithms in mind, as FORTRAN decades ago. Julia achieves both performance and productivity via a type-inference system and a just-in-time compiler that emits efficient machine code. As a result, there is no need to use low-level C/C++ or FORTRAN backends or vectorization of for-loops for getting performance, thus eliminating the two-language problem.

The Gridap project takes advantage of these features of the Julia programming language to create an extensible and easy-to-use finite element (FE) framework. The design of Gridap is quite unique for a numerical PDE software, since it does not follow neither procedural nor object-oriented paradigms. Instead, we exploit the functional programming capabilities provided by Julia like anonymous functions and closures. Even though the project itself is very young, Gridap already provides different types of FE methods, e.g., Lagrangian FEs, Raviart-Thomas FEs and discontinuous Galerkin schemes. The treatment of multi-field problems is also very natural. Gridap is a collaborative open-source project hosted at github (<https://github.com/gridap/>).