# Improved generation of Fortran interfaces for PETSc

PETSc, the Portable, Extensible Toolkit for Scientific Computation, pronounced PET-see, is for the scalable (parallel) solution of scientific applications modeled by partial differential equations (PDEs). It has bindings for C, Fortran, and Python (via petsc4py). PETSc also contains TAO, the Toolkit for Advanced Optimization, software library. It supports MPI, and GPUs through CUDA, HIP, Kokkos, or OpenCL, as well as hybrid MPI-GPU parallelism; it also supports the NEC-SX Tsubasa Vector Engine.

Currently, only a part of the Fortran interfaces can be generated automatically using bfort. Since the manual generation of the remaining interfaces is tedious and error prone, this project is about an improved generation of Fortran interfaces from PETSc's C code.

The main tasks of this project are

- Definition of a robust and future-proof structure for the Fortran interfaces
- Selection and/or development of a tool that creates the interfaces automatically

More specifically, the first task is about finding a suitable structure of the C-to-Fortran interface that reduces the need of 'stubs' on the C and Fortran side making use of modern Fortran features where appropriate. This task will involve evaluating different approaches found in other projects taking into account the object-oriented approach of PETSc. Prototypes will be implemented manually and evaluated with the help of the PETSc community. The second task is then the automated generation of the Fortran interfaces using the approach selected in the first task. To this end, it will be evaluated whether an extension of bfort, the use of another existing tool, or the development of a completely new tool (probably in Python) is the most suitable approach.

**Links**:

- PETSc
- bfort
- Fortran Wiki: Generating C Interfaces
- Fortran Discourse: ISO_C_binding

**Expected outcomes**: Stable and robust autogeneration of Fortran interfaces for PETSc that works for almost all routines

**Skills preferred**: Programming experience in multiple languages, ideally C and/or Fortran

**Difficulty**: Intermediate, 320 hours

**Mentors**: Martin Diehl (@MarDiehl)