Study of MPI tools to implement two-level parallelism
Simulation of initial boundary value problems

Project description:
The background setting for this project is the following. Consider the numerical simulation of time-dependent partial differential equations (PDEs), where we compute numerically an approximation of a unknown function that depends on space and time. Apart from the space discretization, there are two approaches to discretize the problem in time, namely, using explicit schemes or using implicit schemes. The major difference between explicit/explicit with their proc and cons are the following:

Pros	Cons
Explicit methods	
Each time step is cheap in terms of computational cost	The timestep must be small to ensure stability of the discretization scheme
Implicit methods	
The implicit schemes are unconditionally stable, i.e., the time step can be large, which is very suitable for long time intervals.	During each timestep an algebraic system of equations must be solved, which increases the computational complexity.

In this project we will deal with one particular type of implicit time integration methods, referred to as Implicit Runge-Kutta (IRK) methods of Radau type, which possess both unconditional stability and high order of accuracy. The methods perform as follows. At each time step, we compute a number of stage variables by solving a large scale linear system of equations. Then the approximation of the solution on the next time level is formed as a linear combination of these stages.

Let q be the number of stages, n be the number of space discretization points, usually large. Then, the size of the algebraic systems to be solved is qn. In problems of practical importance n can easily be of order 10^6 or larger. Therefore, iterative methods become a suitable choice, combined with some appropriate technique to accelerate the convergence of the iterations. The acceleration is referred to as a preconditioner and is most often in a form of another matrix. In this way each time step requires matrix-vector multiplications with the large qn-sized matrix and solutions with the preconditioning matrix.

To gain parallelism when using IRK methods, there exist a method to construct the preconditioner as a block-diagonal matrix, where the blocks are of size $n \times n$ and there are q such blocks, where q is the number of stages. Thus, we have one level of parallelism across the stages and can solve the diagonal blocks in a fully parallel manner. Each block is, however, of large size and we want to solve it also in parallel. In this way we have two-level parallelism in the algorithm - across the stages and within the solution of each block. In general, each block can be solved also with another readily parallelized iterative method.

The task of the project is to implement such two-level parallelism in MPI. One can envision two strategies, (i) to ask initially for q number of processes and then dynamically to allocate more resources by, for instance, `MPI_COMM_SPAWN`, or (ii) to allocate all resources in the beginning and split them as needed via communicators.

The scientific library to be used is the *Portable Extensible Toolkit for Scientific Computation (PETSc)* and as hardware - the HPC cluster Rackham at UPPMAX, where PETSc is installed.

PETSc contains an open source set of C tools for the parallel solution of PDEs as well as interface to C/C++, Fortran, Matlab and Python.

A running prototype code in Matlab is available. Initial tests can be done with synthetic data.

As a background, the students should have passed a Finite Element course and Scientific Computing 2 (where numerical solution of ODEs is included). Having passed Scientific computing 3 and some knowledge about iterative solution methods is an advantage.

References:

PETSc `https://en.wikipedia.org/wiki/Portable,_Extensible_Toolkit_for_Scientific_Computation`

PETSc Tutorials `https://www.mcs.anl.gov/petsc/documentation/tutorials/index.html`

Ernst Hairer, Gerhard Wanner, *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*, Springer, 1996. (Available at Ångström's library)