# This is the title of the project report

First author[1], Second author[2], Third author[1,2], and Fourth author[1,2]

[1]Affiliation of first author
[2]Other affiliation

May 2, 2022

**Abstract**

TODO: This is the abstract of the proposal report: in this document we provide a template with the required sections. Additional guidelines are provided for selected sections below. The project report should be in the range of **8-10 pages** including graphs and references, and must contain the following information:

- Abstract
- Background and significance
- Scientific goals and objectives
- Algorithms and code parallelization (including memory requirements)
- Representative performance benchmarks and scaling analysis (strong and weak scaling)
- Resource justification

The report should be typeset using LaTeX[1]. You can use this template as a baseline.

## 1 Background and Significance

TODO: In this section, describe the problem you are going to solve, the research methods, its significance and a comparison with existing work.

## 2 Scientific Goals and Objectives

TODO: What are your scientific goals and objective with this project? Justify the need for compute hours on a HPC architecture.

## 3 Algorithms and Code Parallelization

TODO: Please insert in this section a description of the methods and algorithms of all the codes adopted for your computational study, justifying your choices and describing possible alternatives. Furthermore, please specify whether you are the main developer, a contributing developer or a user of the code. You should include a brief list of the main scientific libraries employed (if any) and a description of the parallelization approach, with specific memory and I/O requirements as well.

Report if the code employs OpenMP shared memory parallelism, MPI distributed parallelism or hybrid MPI/OpenMP, which type of MPI communication has been implemented and if it makes use of shared memory parallelism, GPU accelerators or OpenACC/CUDA (optional), with specific memory and I/O requirements as well.

### Validation, Verification

Please explain how to validate your method against experiments or other established reference data and verify the numerical consistency of your model, citing the relevant references to peer reviewed papers.

## 4 Performance Benchmarks and Scaling Analysis

TODO: Please report in this section the results of the mandatory performance analysis of your code (roofline analysis) as well as strong and weak scaling tests. You should report scaling data (tables) and plots for the code of your project.

You should select meaningful job sizes to simulate the representative systems, compatible with reasonably short runtimes: the lowest number of nodes is determined in general by memory and wall time constraints, while the highest node counts should let you identify the job size at which you reach $\sim 50\%$ of the parallel efficiency with respect to ideal scaling.

You are also requested to provide some key figures of the jobs that you ran during the development of the project:

- the wall clock time of your typical submission in hours
- the expected job size of your typical job in number of nodes
- the memory per node requested by the typical job in GB
- the maximum number of input files read by a job
- the maximum number of output files written by a job
- the library used for I/O (if any)

Examples for I/O libraries are: HDF5, NetCDF or MPIIO.

Table 1 provides a template of how these figures should be presented in the report. The given example describes two test cases used for the development presented in two columns. Note that the given numbers are dummy data, the typical wall clock time for your tests in this project should be less than what is shown here.

| | Test case A | Test case B |
|---:|---:|---:|
| Typical wall clock time (hours) | 6 | 12 |
| Typical job size (nodes) | 16 | 16 |
| Memory per node (GB) | 12 | 18 |
| Maximum number of input files in a job | 4 | 6 |
| Maximum number of output files in a job | 8 | 14 |
| Library used for I/O | HDF5 | HDF5 |

Table 1: Workflow parameters of the two test cases used during project development.

## 5 Resource Justification

TODO: The request of the annual amount of node hours should be clearly linked with the node hours used by the representative benchmarks: the number of node hours consumed by a simulation is computed multiplying the number of nodes by the wall time expressed in hours for a typical production run.

For example, the optimal job size of the representative benchmark is 16 nodes. Assuming the corresponding wall time for a production run is 141 s, which is then equivalent to $\sim 0.6267$ node hours, as a result of the following product:

$$0.6267 \text{ node hours} = 16 \text{ nodes} \times \frac{141s}{3600\frac{s}{\text{hour}}}$$

The benchmark is short and represents in general a small number of iterations (cycles, timesteps or an equivalent measure), while in a real production simulation we will need to extend it.

Therefore we will estimate how many iterations should be necessary to complete a simulation in production. Furthermore, your project might contain multiple runs to complete a task, each of them requiring several sets of simulations to complete. The total resource request will sum the corresponding node hours obtained multiplying all the factors reported in Table 2.

| | Test case A | Test case B |
|---:|---:|---:|
| Simulations per task | 2 | 4 |
| Iterations per simulation | 5000 | 10000 |
| node hours per iteration | 0.6267 | 0.6267 |
| Total node hours | 6267 | 25068 |

Table 2: Justification of the resource request

The small example above will request a total of 31335 node hours to cover both test cases in the proposal to the supercomputing center. Note that the simulations listed in Table 2 must usually be motivated by a project plan that accommodates the proposal. We will omit this section for the purpose of the CS205 project.

## References

[1] Leslie Lamport. *LaTeX: a document preparation system.* Addison-Wesley, Reading, Massachusetts, 1993.