

Serie Profiling

Exercise 1. Hand measuring

- Add a timer around the call to `compute_step()`, you can see how it is done in `pi.cc`
- Run the code for different N and record the time per call of `compute_step()`
- Plot the times versus N

Exercise 2. Sequential profiling with `gprof`

Reuse a small size of N for this exercise

- To use **`gprof`** you have to add the **`-pg`** option at compilation. Due the fact this is a C++ code it is highly recommended to at least have one level of optimization.

```
$ CXXFLAGS='-pg -g -O1' make
```

- Run you code, it should generate a `gmon.out` file.
- This file can be visualized with **`gprof`**

```
$ gprof ./poisson
```

- Where is most of the time spend, what can you do about it ?

Exercise 3. Reduing everything differently

- Be sure you removed the dumping function
- Recompile with the following flags

```
$ CXXFLAGS='-pg -g -O2' make
```
- Recompile with the following flags

```
$ CXXFLAGS='-pg -g -O3' make
```
- Recompile with the following flags

```
$ CXXFLAGS='-pg -g -O0' make
```