Performance Metrics

Rules:

- This homework is graded and grants you additional points in your final grade.
- You have to work on the exercises and submit your answers individually.
- You have to submit one single PDF document.
- Use preferably LaTeX or a word processor. Pay attention to the formatting of equations.
- Submit by **email** to `fabregat@aices.rwth-aachen.de` .
- Deadline: **18th of November, 23:59**.

1 Performance Metrics

1.1 Speedup

You are given the following code.

```
int simulation( int n, int k )
{
   double *M = (double *) malloc( ... );
   double res;

   initialize( M, n, k );

   res = process( M, n, k );

   post_process( res );

   cleanup( M, n, k );

   return 0;
}
```

Independently of `n` and `k`, the following relations hold.

- T_1(`initialize`) $= \frac{1}{10}\, T_1$(`simulation`)
- T_1(`process`) $= \frac{15}{20}\, T_1$(`simulation`)
- T_1(`post_process`) $= \frac{1}{10}\, T_1$(`simulation`)
- T_1(`cleanup`) $= \frac{1}{20}\, T_1$(`simulation`)

The functions `initialize` and `cleanup` are strictly sequential.

Answer the following questions.

a) When parallelizing `simulation`, what is the maximum achievable speedup?

b) Use your favorite tool to plot the speedup of `simulation` vs the number of processors in the range [1, 1024].

Now assume that the speedup of the function `process` with p processors is $p \times 0.99^p$, and the speedup of `post_process` is still p.

c) Plot once again the speedup of `simulation` vs the number of processors in the same range.

d) Give an estimate of the optimal number of processors to maximize the speedup.

1.2 Scalability

To solve a certain problem $\mathcal{P}$, you are given algorithms $\mathcal{A}$ and $\mathcal{B}$. Both algorithms have a quadratic complexity with respect to n, $O(n^2)$, where n is the size of the input to $\mathcal{P}$. In terms of workspace, both algorithms use $3n$ memory locations.

When solving a problem of size $n = 10{,}000$ with p processors, the following execution times (in seconds) were observed.

p	Alg. $\mathcal{A}$	Alg. $\mathcal{B}$
512	322	10.8
1024	162	10.4
2048	81.5	10.2
4096	41	10.1

Answer the questions below. In all cases, justify your answer with the necessary calculations.

Strong scalability

You have access to a computer with 32,000 processors. Assuming the parallel efficiency follows the trend observed in the table above, which algorithm do you choose to solve a problem of size 10,000? Why?

Weak scalability

Take the configuration $n = 10{,}000$ and $p = 512$ as the reference load in terms of workspace per processor. You want to solve a problem of size 40,000, maintaning the ratio workspace/processor. Which algorithm do you choose? Why?