

APMA 4302 Methods - Homework 1

Marc Spiegelman

January 26, 2026

Problem 1

A slightly different modification of problem 1.1 from the Beuler textbook.

Modify the code `expx.c` to compute a well-balanced N term Taylor Polynomial approximation for e^x for both positive and negative values of x . The command line should read

```
$ mpiexec -n nP ./expx -x x -N N
```

where x is the point at which to evaluate e^x , N is the number of terms in the Taylor polynomial, and nP is the number of processes to use.

The code should have the following properties:

- you should use PETSc options handling to read in the command line arguments.
- The Taylor polynomial should be computed in parallel using nP processes.
- For each process, the work should be roughly $O(N/nP)$.
- The code should work for both positive and negative values of x .
- The answer should be printed out from process 0 only, and should include the relative error compared to the value computed by the `exp()` function from the standard C, in multiples of machine precision.
- You can use the constant `PETSC_MACHINE_EPSILON` to get machine precision.