CUDA: Polynomial Expansion

1 Preliminary

To use CUDA, add `module load cuda/11.2` in your `.bashrc`.

2 Polynomial expansion

The problem is simple. It is to compute a polynomial function F of degree d:

$$F(x) = \sum_{i=0}^{d} a_i x^i$$

for an array of n values of x.

(The code for polynomial expansion on the CPU is provided.)

Question: Write a simple CUDA code that allocates and fill an array on the CPU and transfer it to the GPU. (Take array size as a parameter)

Question: Compute the polynomial expansion of each element of the array on the GPU. (Take block size and degree of the polynomial as a parameter.)

Question: Bring the results back on the CPU and confirm the GPU code is correct. You can test your code by running `make test` which will queue both CPU and GPU jobs to the cluster which will ouput `cputest-`***`xyz`***`.out` which contains the desired answer, and `gputest-`***`xyz`***`.out` which contain your gpu code answer.

Hint: Remember to check explicitly for errors!

3 Measurements

Question: Measure the runtimes of the codes with `make bench` for polynomial function ofdegree 1, 10, 100, and 1000.

Question: Compare to the performance of the CPU implementation. You can compute a speedup table with `make table`, this will output the time and speedup in the form of a text table in `resulttable.txt`. Which is faster in which configuration? Why do you think that is?