CUBLAS

In this exercise, we use the BLAS3 routine DGEMM in CUBLAS to perform a matrix-matrix multiplication using double precision. GEMM is defined as

$$C = \alpha AB + \beta C.$$

where A,B, and C are matrices and α and β are scalars. In our case, we only want to multiply *A* and *B* so we set $\beta = 0$.

The program (*dgemm_um.cpp*) allocates space for two square matrices *A* and *B* and fills them with random numbers. The data is then copied to the GPU where the DGEMM routine is executed and the result is returned back to the host.

Todo

Compile the code by calling `make dgemm_um`. Run the code for a few different matrix sizes. Is the performance what you would have expected? Compare with the results from the previous exercise.

To convert the time *t* needed to calculate the matrix product of two matrices of size *n* into GFLOP/s use

$$\text{GFLOP/s} = 2 * n * n * n/t * 10^{-9}$$

Read the code and find the CUDA specific calls.

Extra credit

Replace random number generator with call to CURAND and remove unneccessary code.