

Practice 4: Implementation of Matrix Multiplication

Objective:

- To implement a basic matrix multiplication routine using global memory.
- To implement a tiled matrix multiplication using shared memory.

Practice 4.1: Edit the code to perform the following instructions:

- Allocate device memory
 - For input and output matrices of size 200 x 200
 - You may initialize input matrices with arbitrary values
- Copy host memory to device
- Implement a CUDA kernel to perform matrix multiplication on global memory
- Implement a host function to test the resulting output matrix
- Initialize thread block and kernel grid dimensions (thread block dimension is 16 x 16)
- Invoke the CUDA kernel
- Copy results from device to host
- Deallocate device memory
- Invoke the host function to check the result

Practice 4.2: Edit the code to perform the following instructions:

- Allocate device memory
 - For input and output matrices of size 200 x 200
 - You may initialize input matrices with arbitrary values
- Copy host memory to device
- Implement a CUDA kernel to perform matrix multiplication using shared memory and tiling
- Implement a host function to test the resulting output matrix
- Initialize thread block and kernel grid dimensions (thread block dimension is 16 x 16)
- Invoke the CUDA kernel
- Copy results from device to host
- Deallocate device memory
- Invoke the host function to check the result