



Institut of Numerical Mathematics

Dr. Andreas F. Borchert and Dr. Michael C. Lehn
Mladjan Radic

20 November 2017
Quiz 4

High Performance Computing I (WS 2016/2017)

Deadline: 27 November 2017, 2pm

Implement the cache-optimized GEMM operation discussed in session 9. Your submitted file “quiz04.cpp” should contain only:

- A C interface `dgemm` that calls your C++ implementation. How such an interface must be defined is shown in the skeleton below.
- Your C++ implementation must be defined in the namespace ‘`ulmblas`’. Include only what is needed for your C++ implementation of the cache-optimized GEMM operation.
- Include directives for all headers needed by your code.

Try to compile an object file in order to test (for yourself) whether your code contains all required auxiliary functions:

```
thales$ g++ -Wall -std=c++11 -m64 -c quiz04.cpp
```

This should generate (without any warnings) a file named `quiz04.o`.

You can extend the following skeleton with your implementation:

Listing 1: Skeleton for "quiz04.cpp"

```
/* TODO: include header files required by your code */

/* TODO: functions required by dgemm */

extern "C" {

void
dgemm(std::size_t m, std::size_t n, std::size_t k,
      double alpha,
      const double *A, std::ptrdiff_t incRowA, std::ptrdiff_t incColA,
      const double *B, std::ptrdiff_t incRowB, std::ptrdiff_t incColB,
      double beta,
      double *C, std::ptrdiff_t incRowC, std::ptrdiff_t incColC)
{
    ulmblas::gemm(m, n, k,
                  alpha,
                  A, incRowA, incColA,
                  B, incRowB, incColB,
                  beta,
                  C, incRowC, incColC);
}

} // extern "C"
```

Please submit your program "quiz04.cpp" as follows:

```
thales$ submit hpc quiz04 quiz04.cpp
```