



Institut of Numerical Mathematics

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

10 November 2017  
Quiz 3

## High Performance Computing I (WS 2016/2017)

Deadline: 17 November 2017, 2pm

Implement the GEMM operation using buffers as discussed in session 6. Your submitted file “quiz03.c” should contain only:

- The function `dgemm_buf` with the signature as specified in session 6.
- Global variables `M_C`, `N_C` and `K_C` of type `size_t`. These global variables will be initialized by our test suite. So your code can assume that these variables have proper values for blocking matrices.
- Any function called from your function `dgemm_buf`.
- Include directives for all headers needed by your code.

It must not contain any other global variable. So you have to use local buffers and dynamic memory allocation. Make sure you release allocated memory. Also check whether `malloc` returns a null pointer. In this case, release any previously allocated memory and call function `abort()` declared in `stdlib.h`.

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

```
thales$ gcc -Wall -std=c11 -m64 -c quiz03.c
```

This should generate (without any warnings) a file named `quiz03.o`.

You can extend the following skeleton with your implementation:

Listing 1: Skeleton for "quiz03.c"

---

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

size_t M_C, K_C, N_C;

/* TODO: functions required by dgemm_buf */

void
dgemm_buf(size_t m, size_t n, size_t k,
          double alpha,
          const double *A, ptrdiff_t incRowA, ptrdiff_t incColA,
          const double *B, ptrdiff_t incRowB, ptrdiff_t incColB,
          double beta,
          double *C, ptrdiff_t incRowC, ptrdiff_t incColC)
{
    /* TODO: your code */
}
```

---

Please submit your program "quiz03.c" as follows:

```
thales$ submit hpc quiz03 quiz03.c
```