



Institut of Numerical Mathematics

Dr. Andreas F. Borchert and Dr. Michael C. Lehn  
Constantin Greif

26 October 2018  
Quiz 1

## High Performance Computing I (WS 2018/2019)

Deadline: 02 November 2018, 2pm

### Notice:

Please register yourself in SLC for HPC I. The registration will allow you to submit your answers to the following questions. Please type your responses in two simple text files named "quiz01.txt" and "quiz01.c". Submit these files to our server *theon* using the following command:

```
theon$ submit hpc quiz01 quiz01.txt quiz01.c
```

### Question 1

In BLAS, the GEMV (general matrix-vector product) operation is defined as

$$y \leftarrow \alpha Ax + \beta y \quad \text{where} \quad \alpha, \beta \in \mathbb{F}, A \in \mathbb{F}^{m \times n}, x \in \mathbb{F}^n, \text{ and } y \in \mathbb{F}^m$$

where  $\mathbb{F}$  denotes a set of floating point numbers (e.g. single or double precision floating point numbers). Please note:

1. In case  $\beta = 0$  or  $n = 0$  the vector  $y$  might contain NaN (not a number) entries.
2. In case  $\alpha = 0$  then vector  $x$  or matrix  $A$  might contain NaN (not a number) entries.

Write down the algorithm in *quiz01.txt* in mathematical terms for a simple cache optimization where it does not matter whether  $A$  is organized in column or row major.

## Question 2

Implement the algorithm for values of type *double* as function *dgemv* in *quiz01.c*. Use the following skeleton:

```
#include <stddef.h>

void
dgemv(size_t m, size_t n,
      double alpha,
      const double *A, ptrdiff_t incRowA, ptrdiff_t incColA,
      const double *x, ptrdiff_t incX,
      double beta,
      double *y, ptrdiff_t incY)
{
    // your code
}
```

The source file *quiz01.c* must not contain a *main* function. Beside that, you can define further auxiliary functions if needed.

## Question 3

Please add the answers to the following questions to *quiz01.txt*:

- What is the difference between wall time and CPU time?
- What is the purpose of the *MIN\_T* macro parameter in the benchmark program?
- Which matrix organization is favored by the *initMatrix* function in *bench\_gemm\_sol.c* in the third session?