

## Blatt 9 Osterchallenge

### Aufgabe 1 (Normal schnell;8 Punkte; schriftlich)

Schreiben Sie ein Programm, das auf einem Core  $10^8$  normalverteilte Zufallszahlen in doppelter Präzision erzeugt und für die Erzeugung von Pseudozufallszahlen einen Mersenne-Twister verwendet.

Hilfreich zu der Umwandlung zu normalverteilten Zufallszahlen können die Hinweise aus dem Skript zur Vorlesung ‘Monte Carlo Methods’ sein.

Verwenden Sie dabei die folgende Funktionssignatur in einer separaten Header-Datei:

```
void rnorm_NAME(double **result , unsigned int n);
```

Das heißt, übergeben wird ein Pointer auf einen Pointer, der dann auf den Ergebnisvektor zeigt. Die Speicherallokation erfolgt ebenso innerhalb dieser Funktion. Eine Main-Funktion könnte dann derart aussehen:

```
#include "NAME_ostern.h"

int main(){
    unsigned int n = 100'000'000;
    unsigned int m = 10;
    double *result;

    rnorm_NAME(&result , n);

    for(int i = 0; i < m; i++)printf("%f\n" , result[i]);
    printf("\n");

    free(result);
    return 0;
}
```

Abgeben werden Sie dann nur die Header-Datei mit Bezeichnung: "*NAME\_ostern.h*". Als Kommentar am Anfang der Header-Datei können Sie ihr compile-flags angeben. Ist dies nicht der Fall wird die compilierung in der einfachsten funktionierenden Version vorgenommen.

Verglichen wird die Performance der obigen Funktion auf einem einzigen Kern eines AMD EPYC 9454 48-Core Processor (Login-Node auf dem Cluster3). Zur Evaluation wird der Durchschnitt von 10 Versuchen á  $10^8$  Zufallszahlen verwendet.

Abgabe der Header-Datei bis zum 27.04.25. 23:59 Uhr an daniel.mann@uni-mannheim.de.