

---

Übung zur Vorlesung  
**Wissenschaftliches Rechnen**  
WS 2019/20 — Blatt 6

---

**Abgabe:** 25.11.2019, 10:00 Uhr, Briefkasten 112  
Code zusätzlich per e-mail an `marcel.koch@uni-muenster.de`

**Achtung:** Achten Sie darauf, Ihre Programme ordentlich zu formatieren und gut zu kommentieren. Die Form wird mit in die Bewertung eingehen.

**Aufgabe 1** (Roofline Model für PCG) (4 Punkte)

Betrachten Sie folgende Beschreibung des PCG Algorithmus:

```
1  r0 := b - Ax0
2  if r0 is sufficiently small, then return x0
3  z0 := M^-1 r0
4  p0 := z0
5  k := 0
6  for k:= 0 to maxit do{
7      alpha_k := (r_k^T z_k) / (p_k^T A p_k)
8      x_{k+1} := x_k + alpha_k p_k
9      r_{k+1} := r_k - alpha_k A p_k
10     if r0 is sufficiently small, then return x_{k+1}
11     z_{k+1} := M^-1 r_{k+1}
12     beta_k := (z_{k+1}^T r_{k+1}) / (z_k^T r_k)
13     p_{k+1} := z_{k+1} + beta_k p_k
14 }
```

$A \in \mathbb{R}^{N \times N}$  und  $M^{-1} \in \mathbb{R}^{N \times N}$  werden als dünnbesetzte Matrizen betrachtet. Nehmen Sie für die Aufgaben an, dass  $A$  im Durchschnitt  $n$  und  $M^{-1}$  im Schnitt  $m$  nicht-null Einträge pro Zeile haben. Sie können davon ausgehen, dass  $N$  so groß ist, dass die Matrizen und alle Vektoren immer aus dem Hauptarbeitspeicher geladen werden.

Im weitem werden zwei Prozessoren mit den folgenden Eigenschaften (auf einen Kern bezogen) betrachtet:

	Skylake Server (2017)	PowerPC 450 (2006)
Taktrate	2.2 GHz/s	850 MHz/s
FLOP/cycle	32	1
Bandbreite	10 GB/s	3.7 GB/s

(a) Bestimmen Sie die arithmetische Intensität der einzelnen Schritte einer PCG-Iteration.

- (b) Skizzieren Sie ein Roofline Model für beide Prozessoren und tragen Sie die Teilschritte des PCG mit unterschiedlicher arithmetischer Intensität dort ein.
- (c) Bestimmen Sie die Laufzeit einer Iteration in Abhängigkeit von  $N$  und  $n, m$ .

**Aufgabe 2** (SIMD Instruktionen)

(4 Punkte)

Besuchen Sie die Seite [godbolt.org](http://godbolt.org) und fügen Sie in dem Editor Window eine Funktion ein, die zwei `std::array<Real,d>` als Referenz bekommt und das erste Array auf das Zweite addiert.

- (a) Betrachten Sie den Assembler-Output für die Fälle `Real=double` und  $d \in \{3, 4\}$ . Erklären Assemblercode in beiden Fällen zeilenweise (beispielsweise durch Kommentare im Assemblercode).
- (b) Worin liegt der Grund für die Benutzung unterschiedlicher Assemblerbefehle für die beiden Fälle?
- (c) Was können Sie daraus für Ihren Code zum  $n$ -Körper Problem schließen, insbesondere bzgl. der Datenstruktur und den Vektoroperationen?

Hinweis 2: Falls Sie Informationen bezüglich der Assemblerbefehle benötigen, hilft Ihnen das *Intel 64 and IA-32 Architectures Software Developer's Manual* weiter. Dieses ist unter <http://www.intel.com> zum Download verfügbar.