

Serie 3

Aufgabe 3.1 (Funktionsauswertungen)

Moderne Compiler sind bereits in der Lage, viele Schleifen automatisch zu vektorisieren. Bei Aufrufen komplexer und verschachtelter Funktionen, wie etwa Sinus oder Cosinus, stoßen sie jedoch an ihre Grenzen.

Im Folgenden wollen wir uns exemplarisch mit der Auswertung der Cosinus-Funktion beschäftigen:

- (a) Werten Sie die `cos`-Funktion für ein großes Array sowohl skalar als auch vektorisiert unter Verwendung von AVX bzw. AVX512 aus.

Nutzen Sie hierzu für $m \in \mathbb{N}$ die *Taylor-Reihe* des Cosinus:

$$\cos(x) \approx \sum_{k=0}^m (-1)^k \frac{x^{2k}}{(2k)!}.$$

Bei der Auswertung der Taylor-Reihe verwenden Sie bitte das *Horner-Schema* und stellen Sie sicher, dass der absolute Fehler der Approximation im Bereich der Maschinengenauigkeit für den Datentyp `double` liegt.

Was erwarten Sie hinsichtlich der notwendigen Anzahl an Termen, wenn der Definitionsbereich vergrößert bzw. verkleinert wird?

- (b) In der Signalverarbeitung wird häufig die *Diskrete Cosinus-Transformation (DCT)* eingesetzt. Für deren Berechnung ist es entscheidend, Summen mit $N \in \mathbb{N}$ Cosinus-Termen effizient auswerten zu können:

$$f(x) = \sum_{\ell=1}^N \beta_{\ell} \cos(\alpha_{\ell} x), \quad \alpha_{\ell}, \beta_{\ell} \in \mathbb{R} \text{ für alle } \ell \in \{1, \dots, N\}.$$

Implementieren Sie diese Funktion ebenfalls sowohl skalar als auch vektorisiert in Ihrem Code.

Welchen zusätzlichen Rechenaufwand erwarten Sie im Vergleich zur einfachen Auswertung eines einzelnen Cosinus-Terms wie in Aufgabenteil (a)?