Prof. Dr. Gerhard Zumbusch
Wissenschaftliches Rechnen I, Parallele Algorithmen – WS 2017/18

3. Übungsblatt

Thema: Vektorisierung mit SIMD

Moderne Prozessoren sind in der Lage, in einen Befehl mehrere Daten parallel zu verarbeiten, was als SIMD (Single Instruction Multiple Data) bezeichnet wird. Für die x86 Architektur gibt es inzwischen die Befehlssätze SSE (128 bit), AVX (256 bit) und AVX-512 in verschiedenen Versionen.

In SSE werden 128-Bit-Register zur Verfügung gestellt, die beispielsweise für 4 Gleitkommazahlen einfacher Genauigkeit (32Bit, Typ `float`) genutzt werden können. Operationen können dann gleichzeitig mit allen vier Zahlen ausgeführt werden. Intrinsics-Anleitung:
`https://software.intel.com/sites/landingpage/IntrinsicsGuide`.

Aufgabe 6

Skalarprodukt

Exemplarisch wollen wir wieder das Skalarprodukt der Vektoren

$$\vec{a} = (1, 2, 3, \ldots, n)\,, \quad \vec{b} = (n, n-1, n-2, \ldots, 1)$$

berechnen. Der Einfachheit nehmen an, dass n durch 4 teilbar ist. Es sollen immer je 4 Komponenten von $\vec{a}$ und $\vec{b}$ elementweise multipliziert werden und das Ergebnis auf eine ebenfalls 4-elementige Summe s addiert werden. D. h. $s_1 = a_1b_1 + a_5b_5 + a_9b_9 + \ldots$, $s_2 = a_2b_2 + a_6b_6 + \ldots$, etc. Zum Schluss müssen dann noch die 4 Komponenten von s addiert werden.

Verwenden Sie dazu die folgenden (SSE-) Befehle:

`posix_memalign, _mm_load_ps, _mm_mul_ps, _mm_add_ps, _mm_store_ss`

und eine Kombination von SSE Befehlen für die Summe von 4 Komponenten (horizontal add). Übersetzen Sie das Programm mit der Option `-msse4`.

Vergleichen Sie die Laufzeit des Programm mit einem Programm ohne SIMD-Verwendung. Nehmen Sie nicht zu große Vektoren, sonst messen Sie hauptsächlich die Speicherzugriffszeit, anstatt eigentlichen Rechenzeit. Nehmen Sie beispielsweise $n = 5000$ Elemente und 100000 Wiederholungen, die Rechenzeit können Sie wieder innerhalb des Programms mit dem `clock gettime`-Kommando messen.

Aufgabe 7

Newton Iteration

Schreiben Sie ein Programm, dass nur mit SSE-Vektor-Additionen, Subtraktionen und Multiplikationen eine Iteration einer Newton Iteration implementiert für die Berechnung von jeweils vier mal

a) Kehrwert $\frac{1}{x}$

b) Kehrwert der Wurzel $\frac{1}{\sqrt{x}}$

Verwenden Sie als Startwert der Iteration die jeweiligen SSE-Näherungsfunktionen. Testen Sie die Programme auf Korrektheit und Genauigkeit für verschiedene Werte x, und auf Rechenzeit/ Zahl der Prozessortakte im Vergleich zur direkten Berechnung der Werte mit Division und Quadratwurzel.

Aufgabe 8

Bedingte Operationen

Schreiben Sie ein Programm, dass nur mit SSE-Vektoroperationen eine Vektor-Exponentialfunction implementiert. Dazu wird für positive Argumente die Potenzreihe ausgewertet, für negative Werte ist $e^x = 1/e^{-x}$. Die Potenzreihe soll mit dem Hornerschema ausgwertet werden.

a) verwenden Sie eine feste Zahl von Termen

b) konstruieren Sie ein Kriterium, um mit einer vom Argument abhängigen Zahl von Termen zu arbeiten.

Testen Sie die Programme auf Korrektheit und Genauigkeit für verschiedene Werte x, und auf Rechenzeit/ Zahl der Prozessortakte im Vergleich zur skalaren Berechnung der Werte mit der `exp` Funktion.

Aufgabe 9

Benutzen Sie ihr serielles Programm aus der vorhergehenden Aufgabe, (Jacobi Iteration und upwind-Differenzen) um dieses nun zu vektorisieren.

Die Vektorisierung gestaltet sich sehr einfach, Schleifen werden nicht mehr über einzelne Elemente, sondern über ganze Vektoren laufen. Wie gehen Sie mit linken/rechten Nachbar-Gitterpunkten und wie mit oberen/unteren Nachbar-Gitterpunkten um?

a) Verwenden Sie 2000 Iterationen und 1024×1024 innere Gitterpunkte um die Laufzeiten des Programms mit und ohne SIMD Vektoren zu messen. Diskutieren Sie das Ergebnis.

b) Was ändert sich beim Übergang zum Gauß-Seidel-Verfahren, wie könnte man nun vorgehen?

Ersetzen Sie nun die Jacobi-Iteration durch ein Gradientenverfahren. Wie vektorisieren Sie die Skalarprodukte?

a) Verwenden Sie 2000 Iterationen und 1024×1024 innere Gitterpunkte um die Laufzeiten des Programms mit und ohne SIMD Vektoren zu messen. Diskutieren Sie das Ergebnis.