

Projekt 4: Multiplikation großer Zahlen mit Standard-FFT

Es wird die Multiplikation großer ganzer Zahlen mittels der Standard-FFT untersucht. Sei $b \in \mathbb{N}$, $b \geq 2$ die Basis. Die Zahlen $x, y \in \mathbb{N}$ sollen die Darstellung

$$x = \sum_{i=0}^n x_i b^i, \quad y = \sum_{i=0}^n y_i b^i,$$

haben, wobei die Zahlen $x_i, y_i \in \{0, \dots, b-1\}$ sind.

1. Formulieren Sie einen Algorithmus, der zu gegebenen Vektoren $(x_i)_{i=0}^n, (y_i)_{i=0}^n$, das Produkt $z = xy$ in der Darstellung $z = \sum_{i=0}^{2n} z_i b^i$, $z_i \in \{0, \dots, b-1\}$ mit Rechenzeitaufwand $O(n \log n)$ bestimmt.
2. Programmieren Sie Ihren Algorithmus mit der "normalen" FFT aus der Vorlesung. Gehen Sie hierzu wie folgt vor:
 - a) Programmieren Sie den FFT- und IFFT-Algorithmus für `double complex` Zahlen¹. Gehen Sie insbesondere so vor, daß die Potenzen aller benötigten Einheitswurzeln *vor* den Aufrufen von FFT und IFFT berechnet werden.
 - b) Nach der Berechnung der Faltungsprodukte haben Sie eine Darstellung der Form $z = \sum_i \tilde{z}_i b^i$ mit $\tilde{z}_i \in \mathbb{C}$, aber nicht unbedingt $\tilde{z}_i \in \{0, \dots, b-1\}$. Runden Sie die \tilde{z}_i zu ganzen Zahlen, und berechnen Sie eine Darstellung $z = \sum_i z_i b^i$ mit $z_i \in \{0, \dots, b-1\}$.

Auf der Web-page http://www.math.tuwien.ac.at/~melenk/teach/numerik_WS0809/projekte finden Sie ein kleines C++-Programm (mit Hinweisen, wie es in ein C-Programm einzubinden ist), welches Dezimalzahlen aus einer Datei einliest und als Vektor von Ziffern bezüglich einer gewählten Basis b umwandelt.

3. Sei $\mathcal{F}_n : \mathbb{C}_{per}^n \rightarrow \mathbb{C}_{per}^n$ die DFT auf dem Raum der n -periodischen Folgen. Definieren Sie auf \mathbb{C}_{per}^n die Normen $\|\cdot\|_p$ für $p \in \{1, 2, \infty\}$ durch

$$\|f\|_2 := \left(\sum_{j=0}^{n-1} |f_j|^2\right)^{1/2}, \quad \|f\|_1 := \sum_{j=0}^{n-1} |f_j|, \quad \|f\|_\infty := \max_{j=0, \dots, n-1} |f_j|,$$

Zeigen Sie:

$$\begin{aligned} \|\mathcal{F}_n(f)\|_2 &= \sqrt{n} \|f\|_2, & \|\mathcal{F}_n(f)\|_\infty &\leq \|f\|_1, & \|\mathcal{F}_n(f)\|_1 &\leq n^2 \|f\|_\infty, \\ \|\mathcal{F}_n^{-1}(f)\|_\infty &\leq n^{-1} \|f\|_1, & \|\mathcal{F}_n^{-1}(f)\|_1 &\leq n \|f\|_\infty. \end{aligned}$$

4. Der Algorithmus aus Aufg. 1 verwendet die FFT und die IFFT zur Berechnung von Faltungsprodukten von Vektoren, die nur ganze Zahlen enthalten. Das Ergebnis der Faltung ist wieder ein Vektor, der nur ganze Zahlen enthält. Die von Ihnen programmierte FFT und IFFT arbeitet jedoch mit Gleitkommaarithmetik. Versuchen Sie nun, den Rundungsfehler abzuschätzen. Gehen Sie hierzu wie folgt vor: Nehmen Sie an, daß die Realisierungen $\{+*, -*, **, /*\}$ der

¹in C müssen Sie dazu natürlich die Arithmetik realisieren

4 Grundrechenarten auf dem Computer folgendes erfüllen: Für zwei Gleitkommazahlen x, y gibt es ein $\delta = \delta(x, y)$ mit $|\delta| \leq \mathbf{eps}$, so daß

$$x \mathop{\text{op}}^* y = (x \mathop{\text{op}} y)(1 + \delta) \quad \mathop{\text{op}} \in \{+, -, *, /\}.$$

Weiter nehmen Sie an, daß Sie die n -te Einheitswurzel ω mit einem relativen Fehler \mathbf{eps} bestimmen können, d.h. die Einheitswurzel $\tilde{\omega}$, die Sie berechnen, erfüllt

$$\omega = \tilde{\omega}(1 + \delta), \quad |\delta| \leq \mathbf{eps}.$$

Weiter definieren wir für $j \in \mathbb{N}_0$ die Zahl

$$\gamma_j := \frac{j \mathbf{eps}}{1 - j \mathbf{eps}}$$

mit der impliziten Annahme, daß $j \mathbf{eps} < 1$ wann immer man γ_j hinschreibt.

Verwenden Sie folgendes Resultat, welches sich recht einfach bestätigen läßt : Falls zwei Zahlen θ_j, θ_n die Bedingungen $|\theta_j| \leq \gamma_j, |\theta_n| \leq \gamma_n$ erfüllen, dann gilt

$$(1 + \theta_j)(1 + \theta_n) = 1 + \theta_{j+n}$$

für geeignetes θ_{j+n} mit $|\theta_{j+n}| \leq \gamma_{j+n}$.

a) Zeigen Sie, daß

$$\prod_{j=1}^i \left(1 + \gamma_{\frac{n}{2^{j-1}}-1}\right) \leq (1 + \gamma_{2n}). \quad (1)$$

b) Überlegen Sie sich, daß alle berechneten Potenzen $\tilde{\omega}^j$ der Einheitswurzel, die Sie berechnen, die Bedingung

$$\tilde{\omega}^j = \begin{cases} \omega^j(1 + \theta_{2^{j-1}}) & j \geq 1 \\ \omega^j & j = 0. \end{cases}$$

für geeignetes $|\theta_j| \leq \gamma_n$ erfüllen.

c) Bezeichne \mathbf{FFT}_n die exakte FFT und $\widetilde{\mathbf{FFT}}_n$ die in Gleitkommaarithmetik mittels Ihres Algorithmus berechnete. Sei für jedes $n \in \mathbb{N}$ die Zahl $E_\infty(n)$ die bestmöglichen Konstanten, so daß für alle Vektoren $y \in \mathbb{C}_{per}^n$, die aus Gleitkommazahlen bestehen, gilt:

$$\|\mathbf{FFT}_n(y) - \widetilde{\mathbf{FFT}}_n(y)\|_\infty \leq E_\infty(n) \|y\|_1. \quad (2)$$

Zeigen Sie, daß für $n = 2^L$ gilt:

$$E_\infty(n) \leq \sum_{i=1}^L \gamma_{\frac{n}{2^{i-1}}-1} \prod_{j=1}^{i-1} (1 + \gamma_{\frac{n}{2^{j-1}}-1}) \leq \gamma_{n-1} L (1 + \gamma_{2n}). \quad (3)$$

Hinweise: Überlegen Sie sich, daß die Folgen $(\tilde{g}_j)_{j=0}^{n/2-1}, (g_j)_{j=0}^{n/2-1}$, sowie $(\tilde{h}_j)_{j=0}^{n/2-1}, (h_j)_{j=0}^{n/2-1}$, die durch

$$\tilde{g}_j = (y_j +^* y_{j+n/2}), \quad g_j = (y_j + y_{j+n/2}), \quad \tilde{h}_j = (y_j -^* y_{j+n/2}) * \tilde{\omega}^j, \quad h_j = (y_j - y_{j+n/2}) * \omega^j$$

definiert sind, die Abschätzungen $|g_j - \tilde{g}_j| \leq \gamma_1 |g_j| \leq \gamma_{n-1} |g_j|, |h_j - \tilde{h}_j| \leq \gamma_{n-1} |h_j|$ erfüllen. Versuchen Sie, die Rekursion $E_\infty(n) \leq \gamma_{n-1} + (1 + \gamma_{n-1}) E_\infty(n/2)$ zu zeigen.

Bemerkung: Völlig analog kann man $E_1(n)$ definieren als die bestmögliche Abschätzung der Form

$$\|\text{FFT}_n(y) - \widetilde{\text{FFT}}_n(y)\|_1 \leq E_1(n)\|y\|_\infty \quad (4)$$

definieren. Analog zum Vorgehen bei der Abschätzung von $E_\infty(n)$ könnte man dann die Ungleichung $E_1(n) \leq \gamma_{n-1}n^2L(1 + \gamma_{2n})$ zeigen.

Verwenden Sie im Weiteren die vereinfachte (und ziemlich großzügige!) Abschätzung

$$\begin{aligned} \|\text{FFT}_n(y) - \widetilde{\text{FFT}}_n(y)\|_1 &\leq \varepsilon_n\|y\|_\infty, & \varepsilon_n &:= (1 + \gamma_{2n})(n^2 \log_2 n)\gamma_{n-1} \\ \|\text{FFT}_n(y) - \widetilde{\text{FFT}}_n(y)\|_\infty &\leq \varepsilon_n\|y\|_1, \end{aligned}$$

sowie (ohne Beweis) die analoge Abschätzung für die inverse FFT:

$$\begin{aligned} \|\text{IFFT}_n(y) - \widetilde{\text{IFFT}}_n(y)\|_1 &\leq \frac{1}{n}\varepsilon_n\|y\|_\infty \\ \|\text{IFFT}_n(y) - \widetilde{\text{IFFT}}_n(y)\|_\infty &\leq \frac{1}{n}\varepsilon_n\|y\|_1. \end{aligned}$$

- d) Damit Ihr Programm das Gewünschte leistet, müssen n , die Basis b sowie die Maschinengenauigkeit eps so sein, daß der Fehler im Endergebnis in jeder Komponente $< 1/2$ ist (Sie runden ja auf den Typ `int`). Schätzen Sie den Fehler ab, der durch Gleitkommaarithmetik in Ihrem Programm bei der Bestimmung der Faltung der Folgen $(x_j)_j$ und $(y_j)_j$ entsteht. Gehen Sie z.B. so vor:

1. Starten Sie mit $\|\tilde{x} - \hat{x}\|_\infty \leq \varepsilon_n\|x\|_1$ und $\|\tilde{y} - \hat{y}\|_1 \leq \varepsilon_n\|y\|_\infty$.
2. Überlegen Sie sich, daß der Vektor $(c_j)_j$, dessen Komponenten in exakter Arithmetik $c_j = \hat{x}_j \cdot \hat{y}_j$ sein sollten, nun approximiert wird durch $\tilde{c} = c + \eta$ mit einem Vektor η , der

$$\|\eta\|_1 \leq \|x\|_1\|y\|_\infty [(2 + \gamma_1)\varepsilon_n^2 + (1 + \gamma_1)(1 + n^2)\varepsilon_n + \gamma_1n^2]$$

erfüllt.

3. Überlegen Sie sich, wie Sie $\|\text{IFFT}_n(c) - \widetilde{\text{IFFT}}_n(\tilde{c})\|_\infty$ abschätzen können.

Reicht $n = 2^8$, $b = 2^3$ bei $\text{eps} = 10^{-16}$? Wieviele Dezimalstellen können die Zahlen haben, die sie bei dieser Kombination von n und b multiplizieren können?

- e) In der vorangegangenen Aufgabe haben Sie theoretisch die Auswirkungen der Gleitkommaarithmetik untersucht. Dies läßt sich auch numerisch überprüfen: Ihr Programm rundet am Schluß das Ergebnis der IFFT auf eine Integerzahl (z.B. `int32`). An dieser Stelle kann die Abweichung des berechneten Ergebnisses von einer ganzen Zahl (wie es in exakter Arithmetik sein sollte) bestimmt werden. Plotten Sie doppelt logarithmisch diesen Fehler über n auf, wobei Sie $b = 2^7$, $x = y = (b - 1)\text{ones}(n, 1)$ wählen und $n = 2^i$, $i = 1, \dots, 15$. Was beobachten Sie?

5. Machen Sie Zeitmessungen für Ihr Programm mit wachsendem n und fester Basis b . Sehen Sie das erwartete Verhalten?