Projekt 10: 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

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

haben, wobei die Zahlen x_i, $y_i \in \{0, \ldots, b-1\}$ 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 $z = \sum_{i=0}^{2n} z_i b^i$, $z_i \in \{0, \ldots, b-1\}$ 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` Zahlen[1]. Gehen Sie insbesondere so vor, daß die Potenzen aller benötigen Einheitswurzeln *vor* den Aufrufen von FFT und IFFT berechnet werden.

b) Nach der Berechnung der Faltungsprodukte haben Sie eine Darstellung der Form $z = \sum_i \widetilde{z}_i b^i$ mit $\widetilde{z}_i \in \mathbb{C}$, aber nicht unbedingt $\widetilde{z}_i \in \{0, \ldots, b-1\}$. Runden Sie die $\widetilde{z}_i$ zu ganzen Zahlen, und berechnen Sie eine Darstellung $z = \sum_i z_i b^i$ mit $z_i \in \{0, \ldots, b-1\}$.

Auf der Web-page `http://www.math.tuwien.ac.at/~melenk/teach/numerik_WS0708/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 $\mathcal{F}_n : \mathbb{C}^n_{per} \to \mathbb{C}^n_{per}$ die DFT auf dem Raum der n-periodischen Folgen. Definieren Sie auf $\mathbb{C}^n_{per}$ die Normen $\|\cdot\|_p$ für $p \in \{1, 2, \infty\}$ durch

$$\|\mathfrak{f}\|_2 := (\sum_{j=0}^{n-1} |\mathfrak{f}_j|^2)^{1/2}, \qquad \|\mathfrak{f}\|_1 := \sum_{j=0}^{n-1} |\mathfrak{f}_j|, \qquad \|\mathfrak{f}\|_\infty := \max_{j=0,\ldots,n-1} |\mathfrak{f}_j|,$$

Zeigen Sie:

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

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

[1] in `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 $|\delta| \leq \texttt{eps}$, so daß

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

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

$$\omega = \widetilde{\omega}(1+\delta), \qquad |\delta| \leq \texttt{eps}.$$

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

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

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

Verwenden Sie folgendes Resultat aus Projekt 3 und 4: 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) Überlegen Sie sich, daß alle berechneten Potenzen $\widetilde{\omega}^j$ der Einheitswurzel, die Sie berechnen, die Bedingung

$$\widetilde{\omega}^j = \omega^j(1+\theta_{2j-1})$$

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

b) Bezeichne $\texttt{FFT}_n$ die exakte FFT und $\widetilde{\texttt{FFT}}_n$ die in Gleitkommaarithmetik mittels Ihres Algorithmus berechnete. Zeigen Sie, daß alle Vektoren $y \in \mathbb{C}^n_{per}$, die aus Gleitkommazahlen bestehen, gilt:

$$\|\texttt{FFT}_n(y) - \widetilde{\texttt{FFT}}_n(y)\|_\infty \;\leq\; \sum_{i=0}^{L-1}(1+\gamma_{n+2})^i\gamma_{n+2}\|y\|_1 \leq C_n\gamma_{n+2}\|y\|_1, \tag{1}$$

$$\|\texttt{FFT}_n(y) - \widetilde{\texttt{FFT}}_n(y)\|_1 \;\leq\; C_n\gamma_{n+2}\|y\|_\infty \tag{2}$$

wobei die Rekursionstiefe $L = \log_2 n$ ist, und die Konstante C_n definiert ist als

$$C_n = L(4(1+\gamma_{n+2}))^L, \qquad L = \log_2 n.$$

Hinweis: Überlegen Sie sich, daß die Folgen $(\widetilde{g}_j)_{j=0}^{n/2-1}$ und $(\widetilde{h}_j)_{j=0}^{n/2-1}$, die durch

$$\widetilde{g}_j = (y_j +^* y_{j+n/2}), \qquad \widetilde{h}_j = (y_j -^* y_{j+n/2}) *^* \widetilde{\omega}^j$$

definiert sind, die Abschätzungen $|g_j - \widetilde{g}_j| \leq \gamma_{n+2}|g_j|$, $|h_j - \widetilde{h}_j| \leq \gamma_{n+2}|h_j|$ erfüllen. Denken Sie sich anschließend γ_{n+2} als fest (d.h. unabhängig von n) und gehen Sie induktiv vor.

Verwenden Sie im Weiteren die vereinfachte Abschätzung

$$\|\texttt{FFT}_n(y) - \widetilde{\texttt{FFT}}_n(y)\|_1 \leq C_n\gamma_{n+2}\|y\|_\infty$$
$$\|\texttt{FFT}_n(y) - \widetilde{\texttt{FFT}}_n(y)\|_\infty \leq C_n\gamma_{n+2}\|y\|_1,$$

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

$$\|\texttt{IFFT}_n(y) - \widetilde{\texttt{IFFT}}_n(y)\|_1 \leq \frac{1}{n}C_n\gamma_{n+2}\|y\|_\infty$$
$$\|\texttt{IFFT}_n(y) - \widetilde{\texttt{IFFT}}_n(y)\|_\infty \leq \frac{1}{n}C_n\gamma_{n+2}\|y\|_1,$$

c) Damit Ihr Programm das Gewünschte leistet, müssen n, die Basis b sowie die Maschinengenauigkeit `eps` so sein, daß der Fehler im Endergebis in jeder Komponenten $< 1/2$ 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 $\|\widetilde{\hat{x}} - \hat{x}\|_\infty \leq C_n \gamma_{n+2} \|x\|_1$ und $\|\widetilde{\hat{y}} - \hat{y}\|_1 \leq C_n \gamma_{n+2} \|y\|_\infty$.
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 $\widetilde{c} = c + \eta$ mit einem Vektor η, der
$$\|\eta\|_1 \leq \|x\|_1 \|y\|_\infty \left[(1+\gamma_1) C_n \gamma_{n+2} (C_n \gamma_{n+2} + n^2 + 1) + \gamma_1 n^2\right]$$
erfüllt.
3. Überlegen Sie sich, wie Sie $\|\mathtt{IFFT}_n(c) - \widetilde{\mathtt{IFFT}}_n(\widetilde{c})\|_\infty$ abschätzen können.

Reicht $n = 2^{10}$, $b = 2^7$ bei $\mathtt{eps} = \mathtt{10}^{-\mathtt{16}}$? Wieviele Dezimalstellen können die Zahlen haben, die sie bei dieser Kombination von n und b multiplizieren können?

d) In der vorangegangen 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 $b = 2^{15}$ $x = (b-1)\mathtt{ones(n,1)}$, $y = (1, 0, 0, \ldots)$ wählen und $n = 2^i$, $i = 1, \ldots, 15$. Was beobachten Sie? Wiederholen Sie die Berechnung mit $b = 2^7$ und $y = x$. Denken Sie, daß Ihre theoretische Abschätzung aus der vorangegangenen Aufgabe scharf in ihrer Abhängigkeit von n ist?

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