NUM UE4

Ida Hönigmann

November 3, 2021

1 Aufgabe 14:

Wir zeigen $V_nA = DV_n$ indem wir auf jeder Seite den Wert in der j-ten Zeile und k-ten Spalte berechnen.

Wir wissen, dass

$$(V_n)_{jk} = w_n^{j*k}.$$

Durch die Diagonalstruktur der Matrix D wird bei einer Multiplikation mit D von links jede Zeile mit dem entsprechenden Wert aus D skaliert, d.h.

$$(DV_n)_{jk} = p(w_n^j) * w_n^{j*k} = (\sum_{l=0}^{n-1} a_l * w_n^{j*l}) * w_n^{j*k} = \sum_{l=0}^{n-1} a_l * w_n^{j(l+k)}.$$

Um $(V_nA)_{jk}$ auszurechnen wollen wir uns zunächst die entsprechende Zeile von V_n und Spalte von A ansehen:

$$\begin{aligned}
(V_nA)_{jk} &= (w_n^{j*0}, w_n^{j*1}, ..., w_n^{j*(n-1)}) * (a_{n-k}, a_{n-k+1}, ..., a_{n-1}, a_0, a_1, ..., a_{n-k-1})^T \\
&= \sum_{l=0}^{k-1} (w_n^{j*l} * a_{n-k+l}) + \sum_{l=0}^{n-k-1} (w_n^{j*(l+k)} * a_l) \\
&= \sum_{l=n-k}^{k-1+n-k} (w_n^{j*(l-n+k)} * a_{n-k+l-n+k}) + \sum_{l=0}^{n-k-1} (w_n^{j*(l+k)} * a_l) \\
&= \sum_{l=n-k}^{n-1} (w_n^{j*(l+k-n)} * a_l) + \sum_{l=0}^{n-k-1} (w_n^{j*(l+k)} * a_l)
\end{aligned}$$

Da w_n die n-te Einheitswurzel ist gilt $(w_n)^n = 1$ und somit

$$w_n^{j(l+k-n)} = \frac{w_n^{j(l+k)}}{w_n^{j*n}} = \frac{w_n^{j(l+k)}}{((w_n)^n)^j} = \frac{w_n^{j(l+k)}}{1} = w_n^{j(l+k)}.$$

Insgesamt ergibt das

$$(V_nA)_{jk} = \sum_{l=n-k}^{n-1} (w_n^{j*(l+k)} * a_l) + \sum_{l=0}^{n-k-1} (w_n^{j*(l+k)} * a_l) = \sum_{l=0}^{n-1} (w_n^{j*(l+k)} * a_l) = (DV_n)_{jk}.$$

Somit ist gezeigt, dass $V_nAV_n^{-1} = D$.

2 Aufgabe 15:

Nach Aufgabe 14 gilt $V_n A V_n^{-1} = D$, was uns $A = V_n^{-1} D V_n$ und weiter $A^{-1} = V_n^{-1} D^{-1} V_n$ liefert. Die Inverse von A existiert wegen der Regularität.

Von V_n wissen wir, dass $V_n^{-1} = \frac{1}{n}\overline{V_n}$ und durch die Diagonalform von D gilt $D^{-1} = \overline{D}$.

Um $Ax = b$ zu lösen können wir einfach $x = A^{-1}b = \frac{1}{n}\overline{V_n}\bar{D}V_n b$ berechnen.

Schauen wir uns zuerst die Berechnung von D an.

Es sei a der Vektor aus dem die zirkulante Matrix A aufgebaut ist und v_1 die erste Spalte von V_n. Für v_1 gilt $v_1 = (1)_{j=0}^{n-1}$. Es gilt $V_n A = D V_n$ und somit auch für die erste Spalte $V_n a = D v_1 = D * (1, 1, ..., 1)^T = (p(1), p(w_n), ..., p(w_n^{n-1}))$, was den Diagonaleinträgen von D entspricht.

Wir können die Multiplikation von $\overline{D}$ mit dem Ergebnis von $w := V_n b$ durch die komponentenweise Multiplikation der Vektoren $\bar{d} := (\overline{p(1)}, \overline{p(w_n)}, ..., \overline{p(w_n^{n-1})})$ mit w vereinfachen, da $\overline{D}$ Diagonalgestalt hat.

Wenn wir nun $\frac{1}{n}\overline{V_n}\bar{D}V_n b$ von rechts nach links ausrechnen erhalten wir folgenden Algorithmus zur Berechnung einer Lösung von $Ax = b$:

Input: Vektor a aus dem die zirkulante Matrix A aufgebaut ist, Vektor b

1. Berechnen von $d := V_n a$ mittels FFT
2. Berechnen von $\bar{d}$ komponentenweise
3. Berechnen von $w := V_n b$ mittels FFT
4. Berechnen von $y := \bar{d} * w$ komponentenweise
5. Berechnen von $\bar{y}$ komponentenweise
6. Berechnen von $z := V_n \bar{y}$ mittels FFT
7. Berechnen von $\bar{z}$ komponentenweise
8. Berechnen von $x := \frac{1}{n}\bar{z}$ komponentenweise

Ergebnis: Vektor x löst nun $Ax = b$

Der Aufwand beträgt 3 FFTs und 5 komponentenweisen Berechnungen, also $3 * (n log_2 n) + 5n$ bzw. $O(n log n)$.

3 Aufgabe 16:

Wenn $A \in \mathbb{K}^{n\times n}$ eine Toeplitz-Matrix ist, können wir sie wie folgt in eine zirkulante Matrix $B \in \mathbb{K}^{2n\times 2n}$ einbetten:

$$B := \begin{pmatrix} a_{-n+1} & a_{n-1} & \cdots & a_1 & a_0 & \cdots & a_{-n+2} \\ a_{-n+2} & a_{-n+1} & \cdots & a_2 & a_1 & \cdots & a_{-n+3} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{-1} & a_{-2} & \cdots & a_{n-1} & a_{n-2} & \cdots & a_0 \\ \multicolumn{4}{c}{\multirow{5}{*}{$\underbrace{\begin{bmatrix} a_0 & a_{-1} & \cdots & a_{-n+1} \\ a_1 & a_0 & \cdots & a_{-n+2} \\ a_2 & a_1 & \cdots & a_{-n+3} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{bmatrix}}_{=A}$}} & a_{n-1} & \cdots & a_1 \\ & & & & a_{-n+1} & \cdots & a_2 \\ & & & & a_{-n+2} & \cdots & a_3 \\ & & & & \vdots & \ddots & \vdots \\ & & & & a_{-1} & \cdots & a_{-n+1} \end{pmatrix}$$

Nennen wir $a := (a_{-n+1}, ..., a_0, ..., a_{n-1})$ den Vektor aus dem die Toeplitz-Matrix A aufgebaut ist. Dann gilt für b, also den Vektor aus dem die zirkulante Matrix B aufgebaut ist $b = a$.

Wenn wir $y := (x_1, x_2, ..., x_n, 0, .., 0)^T \in \mathbb{K}^{2n}$ (als "Erweiterung" von x) setzten. Erhalten wir $z := A*x = (z_1, z_2, ..., z_n)^T \in \mathbb{K}^n$ indem wir uns die letzen n Werte von $B*y = (?, ?, ..., ?, z_1, z_2, ..., z_n)^T$ ansehen.

Nun können wir mit der zirkulanten Matrix B arbeiten, von der wir aus Beispiel 14 wissen, dass $B = V_n^{-1} D V_n = \frac{1}{n}\overline{V_n} D V_n$ gilt.

Um nun $B * y = \frac{1}{n}\overline{V_n} D V_n y$ und somit $A * x$ zu berechnen können wir mit den gleichen Tricks aus dem vorherigen Beispiel folgendem Algorithmus folgen:

Input: Vektor $a = b$ aus dem die Matrizen aufgebaut sind, Vektor x

1. Erweitern von x zu $y := (x_1, ..., x_n, 0, ..., 0) \in \mathbb{K}^{2n}$
2. Berechnen von $d := V_n b$ mittels FFT
3. Berechnen von $f := V_n y$ mittels FFT
4. Berechnen von $g := d * f$ komponentenweise
5. Berechnen von $\bar{g}$ komponentenweise
6. Berechnen von $h := V_n \bar{g}$ mittels FFT
7. Berechnen von $\bar{h}$ komponentenweise
8. Berechnen von $m := \frac{1}{n}\bar{h}$ komponentenweise
9. Abschneiden der letzten n Werte von m zu $z := (m_n, m_{n+1}, ..., m_{2n})$

Ergebnis: $z = Ax$

Der Aufwand des Algorithmus beträgt 3 Mal den Aufwand einer FFT $(2n * log_2(2n))$ und 4 komponentenweise Rechnungen (Aufwand von $2n$), also $O(nlogn)$.