

## 9. Matrixmultiplikation à la Strassen

Betrachte das Produkt zweier  $2 \times 2$ -Matrizen:

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}; \quad c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j}$$

Man beachte, dass diese und die folgenden Formeln **nicht** die Kommutativität der Multiplikation voraussetzen. Sie gelten also auch, falls die  $a_{ij}$ ,  $b_{ij}$ ,  $c_{ij}$  eigentlich  $n \times n$ -Matrizen  $A_{ij}$ ,  $B_{ij}$ ,  $C_{ij}$  sind (jeweils  $i, j \in \{1, 2\}$ ).

Bilde:

$$m_1 := (a_{12} - a_{22})(b_{21} + b_{22})$$

$$m_2 := (a_{11} + a_{22})(b_{11} + b_{22})$$

$$m_3 := (a_{11} - a_{21})(b_{11} + b_{12})$$

$$m_4 := (a_{11} + a_{12})b_{22}$$

$$m_5 := a_{11}(b_{12} - b_{22})$$

$$m_6 := a_{22}(b_{21} - b_{11})$$

$$m_7 := (a_{21} + a_{22})b_{11}$$

Dann gilt:

$$c_{11} := m_1 + m_2 - m_4 + m_6$$

$$c_{12} := m_4 + m_5$$

$$c_{21} := m_6 + m_7$$

$$c_{22} := m_2 - m_3 + m_5 - m_7$$

Sei  $n = 2^k$  und  $M(n)$  die Anzahl arithmetischer Operationen (Mult, Add, Sub), die Strassens Algorithmus bei  $n \times n$  Matrizen benötigt:

$$M(1) = 1$$

$$M(n) = 7M\left(\frac{n}{2}\right) + 18\left(\frac{n}{2}\right)^2$$

Expansion dieser Rekursionsgleichung  $\Rightarrow$

$$\begin{aligned}M(n) &= 7^{k+1} - 6n^2 = 7 \cdot 2^{k \log_2 7} - 6n^2 \\&= 7n^{\log_2 7} - 6n^2 \\&< 7n^{2,80736} - 6n^2 \\&= \mathcal{O}(n^{2,81}).\end{aligned}$$

Der Exponent  $\omega$  für die Matrixmultiplikation ist also  $< 2,81$ .

```

proc MATMULT( $A, B, n$ ) =
  co  $A$  und  $B$  sind  $n \times n$  Matrizen oc
  if  $n < 16$  then berechne  $A \cdot B$  mit klassischer Methode
  elif  $n$  gerade then
    spalte  $A$  und  $B$  in je 4  $\frac{n}{2} \times \frac{n}{2}$  Matrizen auf und wende
    Strassen's Formeln an. Führe die Multiplikationen rekursiv
    mit MATMULT aus.
  else
    spalte  $A$  und  $B$  in je eine  $(n - 1) \times (n - 1)$  Matrix
    ( $A_{11}, B_{11}$ ) und drei verbleibende Blöcke auf. Wende
    MATMULT rekursiv auf  $A_{11}, B_{11}$  an und berechne die
    anderen Produkte mit der klassischen Methode.
  fi

```

## Satz 125

Der MATMULT-Algorithmus hat folgende Eigenschaften:

- 1 Für  $n < 16$  ist die Anzahl der arithmetischen Operationen genauso hoch wie bei der klassischen Methode.
- 2 Für  $n \geq 16$  ist die Anzahl der arithmetischen Operationen echt kleiner als bei der klassischen Methode.
- 3 MATMULT braucht nie mehr als  $4,91n^{\log_2 7}$  arithmetische Operationen.

Beweis:

- 1 ✓

## Beweis (Forts.):

- ② Sei  $n$  gerade und seien  $A, B$  zwei  $n \times n$  Matrizen. Wir wenden Strassen's Formeln an, führen aber die 7 Multiplikationen der  $\frac{n}{2} \times \frac{n}{2}$  Matrizen mit der klassischen Methode aus.  
Gesamtzahl arithmetischer Operationen:

$$7 \left( 2 \left( \frac{n}{2} \right)^3 - \left( \frac{n}{2} \right)^2 \right) + 18 \left( \frac{n}{2} \right)^2 = \frac{7}{4}n^3 + \frac{11}{4}n^2$$

Dies ist besser als die klassische Methode, falls:

$$\frac{7}{4}n^3 + \frac{11}{4}n^2 < 2n^3 - n^2 \Leftrightarrow \frac{15}{4}n^2 < \frac{1}{4}n^3 \Leftrightarrow 15 < n \Leftrightarrow n \geq 16$$

Sei  $n$  ungerade. Multiplizieren wir  $A \cdot B$  mit der klassischen Methode, so auch  $A_{11} \cdot B_{11}$ . Also brauchen wir durch Anwendung von Strassen's Formeln auf die  $(n-1) \times (n-1)$  Matrizen ( $n-1$  gerade) weniger Operationen, wenn  $n-1 \geq 16$  ist.

## Beweis (Forts.):

Sei  $M'(n)$  die Anzahl arithmetischer Operationen in MATMULT.  
Dann ist:

$$M'(n) = \begin{cases} 2n^3 - n^2 & \text{falls } n < 16 \\ 7M'(\frac{n}{2}) + \frac{18}{4}n^2 & \text{falls } n \geq 16 \text{ gerade} \\ 7M'(\frac{n-1}{2}) + \frac{42}{4}n^2 - 17n + \frac{15}{2} & \text{falls } n \geq 16 \text{ ungerade} \end{cases}$$

Wir definieren für  $x \in \mathbb{R}^+$ :

$$\bar{M}(x) = \begin{cases} 2x^3 - x^2 & \text{falls } x < 32 \\ 7\bar{M}(\frac{x}{2}) + \frac{42}{4}x^2 & \text{falls } x \geq 32 \end{cases}$$

Dann gilt:

$$\bar{M}(n) \geq M'(n) \text{ für alle } n \in \mathbb{N}.$$

## Beweis (Forts.):

Es ist

$$\bar{M}(x) = \sum_{i=0}^{k-1} \left[ 7^i \cdot \frac{42}{4} \left( \frac{x}{2^i} \right)^2 \right] + 7^k \left( 2 \cdot \left( \frac{x}{2^k} \right)^3 - \left( \frac{x}{2^k} \right)^2 \right)$$

für  $x \geq 32$ , wobei  $k := \min \{ l \mid \frac{x}{2^l} < 32 \}$ .

Mit  $k = \lfloor \log_2 x \rfloor - 4 =: \log_2 x - t$  erhalten wir  $t \in [4, 5[$  und

$$\bar{M}(x) \leq 7^{\log_2 x} \left( 14 \left( \frac{4}{7} \right)^t + 2 \left( \frac{8}{7} \right)^t \right) \leq 4,91 \cdot 7^{\log_2 x} \text{ für } x \geq 32.$$

Für  $x < 32$  direkt nachrechnen. □





Volker Strassen:

*Gaussian elimination is not optimal*

Numer. Math. **13**, pp. 354–356 (1969)



Victor Ya. Pan:

*New fast algorithms for matrix operations*

SIAM J. Comput. **9**(2), pp. 321–342 (1980)



Don Coppersmith, Shmuel Winograd:

*Matrix multiplication via arithmetic progressions*

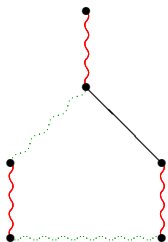
J. Symbolic Computation **9**(3), pp. 251–280 (1990)

# Kapitel VI Matchings in Graphen

## 1. Grundlagen

### Definition 126

Sei  $G = (V, E)$  ein ungerichteter, schlichter Graph. Ein **Matching**  $M$  in  $G$  ist eine Teilmenge von  $E$ , so dass keine zwei Kanten aus  $M$  einen Endpunkt gemeinsam haben.



Matching:

— Variante 1

— Variante 2

## Definition 127

- ① Ein Matching  $M$  in  $G = (V, E)$  heißt **perfekt** (oder **vollkommen**), falls

$$|M| = \frac{|V|}{2}$$

- ② Ein Matching  $M$  heißt „**Matching maximaler Kardinalität**“ (engl. **maximum matching**) in  $G$ , falls es in  $G$  kein Matching  $M'$  mit  $|M'| > |M|$  gibt (durchgehend geringelt/**rot** im Beispiel)
- ③ Ein Matching  $M$  heißt **maximal** in  $G$ , falls es bezüglich „ $\subseteq$ “ maximal ist (engl. **maximal matching**) (gepunktet geringelt/**grün** im Beispiel)

## Definition 128

Sei  $G = (V, E)$  ein Graph. Wenn  $V$  in zwei nichtleere Teilmengen  $V_1$  und  $V_2$  partitioniert werden kann ( $V_1 \cup V_2 = V$ ,  $V_1 \cap V_2 = \emptyset$ ), so dass  $E \subseteq V_1 \times V_2$  ist, dann heißt  $G$  **bipartit** ( $G = (V_1, V_2, E)$ ).