Computational Physics

Übungsblatt 2

Miriam Simm
miriam.simm@tu-dortmund.de

Katrin Bolsmann
katrin.bolsmann@tu-dortmund.de

Mario Alex Hollberg
mario-alex.hollberg@tu-dortmund.de

Abgabe: 8. Mai 2020

Aufgabe 1: Singulärwertzerlegung

a)

Das Bild wird mittels der Datei `service.cpp` eingelesen und zunächst durch Transposition gedreht, sodass der Mandrill einem in die Augen sieht (siehe Abbildung 1). Im nächsten Schritt wird mithilfe von `Eigen::BDCSVD` eine Singulärwertzerlegung durchgeführt.

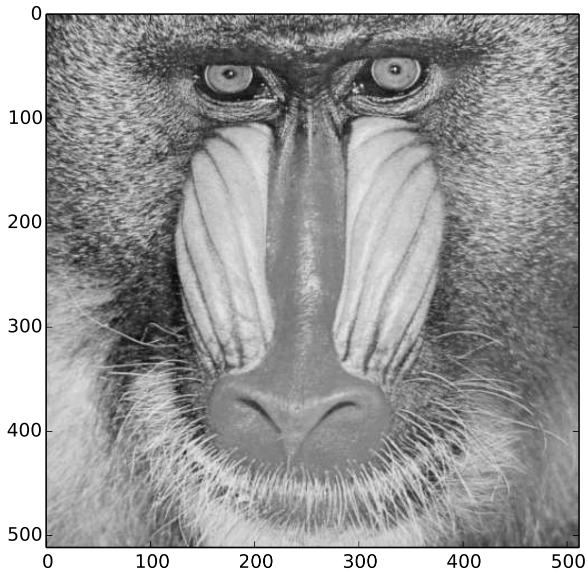
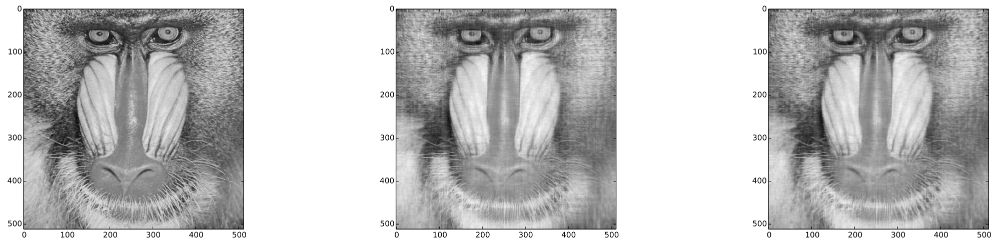


Abbildung 1: Transponierte Bild-Daten

b)

In diesem Aufgabenteil wird eine Rang-k-Approximation gemäß der Vorlesung, für k-Werte von 10, 20 und 50, durchgeführt und im folgenden als Heatmap mit Graustufen-Farbskala dargestellt.



k=10 k=20 k=50

Abbildung 2: Singulärwertzerlegte Bild-Daten nach Rang-k-Approximation

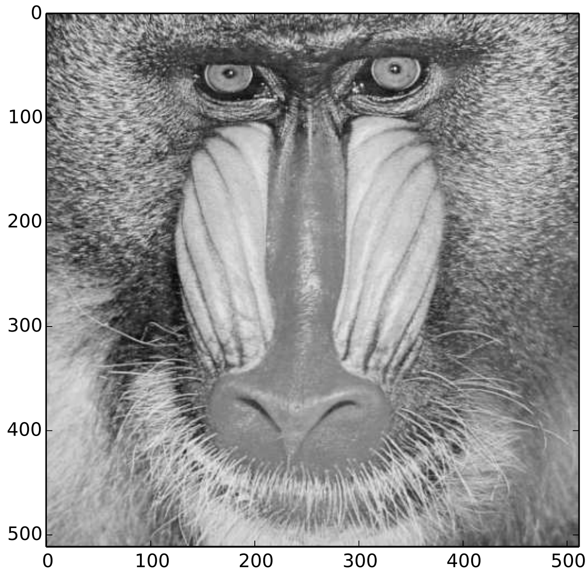


Abbildung 3: Singulärwertzerlegte Bild-Daten nach Rang-k=10-Approximation

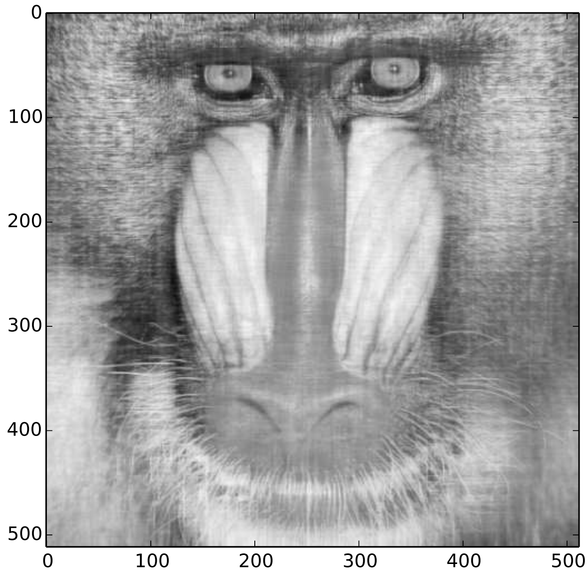


Abbildung 4: Singulärwertzerlegte Bild-Daten nach Rang-k=20-Approximation

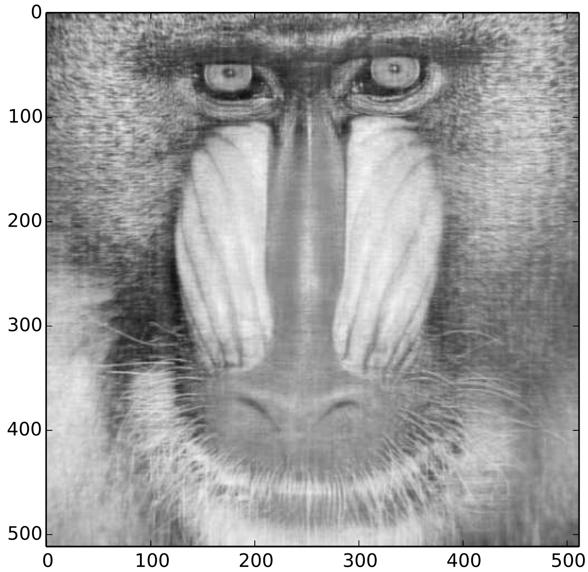


Abbildung 5: Singulärwertzerlegte Bild-Daten nach Rang-k=50-Approximation

Aufgabe 2: Profiling zur Untersuchung eines Algorithmus

In dieser Aufgabe soll ein zufälliges Lineares Gleichungssystem, aus einer quadratischen Matrix M der Dimension N und N-dimensionalen Vektoren x und b

$$Mx = b$$

mit einer LU-Zerlegung gelöst werden. Dabei soll die Laufzeit der einzelnen Arbeitsschritte

1) Erstellen einer zufälligen N-dimensionalen Matrix M
2) Durchführung der LU-Zerlegung
3) Lösen des Gleichungssystems mit LU-Zerlegung

für verschiedene Dimensionen N verglichen werden.

b) Plotten der Laufzeiten

In diesem Aufgabenteil wurde N logarithmisch vergrößert und die Laufzeiten gemessen. Diese sind in Plot 6 doppellogarithmisch aufgetragen. Zusätzlich wurde auch die Gesamtlaufzeit des Algorithmus berechnet.

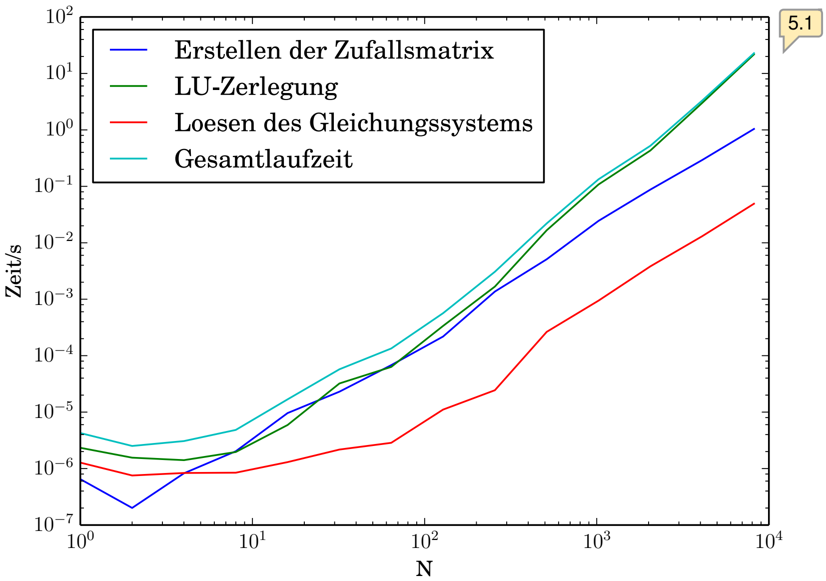


Abbildung 6: Aufgabe 2b: Laufzeiten der einzelnen Schritte doppellogarithmisch aufgetragen.

c) Deutung der Ergebnisse

Gesamtlaufzeit: Abschätzung der Laufzeit für N = 1.000.000

Offensichtlich besteht ein annähernd linearer Zusammenhang zwischen den logarithmierten Werten von N und t_{ges}. Es gilt also

$$\ln(t_{\text{ges}}) = a \ln(N) + b \qquad .$$

Zur Berechnung der Parameter a und b werden zwei beliebig gewählte Wertepaare eingesetzt und nach a und b umgeformt.

6.1

$$\begin{aligned} N &= 8192 \qquad t_{\text{ges}} \approx 24\text{s} \\ N &= 4096 \qquad t_{\text{ges}} \approx 3\text{s} \end{aligned}$$

$$\begin{aligned} \Rightarrow a &= \frac{\ln\left(\frac{3}{24}\right)}{\ln\left(\frac{1}{2}\right)} = 3 \\ b &= \ln(24) - a \ln(8192) \approx -23,85 \end{aligned}$$

Somit ergibt sich für ein Gleichungssystem mit einer 1Mio×1Mio-Matrix eine Laufzeit von

$$t_{\text{ges}} = \exp\left(3 \ln(10^6) - 23,85\right) \text{s} \approx 4,4 \cdot 10^7 \text{s} \qquad .$$

Optimierungspotential:

Werden die Laufzeiten nicht logarithmisch aufgetragen, wird noch deutlicher, dass die LU-Zerlegung den größten Anteil der Gesamtlaufzeit ausmacht, wie in Abbildung 7 zu sehen ist. Somit würde es sich am meisten auszahlen die LU-Zelegung bezüglich der Laufzeit zu optimieren.

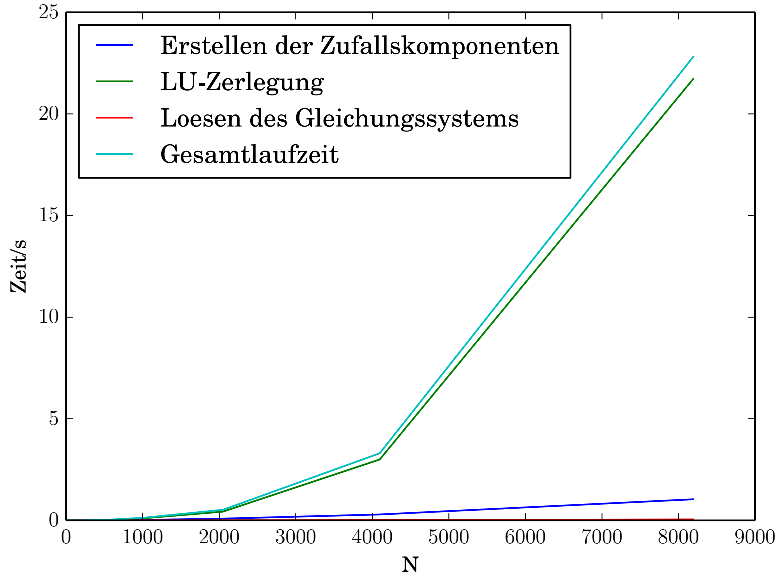


Abbildung 7: Aufgabe 2b: Laufzeiten der einzelnen Schritte.

d) Welche Faktoren schränken die Berechnung der Eigenwerte für große Matrizen weiter ein?

Für sehr große Matrizen ist die LU-Zerlegung anfällig gegenüber Rundungsfehlern und wird somit instabil. 7.1

Aufgabe 3: Profiling zum Vergleich von Algorithmen

Ziel der Aufgabe ist die Anwendung eines einfachen Profiling zur Untersuchung verschiedener Routinen der Bibliothek `Eigen`. Dazu wird, wie auch in Aufgabe 2, ein lineares Gleichungssystem $Mx = b$ mit einer Matrix M und einer rechten Seite b mit jeweils zufälligen Einträgen erstellt, wofür `Eigen::MatrixXd::Random` und `Eigen::VectorXd::Random` verwendet wird. Die Dimensionalität der Matrix sei N.

a) Zur Lösung des Gleichungssystems werden nun die folgenden Methoden implementiert und verglichen

A) Lösung durch Bestimmung und Anwendung der inversen Matrix M^{-1} mit `M.inverse()`

B) Lösung über eine LU-Zerlegung mit vollständiger Pivotisierung durch `M.fullPivLu().solve(b)`

C) Lösung über eine LU-Zerlegung mit teilweiser Pivotisierung durch `M.partialPivLu().solve(b)`

Zusätzlich wird überprüft ob das System invertierbar ist, wozu die Determinante verwendet wird.

b) Im Programm wird außerdem überprüft, ob alle Methoden A, B und C die gleichen Ergbnisse liefern. Das ist jedoch hier auch immer der Fall.

c) Die Ergebnisse der Laufzeitmessung der verschiedenen Vorgehensweisen sind in Abbildung 8 dargestellt. Verwendet wurde ein linear anwachsendes N mit Maximalwert N = 1000.

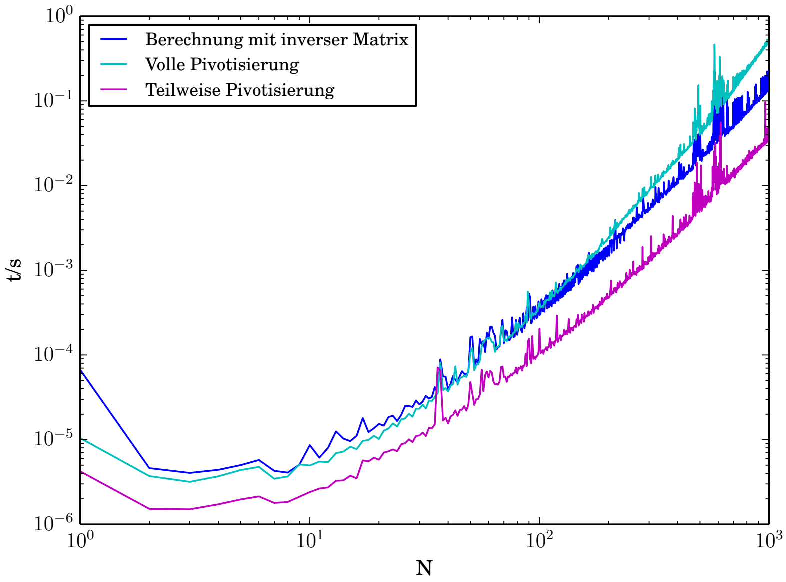


Abbildung 8: Aufgabe 3: Vergleich der Laufzeit der verschiedenen Algorithmen zur Lösung des linearen Gleichungssystems. Verwendet wird eine doppellogarithmische Skalierung.

d) Algorithmenvergleich: Am effizientesten ist in diesem Bereich von N die Berechnung über LU-Zerlegung mit teilweiser Pivotisierung. Weniger effizient ist die Berechnung über LU-Zerlegung mit voller Pivotisierung und über die Inverse, wobei der Zeitaufwand für beide Methoden sich wenig unterscheidet. Die Berechnung mit der inversen Matrix ist jedoch sinnvoll, falls diese noch weiter benötigt wird. Beachtet werden muss auch, dass die teilweise LU-Zerlegung nur für quadratische invertierbare Matrizen möglich ist. Mit voller Pivotisierung funktioniert die LU-Zerlegung für alle Matrizen und hat zudem den Vorteil, dass sie genauer ist.

Index der Kommentare

5.1 Plottet am besten auch Marker, damit man einfacher abschätzen kann, wie feinmaschig die Daten aufgenommen wurden.
Wenn ihr das grid aktiviert (und vielleicht sogar die Basis auf 2 ändert), kann man sehr schön die Exponenten durch Ablesen abschätzen.

6.1 Ich bin überrascht, dass ihr mit den letzten beiden Zeiten so gut auf den Exponenten 3 kommt.
Normalerweise sollte man für eine lineare Approximation entweder mehrere Werte fitten oder bei einer Sekante zumindest nicht direkt benachbarte Punkte hernehmen. Da es hier aber um den Trend für große N geht, ist eure Methodik in meinen Augen aber gerechtfertigt.

7.1 Und der Arbeitsspeicher wir knapp