
Übung 1 *Schurkomplement*
Sei $\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{C}$. Gegeben sei eine Blockpartitionierung einer Matrix $A \in \mathbb{K}^{n\times n}$ der Form

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix},$$

wobei die Teilmatrix $A_{11} \in \mathbb{K}^{p\times p}$ $(1 \leq p \leq n)$ invertierbar sei. Für die anderen Teilmatrizen gilt $A_{12} \in \mathbb{K}^{p\times(n-p)}$, $A_{21} \in \mathbb{K}^{(n-p)\times p}$ und $A_{22} \in \mathbb{K}^{(n-p)\times(n-p)}$. Die Block LR Zerlegung von A lässt sich dann darstellen als

$$A = \begin{pmatrix} Id & 0 \\ A_{21}A_{11}^{-1} & Id \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ 0 & S \end{pmatrix}$$

mit entsprechend dimensionierten Einheitsmatrizen Id. Die Teilmatrix $S \in \mathbb{K}^{(n-p)\times(n-p)}$ wird Schurkomplement von A_{11} in A genannt.

a) Zeigen Sie

$$S = A_{22} - A_{21}A_{11}^{-1}A_{12}.$$

b) Zeigen Sie: Wenn A hermitesch positiv definit ist, dann sind auch A_{11} und S hermitesch positiv definit.

(3 Punkte)

Übung 2 *LR-Zerlegung tridiagonaler Matrizen*
Gegeben sei eine Tridiagonalmatrix

$$A = \begin{pmatrix} a_1 & b_1 & & & 0 \\ c_2 & a_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & b_{n-1} \\ 0 & & & c_n & a_n \end{pmatrix}$$

mit

$$\begin{aligned} &|a_1| > |b_1| > 0 \\ &|a_j| \geq |b_j| + |c_j| > 0, \quad b_j, c_j \neq 0, \quad j \in \{2, ..., n-1\} \\ &|a_n| \geq |c_n| > 0\,. \end{aligned}$$

Zeigen Sie: Der Algorithmus

$$\begin{aligned} &r_1 := a_1 \\ &l_j := c_j / r_{j-1} \qquad j \in \{2, ..., n\} \\ &r_j := a_j - l_j b_{j-1} \quad j \in \{2, ..., n\} \end{aligned}$$

ist durchführbar (d.h. $r_1, \cdots, r_n \neq 0$) und liefert die LR-Zerlegung

$$A = \begin{pmatrix} 1 & & & 0 \\ l_2 & 1 & & \\ & \ddots & \ddots & \\ 0 & & l_n & 1 \end{pmatrix} \begin{pmatrix} r_1 & b_1 & & 0 \\ & r_2 & \ddots & \\ & & \ddots & b_{n-1} \\ 0 & & & r_n \end{pmatrix}$$

Zeigen Sie, dass $\det(A) =\neq 0$ gilt (woraus die Invertierbarkeit von A folgt).

(5 Punkte)

Übung 3 *LR Zerlegung im Besonderen*
Sei $A \in \mathbb{R}^{n \times n}$ gegeben durch

$$a_{ij} = \begin{cases} +1 & \text{wenn } i = j \text{ oder } j = n \\ -1 & \text{wenn } i > j \\ 0 & \text{sonst.} \end{cases}$$

a) Begründen Sie, dass die LR Zerlegung ohne Pivotisierung von A die Eigenschaften

$$|l_{ij}| \leq 1 \qquad \text{und} \qquad r_{nn} = 2^{n-1}$$

erfüllt.

b) Zeigen Sie, dass eine LR Zerlegung von A mit Zeilen- und Spaltenvertauschung existiert mit:

$$|r_{nn}| = 2 = \max_{i,j=1..n} \{r_{ij}\}.$$

Tipp: Erst mal für $n = 4$ betrachten um die beste Verstauschungsstrategie zu finden.

(5 Punkte)

Übung 4 *Cholesky Zerlegung (Parktische Übung)*
In dieser Aufgabe sollen Sie die Cholesky Zerlegung für symmetrisch positiv definite Matrizen implementieren und die Anzahl der benötigten Rechenoperationen untersuchen.

a) Schreiben Sie eine neue Headerdatei `cholesky.hh` in der Sie die template Funktion

```
template<typename NUMBER>
void cholesky(hdnum::DenseMatrix<NUMBER>& A){
  ...
}
```

zur Berechnung der Cholesky Zerlegung $A = \tilde{L}\tilde{L}^T$ einer symmetrisch positiv definiten Matrix $A \in \mathbb{R}^n$ implementieren. Dabei soll die untere Dreicksmatrix $\tilde{L}$ direkt in der Matrix A gespeichert werden. Beachten Sie:

- Die Werte oberhalb der Diagonalen werden durch den Aufruf der Funktion `cholesky(A)` nicht verändert und haben nichts mit $\tilde{L}$ zu tun.
- Für die Implentierung dürfen Sie folgenden Algorithmus verwenden:

for ($i = 1$; $i \leq n$, $i = i + 1$) **do**
  for ($j = 1$; $j \leq i$, $j = j + 1$) **do**
    $z = a_{ij}$
    for ($k = 1$; $k \leq j - 1$, $k = k + 1$) **do**
      $z = z - a_{ik}\, a_{jk}$
    end for
    if $i > j$ **then**
      $a_{ij} = \frac{z}{a_{jj}}$
    else if $z > 0$ **then**
      $a_{ii} = \sqrt{z}$
    else
      Error: The matrix A is not symmetric positive definite!
    end if
  end for
end for

Testen Sie Ihre Cholesky Zerlegung an dem auf der Vorlesungshomepage zur Verfügung gestellten Beispiel `cholesky.cc`. Damit das Programm kompiliert benötigen Sie außerdem die Headerdatei `opcounter.hh`, die ebenfalls auf der Vorlesungshomepage zur Verfügung gestellt wird.

b) In dieser Teilaufgabe sollen Sie die Anzahl der Rechenoperationen der Cholesky Zerlegung bestimmen. Verwenden Sie dafür die `OpCounter` Klasse. Folgendes Beispiel erläutert die Verwendung:

```
#include <iostream>
#include <cmath>
#include "opcounter.hh"

// This function just does some basic arithmetic
template <typename T>
void someOperations(T& a){
  T tmp(2.0);
  a = tmp*a+tmp;

  // Make sure to use sqrt like this! If you use a = std::sqrt(a)
  // instead it won't work for the OpCounter class.
  using std::sqrt;
  a = sqrt(a);
}

int main(){
  // Version using double
  typedef double Number;
  Number a(5.0);
  someOperations(a);
  std::cout << "Double version: a=" << a << std::endl;

  // Version using OpCounter leads to the same result
  typedef oc::OpCounter<double> OC_Number;
  OC_Number b(5.0);
  someOperations(b);
  std::cout << "OpCounter version: b=" << b << std::endl;

  // If you want to print the total amount of operations it doesn't
  // matter what OpCounter class instance you use. Just use any
  // OpCounter variable (in this case c) for reporting.
  OC_Number c;
  std::cout << "Operation count: " << c.totalOperationCount() << std::endl;

  // Reset counter
  c.reset();
  std::cout << "Operation count: " << c.totalOperationCount() << std::endl;

  return 0;
}
```

Passen Sie Ihre Implementierung der Cholesky Zerlegung an und Berechnen Sie die Anzahl der benötigten Rechenoperationen für die gegebene Beispielmatrix. Wie viele Operationen werden benötigt?

c) Stellten Sie für $n = 10, \ldots, 100$ eine symmetrisch positiv definite Matrix $B \in \mathbb{R}^{n \times n}$ auf, indem Sie `hdnum::spd(B)` verwenden. Bestimmen Sie die Anzahl der Rechenoperationen der Cholesky Zerlegung (`oc_cholesky`) und der LR Zerlegung (`oc_lr`) von B. Verwenden Sie dafür die LR Zerlegung aus der hdnum:

```
hdnum::Vector<std::size_t> perm(n);
hdnum::lr(B, perm);
```

Geben Sie n und `double(oc_lr)/oc_cholesky` auf dem Termnial aus und visualisieren Sie das Ergebnis mit gnuplot. Entspricht das Ergebnis Ihren Erwartungen?

(7 Punkte)