Otto-von-Guericke-University Magdeburg
Max Planck Institute for Dynamics of Complex Technical Systems
Computational Methods in Systems and Control Theory

Dr. Jens Saak, Dipl.-Math. Martin Köhler
Website: http://www.mpi-magdeburg.mpg.de/csc/teaching/16ws/sc1/

Winterterm 2016/2017

Scientific Computing 1
10th Homework

Handout: 01/04/2017 **Return:** 01/13/2017

Exercise 1: (3 Points)

Let $V, W \in \mathbb{R}^{n\times k}$, $k < n$, be two matrices of rank k. Show that

$$A = VW^T \quad \in \mathbb{R}^{n\times n}$$

has exactly rank k.

Exercise 2: (4 Points)

The LU decomposition is the main technique to solve moderate size linear systems and has an asymptotic flop count of $\frac{2}{3}n^3$ flops. Beside the classical scalar implementation nowadays block implementations with a blocking parameter r are used to exploit modern computer architectures.

a.) Show that the asymptotic flop count for the block LU decomposition is invariant under the chosen blocking parameter r.

b.) Compare the exact flop count (including the lower order terms) of the unblocked and the block implementation.

Exercise 3: (14 Points)

The outer product Gaussian elimination defines one possible way to compute an LU decomposition of a matrix $A \in \mathbb{R}^{m\times n}$.

a.) Implement this algorithm **without** blocking techniques as a C function with the following signature:

```
int LU2(int m, int n, double * A, int lda);
```

The input `A` with leading dimension `lda` should be overwritten by its LU decomposition as shown in the lecture. The integer return value should be 0 if the matrix was decomposed successfully or a non zero value otherwise.

Hint: The appearing rank-1 updates can be performed using your `gemm` implementation from the previous exercises.

b.) Implement this algorithm **with** blocking techniques as a C function with the following signature:

```
int LU3(int m, int n, double * A, int lda, int r);
```

The input A with leading dimension `lda` should be overwritten by its LU decomposition as shown in the lecture. The blocking parameter `r` should be freely selectable by the caller. The integer return value should be 0 if the matrix was decomposed successfully or a non zero value otherwise. The the LU decomposition function from a.) to decompose the small panels.

Hint: The appearing rank-r updates can be performed using your `gemm` implementation from the previous exercises.

c.) Write a solver function that takes a square LU decomposed matrix from a.) or b.), $L \in \mathbb{R}^{n\times n}$ and $U \in \mathbb{R}^{n\times n}$, and a right hand side $b \in \mathbb{R}^n$ as inputs and overwrites b with the solution of $LUx = b$. Use the BLAS function `DTRSV` or `DTRSM` for this purpose. The function header should be

```
void LU_solve(int n, double *LU, int ldlu, double *b);
```

where `ldlu` is the leading dimension of the matrix containing the LU decomposition.

d.) Solve the following linear system to check your code:

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} \\ \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} \end{pmatrix} x = \begin{pmatrix} \frac{49}{20} \\ \frac{223}{140} \\ \frac{341}{280} \\ \frac{2509}{2520} \\ \frac{2131}{2520} \\ \frac{20417}{27720} \end{pmatrix}$$

What can you recognize? Is the solution sufficiently accurate when you think about the possibilities of double precision floating point numbers?

Hint: A skeleton code with the data for d.) is provided on the lecture's website.

Exercise 4: **(4 Points)**

The Jacobi method is a classical splitting technique to solve linear systems iteratively. It splits a matrix $A \in \mathbb{R}^{n\times n}$ into the diagonal $D = \operatorname{diag}\{a_{11}, a_{22}, \dots, a_{nn}\}$ and off-diagonal elements and repeatedly solves the equation

$$x_{i+1} = M_J x_i + D^{-1} b$$

where the iteration matrix is

$$M_J = D^{-1}(D - A).$$

Show that the Jacobi method converges to the solution $x = A^{-1}b$ if the matrix A is strictly diagonal-dominant, i.e.,

$$\sum_{j=1,\, j\neq i}^{n} |a_{ij}| < |a_{ii}|, \qquad \forall i = 1, \dots, n.$$

Hint: Use Theorem 6.25.

Overall Points: 25