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/2802004/sc1/

Winterterm 2014/2015

Scientific Computing 1
10th Homework

Handout: 01/14/2015 **Return:** 01/21/2015

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. It has an asymptotic flop count of $\frac{2}{3}n^3$ flops. Furthermore, a block implementation is necessary 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: (10 Points)

The outer product Gaussian elimination defines one possible way to compute an LU decomposition of a matrix.

a.) Implement this algorithm as a C function with the following header:

```
void LU(struct my_matrix_st A);
```

The input `A` should be overwritten by its LU decomposition as shown in the lecture. Use the `scale_col` and the `r1_update` functions from the previous Homework to perform the necessary operations.

b.) Write a solver function that takes the LU decomposed matrix from a.) 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/DTRSM` for this purpose. The function header should be

```
void LU_solve(struct my_matrix_st LU, double *b);
```

c.) 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 a reference implementation of `scale_col` and `r1_update` is available on the web page.

Exercise 4: **(8 Points)**

The accuracy of the solution of a linear system can be improved using iterative refinement. Implement this procedure as a C function

```
int refine(struct my_matrix_st A, double *b, int maxiter, double tol);
```

where the right hand side `b` is overwritten with the solution. The `maxiter` parameter defines the maximum number of iterations and the `tol` argument stops the iteration when

$$||b - Ax_i||_2 < \texttt{tol}.$$

The return value of the function is the number of actually performed iteration steps.

Solve the linear system from Exercise 3 with this function and compare the results.

Hints:

- Use `DGETRF` and `DGETRS` from LAPACK to solve the linear system.
- Take care that `DGETRF` overwrites the input matrix.
- The matrix vector product $\alpha Ax + \beta b \to b$ is realized by the `DGEMV` BLAS routine.
- The 2-norm of a vector is computed by the BLAS function `DNRM2`.

Overall Points: 25