Winterterm 2012/2013

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

Dr. Jens Saak, Dipl.-Math. Martin Köhler
Website: http://www.mpi-magdeburg.mpg.de/mpcsc/lehre/2012_WS_SC/

Scientific Computing 1
10th Homework

Handout: 12/06/2012 **Return:** 12/13/2012

Exercise 1: (5 Points)

Prove that

$$\kappa_2(A) = 1$$

holds for all unitary matrices $A \in \mathbb{C}^{n\times n}$. In which way does this influence the design of numerical algorithms?

Exercise 2: (8 Points)

Consider the linear system $Ax = b$ with

$$A = \begin{bmatrix} 2 & -1 & 1 \\ -1 & 10^{-10} & 10^{-10} \\ 1 & 10^{-10} & 10^{-10} \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 2(1+10^{-10}) \\ -10^{-10} \\ 10^{-10} \end{bmatrix}.$$

a.) Compute the solution x of the linear system.

b.) Show that $\kappa_\infty(A) = 2 \cdot 10^{10}$ holds.

c.) Consider a disturbed linear system $(A + \Delta A)\hat{x} = b$ with $|\Delta A| \leq 10^{-8}|A|$. Prove that the solution $\hat{x}$ of the disturbed system fulfills
$$|x - \hat{x}| \leq 10^{-7}\,|x|.$$

d.) Let $D = \operatorname{diag}(10^{-5}, 10^5, 10^5)$. Check that $\kappa_\infty(DAD) \leq 5$ is true.

Exercise 3: (12 Points)

The solution of triangular linear systems $Lx = b$ is a key operation for many high level linear algebra operations. The matrix $L \in \mathbb{R}^{n\times n}$ is stored in the `my_matrix_st` structure, which was presented in the lecture. A skeleton code which provides a read, a print, a random matrix function and a Makefile is available from the lecture website. There also exist some example data sets.
Implement the following functions inside the `main.c` file:

a.) The function

```
void L_solvev(struct my_matrix_st *L, double *b)
```

which takes a lower triangular matrix $L \in \mathbb{R}^{n\times n}$ and a right hand side $b \in \mathbb{R}^n$ as inputs and overwrites b with the solution of $Lx = b$. Implement the naive forward elimination scheme.

b.) The function

```
void L_solvev_trsv(struct my_matrix_st *L, double *b)
```

which overwrites b with the solution of $Lx = b$. Use the BLAS level 2 routine `DTRSV` instead of implementing the forward elimination scheme.

c.) The function

```
void L_solvem(struct my_matrix_st *L, struct my_matrix_st *B)
```

which solves $LX = B$ for multiple right hand sides, i.e. $B \in \mathbb{R}^{n \times p}$, using the triangular solve from a.

d.) The function

```
void L_solvem_trsv(struct my_matrix_st *L, struct my_matrix_st *B)
```

which solves $LX = B$ for multiple right hand sides, i.e. $B \in \mathbb{R}^{n \times p}$, using the triangular solve from b.

e.) The function

```
void L_solvem_trsm(struct my_matrix_st *L, struct my_matrix_st *B)
```

which solves $LX = B$ for multiple right hand sides, i.e. $B \in \mathbb{R}^{n \times p}$, using the BLAS level 3 triangular solve `DTRSM`.

Generate a random matrix $B \in \mathbb{R}^{n \times 100}$ and measure the time for one solve $LX = B$ where $L \in \mathbb{R}^{n \times n}$ is one of the demo matrices. What can you recognize?

Hints:

- In order to get fast BLAS level 3 subroutines install the `libopenblas-dev` package inside the virtual machine using:

  ```
  sudo apt-get install libopenblas-dev
  ```

  The neccessary password is "user".

- The function headers for the BLAS subroutines `DTRSV` and `DTRSM` are prepared in the skeleton code.

- The runtime of a piece of code can be measured using the `wtime` function from the skeleton code:

  ```
  double tic, toc;
  tic = wtime();
  ... Your Code...
  toc = wtime();
  printf("The code took %lg seconds\n", toc-tic);
  ```

- The skelton code provides a Makefile which does all the compilation steps.

Overall Points: 25