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

Winterterm 2018/2019

Website: http://www.mpi-magdeburg.mpg.de/csc/teaching/18ws/sc1/

Scientific Computing 1
10th Homework

Handout: 12/13/2018 **Return:** 12/21/2018

Exercise 1: (2 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: (6 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.) Let $D = \operatorname{diag}(10^{-5}, 10^5, 10^5)$. Check that $\kappa_\infty(DAD) \leq 5$ is true.

Exercise 3: (5 Points)

LAPACK is the key software package for high level numerical linear algebra. It provides Fortran subroutines to solve linear systems, eigenvalue problems and many more. The solution of linear systems is mostly done using the computational subroutines `DGETRF` and `DGETRS` or the driver `DGESV`.
Use the skeleton code provided on the lecture's website and solve the linear systems

$$Ax = b_1 \quad \text{and} \quad Ay = b_2$$

using a proper choice of the above LAPACK routines. The matrix $A \in \mathbb{R}^{487\times 487}$ and the right hand sides $b_1 \in \mathbb{R}^{487}$ and $b_2 \in \mathbb{R}^{487}$ are provided in the skeleton archive. Once, you have computed x and y write them to a file called `solution.dat` with the following style:

```
x_1 y_1
x_2 y_2
...
...
x_487 y_487
```

The `solution.dat` file is used together with `gnuplot` and the `display.plot` script at the end of the program to present the solution. What can you see?

Hints:

- The skeleton code already reads the matrix A and the right hand sides b_1 and b_2.
- If you do not use the virtual machine of the lecture please install `gnuplot` on your system before.
- If the plot does not work, you can use MATLAB as well. Therefore use

  ```
  P = load('./solution.dat');
  plot(P(:,1), P(:,2),'*')
  ```

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

Exercise 4: (4 Points)

Let $A \in \mathbb{R}^{n\times n}$ be a matrix and $A_{ik} \in \mathbb{R}^{(n-1)\times(n-1)}$ be the submatrix which is created by removing the i-th row and the k-th column of A. We can compute the determinant of the matrix by the recursion formula

$$\det(A) = \sum_{i=1}^{n} (-1)^{i+k} a_{ik} \det(A_{ik}).$$

Derive a recursion formula for the number of necessary floating point operations to compute $\det(A)$. Approximate the computation time for the determinant of a 100×100 matrix on a current CPU. Assume that the CPU has a peak performance of 300 GFlops/s ($300 \cdot 10^9$ floating point operations per second).

Exercise 5: (6 Points)

In many cases it is not necessary to compile and link the whole LAPACK library to a program, e.g., if one only needs a single driver routine from it.

a.) Search on `http://www.netlib.org/lapack/double/` for a driver which computes the eigenvalues and eigenvectors of a general matrix and download it together with its dependencies.

b.) Write a `Makefile` which creates a small static library called `liblapack_pocket.a` containing the eigenproblem solver and its dependencies.

c.) Write a small C program which uses this library to compute all eigenvalues and eigenvectors of

$$A = \begin{pmatrix} -3 & 8 & -2 & 1 \\ 6 & -4 & 1 & 5 \\ 2 & 5 & -8 & 8 \\ 1 & 5 & -8 & -7 \end{pmatrix}$$

Hint: Subroutines from BLAS are not automatically included as dependencies in the download. That means either BLAS needs to be linked to the program, or the corresponding files need to be downloaded separately.

Overall Points: 23