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
11th Homework

Handout: 01/15/2017 **Return:** 01/20/2017

Exercise 1: (12 Points)

One efficient storage scheme for sparse matrices is the Compressed Sparse Row (CSR) format introduced in the lecture. In this scheme the matrix is stored using two integer arrays, `rowptr` and `colptr`, and a `values` array containing the entry values. This storage scheme can be realized using a C structure:

```
struct sparse_matrix_st {
       int cols;
       int rows;
       int nnz;
       int *rowptr;
       int *colptr;
       double *values;
};
```

The skeleton implementation available from the lecture homepage contains a function to read the matrix from a file into CSR format called `sparse_matrix_read`. Additionally the `sparse_matrix_print` function prints a sparse matrix to the screen.

Based on the Compressed Row Storage implement the following functions:

a.) The sparse matrix vector product

$$y = Ax$$

as a C function:

```
void sparse_mvp(struct sparse_matrix_st *A, double *x, double *y);
```

b.) The Conjugate Gradient Algorithm to solve $Ax = b$ as a C function

```
int cg(struct sparse_matrix_st *A, double *x, double * b, int maxit,
                                   double tol);
```

The return value should be used to indicate if the algorithm has converged or not. Use the matrix vector product implementation from a.) and BLAS for all other linear algebra operations. The parameter `maxit` gives the maximum number of iterations allowed and the parameter `tol` stops the iteration if

$$||r_i||_2 < tol \cdot ||b||_2$$

is fulfilled. The parameter `x` is used as x_0 on input and contains the solution x_* on output.

c.) The Preconditioned Conjugate Gradient Algorithm with diagonal preconditioning as a C function:

```
int pcg(struct sparse_matrix_st *A, double *x, double * b, int maxit,
                                   double tol);
```

The parameters have the same meanings as in b.). Think about an efficient implementation of the preconditioner.

d.) Demonstrate both functions in a main program. Use $x_0 = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$ and $b = A \cdot \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$ to check if the algorithm works properly.

A skeleton code with some symmetric positive definite matrices is available on the website.

Exercise 2: (4 Points)

Consider a sparse matrix $A \in \mathbb{R}^{1\,000\,000 \times 1\,000\,000}$ with $4\,996\,000$ non zeros elements stored in double precision. We compute the sparse matrix vector $y = Ax$ on a CPU with a peak double precision performance of 42 GFlop/s (Intel® Core-2 Quad @ 2.66 GHz).

a.) We measured that one matrix-vector product takes $0.0143s$. Compute the flops per second for the computation of one matrix vector product with this matrix. Calculate the efficiency of the matrix vector product (the ratio between the achieved performance and the peak performance).

b.) Compute the memory transfer rate for the matrix from the flop rate. Does the result explain the efficiency from a.)?

Exercise 3: (3 Points)

Consider the following matrix vector product

$$y = \underbrace{\left(A^T Z Z^T + Z Z^T A - Z Z^T B B^T Z Z^T + C^T C\right)}_{\mathfrak{A}} x$$

with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times q}$, $C \in \mathbb{R}^{r \times n}$ and $Z \in \mathbb{R}^{n \times p}$. Assume that $p \ll n$, $q \ll n$ and $r \ll n$ holds.

a.) How many floating point operations are necessary to compute y naively?

b.) Rearrange the evaluation such that it takes less floating point operations.
Hint: It is necessary to introduce some extra variables.

Exercise 4: (2 Points)

Consider the full-rank matrices $A \in \mathbb{R}^{n \times n}$, $V_m \in \mathbb{R}^{n \times m}$ and $W_m \in \mathbb{R}^{n \times m}$.

a.) Prove that

$$P := I - A V_m (W_m^H A V_m)^{-1} W_m^H$$

defines a projection.

b.) Show that

$$P A V_m = 0$$

holds.

Exercise 5: **(6 Points)**

Consider the following sparse matrix $A \in \mathbb{R}^{8\times 8}$:

$$A := \begin{bmatrix} 1 & * & * & & & * & & * \\ * & 2 & & * & * & * & & \\ * & & 3 & * & & & & * \\ & * & * & 4 & & * & & \\ & * & & & 5 & & * & \\ * & * & & * & & 6 & * & \\ & & & & * & * & 7 & * \\ * & & * & & & & * & 8 \end{bmatrix},$$

where $*$ denote an arbitrary non-zero entries.

a.) Determine the sequence of elimination graphs $\mathcal{G}_i(A)$, $i = 1 \cdots 8$.

b.) Give the filled graph $\mathcal{G}^+(A)$.

Overall Points: 27