Homework Sheet 3

VU Numerical Algorithms, WiSe 2021

due date: 15.12.2021, 18:00

Programming Exercise

Note

Please stick carefully to the interface described in the second section below. This is necessary for us to evaluate your code, which also means that you will not get points for any piece of code that fails to run for interface related reasons. You can again add more parameters, so you do not need to recompute parts of the program and accelerate your computation. However, you should add default parameters, so the program still works with the predefined parameters given in the assignment description.

If you have not already done so, give the homework report guidelines (on moodle) a read. The aspects described therein form the basis for grading the formal aspects of the report.

Effects of Preconditioners on the Conjugate Gradients Method (16 points)

Implement the conjugate gradient method CG and experimentally investigate the effect of different preconditioners on its convergence. Please compare standard CG and preconditioned CG (PCG) for three given symmetric positive definite test problems in terms of the number of iterations and in terms of the runtime (including the time for computing and applying the preconditioner) until convergence. Show the convergence histories (norm of relative residual vs. iteration number) graphically.

- Implement standard CG *efficiently* (*you are not allowed to use the CG implementation available in Octave!*). In particular, store the sparse matrix in a sparse matrix format. (8 points)
- Use the following preconditioners:
  - Diagonal preconditioner (2 points)
  - Block diagonal preconditioner with diagonal subblocks of sizes $\{5, 25, 100\}$ (2 points)
  - Incomplete Cholesky factorization[1] (2 points)
  - Incomplete Cholesky factorization with threshold dropping: experiment with different thresholds and discuss the effect of the choice of threshold (2 points)

[1] Available in Octave via `ichol()`.

- Test matrices: Please use (at least) the following three test matrices from the "Matrix Market" (http://math.nist.gov/MatrixMarket/) for your experiments:
  - http://math.nist.gov/MatrixMarket/data/Harwell-Boeing/lanpro/nos5.html
  - http://math.nist.gov/MatrixMarket/data/Harwell-Boeing/lanpro/nos6.html
  - http://math.nist.gov/MatrixMarket/data/misc/cylshell/s3rmt3m3.html

  For importing MatrixMarket format (.mtx) files into Octave, the script `mmread.m` from https://math.nist.gov/MatrixMarket/mmio/matlab/mmiomatlab.html may prove helpful.

Formalities

1. Basics:
   - Please use Octave *version 4.4* or higher and indicate the Octave version in your report. Your submission will be evaluated.
   - Do not import additional packages and do not use global variables.
   - Pay attention to the interface definitions, i.e., use the specified terms. In/output parameters must be in the specified order.
   - Do not use any existing code which you did not write yourself! However, for solving linear systems in this assignment, feel free to use Octave routines.
   - You can define your own routines in order to write modular code but please stay consistent with the predefined interface.
   - When plotting the relative residuals, use a logarithmic scale for the y-axis.
2. Interface:
   a) Create a file *residual.m* of the following form:

   ```
   [r] = residual(A, x_hat, b)
   ```

   - Input: A is an $n \times n$ matrix, $\hat{x}$, b are vectors of size n.
   - Output: scalar r, with
   $$r := \frac{||A\hat{x} - b||_2}{||A||_2||\hat{x}||_2}.$$

   Remark: Use this routine to compute the *relative residual* of the approximate solutions obtained in the `cg` routines.

   b) Create a file `cg.m` of the following form:

   ```
   [x, iter, res_vec] = cg(A, b, tol, maxit, x0)
   ```

   - Input: $n \times n$ matrix A, $n \times 1$ vector b, scalar tol, integer $maxit$ and $n \times 1$ vector x_0.
   - Output: $n \times 1$ vector x, integer $iter$, $(iter + 1) \times 1$ vector res_vec

Remark: The parameter *tol* denotes the largest admissible relative residual r. The iteration is terminated either when the residual drops below *tol*, or when the number of iterations exceeds a specified amount *maxit*. The initial guess is supplied via the vector x_0.

The list of return values contains, apart from the solution vector x, the number *iter* of iterations actually performed and the residual history in res_vec, where $res\_vec(i)$ contains the relative residual after iteration $i-1$ and $res\_vec(1)$ contains the relative residual of the initial guess.

c) Create a file `p_cg.m` of the following form:

```
[x, iter, res_vec] = p_cg(A, b, tol, maxit, x0, M1, M2)
```

- Input: As with `cg.m`; additionally two $n \times n$ matrices M_1, M_2.
- Output: As with `cg.m`.

Remark: The matrices M_1, M_2 describe the preconditioner. If M_2 is empty (passed as `[]`), the preconditioner is $M = M_1$. This will be the case with the (block) diagonal preconditioners. If the preconditioner is available in a factorized form, as in the case of incomplete Cholesky factorization, M_1 and M_2 contain the factors and the preconditioner is given as $M = M_1 M_2$. In this case, be careful to compute $M^{-1}\boldsymbol{r}_{k+1}$ using the factorization and not by explicitly inverting (that would ruin the purpose of the algorithm).

d) Write a script `assignment3.m` to test your routines and plot your results.[2]

3. Submission:

- Upload a single zip archive with all your source code files and your report (as a single PDF file named *report.pdf* with all plots and discussions of results as well as your solution to the paper-and-pencil exercises) on the course page in Moodle.
- Name your archive `a<matriculation number>_<last name>.zip` (e.g. *a01234567_Mustername.zip*)
- Directories in the archive are not allowed.
- A complete submission should include the following files:
  a) Routines: `residual.m, cg.m, p_cg.m`, and self defined routines (optional)
  b) Script: `assignment3.m`
  c) Documentation: *report.pdf*

[2] In case you do not use Octave for plotting, use `assignment3.m` to export the data plotted in `.csv` files.