Uppsala University
Graduate School in Mathematics and Computing
Institute for Information Technology

Numerical Linear Algebra FMB and MN1
Fall 2007

Mandatory Assignment 3a:

Iterative solution methods and their rate of convergence

Exercise 1 (Implementation tasks) Implement in `Matlab` the following iterative schemes:

M1: the (pointwise) Jacobi method;

M2: the second order Chebyshev iteration, described in Appendix A;

M3: the unpreconditioned conjugate gradient (CG) method (see the lecture notes);

M4: the Jacobi (diagonally) preconditioned CG method (see the lecture notes).

For methods M3 and M4, you have to write your own `Matlab` code. For the numerical experiments you should use your implementation and not the available `Matlab` function `pcg`. The latter can be used to check the correctness of your implementation.

Exercise 2 (Generation of test data) Generate four test matrices which have different properties:

A: `A=matgen_disco(s,1);`
B: `B=matgen_disco(s,0.001);`
C: `C=matgen_anisot(s,1,1);`
D: `D=matgen_anisot(s,1,0.001);`

Here s determines the size of the problem, namely, the matrix size is $n = 2^s$.
Check the sparsity of the above matrices (`spy`). Compute the complete spectra of A, B, C, D (say, for $s = 3$) by using the `Matlab` function `eig`. Plot those and compare. Compute the condition numbers of these matrices. You are advised to repeat the experiment for a larger s to see how the condition number grows with the matrix size n.

Exercise 3 (Numerical experiments and method comparisons) Test the four methods M1, M2, M3, M4 on the matrices A, B, C, D.
Choose a right-side vector as `b=rand(n,1);` and compute the corresponding 'exact' solution as `sol_A=A\b`. Recall, that $n = 2^s$ is the size of A.
For methods M1, M3 and M4 use as a stopping criterion $\|\mathbf{r}_k\|/\|\mathbf{r}_0\|/ < 10^{-6}$. For methods M3 and M4 $\mathbf{r}_k$ should be the iteratively computed (not the true residual, computed as $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$) residual. For method M2, first predict the number of iterations in advance using formula (7), do that many iterations and compare the obtained error and residual reduction with the expected ones.

Plot the corresponding residual convergence curve for the four methods. For the CG methods plot both the iteratively computed residual and the true residual obtained as `r_true=b-A*x_k`. Perform a series of experiments for $s = 3, 4, 5, 6$ to see the the dependence of the methods behaviour when n is increased.

Exercise 4 (A theoretical task) Derive that the convergence of M4 is better than that of M3.
Here the intention is to try to show it yourself or to do a literature search to find a proper result to cite. A good source to check could be Anne Greenbaum, *Iterative methods for solving linear systems*, SIAM, 1997.

Instructions for performing the numerical tests

Download all the files from the course web-page. The main files to call are the following:

- `matgen_disco.m`
  The routine `matgen_disco.m` generates a finite element stiffness matrix for the Laplace equation

$$-\frac{\partial}{\partial x}\left(a\frac{\partial u}{\partial x}\right) - \frac{\partial}{\partial y}\left(a\frac{\partial u}{\partial y}\right) = f \tag{1}$$

  in $\Omega \equiv (0,1)^2$ where a is a constant and $a = \varepsilon \ll 1$ in a subset $\widetilde{\Omega} \subset \Omega$ and $a = 1$ elsewhere with $\widetilde{\Omega} \equiv \{1/4 \leq x \leq 3/4,\ 1/4 \leq y \leq 3/4\}$. The problem is then discretized using regular isosceles triangles and piece-wise linear basis functions. The mesh-size parameter h is equal to 2^{-s} for some integer s and is chosen always such that there will be mesh-lines along the edges of $\widetilde{\Omega}$

  The routine is called as `A=matgen_disco(s,a);`. When `a` is one, the generated matrix corresponds to $-\Delta u = f$.

  The above routine needs auxiliary files `disco_stiff.m`, `disco_rule.m`, `Xdir.m` and `Ydir.m`.

- `matgen_anisot.m`
  The routine `matgen_anisot.m` generates a finite difference (5-point) discretization of the anisotropic Laplacian

$$-\varepsilon_x u_{xx} - \varepsilon_y u_{yy}.$$

  The routine is called as `A=matgen_anisot(s,epsx,epsy);`, where again mesh-size parameter h is equal to 2^{-s}.

  If both `epsx` and `epsy` are equal to one, the generated matrix corresponds to $-\Delta u = f$ discretized with central differences.

- `Lanczos.m`
  In order to use the Chebyshev iterative method one needs to estimate the extremal eigenvalues of A. The routine `Lanczos` computes such *approximations*. Note, however that there is no guarantee that we will obtain a lower bound for λ_{min} and an upper bound for λ_{max}!

The routine is called as `[lanmin,lanmax]=Lanczos(A);` It performs internally a number of Lanczos steps until the following criteria are met:

$$|\lambda_1^{(k)} - \lambda_1^{(k-1)}| \leq \epsilon \text{ and } |\lambda_n^{(k)} - \lambda_n^{(k-1)}| \leq \epsilon$$

with a default value of $\epsilon = 0.01$.

Remark: You can estimate the extreme eigenvalues using some other technique, for example, using Gershgorin's theorem. In this case, you should include in the report a description how the bounds are found.

It is possible to use some other program (or program implementation of the Lanczos method) to compute approximations of the extreme eigenvalues of A.

Writing a report on the results

The report has to have the following structure:

(i) Brief problem description, namely, the solution of linear systems of equations with symmetric positive definite matrices, using iterative solution methods.

(ii) Theory
Describe briefly the theory regarding the convergence of any of the methods M1-to M4, including derivation of Task 4.

(iii) Numerical experiments

(a) Describe the experiments. Include table(s) with iteration counts for various problem sizes. Present some typical plots of the the relative residual ($\|\mathbf{r}_k\|/\|\mathbf{r}_0\|$) convergence and the error convergence $\|\mathbf{x}^* - \mathbf{x}_k\|$.
It is important to describe what is plotted on the different coordinate axes. Another suggestion for the residual plots is to use `semilogy` in order to better see the convergence history.
OBS! The numerical tests possible to performed are numerous. Do not include everything (or rather too much) in the report. Choose a representative information to show the typical behaviour and comment on the rest of the tests run, if necessary. For instance, do not run inefficient (slowly converging) methods on very large problems. The behaviour can be seen on small-to-medium sized problems. Still, one needs, say, three consecutive problem sizes to see how does the iteration count grow with the problem size.

(b) Analyse the numerical results in comparison with the theoretical.
Do you see iteraion counts for the CG as predicted from the theoretical estimates? If the condition number of the matrices is proportional to h^{-2}, how does the iteration count change when h is decreased?
Does the convergence of the Chebyshev method improve significantly if you use the exact eigenvalues instead of their approximations obtained by the Lanczos method? For which problems?

(c) (Not compulsory) Upon your time, experience and interests, you could try another preconditioner, for example, the incomplete Cholesky preconditioner, produced by `Matlab`'s command `U = cholinc(A,tol);`, where `tol` can be chosen for instance as $0.1, 0.05, 0.01$. How much does the preconditioner you have chosen improve the convergence of the CG method? How does the number of iterations depend on the size of the matrix, compared to the unpreconditioned CG?

(iv) Give your conclusions. Which is you method of choice? Motivate your answer.

Printouts of the program codes should be attached to the report.
Working in pairs is recommended. However, all topics in the assignment should be covered by each participant.
For your convenience, the Latex source of this assignment is at your disposal and you can use it for the report if you wish.

Deadline: The solutions should be delivered to me no later than **December 10, 2007**.
Success!

Maya (`Maya.Neytcheva@it.uu.se`, room 2307)

Any comments on the assignment will be highly appreciated and will be considered for further improvements. Thank you!

Apendix: The Second Order Chebyshev iterative solution method

Description

Let A be a real symmetric positive definite (s.p.d.) matrix of order n. Consider the solution of the linear system $A\mathbf{x} = \mathbf{b}$ using the following iterative scheme, known as the *Second Order Chebyshev iterative solution method*:

$$\begin{array}{rl} \mathbf{x}_0 \text{ given}, & \mathbf{x}_1 = \mathbf{x}_0 + \frac{1}{2}\beta_0 \mathbf{r}_0 \\ \text{For} & k = 0, 1, \cdots \text{ until convergence} \\ & \mathbf{x}_{k+1} = \alpha_k \mathbf{x}_k + (1-\alpha_k)\mathbf{x}_{k-1} + \beta_k \mathbf{r}_k. \\ & \mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k. \end{array} \tag{2}$$

Let $\mathbf{x}^*$ be the exact solution of the above linear system. Denote by $\mathbf{e}_k = \mathbf{x}^* - \mathbf{x}_k$ the iterative error at step k. Clearly, $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k = A(\mathbf{x}^* - \mathbf{x}_k) = A\mathbf{e}_k$.

It is seen from the recursive formula (2) that for each k, the errors satisfy a relation of the form

$$\mathbf{e}_{k+1} = \alpha_k \mathbf{e}_k + (1-\alpha_k)\mathbf{e}_{k-1} + \beta_k A\mathbf{e}_k = Q_k(A)\mathbf{e}_0,$$

where $Q_k(\cdot)$ is some polynomial of degree k. Furthermore, the polynomials $Q_k(A)$ are related among themselves as follows:

$$Q_{k+1}(A) - \alpha_k Q_k(A) - \beta_k A Q_k(A) + (1-\alpha_k)Q_{k-1}(A) = 0, \ k = 1, 2, \cdots \tag{3}$$

We compare the recurrence (3) with the recursive formula for the Chebyshev polynomials, namely,

$$T_0(z) = 1, \quad T_1(z) = z, \quad T_{k+1}(z) - 2T_k(z) + T_{k-1}(z) = 0. \tag{4}$$

One can easily see that for the following special choice of the method parameters

$$\alpha_k = \frac{2c\,T_k(z)}{T_{k+1}(z)} = 1 + \frac{T_{k-1}(z)}{T_{k+1}(z)}$$

and

$$\beta_k = \frac{4}{b-a} T_k(z)/T_{k+1}(z), \quad \text{where } z = \frac{b+a}{b-a}, \quad 0 < a \le \lambda_{min}(A), \ \lambda_{max}(A) \le b,$$

we get

$$Q_k(A) = \frac{T_k(z)}{T_k(z)}, \qquad Z = \frac{1}{b-a}\left[(b+a)I - 2A\right].$$

We now recall that the Chebyshev polynomials possess the following optimal approximation property - among all normalized polynomials of degree n defined in an interval $[a, b]$, the nth degree Chebyshev polynomial is the one which differs from zero least (measured in local min and max in $[a, b]$).

Therefore, for this particular polynomial $Q_k(A)$ we have (due the above approximation properties of the Chebyshev polynomials)

$$\max_z |Q_k(A)z| \le \min_{P \in \Pi_k} |P(A)z|,$$

i.e., at each step we achieve the best possible error reduction. (Here Π_k is the set of all polynomials of degree k.)

In order to use the Chebyshev iteration method we need to estimate the extreme eigenvalues of A and to determine an interval $[a, b]$ which contains the spectrum of A.
Having done that, one finds the following formulas to compute the method parameters recursively:

$$\alpha_k = \frac{a+b}{2}\beta_k, \quad \frac{1}{\beta_k} = \frac{a+b}{2} - \left(\frac{b-a}{4}\right)^2 \beta_{k-1}, \quad \beta_0 = \frac{4}{a+b}. \tag{5}$$

Note that $\alpha_k > 1, k \geq 1$.

Theorem 1 *The following results hold:*

$$\lim_{k\to\infty} \beta_k = \frac{4}{\left(\sqrt{a}+\sqrt{b}\right)^2},$$

$$\frac{\|\mathbf{e}_k\|_A}{\|\mathbf{e}_0\|_A} \leq \frac{1}{T_k\left(\frac{b+a}{b-a}\right)} \leq 2\frac{\sigma^k}{1+\sigma^{2k}}, \qquad \sigma = \frac{1-\sqrt{a/b}}{1+\sqrt{a/b}}.$$

It follows then that $\frac{\|\mathbf{e}_k\|_A}{\|\mathbf{e}_0\|_A} \to 0$ *monotonically.*

(For the special choice $a = \lambda_{min}(A)$, $b = \lambda_{max}(A)$, *we have* $\sigma = \frac{1-\sqrt{\varkappa(A)^{-1}}}{1+\sqrt{\varkappa(A)^{-1}}}$.*)*
From the above convergence rate estimate one can determine a priory the number of Chebyshev iterations needed to be performed in order to achieve an error reduction

$$\frac{\|\mathbf{e}_k\|_A}{\|\mathbf{e}_0\|_A} \leq \varepsilon. \tag{6}$$

Indeed, to insure (6), it suffices to perform

$$k \geq \ln\left(\frac{1}{\varepsilon} + \sqrt{\frac{1}{\varepsilon^2}-1}\right) / \ln(\sigma^{-1}) \quad \textit{or} \quad k^* = \left\lceil \frac{1}{2}\sqrt{\frac{b}{a}}\ln\frac{2}{\varepsilon} \right\rceil \quad \textit{iterations.} \tag{7}$$