Introduction to Numerical Simulation (Fall 2003)
Problem Set #4 - Solutions

1 Problem 4.1

a) Both *sgcr.m* and *tgcr.m* are identical under exact arithmetic, but they are different under finite precision arithmetic, like the one based on floating point representation. *sgcr.m* uses the classical Gram-Schmidt (CGS) which is numerically unstable, and *tgcr.m* is based on the modified Gram-Schmidt (MGS) which is more stable.

Consider the following example where we want to orthogonalize a vector $v = [1+\epsilon,\ 1-\epsilon]^T$ with respect to two orthonormal vectors

$$q_1 = \frac{1}{\sqrt{2}}[1,\ 1]^T,$$
$$q_2 = \frac{1}{\sqrt{2}}[1,\ -1]^T.$$

For both CGS and MGS, the first step is the same:

$$\beta = v^T q_1 = \frac{1}{\sqrt{2}}\{(1+\epsilon)+(1-\epsilon)\} = \sqrt{2},$$

where there is not much cancellation error. Hence the updated v, is

$$v' = v - \beta q_1 = [\epsilon',\ -\epsilon']^T,$$

where $\epsilon' = \epsilon + (\text{rounding errors})$ is used since $(1+\epsilon)-1$ generates some cancellation errors.

Now, the CGS and MGS differs in the next step. For MGS,

$$\beta = (v')^T q_2 = \sqrt{2}\epsilon',$$
$$v'' = v' - \beta q_2 = 0,$$

The projection step does not introduce much numerical error, since the operands, ϵ''s, are in the same order. But, for CGS,

$$\begin{aligned} \beta &= v^T q_2 = \frac{1}{\sqrt{2}}\{(1+\epsilon')-(1-\epsilon')\} = \sqrt{2}\epsilon'', \\ v'' &= v' - \beta q_2 = [\epsilon' - \epsilon'',\ -\epsilon' + \epsilon''], \end{aligned} \tag{1}$$

where $\epsilon'' = \epsilon' + (\text{rounding errors})$ models the rounding error introduced in evaluating β, because 1 and ϵ are very different in magnitude.

b) If the matrix is symmetric, the back orthogonalization process, which are CGS for *sgcr.m* and MGS for *tgcr.m*, is effectively one step. That is, you have to normalize the kth search direction to only $(k-1)$th search direction. MGS and CGS behave exactly the same

for the first orthogonalization step, hence, it is hard for you to see any difference.

c) Your matrix has to satisfy some criteria:

1. It should not be symmetric.
2. It should be large enough so that the error can accumulate.
3. The columns should be nearly linearly dependent as much as possible, but not exactly dependent.

One easy example that one can come up with is a large enough random matrix, say 70-by-70. You can generate one easily by typing *A=rand(70).* But when you do this, you have to check the cause of a break down because A can be indefinite and *alpha* becomes 0. Think about how you can do this.

A better one is the Vandermonde matrix, which arises when you want to do polynomial fitting in a naive manner. (Thank you Carlos for letting me know about this. =)) Type *A=vander([0:0.1:0.9])* for a 10-by-10 Vandermonde matrix, and this will make the *sgcr.m* break down.

2 Problem 4.2

a) We can use just the following simple function which is zero on outlying eigenvalues:

$$p_3(x) = \frac{1}{4000}(x-1)(x-20)(x-200)$$

We can easily see that on all of the eigenvalues this polynomial is less than one, therefore we are done.

b) For $k \geq 3$,

$$\begin{aligned}\frac{||r^k||}{||r^0||} &\leq \max_i |p_k(\lambda_i)| \\ &= \max_i |p_2(\lambda_i)p_{k-2}(\lambda_i)| \\ &\leq \max_i |p_2(\lambda_i)| \cdot \max_i |p_{k-2}(\lambda_i)|\end{aligned}$$

Then, we do the following for each polynomial:

1. Since we know that one eigenvalue is at 20 and another at 200, we make $p_2(\lambda_i)$ to become zero at these points. To ensure the condition $p_k(0) = 1$, $p_2(x)$ should be

$$p_2(x) = \frac{1}{4000}(x-20)(x-200).$$

This polynomial is always less than unity for $x \in [.5, 3]$.

2. Because $p_2(x)$ is zero at $x = 10$ and 100, the largest value of $|p_{k-2}(x)|$ should be minimized over $[.5, 3]$,

$$\max_i |p_{k-2}(\lambda_i)| = \max_{x\in[0.5,3]} |p_{k-2}(x)|.$$

Then, the optimal polynomial that does this is the Chebyshev polynomial, which gives the bound

$$\max_{x\in[0.5,3]} |p_{k-2}(x)| = 2(\frac{\sqrt{6}-1}{\sqrt{6}+1})^{k-2}. \tag{2}$$

Hence, the overall bound when $k \geq 3$ is

$$\frac{||r^k||}{||r^0||} \leq 2(\frac{\sqrt{6}-1}{\sqrt{6}+1})^{k-2}. \tag{3}$$

For $k \leq 2$, we have to find them separately:

$$\begin{aligned} \frac{||r^0||}{||r^0||} &= 1 \\ \frac{||r^1||}{||r^0||} &= |\frac{1}{200}(1-200)| = 0.99 \\ \frac{||r^2||}{||r^0||} &= |\frac{1}{4000}(1-20)(1-200)| = 0.97 \end{aligned} \tag{4}$$

c) For 10 test cases, the residual of the *tgcr.m* is shown in Figs. 1.

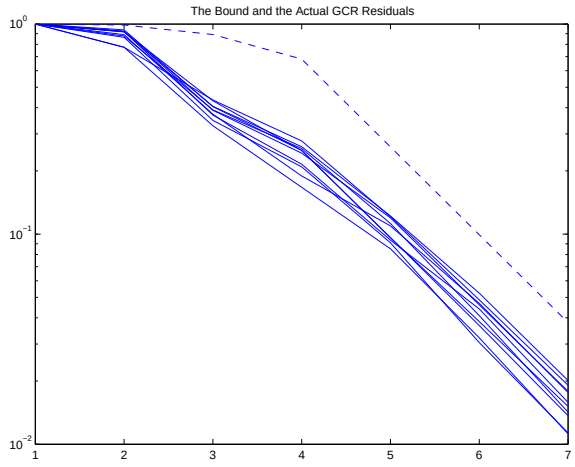

Figure 1: Plot of the residual for *tgcr.m* with the bound in b).

3 Problem 4.3

a) Suppose $p(x, y)$ has a constant term c such that

$$p(x, y) = p'(x, y) + c \quad \text{where } p'(0, 0) = 0. \tag{5}$$

Then, the differential equation and the boundary conditions for the pure Neumann problem can be written as

$$\begin{aligned} \frac{\partial^2 p(x,y)}{\partial x^2} + \frac{\partial^2 p(x,y)}{\partial y^2} = \frac{\partial^2 p'(x,y)}{\partial x^2} + \frac{\partial^2 p'(x,y)}{\partial y^2} = b(x, y), \\ \frac{\partial p(x,y)}{\partial x} = \frac{\partial p'(x,y)}{\partial x} = 0, \\ \frac{\partial p(x,y)}{\partial y} = \frac{\partial p'(x,y)}{\partial y} = 0, \end{aligned}$$

and none of them specifies the value of c. This means that any value of c can be used for the solution, in other words, the A matrix is singular. Hence, a potential at one point should specified to get a unique value of c.

b) We got 0.8280 sec for *tgcr_sym.m* which is modified code for symmetric matrices, and 1.7040 sec for ordinary *tgcr.m*. A machine with Pentium 4, 2GHz CPU and 1GByte memory is used for the measurement. The difference comes from the fact that the cost of orthogonalization does not increase as the GCR algorithm progresses if truncated back orthogonalization is used. (You can easily measure the time taken using *tic* and *toc* in MATLAB.)

c) The change in the residuals for the purely Neumann problem and partly Dirichlet problem are given in Figs. 2. One reason that the problem with partly Dirichlet problem converges faster can be found from the eigenvalue distribution. From Figs. 3, one can see that the smallest eigenvalue for the purely Neumann problem is much smaller than the one for the partly Dirichlet problem. Hence, eliminating this eigenvalue will make the polynomial overshoot at the other eigenvalues, and the residual can only be made small when we put zeros near sufficiently many eigenvalues. For example, if $p(x)$ is

$$p(x) = \frac{(x - 10^{-6})(x - 1)}{10^{-6} \cdot 1}, \tag{6}$$

to eliminate a zero at 10^{-6}, $p(2) \approx 10^6$ and we have to put zeros at larger eigenvalues to make the value small.

Figs. 4, which is the magnified plot of Figs. 2, shows an interesting behavior of the partly Dirichlet case. That is, the residual drops quite a lot after 20 iterations. This means that it takes about 20 iterations for the boundary condition at one end to propagate to the other end, which makes sense for our case.

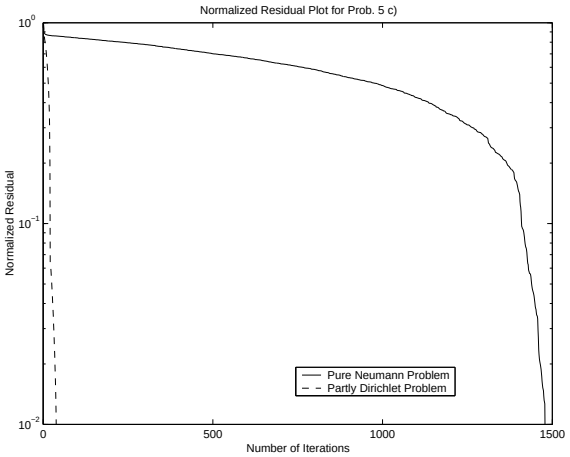

Figure 2: Plot of the residual for the purely Neumann and partly Dirichlet problems.

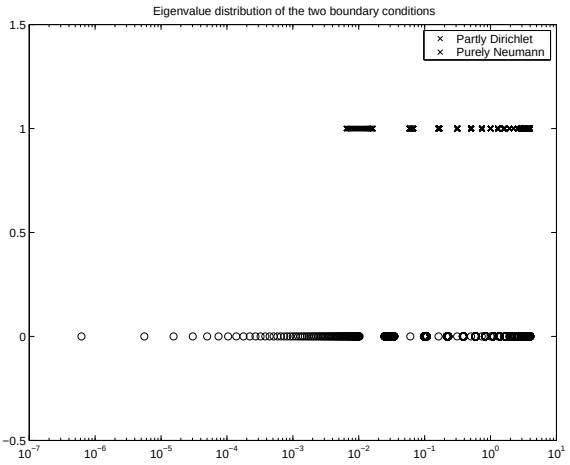

Figure 3: Eigenvalue distributions for purely Neumann and partly Dirichlet problems.

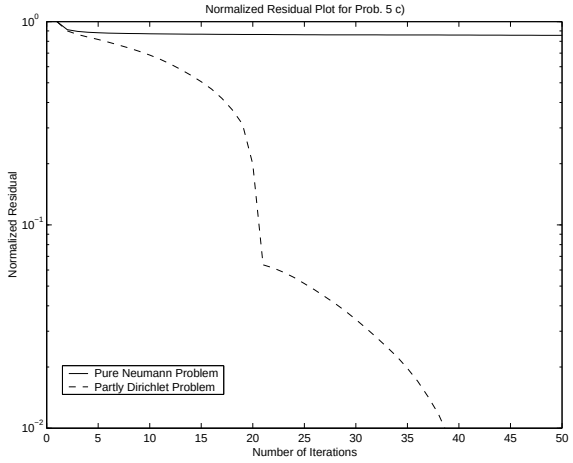

Figure 4: Magnified plot of the residual for the purely Neumann and partly Dirichlet problems.