Final Project for Math 6644 : Iterative Methods for Systems of Equations
(Due on April 23)

You are encouraged to design your own projects if you are particularly interested in solving some practical problems using the course materials. Please come to see me if you have any idea in your mind.

Preconditioned Conjugate Gradient Methods

A $n \times n$ matrix A_n is called Toeplitz matrix if it has the form

$$A_n = \begin{bmatrix} a_0 & a_{-1} & \cdots & a_{2-n} & a_{1-n} \\ a_1 & a_0 & a_{-1} & & a_{2-n} \\ \vdots & a_1 & a_0 & \ddots & \vdots \\ a_{n-2} & & \ddots & \ddots & a_{-1} \\ a_{n-1} & a_{n-2} & \cdots & a_1 & a_0 \end{bmatrix},$$

i.e. A_n is constant along its diagonal. In this part of the project, we consider solving symmetric positive definite Toeplitz systems by preconditioned CG methods using circulant matrices as the preconditioners.

A $n \times n$ matrix C_n is called circulant matrix if

$$C_n = \begin{bmatrix} c_0 & c_{-1} & \cdots & c_{2-n} & c_{1-n} \\ c_{1-n} & c_0 & c_{-1} & & c_{2-n} \\ \vdots & c_{1-n} & c_0 & \ddots & \vdots \\ c_{-2} & & \ddots & \ddots & c_{-1} \\ c_{-1} & c_{-2} & \cdots & c_{1-n} & c_0 \end{bmatrix}.$$

Circulant matrices are diagonalized by the Fourier matrix F_n, i.e.

$$C_n = F_n^* \Lambda_n F_n,$$

where $[F]_{j,k} = \frac{1}{\sqrt{n}} e^{2\pi i jk/n}$ for $0 \le j, k \le n-1$ and Λ_n is a diagonal matrix holding the eigenvalues of C_n. In fact, Λ_n can be obtained in $O(n \log n)$ operations by taking FFT (Fast Fourier Transform) of the first column of C_n (why?). Once Λ_n is obtained, the products $C_n \vec{y}$ and $C_n^{-1} \vec{y}$ can be computed by FFT using $O(n \log n)$ operations for any given vector $\vec{y}$.

Solving a symmetric positive definite Toeplitz system

$$A_n \vec{x} = \vec{b}$$

by CG can be speed up by using circulant preconditioners. Commonly used circulant preconditioners include:

(i) G. Strang's circulant preconditioner which is defined as $[C_n]_{k,l} = c_{k-l}$ for $0 \le k, l < n$, where

$$c_j = \begin{cases} a_j & 0 \le j \le [n/2] \\ a_{j-n} & [n/2] < j < n \\ c_{n+j} & 0 < -j < n \end{cases},$$

where operator $[x]$ returns the closest integer (smaller) than x.

(ii) T. Chan's circulant preconditioner

$$c_j = \begin{cases} \frac{(n-j)a_j + j a_{j-n}}{n} & 0 \le j < n \\ c_{n+j} & 0 < -j < n \end{cases}$$

Write your code to perform the CG and PCG for Toeplitz systems using both circulant matrices as the preconditioners. The right hand side $\vec{b}$ is a random vector selected by you.

Experiment your code for the following symmetric toeplitz systems with n varying from $50, 100, 200, 400, \cdots$:

(a) $a_k = |k+1|^{-p}$ for the lower triangular part of A_n, where $p = 2, 1, 1/10, 1/100$. (for $p \le 1$, is the system still positive definite?)

(b) The elements of the Toeplitz system is defined by

$$a_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\theta) e^{-ik\theta} d\theta, \quad k = 0, \pm 1, \pm 2, \ldots,$$

where $f(\theta) = \theta^4 + 1$ for $-\pi \le \theta \le \pi$. This means that a_k is the Fourier coefficients of function $f(\theta)$, and you should obtain a_k by FFT.

Terminate your computations if the relative residual $||\vec{r}_k||/||\vec{r}_0|| \le 10^{-6}$. List the number of iterations needed for CG without preconditioner, and PCG with G. Strang's and T. Chan's circulant preconditioners in a table in each case. Also compare the number of flops (in case you can not count the flops easily, use the CPU time instead) against the matrix size N in each case. Do you get the $O(n \log n)$ growing pattern? Comment on your results.

Remark: Computing $A_n\vec{x}$ for a Toeplitz system can also be carried out in $O(n \log n)$ operations by using FFT. A hint is to embed A_n into a $2n \times 2n$ circulant matrix and extend $\vec{x}$ to a $2n$-vector by adding zeros. Then by using FFT, one can compute the circulant matrix-vector multiplication and obtain the corresponding $A_n\vec{x}$ from the resulting $2n$-vector.

Fixed-point and Newton's methods for nonlinear systems

Consider the discrete Chandrasekhar H-equation

$$\mathbf{F}_i(\vec{x}) = x_i - \left(1 - \frac{c}{2N} \sum_{j=1}^{N} \frac{\mu_i x_j}{\mu_i + \mu_j}\right)^{-1} = 0, \tag{1}$$

where $c \in (0, 1)$ is a give constant, $\mu_i = (i - 1/2)/N$ for $1 \le i \le N$, and N is the dimension of the unknown vector $\vec{x}$.

Write your own code and compute the solution of the equation for $N = 200$ and $c = 0.9$ by using

(1) Fixed-point method,
(2) Chord method,
(3) Newton method,
(4) Shamanskii method with $m = 2$.

In all of your computations, the initial guess is taken as $\vec{x} = [1, 1, \cdots, 1]^T$, the stopping condition is that

$$\|\mathbf{F}(\vec{x})\| \le \tau_r r_0 + \tau_a,$$

where $\tau_r = \tau_a = 10^{-6}$.

Compare your computation results (by tabulating or plotting the iteration numbers, error reduction, and cost) and comment on your computations.

Remark: You may use MATLAB code *diffjac* to compute the Jacobian in your computations.

References

[1] G. Strang, *A Proposal for Toeplitz Matrix Calculations*, Stud. Appl. Math., 74(1986), pp171-176.

[2] R. Chan and G. Strang, *Toeplitz Equations by Conjugate Gradients with Circulant Preconditioner*, SIAM J. Sci. Stat. Comput., 10(1989), pp 104-117.

[3] T. Chan, *An Optimal Circulant Preconditioner for Toeplitz Systems*, SIAM J. Sci. Stat. Comput. 9 (1988) pp 766-771.