CS 290N/219: Sparse matrix algorithms: Homework 4

Assigned October 28, 2009

Due by class Wednesday, November 4

1. [20 points]

(a) Find a 2-by-2 matrix A that is symmetric and nonsingular, but for which neither A nor $-A$ is positive definite. What are the eigenvalues of A? Find a 2-vector y such that $y^T Ay < 0$.

(b) For A as above, find a 2-vector b such that the conjugate gradient algorithm, when started with the zero vector as an initial guess, does not converge to the solution of $Ax = b$. Show what happens on the first two iterations of CG, as in the October 28 class slides. How do you know it won't converge to the right answer?

2. [40 points] In this problem you'll actually prove that CG works in at most n steps, assuming that real numbers are represented exactly. (This is not a realistic assumption in floating-point arithmetic, or on any computer with a finite amount of hardware, but it gives a solid theoretical underpinning to CG.) Let A be an n-by-n symmetric, positive definite matrix, and let b be an n-vector.

We start with the idea of searching through n-dimensional space for the value of x that minimizes $f(x) = \frac{1}{2}x^T Ax - b^T x$, which is the x that satisfies $Ax = b$. We begin by picking a set of n linearly independent search directions, called d_0, d_1, …, d_{n-1}. (Actually we don't know them in advance, but that's a detail.) At each iteration we proceed along the next direction until we are "lined up" with the final answer, the value of x at which $Ax = b$. In n-space, once we are lined up with the answer from n independent directions, we will be exactly on the answer.

The **first magic of CG** is that for the right kind of search directions, there is a way to define "lined up" for which we can actually compute how far to go along each search direction. The key definition uses A-*conjugate* vectors. Then "lined up" means that the *error* $e_i = x_i - x$ is exactly crossways to the search direction d_{i-1}, not in the sense of being perpendicular (which would mean $e_i^T d_{i-1} = 0$), but in the sense of being A-conjugate: $e_i^T Ad_{i-1} = 0$.

An informal way to say that is, we proceed along the search direction until we are lined up with the solution as seen through A-glasses. The reason for lining up through A-glasses rather than bare eyes is that we can compute where to stop without knowing where the final answer is. We can't see and compute with x-space directly, but we can see the space where Ax and b live. And after lining up each of n independent directions in an n-dimensional space we are guaranteed to be sitting on top of the right answer, whether the independent directions are the conventional coordinate axes or the A-conjugate axes we see through our A-glasses.

To go along with this, we need to choose the search directions themselves to be mutually A-conjugate: we will require each d_i to be A-conjugate to all the earlier d_j's, so $d_i^T Ad_j = 0$ if $i \neq j$.

(a) Suppose we are given i mutually A-conjugate vectors d_0, …, d_{i-1}. Suppose $x_0 = 0$, and for each $j < i$ we have $x_j = x_{j-1} + \alpha_j d_{j-1}$. Write down and prove correct an expression for a scalar α_i such that, if we take $x_i = x_{i-1} + \alpha_i d_{i-1}$, then the error $e_i = x_i - x$ is A-conjugate to d_{i-1}.

Now, how do we get a sequence of A-conjugate directions to search along? In fact, we can start with any sequence of linearly independent directions, and convert them to A-conjugate directions by projecting out all the earlier search directions from each one, using Gram-Schmidt orthogonalization, as follows.

(b) Suppose we are given i mutually A-conjugate vectors $d_0, \ldots, d_{i-1}$, and one more vector u_i that does not lie in their span. Write down and prove correct an expression for scalars $\beta_{i,j}$ such that, if we take

$$d_i = u_i + \sum_{j=0}^{i-1} \beta_{i,j} d_j,$$

then d_i is A-conjugate to all the earlier d_j.

Finally, the **second magic of CG** is that there is a way to choose a particular sequence of directions for which the Gram-Schmidt orthogonalization is really easy. If we choose the right directions to start with, we only need to project out *one* earlier direction, not all i of them. This is why the cost of one CG iteration is only $O(n)$, not $O(n^2)$.

(c) Suppose the vectors $d_0, \ldots, d_{i-1}$, the vectors $x_0, \ldots, x_{i-1}$, and the scalars α_j and $\beta_{i,j}$ are as above. Suppose in addition that at each stage we take $u_i = b - Ax_i$ (which is also known as r_i, the residual). First, prove that if this choice of u_i lies in the span of $d_0, \ldots, d_{i-1}$, the CG iteration can stop with $x_i = x$. Second, show that this direction u_i is already A-conjugate to all of the d_j except d_{i-1}, and therefore we can take $\beta_{i,j} = 0$ for $j < i - 1$.

(d) One last detail: Prove that the CG code on the course slide does in fact compute the residual r_i correctly; that is, prove that $r_{i-1} - \alpha_i A d_{i-1}$ is in fact equal to $b - Ax_i$.