Krylov Subspace Methods

When solving differential equations one can encounter matrices so large that they don't fit in memory. Such matrices are generally given by a formula that allows each entry a_{ij} to be recomputed whenever it is needed. An example of such a matrix is given by

$$a_{ij} = \begin{cases} 4 & \text{if } i = j \\ -1 & \text{if } i = j+1 \text{ or } i+1 = j \\ 0 & \text{otherwise.} \end{cases}$$

Note many of the entries a_{ij} are identically zero. This is so often the case that Matlab has a matrix type called sparse that stores only the non-zero entries of A. For the tridiagonal matrix given above one could also store the upper, lower and main diagonal entries as three vectors. Moreover, because the location of the zeros in tridiagonal matrices is particularly convenient, in this case Gaussian elimination can be efficiently used to solve $Ax = b$.

This is not always the case. Consider the matrix given by

$$a_{ij} = \begin{cases} 5 & \text{if } i = j \\ -1/|i-j|^2 & \text{otherwise.} \end{cases}$$

All entries of this matrix are non-zero. Still, multiplication by A can be represented by a function that computes Ax for any vector x. In Matlab we may express this matrix as

Matlab Example 22a

```
function y=multA(x)
    n=length(x);
    c=[-1.0./[n-1:-1:1].^2,5,-1.0./[1:n-1].^2];
    for j=n:-1:1
        y(j,1)=c(n-j+1:2*n-j)*x;
    end
```

Our goal is to find an approximate solution to the equation $Ax = b$. The difficulty is that since A doesn't fit into memory we can't use the LU or QR decompositions directly.

Define the m-th Krylov subspace as

$$\mathcal{K}_m = \text{span}\{b, Ab, A^2b, \ldots, A^{m-1}b\}.$$

Take as an approximate solution of $Ax = b$ the $x^* \in K_m$ that minimizes $\|Ax - b\|$. Note that the vectors $b, Ab, \ldots, A^{m-1}$ may not all be linearly independent. When they are linearly dependent there is a non-trivial linear combination of vectors such that

$$c_0 b + c_1 Ab + c_2 A^2 b + \cdots + c_{m-1} A^{m-1} b = 0.$$

In this case we assume may that $c_0 \neq 0$ and then solve to find that

$$x^* = \frac{-1}{c_0}\big(c_1 b + c_2 Ab + \cdots + c_{m-1} A^{m-2} b\big) \in \mathcal{K}_m$$

is an exact solution to $Ax = b$.

Since it is impossible to have more than n linearly independent vectors in $\mathbf{R}^n$, then taking $m \geq n$ guarantees that the exact solution to $Ax = b$ is contained in $\mathcal{K}_m$. Thus, for large enough m there is $x^* \in \mathcal{K}_m$ which is a good approximation to $Ax = b$. On the other hand, if m is not too large then we can store a basis for $\mathcal{K}_m$ in memory. Our hope is to obtain a good approximation for small m. This requires A to have special properties.

Generally, we prefer to work with an orthogonal basis, so let

$$\Gamma_m = \left[b \middle| Ab \middle| \cdots \middle| A^{m-1}b \right]$$

and form the reduced QR decomposition $\Gamma_m = Q_m R_m$ where Q_m is an $n \times m$ matrix and R_m is $m \times m$. It follows that $x^* = Q_m z$ where z is the least squares solution to the over determined system of equations $AQ_m z = b$.

Let us illustrate this technique, called GMRES for generalized minimum residual method, to solve $Ax = b$ using the matrix defined in Matlab Example 22a. We take $m = 10$ and $n = 10000$. Note that a 10000×10000 matrix would take 762MB of memory to store whereas a 10×10000 matrix only takes 762KB.

Matlab Example 22b

```
>> m=10;
>> n=10000;
>> b=rand(n,1);
>> Gamma(:,1)=b;
>> for j=2:m
>>     Gamma(:,j)=multA(Gamma(:,j-1));
>> end
>> [Q,R]=qr(Gamma,0);
>> clear Gamma;
>> for j=1:m
>>     AQ(:,j)=multA(Q(:,j));
>> end
>> z=AQ\b;
>> x=Q*z;
>> r=b-multA(x);
>> norm(r)
ans = 0.00050635
```

A better approximation can be found by increasing m or by iterating. Iteration is accomplished by defining the residual $r = b - Ax^*$ and then approximating the solution y^* to $Ay = r$ using the same techniques as before. In this case

$$A(x^* + y^*) = Ax^* + Ay^* \approx Ax^* + r = b.$$

Therefore $x^* + y^*$ should be a better approximation to the solution of $Ax = b$. Iterating reduces the error in Matlab Example 22b to $\|A(x^* + y^*) - b\| = 1.0554 \times 10^{-8}$.

Next time we discuss some refinements that allow efficient computation of the approximation lying in $\mathcal{K}_{m+1}$ after already having found the approximation in $\mathcal{K}_m$.