September 13, 2017

Matlab Assignments – Lecture 6, Fall 2016

Assignment 6.1. **Conjugate Residuals.** If $\mathbf{A}$ is Hermitian ($\mathbf{A}^* = \mathbf{A}$), then the orthogonalization loop in GCR can be reduced to a single update. The resulting algorithm is called CR (*Conjugate Residuals*). Write a function subroutine `CR.m` for CR. Compare the convergence (in number of iteration steps and in computational time) of GCR and CR for a symmetric matrix. (Diagonal matrices can be used for experiments (why?). Take $\mathbf{A} = \text{diag}(\Lambda)$ with $\Lambda = 1 : 400$ and $\Lambda = \sqrt{1 : 400}$. Consider also shifted versions $\mathbf{A} \leftarrow \mathbf{A} - \sigma\mathbf{I}$ with σ between 1 en 10.)

How does CR perform on a (weakly) non Hermitian problem (Take a matrix of the form $\mathbf{TDT}$ with $\mathbf{D}$ diagonal and $\mathbf{T} = \mathbf{I} - \gamma\mathbf{S}$ with γ small and $\mathbf{S}$ the *Shift matrix*, that is, all entries are equal to 0 except for $S_{i+1,i} = 1$ all i)?

Assignment 6.2. **Restart and truncation.** The computational costs as well as storage requirements of both GCR and GMRES increase with increasing step number. To limit the computational costs, the methods can be *restarted* after ℓ steps, i.e., every ℓth steps, solve $\mathbf{Ax} = \mathbf{b}$ with ℓ steps of GCR or GMRES with initial guess the approximate $\mathbf{x}_{j\ell}$ from the preceding sweep of ℓ steps. Show that this strategy can conveniently be incorporated in GCR by replacing the line '`for` $j = 0, 1, \ldots, k-1$ `do`' by '`for` $j = \pi(k), \pi(k)+1, \ldots, k-1$', with $\pi(k) \equiv m\ell$ if $k \geq m\ell$ ($m \in \mathbb{N}_0$ maximal). (Note that the loop in the MATLAB code starts with 1 rather than 0: MATLAB does not allow 0 index such as $u(:, 0)$.) Write a restarted version for GCR and GMRES. Investigate experimentally the effect of the restart on the convergence.

Another strategy to limit the computational costs in GCR is to limit the orthogonalization loop to the ℓ most recent vectors c_j, that is, take $\pi(k) = \max(0, k-\ell)$. Incorporate this *truncation* strategy in GCR (adapt the code also to minimise storage). What is the effect of truncation on the convergence? Try also symmetric matrices and weakly non-symmetric ones.

Can truncation also be incorporated in GMRES?

Assignment 6.3. **Stability GMRES.** GCR and GMRES are mathematically equivalent, that is, started with the same initial guess and assuming exact arithmetic, the kth GMRES residual is equal to the kth GCR residual. The number of MVs (matrix vector multiplications) are the same. With respect to the other computational costs (and memory requirements), GCR is approximately twice as expensive as GMRES.

Is this visible in the computational time?

To compare stability, select a step number k_0 (say, $k_0 = 2$ or $k_0 = 20$) and perturb the k_0 generating vector (with $\mathbf{w} = \mathbf{r}_{k_0}$, $u_{k_0} = \mathbf{w} + \Delta$ in GCR, with $\mathbf{w} = \mathbf{Av}_{k-1}$, orthogonalize $\mathbf{w} + \Delta$ in GMRES. Take Δ a random vector of size $\delta\|\mathbf{w}\|_2$ ($\delta = 10^{-5}$, $\delta = 10^{-2}$). How does this affect convergence?