January 24, 2011

Notes on numerical experiments with iterative Methods

If you evaluate the performance of numerical solvers for linear systems, make sure that you compare comparable quatities.

• Make sure that you not only apply each method to the same problem $\mathbf{A}\mathbf{x} = \mathbf{b}$, but that you also use the same initial value $\mathbf{x}_0$ (and, if there is an initial shadow residual $\widetilde{\mathbf{r}}_0$, that you also take the same shadow residual. Sometimes a random initial shadow residual gives better results than a structured one as $\widetilde{\mathbf{r}}_0 = \mathbf{r}_0$. If you use a random shadow residual, then make sure that you use the same random one for each method (set the seed for random generator to, say, 0, or save your initial shadow residual).).

• If you use random choices, please make sure that your conclusions do not depend on your random choice: try other random choices as well.

• In general, the convergence behavior of a method does not heavily depend on the specific value of $\mathbf{b}$. So you can do meaningful experiments with, for instance, $\mathbf{b} = \mathbf{1}$ (all coordinates equal to 1) or $\mathbf{b}$ a random vector (of correct size). (You can try to demonstrate this observation numerically).

• Usually it is tried to gain insight in the performance of the method by plotting the 10-log of the 2-norm $\|\mathbf{r}_k\|_2$ of the residual along the vertical axis.

If you use several dicretisations of the same partial differential equation, then you need a scaled 2-norm (if $\mathbf{x}$ is a vector version of a discretized function ϕ, then the scaling should be such that $\|\mathbf{x}\|_2 \approx \|\phi\|_2$, where $\|\phi\|_2^2 = \int_\Omega |\phi|^2 \,\mathrm{d}_V$ and in the limit, for discretization step size to 0, $\|\mathbf{x}\|_2$ should aproach $\|\phi\|_2$.).

Usually, a graph of $\|\mathbf{r}_k\|_2$ is easier to read if you plot the residual *reduction*, i.e., $\|\mathbf{r}_k\|_2/\|\mathbf{r}_0\|_2$, and you stop the iteration if the residual is reduced with some specific factor: $\|\mathbf{r}_k\|_2/\|\mathbf{r}_0\| \leq \mathrm{tol}$ and **not** as $\|\mathbf{r}_k\|_2 \leq \mathrm{tol}$.

• Think carefully about what you put along the horizontal axis. There are several possibilities.

1) The number of iteration steps.

2) The dimension of the Krylov space from which $\mathbf{x}_k$ with $\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$ has been extracted.

3) The number of MVs (multiplication by the $n \times n$ matrix $\mathbf{A}$ with an n-vector).

4) The (cumulative) time t_k needed to compute $\mathbf{x}_k$ (and $\mathbf{r}_k$).

Comments. The number of iteration steps (option 1) is a dubious measure. It depends on what you call one iteration step. Do you call one loop in BiCGSTAB one iteration step or two steps (one step in the BiCG part and a second one in the minimization part)? If you restart GMRES every five steps, GMRES(5), is one superstep one iteration, or are the steps in the inner loop iteration steps? This objection does not apply to the measures in option 2 and option 3.

Actually, you would like to see t_k along the horizontal axis (option 4). But timing often depends on the skill of programmer (or how intensively your machine is being used at the time of running experiments or the programming language [as Matlab] that has been used) and not on the method. So especially if you use codes from different sourses, timing does not have to be informative on the methods. Usually, only a table is displayed that contains only the total time needed to compute $\mathbf{x}_k$ for which $\|\mathbf{r}_k\|_2 \leq \mathrm{tol}\|\mathbf{r}_0\|_2$.

Often the MV (especially if the MV is preconditioned) is considerably more expensive than a few inner products and vector updates. If, in such a case, you work with short recursion methods, then the number of (preconditioned) MVs (option 3) a good indication of the time (option 4). In practice, the matrix will be preconditioned, while you like to compare the convergence behavior of iterative methods without preconditioning. For this reason it is wise to put the number of MVs along the horizontal axis position (option 3).

For GMRES, the number of MVs is not a good indication for the time. Even if the costs of an (preconditioned) MV is significantly greater than a few inner products and vector updates, if the required number of MVs is large (which may depend on the quality of the preconditioneerder), then, in the long run, the cost of the inner products and vector updates in GMRES are still dominant.

If you want to illustrate your statements about the quality of the search space (the Krylov space) then you will go for option 2. Recall that in, for instance, Bi-CG you need two MVs (one with an A and one with the transpose of A) to get the Krylov subspace expanded with one dimension, while Bi-CGSTAB expands the Krylov subspace by one dimension with each MV. For more expensive MVs, the extension of the Krylov space by one dimension with Bi-CG is twice as expensive as with Bi-CGSTAB.

Summary. Generally, it is a good choice to use option 3 for your graphs and supplement them with tables with total times.

If you get codes from the Internet, make sure that those codes display the same quantities along the same axis (and use the same scale). If that is not the case, change the code or adjust the graphs.

You can use the Matlab command ”profile on” to obtain timings of individual steps.