ERROR ESTIMATION IN ITERATIVE SOLUTION OF LINEAR SYSTEMS : ALGORITHMS AND ANALYSIS FOR CG, BI-CG AND GMRES

PUNEET JAIN, KRISHNA MANGLANI AND MURUGESAN VENKATAPATHI

Abstract. We analyzed the uncertainty in convergence when using relative residue as a stopping criterion, and the resulting over/under computation for a tolerance in error. We show that error estimation is significant for efficient and accurate solution of moderate to high condition problems ($\kappa > 100$). An $\mathcal{O}(1)$ estimator (at every iteration) was proposed more than a decade ago, for efficient solving of symmetric positive-definite linear systems by the CG algorithm. Later, an $\mathcal{O}(k^2)$ estimator was described for the GMRES algorithm which allows for non-symmetric linear systems as well, and here k is the iteration number. We suggest a minor modification in this GMRES estimation for increased stability. In this work, we also propose an $\mathcal{O}(n)$ error estimator for A-norm and l_2 norm of the error vector in Bi-CG algorithms that can as well solve non-symmetric linear systems. Note that computational cost of the estimator is expected to be significantly less than the $\mathcal{O}(n^2)$ evaluation at every iteration of these methods in solving problems of dimension n. The robust performance of these estimates as a stopping criterion results in increased savings and accuracy in computation, as condition number and size of problems increase.

Key words. : error; stopping criteria; condition number; Conjugate Gradients; Bi-CG; GMRES

AMS subject classifications.

1. Introduction.

Solving a system of linear equations in the form $Ax = b$ is a ubiquitous requirement in science and engineering (where A is a given matrix, x and b are the unknown and known vectors respectively; $x \in \mathbb{R}^n$ and $b \in \mathbb{R}^n$ if $A \in \mathbb{R}^{n\times n}$). Iterative methods like CG (Conjugate Gradient), Bi-CG (Bi-Conjugate Gradient) and GMRES (Generalized Minimal Residual) are commonly used to solve large linear problems as they require $\mathcal{O}(n^2)$ operations compared to direct solvers which can evaluate A^{-1} explicitly in $\mathcal{O}(n^3)$ operations. Iterations should be stopped when the norm of the error $\epsilon_k = x - x_k$ is less than a desired tolerance, where x is the final solution to the linear system and x_k is the iterate. Since the the actual error is unknown, relative residue ($\frac{\|r_k\|}{\|b\|}$) is considered as stopping criteria where $r_k = b - Ax_k$ is the residual vector at k^{th} iteration. Such stopping criteria can work when the system is well-conditioned and can be erroneous depending on the condition number of A and the choice of initial approximation. It can stop the iterations too early when the norm of error is still much larger than tolerance, or not stop early enough and too many floating point operations having done for the required accuracy. Also when condition number of the matrix is large, the residual of a CG/Bi-CG algorithm need not show monotonic behaviour and oscillate while the actual error might still be (however slowly) converging (and vice versa for the GMRES algorithm). The norm of the relative residue can be as large as κ times or as small as $\frac{1}{\kappa}$ times the norm of the relative error.

Even when most iterative algorithms are used with preconditioners, it is not guaranteed that the condition number of the matrix is reduced, and this is observed with matrices of larger dimensions. In cases where the condition number of the matrix is indeed reduced, one would like the reduction of condition numbers of both the backward problem (i.e. computing x given A and b) and the forward problem (i.e. computing b given A and x). This is because while the former accelerates the convergence, the latter reduces the uncertainty in the relationship between the residue and the actual error allowing one to stop iterations efficiently. Note that the condition number of matrix is given by the product of the forward and backward components, and hence both are inversely related to each other for any given $\kappa(A)$. Thus the relative residue remains a poor and inefficient indicator of convergence in general. Moreover, the condition number of matrix $\kappa(A)$ is typically unknown and costly to compute. Thus for even marginally high condition numbers of matrices ($\kappa(A) > 10^2$), either the accuracy or the efficiency of computation is degraded by the above conundrum. The precision in measurements and engineering today renders both the size and condition number of most problems large; making accurate stopping and restarting criteria indispensable in ensuring computational efficiency of solvers. This motivated methods to compute estimates of some norms of the error in iterative solvers. Such estimators (e.g CGQL) are available for CG algorithm [6]. For solving non-symmetric linear systems using FOM (Full Orthogonalization method) and GMRES (Generalized Minimal Residual) methods, formulas for estimation of errors have been suggested [5] recently. We suggest a minor modification to this estimator proposed by Meurant, to increase its stability and precision. Our objective is to derive an efficient estimator for solving non-symmetric linear systems using BiCG, and present an analysis highlighting the significance of these estimation algorithms for CG, BiCG and GMRES methods. This analysis shows that these estimators are robust and increase the efficiency/accuracy of computing notably. Note that this gain is expected even when errors in estimation itself may not be negligible, as the factor scaling the relative residue to the actual relative error can be more significant, i.e. as large as $\kappa(A)$.

In the second part of this work, we also show that the *expected* reduction of uncertainty in the relative error by using an estimator is proportional to $\kappa(A, x)$, the forward condition number of the problem for all $\kappa(A, x) >> 1$. A point to note is that the error and residue are related trivially by a factor $\kappa(A, x)$, but this same factor relating the *relative* residue and *relative* error *on average*, required a detailed analysis. Later, numerical results averaged over a large number of linear systems were used to verify the derived theorems on this uncertainty in convergence. The under and over computation due to this uncertainty is demonstrated, and its large increase with the dimension of the linear system is relatively straight-forward to infer. We show that the *average* under/over computing for problems with matrices of any given $\kappa(A)$ can be approximated with $\delta \sim \frac{log(\kappa(A))}{2log(1/\epsilon_{mc})-log(\kappa(A))}$, and here ϵ_{mc} is the machine precision. Thus error estimation becomes very significant as condition number and size of problems increase.

Section 2 presents related work and discuss CG, Bi-CG and GMRES algorithms and their error estimates. Section 3 presents the analysis of error estimates and its performance as an efficient stopping criteria for these iterative methods of solving linear systems.

2. Methods.

2.1. Related work: Algorithms for CG and GMRES.

2.1.1. CGQL Algorithm (Conjugate Gradient and Quadrature Lanczos). One of the most commonly used methods for solving linear systems with a real Symmetric Positive Definite (SPD) matrix is the Conjugate Gradient (CG) algorithm. It can be derived from several different perspectives, (i) an orthogonalization problem (ii) minimization problem and (iii) Lanczos algorithm.

CGQL algorithm [3] helps in finding estimates of error i.e ($\|x - x_k\|$) at each iteration of CG by inducing a delay in the estimation. The idea of CGQL algorithm is to use CG instead of the Lanczos algorithm, to compute explicitly the entries of the some tridiagonal matrices (T_k) at each iteration k and derive recursive formulas to compute the A-norm of error.

The square of the A-norm of error at CG iteration k is given by:

$$\|\epsilon_k\|_A^2 = \|r_0\|^2[(T_n^{-1})_{(1,1)} - (T_k^{-1})_{(1,1)}] \tag{2.1}$$

Here T_k is the tridiagonal matrix produced by the Lanczos algorithm whose entries can be computed from coefficients of CG algorithm equally well. Also ϵ and r are the error and residual vectors respectively. The main essence of the CGQL lies in computing difference between $(1, 1)$ elements of the inverse of two tridiagonal matrices generated from a Lanczos algorithm with the same starting vectors as CG algorithm. Let d be a delay integer, the approximation of the A-norm of error at iteration $k - d$ is given by

$$\|\epsilon_{k-d}\|_A^2 = \|r_0\|^2[(T_k^{-1})_{(1,1)} - (T_{k-d}^{-1})_{(1,1)}] \tag{2.2}$$

2.1.2. Estimator for GMRES. A delay based error estimator for FOM and GMRES was proposed by Meurant in [5]. Unlike CGQL which uses a Tridiagonal matrix to compute error estimates, GMRES error estimator uses Hessenberg matrix (or its some form). Error estimate denoted by χ_{k-d} at $(k-d)^{th}$ iteration in GMRES as provided by Meurant[5] is given by:

$$\frac{\chi_{k-d}^2}{\|r_0\|^2} = \gamma_{k-d}^2 \left\|\tilde{H}_{k-d}^{-1} e_1\right\|^2 + \left\|\gamma_{k-d} H_{k-d}^{-1} w_{k-d} + (e_{k-d}, H_{k-d}^{-1} e_1) u_{k-d}\right\|^2 \tag{2.3}$$

Here H_k represents the Hessenberg matrix ($H_k = V_k^T A V_k$) at any k^{th} iteration where V_k is a $n \times k$ matrix whose columns are orthonormal basis vectors. e_k is the canonical (identity) vector and all other constants like u, w and γ are derived from GMRES algorithm. A brief explanation regarding the derivation of this estimator can be found in the Appendix.

2.1.3. Proposed modification to the GMRES error estimator. It can be seen that the estimator for GMRES [7] proposed by Meurant in [5] satisfies the

Equation 2.4.

$$\frac{\chi^2_{k-d}}{\|r_0\|^2} = \underbrace{\frac{\|\epsilon_{k-d}\|^2 - \|\epsilon_k\|^2}{\|r_0\|^2}}_{\text{term1}} + \underbrace{\|s_k\|^2}_{\text{term2}} + \tag{2.4}$$

$$\underbrace{\begin{aligned} &\Big[2h_{k+1,k}\left(e_k, H_k^{-1}e_1\right)\left((H_n^{-1}e_{k+1})^k, H_k^{-1}e^1 + s_k\right) \\ &\text{- } 2\mathrm{h}_{k-d+1,k-d}\left(e_{k-d}, H_{k-d}^{-1}e_1\right)\left((H_n^{-1}e_{k-d+1})_{k-d} - (H_k^{-1}e_{k-d+1})^{k-d}, H_{k-d}^{-1}e_1 + s_{k-d}\right)\Big] \end{aligned}}_{\text{term3}}$$

where $s_k = (e_k, H_k^{-1}e_1)u_k$

Equation 2.4 consists of three terms and the third term inside the square bracket is negligible compared to other two terms. Note that for large d which is a delay in estimation, $\|s_k\|^2$ also becomes negligible. However d is kept small in practice (≈ 10), hence in such cases, term $\|s_k\|^2$ significantly contributes to the value of error estimate and needs to be accounted.

For converging problems, $\|\epsilon_{k-d}\|^2 \gg \|\epsilon_k\|^2$,

$$\chi^2_{k-d} \approx \|\epsilon_{k-d}\|^2 + \|r_0\|^2 \|s_k\|^2 \tag{2.5}$$

Equation 2.5 shows that the estimator is offset from exact value of error by term $\|r_0\|^2 \|s_k\|^2$ and should be accounted in the error estimator. This offset term can also cause unstable overshoots in estimation as seen in figure 2.1.

$$\begin{aligned} \frac{\chi^2_{k-d}}{\|r_0\|^2} = |\gamma^2_{k-d}\left\|\tilde{H}^{-1}_{k-d}e_1\right\|^2 \\ + \left\|\gamma_{k-d}H^{-1}_{k-d}w_{k-d} + (e_{k-d}, H^{-1}_{k-d}e_1)u_{k-d}\right\|^2 - \left\|(e_k, H_k^{-1}e_1)u_k\right\|^2| \end{aligned} \tag{2.6}$$

The absolute operation in Equation 2.6 is necessary as in non-converging situations, the operand can become negative.

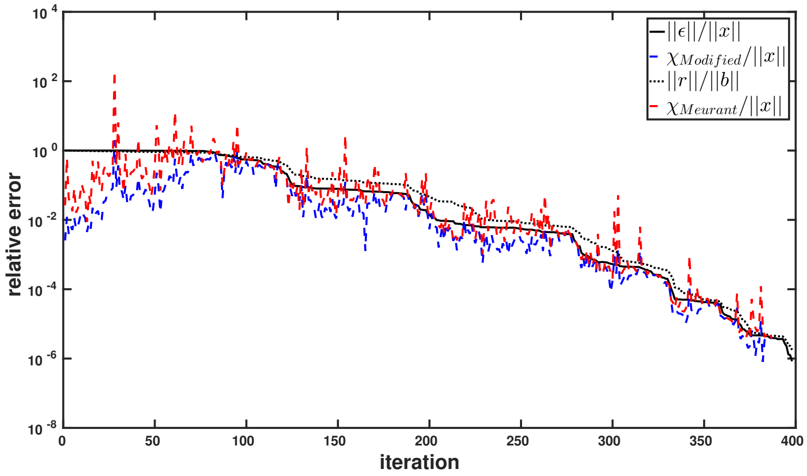


Fig. 2.1. *Convergence plot showing comparison of different relative error estimates and true relative error for randomly generated nonsymmetric positive definite matrix ($\kappa(A) \approx 10^6$) with random right hand side vector b. (d = 10). Note that undershoots of the modified estimator (blue) are much smaller than the overshoots of the original estimator (red), and the logarithmic scale should be noted.*

It was also seen that GMRES error estimator even after correction may behave erratically when numerical precision of computing system is exhausted as seen in figure 2.2. The exhaustion of numerical precision can lead to near singularity of Hessenberg matrix H_k formed during the Arnoldi iteration which can cause large errors in H_k^{-1} and its functions. Note that $\|s_k\|$ is also a function of H_k^{-1} and can be used as trigger to predict this exhaustion of numerical precision as seen in figure 2.2 making this error estimate a robust stopping criterion.

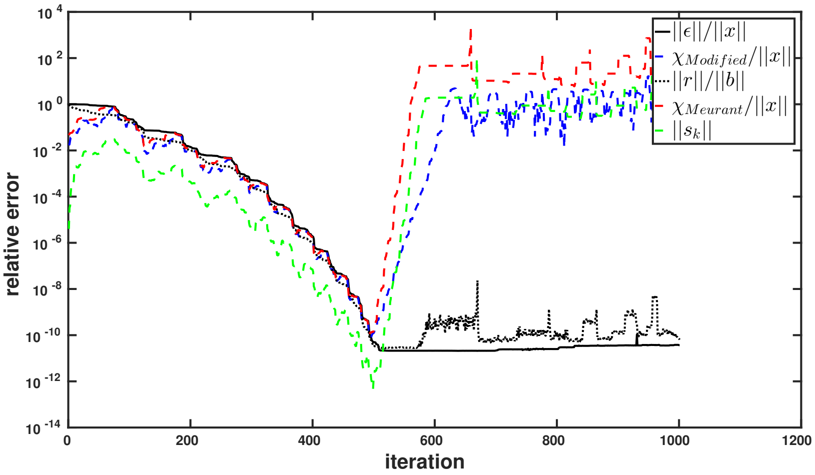


Fig. 2.2. *Behaviour of relative error estimates in GMRES on symmetric positive definite matrix ($\kappa(A) = 10^6$) when the numerical precision is exhausted. The term $\|s_k\|$ can be used as trigger to improve the estimator and for detection of exhaustion of numerical precision.*

2.2. BiCGQL: Proposed estimator for A-norm and l_2 norm of errors in Bi-Conjugate gradient Algorithm (BiCG).

Similar to the CGQL for CG, A-norm of error in this case can be represented in terms of residual vector of BiCG algorithm and tridiagonal matrices produced by a non-symmetric version of the Lanczos algorithm. A-norm of error (which we better denote as A-measure for matrices which are not positive definite) when matrix is non-symmetric, is given by:

$$\|\epsilon_k\|_A^2 = \epsilon_k^T A \epsilon_k = r_k^T (A^T)^{-1} r_k = r_k^T A^{-1} r_k \tag{2.7}$$

And, when $A \in \mathbb{R}^{N \times N}$ and $r \in \mathbb{R}^N$; $r_k^T(A^T)^{-1}r_k$ is a scalar quantity whose transpose will be itself and thus $r_k^T(A^T)^{-1}r_k = r_k^T A^{-1} r_k$. Here, r is the residual vector pertaining to the BiCG method. When A is positive definite, the right side of the above equation is always positive. In case of indefinite matrices, the absolute value of the above equation is considered and we define such an A-measure of the error in these cases. Moreover the l_2 norm of error is given by:

$$\|\epsilon_k\|_2^2 = \epsilon_k^T \epsilon_k = r_k^T (A^T)^{-1} A^{-1} r_k \tag{2.8}$$

If $A = A^T$, A-norm and l_2 norm of error is given by $r_k^T A^{-1} r_k$ and $r_k^T A^{-2} r_k$ respectively. We are interested in approximating (2.7) and (2.8). In the following sections we derive approximation of A-norm and l_2 norm of error at every iteration of BiCG. BiCG is included as Algorithm 1 in the appendix for reference.

2.2.1. $\mathcal{O}(n)$ expression to estimate A-norm of error. Writing the consecutive difference between A-norm of error at iteration k and $k+1$ we get the following relation when A is a non-symmetric matrix (See Appendix for BiCG Algorithm):

(2.9)
$$\begin{aligned} r_k^T(A^T)^{-1}r_k - r_{k+1}^T(A^T)^{-1}r_{k+1} &= r_k^T(A^T)^{-1}r_k - (r_k - \alpha_k A p_k)(A^T)^{-1}(r_k - \alpha_k A p_k) \\ &= -\alpha_k r_k^T p_k - \alpha_k r_k^T (A^T)^{-1} A p_k + \alpha_k^2 p_k^T A p_k \end{aligned}$$

In the above equation the first and third terms of R.H.S can be trivially computed using iterates of the Bi-CG Algorithm. Second term involves computation of A^{-1} hence we further reduce it. As $r_{k+1} = r_k - \alpha_k A p_k$ and $Ap_k = \dfrac{r_k - r_{k+1}}{\alpha_k}$ we can derive the following relation:

$$\begin{aligned} r_k^T A^{-1} r_k - r_{k+1}^T A^{-1} r_{k+1} &= \alpha_k r_k^T p_k - \alpha_k r_k^T (A^T)^{-1} (\frac{r_k - r_{k+1}}{\alpha_k}) - \alpha_k^2 p_k^T A p_k \\ \implies r_{k+1}^T A^{-1} r_{k+1} &= -\alpha_k r_k^T p_k + r_{k+1}^T A^{-1} r_k + \alpha_k^2 p_k^T A p_k \end{aligned} \tag{2.10}$$

Error ($\epsilon_k = x - x_k$) is given by $A^{-1} r_k$. Also, ϵ_k can be written as a weighted sum of search directions from k to n given below in Equation 2.11.

$$\epsilon_k = \sum_{j=k}^{n} \alpha_j p_j \tag{2.11}$$

If ϵ_{k+d} denotes the error at $k+d$ iteration then $\epsilon_{k+d} \ll \epsilon_k$ for some $d > 0$ (when error reduces with iterations) and we neglect the further terms of the series sum. We use the error vector in Equation 2.11 to give an estimate of A-norm of error after inducing a delay of d iterations. So Equation 2.10 now becomes:

$$r_{k+1}^T A^{-1} r_{k+1} \approx -\alpha_k r_k^T p_k + r_{k+1}^T (\sum_{j=k}^{k+d} \alpha_j p_j) + \alpha_k^2 p_k^T A p_k \tag{2.12}$$

It should be noted that when A is positive definite, the above expression is always positive and thus provides a lower bound for the square of A-norm of error. Note that the above is derived from Bi-CG, and when $A = A^T$, this estimator for A-norm is equivalent to the CGQL estimator for A-norm (Section 5.3 in Appendix). Moreover, RHS of Equation 2.12 is not a unique relation for evaluation of A-measure of error using Bi-CG iterates.

2.2.2. $\mathcal{O}(n)$ expression to estimate l_2 norm of error. Rewriting the consecutive difference between l_2 norm of error at iteration k and $k+1$ we get the following relation when A is a non-symmetric matrix:

$$\begin{aligned} r_k^T (A^T)^{-1} A^{-1} r_k - r_{k+1}^T (A^T)^{-1} A^{-1} r_{k+1} = r_k^T (A^T)^{-1} A^{-1} r_k \\ - (r_k - \alpha_k A p_k)(A^T)^{-1} A^{-1} (r_k - \alpha_k A p_k) \end{aligned} \tag{2.13}$$

Rearranging Equation 2.13, we get:

$$r_k^T (A^T)^{-1} A^{-1} r_k - r_{k+1}^T (A^T)^{-1} A^{-1} r_{k+1} = 2\alpha_k p_k^T A^{-1} r_k - \alpha_k^2 \|p_k\|^2 \tag{2.14}$$

Using 2.11 and $\epsilon_k = A^{-1} r_k$ we can rewrite above as

$$r_{k+1}^T (A^T)^{-1} A^{-1} r_{k+1} \approx -2\alpha_k p_k^T (\sum_{j=k+1}^{k+d} \alpha_j p_j) + \| \sum_{j=k+1}^{k+d} \alpha_j p_j \|^2 \tag{2.15}$$

Here, d signifies the delay in approximation. Also, BiCG method shows irregular convergence, and in such cases larger values of d can result in less accurate approximations. Hence, values of $d < 10$ is recommended which is much less than N resulting in an efficient estimation. Note that by $\mathcal{O}(n)$ we mean arithmetic complexity. Lemma 2.1 further justifies the arithmetic complexity for the above expressions.

Lemma 2.1. *Equation 2.12 and 2.15 involve arithmetic operations of $\mathcal{O}(n)$.*

Proof. Equation 2.12 is an approximation to A norm of error at $k+1$ iteration and it involves three terms. All the terms involve inner products of vectors in $\mathbb{R}^n$ and require n multiplication operations; referred to as $\mathcal{O}(n)$ arithmetic operations here. Note that the third term involves the matrix-vector product Ap_k which is provided by the Bi-CG Algorithm at every iteration with no additional cost to this estimator. Similarly, the moving-window summation in the second term involves a single inner product at every iteration. Thus estimation of A norm of error in BiCG algorithm requires $\mathcal{O}(3n)$ arithmetic operations in total. Here d denotes the delay in estimation where α_k is a scalar.

Similarly, evaluation of Equation 2.15 requires two inner products in the moving-window summations and result in $\mathcal{O}(2n)$ operations.
□

3. Numerical results and an analysis of estimators.

3.1. A-norm and l_2 norm estimators for Bi-CG. In Figure 3.1 plot of estimator along with A-norm of the error is shown when A is a Non-symmetric matrix.

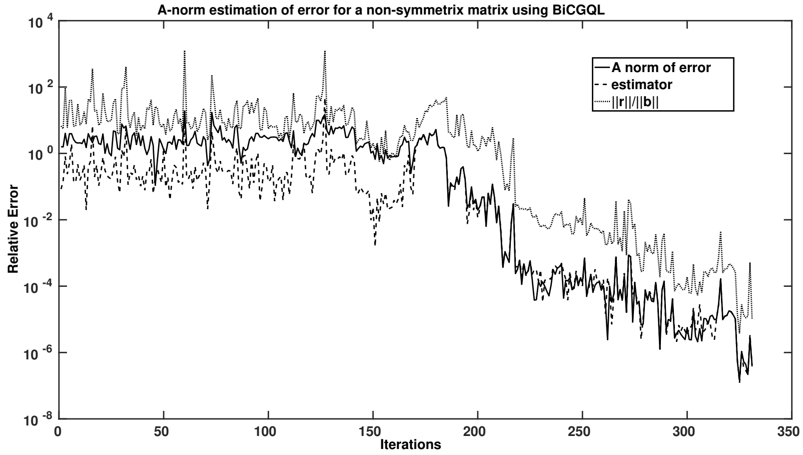


FIG. 3.1. *BiCGQL estimator for a Non-symmetric matrix (indefinite); absolute values are considered for $r^TA^{-1}r$ and its approximation; dimension of the matrix* $= 500 \times 500$*; condition number of the matrix is* 10^6 *and* $d = 10$

Figure 3.2 shows the comparison between l_2 norm approximation, actual l_2 norm of the error and l_2 norm of the residue.

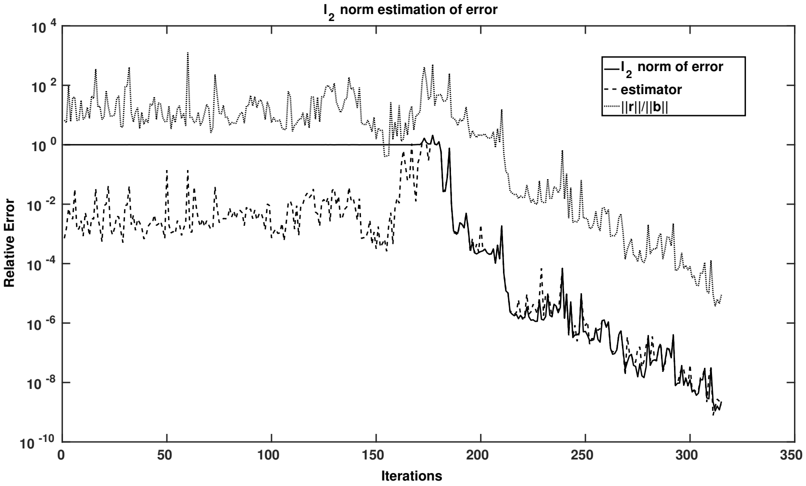


FIG. 3.2. *Comparison between BiCGQL l_2 norm of estimator, actual l_2 norm of the error and l_2 norm of the residue; dimension of the matrix $= 500 \times 500$; condition number of the matrix is 10^6 and $d = 10$*

It is evident that BiCGQL estimators work efficiently both cases. In both the figures we see that the error norm is also more stable than the norm of the residue. Similar behaviour can be seen when the matrix A is Non-symmetric positive definite, and the convergence is faster and stable as compared to indefinite cases.

3.2. Analysis of estimators.

3.2.1. Condition number of the problem. Condition number is useful in matrix computations as they enable us to estimate the accuracy of computed result. Condition number of a forward problem (that is computing b given A and x) and backward problem (computing x from A and b) respectively are:

$$\kappa(A, x) = \|A\| \frac{\|x\|}{\|Ax\|} \tag{3.1}$$

$$\kappa(A, b) = \|A^{-1}\| \frac{\|b\|}{\|A^{-1}b\|} \tag{3.2}$$

Condition number of the matrix is given by the product of $\kappa(A, x)$ and $\kappa(A, b)$ and is given by:

$$\kappa = \|A\|\|A^{-1}\| \tag{3.3}$$

3.2.2. Evaluation metrics. Equation 3.1 and 3.2 relate to the condition number of the forward and backward problem respectively when one solves for a linear system. In order to test the estimator, we choose the relative error in estimating norm (l_2-norm or A-norm) of error by the estimator as an uncertainty metric of estimator. This metric for k^{th} iteration can be expressed as follows:

$$\left|\frac{\dfrac{\chi_k}{\|x\|} - \dfrac{\|\epsilon_k\|}{\|x\|}}{\dfrac{\|\epsilon_k\|}{\|x\|}}\right| \tag{3.4}$$

where χ_k is the estimate of norm of error at k^{th} iteration. However, we are more interested in comparing the estimator and relative residual in order to come up with robust stopping criterion in krylov subspace based algorithms. Also the metric should consider all iterations on which we could measure the uncertainties, Hence, we define uncertainty ratio $U.R.^{(j)}$ of j^{th} order as Performance or Uncertainty metric in estimating the norm of error as follows:

$$U.R.^{(j)} = \frac{1}{n-d} \sum_{k=0}^{n-d-1} \left(\left| \frac{\dfrac{\|r_k\|^j}{\|b\|^j} - \dfrac{\|\epsilon_k\|^j}{\|x\|^j}}{\dfrac{\chi_k^j}{\|x\|^j} - \dfrac{\|\epsilon_k\|^j}{\|x\|^j}} \right| \right) \tag{3.5}$$

where n is dimension of matrix and d is the delay in estimation. We consider j = 1 and 2 for subsequent analysis which are defined as follows:

$$U.R.^{(1)} = \frac{1}{n-d} \sum_{k=0}^{n-d-1} \left(\left| \frac{\dfrac{\|r_k\|}{\|b\|} - \dfrac{\|\epsilon_k\|}{\|x\|}}{\dfrac{\chi_k}{\|x\|} - \dfrac{\|\epsilon_k\|}{\|x\|}} \right| \right) \tag{3.6}$$

&

$$U.R.^{(2)} = \frac{1}{n-d} \sum_{k=0}^{n-d-1} \left(\left| \frac{\dfrac{\|r_k\|^2}{\|b\|^2} - \dfrac{\|\epsilon_k\|^2}{\|x\|^2}}{\dfrac{\chi_k^2}{\|x\|^2} - \dfrac{\|\epsilon_k\|^2}{\|x\|^2}} \right| \right) \tag{3.7}$$

According to 3.6 and 3.7, it can be seen that $U.R.^{(1)}$ and $U.R.^{(2)}$ are functions of matrix A and vectors b, x^0 and delay parameter d of estimator. Further we show why the condition number of forward problem (Equation 3.1) encapsulates the parameters of the problem in this estimation. Calculating $\|x\|$ is not trivial but we use the norm of x at each iterate (i.e $\|x_k\|$) in place of $\|x\|$ since $\|x_k\|$ converges to $\|x\|$ in first few iterations and doing so brings marginal changes in the estimation.

3.2.3. Theorem on Expectation of $U.R.^{(1)}$ and $U.R.^{(2)}$.

THEOREM 3.1. *For fixed singular values of non-singular square real matrix A and forward condition number $\kappa(A,x)$,*

$$E\left(U.R.^{(1)}\right) \geq \sqrt{\frac{8}{3}} f_{rc}(f_e)^d \left(\frac{\|A\|_F}{\sqrt{n}}\frac{\|x\|}{\|b\|}\right)\left(1+\frac{d}{n-d}\log(n-d)\right)$$

$$\&$$

$$E\left(U.R.^{(2)}\right) \geq f_{rc}^2(f_e^2)^d \left(\frac{\|A\|_F^2}{n}\frac{\|x\|^2}{\|b\|^2}\right)\left(1+\frac{d}{n-d}\log(n-d)\right)$$

where d is the delay in estimation, n is the dimension of the matrix A, $\kappa(A,x) \gg 1$, f_{rc} is the minimum relative uncertainty in residual with respect to error and f_e is the minimum convergence rate of error. The expectation is taken over the problems ($Ax = b$) satisfying above constraints.

Proof. This is the brief version of proof for the above theorem. The detailed version is given in the appendix 5.5.

Derivation for $E(U.R.^{(2)})$:
The $U.R.^{(2)}$ can be written as

$$U.R.^{(2)} = \frac{1}{n-d}\sum_{k=0}^{n-d-1}\left|\frac{\dfrac{\|r_k\|^2}{\|\epsilon_k\|^2}\dfrac{\|x\|^2}{\|b\|^2}-1}{\dfrac{\chi_k^2}{\|\epsilon_k\|^2}-1}\right| \tag{3.8}$$

The direction of error vector ϵ_k can be considered as randomly uniform on the surface of n-dimensional sphere. Hence, we have,

$$E\left(\frac{\|r_k\|^2}{\|\epsilon_k\|^2}\right) = E\left(\frac{\|A\epsilon_k\|^2}{\|\epsilon_k\|^2}\right) = \frac{\|A\|_F^2}{n}$$

As $\kappa(A,x)$ is fixed, the quantity $\kappa_F(A,x) = \dfrac{\|A\|_F}{\sqrt{n}}\dfrac{\|x\|}{\|b\|}$ is also fixed due to the following inequality of matrices.

$$\frac{\|A\|}{\sqrt{n}} \leq \frac{\|A\|_F}{\sqrt{n}} \leq \|A\|$$

Since, $\kappa(A,x) \gg 1$, we can write,

$$E\left(U.R.^{(2)}\right) \approx \frac{\kappa_F^2(A,x)}{n-d}\sum_{k=0}^{n-d-1}\frac{1}{E\left(\left|\dfrac{\chi_k^2}{\|\epsilon_k\|^2}-1\right|\right)}$$

For delay-based estimators, $\chi_k^2 \approx \|\epsilon_k\|^2 - \|\epsilon_{k+d}\|^2$ which implies

$$\frac{\chi_k^2}{\|\epsilon_k\|^2} - 1 \approx -\frac{\|\epsilon_{k+d}\|^2}{\|\epsilon_k\|^2}$$

Therefore,

$$E\left(U.R.^{(2)}\right) \approx \frac{\kappa_F^2(A,x)}{n-d} \sum_{k=0}^{n-d-1} \frac{1}{E\left(\dfrac{\|\epsilon_{k+d}\|^2}{\|\epsilon_k\|^2}\right)} \approx \frac{\kappa_F^2(A,x)}{n-d} \sum_{k=0}^{n-d-1} \frac{1}{\dfrac{E(\|\epsilon_{k+d}\|^2)}{E(\|\epsilon_k\|^2)}}$$

$$E\left(U.R.^{(2)}\right) = \frac{\kappa_F^2(A,x)}{n-d} \sum_{k=0}^{n-d-1} \frac{E(\|\epsilon_k\|^2)}{E(\|\epsilon_{k+d}\|^2)}$$

At the k^{th} iteration, the computed solution lies in k-dimensional krylov subspace (for nonsingular matrix A). Hence, the error vector ϵ_k lies in $n-k$ dimensional subspace with gaussian distribution. We can write,

$$E(\|\epsilon_k\|^2) = E(c_1^2 + c_2^2 + ... + c_{n-k}^2) \approx n-k$$

Thus,

$$E\left(U.R.^{(2)}\right) \approx \frac{\kappa_F^2(A,x)}{n-d} \sum_{k=0}^{n-d-1} \frac{n-k}{n-k-d} \approx \frac{\kappa_F^2(A,x)}{n-d} \sum_{k=0}^{n-d-1} \left(1 + \frac{d}{n-k-d}\right)$$

$$E\left(U.R.^{(2)}\right) \approx \left(\frac{\|A\|_F^2}{n} \frac{\|x\|^2}{\|b\|^2}\right) \left(1 + \frac{d}{n-d} \log(n-d)\right) \tag{3.9}$$

Derivation for $E(U.R.^{(1)})$:

The random variable $\dfrac{\|r_k\|}{\|\epsilon_k\|}\dfrac{\|x\|}{\|b\|}$ has well defined bounds as follows:

$$\frac{\kappa(A,x)}{\kappa} \leq \frac{\|r_k\|}{\|\epsilon_k\|}\frac{\|x\|}{\|b\|} \leq \kappa(A,x)$$

We can assume that $\dfrac{\|r_k\|}{\|\epsilon_k\|}\dfrac{\|x\|}{\|b\|}$ follows a triangular distribution with mode at unity. The triangular distribution for every k is static in nature and is justified by gaussian assumption on error vector. The gaussian distribution has well known property that linear combination of independent gaussian variables is gaussian. As $\kappa(A,x) >> 1$, the triangular distribution on term $\dfrac{\|r_k\|}{\|\epsilon_k\|}\dfrac{\|x\|}{\|b\|}$ is highly skewed.

For such a skewed triangular distribution, we can write,

$$E\left(\frac{\|r_k\|^2}{\|\epsilon_k\|^2}\frac{\|x\|^2}{\|b\|^2}\right) \approx \frac{3}{2} E\left(\frac{\|r_k\|}{\|\epsilon_k\|}\frac{\|x\|}{\|b\|}\right)^2$$

Thus,

$$E\left(\frac{\|r_k\|}{\|\epsilon_k\|}\frac{\|x\|}{\|b\|}\right) \approx \sqrt{\frac{2}{3}}\left(\frac{\|A\|_F}{\sqrt{n}}\frac{\|x\|}{\|b\|}\right)$$

Since, $\kappa(A, x) \gg 1$, we can write,

$$E\left(\left|\frac{\|r_k\|}{\|\epsilon_k\|}\frac{\|x\|}{\|b\|} - 1\right|\right) \approx \sqrt{\frac{2}{3}}\left(\frac{\|A\|_F}{\sqrt{n}}\frac{\|x\|}{\|b\|}\right) \qquad (3.10)$$

Consider,

$$\psi_k = \frac{\chi_k}{\|\epsilon_k\|} \approx \sqrt{1 - \frac{\|\epsilon_{k+d}\|^2}{\|\epsilon_k\|^2}}$$

It is required that $|\psi_k - 1| \to 0 \quad \forall k$ as $d \to n$, since it can be proven that estimate has zero error if $d = n$. It is also required that mode of ψ_k to be at 1 since χ_k is an estimate of $\|\epsilon_k\|$. Hence, we assume that $|\psi_k - 1|$ has exponential distribution with mean λ_k^{-1} i.e. $|\psi_k - 1| \sim \text{expo}(\lambda_k)$ (Here expo refers to exponential distribution). The parameter λ_k can be estimated by the knowledge of $E\left(\left|\frac{\chi_k^2}{\|\epsilon_k\|^2} - 1\right|\right)$ i.e. $E(|\psi_k^2 - 1|)$ by using the relation 3.11. The derivation of equation 3.11 can be found in appendix 5.7.

$$E(|\psi_k^2 - 1|) = \frac{2}{\lambda_k} + \left(1 + \frac{2}{\lambda_k} + \frac{2}{\lambda_k^2}\right)e^{-\lambda_k} \qquad (3.11)$$

Since,

$$E\left(|\psi_k^2 - 1|\right) = \frac{n-k-d}{n-k} = 1 - \frac{d}{n-k}$$

we have,

$$\frac{2}{\lambda_k} + \left(1 + \frac{2}{\lambda_k} + \frac{2}{\lambda_k^2}\right)e^{-\lambda_k} = 1 - \frac{d}{n-k}$$

If $\lambda_k > 1$,

$$\frac{2}{\lambda_k} \approx 1 - \frac{d}{n-k}$$

and

$$E\left(\left|\frac{\chi_k}{\|\epsilon_k\|} - 1\right|\right) = \frac{1}{\lambda_k} \approx \frac{1}{2}\left(1 - \frac{d}{n-k}\right)$$

The probability density function of ψ_k can now be computed only by the knowledge of $\frac{d}{n-k}$. The density function has mode at 1 which is the requirement of being the estimator and $\left|\frac{\chi_k}{\|\epsilon_k\|} - 1\right| \to 0$ in expectation as $d \to n$.

Now,

$$E\left(U.R.^{(1)}\right) \approx \frac{1}{n-d}\sum_{k=0}^{n-d-1}\frac{E\left(\left|\frac{\|r_k\|}{\|\epsilon_k\|}\frac{\|x\|}{\|b\|}-1\right|\right)}{E\left(\left|\frac{\chi_k}{\|\epsilon_k\|}-1\right|\right)}$$

$$\approx \sqrt{\frac{2}{3}}\left(\frac{\|A\|_F}{\sqrt{n}}\frac{\|x\|}{\|b\|}\right)\left(\frac{1}{n-d}\sum_{k=0}^{n-d-1}\frac{1}{\frac{1}{2}\left(1-\frac{d}{n-k}\right)}\right)$$

$$\approx \sqrt{\frac{8}{3}}\left(\frac{\|A\|_F}{\sqrt{n}}\frac{\|x\|}{\|b\|}\right)\left(\frac{1}{n-d}\sum_{k=0}^{n-d-1}1+\frac{d}{n-k-d}\right)$$

$$E\left(U.R.^{(1)}\right) \approx \sqrt{\frac{8}{3}}\left(\frac{\|A\|_F}{\sqrt{n}}\frac{\|x\|}{\|b\|}\right)\left(1+\frac{d}{n-d}\log(n-d)\right) \tag{3.12}$$

□

Lemma 3.2.

$$U.R.^{(1)} = O\left(\kappa(A,x)\left(\frac{1}{E^2}\right)^{\frac{d}{n}}\right)$$

&

$$U.R.^{(2)} = O\left(\kappa(A,x)^2\left(\frac{1}{E^2}\right)^{\frac{d}{n}}\right)$$

where E is the stopping tolerance on relative error.

Proof. Using Cauchy-Schwartz inequality, the $U.R.^{(1)}$ can be seen to satisfy equation 3.13.

$$U.R.^{(1)} \leq \frac{\kappa(A,x)}{n-d}\left(\sum_{k=0}^{n-d-1}\frac{1}{\left|1-\frac{\chi_k}{\|\epsilon_k\|}\right|}\right) \tag{3.13}$$

For delay-based estimators, $\frac{\chi_k}{\|\epsilon_k\|} \approx \sqrt{1-\frac{\|\epsilon_{k+d}\|^2}{\|\epsilon_k\|^2}}$.

Since, we are interested in upperbound, term $\frac{\|\epsilon_{k+d}\|^2}{\|\epsilon_k\|^2} << 1$ and error decreases exponentially. The positive definite matrices have exponential convergence rates and are considered as good matrices from convergence point of view. Hence, use of exponential convergence rates is justified for indefinite matrices in order to find upperbound on convergence rate. Therefore, we have,

$$\sqrt{1-\frac{\|\epsilon_{k+d}\|^2}{\|\epsilon_k\|^2}} \approx 1-\frac{1}{2}\frac{\|\epsilon_{k+d}\|^2}{\|\epsilon_k\|^2}$$

Let UB be upperbound on $U.R.^{(1)}$. Then,

$$UB \approx 2\left(\frac{\kappa(A,x)}{n-d}\right)\left(\sum_{k=0}^{n-d-1}\frac{\|\epsilon_k\|^2}{\|\epsilon_{k+d}\|^2}\right)$$

Now, we know that error can fall up to prescribed tolerance level only and we can assume that relative error at starting iteration is $O(1)$. Therefore,

$$\frac{\|\epsilon_k\|^2}{\|\epsilon_{k+d}\|^2} \approx \left(\frac{1}{E^2}\right)^{\frac{d}{n}} \tag{3.14}$$

Thus,

$$UB \approx 2\left(\kappa(A,x)\right)\left(\frac{1}{E^2}\right)^{\frac{d}{n}}$$

Hence,

$$U.R.^{(1)} = O\left(\kappa(A,x)\left(\frac{1}{E^2}\right)^{\frac{d}{n}}\right) \tag{3.15}$$

Again using Cauchy-Schwartz inequality, the $U.R.^{(2)}$ can be seen to satisfy equation 3.16.

$$U.R.^{(2)} \leq \frac{\kappa(A,x)^2}{n-d}\left(\sum_{k=0}^{n-d-1}\frac{1}{\left|1-\frac{\chi_k^2}{\|\epsilon_k\|^2}\right|}\right) \tag{3.16}$$

i.e.

$$U.R.^{(2)} \leq \frac{\kappa(A,x)^2}{n-d}\left(\sum_{k=0}^{n-d-1}\frac{\|\epsilon_k\|^2}{\|\epsilon_{k+d}\|^2}\right)$$

According to equation 3.14,

$$U.R.^{(2)} = O\left(\kappa(A,x)^2\left(\frac{1}{E^2}\right)^{\frac{d}{n}}\right) \tag{3.17}$$

□

The upperbounds 3.15 and 3.17 are used to study the trends of $U.R.^{(1)}$ and $U.R.^{(2)}$ with $\frac{d}{n}$ respectively in figures 3.6 and 3.7.

3.3. Experimental results on performance of error estimates as the stopping criteria.

The Dataset consists of 10000 problems of high condition numbers for dimensions of 100,250,500,1000 on which the performance metrics are measured. The convergence of krylov subspace based iterative methods mostly depend on the eigenvalue spectrum of a matrix when it is nearly normal as shown by Liesen et. al[4]. The uncertainties in the estimation of error by estimator or residual can also depend on the convergence behaviour apart from other parameters. Hence, we consider two kinds of non-symmetric matrices i.e. positive definite and general. The performance of the CG estimator is implicitly described by the Bi-CG estimator in the case of symmetric positive definite matrices.

Figure 3.3 show the linear increase in Uncertainty Ratio ($U.R.^{(1)}$) (for A-norm of error estimation) as the condition number of the forward problem increases when matrices are taken to be a general non-symmetric in the case of the Bi-CG estimator. Here each blue dot signifies the mean average value of uncertainty ratio (Equation 3.6) for a particular problem (i.e for a particular matrix A and a right hand side vector b with x_0 sampled randomly from surface of n-dimensional sphere).

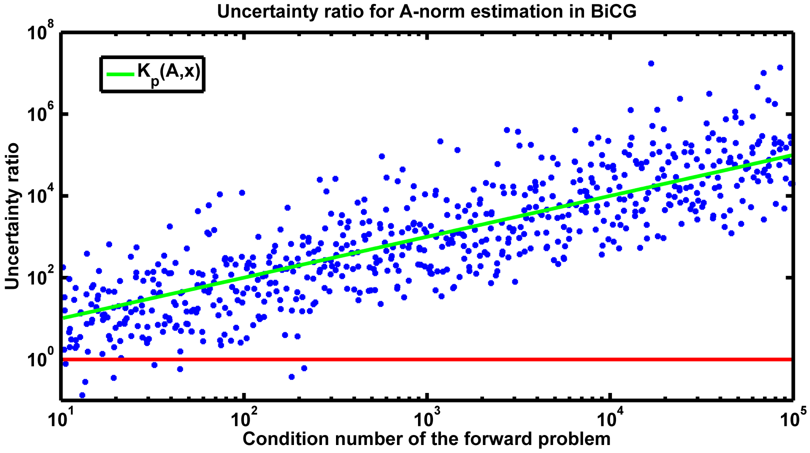
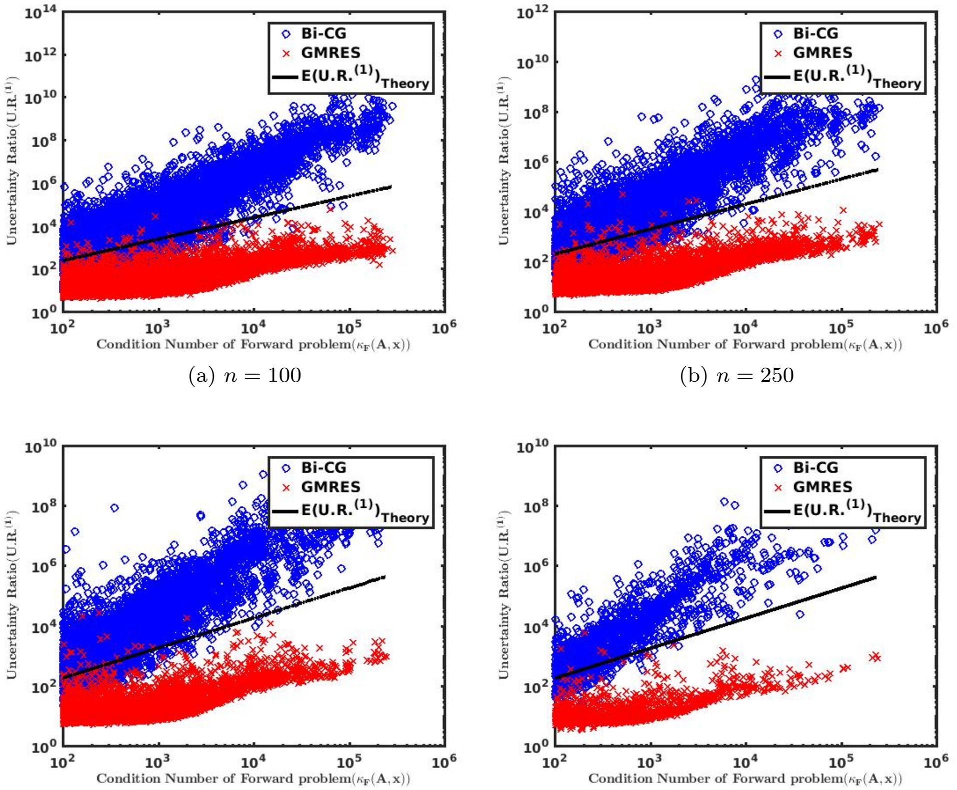


FIG. 3.3. *Linear increase in uncertainty ratio $U.R.^{(1)}$ for estimation of A-norm of error with increase in condition number of the problem when A is general non-symmetric, green line represents $\kappa(A,x)$ and red line represents the threshold line below which $\frac{\|r\|}{\|b\|}$ is a better estimator of error as compared to $\frac{\chi_k}{\|x\|}$*

Figures 3.5 and 3.4 shows the behaviour of Uncertainty Ratio ($U.R.^{(1)}$) for general nonsymmetric matrices and nonsymmetric positive definite matrices respectively. The linear trend with the forward condition number of the problem indicates the robustness of BiCGQL and GMRES error estimators with respect to relative residual in high condition number problems and also validates the theorem presented in section 3.1.



(a) $n = 100$

(b) $n = 250$

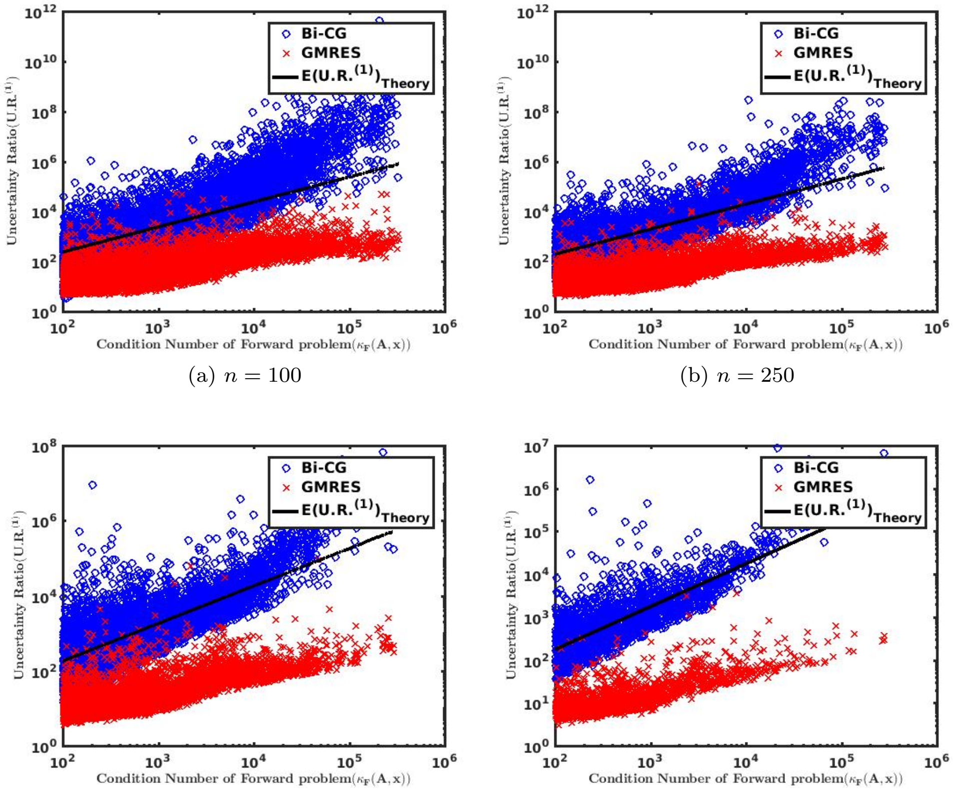
(c) $n = 500$

(d) $n = 1000$

FIG. 3.4. *Linear increase in uncertainty ratio ($U.R.^{(1)}$) for estimation of l_2 norm of error with increase in forward condition number of the problem ($\kappa_F(A,x)$) for nonsymmetric positive definite matrices of different dimensions with $d = 10$.*

3.3.1. Uncertainty Ratio and the delay parameter.

It is evident that increase in delay d will increase the estimator performance and in order to assess that performance experimentally, uncertainty ratio $U.R.^{(1)}$ (normalized by forward condition number based on frobenius norm) i.e. $\kappa_F(A,x) = \frac{\|A\|_F}{\sqrt{n}} \frac{\|x\|}{\|b\|}$ is averaged over 10000 problems for each d with dimension $n = 100$. The same experiment was done for $U.R.^{(2)}$ and results are shown in Figures 3.6 and 3.7. The experimental results are compared with theoretical results provided in 3.1 and 3.2.



(a) $n = 100$

(b) $n = 250$

(c) $n = 500$

(d) $n = 1000$

FIG. 3.5. *Linear increase in uncertainty ratio (U.R.$^{(1)}$) for estimation of l_2 norm of error with increase in forward condition number of the problem ($\kappa_F(A,x)$) for general nonsymmetric matrices of different dimensions with $d = 10$.*

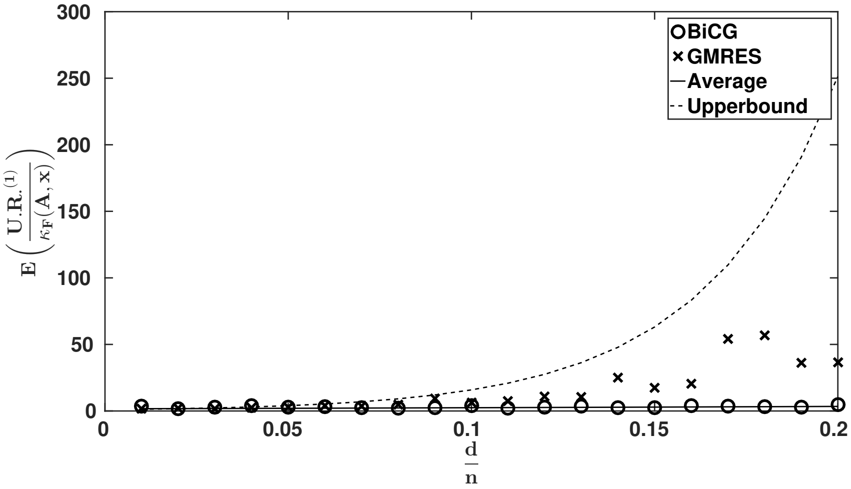


FIG. 3.6. *Comparison of theoretical and experimental results on behaviour of U.R.$^{(1)}$ with ratio of delay parameter to dimension of matrix $\left(\frac{d}{n}\right)$ for BiCG and GMRES algorithms.*

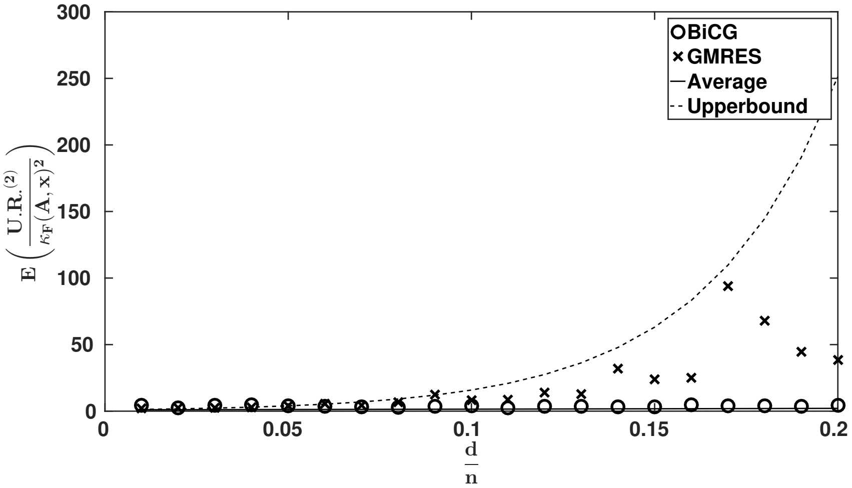


FIG. 3.7. *Comparison of theoretical and experimental results on behaviour of* $U.R.^{(2)}$ *with ratio of delay parameter to dimension of matrix* $\left(\frac{d}{n}\right)$ *for BiCG and GMRES algorithms.*

The dataset consists of non-symmetric positive definite matrices of high condition number of magnitude of 10^6. The results for both BiCG and GMRES algorithms are above the average line and this could be explained by the increased convergence rates due to positive definiteness of matrices. However, the Upperbound line tightly bounds the average $U.R.$ normalised by $\kappa_F(A,x)$ for tolerance level of 10^{-6}.

3.4. Computation saved with BiCGQL and GMRES estimators.

There can always be an under computation or an over computation involved for a desired relative error, when only the relative reside is used as the indicator of convergence. One might stop the iterations too early or deploy too many iterations due to this uncertainty in error in a solution, resulting in loss of efficiency or accuracy (see figures 3.8 and 3.9).

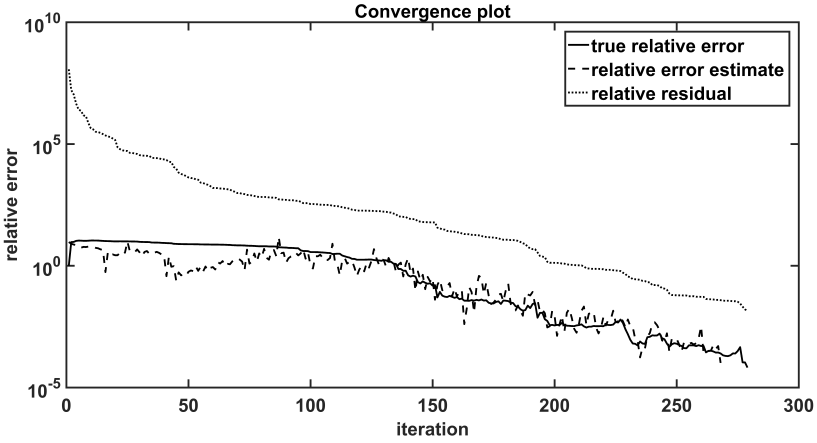


Fig. 3.8. *Illustration of over computation due to large uncertainty of relative residue leading to computation loss iterations for* 1000×1000 *nonsymmetric positive definite matrix with condition number having order of magnitude* 10^{11} *with random right hand side vector b. (d* $= 10$*)*

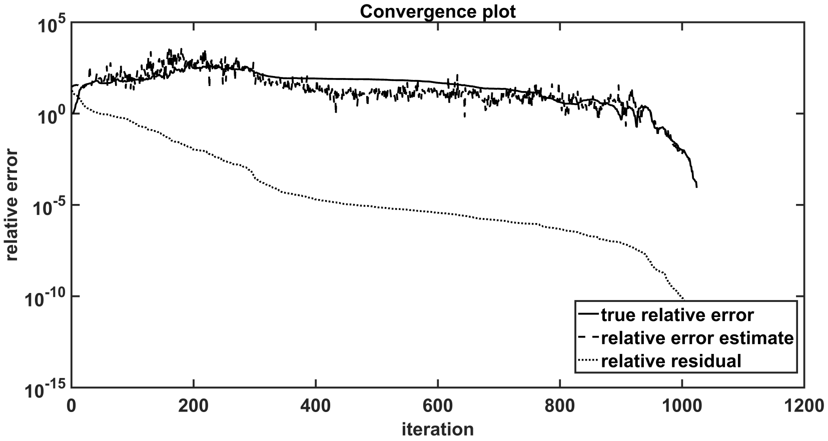


Fig. 3.9. *Illustration of under computation due to large uncertainty of relative residue leading to accuracy loss iterations for matrix 'sherman2' from matrix market provided along with its right hand side. (d* $= 10$*)*

Recall that the expected uncertainty ratio $U.R.^{(1)}$ is given by

$$E\left(U.R.^{(1)}\right) \approx \sqrt{\frac{8}{3}} \left(\frac{\|A\|_F}{\sqrt{n}} \frac{\|x\|}{\|b\|}\right) \left(1 + \frac{d}{n-d} \log(n-d)\right)$$

It can be observed that, if ratio $\dfrac{d}{n}$ is kept constant, the average value of $U.R.^{(1)}$ or $U.R.^{(2)}$ increases as $O(\log\ n)$ with dimension n of problem. The estimator becomes increasingly useful as the dimensionality of problem increases because of $O(n^2)$ cost of each iteration. Following analysis shows that for any given condition number of problem, this over and under computing scales with dimension of matrix as $O(n^3)$ i.e. loss in computing is of form $(\delta \times n) \times n^2$ where $0 < \delta < 1$. Note that our analysis showed only a weak (logarithmic) increase of uncertainty with dimension of the linear system, and we assume a dimension independent uncertainty $\kappa(A,x)$ in this section to derive a conservative estimate of the change in computation due to an estimator of error. The change in number of iterations deployed due to the estimator depends upon the convergence of relative error and relative residual, and also the stopping criterion used. Given an average uncertainty in convergence due to relative residue as the only indicator, we can estimate the under or over computing by averaging over stopping criteria for nominal types of convergence. Later we verify that this estimate is conservative, again using a large number of solutions of linear systems of manageable dimensions. We consider linear, super-linear and sub-linear convergence and estimate δ as follows; starting with expected ratio of the relative residue and error given earlier in 3.1.

$$\mathbb{E}(\frac{\|r_k\|\,\|x\|}{\|\epsilon_k\|\,\|b\|}) \approx \kappa(A,x) \tag{3.18}$$

For the following analysis, let $f_r(k) = \dfrac{\|r_k\|}{\|b\|}$ and $f_e(k) = \dfrac{\|\epsilon_k\|}{\|x\|}$.

Let the sub-linear convergence be of the form $f = Re^{-\alpha k}$ and super-linear convergence of the form $f = R(1 - e^{-\alpha(n-k)})$, where R represents the initial relative residue or the relative error respectively with corresponding convergence rates α_r and α_e. Here we assume that (3.18) is true for each type of convergence. To find a δ that satisfies

$$\mathbb{E}(\frac{f_r(k \pm \delta n)}{f_e(k)}) \approx 1 \tag{3.19}$$

Case (i): Sub-linear Convergence

Substituting a nominal model of sub-linear convergence of the form $f = Re^{-\alpha k}$, we divide (3.18) with (3.19) to get a corresponding estimate.

$$e^{\alpha_r(\pm\delta n)} \approx \kappa(A,x)$$

where $\frac{1}{n}\log(1/\epsilon_{mc}) \geq \alpha_r \geq \frac{1}{n}(\log(1/\epsilon_{mc}) - \log(\kappa(A,x)))$ for $+\delta$; and $\alpha_r \approx \frac{1}{n}\log(1/\epsilon_{mc})$ for $-\delta$ in the above equation. These correspond to the end conditions after full convergence, depending on whether $f_r(n) > f_e(n)$ or $f_r(n) \leq f_e(n)$. This gives us the conservative estimate of

$$\frac{\log(\kappa(A,x))}{\log(1/\epsilon_{mc})} \leq \delta \leq \frac{\log(\kappa(A,x))}{\log(1/\epsilon_{mc}) - \log(\kappa(A,x))}$$

Case (ii): Super-linear Convergence

Substituting a nominal model of super-linear convergence of the form $R(1 - e^{-\alpha(n-k)})$, we divide (3.18) with (3.19) to get an inequality

$$e^{2\alpha_r(\pm\delta n)} \leq \kappa(A,x)$$

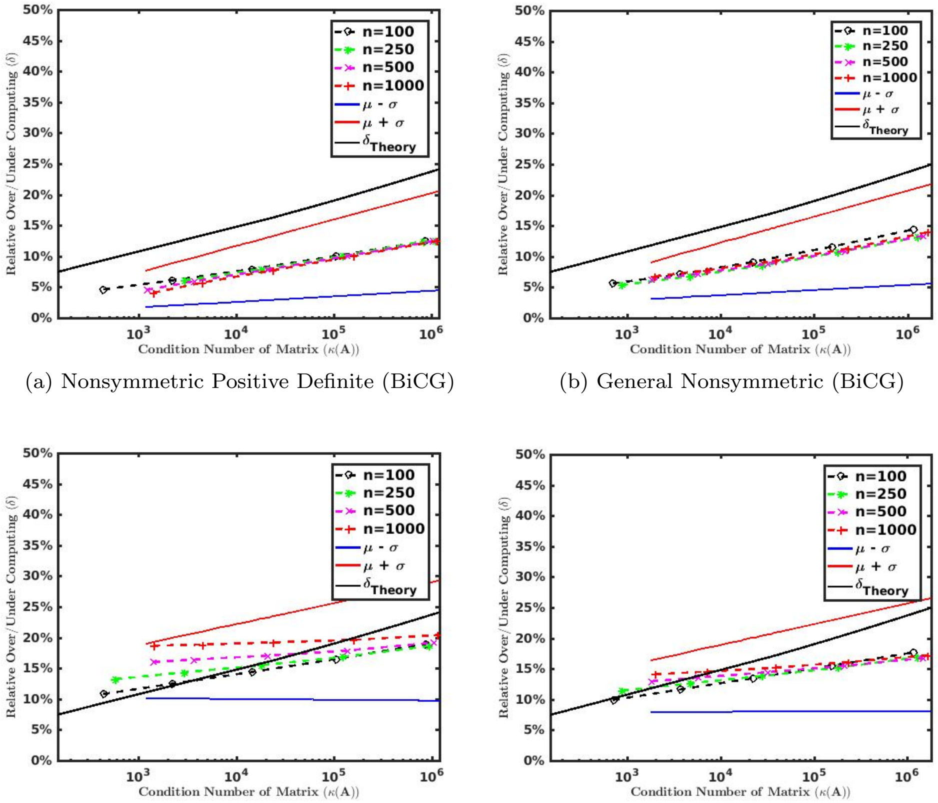
where $\frac{1}{n}\log(1/\epsilon_{mc}) \geq \alpha_r \geq \frac{1}{n}(\log(1/\epsilon_{mc}) - \log(\kappa(A,x)))$ for $+\delta$; and $\alpha_r \approx \frac{1}{n}\log(1/\epsilon_{mc})$ for $-\delta$ in the above. These correspond to the end conditions after full convergence, depending on whether $f_r(n) > f_e(n)$ or $f_r(n) \leq f_e(n)$. This gives us the conservative estimate of

$$\frac{\log(\kappa(A,x))}{2\log(1/\epsilon_{mc})} \leq \delta \leq \frac{\log(\kappa(A,x))}{2(\log(1/\epsilon_{mc}) - \log(\kappa(A,x)))}$$

While, the above analysis assumed a nominal smooth convergence, actual over and under computations can be larger. Thus, as a conservative indicator in recommending the use of error estimates, the following relation is useful.

$$\delta \sim \frac{\log(\kappa(A))}{2\log(1/\epsilon_{mc}) - \log(\kappa(A))} \tag{3.20}$$

where we have replaced $\kappa(A,x)$ with its field averaged value $\kappa(A)^{\frac{1}{2}}$.



(a) Nonsymmetric Positive Definite (BiCG)

(b) General Nonsymmetric (BiCG)

(c) Nonsymmetric Positive Definite (GMRES)

(d) General Nonsymmetric (GMRES)

FIG. 3.10. *Comparison of theoretical and experimental results on average relative over/under computation with condition number of matrix for different dimensions in BiCG and GMRES Algorithms on different datasets. The red and blue solid lines bounds the uncertainty region of the estimates of relative over/under computing with confidence level of one standard deviation. The black solid line shows the theoretical estimate of δ mentioned in equation 3.20.*

4. Conclusions.

The importance of error estimators for efficient stopping (or restarting) are clearly evident for problems with even moderately high condition number $\kappa > 100$, and is emphasized by numerical examples and an analysis of the expected uncertainty in convergence when using the norm of the residual vector. Results showing the reduction of uncertainty in convergence while using error estimators were presented for both Bi-CG and GMRES algorithms. It was also highlighted that the proposed estimator for Bi-CG converges to the earlier proposed estimator for the CG algorithm in the case of symmetric positive-definite matrices. Further, a simple analysis of the change in computation due to the estimators was presented along with numerical results of large number of problems of manageable dimensions. While Bi-CG has much larger uncertainties in convergence because of its oscillatory behavior (and this can be improved by using extended versions of the algorithm [1] [10] [9] [11] [2]), the actual under and over computations for any required error are similar for both these algorithms, as expected. Based on the results discussed in the previous sections, we conclude that the estimate for the A-norm or the l_2 norm of the error should be implemented into software realization of iterative solvers, instead of using only the relative norm of the residual vector as a criterion for stopping.

5. Appendix.

5.1. Bi-CG and its relation to Non-symmetric Lanczos algorithm.

Bi-Conjugate Gradient algorithm is an extension to CG algorithm which is used to solve a system of linear equations and works even for a Non-symmetric (possibly indefinite) matrix.

Algorithm 1 Bi-Conjugate Gradient Algorithm

```
procedure
    input A, $A^T$, b, $x_0$, $y_0$
  $r_0$ = b - A$x_0$
  $\tilde{r_0}$ = b - $A^T$ $y_0$
  $p_0 = r_0$
  $q_0 = \tilde{r_0}$
  for k = 1...until convergence do
      $\alpha_{k-1} = \frac{\tilde{r}^T_{k-1} r_{k-1}}{q^T_{k-1} A p_{k-1}}$
      $x_k = x_{k-1} + \alpha_{k-1} p_{k-1}$
      $y_k = y_{k-1} + \alpha_{k-1} q_{k-1}$
      $r_k = r_{k-1} - \alpha_{k-1} A p_{k-1}$
      $\tilde{r}_k = \tilde{r}_{k-1} - \alpha_{k-1} A^T q_{k-1}$
      $\beta_{k-1} = \frac{\tilde{r_k}^T r_k}{\tilde{r}^T_{k-1} r_{k-1}}$
      $p_k = r_k + \beta_{k-1} p_{k-1}$
      $q_k = \tilde{r_k} + \beta_{k-1} q_{k-1}$
  end for
end procedure
```

Bi-CG can also be derived from Non-symmetric Lanczos algorithm, for example considering v_1 and $\tilde{v_1}$ be the given starting vectors to Non-symmetric lancozs algorithm (such that $\|v_1\| = 1$ and $(v_1, \tilde{v_1}) = 1$), the two three term recurrences which help in forming two bi-orthogonal subspaces can be as follows:

For k=1,2....

$$\begin{aligned} z_k &= Av_k - w_k v_k - \eta_{k-1} v_{k-1} \\ \tilde{z_k} &= A^T \tilde{v_k} - w_k \tilde{v_k} - \tilde{\eta}_{k-1} \tilde{v}_{k-1} \end{aligned} \tag{5.1}$$

The coefficient w_k being computed as $w_k = (\tilde{v_k}, Av_k)$. The other coefficients η_k and $\tilde{\eta_k}$ are chosen (provided $(\tilde{z_k}, v_k) = 0$) such that $\eta_k \tilde{\eta_k} = (\tilde{z_k}, z_k)$ and the new vectors at step $k+1$ are given by:

$$v_{k+1} = \frac{z_k}{\eta_k}$$

(5.2)

$$\tilde{v}_{k+1} = \frac{\tilde{z_k}}{\tilde{\eta_k}}$$

The relationship between the bi-orthogonal subspaces and matrix A can be written in the form of a non-symmetric tri-diagonal matrix form (under the condition $\tilde{V_k}^T AV_k = T_k$) as:

$$T_k = \begin{bmatrix} \omega_1 & \eta_1 & 0 & \cdots & 0 \\ \tilde{\eta_1} & \omega_2 & \eta_2 & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \tilde{\eta}_{k-2} & \omega_{k-1} & \eta_{k-1} \\ 0 & \cdots & 0 & \tilde{\eta}_{k-1} & \omega_k \end{bmatrix} \tag{5.3}$$

5.2. GMRES error estimator. Let V_k be a matrix whose columns are orthonormal basis vectors $v_j,\ j = 1, ..., k$, of Krylov subspace $K_k(A, r_0)$, where r_0 is initial residual. The iterates of GMRES are defined as $x_k = x_0 + V_k z_k$ where z_k is vector of weights for each of orthonormal basis vectors v_j. We also have $H_k = V_k^T AV_k$ and $AV_n = V_n H_n$ with the assumption that Arnoldi process does not terminate early, that is, $h_{k+1,k} \neq 0$ for $k = 1, 2, ..., n-1$.

At k^{th} iteration, we have H_k that can be decomposed blockwise as:

$$H_k = \begin{pmatrix} H_{k-d} & W_{k-d} \\ Y_{k-d}^T & \tilde{H}_{k-d} \end{pmatrix}$$

Let

$$\gamma_{k-d} = \frac{h_{k-d+1,k-d}\big(e_{k-d}, H_{k-d}^{-1} e_1\big)}{1 - h_{k-d+1,k-d}\big(e_{k-d}, H_{k-d}^{-1} w_{k-d}\big)}$$

where $w_{k-d} = W_{k-d} \tilde{H}_{k-d}^{-1} e_1$.

Let the vector t_k be the last column of $\big(H_k^T H_k\big)^{-1}$, t_{kk} its last element and

$$\delta_{k+1} = \frac{h_{k+1,k}^2}{1 + h_{k+1,k}^2 t_{kk}}$$

and

$$u_k = \delta_{k+1} t_k$$

Hence error estimate denoted by χ_{k-d} at $(k-d)^{th}$ iteration in GMRES as provided by Meurant[5] is given by:

$$\frac{\chi^2_{k-d}}{\|r_0\|^2} = \gamma^2_{k-d}\left\|\tilde{H}^{-1}_{k-d}e_1\right\|^2 + \left\|\gamma_{k-d}H^{-1}_{k-d}w_{k-d} + (e_{k-d}, H^{-1}_{k-d}e_1)u_{k-d}\right\|^2 \tag{5.4}$$

5.3. Relating BiCGQL to CGQL . The difference between two consecutive A-norm of error (A-measure) in case of a Conjugate gradient Algorithm at iteration $'k'$ and $'k+1'$ can be given by:

$$\|\epsilon_k\|^2_A - \|\epsilon_{k+1}\|^2_A = \alpha_k r_k^T r_k \tag{5.5}$$

Hence by inducing a delay of $'d'$ iterations we can easily compute A-norm of error.

$$\begin{gathered} \|\epsilon_k\|^2_A - \|\epsilon_{k+d}\|^2_A = \textstyle\sum_{j=k}^{k+d} \alpha_j r_j^T r_j \\ \|\epsilon_k\|^2_A \approx \textstyle\sum_{j=k}^{k+d} \alpha_j r_j^T r_j \end{gathered} \tag{5.6}$$

For A-norm estimation of error in BiCGQL algorithm we derived the following results:

$$\|\epsilon_{k+1}\|^2_A = r_{k+1}^T A^{-1} r_{k+1} = -\alpha_k r_k^T p_k + r_{k+1}^T A^{-1} r_k + \alpha_k^2 p_k^T A p_k \tag{5.7}$$

By using $r_{k+1} = r_k - \alpha_k A p_k$ the above result can also be written as:

$$r_k^T A^{-1} r_k - r_{k+1}^T A^{-1} r_{k+1} = \alpha_k r_k^T p_k + \alpha_k r_k^T (A^T)^{-1} A p_k - \alpha_k^2 p_k^T A p_k \tag{5.8}$$

For a symmetric matrix $A = A^T$ the above equation can be further written as:

$$r_k^T A^{-1} r_k - r_{k+1}^T A^{-1} r_{k+1} = \alpha_k r_k^T p_k + \alpha_k r_k^T p_k - \alpha_k^2 p_k^T A p_k \tag{5.9}$$

For an algorithm like CG $p_i^T r_j = 0$ for $i \neq j$. Also $r_{k+1} = r_k - \alpha_k A p_k$ and thus substituting $Ap_k = \dfrac{r_k - r_{k+1}}{\alpha_k}$ in the last term we get:

$$\begin{gathered} r_k^T A^{-1} r_k - r_{k+1}^T A^{-1} r_{k+1} = \alpha_k r_k^T p_k + \alpha_k r_k^T p_k - \alpha_k p_k^T r_k + \alpha_k p_k^T r_{k+1} \\ \implies r_k^T A^{-1} r_k - r_{k+1}^T A^{-1} r_{k+1} = \alpha_k r_k^T p_k \\ \implies r_k^T A^{-1} r_k - r_{k+1}^T A^{-1} r_{k+1} = \alpha_k r_k^T r_k + \beta_{k-1} r_k^T p_{k-1} \\ \implies r_k^T A^{-1} r_k - r_{k+1}^T A^{-1} r_{k+1} = \alpha_k r_k^T r_k \end{gathered} \tag{5.10}$$

Equation 5.10 is equivalent to the A-norm estimation in CGQL Algorithm [3], and thus the proposed A-norm estimator for BiCG is equivalent to the CGQL estimator when $A = A^T$.

5.4. Relations between Non-Symmetric Lanczos Tridiagonal Matrix (T_k) and Residual vectors (r and $\tilde{r}$) of BiCG algorithm . A direct relationship between T_k , r_k and $\tilde{r_k}$ can be given by:

$$(T_n^{-1})_{(1,1)} = (T_k^{-1})_{(1,1)} + \frac{\tilde{r_k}^T A^{-1} r_k}{\|r_0\|^2} \tag{5.11}$$

$$(T_n^{-1})_{(1,1)} = (T_{k+1}^{-1})_{(1,1)} + \frac{\tilde{r}_{k+1}^T A^{-1} r_{k+1}}{\|r_0\|^2} \tag{5.12}$$

Subtracting 5.11 and 5.12 we get:

$$(T_{k+1}^{-1})_{(1,1)} = (T_k^{-1})_{(1,1)} + \frac{(\tilde{r_k}^T A^{-1} r_k - \tilde{r}_{k+1}^T A^{-1} r_{k+1})}{\|r_0\|^2} \tag{5.13}$$

As $\tilde{r_k}^T A^{-1} r_k - \tilde{r}_{k+1}^T A^{-1} r_{k+1} = \alpha_k \tilde{r}_k^T r_k$ for a Non-Symmetric matrix in BiCG:

$$(T_{k+1}^{-1})_{(1,1)} = (T_k^{-1})_{(1,1)} + \frac{(\alpha_k \tilde{r}_k^T r_k)}{\|r_0\|^2} \tag{5.14}$$

Thus knowing the relation between two consecutive T_k inverse first elements we can relate it with the estimator we developed for A-norm of error and the approach of CGQL Algorithm. However relating Non-Symmetric Lanczos and BiCG through Quadrature based methods will involve Complex Gaussian Quadratures ([8]). Hence we have followed an equivalent but direct approach of estimation using the relations of Bi-CG explicitly.

5.5. Proof for Theorem on Expectation of $U.R.^{(2)}$. This is the detailed proof for theorem mentioned in reference 3.1. The proof for Expectation of $U.R.^{(1)}$ is same as given in reference 3.1.

Proof.

Derivation for $E(U.R.^{(2)})$:
The $U.R.^{(2)}$ can be written as

$$U.R.^{(2)} = \frac{1}{n-d} \sum_{k=0}^{n-d-1} a_k \tag{5.15}$$

where

$$a_k = \left| \frac{\frac{\|r_k\|^2}{\|\epsilon_k\|^2} \frac{\|x\|^2}{\|b\|^2} - 1}{\frac{\chi_k^2}{\|\epsilon_k\|^2} - 1} \right| \tag{5.16}$$

In order to find $E(a_k)$ we deal with numerator and denominator of a_k separately. Now, consider the error at k^{th} iteration in GMRES or krylov-subspace based methods,

$$\epsilon_k = \epsilon_0 - V_k z_k \tag{5.17}$$

where V_k is the matrix representing basis of subspace of krylov vectors or search directions.

The error in krylov subspace based iterative algorithm generally decreases and convergence is guaranteed in atmost n iterations where n is dimension of matrix A. The basis V_k can atmost span k dimensions and if we average across the problems with constant forward condition number and singular values then the error vector ϵ_k on average will lie in n-k dimensional subspace provided ϵ_0 lies in entire n-dimensional space with any direction being equally probable. Hence, we can write,

$$\epsilon_k = c_1 v_{i_1} + c_2 v_{i_2} + ... + c_{n-k} v_{i_{n-k}} \tag{5.18}$$

&

$$r_k = c_1 \sigma_{i_1} u_{i_1} + c_2 \sigma_{i_2} u_{i_2} + ... + c_{n-k} \sigma_{i_{n-k}} u_{i_{n-k}} \tag{5.19}$$

where $(i_1, .., i_{n-k})$ is a single combination out of $\binom{n}{n-k}$ possible combinations and u_{i_k} and v_{i_k} are the corresponding left and right singular vectors of matrix A respectively.

Now, lets consider the problems where σ_{i_k} are fixed. In such a scenario,

$$E_{\sigma_{i_1},..,\sigma_{i_{n-k}}}\left(\frac{\|r_k\|^2}{\|\epsilon_k\|^2}\frac{\|x\|^2}{\|b\|^2}\right) = \left(\frac{\|A\|_F^2\,\|x\|^2}{\|b\|^2}\right)\left(\frac{1}{\|A\|_F^2}\right) E_{\sigma_{i_1},..,\sigma_{i_{n-k}}}\left(\frac{\|r_k\|^2}{\|\epsilon_k\|^2}\right)$$

The operator $E_{\sigma_{i_1},..,\sigma_{i_{n-k}}}$ will be replaced by $\hat{E}_{n-k}$ for following analysis.

$$\begin{aligned}
\hat{E}_{n-k}\left(\frac{\|r_k\|^2}{\|\epsilon_k\|^2}\frac{\|x\|^2}{\|b\|^2}\right) &= \left(\frac{\|x\|^2}{\|b\|^2}\right)\hat{E}_{n-k}\left(\frac{\left\|c_1\sigma_{i_1}u_{i_1} + ... + c_{n-k}\sigma_{i_{n-k}}u_{i_{n-k}}\right\|^2}{\left\|c_1 v_{i_1} + ... + c_{n-k}v_{i_{n-k}}\right\|^2}\right) \\
&= \left(\frac{\|x\|^2}{\|b\|^2}\right)\hat{E}_{n-k}\left(\frac{(c_1^2\sigma_{i_1}^2 + c_2^2\sigma_{i_2}^2 + ... + c_{n-k}^2\sigma_{i_{n-k}}^2)}{(c_1^2 + c_2^2 + ... + c_{n-k}^2)}\right)
\end{aligned}$$

As all directions in $n-k$ dimensional space are equally probable for error to point at, hence, the components follow gaussian distribution for such a vector and hence, $\hat{E}_{n-k}(c_i^2) = 1$. Hence by first order approximation on expectation of ratio,

$$\begin{aligned}
\hat{E}_{n-k}\left(\frac{\|r_k\|^2}{\|\epsilon_k\|^2}\frac{\|x\|^2}{\|b\|^2}\right) &\approx \left(\frac{\|x\|^2}{\|b\|^2}\right)\left(\frac{\hat{E}_{n-k}(c_1^2\sigma_{i_1}^2 + c_2^2\sigma_{i_2}^2 + ... + c_{n-k}^2\sigma_{i_{n-k}}^2)}{\hat{E}_{n-k}(c_1^2 + c_2^2 + ... + c_{n-k}^2)}\right) \\
&\approx \left(\frac{\|x\|^2}{\|b\|^2}\right)\left(\frac{(\sigma_{i_1}^2 + \sigma_{i_2}^2 + ... + \sigma_{i_{n-k}}^2)}{n-k}\right)
\end{aligned}$$

However, to find total expectation, we should consider all possible combinations of singular values to be equally probable and thus,

$$E\left(\frac{\|r_k\|^2}{\|\epsilon_k\|^2}\frac{\|x\|^2}{\|b\|^2}\right) \approx \left(\frac{\|x\|^2}{\|b\|^2}\right)\frac{1}{\binom{n}{n-k}}\sum_{(i_1,..,i_{n-k})\in S}\left(\frac{(\sigma_{i_1}^2 + \sigma_{i_2}^2 + ... + \sigma_{i_{n-k}}^2)}{n-k}\right)$$

where S is index set containing all $\binom{n}{n-k}$ combinations.

Now, for each combination $(i_1, .., i_{n-k})$, we can write,

$$\sigma_{i_1}^2 + \sigma_{i_2}^2 + ... + \sigma_{i_{n-k}}^2 = \|A\|_F^2 - (\sigma_{i_n}^2 + \sigma_{i_{n-1}}^2 + ... + \sigma_{i_{n-k+1}}^2)$$

Thus,

$$E\left(\frac{\|r_k\|^2}{\|\epsilon_k\|^2}\frac{\|x\|^2}{\|b\|^2}\right) \approx \left(\frac{\|x\|^2}{\|b\|^2}\right)\frac{1}{(n-k)}\left(\|A\|_F^2 - \frac{\sum_{(i_1,..,i_k)\in S'}\left(\sigma_{i_1}^2 + \sigma_{i_2}^2 + ... + \sigma_{i_k}^2\right)}{\binom{n}{k}}\right)$$

where S' is the complementary index set which contains all $\binom{n}{k}$ combinations.

In this sum $\sum_{(i_1,..,i_k)\in S'}\left(\sigma_{i_1}^2 + \sigma_{i_2}^2 + ... + \sigma_{i_k}^2\right)$, if σ_{i_1} is fixed to particular singular value, then there are $\binom{n-1}{k-1}$ possible combinations for choosing $\sigma_{i_2}, .., \sigma_{i_k}$. Since σ_{i_1} can be any one of the n singular values, the above sum can be written as

$$\sum_{(i_1,..,i_k)\in S'}\left(\sigma_{i_1}^2 + \sigma_{i_2}^2 + ... + \sigma_{i_k}^2\right) = \|A\|_F^2\binom{n-1}{k-1}$$

Therefore,

$$E\left(\frac{\|r_k\|^2}{\|\epsilon_k\|^2}\frac{\|x\|^2}{\|b\|^2}\right) \approx \left(\frac{\|x\|^2}{\|b\|^2}\right)\frac{1}{(n-k)}\left(\|A\|_F^2 - \frac{\|A\|_F^2\binom{n-1}{k-1}}{\binom{n}{k}}\right)$$

$$E\left(\frac{\|r_k\|^2}{\|\epsilon_k\|^2}\frac{\|x\|^2}{\|b\|^2}\right) \approx \left(\frac{\|A\|_F^2\|x\|^2}{\|b\|^2}\right)\frac{1}{(n-k)}\left(1-\frac{k}{n}\right) \approx \frac{\|A\|_F^2}{n}\frac{\|x\|^2}{\|b\|^2} \tag{5.20}$$

Consider

$$\hat{Q} = \frac{\|r_k\|^2}{\|\epsilon_k\|^2}\frac{\|x\|^2}{\|b\|^2} \quad \forall k$$

It can be seen that the random variable $\hat{Q}$ has well defined bounds as per Eqn. 5.21 which can be derived using Cauchy-Schwartz inequality:

$$\frac{\kappa(A,x)^2}{\kappa^2} \leq \hat{Q} \leq \kappa(A,x)^2 \tag{5.21}$$

let $a = \dfrac{\kappa(A,x)^2}{\kappa^2}$.
The lower bound a lies between 0 and 1 where as upperbound is much greater than 1. The random variable $\hat{Q}$ can be seen to follow the relation 5.22 (proof in section 5.6 of Appendix):

$$E\left(\left|\hat{Q}-1\right|\right) = E\left(\hat{Q}\right) - 1 + 2\int_a^1 (1-q)f_{\hat{Q}}(q)dq \tag{5.22}$$

Since, $E\left(\hat{Q}\right) \approx \frac{\|A\|_F^2}{n}\frac{\|x\|^2}{\|b\|^2} >> 1$,

$$E\left(\left|\hat{Q}-1\right|\right) \approx \frac{\|A\|_F^2}{n}\frac{\|x\|^2}{\|b\|^2} \tag{5.23}$$

From Eqn. 5.16,

$$E(a_k) \approx \frac{E\left(\left|\frac{\|r_k\|^2}{\|\epsilon_k\|^2}\frac{\|x\|^2}{\|b\|^2} - 1\right|\right)}{E\left(\left|\frac{\chi_k^2}{\|\epsilon_k\|^2} - 1\right|\right)}$$

For delay-based estimators, $\chi_k^2 \approx \|\epsilon_k\|^2 - \|\epsilon_{k+d}\|^2$ which implies

$$\frac{\chi_k^2}{\|\epsilon_k\|^2} - 1 \approx -\frac{\|\epsilon_{k+d}\|^2}{\|\epsilon_k\|^2}$$

Therefore,

$$E(a_k) \approx \frac{\frac{\|A\|_F^2}{n}\frac{\|x\|^2}{\|b\|^2}}{E\left(\frac{\|\epsilon_{k+d}\|^2}{\|\epsilon_k\|^2}\right)} \approx \frac{\frac{\|A\|_F^2}{n}\frac{\|x\|^2}{\|b\|^2}}{\frac{E(\|\epsilon_{k+d}\|^2)}{E(\|\epsilon_k\|^2)}}$$

Thus, According to Eqn. 5.18,

$$E(a_k) \approx \frac{\frac{\|A\|_F^2}{n}\frac{\|x\|^2}{\|b\|^2}}{\frac{E(c_1^2 + c_2^2 + ... + c_{n-k-d}^2)}{E(c_1^2 + c_2^2 + ... + c_{n-k}^2)}}$$

It can be stated that $E(c_i^2) = 1$ as averaging over all combinations of singular values will not change the average since every combination is equally probable.
Thus,

$$E(a_k) \approx \frac{\frac{\|A\|_F^2}{n}\frac{\|x\|^2}{\|b\|^2}}{\frac{n-k-d}{n-k}}$$

$$E(a_k) \approx \frac{\|A\|_F^2}{n}\frac{\|x\|^2}{\|b\|^2}\left(1 + \frac{d}{n-k-d}\right)$$

Thus,

$$E\left(U.R.^{(2)}\right) \approx \left(\frac{\|A\|_F^2}{n}\frac{\|x\|^2}{\|b\|^2}\right)\frac{1}{n-d}\sum_{k=0}^{n-d-1}\left(1 + \frac{d}{n-k-d}\right)$$

$$E\left(U.R.^{(2)}\right) \approx \left(\frac{\|A\|_F^2}{n}\frac{\|x\|^2}{\|b\|^2}\right)\left(1+\frac{d}{n-d}\log(n-d)\right) \tag{5.24}$$

□

5.6. Expectation of $\left|\hat{Q}-1\right|$.

The random variable $\hat{Q}$ takes values between a and b where $a \leq 1 \leq b$ and let $f_{\hat{Q}}(q)$ be the probability density function of $\hat{Q}$. In such case,

$$\begin{aligned} E\left(\left|\hat{Q}-1\right|\right) &= \int_a^b |q-1|\, f_{\hat{Q}}(q)dq \\ &= \int_a^1 (1-q)f_{\hat{Q}}(q)dq + \int_1^b (q-1)f_{\hat{Q}}(q)dq \\ &= 2\int_a^1 (1-q)f_{\hat{Q}}(q)dq + \int_a^b (q-1)f_{\hat{Q}}(q)dq \\ &= E(\hat{Q}) - 1 + 2\int_a^1 (1-q)f_{\hat{Q}}(q)dq \end{aligned}$$

5.7. Expectation of $\left|\psi_k^2-1\right|$.

We have,

$$|\psi_k - 1| \sim expo(\lambda_k)$$

The random variable ψ_k can take values from 0 to ∞. We assume that ψ_k is symmetrically distributed about 1 in range of 0 to 2. Thus, we can write,

$$P(1-a \leq \psi_k \leq 1+a) = 1 - e^{-\lambda_k a} \quad \forall a \in [0,1]$$

Also,

$$P(1-a \leq \psi_k \leq 1) = \frac{1}{2}\left(1 - e^{-\lambda_k a}\right) \quad \forall a \in [0,1]$$

and

$$P(1 \leq \psi_k \leq 1+a) = \frac{1}{2}\left(1 - e^{-\lambda_k a}\right) \quad \forall a \in [0,1]$$

and

$$\begin{aligned} P(2 \leq \psi_k \leq b) &= P(1 \leq |\psi_k - 1| \leq b-1) \\ &= e^{-\lambda_k} - e^{-\lambda_k(b-1)} \quad \forall b \in (2,\infty) \end{aligned}$$

Using the above equations, one can derive the cumulative distribution function and density function of ψ_k which are given respectively as follows.

$$\begin{aligned} F_k(y) &= \frac{1}{2}e^{-\lambda_k}\left(e^{\lambda_k y} - 1\right), \quad 0 \leq y \leq 1 \\ &= 1 - \frac{1}{2}e^{-\lambda_k}\left(1 + e^{-\lambda_k(y-2)}\right), \quad 1 \leq y \leq 2 \\ &= 1 - e^{-\lambda_k(y-1)}, \quad y \geq 2 \end{aligned}$$

$$\begin{aligned} f_k(y) &= \frac{\lambda_k}{2} e^{\lambda_k(y-1)}, \quad 0 \le y \le 1 \\ &= \frac{\lambda_k}{2} e^{-\lambda_k(y-1)}, \quad 1 \le y \le 2 \\ &= \lambda_k e^{-\lambda_k(y-1)}, \quad y \ge 2 \end{aligned}$$

Using this density function, one can easily derive an expression for $E\left(\left|\psi_k^2 - 1\right|\right)$ and it is given by

$$E(\left|\psi_k^2 - 1\right|) = \frac{2}{\lambda_k} + \left(1 + \frac{2}{\lambda_k} + \frac{2}{\lambda_k^2}\right) e^{-\lambda_k}$$

REFERENCES

[1] Roger Fletcher, *Conjugate gradient methods for indefinite systems*, in Numerical analysis, Springer, 1976, pp. 73–89.

[2] Roland W Freund and Noël M Nachtigal, *Qmr: a quasi-minimal residual method for non-hermitian linear systems*, Numerische mathematik, 60 (1991), pp. 315–339.

[3] Gene H Golub and Gérard Meurant, *Matrices, moments and quadrature ii; how to compute the norm of the error in iterative methods*, BIT Numerical Mathematics, 37 (1997), pp. 687–705.

[4] Jörg Liesen and Petr Tichỳ, *Convergence analysis of krylov subspace methods*, GAMM-Mitteilungen, 27 (2004), pp. 153–173.

[5] Gérard Meurant, *Estimates of the norm of the error in solving linear systems with fom and gmres*, SIAM Journal on Scientific Computing, 33 (2011), pp. 2686–2705.

[6] Gérard Meurant and Zdenek Strakoš, *The lanczos and conjugate gradient algorithms in finite precision arithmetic*, Acta Numerica, 15 (2006), pp. 471–542.

[7] Youcef Saad and Martin H Schultz, *Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on scientific and statistical computing, 7 (1986), pp. 856–869.

[8] Paul E Saylor and Dennis C Smolarski, *Why gaussian quadrature in the complex plane?*, Numerical Algorithms, 26 (2001), pp. 251–280.

[9] Gerard LG Sleijpen and Diederik R Fokkema, *Bicgstab (l) for linear equations involving unsymmetric matrices with complex spectrum*, Electronic Transactions on Numerical Analysis, 1 (1993), p. 2000.

[10] Gerard LG Sleijpen, Henk A Van der Vorst, and Diederik R Fokkema, *Bicgstab (l) and other hybrid bi-cg methods*, Numerical Algorithms, 7 (1994), pp. 75–109.

[11] Henk A Van der Vorst, *Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems*, SIAM Journal on scientific and Statistical Computing, 13 (1992), pp. 631–644.