SIGNIFICANCE OF ERROR ESTIMATION IN ITERATIVE SOLUTION OF LINEAR SYSTEMS: ESTIMATION ALGORITHMS AND ANALYSIS FOR CG, BI-CG AND GMRES
PUNEET JAIN, KRISHNA MANGLANI AND MURUGESAN VENKATAPATHI

Abstract. We show that estimation of error in the iterative solution can reduce uncertainty in convergence by a factor \( \sim \kappa(A,x) \) compared to the case of using the relative residue as a stopping criterion. Here \( \kappa(A,x) \) is the condition number of the forward problem of computing \( Ax \) given \( x, A, \) and \( 1 \leq \kappa(A,x) \leq \kappa(A) \) where \( \kappa(A) \) is the condition number of matrix \( A \). This makes error estimation as important as preconditioning for efficient and accurate solution of moderate to high condition problems \( (\kappa(A) > 10) \). 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. Note that computational cost of the estimator is expected to be significantly less than the \( \mathcal{O}(n^3) \) evaluation at every iteration of these methods in solving problems of dimension \( n \). In this work, we first 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. Secondly, we present an analysis of performance of these error estimates proposed for CG, Bi-CG and GMRES methods. 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. : Conjugate Gradients; Bi-CG; GMRES; error; stopping criteria; condition number

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 for a square matrix. 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 \( \left( \frac{\|r_k\|}{\|b\|} \right) \) 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, as 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 (denoted by \( \kappa \)) of the matrix is large \( (\kappa > 10^2) \), 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 \( \kappa \) 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 problems will be reduced and this is observed with matrices of larger dimensions. In cases where the condition number of the matrix is indeed reduced, a reduction of condition number of the backward problem (i.e. computing \( x \) given \( A \) and \( b \)) might not be guaranteed. Further more, a reduction in the
Significance of error estimation in iterative solution of linear systems

Condition number of the backward problem by an ideal preconditioner will accelerate convergence but still leave a large uncertainty in the relationship between the residue and the error, and 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 ($\kappa(A) > 10$), 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 [5]. 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 [4] 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 residue to the actual error can be more significant, i.e. as large as $\kappa(A)$ or as small as $1/\kappa(A)$.

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 (Algorithm 1 in appendix).

The idea of CGQL algorithm is to use CG instead of the Lanczos algorithm, to compute explicitly the entries of the corresponding tridiagonal matrices ($T_k$), and then to derive recursive formulas to compute the A-norm of error. The formulas are summarized as CGQL Algorithm 3 in the appendix (QL standing again for Quadrature and Lanczos), whose most recent version is described [2].

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

\begin{equation}
\|\epsilon_k\|_A^2 = \|r_0\|^2[(T_n^{-1})_{(1,1)} - (T_k^{-1})_{(1,1)}]
\end{equation}

Here $T_k$ is the tridiagonal matrix from Lanczos algorithm whose coefficients can be computed from Equation 5.9. Also $\epsilon_k, r_0$ are the error and residual respectively.
Let $d$ be a delay integer, the approximation of the A-norm of error at iteration $k - d$ is given by:

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

(2.2)

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. If $s_{k-1}$ be the estimate of $\|\epsilon_k\|_A$, for sufficiently large $k$ and $d = 1$ the estimator is given by:

$$
s_{k-1} = \|r_0\|^2 \frac{\eta_{k-1}^2 \epsilon_{k-1}}{\delta_k (\alpha_k \delta_{k-1} - \eta_k)}
$$

(2.3)

All above coefficients are related to CGQL algorithm as shown in the appendix (Algorithm 3).

2.1.2. Estimator for GMRES. Let $V_k$ be a matrix whose columns are orthonormal basis vectors $v_j$, $j = 1, \ldots, k$, of Krylov subspace $K_k(A,r^0)$. The iterates of GMRES are defined as $x_k = x_0 + V_k z_k$. We also have $H_k = V_k^T A V_k$ 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 $k = 1, 2, \ldots, n-1$. See appendix for a brief description of GMRES method and the algorithm for estimation of its error.

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

$$
H_k = \begin{pmatrix}
H_{k-d} & W_{k-d} \\
Y_{k-d} & H_{k-d}
\end{pmatrix}
$$

Let

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

where $w_{k-d} = W_{k-d} H_{k-d}^{-1} e_1$

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

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

and

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

Then, error estimates at $(k-d)^{th}$ iteration for GMRES as provided by Meurant[4] are given by:

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

(2.4)
2.1.3. Proposed modification to the GMRES error estimator. It can be seen that the estimator for GMRES proposed by Meurant [4] satisfies the equation (2.5).

\[
\chi^2_{k-d} \parallel r_0 \parallel^2 = \frac{\parallel e_{k-d} \parallel^2 - \parallel e_k \parallel^2}{\parallel r_0 \parallel^2} + \parallel s_k \parallel^2 + \left[ 2h_{k+1,k} (e_k, H_k^{-1} e_1) \left( (H_n^{-1} e_{k+1})^k, H_k^{-1} e^1 + s_k \right) 
- 2h_{k-d+1,k-d} (e_{k-d}, H_k^{-1} e^{1}) \left( (H_n^{-1} e_{k-d+1})^{k-d} - (H_k^{-1} e_{k-d+1})^{k-d}, H_k^{-1} e_1 + s_{k-d} \right) \right]
\]

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

The terms inside the square bracket are negligible compared to other terms, hence ignoring that, we get,

\[
\chi^2_{k-d} \parallel r_0 \parallel^2 \approx \parallel e_{k-d} \parallel^2 - \parallel e_k \parallel^2 + \parallel s_k \parallel^2
\]

Note that for large \( d \), term \( \parallel s_k \parallel^2 \) also becomes negligible. However, the delay \( d \) is kept small in practice (10), hence in such cases, term \( \parallel s_k \parallel^2 \) significantly contributes to the value of error estimate and needs to be accounted.

For converging problems, \( \parallel e_{k-d} \parallel^2 \gg \parallel e_k \parallel^2 \),

(2.6)

\[
\chi^2_{k-d} \approx \parallel e_{k-d} \parallel^2 + \parallel r_0 \parallel^2 \parallel s_k \parallel^2
\]

The equation (2.6) shows that the estimator is offset from exact value of error by term \( \parallel r_0 \parallel^2 \parallel s_k \parallel^2 \) and should be accounted in the formula of error estimator. This offset term can also cause unstable overshoots in estimation as seen in figure 2.1(b). This figure illustrates that original estimator can have very large errors in estimation due to term \( \parallel s_k \parallel^2 \) and removal of the term gives better results. Hence, we propose the modification in estimator with new formula given in equation (2.7).

(2.7)

\[
\chi^2_{k-d} \parallel r_0 \parallel^2 = \left( \gamma_{k-d} \parallel H_k^{-1} e_1 \parallel^2 + \parallel \gamma_{k-d} H_k^{-1} w_{k-d} + (e_{k-d}, H_k^{-1} e^1) u_{k-d} \parallel^2 - \parallel (e_k, H_k^{-1} e_1) u_k \parallel^2 \right)
\]

The absolute operation is necessary as in the non-converging situations, the operand can become negative. The comparison of the Meurant [4] estimator and proposed correction is demonstrated in the figure 2.1.
(a) orani678 matrix

(b) Positive definite matrix

Fig. 2.1. Convergence plot for two different non-symmetric matrices with random right hand side vectors $b$ and random initial solution $x_0$. Fig. 2.1(a) shows the comparison of different relative error estimates and true relative error for indefinite matrix "orani678" taken from matrix market. Fig. 2.1(b) shows the same comparison for the nonsymmetric positive definite matrix with condition number of $O(10^6)$.

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 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. Proposed estimator for $A$-norm and $l_2$ norm of errors in Bi-Conjugate gradient Algorithm.

Similar to the CGQL for CG, $A$-norm of error in this case can be represented
in term of residual vector of BiCG algorithm and tri-diagonal matrices of the corresponding Non-symmetric 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:

\[
\|\epsilon_k\|_A^2 = \epsilon_k^T A \epsilon_k = r_k^T (A^T)^{-1} r_k = r_k^T A^{-1} r_k
\]

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:

\[
\|\epsilon_k\|_2^2 = \epsilon_k^T \epsilon_k = r_k^T (A^T)^{-1} A^{-1} r_k
\]

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 \( \|\epsilon_k\|_A^2 \) and \( \|\epsilon_k\|_2^2 \). In the following sections we derive approximation of A-norm and \( l_2 \) norm of error for every iteration of BiCG iteration. The BiCG method is shown as Algorithm 4.

**2.2.1. \( O(n) \) 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:

\[
r_k^T (A^T)^{-1} r_k - r_{k+1}^T (A^T)^{-1} r_{k+1} = r_k^T (A^T)^{-1} r_k - (r_k - \alpha_k A p_k)(A^T)^{-1}(r_k - \alpha_k A p_k)
\]

\[
= -\alpha_k r_k^T p_k - \alpha_k r_k^T (A^T)^{-1} A p_k + \alpha_k^2 r_k^T A p_k
\]

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

\[
r_k^T A^{-1} r_k - r_{k+1}^T A^{-1} r_{k+1} = \alpha_k r_k^T p_k - \alpha_k r_k^T (A^T)^{-1}(r_k - r_{k+1}) - \alpha_k^2 r_k^T A p_k
\]

\[
\Rightarrow r_{k+1}^T A^{-1} r_{k+1} = -\alpha_k r_k^T p_k + r_k^T A^{-1} r_k + \alpha_k^2 r_k^T A p_k
\]

Error (\( \epsilon_k \)) is given as \( A^{-1} r_k \). Also, \( \epsilon_k = x - x_k \) and we know \( x_k \) but we do not know the final \( x \) and hence error at each iteration is difficult to compute. Also \( \epsilon_k \) is given as weighted sum of search directions from \( k \) to \( n \) given by Equation 2.12

\[
\epsilon_k = \sum_{j=k}^{n} \alpha_j p_j
\]

If \( \epsilon_k^{d+1} \) denotes the error at \( k + d \) iteration then \( \epsilon_k^{d+1} \ll \epsilon_k \) (assuming error falls as the iteration increases and \( d \geq 0 \)) so we can neglect the further terms of the series.
when since the above derivation is done using Bi-CG which is an extended version of CG, positive and thus provides a lower bound for the square of A-norm of error. Note that formula of evaluation, it provides a convenient way to use Bi-CG iterates.

\[ (2.13) \quad r_{k+1}^T A^{-1} r_{k+1} \approx -\alpha_k r_k^T p_k + r_{k+1}^T \left( \sum_{j=k}^{k+d} \alpha_j p_j \right) + \alpha_k^2 p_k^T A p_k \]

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 since the above derivation is done using Bi-CG which is an extended version of CG, when \( A = A^T \), our estimator for \( A \)-norm (Section 5.2 from Appendix). Though RHS of Equation (2.13) is not a unique formula of evaluation, it provides a convenient way to use Bi-CG iterates.

2.2.2. \( O(n) \) expression to estimate \( l_2 \) norm of error. Writing 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:

\[ (2.14) \quad r_k^T (A^T)^{-1} A^{-1} r_k - r_{k+1}^T (A^T)^{-1} A^{-1} r_{k+1} = r_k^T (A^T)^{-1} A^{-1} r_k \]
\[ \quad - (r_k - \alpha_k A p_k) (A^T)^{-1} A^{-1} (r_k - \alpha_k A p_k) \]

i.e.,

\[ (2.15) \quad r_k^T (A^T)^{-1} A^{-1} r_k - r_{k+1}^T (A^T)^{-1} A^{-1} r_{k+1} = \alpha_k^2 r_k^T (A^T)^{-1} p_k + \alpha_k p_k^T A^{-1} r_k - \alpha_k^2 p_k^T p_k \]

Taking transpose of first term of Equation (2.15) and rewriting the equation:

\[ (2.16) \quad r_k^T (A^T)^{-1} A^{-1} r_k - r_{k+1}^T (A^T)^{-1} A^{-1} r_{k+1} = 2\alpha_k p_k^T A^{-1} r_k - \alpha_k^2 \| p_k \|^2 \]

Rearranging Equation (2.16) we get:

\[ (2.17) \quad r_{k+1}^T (A^T)^{-1} A^{-1} r_{k+1} = -\alpha_k p_k^T A^{-1} r_k + \epsilon_k^T \epsilon_k - \alpha_k p_k^T A^{-1} r_k + \alpha_k^2 \| p_k \|^2 \]

Again using (2.12) we can estimate (2.18)

\[ (2.18) \quad r_{k+1}^T (A^T)^{-1} A^{-1} r_{k+1} \approx -2\alpha_k p_k^T \left( \sum_{j=k}^{k+d} \alpha_j p_j \right) + \| \sum_{j=k}^{k+d} \alpha_j p_j \|^2 + \alpha_k^2 \| p_k \|^2 \]

\[ \Rightarrow r_{k+1}^T (A^T)^{-1} A^{-1} r_{k+1} \approx -2\alpha_k p_k^T \left( \sum_{j=k+1}^{k+d} \alpha_j p_j \right) + \| \sum_{j=k+1}^{k+d} \alpha_j p_j \|^2 \]

Here, \( d \) again signifies the delay in approximation. Also, the Bi-CG method shows irregular convergence, in such cases larger values of \( d \) can result in less accurate approximations. Hence, values of \( d < 10 \) is recommended which are anyway much lesser than \( N \) for both accurate and efficient estimation. Note that by \( O(n) \) we mean arithmetic complexity. Lemma (2.1) further justifies the arithmetic complexity for the above expressions.
Lemma 2.1. Equations 2.13 and 2.18 involve arithmetic operations of $O(n)$.

Proof. Equation 2.13 is an approximation to $A$ norm of error at $k + 1$ iteration and it involves three terms.

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

All the terms individually are inner products of vectors in $\mathbb{R}^n$ and require $n$ multiplication operations; referred to as $O(n)$ arithmetic operations here. Here $d$ denotes the delay in estimation where $\alpha_k$ is a scalar. Note that the third term involves $Ap_k$ which is a vector provided by the Bi-CG Algorithm at every iteration with no additional cost to this estimator. Thus estimation of $A$ norm of error in BiCG algorithm requires $O(3n + nd)$ arithmetic operations in total.

Similarly for Equation 2.17 the first term requires $O(n + nd)$ operations in total and result of first term is partially used to compute the second term, hence requiring only $O(n)$ operations. Thus estimation of $l_2$ norm requires $O(2n + nd)$ operations.

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

But note that in both of the above expressions $d$ vectors are available at each iteration from previous iterations and thus estimation of $A$ norm and $l_2$ norm of error only requires two or three instances of inner-products of vectors in $\mathbb{R}^n$, thus making the arithmetic complexity $O(3n)$ and $O(2n)$ respectively.

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 $r^T A^{-1} r$ 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 more sable as compared to indefinite cases.

3.2. Analysis of estimators.

3.2.1. Condition number of the problem. Condition number plays a valuable role in matrix computations as they enable us to estimate the accuracy of com-
Significance of error estimation in iterative solution of linear systems

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:

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

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

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

$$(3.3) \quad \kappa = \|A\|\|A^{-1}\|$$

### 3.2.2. Data set and Performance 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:

$$(3.4) \quad \frac{|\chi_k\|_2}{\|x\|} - \frac{\|\epsilon_k\|_2}{\|x\|}$$

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:

$$(3.5) \quad U.R. (j) = \frac{1}{n-d} \sum_{k=0}^{n-d-1} \left( \frac{\|r_k\|_2^2}{\|b\|_2^2} - \frac{\|\epsilon_k\|_2^2}{\|x\|_2^2} \right) \left( \frac{\chi_k^2}{\|x\|_2^2} - \frac{\|\epsilon_k\|_2^2}{\|x\|_2^2} \right)$$

where $\chi_k$ is the estimate of norm of error at $k^{th}$ iteration and $n$ is dimension of matrix. We consider $j = 1$ and 2 for subsequent analysis which are defined as follows:

$$(3.6) \quad U.R. (1) = \frac{1}{n-d} \sum_{k=0}^{n-d-1} \left( \frac{\|r_k\|_2}{\|b\|_2} - \frac{\|\epsilon_k\|_2}{\|x\|_2} \right) \left( \frac{\chi_k}{\|x\|_2} - \frac{\|\epsilon_k\|_2}{\|x\|_2} \right)$$
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. Below, we show why the forward condition number of problem 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.

The Dataset consists of $O(n^2)$ problems of high condition numbers on which the performance metrics are measured. The convergence of an krylov subspace based iterative methods largely depend on the eigenvalue spectrum of matrix $A$. 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 matrices i.e. positive definite and indefinite matrices.

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

Theorem 3.1. For given singular value distribution and $\kappa(A,x)$,

$$E\left(U.R.(1)\right) \approx \sqrt{\frac{2}{3}} \left(\frac{\|A\|_F \|x\|}{\sqrt{n} \|b\|}\right) \left(1 + \frac{d}{n-d} \log \left(\frac{\sqrt{n} - \sqrt{d}}{\sqrt{n+d} - \sqrt{d}}\right) + \left(\frac{\sqrt{2d}}{\sqrt{n+d}}\right)\right),$$

$$\&$$

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

Provided, $\kappa(A,x) \gg 1$.

Proof.

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

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

where

$$a_k = \frac{\|r_k\|^2 \|x\|^2}{\|\epsilon_k\|^2 \|b\|^2 - 1}$$

$$\&$$

$$\frac{\|r_k\|^2 \|x\|^2}{\|\epsilon_k\|^2 \|b\|^2 - 1} - \frac{\chi_k^2}{\|\epsilon_k\|^2 - 1}$$
Now, consider the error at \(^{k}\text{th}\) iteration in GMRES or other krylov-subspace based methods,
\[
\epsilon_{k} = \epsilon_{0} - V_{k}d_{k}
\]

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 \(\epsilon_{k}\) on average will lie in \(n-k\) dimensional subspace provided \(\epsilon_{0}\) lies in entire \(n\)-dimensional space with any direction being equally probable. Hence, we can write,
\[
\epsilon_{k} = c_{1}v_{i_{1}} + c_{2}v_{i_{2}} + \ldots + c_{n-k}v_{i_{n-k}}
\]

where \(u_{i_{k}}\) and \(v_{i_{k}}\) are the left and right singular vectors of matrix \(A\) respectively. Now, lets consider the problems where \(\sigma_{i_{k}}\) are fixed. In such scenario,
\[
E_{\sigma_{1},...,\sigma_{n-k}} \left( \frac{\|r_{k}\|^{2} \|x\|^{2}}{\|\epsilon_{k}\|^{2} \|b\|^{2}} \right) = \left( \frac{\|A\|_{F}^{2} \|x\|^{2}}{\|b\|^{2}} \right) \left( \frac{1}{\|A\|_{F}^{2}} \right) E_{\sigma_{1},...,\sigma_{n-k}} \left( \frac{\|r_{k}\|^{2}}{\|\epsilon_{k}\|^{2}} \right)
\]

This can be done as forward condition number and singular values are constant across problems on which we are averaging. The operator \(E_{\sigma_{1},...,\sigma_{n-k}}\) will be replaced by \(P_{n-k}\) for following analysis.

\[
P_{n-k} \left( \frac{\|r_{k}\|^{2} \|x\|^{2}}{\|\epsilon_{k}\|^{2} \|b\|^{2}} \right) = \left( \frac{\|x\|^{2}}{\|b\|^{2}} \right) P_{n-k} \left( \frac{\|c_{1}\sigma_{i_{1}}u_{i_{1}} + c_{2}\sigma_{i_{2}}u_{i_{2}} + \ldots + c_{n-k}\sigma_{i_{n-k}}u_{i_{n-k}}\|^{2}}{\|c_{1}v_{i_{1}} + c_{2}v_{i_{2}} + \ldots + c_{n-k}v_{i_{n-k}}\|^{2}} \right) = \left( \frac{\|x\|^{2}}{\|b\|^{2}} \right) P_{n-k} \left( \frac{(c_{1}^{2}\sigma_{i_{1}}^{2} + c_{2}^{2}\sigma_{i_{2}}^{2} + \ldots + c_{n-k}^{2}\sigma_{i_{n-k}}^{2})}{(c_{1}^{2} + c_{2}^{2} + \ldots + c_{n-k}^{2})} \right)
\]

As all directions in \(n-k\) dimensional space are equally probable for error to point at, hence, the coefficients will follow gaussian distribution for such a vector and hence,
\(P_{n-k}(c_{i}^{2}) = 1\).

\[
P_{n-k} \left( \frac{\|r_{k}\|^{2} \|x\|^{2}}{\|\epsilon_{k}\|^{2} \|b\|^{2}} \right) \approx \left( \frac{\|x\|^{2}}{\|b\|^{2}} \right) \left( \frac{P_{n-k}(c_{1}^{2}\sigma_{i_{1}}^{2} + c_{2}^{2}\sigma_{i_{2}}^{2} + \ldots + c_{n-k}^{2}\sigma_{i_{n-k}}^{2})}{P_{n-k}(c_{1}^{2} + c_{2}^{2} + \ldots + c_{n-k}^{2})} \right) \approx \left( \frac{\|x\|^{2}}{\|b\|^{2}} \right) \left( \frac{(\sigma_{i_{1}}^{2} + \sigma_{i_{2}}^{2} + \ldots + \sigma_{i_{n-k}}^{2})}{n-k} \right)
\]

However, to find total expectation, we should consider all possible combinations of singular values to be equally probable and thus,
\[
E \left( \frac{\|r_{k}\|^{2} \|x\|^{2}}{\|\epsilon_{k}\|^{2} \|b\|^{2}} \right) \approx \left( \frac{\|x\|^{2}}{\|b\|^{2}} \right) \frac{1}{(n-k)} \sum_{S(\{i_{1},...,i_{n-k}\}) \in S} \left( \frac{(\sigma_{i_{1}}^{2} + \sigma_{i_{2}}^{2} + \ldots + \sigma_{i_{n-k}}^{2})}{n-k} \right)
\]
where $S$ is index set containing all $\binom{n}{n-k}$ combinations.

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

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

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

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

Now, consider

$$S = \frac{\|r_k\|^2 \|x\|^2}{\|\epsilon_k\|^2 \|b\|^2}$$

It can be seen that the random variable $S$ has well defined bounds as per Eqn. 3.13 which can be derived using Cauchy-Schwartz inequality:

$$\frac{\kappa(A, x)^2}{\kappa^2} \leq S \leq \kappa(A, x)^2$$

let $a = \frac{\kappa(A, x)^2}{\kappa^2}$. The lower bound $a$ lies between 0 and 1 where as upperbound is much greater than 1. The random variable $S$ can be seen to follow the relation 3.14 (proof in section 5.5 of Appendix):

$$E(|S - 1|) = E(S) - 1 + 2 \int_a^1 (1 - s)f(s)ds$$

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

$$E(|S - 1|) \approx \frac{\|A\|^2_F \|x\|^2}{n \|b\|^2}$$

Now,

$$E(a_k) \approx \frac{E\left(\frac{\|r_k\|^2 \|x\|^2}{\|\epsilon_k\|^2 \|b\|^2} - 1\right)}{E\left(\frac{\chi_k^2}{\|\epsilon_k\|^2} - 1\right)}$$
For delay-based estimators, $\chi^2_k \approx \|\epsilon_k\|^2 - \|\epsilon_{k+d}\|^2$, which means estimator is slightly less than actual error in general and thus the probability mass should be significant for $\frac{\chi^2_k}{\|\epsilon_k\|^2} < 1$. This justification allows us to write the following step.

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

Thus, According to Eqn. 3.10

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

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,

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

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

Thus,

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

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

**Derivation for $E(U.R.(1))$:**

Consider

$$Q = \frac{\|r_k\| \|x\|}{\|\epsilon_k\| \|b\|}$$

The random variable Q has well defined bounds as follows:

$$\frac{\kappa(A, x)}{\kappa} \leq Q \leq \kappa(A, x)$$
It can be assumed to follow the triangular distribution with mode at 1. This can be considered as safe assumption as it is in the favor of residual stopping criterion. However, this assumption also gives good intuition that most likely value of relative residual is relative error however due to huge variation, the relative residual on average is too far from relative error. The figure 3.3 depicts the distribution of Q.

For such a skewed triangular distribution,

\[ E(Q^2) \approx \frac{3}{2} E(Q)^2 \]

Thus,

\[ E(Q) \approx \sqrt{\frac{2}{3}} \left( \frac{\|A\|_F \|x\|}{\sqrt{n} \|b\|} \right) \]

Let

\[ \psi = \frac{\chi_k}{\|\epsilon_k\|} \]

The distribution of random variable \( \psi \) can be approximated well by the exponential distribution as the significant mass of probability lies between 0 and 1 and decreases very fast after 1. For exponential distribution,

\[ E(\psi^2) = 2 (E(\psi))^2 \]

Hence,

\[ E(\psi^2) = 1 - \frac{n - k - d}{n - k} = \frac{d}{n - k} \]
Significance of error estimation in iterative solution of linear systems

\[ E \left( \frac{\chi_k}{\| \epsilon_k \|} \right) = \sqrt{\frac{1}{2} \left( \frac{d}{n-k} \right)} \]

Now,

\[ E \left( \text{U.R.}^{(1)} \right) \approx \sqrt{\frac{2}{3} \left( \frac{\| A \|_F \| x \|}{\sqrt{n} \| b \|} \right)} \left( \frac{1}{n-d} \sum_{k=0}^{n-d-1} \frac{1}{1 - \sqrt{\frac{d}{n-k}}} \right) \]

\[ E \left( \text{U.R.}^{(1)} \right) \approx \sqrt{\frac{2}{3} \left( \frac{\| A \|_F \| x \|}{\sqrt{n} \| b \|} \right)} \left( \frac{1}{n-d} \sum_{k=0}^{n-d-1} \left( 1 + \frac{d}{n-k} - \frac{d}{2} \right) \right) \]

(3.17)

\[ E \left( \text{U.R.}^{(1)} \right) \approx \sqrt{\frac{2}{3} \left( \frac{\| A \|_F \| x \|}{\sqrt{n} \| b \|} \right)} \left( 1 + \frac{d}{n-d} \log \left( \frac{\sqrt{n} - \sqrt{\frac{d}{2}}}{\sqrt{d+1} - \sqrt{\frac{d}{2}}} \right) + \left( \frac{\sqrt{2d}}{\sqrt{n} + \sqrt{d}} \right) \right) \]

Lemma 3.2.

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

&

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

where \( E \) is the stopping tolerance on relative error.

Proof. Using Cauchy-Schwartz inequality, the \( \text{U.R.}^{(1)} \) can be seen to satisfy equation 3.18

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

(3.18)

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,

$$(3.19) \quad \frac{\|\epsilon_k\|^2}{\|\epsilon_k+d\|^2} \approx \left( \frac{d}{n} \right)$$

Thus,

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

Hence,

$$(3.20) \quad U.R.(1) = O \left( \kappa(A,x) \left( \frac{1}{E^2} \right) \frac{d}{n} \right)$$

Again using Cauchy-Schwartz inequality, the $U.R.(2)$ can be seen to satisfy equation $3.21$

$$(3.21) \quad U.R.(2) \leq \frac{\kappa(A,x)^2}{n-d} \left( \sum_{k=0}^{n-d-1} \frac{1}{1 - \frac{\chi_k^2}{\|\epsilon_k\|^2}} \right)$$

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 $3.19$

$$(3.22) \quad U.R.(2) = O \left( \kappa(A,x)^2 \left( \frac{1}{E^2} \right) \frac{d}{n} \right)$$
3.3. Experimental results on performance of error estimates as the stopping criteria.

Figures 3.4 and 3.5 show the increase in performance ratio (both for $A$-norm and $l_2$ norm estimation) as the condition number of the problem increases when matrices are taken to be Non-symmetric and Indefinite for Bi-CG Algorithm. 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 $b$).
Fig. 3.5. Linear increase in performance ratio for estimation of $l_2$ norm of error with increase in condition number of the problem when $A$ is Non-symmetric and Indefinite, 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 $\frac{\|f\|}{\|x\|}$.

There is a linear increase in the performance ratio with condition number of the problem, showing the robustness of our estimator. The same trend is expected for Non-symmetric but positive definite matrices as shown in Figure 3.6. The red line in all the above mentioned figures is just a threshold line and a point below the red line only depicts that relative residual was a better estimator of error as compared to our estimator (any measure) for a particular backward problem.

Fig. 3.6. Linear increase in performance ratio for estimation of $l_2$ norm of error with increase in condition number of the problem when matrix $(100 \times 100)$ is Non-symmetric positive definite.

Note that condition number of the problem for A-norm performance ratio will be defined by A-norm rather than a 2-norm condition number. Moreover the similar trend is expected in case of CGQL and GMRES algorithms where their estimators...
are compared to relative residual as a stopping criteria as shown in Figures 3.7 and 3.9.

Moreover the BiCGQL estimator is equivalent to a CGQL estimator when the matrix is Symmetric and Positive definite which is a requirement for guaranteed convergence of Conjugate Gradient Algorithm. Thus we expect to see a similar pattern of the trend line of performance ratio as the condition number of the problem increases for a CGQL Algorithm in Figure 3.7. This confirms the robustness of all three estimators (Bi-CG, CG and GMRES).
Fig. 3.9. Linear increase in performance ratio for estimation of $l_2$ norm of error with increase in condition number of the problem when matrix $(100 \times 100)$ is Non-symmetric in GMRES algorithm

### 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. $\kappa_F(A, x) = \frac{\|A\|_F \|x\|}{\sqrt{n} \|b\|}$ is averaged over $O(n^2)$ problems for each $d$ with dimension $n = 100$. The same experiment was done for $U.R.^{(2)}$ and results are shown in Fig. 3.10. The experimental results are compared with theoretical results provided in 3.1 and 3.2.
Significance of error estimation in iterative solution of linear systems

Fig. 3.10. Comparison of theoretical and experimental results on behaviour of $U.R.(1)$ and $U.R.(2)$ with delay parameter $d$ for BiCG and GMRES algorithms. Fig. 3.10(a) shows the effect of delay parameter on estimator performance with respect to $U.R.(1)$ measure and Fig. 3.10(b) shows the same with respect to $U.R.(2)$ measure.

The dataset consists of non-symmetric positive definite matrices of high condition number of $O(10^6)$. 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 desired relative error when dealing with the convergence of an iterative method. Relying on relative residue might stop the iterations too early or might take too many
iterations to converge to the final solution. The early stopping leads to loss of accuracy of solution which is proportional to difference of iterations required for relative error and relative residual to reach same stopping tolerance. We will call these iterations as accuracy loss iterations. The late stopping of leads to wastage of computation in the form of iterations. we will call these iterations as computation loss iterations. Use of error estimators owing to very less uncertainty in estimating relative error as compared to relative residue can reduce these lost iterations to significant amount as shown in figures [3.11 and 3.12](#).

![Convergence plot](image1.png)

**Fig. 3.11.** Illustration of over computation due to large uncertainty of relative residue leading to computation loss iterations for $1000 \times 1000$ nonsymmetric positive definite matrix with condition number of $O(10^{11})$ with random right hand side.

![Convergence plot](image2.png)

**Fig. 3.12.** 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.

The amount of savings should increase with dimensionality ($n$) of matrix as the uncertainty ratio according to equation ?? increases with size of matrix. It is clear that convergence rate will keep on decreasing as the size of matrix increases leading to more iterations saved by estimator and this could be proportional to dimension of matrix ($n$).
4. **Conclusions.** The importance of error estimators for efficient stopping (or restarting) are strongly evident for problems with even moderately high condition number \( \kappa > 100 \), and is emphasized by numerical examples, and the expected uncertainty in convergence otherwise using the residual. One might choose to use preconditioners for a highly conditioned problem but in most cases the structure of the matrix as well as properties of the matrix like condition number, spectra are not known. Even knowing that the condition number of the matrix is very high may not be helpful, as if one were able to reduce the condition number of a matrix it does not always imply reducing the condition number of the backward problem solved. Also one might actually increase the condition number of the forward problem and hence the uncertainty in error.

For Non-Symmetric matrices Bi-CG shows an irregular convergence which can be improved by using an extended version of the algorithm, BiCGSTAB \[11\] \[10\] \[9\]. BiCGSTAB acts as a moving window average of Bi-CG iterates which smoothen the convergence and there are other versions of BiCGSTAB that can be interpreted as the product of BiCG \[1\] and repeated application of the Generalized minimal residual method (GMRES) \[7\] in which a residual vector is minimized, which leads to a considerably smoother convergence behavior. It should also be noted that relations \[2.13\] and \[2.17\] hold valid for BiCGSTAB algorithms. Similarly, results showing the reduction of uncertainty in convergence while using error estimators were presented for GMRES. For HPD systems the Bi-CG delivers the same results as CG, but at twice the cost per iteration. Based on the results discussed in the previous sections, we believe that the estimate for the A-norm or the \( l_2 \) norm of the error should be implemented into software realization of iterative solvers, to use errors as stopping criteria instead of the residual.

5. **Appendix.**

**Algorithm 1** Lanczos Algorithm

1: procedure
2: \[\text{input: } A, v\]
3: \( \beta_0 = 0, v_0 = 0 \)
4: \( v_1 = \frac{\|v\|}{v} \)
5: for \( k = 1 \) to \( N \) do
6: \( m = Av_k - \eta_{k-1}v_{k-1} \)
7: \( \omega_k = \frac{v_k^T m}{\|m\|} \)
8: \( m = m - \omega_k v_k \)
9: \( \eta_k = \frac{\|m\|}{m} \)
10: \( v_{k+1} = \frac{m}{\eta_k} \)
11: end for
12: end procedure

**5.1. CG and its relation to (symmetric) Lanczos algorithm.** The Lanczos Algorithm \[3\] \[6\]) can be viewed as a simplified Arnoldi’s algorithm in that it applies to Hermitian matrices. The \( k^{th} \) step of the algorithm transforms the matrix \( A \) into a tri-diagonal matrix \( T_k \); when \( k \) is equal to the dimension of \( A \), \( T_k \) is similar to \( A \).

Given a starting vector \( v_1 \) and a symmetric matrix \( A \), the Lanczos algorithm
Algorithm 1 computes an orthonormal basis $v_1, v_2, ..., v_{k+1}$ of the krylov subspace $K_{k+1}(A, v_1)$ and transforms the matrix $A$ to a tri-diagonal matrix $T$.

\begin{equation}
K_{k+1}(A, v_1) = \text{span}\{v_1, Av_1, \cdots A^k v_1\}
\end{equation}

The basis vectors $v_k$ satisfy the matrix relation:

\begin{equation}
AV_k = V_k T_k + \eta_{k+1} v_{k+1} e_T
\end{equation}

Here, $e_k$ is the $k^{th}$ canonical vector, where $V_k = [v_1, ..., v_k]$ and $T_k$ is the $k \times k$ symmetric tri-diagonal matrix of recurrence coefficients in Algorithm 1:

\begin{equation}
T_n = \begin{bmatrix}
\omega_1 & \eta_1 & 0 & \cdots & 0 \\
\eta_1 & \omega_2 & \eta_2 & \vdots \\
0 & \ddots & \ddots & \ddots & 0 \\
\vdots & & \eta_{n-1} & \omega_{n-1} & \eta_{n-1} \\
0 & \cdots & 0 & \eta_{n-1} & \omega_n
\end{bmatrix}
\end{equation}

When solving a system of linear algebraic equations $Ax = b$ with symmetric and positive definite matrix $A$, the CG method (Algorithm 2) computes iterates $x_k$ that are optimal since the $A$-norm of error defined in (??) is minimized over $x_0 + \kappa_k(A, r_0)$,

\begin{equation}
\|x - x_k\|_A = \min_{y \in x_0 + \kappa_k(A, r_0)} \|x - y\|_A
\end{equation}

**Algorithm 2 Conjugate Gradient Algorithm**

1: procedure
2: input: $A, b, x_0$
3: $r_0 = b - Ax_0$
4: $p_0 = r_0$
5: for $k = 0$ to $N$ do
6: $\alpha_k = \frac{r_k^T r_k}{p_k^T Ap_k}$
7: $x_{k+1} = x_k + \alpha_k p_k$
8: $r_{k+1} = r_k - \alpha_k Ap_k$
9: $\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$
10: $p_{k+1} = r_{k+1} + \beta_k p_k$
11: end for
12: end procedure

CG can be derived from Lanczos Algorithm [3] [6] and relation between CG and Lanczos coefficients are as follows:
Significance of error estimation in iterative solution of linear systems

\[ \eta_k = \frac{\sqrt{\beta_k}}{\alpha_k}, \omega_k = \frac{1}{\alpha_{k-1}} + \frac{\beta_{k-1}}{\alpha_{k-2}} \]

5.2. 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:

\[ \| \epsilon_k \|^2_A - || \epsilon_{k+1} ||^2_A = \alpha_k r_k^T r_k \]

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

\[ \| \epsilon_k \|^2_A - || \epsilon_{k+d} ||^2_A = \sum_{j=k}^{k+d} \alpha_j r_j^T r_j \]

\[ \| \epsilon_k \|^2_A \approx \sum_{j=k}^{k+d} \alpha_j r_j^T r_j \]

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

\[ r_{k+1}^T A^{-1} r_{k+1} = -\alpha_k r_k^T p_k + r_{k+1}^T A^{-1} r_k + \alpha_k p_k^T A p_k \]

The above result can also be written as:

\[ r_k^T A^{-1} r_k - r_{k+1}^T A^{-1} r_{k+1} = \alpha_k r_k^T p_k + \alpha_k r_k^T (A^T)^{-1} A p_k - \alpha_k p_k^T A p_k \]

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

\[ r_k^T A^{-1} r_k - r_{k+1}^T A^{-1} r_{k+1} = \alpha_k r_k^T p_k + \alpha_k r_k^T p_k - \alpha_k p_k^T A p_k \]

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 \( A p_k = \frac{r_k - r_{k+1}}{\alpha_k} \) in the last term we get:

\[ r_k^T A^{-1} r_k - r_{k+1}^T A^{-1} r_{k+1} = \alpha_k r_k^T p_k + \alpha_k r_k^T p_k - \alpha_k p_k^T A p_k \]

\[ r_k^T A^{-1} r_k - r_{k+1}^T A^{-1} r_{k+1} = \alpha_k r_k^T r_k - \alpha_k r_{k+1}^T A^{-1} r_{k+1} + \alpha_k r_{k+1}^T r_k + \alpha_k p_k^T p_k \]

\[ \Rightarrow r_k^T A^{-1} r_k - r_{k+1}^T A^{-1} r_{k+1} = \alpha_k r_k^T r_k - \alpha_k r_{k+1}^T A^{-1} r_{k+1} + \alpha_k r_{k+1}^T r_k + \alpha_k p_k^T r_k \]

Equation 5.11 is equivalent to the result for A-norm estimation in CGQL Algorithm. Thus our A-norm estimator (BiCGQL) is equivalent to a CGQL estimator when \( A = A^T \).
5.2.1. 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:

\begin{equation}
(T^{-1}_n)_{(1,1)} = (T^{-1}_k)_{(1,1)} + \frac{\tilde{r}_k^T A^{-1} r_k}{\|r_0\|^2}
\end{equation}

\begin{equation}
(T^{-1}_n)_{(1,1)} = (T^{-1}_{k+1})_{(1,1)} + \frac{\tilde{r}_{k+1}^T A^{-1} r_{k+1}}{\|r_0\|^2}
\end{equation}

Subtracting (5.12) and (5.13) we get:

\begin{equation}
(T^{-1}_{k+1})_{(1,1)} = (T^{-1}_k)_{(1,1)} + \frac{\tilde{r}_k^T A^{-1} r_k - \tilde{r}_{k+1}^T A^{-1} r_{k+1}}{\|r_0\|^2}
\end{equation}

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:

\begin{equation}
(T^{-1}_{k+1})_{(1,1)} = (T^{-1}_k)_{(1,1)} + \frac{\alpha_k \tilde{r}_k^T r_k}{\|r_0\|^2}
\end{equation}

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.
**Algorithm 3** Conjugate Gradients and Quadrature via Lanczos coefficients

1: procedure
2: input: A,b,x₀,λₘ,λₘ
3: r₀ = b - Ax₀, p₀ = r₀
4: η₀ = 0, α₋₁ = 1, c₁ = 1, β₀ = 0, δ₀ = 1, ω(µ) = λₘ, ω(η) = λₘ
5: for k = 1...until convergence do
6:   ωₖ = 1/α₋₁ + β₋₁
7:   ηₖ = βᵏ/α₋₁
8:   δₖ = ωₖ - β₋₋₁
9:   gₖ = ||r₀||₂ ω₋₁δ₀
10:  δ₋₋₁ = ω₋₋₁, ω₋₋₁ = λₘ + β²
11:  f₋₋₁ = ||r₀||₂ η₋₋₁δ₀
12:  δ₋₋₁ = ω₋₋₁ - ω₋₋₁, ω₋₋₋₁ = λₘ + β²
13:  f₋₋₋₁ = ||r₀||₂ η₋₋₋₁δ₀
14:  ω₋₋₋₋₁ = λₘ - λₘ
15:  η₋₋₋₋₁ = δ₋₋₋₋₁
16:  f₋₋₋₋₁ = ||r₀||₂ (η₋₋₋₋₁δ₀)
17:  c₂₋₋₋₋₁ = η₋₋₋₋₁δ₀
18: end for
19: end procedure
5.2.2. 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 4 Bi-Conjugate Gradient Algorithm

1: procedure
2: input $A$, $A^T$, $b$, $x_0$, $y_0$
3: $r_0 = b - Ax_0$
4: $\tilde{r}_0 = b - A^Ty_0$
5: $p_0 = r_0$
6: $q_0 = \tilde{r}_0$
7: for $k = 1$...until convergence do
8: $\alpha_{k-1} = \frac{\tilde{r}_k^T r_{k-1}}{\eta_{k-1} p_{k-1}}$
9: $x_k = x_{k-1} + \alpha_{k-1} p_{k-1}$
10: $y_k = y_{k-1} + \alpha_{k-1} q_{k-1}$
11: $\tilde{r}_k = \tilde{r}_{k-1} - \alpha_{k-1} A^T q_{k-1}$
12: $\tilde{\eta}_{k-1} = \frac{\tilde{r}_k^T \tilde{r}_{k-1}}{\tilde{r}_{k-1}^T \tilde{r}_{k-1}}$
13: $\eta_k = \eta_{k-1} + \tilde{\eta}_{k-1}$
14: $\beta_k = \tilde{\eta}_k^T \eta_k$
15: $p_k = r_k + \beta_k p_{k-1}$
16: $q_k = \tilde{r}_k + \beta_k q_{k-1}$
17: end for
18: 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 $k=1,2,\ldots$

\begin{align}
(5.16) \\
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{align}

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

\begin{align}
(5.17) \\
v_{k+1} &= \frac{z_k}{\eta_k} \\
\tilde{v}_{k+1} &= \frac{\tilde{z}_k}{\tilde{\eta}_k}
\end{align}

These relations can again be written in the form of a non-symmetric tri-diagonal matrix form (under the condition $V_k^T AV_k = T_k$) as:
Significance of error estimation in iterative solution of linear systems

\[ T_k = \begin{bmatrix}
\omega_1 & \eta_1 & 0 & \cdots & 0 \\
\tilde{\eta}_1 & \omega_2 & \eta_2 & \ddots & \\
0 & \ddots & \ddots & \ddots & 0 \\
\vdots & & \tilde{\eta}_{k-2} & \omega_{k-1} & \eta_{k-1} \\
0 & \cdots & 0 & \tilde{\eta}_{k-1} & \omega_k
\end{bmatrix} \]

5.3. Generalized minimal residual method (GMRES). (GMRES) [7] is an iterative method for the numerical solution of a non-symmetric system of linear equations. The method approximates the solution by the vector in a Krylov subspace with minimal residual.

The krylov vectors in GMRES follow the (n+1)-term recurrence relation as follows:

\[ Av_k = h_{1,k}v_1 + \ldots + h_{k,k}v_k + h_{k+1,k}v_{k+1} \]

where \( h_{i,j} \) are the entries of Hessenberg matrix \( H_k \).

The Arnoldi iteration in GMRES exploits this relation to compute these entries and vectors using Modified Gram-Schmidt (MGS) technique, as in Algorithm 5.3.

**Algorithm 5 GMRES Algorithm**

1. set \( tol \)
2. set \( maxit \) (maximum iterations)
3. set \( k = 1 \)
4. \( n = \text{dim}(b) \)
5. \( r_0 = b - Ax_0, \beta = \| r_0 \|, v_1 = r_0 / \beta, \text{res} = \frac{\beta}{\| b \|} \)
6. while \( \text{res} > \text{tol} \) and \( k \leq \text{maxit} \) and \( k \leq n \) do
   1. \( s = Av_k \)
   2. for \( j = 1 \) to \( k \) do
      1. \( h_{j,k} = (v_j)^T s \)
      2. \( s = s - h_{j,k}v_j \)
   3. end for
   4. \( h_{k+1,k} = \| s \| \).
   5. if \( h_{k+1,k} \neq 0 \) then
      1. \( v_{k+1} = s / h_{k+1,k} \)
   6. end if
   7. Define the \((k + 1 \times k)\) Hessenberg matrix \( H_k^e \)
   8. Compute \( z_k \), the minimizer of \( \| \beta e_1 - H_k^e z_k \| \)
   9. \( x_k = x_0 + V_k z_k \)
   10. \( \text{res} = \frac{\| b - Ax_k \|}{\| b \|} \)
   11. \( k = k + 1 \)
7. end while
5.4. Bin experiment showing the average relative error in estimation of $l_2$ norm of error vector. For the purpose of extended tests on the Bi-CG estimator we proposed, six different bins of hundred matrices were generated with varying condition number of matrix $A$: $10^1$ to $10^6$. For each matrix $A$, 100 different instances of vector $'b'$ were created, each being unique canonical form of order 100. Thus each bin represents the result accumulated from 10,000 different cases.

In order to verify the relative precision of the estimator for $l_2$ norm of error, we perform the average of equation 5.20 over all iterations for each condition number of the matrix ranging from $10^1 - 10^6$.

$$\frac{\| f_k \| - \| e_k \|}{\| x \|} \frac{\| e_k \|}{\| x \|}$$

(5.20)

![Average relative error for $l_2$-norm estimation in BiCGQL](image)

**Fig. 5.1.** Average relative error between $l_2$-norm and it’s estimator for $d = 10$

Figure 5.1 shows that Equation 5.20 is independent of the condition number.

5.5. Expectation of $|S - 1|$.

The random variable $S$ takes values between $a$ and $b$ where $a \leq 1 \leq b$ and let $f(s)$ be the probability density function of $S$. In such case,

$$E(|S - 1|) = \int_a^b |s - 1| f(s)ds$$

$$= \int_a^1 (1 - s)f(s)ds + \int_1^b (s - 1)f(s)ds$$

$$= 2\int_a^1 (1 - s)f(s)ds + \int_a^b (s - 1)f(s)ds$$

$$= E(S) - 1 + 2\int_a^1 (1 - s)f(s)ds$$
References

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

[2] 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.

[3] Cornelius Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, United States Governm. Press Office Los Angeles, CA, 1950.

[4] 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.

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

[6] Beresford N Parlett and David S Scott, *The lanczos algorithm with selective orthogonalization*, Mathematics of computation, 33 (1979), pp. 217–238.

[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.