

Solving Complex-valued Linear Systems via Equivalent Real Formulations *

David Day and Mike Heroux

May 9, 2000

RECEIVED
JUN 07 2000
OSTI

Abstract

Most algorithms used in preconditioned iterative methods are generally applicable to complex valued linear systems, with real valued linear systems simply being a special case. However, most iterative solver packages available today focus exclusively on real valued systems, or deal with complex valued systems as an afterthought. One obvious approach to addressing this problem is to recast the complex problem into one of a several equivalent real forms and then use a real valued solver to solve the related system. However, well-known theoretical results showing unfavorable spectral properties for the equivalent real forms have diminished enthusiasm for this approach.

At the same time, our experience has shown us that there are situations where using an equivalent real form can be very effective. In this paper, we explore this approach, giving both theoretical and experimental evidence that an equivalent real form can be useful for a number of practical situations. Furthermore, we show that by making good use of some of the advance features of modern solver packages, we can easily generate equivalent real form preconditioners that are computationally efficient and mathematically identical to their complex counterparts.

Using our techniques, we are able to solve very ill-conditioned complex valued linear systems for a variety of large scale applications. However, more importantly, we shed more light on the effectiveness of equivalent real forms and more clearly delineate how and when they should be used.

1 Introduction

This work addresses the problem of solving a complex valued linear system

$$Cw = d \quad (1)$$

iteratively, where C is an m -by- n known complex matrix, d is a known right hand side and w is unknown. Although most preconditioners and iterative methods are directly applicable to complex valued linear systems [4], most preconditioned iterative solver packages deal only with real valued systems. There are some notable exceptions, including QMRPACK [6] and PETSc [3, 2, 1]. This work explains when and how to leverage existing real-valued solver packages for use with complex-valued systems.

1.1 The K1 and K2 Formulations

It is well known that if we write $Cw = d$ separated into real and imaginary parts, i.e.,

$$(A + iB)(x + iy) = b + ic \quad (2)$$

then the solution is equivalent to that of the system:

$$\begin{pmatrix} A & -B \\ B & A \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix} \quad (3)$$

*Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

An alternate formulation is:

$$\begin{pmatrix} A & B \\ B & -A \end{pmatrix} \begin{pmatrix} x \\ -y \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix} \quad (4)$$

For future reference, we call Equation 3 the Komplex-1 or K1 formulation (and denote the matrix by K_1) and Equation 4 the Komplex-2 or K2 formulation.

The K1 and K2 formulations have been studied before (see [7]) and the common belief is that these approaches are generally a bad idea. This is primarily because if λ is an eigenvalue of C , then $\bar{\lambda}$ is an eigenvalue of K_1 . If C has all eigenvalues on one side of the imaginary axis, then the spectrum of K_1 should not present a major problem to iterative methods such as GMRES. However, if C has eigenvalues on both sides of the imaginary axis, a property that causes difficulty for GMRES, then K_1 will have twice as many bad eigenvalues. Furthermore, the convex hull containing the eigenvalues of K_1 will contain the origin, an additional bad property for convergence of GMRES.

The situation for the K2 formulation is even worse. If λ is an eigenvalue of K_2 , then $\{-\lambda, \bar{\lambda}, -\bar{\lambda}\}$ are all eigenvalues of K_2 . Thus, no matter how well we precondition the K2 problem, the convex hull of the eigenvalues will always contain the origin.

Because of the above properties, the K1 and K2 formulations have been justly criticized. Particularly, the K2 formulation does not appear to be usable. However, we have found that a variation of the K1 formulation does indeed have merit, if a sufficiently good preconditioner is applied to problem, something that we found is generally required to achieve convergence, even when using a complex solver on the original complex system. In fact, our experience shows that for the classes of problems we are solving, if a good preconditioner is used, then the iteration count of the K formulation discussed below is generally comparable (within 50%) to that of solving the original complex problem with a true complex preconditioned iterative solver, and has the same robustness as a true complex solver. Given the wide availability of good real valued solver packages, we view our results as noteworthy.

2 Approaches to Solving Real Equivalent Systems

In trying to solve the original complex system in Equation 1 via the K1 formulation in Equation 3, the most interesting question is how to precondition K_1 . Applying standard real valued preconditioners such as Jacobi, Gauss-Seidel or ILU directly to K_1 brings a variety of problems. Generally we found that they were not robust enough for our needs. Furthermore, applying ILU to K_1 was complicated by its unusual sparsity pattern.

Another approach we tried was to focus on preconditioning just the (1,1) and (2,2) blocks in Equation 3 by using a standard real valued preconditioner, denoted M_A , applied to A , the real part of C . The preconditioner then became:

$$\begin{pmatrix} M_A & 0 \\ 0 & M_A \end{pmatrix}. \quad (5)$$

Again, we found that this approach was not robust enough for our problems, providing a large degradation in iteration count or an outright failure to converge, while the solution of the complex system via a complex solver succeeded. Furthermore, the related approach of putting B and $-B$ (by swapping block rows in Equation 3) also failed.

2.1 The K Formulation

The final approach that we tried, one which consistently gives us good results, was to make the following observation: If $c_{ij} = a_{ij} + \sqrt{-1} b_{ij}$ is a nonzero entry of C then we can apply the K1 formulation directly to each entry of C to generate a 2-by-2 block entry of the form

$$\begin{pmatrix} a_{ij} & -b_{ij} \\ b_{ij} & a_{ij} \end{pmatrix}. \quad (6)$$

This leads to the K formulation where the matrix K is a $2m$ -by- $2n$ real matrix partitioned in to 2-by-2 blocks and each (i,j) block entry of K is of the form in Equation 6 corresponding to c_{ij} . For example, if

$$C = \begin{pmatrix} c_{11} & 0 & c_{13} & 0 & c_{15} \\ 0 & c_{22} & c_{23} & 0 & 0 \\ c_{31} & 0 & c_{33} & c_{34} & 0 \\ 0 & 0 & c_{43} & c_{44} & 0 \\ c_{51} & 0 & 0 & 0 & c_{55} \end{pmatrix} \quad (7)$$

then

$$K = \begin{pmatrix} a_{11} & -b_{11} & 0 & 0 & a_{13} & -b_{13} & 0 & 0 & a_{15} & -b_{15} \\ b_{11} & a_{11} & 0 & 0 & b_{13} & a_{13} & 0 & 0 & b_{15} & a_{15} \\ \hline 0 & 0 & a_{22} & -b_{22} & a_{23} & -b_{23} & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{22} & a_{22} & b_{23} & a_{23} & 0 & 0 & 0 & 0 \\ \hline a_{31} & -b_{31} & 0 & 0 & a_{33} & -b_{33} & a_{34} & -b_{34} & 0 & 0 \\ b_{31} & a_{31} & 0 & 0 & b_{33} & a_{33} & a_{34} & -b_{34} & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & a_{43} & -b_{43} & a_{44} & -b_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & b_{43} & a_{43} & a_{44} & -b_{44} & 0 & 0 \\ \hline a_{51} & -b_{51} & 0 & 0 & 0 & 0 & 0 & 0 & a_{55} & -b_{55} \\ b_{51} & a_{51} & 0 & 0 & 0 & 0 & 0 & 0 & b_{55} & a_{55} \end{pmatrix} \quad (8)$$

2.2 Implementation within Existing Software Packages

The K formulation defined above has several nice properties that allow us to implement efficient and robust preconditioned iterative solvers for complex linear systems. It allows us to efficiently compute and apply the exact equivalent of a complex valued preconditioner and it has sufficiently good spectral properties that, when properly preconditioned, allow for good convergence relative to a true complex solver.

Several full-featured solver packages, e.g., Aztec [12] and PETSc [2], support block entry matrices. Matrices of this form are sparse, but each entry of the matrix is a dense matrix itself. The matrix K in the Komplex formulation has a natural 2-by-2 block structure that can be exploited by using block entry data structures. By using the block entry features of these packages we get the following benefits:

1. Applying 2-by-2 block Jacobi scaling to K corresponds exactly to applying point Jacobi scaling to C .
2. The block sparsity pattern of K exactly matches the point sparsity pattern of C . Thus, any pattern based preconditioners such as BILU(k) applied to K correspond exactly to ILU(k) applied to C .
3. Any drop tolerance based complex preconditioner has a straight-forward K formulation since the absolute value of a complex entry equals the scaled Frobenius norm of the corresponding block entry in K .

Thus, by using these block entry features, we can easily and efficiently construct preconditioners for K that are equivalent to those we would form for C if we used a true complex formulation. As a result, our preconditioned matrix operation is *identical* to the true complex preconditioned operator up to a permutation. This implies that solving the real equivalent form via the K formulation using a preconditioned iterative method is identical to solving the original complex system using a corresponding preconditioned complex solver, except that they are using different inner product spaces.

3 Properties of the K Formulation

This section is a mathematical justification of the K formulation that takes into account the essential role of preconditioning in iterative solution methods. The K formulation uses an equivalent real form of the

preconditioner as well as the coefficient matrix. A K1 formulation transforms the eigen- and singular value decompositions adversely, from the perspective of an iterative linear solver. In this section we show that the K formulation transforms in preconditioned complex linear system in the same way that the K1 formulation transforms the coefficient matrix. In contrast to the unpreconditioned case though, preconditioned iterative linear solvers are insensitive to the K formulation. Iterative methods applied to poorly preconditioned systems do not converge in either formulation and, if applied to a well preconditioned systems, converge at essentially the same rate.

We use the following framework to comparing preconditioned iterative linear solvers For clarity a zero initial guess is assumed. A left-preconditioned Krylov solver is an algorithm that determines a sequence of approximations to z from the corresponding expanding Krylov subspaces

$$\mathcal{K}^k(M^{-1}C, M^{-1}d) = \text{span}(M^{-1}d, \dots, (M^{-1}C)^{k-1}M^{-1}d) \quad (9)$$

Right-preconditioning is similar. A preconditioned Krylov solver succeeds to the extent that the algorithm converges in a small number of iterations.

Characterizing good preconditioners is hard. A crude measure of the quality of the preconditioner M is the maximum ratio of singular values,

$$\text{cond}(M^{-1}C). \quad (10)$$

The convergence rate of GMRES is bounded in terms of the eigenvalues and condition number of the matrix of eigenvectors of $M^{-1}C$ [11]. In this section we show that in our approach the Krylov subspace, (9), and the condition number, (10), of complex preconditioned linear system are preserved. Additionally, the condition number of the matrix of eigenvectors is also preserved.

We use the function $f()$ to denote the matrix K that corresponds to an arbitrary complex matrix C in the K1 transformation,

$$f(C) = K$$

The K formulation of a preconditioned Krylov subspace method with preconditioner M applies the preconditioner $f(M)$ to $f(C)$. The matrix $K = f(C)$ is formed explicitly, but linear systems

$$f(M) \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix}$$

are only solved implicitly by solving $M(x + iy) = u + iv$ in (simulated) complex arithmetic.

The key to understanding the K formulation is that f is a homomorphism,

$$\begin{cases} f(I) = I \\ f(XY) = f(X)f(Y) \end{cases}$$

This observation appears not to have been made before in this context. From this one can deduce relations such as $f(X^{-1}Y) = f(X)^{-1}f(Y)$.

The proposed method preserves Krylov subspaces in the sense that

$$f((M^{-1}C)^k M^{-1}) = (f(M)^{-1}f(C))^k f(M)^{-1}$$

This statement applies to left preconditioning. A similar statement applies if right preconditioning is used. A prerequisite to discussing condition numbers is to relate the singular values decomposition (SVD) of C ,

$$C = U\Sigma V^*$$

to the SVD of $f(C)$. Problem 8.6.4 in [8] is to show that if $U = U_r + iU_u$ and $V = V_r + iV_u$, where U_r, U_u, V_r, V_u are real then $f(C)$ has the SVD

$$K = \begin{bmatrix} U_r & -U_u \\ U_u & U_r \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & \Sigma \end{bmatrix} \begin{bmatrix} V_r & -V_u \\ V_u & V_r \end{bmatrix}^T$$

In particular f preserves condition numbers. This implies the second property, that conditioning is preserved from the complex case:

$$\begin{aligned} \text{cond}(M^{-1}C) &= \text{cond}(f(M^{-1}C)) \\ &= \text{cond}(f(M)^{-1}f(C)) \end{aligned}$$

For clarity we discuss eigenvalues in the unpreconditioned case. But as above these results to extend to the preconditioned case. As mentioned in the introduction, eigenvalues of K are either eigenvalues of C or their conjugates. More precisely, Proposition 5.1 of [7] states that if C has Jordan normal form,

$$C = XJX^{-1}$$

then $K = f(C) = W \text{diag}(J, \bar{J}) W^{-1}$ for

$$W = \begin{bmatrix} X & \bar{X} \\ -iX & i\bar{X} \end{bmatrix}$$

An observation not made explicitly in [7] though, is that it immediately follows that

$$\text{cond}(W) = \text{cond}(X)$$

3.1 Convergence of the K Formulation

In this section the sensitivity of the convergence rate of an iterative method to a K formulation is quantified for certain classes of model problems. For Hermitian linear systems the K formulation preserves the convergence rate. For linear systems with asymmetric preconditioned spectrum (see Figure 1), the convergence rate degrades mildly in a K formulation.

The most important difference between K and C is that the eigenvalues of K are to eigenvalues of C together with their conjugates; $\sigma(K) = \sigma(C) \cup \overline{\sigma(C)}$ where $\sigma(\cdot)$ denotes the matrix spectrum. The asymptotic convergence factor for polynomial based methods, including preconditioned Krylov subspace methods, is κ if the residual norms decrease like κ^n . Successful preconditioning transforms the spectrum into a disk far from 0. In the K formulation the asymptotic convergence factor is the same if this disk is centered on the real axis. In this section we determine the influence on the convergence rate of preconditioners that map the spectrum to a disk not centered on the real axis. Fortunately in practice preconditioners cluster the spectrum about 1. For example if C is Hermitian, then $\sigma(K) = \sigma(C)$, and the asymptotic convergence factors are the same in the complex and Komplex formulations. But if C is skew-Hermitian, then $\sigma(C)$ is a subset of the imaginary axis, and in the K formulation iterative linear solution methods can converge much more slowly. In the skew Hermitian case, the slower convergence is avoided by using the K formulation of the Hermitian linear system $\sqrt{-1}Cw = \sqrt{-1}d$.

A sharp upper bound for the 'asymptotic convergence factor' can be determined using the complex Greens function for the convex hull of the spectrum. Following [5] pages 90-93, an iterative method for $Cw = d$ (hopefully) determines $\{w_i\}$ that converge to w and residual, $r_i = d - Cw_i$ that converge to zero. Consider the polynomial based iterative solution method

$$r_n = p_n(C)r_0, \quad p_n(0) = 1$$

where each $p_n \in \Pi_n$, the space of n th degree polynomials. Next let Ω be a set containing $\sigma(C)$, and define

$$\|p_n\|_\Omega = \max_{\omega \in \Omega} |p_n(\omega)|$$

Common choices for Ω are the convex hull of $\sigma(C)$, a disk, or an ellipse with major axis along a ray through the origin. For clarity, let's assume that C is diagonalizable, $CV = V\Lambda$. The reduction in the residual norm

$$\|r_n\|_2 = \|p_n(C)r_0\|_2 \leq \text{cond}(V) \|p_n\|_{\sigma(C)} \|r_0\|_2 \leq \text{cond}(V) \|p_n\|_\Omega \|r_0\|_2$$

is bounded above by the spectral condition number of V and $\|p_n\|_\Omega$. A residual polynomial p_n that minimizes $\|p_n\|_\Omega$ is an *optimal polynomial* $\mathcal{P}_n(t; \Omega, 0)$ and solves

$$\|\mathcal{P}_n(t; \Omega, 0)\|_\Omega = \min\{\|p\|_\Omega : p \in \Pi_n, p(0) = 1\}, \quad 0 \notin \Omega$$

The convergence of an iteration $r_n = p_n(C)r_0$ is related to the asymptotic convergence factor for the polynomial iteration method induced by p_n ,

$$\kappa(C; p_n) = \limsup_{n \rightarrow \infty} \left(\sup_{r_0 \neq 0} \frac{\|r_n\|_2}{\|r_0\|_2} \right)^{1/n}$$

The asymptotic convergence factor for Ω is defined by

$$\kappa(\Omega) = \inf_{P_n} \sup_{\sigma(C) \subset \Omega} \kappa(C; p_n)$$

The asymptotic convergence factor corresponding to the K formulation is $\kappa(\Omega \cup \bar{\Omega})$.

Next we determine the asymptotic convergence factor for C and K in several characteristic cases. $\kappa(\Omega)$ is determined from the the Green's function $G(z; \Omega^c)$ for the complement Ω^c of Ω with pole at infinity

$$\kappa(\Omega) = |G(0; \Omega^c)|$$

If Ω is connected, then G is a conformal mapping from Ω^c to the open unit disk such that $G(\infty) = 0$. In general G has the following properties:

- $G(z; \Omega^c)$ is an analytic function on Ω^c with single valued modulus $|G(z; \Omega^c)| < 1$ in Ω^c .
- $G(z; \Omega^c)$ has precisely one zero at ∞ .
- If $z \in \partial\Omega$, then $|G(z; \Omega^c)| = 1$.

We give four examples of evaluating $\kappa(\Omega)$ using Green's functions. Examples one and three come from [5]. Example two is classical and reviewed here in detail. Example four appears to be new. The first two examples correspond to the case in which C is Hermitian positive definite and $\sqrt{i}C$ is Hermitian positive definite. In these examples, I denotes the interval $[-1, 1]$.

Example 1. If C is Hermitian positive definite, then the convex hull of $\sigma(C) = \sigma(K)$ is $\Omega = [\alpha, \beta]$ for $0 < \alpha < \beta$. The first step is to derive the Green's function for the complement of the interval $G(z; I^c)$, an inverse of the Joukowski map $\phi(z) = (z + z^{-1})/2$. We select the square root function with branch cut along the negative real axis and

$$\phi^{-1}(z) = \begin{cases} z + \sqrt{z^2 - 1} & 0 \leq \arg(z) \leq \pi \\ z - \sqrt{z^2 - 1} & \pi < \arg(z) < 2\pi \end{cases}$$

Note that the branch cut for the argument function is along the positive real axis, and for $\rho > 1$,

$$\lim_{z \rightarrow \rho, \Im(z) < 0} \arg(z^2 - 1) = 2\pi$$

Here $\Im(z)$ denotes the imaginary part of the complex number z . In this case the singularity in ϕ^{-1} along $[1, \infty]$ has been removed, and because ϕ^{-1} is odd, ϕ^{-1} is analytic on I^c . To show that ϕ^{-1} maps I^c to the open unit disk, note that the equation $\phi(z) = w$ is a quadratic polynomial in z and $\phi(z) = \phi(z^{-1})$.

Now for $\Omega = [\alpha, \beta]$ such that $0 < \alpha < \beta$, $G(z; \Omega^c) = \phi^{-1}(\ell(z))$ for $\ell(t) = (2t - \alpha - \beta)/(\beta - \alpha)$ and

$$\kappa([\alpha, \beta]) = |G(0)| = \frac{1}{-\ell(0) + \sqrt{\ell^2(0) - 1}} = \frac{\beta - \alpha}{\beta + 2\sqrt{\beta\alpha} + \alpha}$$

Example 2. If iC is Hermitian positive definite, then the convex hull of $\sigma(C)$ is $[i\alpha, i\beta]$ for $0 < \alpha < \beta$ and $\kappa(i[\alpha, \beta]) = \kappa([\alpha, \beta])$. But $\sigma(K) \subset [-i\beta, -i\alpha] \cup [i\alpha, i\beta]$ and $\kappa([-i\beta, -i\alpha] \cup [i\alpha, i\beta]) = \kappa([- \beta, -\alpha] \cup [\alpha, \beta])$. The first step is to derive the Green's function for the complement of symmetric intervals $\Omega = [-1, -\eta] \cup [\eta, 1]$ for $\eta = \alpha/\beta < 1$. The Green's function is a branch of the solution of

$$\psi^2 + 2 \frac{2z^2 - (1 + \eta^2)}{1 - \eta^2} \psi + 1 = 0$$

The two branches

$$\psi_{\pm} = \frac{2}{1 - \eta^2} \left(\pm \sqrt{(z^2 - 1)(z^2 - \eta^2)} - z^2 + \frac{1 + \eta^2}{2} \right)$$

satisfy $\psi_+ \psi_- = 1$, and we seek a branch whose range is the open unit disk. We choose the branch of $\pm \sqrt{(z^2 - 1)(z^2 - \eta^2)}$ that is nearest to z^2 . The branch cut for $\sqrt{}$ is along the negative real axis and

$$\arg((z^2 - 1)(z^2 - \eta^2)) = \pm\pi \leftrightarrow z = x + iy \text{ and } x^2 - y^2 = \frac{1 + \eta^2}{2}$$

This gives us the Green's function

$$G(z) = \begin{cases} \frac{2}{1-\eta^2} \left(\sqrt{(z^2-1)(z^2-\eta^2)} - z^2 + \frac{1+\eta^2}{2} \right) & x^2 - y^2 \geq \frac{1+\eta^2}{2} \\ \frac{2}{1-\eta^2} \left(-\sqrt{(z^2-1)(z^2-\eta^2)} - z^2 + \frac{1+\eta^2}{2} \right) & x^2 - y^2 < \frac{1+\eta^2}{2} \end{cases}$$

To show that the range of G is the open unit disk, note that ψ is a quadratic polynomial and $|\psi_{\pm}| = 1$ if and only if $\eta^2 \leq z^2 \leq 1$. In this case $\kappa(\Omega) = (1-\eta)/(1+\eta) = (\beta-\alpha)/(\beta+\alpha)$.

Examples one and two quantify the difference between the K_1 and K_2 formulations for the special case of a Hermitian positive definite linear system with eigenvalues in $[\alpha, \beta]$. In the K_1 formulation,

$$\kappa_1 \approx 1 - 2\sqrt{\frac{\alpha}{\beta}}$$

while for the K_2 formulation the factor is much larger:

$$\kappa_2 \approx 1 - 2\frac{\alpha}{\beta} \gg \kappa_1$$

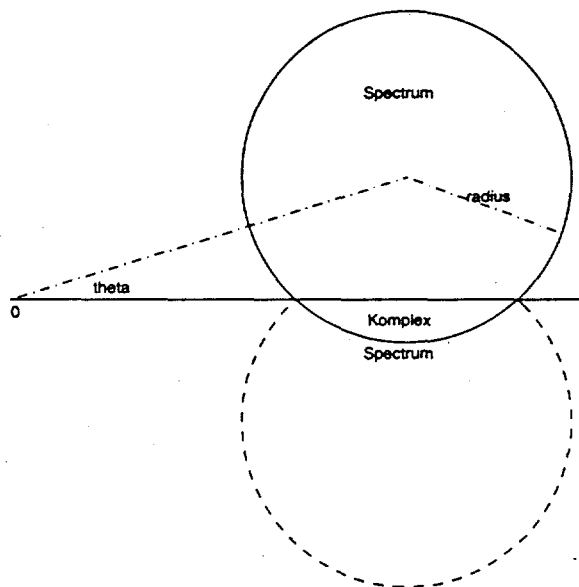


Figure 1: Asymmetric preconditioned spectrum

Examples three and four quantify the penalty for using the K formulation instead of the true complex formulation if the convex hull of the preconditioned spectrum lies inside a disk in the left half plane rotated by θ from the positive real axis. Our analysis only applies in the case in which θ is small enough that the disk intersects its conjugate. In complex arithmetic, the asymptotic convergence factor is independent of θ , but in the K formulation the asymptotic convergence factor mildly with θ .

Example 3. For a disk $\Omega = \{z : |z - w| < r\}$ not containing 0 (i.e. $r < |w|$) a conformal mapping onto the open unit disk is $G(z; \Omega) = \frac{r}{z-w}$ and

$$\kappa(\Omega) = \frac{r}{|w|}$$

Example 4. Here Ω comes from rotating the disk $\{z : |z - \rho| < r\}$ with center $\rho > 0$ by θ and then reflecting through the real axis. I assume that circles intersect, $\sin \theta < \eta = r/\rho$. This is illustrated in figure 1

The conformal mapping of the exterior of two intersecting circles to the open unit disk is given by

$$G(z) = \frac{\left(\frac{z-\mu}{z-\nu}\right)^{\pi/\alpha} - 1}{\left(\frac{z-\mu}{z-\nu}\right)^{\pi/\alpha} + 1}$$

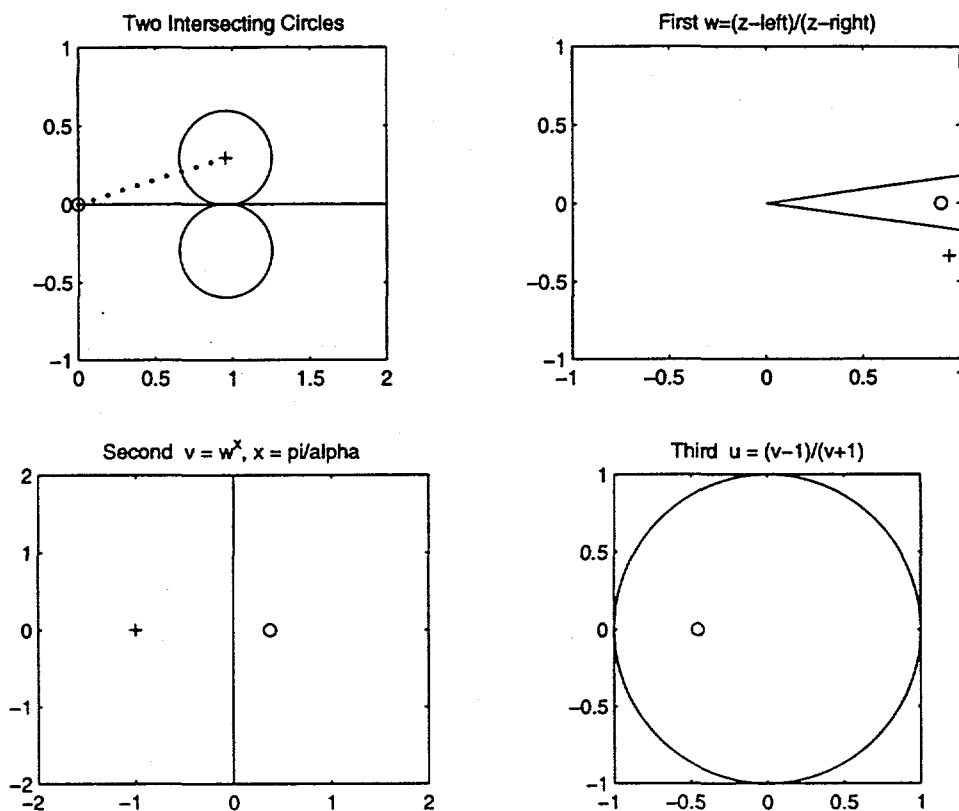


Figure 2: Conformal mapping in three steps

where

$$\mu = \cos \theta - \sqrt{\eta^2 - \sin^2 \theta} \quad \nu = \cos \theta + \sqrt{\eta^2 - \sin^2 \theta} \quad \alpha = \pi - 2 \tan^{-1} \left(\frac{\sin \theta}{\sqrt{\eta^2 - \sin^2 \theta}} \right)$$

Also $w^{\pi/\alpha} = \exp(\frac{\pi}{\alpha} \log w)$ for $\log w = \log |w| + i \arg w$ and the branch for the argument function is placed along the negative real axis, $-\pi < \arg w \leq \pi$. In this case

$$\kappa(\Omega) = (1 - (\mu/\nu)^{\pi/\alpha}) / (1 + (\mu/\nu)^{\pi/\alpha}) = r + \frac{|\theta|}{\pi} (1 - r^2) \log \left(\frac{1+r}{1-r} \right) + O(\theta^2)$$

3.2 Summary of K Formulation Properties

Based on the previous results of this section, we see that the K formulation differs from a true complex iterative solver only in the inner product space used by the iterative method. In other words, by utilizing the block entry data structures mentioned in Section 2.2, we are able to provide the identical preconditioned matrix multiply computations using the K formulation as we would for a true complex solver.

Furthermore, for complex Hermitian problems, there is no difference in asymptotic convergence rates. In fact, as is well known (e.g., see Problem 8.3.6 in [8]), if a complex matrix C is Hermitian (positive definite), the corresponding K matrix $K = f(x)$ is symmetric (positive definite) with identical eigenvalues having twice the multiplicity. Given the ability of the conjugate gradient method to resolve multiple identical eigenvalues simultaneously, we observe in practice that K formulation has identical convergence properties as a true complex solver for complex Hermitian problems.

For the non-Hermitian case, we saw from Examples 3 and 4 above that, if the disk enclosing the spectrum of the preconditioned matrix C is not far from the point (1,0) in the complex plane, then the asymptotic convergence rate of the K formulation is close to the convergence rate of a true complex solver, with the

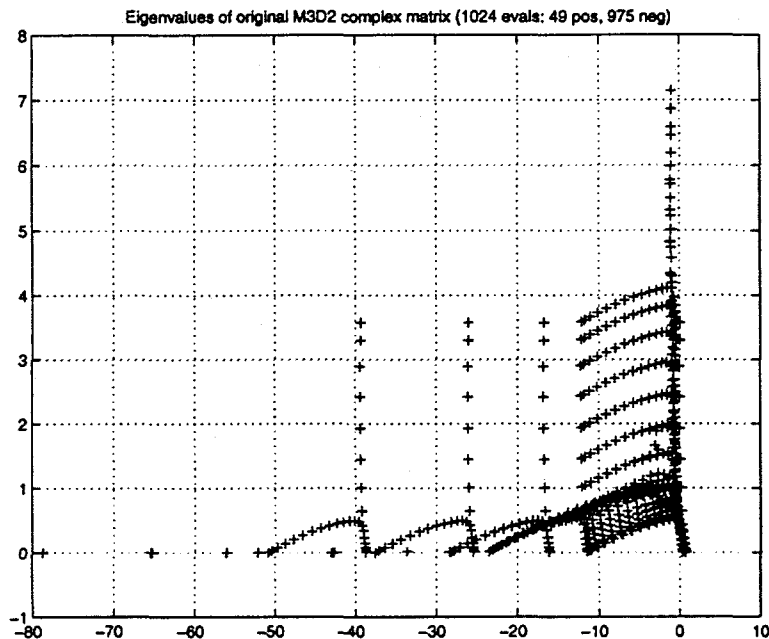


Figure 3: Eigenvalues of the original complex matrix in problem M3D2.

rate dictated by the size of the angle θ in Figure 1. As we will see in the next Section 3.3, a high quality preconditioner tends to move the spectrum of C toward $(1,0)$, setting up very favorable conditions for the K formulation.

3.3 Spectral Case Study

For the problem M3D2 listed in Table 1, we computed the spectrum of the original and preconditioned matrix using the `eig` function of Matlab. Figure 3 shows the distribution of eigenvalues for the original matrix¹. Figure 4 shows the eigenvalues for the K matrix and, as expected, the eigenvalues of the K matrix are the eigenvalues of the complex matrix plus their reflection about the real-axis.

Figure 5 shows the spectrum of preconditioned matrix using `luinc(A, 1e-1)` from Matlab. This is an incomplete LU preconditioner that uses a drop tolerance of 10^{-1} . Note that the eigenvalues start to cluster around the point $(1,0)$ in the complex plane. Figure 6 shows the spectrum using `luinc(A, 1e-2)`. With the exception of one outlier, the eigenvalues are becoming very closely clustered around $(1,0)$.

Coupling this observation with the analysis from Section 3.1, we see that high quality preconditioning, which tends to cluster the eigenvalues around the point $(1,0)$, additionally minimizes the differences in asymptotic convergence rates between the true complex formulation and the K formulation. Thus, it simultaneously improves the convergence of both formulations and reduces the differences in convergence rates between them.

4 Experimental Results

We have used the K formulation to solve complex linear systems coming from two application areas (see Table 1): molecular dynamics and linear stability analysis. Each linear system comes from a real application and is very ill-conditioned. In addition, because some of these users are solving eigenvalues problems via shift-and-invert methods, the solution of the complex system must be very accurate. We have found that the K formulation with a strong ILU preconditioner has been very effective.

¹A note of thanks to Tom Wright and Nick Trefethen. They analyzed pseudospectra of this matrix and determined that the eigenvalues of this matrix obtain via `eig` would be accurately computed.

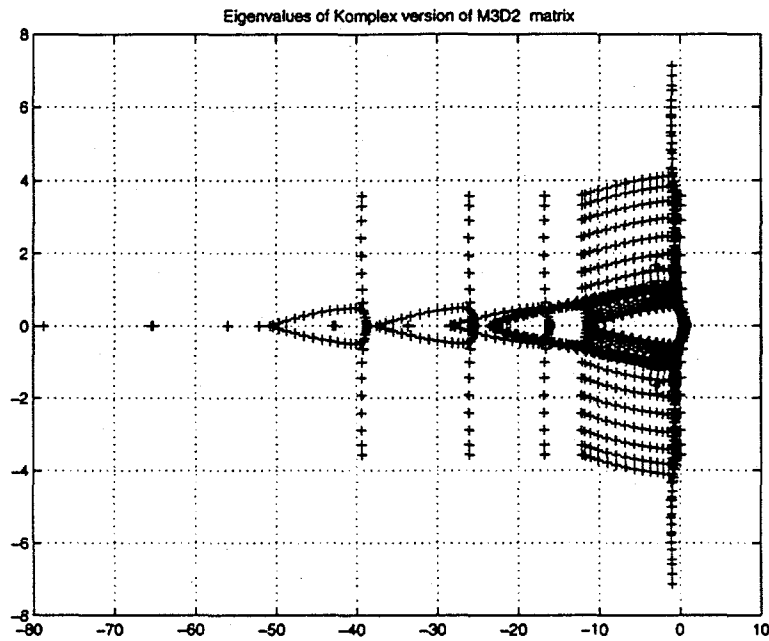


Figure 4: Eigenvalues of the K formulation matrix in problem M3D2.

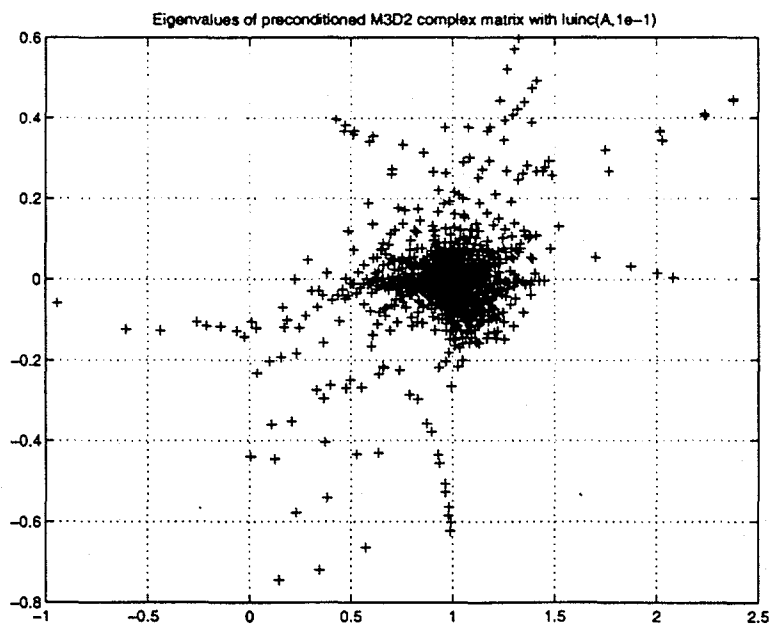


Figure 5: Eigenvalues of the complex matrix in problem M3D2, preconditioned by `luinc(A,1e-1)`.

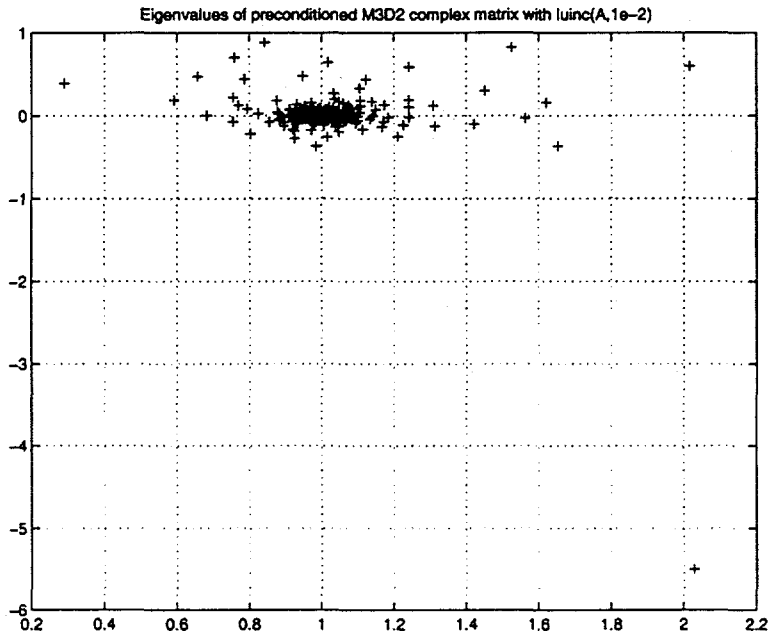


Figure 6: Eigenvalues of the complex matrix in problem M3D2, preconditioned by `luinc(A,1e-2)`.

<i>Problem</i>	<i>Dim</i>	<i># Nonzeros</i>	<i>Description</i>
<i>M3D2</i>	1024	12480	Computational Chemistry Model I, Sherry Li, LBL/NERSC
<i>M4D2</i>	10000	127400	Computational Chemistry Model II, Sherry Li, LBL/NERSC
<i>LINSTAB1</i>	10590	276979	MPSalsa Linear Stability Analysis, Andrew Salinger, Rich Lehoucq, Cayley Transform Approach
<i>LINSTAB2</i>	10590	276979	MPSalsa Linear Stability Analysis, Andrew Salinger, Shift = $33 \cdot \sqrt{-1}$

Table 1: Test Problem Descriptions.

<i>Problem</i>	<i>droptol</i>	$nz(ILU)/nz(A)$	$\ r\ /\ b\ $	<i>C Iters</i>	<i>K Iters</i>
<i>M3D2</i>	1×10^{-3}	5.8	3×10^{-11}	12	12
<i>M3D2</i>	1×10^{-2}	4.5	8×10^{-11}	30	40
<i>M3D2</i>	1×10^{-1}	0.5	5×10^{-11}	107	181
<i>M4D2</i>	1×10^{-4}	13.1	5×10^{-11}	17	23
<i>M4D2</i>	1×10^{-3}	6.7	6×10^{-11}	72	109
<i>LINSTAB1</i>	1×10^{-3}	10.7	9×10^{-11}	71	93

Table 2: MATLAB Test Results Using GMRES(∞) with luinc(droptol) Preconditioning.

<i>Problem</i>	<i>levfill</i>	$\ r\ /\ b\ $	<i>K Iters</i>	<i>Time(s)</i>
<i>M4D2</i>	0	1×10^{-11}	322	354
<i>M4D2</i>	1	1×10^{-11}	179	182
<i>M4D2</i>	2	1×10^{-11}	75	95

Table 3: Aztec Test Results for M4D2 Using GMRES(∞) with Block ILU Preconditioning, 8-by-8 blocks.

The first set of results (in Table 2) come from a Matlab code where we compare a true complex preconditioned iterative solver to the K formulation. For these problems, the preconditioned operators are exactly equivalent, i.e., using the notation from Section 3, if M_C and M_K are the complex and Komplex preconditioners respectively, then $M_K = f(M_C)$. Thus, what we are comparing are the differences due to having a complex inner product over n -space versus a real inner product over $2n$ -space. Note that as the quality of the preconditioner improves, the difference in iteration counts between the two approach diminishes.

Our Matlab results did not have any relevant solution time statistics, so we cannot get precise measurement of relative costs. However, the results in Table 3 show that the higher quality preconditioners also provide the best time to solution. These results are from the Komplex Solver Package [9], an add-on module to Aztec. We used a Block ILU preconditioner with 8-by-8 blocks and non-restarted GMRES. The best time to solution comes from BILU with a level fill of 2. This result suggests a general observation that a high quality preconditioner provides both the best time to solution and makes the difference in iteration counts between the true complex and the K formulations minimal.

The final set of results (in Table 4) comes from using the Komplex Solver package to solve linear stability problems in computational fluid dynamics. The primary purpose of these results is to illustrate that, with minimal new software development, we are able to provide a full-featured parallel preconditioned iterative solver for complex valued linear systems by leveraging existing real valued solvers. Results are given for 1, 2 and 4 processors of a PC-based Beowulf [10] cluster.

<i>Problem</i>	$nz(ILU)/nz(A)$	$\ r\ /\ b\ $	<i># Proc</i>	<i>K Iters</i>	<i>Time(s)</i>
<i>LINSTAB1</i>	4.8	4×10^{-13}	1	61	115.0
<i>LINSTAB1</i>			2	58	51.4
<i>LINSTAB1</i>			4	67	35.8
<i>LINSTAB2</i>	3.0	3×10^{-12}	1	73	79.0
<i>LINSTAB2</i>			2	67	34.8
<i>LINSTAB2</i>			4	72	29.1

Table 4: Aztec Test Results Using GMRES(∞) with ILUT Preconditioning.

5 Conclusions

In this paper we presented a discussion of how to solve complex valued linear systems via equivalent real forms. We listed approaches that failed, and presented the K formulation which works very well. Although it is clear from our results that the K formulation is not superior to a true complex solver, and clearly if you have easy access to a complex valued solver you should use it, we do think that equivalent real forms should receive more attention than they have in the past.

For many challenging problems, a high-quality preconditioner is a requirement for convergence. Such a preconditioner has the tendency to map the spectrum around the point (1,0) in the complex plane. This in turn, as the analysis in Section 3.1 shows, minimizes the spectral difference between a true complex valued iterative solver and the K formulation, and leads to our observation that the requirement of a high quality preconditioner simultaneously provides the best solution times and diminishes the convergence differences between a true complex iterative solver and the K formulation.

Finally, for application such as linear stability analysis where all operators are real valued except for the presence of complex shift value, the equivalent real form can be very attractive since it utilizes the native real valued solver and installation of the K formulation usually involves minimal extra effort on the part of the application developer.

References

- [1] S. Balay, W. Gropp, L. McInnes, and B. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhauser Press, 1997.
- [2] S. Balay, W. Gropp, L. McInnes, and B. Smith. PETSc 2.0 users manual. Technical Report ANL-95/11 - Revision 2.0.22, Argonne National Laboratory, 1998.
- [3] S. Balay, W. Gropp, L. McInnes, and B. Smith. PETSc home page. <http://www.mcs.anl.gov/petsc>, 1998.
- [4] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the solution of linear systems : building blocks for iterative methods*. SIAM, Philadelphia, PA, USA, 1994.
- [5] B. Fischer. *Polynomial based iteration methods for symmetric linear systems*. Wiley Teubner, 1st edition, 1996.
- [6] R. Freund and N. Nachtigal. QMRPACK: a package of QMR algorithms. *ACM Trans. Math. Softw.*, 22(1):46–77, March 1996.
- [7] R.W. Freund. Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices. *SIAM J. Sci. Comput.*, 13:425–448, 1992.
- [8] G. Golub and C. Van Loan. *Matrix computations*. The Johns Hopkins University Press, 3rd edition, 1996.
- [9] Michael A. Heroux. *The Komplex Solver Package Reference Manual 1.0*. Sandia National Laboratories, Albuquerque, NM 87185, 2000.
- [10] Phil Merkey. Beowulf home page. <http://beowulf.gsfc.nasa.gov>, 1999.
- [11] Y. Saad. *Iterative methods for sparse linear systems*. PWS Publishing Company, 1st edition, 1996.
- [12] Ray S. Tuminaro, Michael A. Heroux Scott. A. Hutchinson, and J. N. Shadid. *Official Aztec User's Guide, Version 2.1*. Sandia National Laboratories, Albuquerque, NM 87185, 1999.