Simultaneous preconditioning and symmetrization of non-symmetric linear systems

Nassif Ghoussoub* and Amir Moradifam †
Department of Mathematics, University of British Columbia,
Vancouver BC Canada V6T 1Z2
nassif@math.ubc.ca
a.moradi@math.ubc.ca

January 24, 2008

Abstract

Motivated by the theory of self-duality which provides a variational formulation and resolution for non self-adjoint partial differential equations [6, 7], we propose new templates for solving large non-symmetric linear systems. The method consists of combining a new scheme that simultaneously preconditions and symmetrizes the problem, with various well known iterative methods for solving linear and symmetric problems. The approach seems to be efficient when dealing with certain ill-conditioned, and highly non-symmetric systems.

1 Introduction and main results

Many problems in scientific computing lead to systems of linear equations of the form,

$$Ax = b \text{ where } A \in \mathbb{R}^{n\times n} \text{ is a nonsingular but sparse matrix, and } b \text{ is a given vector in } \mathbb{R}^n, \tag{1}$$

and various iterative methods have been developed for a fast and efficient resolution of such systems. The Conjugate Gradient Method (CG) which is the oldest and best known of the nonstationary iterative methods, is highly effective in solving symmetric positive definite systems. For indefinite matrices, the minimization feature of CG is no longer an option, but the Minimum Residual (MINRES) and the Symmetric LQ (SYMMLQ) methods are often computational alternatives for CG, since they are applicable to systems whose coefficient matrices are symmetric but possibly indefinite.

The case of non-symmetric linear systems is more challenging, and again methods such as CGNE, CGNR, GMRES, BiCG, QMR, CGS, and Bi-CGSTAB have been developed to deal with these situations (see the survey books [9] and [11]). One approach to deal with the non-symmetric case, consists of reducing the problem to a symmetric one to which one can apply the above mentioned schemes. The one that is normally used consists of simply applying CG to the normal equations

$$A^TAx = A^Tb \quad \text{or} \quad AA^Ty = b, \quad x = A^Ty. \tag{2}$$

It is easy to understand and code this approach, and the CGNE and CGNR methods are based on this idea. However, the convergence analysis of these methods depends closely on the *condition number* of the matrix under study. For a general matrix A, the condition number is defined as

$$\kappa(A) = \|A\| \cdot \|A^{-1}\|, \tag{3}$$

*Partially supported by a grant from the Natural Sciences and Engineering Research Council of Canada.
†Partially supported by a UBC Graduate Fellowship.

and in the case where A is positive definite and symmetric, the condition number is then equal to

$$\tilde{\kappa}(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}, \tag{4}$$

where $\lambda_{\min}(A)$ (resp., $\lambda_{\max}(A)$) is the smallest (resp., largest) eigenvalue of A). The two expressions can be very different for non-symmetric matrices, and these are precisely the systems that seem to be the most pathological from the numerical point of view. Going back to the crudely symmetrized system (2), we echo Greenbaum's statement [9] that numerical analysts *cringe* at the thought of solving these normal equations because the *condition number* (see below) of the new matrix $A^T A$ is the square of the condition number of the original matrix A.

In this paper, we shall follow a similar approach that consists of symmetrizing the problem so as to be able to apply CG, MINRES, or SYMMLQ. However, we argue that for a large class of non-symmetric, ill-conditionned matrices, it is sometimes beneficial to replace problem (1) by one of the form

$$A^T MAx = A^T Mb, \tag{5}$$

where M is a symmetric and positive definite matrix that can be chosen properly so as to obtain good convergence behavior for CG when it is applied to the resulting symmetric $A^T MA$. This reformulation should not only be seen as a symmetrization, but also as preconditioning procedure. While it is difficult to obtain general conditions on M that ensure higher efficiency by minimizing the condition number $k(A^T MA)$, we shall show theoretically and numerically that by choosing M to be either the inverse of the symmetric part of A, or its resolvent, one can get surprisingly good numerical schemes to solve (1).

The basis of our approach originates from the selfdual variational principle developed in [6, 7] to provide a variational formulation and resolution for non self-adjoint partial differential equations that do not normally fit in the standard Euler-Lagrangian theory. Applied to the linear system (1), the new principle yields the following procedure. Split the matrix A into its symmetric A_a (resp., anti-symmetric part A_a)

$$A = A_s + A_a, \tag{6}$$

where

$$A_s := \frac{1}{2}(A + A^T) \quad \text{and} \quad A_a := \frac{1}{2}(A - A^T). \tag{7}$$

Proposition 1.1 (Selfdual symmetrization) *Assume the matrix A is positive definite, i.e., for some $\delta > 0$,*

$$\langle Ax, x\rangle \geq \delta |x|^2 \textit{ for all } x \in \mathbb{R}^n. \tag{8}$$

The convex continuous functional

$$I(x) = \frac{1}{2}\langle Ax, x\rangle + \frac{1}{2}\langle A_s^{-1}(b - A_a x), b - A_a x\rangle - \langle b, x\rangle \tag{9}$$

then attains its minimum at some $\bar{x}$ in $\mathbb{R}^n$, in such a way that

$$\begin{aligned} I(\bar{x}) &= \inf_{x\in\mathbb{R}^n} I(x) = 0 \qquad (10)\\ A\bar{x} &= b. \qquad (11) \end{aligned}$$

Symmetrization and preconditioning via selfduality: Note that the functional I can be written as

$$I(x) = \frac{1}{2}\langle \tilde{A}x, x\rangle + \langle A_a A_s^{-1} b - b, x\rangle + \frac{1}{2}\langle A_s^{-1} b, b\rangle, \tag{12}$$

where

$$\tilde{A} := A_s - A_a A_s^{-1} A_a = A^T A_s^{-1} A. \tag{13}$$

By writing that $DI(\bar{x}) = 0$, one gets the following equivalent way of solving (1).

If both $A \in \mathbb{R}^{n\times n}$ and its symmetric part A_s are nonsingular, then x is a solution of the equation (1) if and only if it is a solution of the linear symmetric equation

$$A^T A_s^{-1} Ax = (A_s - A_a A_s^{-1} A_a)x = b - A_a A_s^{-1} b = A^T A_s^{-1} b. \tag{14}$$

One can therefore apply to (14) all known iterative methods for symmetric systems to solve the non-symmetric linear system (1). As mentioned before, the new equation (14) can be seen as a new symmetrization of problem (1) which also preserves positivity, i.e., $A^T A_s^{-1} A$ is positive definite if A is. This will then allow for the use of the Conjugate Gradient Method (CG) for the functional I. More important and less obvious than the symmetrization effect of $\tilde{A}$, is our observation that for a large class of matrices, the convergence analysis on the system (14) is often more favorable than the original one. The Conjugate Gradient method –which can now be applied to the symmetrized matrix $\tilde{A}$– has the potential of providing an efficient algorithm for resolving non-symmetric linear systems. We shall call this scheme the *Self-Dual Conjugate Gradient for Non-symmetric matrices* and we will refer to it as SD-CGN.
As mentioned above, the convergence analysis of this method depends closely on the condition number $k(\tilde{A})$ of $\tilde{A} = A^T A_s^{-1} A$ which in this case is equal to $\tilde{k}(\tilde{A})$. We observe in section 2.3 that even though $k(\tilde{A})$ could be as large as the square of $k(A_s)$, it is still much smaller that the condition number of the original matrix $\kappa(A)$. In other words, the inverse C of $A^T A_s^{-1}$ can be an efficient preconditioning matrix, in spite of the additional cost involved in finding the inverse of A_s. Moreover, the efficiency of C seems to surprisingly improve in many cases as the norm of the anti-symmetric part gets larger (Proposition 2.2). A typical example is when the anti-symmetric matrix A_a is a multiple of the symplectic matrix J (i.e. $JJ^* = -J^2 = I$). Consider then a matrix $A_\epsilon = A_s + \frac{1}{\epsilon} J$ which has an arbitrarily large anti-symmetric part. One can show that

$$\kappa(\tilde{A}_\epsilon) \leq \kappa(A_s) + \epsilon^2 \lambda_{\max}(A_s)^2, \tag{15}$$

which means that the larger the anti-symmetric part, the more efficient is our proposed selfdual preconditioning. Needless to say that this method is of practical interest only when the equation $A_s x = d$ can be solved with less computational effort than the original system, which is not always the case.
Now the relevance of this approach stems from the fact that conjugate gradient methods for nonsymmetric systems are costly since they require the storage of previously calculated vectors. It is however worth noting that Concus and Golub [3] and Widlund [15] have also proposed another way to combine CG with a preconditioning using the symmetric part A_s, which does not need this extended storage. Their method has essentially the same cost per iteration as the preconditioning with the inverse of $A^T A_s^{-1}$ that we propose for SD-CGN and both schemes converge to the solution in at most N iterations.

Iterated preconditioning: Another way to see the relevance of A_s as a preconditioner, is by noting that the convergence of "simple iteration"

$$A_s x_k = -A_a x_{k-1} + b \tag{16}$$

applied to the decomposition of A into its symmetric and anti-symmetric parts, requires that the spectral radius $\rho(I - A_s^{-1}A) = \rho(A_s^{-1}A_a) < 1$. By multiplying (16) by A_s^{-1}, we see that this is equivalent to the process of applying simple iteration to the original system (1) conditioned by A_s^{-1}, i.e., to the system

$$A_s^{-1} Ax = A_s^{-1} b. \tag{17}$$

On the other hand, "simple iteration" applied to the decomposition of $\tilde{A}$ into A_s and $A_a A_s^{-1} A_a$ is given by

$$A_s x_k = A_a A_s^{-1} A_a x_{k-1} + b - A_a A_s^{-1} b. \tag{18}$$

Its convergence is controlled by $\rho(I - A_s^{-1}\tilde{A}) = \rho((A_s^{-1}A_a)^2) = \rho(A_s^{-1}A_a)^2$ which is strictly less than $\rho(A_s^{-1}A_a)$, i.e., an improvement when the latter is strictly less than one, which the mode in which we have convergence. In other words, the linear system (14) can still be preconditioned one more time as follows:

If both $A \in \mathbb{R}^{n\times n}$ and its symmetric part A_s are nonsingular, then x is a solution of the equation (1) if and only if it is a solution of the linear symmetric equation

$$\bar{A}x := A_s^{-1} A^T A_s^{-1} Ax = [I - (A_s^{-1}A_a)^2]x = (I - A_s^{-1}A_a)A_s^{-1}b = A_s^{-1} A^T A_s^{-1} b. \tag{19}$$

Note however that with this last formulation, one has to deal with the potential loss of positivity for the matrix $\tilde{A}$.

Anti-symmetry in transport problems: Numerical experiments on standard linear ODEs (Example 3.1) and PDEs (Example 3.2), show the efficiency of SD-CGN for non-selfadjoint equations. Roughly speeking, discretization of differential equations normally leads to a symmetric component coming from the Laplace operator, while the discretization of the non-self-adjoint part leads to the anti-symmetric part of the coefficient matrix. As such, the symmetric part of the matrix is of order $O(\frac{1}{h^2})$, while the anti-symmetric part is of order $O(\frac{1}{h})$, where h is the step size. The coefficient matrix A in the original system (1) is therefore an $O(h)$ perturbation of its symmetric part. However, for the new system (14) we have roughly

$$\tilde{A} = A_s - A_a A_s^{-1} A_a = O(\frac{1}{h^2}) - O(\frac{1}{h})O(h^2)O(\frac{1}{h}) = O(\frac{1}{h^2}) - O(1), \tag{20}$$

making the matrix $\tilde{A}$ an $O(1)$ perturbation of A_s, and therefore a matrix of the form $A_s + \alpha I$ becomes a natural candidate to precondition the new system (14).

Resolvents of A_s as preconditioners: One may therefore consider preconditioned equations of the form $A^T M A x = A^T M b$, where M is of the form

$$M_\alpha = \big(\alpha A_s + (1-\alpha)I\big)^{-1} \quad \text{or} \quad N_\beta = \beta A_s^{-1} + (1-\beta)I, \tag{21}$$

for some $0 \le \alpha, \beta \in \mathbb{R}$, and where I is the unit matrix.
Note that we obviously recover (2) when $\alpha = 0$, and (14) when $\alpha = 1$. As $\alpha \to 0$ the matrix $\alpha A_s + (1-\alpha)I$ becomes easier to invert, but the matrix

$$A_{1,\alpha} = A^T(\alpha A_s + (1-\alpha)I)^{-1}A \tag{22}$$

may become more ill conditioned, eventually leading (for $\alpha = 0$) to $A^T A x = A^T b$. There is therefore a trade-off between the efficiency of CG for the system (5) and the condition number of the inner matrix $\alpha A_s + (1-\alpha)I$, and so by an appropriate choice of the parameter α we may minimize the cost of finding a solution for the system (1). In the case where A_s is positive definite, one can choose –and it is sometimes preferable as shown in example (3.4)– $\alpha > 1$, as long as $\alpha < \frac{1}{1-\lambda^s_{\min}}$, where $\lambda^s_{\min}$ is the smallest eigenvalue of A_s. Moreover, in the case where the matrix A is not positive definite or if its symmetric part is not invertible, one may take α small enough, so that the matrix M_α (and hence $A_{1,\alpha}$) becomes positive definite, and therefore making CG applicable (See example 3.4). Similarly, the matrix $N_\beta = \beta A_s^{-1} + (1-\beta)I$ provides another choice for the matrix M in (5), for $\beta < \frac{\lambda^s_{\max}}{\lambda^s_{\max}-1}$ where $\lambda^s_{\max}$ is the largest eigenvalue of A_s. Again we may choose α close to zero to make the matrix N_β positive definite. As we will see in the last section, appropriate choices of β, can lead to better convergence of CG for equation (5).
One can also combine both effects by considering matrices of the form

$$L_{\alpha,\beta} = \big(\alpha A_s + (1-\alpha)I\big)^{-1} + \beta I, \tag{23}$$

as is done in example (3.4).
We also note that the matrices $M'_\alpha := (\alpha A'_s + (1-\alpha)I)^{-1}$ and $N'_\beta := \beta(A'_s)^{-1} + (1-\beta)I$ can be other options for the matrix M, where A'_s is a suitable approximation of A_s, chosen is such a way that $M'_\alpha q$ and $N'_\beta q$ can be relatively easier to compute for any given vector q.
Finally, we observe that the above reasoning applies to any decomposition $A = B + C$ of the non-singular matrix $A \in \mathbb{R}^{n\times n}$, where B and $(B-C)$ are both invertible. In this case, $B(B-C)^{-1}$ can be a preconditioner for the equation (1). Indeed, since $B - CB^{-1}C = (B-C)B^{-1}A$, x is a solution of (1) if and only of it is a solution of the system

$$(B-C)B^{-1}Ax = (B - CB^{-1}C)x = b - CB^{-1}b. \tag{24}$$

In the next section, we shall describe a general framework based on the ideas explained above for the use of iterative methods for solving non-symmetric linear systems. In section 3 we present various numerical experiments to test the effectiveness of the proposed methods.

2 Selfdual methods for non-symmetric systems

By *selfdual methods* we mean the ones that consist of first associating to problem (1) the equivalent system (5) with appropriate choices of M, then exploiting the symmetry of the new system by using the various existing iterative methods for symmetric systems such as CG, MINRES, and SYMMLQ, leading eventually to the solution of the original problem (1). In the case where the matrix M is positive definite, one can then use CG on the equivalent system (5). This scheme (SD-CGN) is illustrated in Table (1) below, in the case where the matrix M is chosen to be the inverse of the symmetric part of A. If M is not positive definite, then one can use MINRES (or SYMMLQ) to solve the system (14). We will then refer to them as SD-MINRESN (i.e., Self-Dual MINRES for Nonsymmetric linear equations).

2.1 Exact methods

In each iteration of CG, MINRES, or SYMMLQ, one needs to compute Mq for certain vectors q. Since selfdual methods call for a conditioning matrix M that involves inverting another one, the computation of Mq can therefore be costly, and therefore not necessarily efficient for all linear equations. But as we will see in section 3, M can sometimes be chosen so that computing Mq is much easier than solving the original equation itself. This is the case for example when the symmetric part is either diagonal or tri-diagonal, or when we are dealing with several linear systems all having the same symmetric part, but with different anti-symmetric components. Moreover, one need not find the whole matrix M, in order to compute Mq. The following scheme illustrates the exact SD-CGN method applied in the case where the coefficient matrix A in (1) is positive definite, and when $A^T(A_s)^{-1}Aq$ can be computed exactly for any given vector q.

```
Given an initial guess $x_0$,
Solve $A_s y = b$
Compute $\overline{b} = b - A_a y$.
Solve $A_s y_0 = A_a x_0$
Compute $r_0 = \overline{b} - A_s x_0 + A_a y_0$ and set $p_0 = r_0$.
For k=1,2, . . . ,
Solve $A_s z = A_a p_{k-1}$
Compute $w = A_s p_{k-1} - A_a z$ .
Set $x_k = x_{k-1} + \alpha_{k-1} p_{k-1}$, where $\alpha_{k-1} = \frac{<r_{k-1},r_{k-1}>}{<p_{k-1},w>}$ .
Cpmpute $r_k = r_{k-1} - \alpha_{k-1} w$.
Set $p_k = r_k + b_{k-1} p_{k-1}$, where $b_{k-1} = \frac{<r_k,r_k>}{<r_{k-1},r_{k-1}>}$ .
Check convergence; continue if necessary.
```

Table 1: GCGN

In the case where A is not positive definite, or when it is preferable to choose a non-positive definite conditioning matrix M, then one can apply MINRES or SYMMLQ to the equivalent system (5). These schemes will be then called SD-MINRESN and SD-SYMMLQN respectively.

2.2 Inexact Methods

The SD-CGN, SD-MINRESN and SD-SYMMLQN are of practical interest when for example, the equation

$$A_s x = q \tag{25}$$

can be solved with less computational effort than the original equation (1). Actually, one can use CG, MINRES, or SYMMLQ to solve (25) in every iteration of SD-CGN, SD-MINRESN, or SD-SYMMLQN. But since each sub-iteration may lead to an error in the computation of (25), one needs to control such errors, in order for the method to lead to a solution of the system (1) with the desired tolerance. This leads to the Inexact SD-CGN, SD-MINRESN and SD-SYMMLQN methods (denoted below by ISD-CGN, ISD-MINRESN and ISD-SYMMLQN respectively).

The following proposition –which is a direct consequence of Theorem 4.4.3 in [9]– shows that if we solve the inner equations (25) "accurately enough" then ISD-CGN and ISD-MINRESN can be used to solve (1) with a pre-determined accuracy. Indeed, given $\epsilon > 0$, we assume that in each iteration of ISD-CGN or ISD-MINRESN, we can solve the inner equation –corresponding to A_s– accurately enough in such a way that

$$\|(A_s - A_a A_s^{-1} A_a)p - (A_s p - A_a y)\| = \|A_a A_s^{-1} A_a p - A_a y\| < \epsilon, \tag{26}$$

where y is the (inexact) solution of the equation

$$A_s y = A_a p. \tag{27}$$

In other words, we assume CG and MINRES are implemented on (27) in a finite precision arithmetic with machine precision ϵ. Set

$$\epsilon_0 := 2(n+4)\epsilon, \quad \epsilon_1 := 2(7 + n\frac{\| \, |A_s - A_a A_s^{-1} A_a| \, \|}{\|A_s - A_a A_s^{-1} A_a\|})\epsilon, \tag{28}$$

where $|D|$ denotes the matrix whose terms are the absolute values of the corresponding terms in the matrix D. Let $\lambda_1 \leq ... \leq \lambda_n$ be the eigenvalues of $(A_s - A_a A_s^{-1} A_a)$ and let $T_{k+1,k}$ be the $(k+1) \times k$ tridiagonal matrix generated by a finite precision Lanczos computation. Suppose that there exists a symmetric tridiagonal matrix T, with $T_{k+1,k}$ as its upper left $(k+1) \times k$ block, whose eigenvalues all lie in the intervals

$$S = \cup_{i=1}^{n} [\lambda_i - \delta, \lambda_i + \delta], \tag{29}$$

where none of the intervals contain the origin. let d denote the distance from the origin to the set S, and let p_k denote a polynomial of degree k.

Proposition 2.1 *The ISD-MINRESN residual r_k^{IM} then satisfies*

$$\frac{||r_k^{IM}||}{||r_0||} \leq \sqrt{(1+2\epsilon_0)(k+1)} \quad \min_{p_k} \quad \max_{z=S} |p_k(z)| + 2\sqrt{k}(\frac{\lambda_n}{d})\epsilon_1. \tag{30}$$

If A is positive definite, then the ISD-CGN residual r^{IC} satisfies

$$\frac{||r_k^{IC}||}{||r_0||} \leq \sqrt{(1+2\epsilon_0)(\lambda_n + \delta)/d} \quad \min_{p_k} \quad \max_{z=S} |p_k(z)| + \sqrt{k}(\frac{\lambda_n}{d})\epsilon_1. \tag{31}$$

It is shown by Greenbaum [6] that $T_{k+1,k}$ can be extended to a larger symmetric tridiagonal matrix T whose eigenvalues all lie in tiny intervals about the eigenvalues of $(A_s - A_a A_s^{-1} A_a)$. Hence the above proposition guarantees that if we solve the inner equations accurate enough, then ISD-CGN and ISD-MINRESN converges to the solution of the system 1 with the desired relative residual (see the last section for numerical experiments).

2.3 Preconditioning

As mentioned in the introduction, the convergence of iterative methods depends heavily on the spectral properties of the coefficient matrix. Preconditioning techniques attempt to transform the linear system (1) into an equivalent one of the form $C^{-1}Ax = C^{-1}b$, in such a way that it has the same solution, but hopefully with more favorable spectral properties. As such the reformulation of (1) as

$$A^T A_s^{-1} A x = A^T A_s^{-1} b, \tag{32}$$

can be seen as a preconditioning procedure with C being the inverse of $A^T A_s^{-1}$. The spectral radius, and more importantly the condition number of the coefficient matrix in linear systems, are crucial parameters for the convergence of iterative methods. The following simple proposition gives upper bounds on the condition number of $\tilde{A} = A^T A_s^{-1} A$.

Proposition 2.2 *Assume A is an invertible positive definite matrix, then*

$$\kappa(\tilde{A}) \leq \min\{\kappa_1, \kappa_2\}, \tag{33}$$

where

$$\kappa_1 := \kappa(A_s) + \tfrac{\|A_a\|^2}{\lambda_{\min}(A_s)^2} \quad \textit{and} \quad \kappa_2 := \kappa(A_s)\kappa(-A_a^2) + \tfrac{\lambda_{\max}(A_s)^2}{\lambda_{\min}(-A_a^2)}. \tag{34}$$

Proof: We have

$$\lambda_{min}(\tilde{A}) = \lambda_{min}(A_s - A_a A_s^{-1} A_a) \geq \lambda_{min}(A_s).$$

We also have

$$\begin{aligned}\lambda_{max}(\tilde{A}) &= \sup_{x\neq 0} \frac{x^t \tilde{A} x}{|x|^2} = \sup_{x\neq 0} \frac{x^t(A_s - A_a A_s^{-1} A_a)x}{|x|^2} \\ &\leq \lambda_{max}(A_s) + \frac{||A_a||^2}{\lambda_{min}(A_s)}.\end{aligned}$$

Since $\kappa(\tilde{A}) = \frac{\lambda_{max}(\tilde{A})}{\lambda_{min}(\tilde{A})}$, it follows that $\kappa(\tilde{A}) \leq \kappa_1$.
To obtain the second estimate, observe that

$$\begin{aligned}\lambda_{min}(\tilde{A}) &= \lambda_{min}(A_s - A_a A_s^{-1} A_a) > \lambda_{min}(-A_a A_s^{-1} A_a) \\ &= \inf_{x\neq 0} \frac{-x^T A_a A_s^{-1} A_a x}{x^T x} \\ &= \inf_{x\neq 0} \{\frac{(A_a x)^T A_s^{-1}(A_a x)}{(A_a x)^T (A_a x)} \times \frac{(A_a x)^T (A_a x)}{x^T x}\} \\ &\geq \inf_{x\neq 0} \frac{(A_a x)^T A_s^{-1}(A_a x)}{(A_a x)^T (A_a x)} \times \inf_{x\neq 0} \frac{x^T (A_a)^T (A_a) x}{x^T x} \\ &= \frac{1}{\lambda_{max}(A_s)} \times \lambda_{min}((A_a)^T A_a) \\ &= \frac{1}{\lambda_{max}(A_s)} \times \lambda_{min}(-A_a^2)\end{aligned}$$

With the same estimate for $\lambda_{max}(\tilde{A})$ we get $\kappa(\tilde{A}) \leq \kappa_2$.

Remark 2.1 Inequality (33) shows that SD-CGN and SD-MINRES can be very efficient schemes for a large class of ill conditioned non-symmetric matrices, even those that are almost singular and with arbitrary large condition numbers. It suffices that either κ_1 or κ_2 be small. Indeed,

- The inequality $\kappa(\tilde{A}) \leq \kappa_1$ shows that the condition number $\kappa(\tilde{A})$ is reasonable as long as the anti-symmetric part A_a is not too large. On the other hand, even if $\|A_a\|$ is of the order of $\lambda_{\max}(A_s)$, and $\kappa(\tilde{A})$ is then as large as $\kappa(A_s)^2$, it may still be an improved situation, since this can happen for cases when $\kappa(A)$ is exceedingly large. This can be seen in example 2.2 below.

- The inequality $\kappa(\tilde{A}) \leq \kappa_2$ is even more interesting especially in situations when $\lambda_{\min}(-A_a^2)$ is arbitrarily large while remaining of the same order as $||A_a||^2$. This means that $\kappa(\tilde{A})$ can remain of the same order as $\kappa(A_s)$ regardless how large is A_a.

  A typical example is when the anti-symmetric matrix A_a is a multiple of the symplectic matrix J (i.e. $JJ^* = -J^2 = I$). Consider then a matrix $A_\epsilon = A_s + \frac{1}{\epsilon}J$ which has an arbitrarily large anti-symmetric part. By using that $\kappa(\tilde{A}) \leq \kappa_2$, one gets

  $$\kappa(\tilde{A}_\epsilon) \leq \kappa(A_s) + \epsilon^2 \lambda_{\max}(A_s)^2. \tag{35}$$

Here are other examples where the larger the condition number of A is, the more efficient is the proposed selfdual preconditioning.

Example 2.2 Consider the matrix

$$A_\epsilon = \begin{bmatrix} 1 & -1 \\ 1 & -1+\epsilon \end{bmatrix} \tag{36}$$

which is a typical example of an ill-conditioned non-symmetric matrix. One can actually show that $\kappa(A_\epsilon) = O(\frac{1}{\epsilon}) \to \infty$ as $\epsilon \to 0$ with respect to any norm. However, the condition number of the associated selfdual coefficient matrix

$$\tilde{A_\epsilon} = A_s - A_a(A_s)^{-1}A_a = \begin{bmatrix} \frac{\epsilon}{\epsilon-1} & 0 \\ 0 & \epsilon \end{bmatrix}$$

is $\kappa(\tilde{A_\epsilon}) = \frac{1}{1-\varepsilon}$, and therefore goes to 1 as $\varepsilon \to 0$. Note also that the condition number of the symmetric part of A_ϵ goes to one as $\epsilon \to 0$. In other words, the more ill-conditioned problem (1) is, the more efficient the selfdual conditioned system (14) is.

We also observe that $\kappa(A_s^{-1}A)$ goes to ∞ as ϵ goes to zero, which means that besides making the problem symmetric, our proposed conditioned matrix $A^TA_s^{-1}A$ has a much smaller condition number than the matrix $A_s^{-1}A$, which uses A_s as a preconditioner.

Similarly, consider the non-symmetric linear system with coefficient matrix

$$A_\epsilon = \begin{bmatrix} 1 & -1+\epsilon \\ 1 & -1 \end{bmatrix}. \tag{37}$$

As $\epsilon \to 0$, the matrix becomes again more and more ill-conditioned, while the condition number of its symmetric part converges to one. Observe now that the condition number of $\tilde{A_\epsilon}$ also converges to 1 as ϵ goes to zero. This example shows that self-doual preconditioning can also be very efficient for non-positive definite problems.

3 Numerical Experiments

In this section we present some numerical examples to illustrate the proposed schemes and to compare them to other known iterative methods for non-symmetric linear systems. Our experiments have been carried out on Matlab (7.0.1.24704 (R14) Service Pack 1). In all cases the iteration was started with $x_0 = 0$.

Example 3.1 *Consider the ordinary differential equation*

$$-\epsilon y'' + y' = f(x), \quad on \quad [0,1], \quad y(0) = y(1) = 0. \tag{38}$$

By discretizing this equation with stepsize $1/65$ *and by using backward difference for the first order term, one obtains a nonsymmetric system of linear equations with 64 unknowns. We present in Table 2 below, the number of iterations needed for various decreasing values of the residual* ϵ*. We use ESD-CGN and ISD-CGN (with relative residual* 10^{-7} *for the solutions of the inner equations). We then compare them to the known methods CGNE, BiCG, QMR, CGS, and BiCGSTAB for solving non-symmetric linear systems. We also test preconditioned version of these methods by using the symmetric part of the corresponding matrix as a preconditioner.*

Table 2: Number of iterations to find a solution with relative residual 10^{-6} for equation (38). $f(x)$ is chosen so that $y = x\sin(\pi x)$ is a solution.

N=64	$\epsilon = 10^{-2}$	$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$	$\epsilon = 10^{-6}$	$\epsilon = 10^{-10}$	$\epsilon = 10^{-16}$
ESD-CGN	22	8	5	4	3	2
ISD-CGN(10^{-7})	24	9	6	4	3	2
GCNE	88	64	64	64	64	64
QMR	114	> 1000	> 1000	> 1000	> 1000	> 1000
PQMR	34	51	50	52	52	52
BiCGSTAB	63.5	78.5	92.5	98.5	100.5	103.5
PBiCGSTAB	26.5	46.5	50.5	50	51.5	51.5
BiCG	125	> 1000	> 1000	> 1000	> 1000	> 1000
PBiCG	31	44	50	50	52	52
CGS	> 1000	> 1000	> 1000	> 1000	> 1000	> 1000
PCGS	27	51	46	46	46	48

Table 3: Number of iterations to find a solution with relative residual 10^{-6} for equation (38). $f(x)$ is chosen so that $y = \frac{x(1-x)}{\cos(x)}$ is a solution, while the stepsize used is 1/129.

N=128	$\epsilon = 10^{-2}$	$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$	$\epsilon = 10^{-6}$	$\epsilon = 10^{-10}$	$\epsilon = 10^{-16}$
ESD-CGN	37	11	6	4	3	2
ISD-CGN(10^{-7})	38	12	7	4	3	2
GCNE	266	140	128	128	128	128
QMR	> 1000	> 1000	> 1000	> 1000	> 1000	> 1000
PQMR	40	77	87	92	90	85
BiCGSTAB	136.5	167.5	241	226.5	233.5	237.5
PBiCGSTAB	35.5	87.5	106.5	109	110.5	110.5
BiCG	> 1000	> 1000	> 1000	> 1000	> 1000	> 1000
PBiCG	37	76	84	89	85	91
CGS	> 1000	> 1000	> 1000	> 1000	> 1000	> 1000
PCGS	34	80	96	91	94	90

As we see in Tables 2 and and 3, a phenomenon similar to Example 2.2 is occuring. As the problem gets harder (ϵ smaller), SD-CGN becomes more efficient. These results can be compared with the number of iterations that the HSS iteration method needs to solve equation (38) (Tables 3,4, and 5 in [2]).

Example 3.2 *Consider the partial differential equation*

$$-\Delta u + a(x,y)\frac{\partial u}{\partial x} = f(x,y), \quad 0 \le x \le 1, \quad 0 \le y \le 1, \tag{39}$$

with Dirichlet boundary condition.

The number of iterations that ESD-CGN and ISD-CGN needed to find a solution with relative residual 10^{-6}, are presented in Table 4 below for different coefficients $a(x,y)$.

Table 4: Number of iterations (I) for the backward scheme method to find a solution with relative residual 10^{-6} for equation (39) (Example 3.2)

a(x,y)	N	I (ESD-CGN)	I (ISD-CGN)	Solution
100	49	18	18	random
100	225	40	37	random
100	961	44	46	random
100	961	52	51	$\sin \pi x \sin \pi y . \exp((x/2+y)^3)$
1000	49	10	10	random
1000	225	31	31	random
1000	961	36	37	random
1000	961	31	39	$\sin \pi x \sin \pi y . \exp((x/2+y)^3)$
10^6	49	4	4	random
10^6	225	6	6	random
10^6	961	6	6	random
10^6	961	6	6	$\sin \pi x \sin \pi y . \exp((x/2+y)^3)$
10^{16}	961	2	2	$\sin \pi x \sin \pi y . \exp((x/2+y)^3)$

Table 5: Number of iterations (I) for the centered difference scheme method for equation (39) (Example 3.2)

a(x,y)	N	I (ESD-CGN)	Solution	Relative Residoual
1	49	21	random	6.71×10^{-6}
1	225	73	random	9.95×10^{-6}
1	961	91	random	8.09×10^{-6}
1	961	72	$\sin \pi x \sin \pi y . \exp((x/2+y)^3)$	9.70×10^{-6}
10	49	18	random	9.97×10^{-6}
10	225	65	random	5.90×10^{-6}
10	961	78	random	8.95×10^{-6}
10	961	65	$\sin \pi x \sin \pi y . \exp((x/2+y)^3)$	7.78×10^{-6}
100	49	31	random	6.07×10^{-6}
100	225	42	random	5.20×10^{-6}
100	961	43	random	5.03×10^{-6}
100	961	38	$\sin \pi x \sin \pi y . \exp((x/2+y)^3)$	4.69×10^{-6}
1000	49	65	random	4.54×10^{-6}
1000	225	130	random	8.66×10^{-6}
1000	961	140	random	2.12×10^{-6}
100	961	150	$\sin \pi x \sin \pi y . \exp((x/2+y)^3)$	5.98×10^{-6}

Table 4 and 5 can be compared with Table 1 in [15], where Widlund had tested his Lanczos method for non-symmetric linear systems. Comparing Table 5 with Table 1 in [15] we see that for small $a(x,y)$ (1 and 10) Widlund's method is more efficient than SD-CGN, but for large values of a, SD-CGN turns out to be more efficient than Widlund's Lanczos method.

Remark 3.3 *As we see in Tables 2,3, and 4, the number of iterations for ESD-CGN and ISD-CGN (with relative residual* 10^{-7} *for the solutions of the inner equations) are almost the same One might choose dynamic relative residuals for the solutions of inner equations to decrease the average cost per iterations of ISD-CGN. It is interesting to figure out whether there is a procedure to determine the accuracy of solutions for the inner equations to minimize the total cost of finding a solution.*

Example 3.4 *Consider the partial differential equation*

$$-\Delta u + 10\frac{\partial(\exp(3.5(x^2+y^2)u)}{\partial x} + 10\exp(3.5(x^2+y^2))\frac{\partial u}{\partial x} = f(x), \qquad on\ \ [0,1]\times[0,1], \tag{40}$$

with Dirichlet boundary condition, and choose f so that $\sin(\pi x)\sin(\pi y)\exp((x/2+y)^3)$ is the solution of the equation. We take the stepsize $h = 1/31$ which leads to a linear system $Ax = b$ with 900 unknowns. Table 5 includes the number of iterations which CG needs to converge to a solution with relative residual 10^{-6} when applied to the preconditioned matrix

$$A^T(\alpha A_s^{-1} + (1-\alpha)I)A. \tag{41}$$

Table 5 can be compared with Table 1 in [15], where Widlund has presented the number of iterations needed to solve equation (40).

Table 6: Number of iterations for a solution with relative residual 10^{-6} for example 3.3 when SD-CGN is used with the preconditioner (41) for different values of α.

$\lambda^s_{max}(\frac{1-\alpha}{\alpha})$	I		$\lambda^s_{max}(\frac{1-\alpha}{\alpha})$	I
$\infty(\alpha = 0)$	> 5000		0.1	232
$0(\alpha = 1)$	229		0.2	237
-0.1	221		0.4	249
-0.25	216		0.8	263
-0.5	201		1	272
-0.7	191		5	384
-0.8	186		10	474
-0.9	180		20	642
-0.95	179		50	890
-0.99	177		100	1170
-0.999	180		1000	2790
-0.9999	234		10000	4807

Remark 3.5 *As we see in Table 5, for $\lambda^s_{max}(\frac{1-\alpha}{\alpha}) = -.99$ we have the minimum number of iterations. Actually, this is the case in some other experiments, but for many other system the minimum number of iterations accrues for some other α with $-1 < \lambda^s_{max}(\frac{1-\alpha}{\alpha}) \leq 0$. Our experiments show that for a well chosen $\alpha > 1$, one may considerably decrease the number of iterations. Obtaining theoretical results on how to choose parameter α in 41 seems to be an interesting problem.*

Note that the coefficient matrix of the linear system corresponding to (40) is positive definite. Hence we may also apply CG with the preconditioned symmetric system of equations

$$A^T(A_s - \alpha\lambda^s_{\min}I)^{-1}A = A^T(A_s - \alpha\lambda^s_{min}I)^{-1}b, \tag{42}$$

where $\lambda^s_{\min}$ is the smallest eigenvalue of A_s and $\alpha < 1$. The number of iterations function of α, that CG needs to converges to a solution with relative residual 10^{-6} are presented in Table 7.

Table 7: Number of iterations to find a solution with relative residual 10^{-6} for equation (40) when SD-CGN is used with the preconditioner (42) for different values of α.

α	I
0	229
0.5	204
0.9	177
0.99	166
0.999	168
0.9999	181
0.99999	194
0.999999	222
0.9999999	248
0.99999999	257

Remark 3.6 *As we see in the above table, for $\alpha = 0.99$ in (42) we have the minimum number of iterations. Obtaining theoretical results on how to choose the parameter α seems to be an interesting problem to study.*

We also repeat the experiment by applying CG to the system of equations

$$A^T \left(A_s - 0.99\lambda^s_{\min} I)^{-1} - \frac{0.99}{\lambda^s_{\max}} I \right) A = A^T \left((A_s - o.99\lambda^s_{\min} I)^{-1} - \frac{0.99}{\lambda^s_{\max}} I \right) b. \tag{43}$$

Then CG needs 131 iterations to converge to a solution with relative residual 10^{-6}.
As another experiment we apply CG to the preconditioned linear system

$$A_s^{-1} A^T A_s^{-1} A = A_s^{-1} A^T A_s^{-1} b,$$

to solve the non-symmetric linear system obtained from discritization of the Equation (40). The CG converges in 31 iterations to a solution with relative residual less than 10^{-6}. Since, we need to solve two equations with the coefficient matrix As, the cost of each iteration in this case is towice as much as SD-CGN. So, by the above preconditioning we decrease cost of finding a solution to less that 62/131 of that of SD-CGN (System (43)).

Example 3.7 *Consider now the following equation*

$$-\Delta u + 10 \frac{\partial(\exp(3.5(x^2 + y^2)u)}{\partial x} + 10 \exp(3.5(x^2 + y^2)) \frac{\partial u}{\partial x} - 200u = f(x), \qquad on \;\; [0,1] \times [0,1], \tag{44}$$

If we discretize this equation with stepsize 1/31 and use backward differences for the first order term, we get a linear system of equations $Ax = b$ with A being a non-symmetric and non-positive definite coefficient matrix. We then apply CG to the following preconditioned, symmetrized and positive definite matrix

$$A^T((A_s - \alpha\lambda^s_{\min} I)^{-1} + \beta I)A = A^T((A_s - \alpha\lambda^s_{\min} I)^{-1} + \beta I)b, \tag{45}$$

with $\alpha < 1$. For different values of α the number of iterations which CG needs to converge to a solution with the relative residual 10^{-6} are presented in Table 8.

Table 8: Number of iterations to find a solution with relative residual 10^{-6} for equation (44) when SD-CGN is used with the preconditioner (45) for different values of α and β.

α	$\beta = 0$	$\beta = -.99/\lambda^s_{max}$
10	543	424
5	446	352
2.5	369	288
1.5	342	264
1.1	331	258
1.01	327	259
1.001	333	271
1.0001	368	289
1.00001	401	317

We repeat our experiment with stepsize 1/61 and get a system with 3600 unknowns. With $\alpha = -1.00000001$ and $\beta = 0$, CG converges in one single iteration to a solution with relative residual less than 10^{-6}. We also apply QMR, BiCGSTAB, BiCG, and CGS (also preconditioned with the symmetric part as well) to solve the corresponding system of linear equations with stepsize 1/31. The number of iterations needed to converge to a solution with relative residual 10^{-6} are presented in Table 9.

Table 9: Number of iterations to find a solution with relative residual 10^{-6} for equation (44) using various algorithms.

N=900	I
CGNE	> 5000
QMR	3544
PQMR	490
BiCGSTAB	> 5000
PBiCGSTAB	Breaks down
BiCG	4527
PBiCG	> 1000
CGS	1915
PCGS	649

Acknowledgments: This paper wouldn't have seen the light without the gentle prodding and constant encouragement of Anthony Peirce, and the expert guidance and generous support of Chen Greif. They have our deep and sincere gratitude.

References

[1] O. Axelsson, Z.-Z. Bai, and S.-X. Qiu, *A class of nested iteration schemes for linear systems with a coefficient matrix with a dominant positive definite symmetric part,* Numer. Algorithms, to appear.

[2] Z.-Z. Bai, G. H. Golub, and M. K. NG, Hermitian and skew-Hermitian splitting methods for non-hermitian positive definite linear systems, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 603-626.

[3] P. Concus and G. H. Golub, *A generalized conjugate gradient method for non-symmetric systems of linear equations,* Computing Methods in Applied Sciences and Engineering, Lecture Notes in Econom. and Math. Systems 134, R. Glowinski and J.R. Lions, eds., Springer-Verlag, Berlin, 1976, pp. 56-65; also available online fromh ttp://wwwsccm. stanford.edu.

[4] M. Eiermann, W. Niethammer, and R. S. Varga, *Acceleration of relaxation methods for non-Hermitian linear systems,* SIAM J. Matrix Anal. Appl., 13 (1992), pp. 979-991.

[5] R. Fletcher, *Conjugate gradient methods for indefinite systems,* Lecture Notes in Math., 506 (1976), pp. 73-89.

[6] N. Ghoussoub, *Anti-selfdual Lagrangians: Variational resolutions of non self-adjoint equations and dissipative evolutions*, AIHP-Analyse non linéaire, 24 (2007), 171-205.

[7] N. Ghoussoub, *Selfdual partial differential systems and their variational principles*, Springer-Verlag, Universitext Series, In press (2007) 350 pp.

[8] G. H. Golub and D. Vanderstraeten, *On the preconditioning of matrices with a dominant skew-symmetric component,* Numer. Algorithms, 25 (2000), pp. 223-239.

[9] A. Greenbaum, *Iterative Methods for Solving Linear Systems,* Frontiers Appl. Math. 17, SIAM, Philadelphia, 1997.

[10] J. A. Meijerink and H. A. Van Der Vorst, *An iterative solution method for linear systems of which the coeJficient matrix is a symmetric M-matrix,* Math. Comp., 31 (1977), pp. 148-162.

[11] Y. Saad, *Iterative Methods for Sparse Linear Systems,* PWS Publishing, Boston, 1996.

[12] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,* SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856-869.

[13] P. Sonneveld, *CGS: a fast Lanczos-type solver for nonsymmetric linear systems,* SIAM J. Sci. Statist. Comput., 10 (1989), pp. 36-52.

[14] H. A. Van Der Vorst, *The convergence behaviour of preconditioned CG and CG-S in the presence of rounding errors,* Lecture Notes in Math., 1457 (1990), pp. 126-136.

[15] O. Widlund, *A Lanczos method for a class of nonsymmetric systems of linear equations,* SIAM J. Numer. Anal., 15 (1978), pp. 801-812.