46

Preconditioning of Two-dimensional Singular Integral Equations

Ke Chen

1 Introduction

Numerical solution of integral equations produces dense linear systems that give rise to unsymmetric matrices, in general. In the two-dimensional case, such systems can become too large for direct solution. Here we consider the application of conjugate gradient methods. For singular integral equations, such iterative methods require preconditioning for any convergence.

Preconditioning techniques proposed in the literature, involving sparse matrices, are mostly designed for one-dimensional integral equations, and based on considerations of efficiency. In [Che94] and [Che96], for 1D singular integral equations, we have given a theoretical justification for a class of preconditioners. Here we consider a generalization of this work to the 2D case. Such a theory is based on a suitable splitting of the underlying singular operator. Essentially the domain is divided into many subdomains in order to isolate singularities. Some experiments on Cauchy type bi-singular integral equations are reported. We have applied both the conjugate gradient normal method (CGN) and the generalized minimal residual method (GMRES), in connection with our proposed sparse preconditioners.

Our preliminary results show that the CGN with the proposed sparse preconditioners converges much faster than the GMRES. This conclusion is in agreement with our earlier work [Che96] on the one-dimensional case. The present work appears to be new as no extensive studies have been found in the literature regarding preconditioning bi-singular integral equations.

Our ultimate aim is to design efficient preconditioners for solving 2D singular boundary integral equations arising from 3D Helmholtz equations. This work has pointed out a way to achieve the aim.

To introduce our new work, we shall briefly describe how we decompose an integral operator in 1D based on domain splitting and further design sparse preconditioners.

Ninth International Conference on Domain Decomposition Methods
Editor Petter E. Bjørstad, Magne S. Espedal and David E. Keyes

2 Decomposition of 1D Integral Operators

As in [Che96], denote an operator equation defined over interval $[a, b]$ by

$$\mathcal{A}\mathbf{u} = \mathcal{F},$$

and the corresponding discretized matrix equation by

$$Au = f.$$

The purpose of preconditioning is to choose a matrix M such that the linear system $MAu = Mf$ is more amenable to the use of iterative methods. As M or its inverse M^{-1} must be sparse for efficiency, we choose M from part of matrix A.

It turns out that, for singular integral equations, the following operator $\mathcal{D}_1$ contains all the singularity of dense operator $\mathcal{A}$,

$$\mathcal{D}_1 = \begin{pmatrix} \times & \times & & & & \times \\ \times & \times & \times & & & \\ & \times & \times & \times & & \\ & & \times & \times & \ddots & \\ & & & \ddots & \ddots & \times \\ \times & & & & \times & \times \end{pmatrix}.$$

That is, we have the decomposition $\mathcal{A} = \mathcal{D}_1 + \mathcal{C}_1$. Correspondingly the matrix decomposition will be $A = D_1 + C_1$. Then we take $M = D_1^{-1}$ as a preconditioner, where D_1 is of the same sparsity pattern as $\mathcal{D}_1$.

Depending on the type of numerical methods one intends to use, the following two preconditioners, derived from collocation methods collocating at nodes and at mid-points, respectively, have been found to be effective

$$\mathcal{D}_2 = \begin{pmatrix} \times & & & & & \times \\ \times & \times & & & & \\ & \times & \times & & & \\ & & \times & \times & & \\ & & & \ddots & \ddots & \\ & & & & \times & \times \end{pmatrix}, \qquad \mathcal{D}_3 = \begin{pmatrix} \times & & & & & \\ & \times & & & & \\ & & \times & & & \\ & & & \times & & \\ & & & & \ddots & \\ & & & & & \times \end{pmatrix}.$$

The three types of preconditioners above will be considered in the following sections. The generalization of other preconditioners is currently under investigation; see [Che96], [Vav92] and [Yan94].

3 A 2D Model Equation

As a first step to developing preconditioners for solving 2D singular boundary integral equations arising from 3D Helmholtz equations, we consider the following model integral equation

$$\int_a^b \int_c^e k(x, y; \xi, \eta) u(\xi, \eta) d\xi d\eta = f(x, y), \tag{3.1}$$

with singular kernel $k(x,y;\xi,\eta)$, i.e., $\mathcal{A}\mathbf{u} = \mathbf{F}$. Note that this model equation can represent the bi-singular integral equation arising from aerodynamics modelling ([Ell95]), when

$$k(x,y;\xi,\eta) = \frac{d(x,y;\xi,\eta)}{\pi^2(\xi-x)(\eta-y)},$$

where d denotes a smooth function.

4 Decomposition of 2D Singular Operators

Our main idea in designing sparse preconditioners is based on operator splittings. Bearing in mind that whenever (x,y) and (ξ,η) are distinct there is no singularity in the kernel, we shall consider a general case and two special cases of operator splittings. We assume that when (x,y) and (ξ,η) are on the opposite side of the boundary, there is no singularity. However, for boundary integral equations arising from 3D Helmholtz equations, boundary points coincide so the first part of our splittings should have a wrap-around structure as in the 1D case.

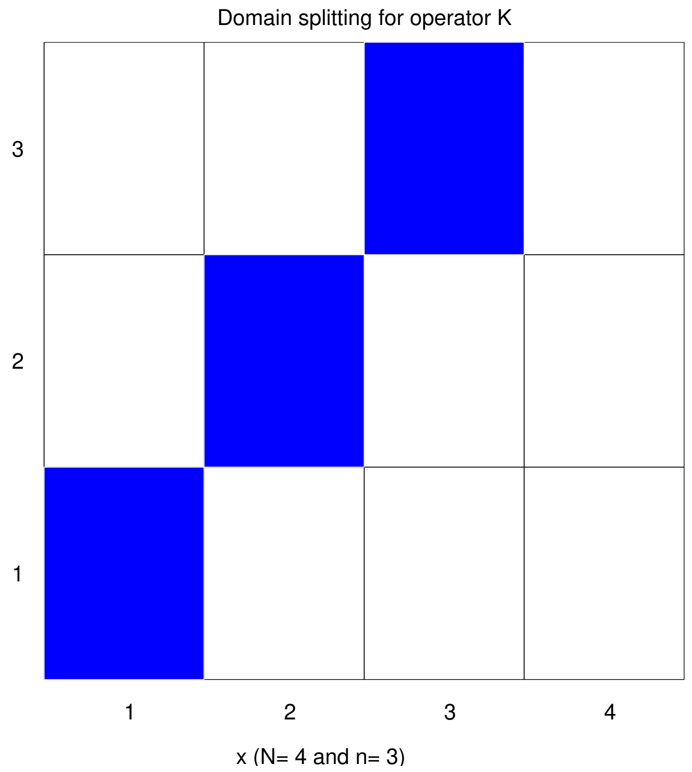
We now consider the decomposition of operator $\mathcal{A}$. To this end, partition the interval $[c,e]$ into n subintervals and $[a,b]$ into N subintervals. For simplicity, we shall take $n=3$ and $N=4$ in following discussions; see Fig. 1.

Then operator $\mathcal{A}$ can be written in a matrix form

$$\left[\begin{array}{cccc|cccc|cccc}
\mathcal{A}^{11}_{11} & \mathcal{A}^{11}_{12} & \mathcal{A}^{11}_{13} & \mathcal{A}^{11}_{1N} & \mathcal{A}^{11}_{21} & \mathcal{A}^{11}_{22} & \mathcal{A}^{11}_{23} & \mathcal{A}^{11}_{2N} & \mathcal{A}^{11}_{n1} & \mathcal{A}^{11}_{n2} & \mathcal{A}^{11}_{n3} & \mathcal{A}^{11}_{nN} \\
\mathcal{A}^{12}_{11} & \mathcal{A}^{12}_{12} & \mathcal{A}^{12}_{13} & \mathcal{A}^{12}_{1N} & \mathcal{A}^{12}_{21} & \mathcal{A}^{12}_{22} & \mathcal{A}^{12}_{23} & \mathcal{A}^{12}_{2N} & \mathcal{A}^{12}_{n1} & \mathcal{A}^{12}_{n2} & \mathcal{A}^{12}_{n3} & \mathcal{A}^{12}_{nN} \\
\mathcal{A}^{13}_{11} & \mathcal{A}^{13}_{12} & \mathcal{A}^{13}_{13} & \mathcal{A}^{13}_{1N} & \mathcal{A}^{13}_{21} & \mathcal{A}^{13}_{22} & \mathcal{A}^{13}_{23} & \mathcal{A}^{13}_{2N} & \mathcal{A}^{13}_{n1} & \mathcal{A}^{13}_{n2} & \mathcal{A}^{13}_{n3} & \mathcal{A}^{13}_{nN} \\
\mathcal{A}^{1N}_{11} & \mathcal{A}^{1N}_{12} & \mathcal{A}^{1N}_{13} & \mathcal{A}^{1N}_{1N} & \mathcal{A}^{1N}_{21} & \mathcal{A}^{1N}_{22} & \mathcal{A}^{1N}_{23} & \mathcal{A}^{1N}_{2N} & \mathcal{A}^{1N}_{n1} & \mathcal{A}^{1N}_{n2} & \mathcal{A}^{1N}_{n3} & \mathcal{A}^{1N}_{nN} \\
\hline
\mathcal{A}^{21}_{11} & \mathcal{A}^{21}_{12} & \mathcal{A}^{21}_{13} & \mathcal{A}^{21}_{1N} & \mathcal{A}^{21}_{21} & \mathcal{A}^{21}_{22} & \mathcal{A}^{21}_{23} & \mathcal{A}^{21}_{2N} & \mathcal{A}^{21}_{n1} & \mathcal{A}^{21}_{n2} & \mathcal{A}^{21}_{n3} & \mathcal{A}^{21}_{nN} \\
\mathcal{A}^{22}_{11} & \mathcal{A}^{22}_{12} & \mathcal{A}^{22}_{13} & \mathcal{A}^{22}_{1N} & \mathcal{A}^{22}_{21} & \mathcal{A}^{22}_{22} & \mathcal{A}^{22}_{23} & \mathcal{A}^{22}_{2N} & \mathcal{A}^{22}_{n1} & \mathcal{A}^{22}_{n2} & \mathcal{A}^{22}_{n3} & \mathcal{A}^{22}_{nN} \\
\mathcal{A}^{23}_{11} & \mathcal{A}^{23}_{12} & \mathcal{A}^{23}_{13} & \mathcal{A}^{23}_{1N} & \mathcal{A}^{23}_{21} & \mathcal{A}^{23}_{22} & \mathcal{A}^{23}_{23} & \mathcal{A}^{23}_{2N} & \mathcal{A}^{23}_{n1} & \mathcal{A}^{23}_{n2} & \mathcal{A}^{23}_{n3} & \mathcal{A}^{23}_{nN} \\
\mathcal{A}^{2N}_{11} & \mathcal{A}^{2N}_{12} & \mathcal{A}^{2N}_{13} & \mathcal{A}^{2N}_{1N} & \mathcal{A}^{2N}_{21} & \mathcal{A}^{2N}_{22} & \mathcal{A}^{2N}_{23} & \mathcal{A}^{2N}_{2N} & \mathcal{A}^{2N}_{n1} & \mathcal{A}^{2N}_{n2} & \mathcal{A}^{2N}_{n3} & \mathcal{A}^{2N}_{nN} \\
\hline
\mathcal{A}^{n1}_{11} & \mathcal{A}^{n1}_{12} & \mathcal{A}^{n1}_{13} & \mathcal{A}^{n1}_{1N} & \mathcal{A}^{n1}_{21} & \mathcal{A}^{n1}_{22} & \mathcal{A}^{n1}_{23} & \mathcal{A}^{n1}_{2N} & \mathcal{A}^{n1}_{n1} & \mathcal{A}^{n1}_{n2} & \mathcal{A}^{n1}_{n3} & \mathcal{A}^{n1}_{nN} \\
\mathcal{A}^{n2}_{11} & \mathcal{A}^{n2}_{12} & \mathcal{A}^{n2}_{13} & \mathcal{A}^{n2}_{1N} & \mathcal{A}^{n2}_{21} & \mathcal{A}^{n2}_{22} & \mathcal{A}^{n2}_{23} & \mathcal{A}^{n2}_{2N} & \mathcal{A}^{n2}_{n1} & \mathcal{A}^{n2}_{n2} & \mathcal{A}^{n2}_{n3} & \mathcal{A}^{n2}_{nN} \\
\mathcal{A}^{n3}_{11} & \mathcal{A}^{n3}_{12} & \mathcal{A}^{n3}_{13} & \mathcal{A}^{n3}_{1N} & \mathcal{A}^{n3}_{21} & \mathcal{A}^{n3}_{22} & \mathcal{A}^{n3}_{23} & \mathcal{A}^{n3}_{2N} & \mathcal{A}^{n3}_{n1} & \mathcal{A}^{n3}_{n2} & \mathcal{A}^{n3}_{n3} & \mathcal{A}^{n3}_{nN} \\
\mathcal{A}^{nN}_{11} & \mathcal{A}^{nN}_{12} & \mathcal{A}^{nN}_{13} & \mathcal{A}^{nN}_{1N} & \mathcal{A}^{nN}_{21} & \mathcal{A}^{nN}_{22} & \mathcal{A}^{nN}_{23} & \mathcal{A}^{nN}_{2N} & \mathcal{A}^{nN}_{n1} & \mathcal{A}^{nN}_{n2} & \mathcal{A}^{nN}_{n3} & \mathcal{A}^{nN}_{nN}
\end{array}\right].$$

In the general case, we obtain the splitting $\mathcal{A} = \mathcal{D}_1 + \mathcal{C}_1$, where all singularity of $\mathcal{A}$ is contained in a $\mathcal{D}_1$ that has a block tridiagonal structure, i.e., represented from the

Figure 1 The domain splitting for a general operator $\mathcal{A}$ ($n = 3$ & $N = 4$)

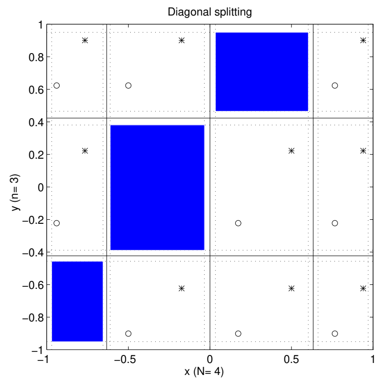


nonzero entries below (boxed and un-boxed)

$$
\left[\begin{array}{cccc|cccc|cccc}
\boxed{\times} & \times & & & \times & \times & & & & & & \\
\boxed{\times} & \boxed{\times} & \times & & \times & \times & \times & & & & & \\
& \boxed{\times} & \boxed{\times} & \times & & \times & \times & \times & & & & \\
& & \boxed{\times} & \boxed{\times} & & & \times & \times & & & & \\
\hline
\boxed{\times} & \times & & & \boxed{\times} & \times & & & \times & \times & & \\
\boxed{\times} & \boxed{\times} & \times & & \boxed{\times} & \boxed{\times} & \times & & \times & \times & \times & \\
& \boxed{\times} & \boxed{\times} & \times & & \boxed{\times} & \boxed{\times} & \times & & \times & \times & \times \\
& & \boxed{\times} & \boxed{\times} & & & \boxed{\times} & \boxed{\times} & & & \times & \times \\
\hline
& & & & \boxed{\times} & \times & & & \boxed{\times} & \times & & \\
& & & & \boxed{\times} & \boxed{\times} & \times & & \boxed{\times} & \boxed{\times} & \times & \\
& & & & & \boxed{\times} & \boxed{\times} & \times & & \boxed{\times} & \boxed{\times} & \times \\
& & & & & & \boxed{\times} & \boxed{\times} & & & \boxed{\times} & \boxed{\times}
\end{array}\right]. \tag{4.2}
$$

To simplify $\mathcal{D}_1$ further, we first consider only part of the underlying domain that surrounds all nodes; see Fig. 2. This corresponds to numerical methods collocating at nodes. We obtain the splitting $\mathcal{A} = \mathcal{D}_2 + \mathcal{C}_2$, where all singularities of $\mathcal{A}$ are contained

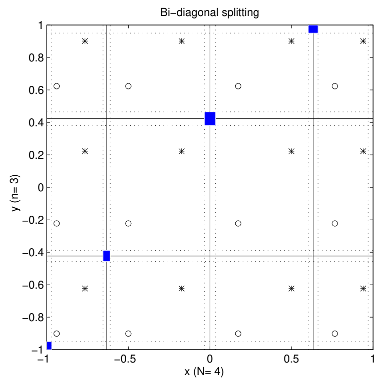
Figure 2 The domain splitting to generate a bi-diagonal splitting of $\mathcal{A}$



in $\mathcal{D}_2$, which has a block bi-diagonal structure, e.g., represented by the boxed entries in (4.2).

We then consider the opposite part of the domain that excludes all nodes; see Fig. 3. This is associated with numerical methods collocating at interior points. Now

Figure 3 The domain splitting to generate a diagonal splitting of $\mathcal{A}$



the operator is split as $\mathcal{A} = \mathcal{D}_3 + \mathcal{C}_3$, where all singularities of $\mathcal{A}$ are contained in $\mathcal{D}_3$, which has a diagonal structure, e.g., the diagonal in (4.2).

In summary, all three operator splitting strategies discussed are such that the preconditioned equation has a compact operator and therefore ideal spectral properties. At the matrix level, we propose three sparse block preconditioners: tri-diagonal ($M = M_1$), bi-diagonal ($M = M_2$), and diagonal ($M = M_3$) matrices.

Table 1 Convergence results of CGN

n	Nodes n^2	Unpreconditioned	Tri-diagonal M_1	Bi-diagonal M_2	Diagonal M_3
4	16	12	13	12	10
8	64	88	33	51	23
16	256	*	88	215	41
32	1024	*	216	972	68

5 Numerical Results

We now present some numerical results of using the above described sparse preconditioners for solving

$$\frac{1}{\pi}\int_{-1}^{1}\frac{1}{\pi}\int_{-1}^{1}\frac{d(x,y;\xi,\eta)}{(\xi-x)(\eta-y)}u(\xi,\eta)d\xi d\eta = f(x,y),$$

with Kutta conditions $\left\{\begin{array}{ll} u(1,y)=0 & -1<y<1, \\ u(x,1)=0 & -1<x<1. \end{array}\right.$

Write the index 0 solution as

$$u(x,y)=\sqrt{\frac{1-x}{1+x}}\sqrt{\frac{1-y}{1+y}}\Psi(x,y)$$

to give the equation

$$\frac{1}{\pi}\int_{-1}^{1}\sqrt{\frac{1-\xi}{1+\xi}}\frac{d\xi}{\xi-x}\left(\frac{1}{\pi}\int_{-1}^{1}\sqrt{\frac{1-\eta}{1+\eta}}\frac{\Psi(\xi,\eta)d\eta}{\eta-y}\right)=f(x,y).$$

Taking $N=n$, an appropriate discretization is given by

$$\sum_{j=1}^{n}\sum_{k=1}^{n}\frac{d(\eta_{rn},\eta_{sn};\xi_{jn},\xi_{kn})}{(\xi_{jn}-\eta_{rn})(\xi_{kn}-\eta_{sn})}\frac{(1-\xi_{jn})(1-\xi_{kn})}{(n+1/2)(n+1/2)}\Psi_{nn}(\xi_{jn},\xi_{kn})=f(\eta_{rn},\eta_{sn})$$

where

$$\left\{\begin{array}{ll} \xi_{jn}=\cos\left(\frac{2j\pi}{2n+1}\right), & j=1,\cdots,n \\ \eta_{jn}=\cos\left(\frac{(2j-1)\pi}{2n+1}\right), & j=1,\cdots,n. \end{array}\right.$$

We consider $\Psi(x,y)=1$, $d(x,y;\xi,\eta)=1$, and $f(x,y)=1$. In Tables 1 and 2, we show the number of steps required to reduce the residual error to below a tolerance of $10^{(-2-\log(n)/\log(2))}=10^{-(\ell+2)}$ for $n=2^\ell$, where '*' denotes no convergence or the iteration steps exceed $3n/2$, CGN stands for the conjugate gradient normal method and GMRES(m) for the generalized minimal residual method; see [NRT92]. This particular

Table 2 Convergence results of re-started GMRES(3)

n	Nodes n^2	Unpreconditioned	Tri-diagonal M_1	Bi-diagonal M_2	Diagonal M_3
4	16	2	6	12	*
8	64	2	22	26	*
16	256	2	*	53	*
32	1024	*	*	198	*

choice of tolerance ensures that the residual is of a comparable magnitude to the truncation error that would result with a direct solver.

We can observe that CGN produces results as predicted and M_3 is the best preconditioner because the numerical method collocates at interior points (see Fig. 3). However the performance of GMRES is somewhat erratic; we have experimented with an increased m and observed similar results. Here each step of GMRES involves m matrix-vector multiplications while CGN involves two matrix-vector multiplications. Therefore our preliminary results suggest that CGN is suitable for bi-singular integral equations with operator splitting based sparse preconditioners (M_3 and M_1). For integral equations, in general, CGN with suitable preconditioners may out perform other iterative solvers; see [Che97] for some discussion.

Acknowledgement

The author wishes to thank Professor David Elliott for suggesting the test problem and its numerical solution.

REFERENCES

[Che94] Chen K. (1994) Efficient iterative solution of linear systems from discretizing singular integral equation. *Elec. Tran. Numer. Anal.* 2: 76–91.

[Che96] Chen K. (1996) Solution of singular boundary element equations based on domain splitting. In Glowinski R., Périaux J., Shi Z.-C., and Widlund O. B. (eds) *Proc. Eighth Int. Conf. on Domain Decomposition Meths.* Wiley and Sons, Chichester.

[Che97] Chen K. (1997) On preconditioning techniques for dense linear systems from bundary elements. *to appear* .

[Ell95] Elliott D. (1995) Private communication. Maths Department, University of Tasmania, Australia.

[NRT92] Nachtigal N., Reddy S. C., and Trefethen L. N. (1992) How fast are nonsymmetric matrix iterations. *SIAM J. Matrix Anal. Appl.* 13(3): 778–795.

[Vav92] Vavasis S. (1992) Preconditioning for boundary integral equations. *SIAM J. Matrix. Anal. Appl.* 13(3): 905–925.

[Yan94] Yan Y. (1994) Sparse preconditioned iterative methods for dense linear systems. *SIAM J. Sci. Comp.* 15(5): 1190–1200.