Block Preconditioners for Linear Systems Arising from Multiscale Collocation with Compactly Supported RBFs

Farrell, Patricio and Pestana, Jennifer

2014

MIMS EPrint: **2014.18**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: http://eprints.maths.manchester.ac.uk/
And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

Block Preconditioners for Linear Systems Arising from Multiscale Collocation with Compactly Supported RBFs

Patricio Farrell and Jennifer Pestana

February 13, 2015

Abstract

Symmetric collocation methods with radial basis functions allow approximation of the solution of a partial differential equation, even if the right-hand side is only known at scattered data points, without needing to generate a grid. However, the benefit of a guaranteed symmetric positive definite block system comes at a high computational cost. This cost can be alleviated somewhat by considering compactly supported radial basis functions and a multiscale technique. But the condition number and sparsity will still deteriorate with the number of data points. Therefore, we study certain block diagonal and triangular preconditioners. We investigate ideal preconditioners and determine the spectra of the preconditioned matrices before proposing more practical preconditioners based on a restricted additive Schwarz method with coarse grid correction (ARASM). Numerical results verify the effectiveness of the preconditioners.

1 Introduction

Radial basis functions (RBFs) are a modern tool to flexibly approximate scattered data [24, 38, 3, 9, 32]. Although they were initially used for interpolation problems, they have recently become an attractive alternative to solve PDEs – especially when the given data is arbitrarily scattered [14, 18, 12, 13, 39, 35, 17]. Their attractiveness stems from the fact that the RBF method dispenses with the expensive generation of a grid. Unlike a method yielding a nonsymmetric system proposed by Kansa [25], we would like to focus on *symmetric* RBF approximation, which has the advantage that it always yields a symmetric positive definite system.

The main problem with RBFs is that for a large number of data sites the condition number of the system one needs to solve becomes prohibitively large. Therefore, multiscale ideas in combination with compactly supported RBFs have been developed to reduce the computational cost [8, 11, 28, 40]. Here, we focus on a symmetric multiscale RBF collocation method for second-order elliptic

PDEs on bounded domains [7, 8]. The method employs RBFs on a sequence of levels, and uses different numbers of data sites and RBF support radii on each level. Consequently, it is particularly suited to problems with multiple scales.

At each level of the multiscale collocation method a linear system of the form

$$\underbrace{\begin{bmatrix} \bar{A} & \bar{B}^T \\ \bar{B} & \bar{C} \end{bmatrix}}_{\bar{\mathcal{A}}} \underbrace{\begin{bmatrix} \alpha^i \\ \alpha^b \end{bmatrix}}_{\alpha} = \underbrace{\begin{bmatrix} b^i \\ b^b \end{bmatrix}}_{b}, \tag{1}$$

must be solved, where $\bar{\mathcal{A}}$ is nonsingular and symmetric positive definite and $\bar{B} \in \mathbb{R}^{m \times n}$ has full rank. The positive definiteness of $\bar{\mathcal{A}}$ ensures that the principal submatrices $\bar{A} \in \mathbb{R}^{n \times n}$ and $\bar{C} \in \mathbb{R}^{m \times m}$ are themselves positive definite. More detailed descriptions of these matrices will be provided in Section 2.2. We make the reasonable assumption that $n \geq m$, which means that more data sites are located in the interior of the domain than on its boundary.

Even though the multiscale approach is already much more efficient than the one-shot method, certain features of $\bar{\mathcal{A}}$ make (1) difficult to solve at later levels. First, the number of data sites, and thus the dimension of the matrix $\bar{\mathcal{A}}$, increases at each level so that finer scales may be resolved. Second, the conditioning of the coefficient matrix deteriorates as the separation between data sites decreases, a fact we state more in carefully in Theorem 2. In particular, if one wants to ensure convergence, the density of nonzero elements increases [7, 8].

When the dimension of $\bar{\mathcal{A}}$ is large, iterative methods are more feasible than direct methods for solving (1). However, the ill-conditioning and density of nonzeros mean that fast convergence will typically only be achieved with suitable preconditioners. For interpolation problems solved by an analogous RBF multiscale method, convergence and constant condition numbers can be achieved at the same time [40]. However, applying the same approach to PDE problems is insufficient for level-independent convergence [7, 8] and a more sophisticated strategy is required.

Preconditioners for RBF matrices have previously been developed, with domain decomposition approaches among the most popular. Beatson *et al.* [2] employed a multiplicative Schwarz method to solve (rather than precondition) linear systems resulting from interpolation by polyharmonic splines. Yokota *et al.* [41] investigated restricted additive Schwarz (RAS) methods for interpolation by Gaussian RBFs, while Deng and Driscoll [6] used a two-level RAS method as a GMRES preconditioner for interpolation by multiquadrics. Additionally, Le Gia *et al.* [27] applied a two-level overlapping additive Schwarz method to the problem of solving PDEs by compactly-supported basis functions on spheres. Alternative preconditioners include those based on approximate cardinal functions [1, 31], which can be combined with domain decomposition [2, 30].

To construct preconditioners for the whole matrix $\bar{\mathcal{A}}$, however, it seems sensible to exploit the block structure. Recently, Le Gia and Tran [29] examined effective block diagonal preconditioners for the RBF multiscale method for PDEs on local spherical regions, with additive Schwarz preconditioners for each block. In this complementary work we consider both block diagonal and

block triangular preconditioners for the multiscale method for PDEs on general bounded domains. In contrast to Le Gia and Tran [29] we attempt to identify ideal preconditioners, from the point of view of fast convergence of the iterative solver of (1), and analytically determine the spectra of the preconditioned matrices. These ideal preconditioners then guide the development of more practical alternatives based on restricted additive Schwarz methods.

The rest of this paper is organised as follows. In Section 2 we describe in more detail the multiscale RBF method and give a brief overview of Krylov subspace methods and preconditioners for solving (1). We present our ideal block diagonal and block triangular preconditioners, and describe the spectra of the preconditioned matrices, in Section 3. We investigate the effect of replacing $\bar{A}$ by an additive Schwarz preconditioner in Section 4 and give numerical results in Section 5. Note that throughout, the transpose of a matrix A is represented by A^T and its nullspace by null(A).

2 Background

In this section we present background material on the multiscale RBF method and on preconditioned Krylov subspace methods.

2.1 Second-order elliptic boundary value problems

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain. We consider second-order elliptic boundary value problems of the form

$$\begin{aligned} \mathcal{L}u &= f \quad \text{in } \Omega, \\ u &= F \quad \text{on } \partial\Omega, \end{aligned} \tag{2}$$

where $\mathcal{L}$ is a second-order elliptic linear differential operator defined by

$$\mathcal{L}u(x) = \sum_{i,j=1}^{d} a_{ij}(x)\partial_{ij}u(x) + \sum_{i=1}^{d} b_i(x)\partial_i u(x) + c(x)u(x),$$

which is strictly elliptic on Ω. That is, there exists a constant $c_E > 0$ such that

$$c_E\|\xi\|_2^2 \leq \sum_{i,j=1}^{d} a_{ij}(x)\xi_i\xi_j$$

for all $x \in \Omega$ and $\xi = (\xi_i) \in \mathbb{R}^d$.

If we assume that the right-hand sides f and F are chosen such that the solution u lies in the Sobolev space $H^\sigma(\Omega)$ with $\sigma > d/2+2$, then the differential operator $\mathcal{L}u$ is in fact well-defined since we know by the Sobolev embedding theorem that $H^\sigma(\Omega) \subseteq C^2(\Omega)$.

To ensure that $\mathcal{L}$ is a bounded operator from $H^\sigma(\Omega)$ to $H^{\sigma-2}(\Omega)$, we impose some restrictions on the coefficients. For $k := \lfloor\sigma\rfloor > 2 + d/2$ we demand that

a_{ij}, b_i, c lie in $W_\infty^{k-1}(\Omega)$. Due to our previous assumption on σ, we see that $(k-1)-1 > d/2$, which implies by the Sobolev embedding theorem that the coefficients are continuously differentiable, see [18] for details.

2.2 Multiscale RBF collocation

In order to solve the boundary value problem (2), we will construct a numerical approximation from a linear combination of translated radial basis functions. These basis functions are particularly useful in the context of scattered data approximation. Therefore, we introduce two measures that help us to describe scattered data points $X = \{x_1, \ldots, x_N\}$ in $\Omega \subset \mathbb{R}^d$. The mesh norm

$$h_{X,\Omega} = \sup_{x \in \Omega} \min_{x_j \in X} \|x - x_j\|_2$$

is the radius of the largest data-free hole that is contained in the domain of interest Ω. On the other hand, the separation distance

$$q_X = \min_{j \neq k} \|x_j - x_k\|_2$$

is the shortest distance between any two data points in X.

Definition 1 (Radial basis function)**.** *A continuous function $\Phi\colon \mathbb{R}^d \to \mathbb{R}$ is called positive definite on $\mathbb{R}^d$ if for any d-dimensional data set $X = \{x_1, \ldots, x_N\}$ of pairwise distinct points the matrix*

$$A_{\Phi,X} := (\Phi(x_j - x_k))_{1 \leq j,k \leq N}$$

is positive definite. We refer to Φ as a radial basis function if it is a radial positive definite function.

There are many different examples of radial basis functions. Among them Gaussians, (inverse) multiquadrics and polyharmonic splines have been popular. Here, we are interested in compactly supported radial basis functions since in this case the matrix $A_{\Phi,X}$ is sparse. The Fourier transform of the compactly supported radial basis function shall satisfy

$$c_1(1 + \|\omega\|_2^2)^{-\sigma} \leq \widehat{\Phi}(\omega) \leq c_2(1 + \|\omega\|_2^2)^{-\sigma} \tag{3}$$

for $0 < c_1 \leq c_2$. For $\sigma > \frac{d+1}{2}$ it is indeed possible to find such a function, see [5].

The most prominent examples of such compactly supported RBFs were given by Wendland [37]. For any given integer smoothness degree and dimension Wendland was able to construct radial basis functions that are polynomials within the unit ball and vanish outside of it. They have the special property that for given smoothness and dimension these polynomials are of the smallest degree such that the compactly supported RBF is still positive definite.

The numerical solution we would like to consider is of the form

$$s(x) = \sum_{j=1}^{n} \alpha_j^i \mathcal{L}^{(2)} \Phi(x - x_j) + \sum_{j=1}^{m} \alpha_j^b \Phi(x - y_j), \tag{4}$$

where $X = \{x_1, \ldots, x_n\} \subset \Omega$ and $Y = \{y_1, \ldots, y_m\} \subset \partial\Omega$ denote scattered data points in the interior and on the boundary, respectively, for which the right-hand sides are known. The superscript next to the differential operator in the first sum means that the operator is first applied to the second argument of the RBF and then evaluated at data site x_j, that is, we apply $\mathcal{L}$ to the second term z in $\Phi(x-z)$ and then evaluate the result at $z = x_j$. Note that $\mathcal{L}^{(2)}\Phi(x-x_j)$ is still a function of the first variable x. The real interior and boundary coefficients α_j^i and α_j^b are determined by applying the boundary value problem (2) to the numerical approximation s at these scattered data points. The system one needs to solve is then given by (1), where the different parts are given by

$$\begin{aligned}
&\bar{A} = (\mathcal{L}^{(1)}\mathcal{L}^{(2)}\Phi(x_i - x_j))_{1\le i,j\le n}, \\
&\bar{B} = (\mathcal{L}^{(2)}\Phi(y_i - x_j))_{\substack{1\le i\le m,\\ 1\le j\le n}}, && \bar{C} = (\Phi(y_i - y_j))_{1\le i,j\le m}, \\
&\alpha^i = (\alpha_j^i)_{1\le j\le n}, && \alpha^b = (\alpha_j^b)_{1\le j\le m}, \\
&b^i = (f(x_j))_{1\le j\le n}, && b^b = (F(y_j))_{1\le j\le m}.
\end{aligned}$$

In the context of generalised interpolation [38], it can be shown that the matrices $\bar{\mathcal{A}}$ and $\bar{A}$ are symmetric and positive definite. The matrix $\bar{C}$, on the other hand, is symmetric and positive definite by Definition 1. This guarantee that the block matrix $\bar{\mathcal{A}}$ is always symmetric and positive definite is the main reason we choose our numerical approximation as in (4). However, the above approach also differs fundamentally from other methods commonly used to approximate PDEs such as finite difference, finite element and finite volume methods. Unlike for these methods, the matrix system (1) is not a discrete approximation of the PDE. Hence, classical preconditioning theory for PDE problems does not directly apply.

Though it would be feasible to use just this one-shot solution as a numerical approximation to the solution of the PDE, it is not very efficient to do so since the system suffers from severe ill-conditioning. One way around this is to employ the following multiscale strategy.

We will choose a sequence of denser data sets as well as smaller support radii. We denote a sequence of point sets in the bounded domain Ω by $X_1, X_2, X_3, ...$ and a sequence of point sets on the domain's boundary $\partial\Omega$ by $Y_1, Y_2, Y_3, ...$ For support radii $\delta_j > 0$ and a compactly supported basis function Φ we define

$$\Phi_j(x - y) = \Phi_{\delta_j}(x - y) = \Phi\left(\frac{x - y}{\delta_j}\right).$$

So if Φ has unit support, the Φ_j indeed have support radii δ_j. Instead of using just the previously introduced one-shot approximation, we will construct several

ones, coming from the spaces

$$\mathcal{L}V_{X_j} + V_{Y_j} = \operatorname{span}\{\mathcal{L}\Phi_j(\cdot - x) \mid x \in X_j\} + \operatorname{span}\{\Phi_j(\cdot - y) \mid y \in Y_j\},$$

where the notation $\Phi_j(\cdot - x)$ again indicates that after fixing x, we still have a function of one variable. At each level a system of the form (1) must be solved, with the entries of $\bar{\mathcal{A}}$ depending on the scaled basis function $\Phi_j = \Phi_{\delta_j}$ which implies that the collocation matrix also depends on the support radius ($\bar{\mathcal{A}} = \bar{\mathcal{A}}(\delta)$). Each approximation from these spaces shall resolve the current residual so that the sum of all those approximations yields a numerical solution to the original PDE. This is achieved quite naturally by the following multiscale RBF collocation algorithm.

Algorithm 1 (Multiscale RBF collocation algorithm)**.** *Given right-hand sides f and F do:*

1. *Set $u_0 = 0, f_0 = f, F_0 = F$*
2. *For $j = 1, 2, 3 \ldots$ do*
   (a) *Determine the correction $s_j \in \mathcal{L}V_{X_j} + V_{Y_j}$ to residuals f_{j-1} and F_{j-1} from the equations*

$$\begin{aligned} \mathcal{L}s_j(x) &= f_{j-1}(x), \qquad x \in X_j, \\ s_j(y) &= F_{j-1}(y), \qquad y \in Y_j. \end{aligned}$$

   (b) *Update the final approximation and the residuals*

$$\begin{aligned} u_j &= u_{j-1} + s_j, \\ f_j &= f_{j-1} - \mathcal{L}(s_j|\Omega), \\ F_j &= F_{j-1} - s_j|\partial\Omega. \end{aligned}$$

A variant of this algorithm for pure interpolation problems was shown to converge [40] if the support radii are chosen proportionally to the mesh norms. This in turn implied that for quasi-uniform data sets (i.e. when the separation distance is comparable to the mesh norm) the condition numbers of the interpolation matrices could be bounded independently of the current level. Thus, the conjugate gradient method would converge in a fixed number of steps, regardless of the size of the problem.

Unfortunately, the same does not hold for the multiscale collocation algorithm. We state the two main results from [7] concerning convergence and stability, omitting some technical details in favour of readability. The main point here is that in order to guarantee convergence, one has to cope with ill-conditioning issues.

Theorem 1 (Convergence)**.** *Let Φ satisfy (3) for $\sigma > 2 + d/2$, and h denote the maximum of the boundary and interior mesh norms. Then, the multiscale collocation algorithm for elliptic boundary value problems converges as h tends to zero if the support radius δ is chosen proportional to $h^{1-2/\sigma}$.*

Theorem 2 (Stability)**.** *Under the same assumptions of Theorem 1, the condition number of the block matrices $\bar{\mathcal{A}} = \bar{\mathcal{A}}(\delta)$ arising in Algorithm 1 can be bounded by*

$$\operatorname{cond}(\bar{\mathcal{A}}) \leq C q_{X\cup Y}^{-8+8/\sigma}.$$

Employing the diagonal preconditioner $\mathcal{M}$ with entries

$$m_{ij} = \begin{cases} 0, & i \neq j, \\ \delta^2, & 1 \leq i \leq n, \\ 1, & n \leq i \leq n+m, \end{cases}$$

the condition number can be bounded by

$$\operatorname{cond}(\mathcal{M}^{-1}\bar{\mathcal{A}}\mathcal{M}^{-1}) \leq C q_{X\cup Y}^{-4}.$$

When implementing the algorithm we need to solve a block system for the coefficients and the function updates will become vector updates.

It might not be intuitively clear why the condition number of the scaled matrix is always bounded essentially by q^{-4}. One might expect that the condition numbers vary with the smoothness of the underlying RBF. Looking more closely at the proof of Theorem 5.3 in [7], we note that the bound actually depends on the smoothness – but in a beneficial way. Let $c > 0$ be independent of the smoothness parameter σ. We have

$$\operatorname{cond}(\mathcal{M}^{-1}\bar{\mathcal{A}}\mathcal{M}^{-1}) \leq c(q^{2/\sigma}+1) q_{X\cup Y}^{-4} = c C_{\sigma,q} q_{X\cup Y}^{-4}.$$

with $C_{\sigma,q} = q_{X\cup Y}^{2/\sigma} + 1$. However since $\sigma > 0$, we can always find a smoothness independent bound on this constant as $q_{X\cup Y}$ tends to zero.

The preconditioner $\mathcal{M}$ aims to mitigate the different scaling of the blocks of $\bar{\mathcal{A}}$ in (1), since due to the chain rule the block $\bar{A}$ scales like $O(\delta^{-4})$, $\bar{B}$ like $O(\delta^{-2})$ and $\bar{C}$ like $O(1)$. The preconditioned system is

$$\underbrace{\mathcal{M}^{-1}\bar{\mathcal{A}}\mathcal{M}^{-1}}_{\mathcal{A}} y = \mathcal{M}^{-1}b, \quad \mathcal{M}^{-1}y = x, \quad \mathcal{A} = \begin{bmatrix} A & B^T \\ B & C \end{bmatrix}, \tag{5}$$

Lastly, we point out that employing the Jacobi preconditioner $\sqrt{\operatorname{diag}(\bar{\mathcal{A}})}$, first suggested by Fasshauer [8], yields even better condition numbers than $\mathcal{M}$ but the result in Theorem 2 still applies as the diagonals only differ by constants. Hence, the asymptotic growth of the condition number is the same for both variants.

Since $\mathcal{M}$ is diagonal, $\mathcal{A}$ can be computed cheaply from the sparse matrix $\bar{\mathcal{A}}$. This requires $O(k*(n+m))$ floating point operations, where $k = k(n+m)$ is the number of nonzeros per row of $\bar{\mathcal{A}}$. Because for a convergent scheme the support radii decrease more slowly than the mesh norm (see Theorem 1), k actually increases mildly for larger linear systems.

However, within the iterative methods we discuss in the next section we only require matrix-vector products with $\mathcal{A}$, i.e., products of the form $\mathcal{A}v$ for some

vector $v \in \mathbb{R}^{n+m}$. These can be computed by first forming $u = \bar{\mathcal{A}}(\mathcal{M}^{-1}v)$, and then computing $\mathcal{M}^{-1}u$. Accordingly, these matrix vector products require only $2(n+m)$ more floating point operations than are needed to compute a matrix-vector product with $\bar{\mathcal{A}}$.

Due to the obvious improvement in the condition number, we assume that the Jacobi preconditioner $\mathcal{M} = \sqrt{\text{diag}(\bar{\mathcal{A}})}$ has been applied to (1). We then solve (5) with this particular preconditioner in the following sections.

2.3 Preconditioned Krylov subspace methods

When the system (5) is large and sparse, iterative methods are often used to obtain its solution, with Krylov subspace methods among the most popular. If the coefficient matrix is symmetric positive definite, as in (5), we can apply the conjugate gradient method (CG) [21]. For nonsymmetric systems, iterative methods such as GMRES [34], QMR [15] or Bi-CGSTAB [36] are required.

Whichever Krylov method is employed, the rate of convergence can be sensitive to the conditioning of the matrix $\mathcal{A}$. In particular, the convergence rate of Krylov methods for symmetric positive definite matrices often decreases when the condition number of $\mathcal{A}$ increases, and this certainly occurs for the linear systems arising from the RBF collocation method. Moreover, for any of the mentioned Krylov methods small eigenvalues can cause slow convergence. The condition numbers of the RBF multiscale matrices $\mathcal{A}$ increase at each level and thus to achieve fast convergence of the Krylov subspace method it is necessary to precondition (5). In right preconditioning $\mathcal{A}\mathcal{P}^{-1}y = b$, $\alpha = \mathcal{P}^{-1}y$, is solved in place of (5). Symmetry can be preserved when the preconditioner is symmetric positive definite, (see, for example, Greenbaum [19, Chapter 8]) and CG can be applied to the preconditioned system. Within the preconditioned conjugate gradient method, we only require the action of $\mathcal{P}$ on a vector, that is, given some vector v we only need to solve the system $\mathcal{P}u = v$.

2.4 Test problem

The next sections are devoted to overcoming the ill-conditioning described by Theorem 2. To illustrate the improvements made, as well as certain features of $\mathcal{A}$, we test our preconditioners using the following example problem.

We have implemented a Poisson problem on the unit square $\Omega = (0,1)^2$ with boundary $\partial\Omega$, which comes from [9], namely

$$\Delta u = -\frac{5}{4}\sin(\pi x)\cos(\pi y/2) \qquad \text{in} \quad \Omega,$$
$$u = \begin{cases} \sin(\pi x) & \text{on} \quad 0 \le x \le 1, y = 0, \\ 0 & \text{elsewhere on} \quad \partial\Omega. \end{cases}$$

We use Wendland's compactly supported radial basis function, $\phi_{2,3}(r) = (1-r)_+^8(32r^3 + 25r^2 + 8r + 1) \in C^6(\mathbb{R}^2)$, which satisfies (3) with $\sigma = 4.5$ as well as

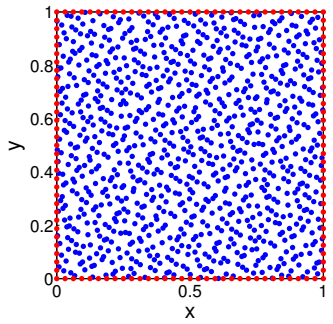


Figure 1: Blue interior Halton points and red uniform boundary points, 961 and 128 points respectively.

support radii of the form

$$\delta = \nu \left(h/\mu\right)^{1-2/4.5}$$

with $\mu = 0.5$ and $\nu = 2.4$. Again, h denotes the maximum of the boundary and interior mesh norms. By Theorem 1, we have ensured convergence of the multiscale collocation algorithm. Theorem 2 states that the condition number of the collocation matrix in (1) should behave like $q_{X\cup Y}^{-6.\overline{2}}$ and the condition number of the preconditioned collocation matrix in (5) should behave like $q_{X\cup Y}^{-4}$. These theoretical results were verified in [7].

In the following we discuss uniform nested data sets as well as nested data sets created from Halton points. Halton points appear to be randomly distributed but are actually deterministically determined via van der Corput sequences. For details we refer to the literature [20, 9]. See Figure 1 for a distribution of Halton points in two space dimensions.

3 Ideal block preconditioners

To achieve better conditioning as the levels increase we need effective preconditioners. In this section we introduce ideal block diagonal and block triangular preconditioners for the linear system (5) that depend on the blocks of $\mathcal{A}$ and on the Schur complement $S = C - BA^{-1}B^T$. (Recall from the end of Section 2.2 that we solve the diagonally preconditioned system (5) because $\mathcal{A}$ is better conditioned than $\bar{\mathcal{A}}$ in (1).) We additionally examine the effect of the preconditioning blocks on the spectrum and conditioning of the preconditioned matrix. Although the preconditioners we obtain may be too expensive to apply in practice, since they all involve linear solves with A, in subsequent sections we show that this solve with A can be replaced by a restricted additive Schwarz method to obtain an efficient alternative.

Our block diagonal preconditioners are positive definite and the preconditioned system can be solved by the conjugate gradient method. On the other

hand, block triangular preconditioners are nonsymmetric and can only be used in conjunction with a Krylov method for nonsymmetric matrices. However, if the speed of convergence is significantly faster with the block triangular preconditioner this may outweigh any extra cost associated with using a nonsymmetric solver; this is the case here (see Table 11).

Note that since $\mathcal{P}^{-1}\mathcal{A}$, $\mathcal{P}^{-\frac{1}{2}}\mathcal{A}\mathcal{P}^{-\frac{1}{2}}$ and $\mathcal{A}\mathcal{P}^{-1}$ are similar they all have the same eigenvalues, although their eigenvectors may differ. Thus, in the following we determine only the eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$.

3.1 Block diagonal preconditioners

For saddle point matrices, for which $C = 0$ in (5), it is well known [23, 26, 33] that if the preconditioner

$$\mathcal{P}_0 = \begin{bmatrix} A & 0 \\ 0 & BA^{-1}B^T \end{bmatrix},$$

is applied then $\mathcal{P}_0^{-1}\mathcal{A}$ has the three eigenvalues 1, $(1 \pm \sqrt{5})/2$. Krylov methods applied to this preconditioned system usually converge rapidly; for example, CG converges in at most three iterations. When C is small in norm, we might expect the preconditioner $\mathcal{P}_0$ to continue to perform well but when the norm of C increases a different choice for the (2,2) block might be preferable. Accordingly, we consider the block diagonal preconditioner

$$\mathcal{P}_D = \begin{bmatrix} A & 0 \\ 0 & \widehat{S} \end{bmatrix}, \tag{6}$$

where $\widehat{S} \in \mathbb{R}^{m\times m}$ is symmetric positive definite, and investigate the spectrum of $\mathcal{P}_D^{-1}\mathcal{A}$ for different choices of $\widehat{S}$. The following lemma describes the spectrum of $\mathcal{P}_D^{-1}\mathcal{A}$.

Lemma 3. *The matrix $\mathcal{P}_D^{-1}\mathcal{A}$, with $\mathcal{P}_D$ and $\mathcal{A}$ defined by* (6) *and* (5)*, respectively, has the eigenvalue 1 with multiplicity $n - m$. Each of the remaining $2m$ eigenvalues λ, with corresponding eigenvector $[u^T v^T]^T$, satisfies*

$$\lambda = \frac{1}{2}\left(1 + \frac{v^T C v}{v^T \widehat{S} v}\right) \pm \sqrt{\frac{1}{4}\left(1 - \frac{v^T C v}{v^T \widehat{S} v}\right)^2 + \frac{v^T B A^{-1} B^T v}{v^T \widehat{S} v}}. \tag{7}$$

Proof. The eigenvalues λ of $\mathcal{P}_D^{-1}\mathcal{A}$ satisfy

$$\begin{aligned} Au + B^T v &= \lambda A u, \\ Bu + Cv &= \lambda \widehat{S} v, \end{aligned} \tag{8}$$

where $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^m$ are not simultaneously 0. Both $\mathcal{A}$ and $\mathcal{P}_D$ are nonsingular and so $\lambda \neq 0$. If $\lambda = 1$, then $B^T v = 0$ which, since B has full rank and $m \leq n$, implies that $v = 0$. Then (8) shows that $u \in$ null(B). We can find

$n-m$ linearly independent vectors $u \in \text{null(B)}$ and so $\lambda = 1$ is an eigenvalue of $\mathcal{P}_D^{-1}\mathcal{A}$ with multiplicity $n-m$.

If $\lambda \neq 1$, then $u = 1/(\lambda-1)A^{-1}B^T v$, from which we see that $v \neq 0$. Substituting for u in (8), pre-multiplying by v^T, dividing by $v^T\widehat{S}v$ and simplifying gives (7). □

The $2m$ non-unit eigenvalues (7) lie in two intervals on the real line that should be small if the rate of convergence of the Krylov subspace method is to be fast. Note that these $2m$ eigenvalues result from the imposition of a boundary condition (c.f. the analogous matrix for PDEs on the sphere [28].)

Since (7) contains the terms v^TCv and $v^TBA^{-1}B^Tv$, we consider the choices C, S, and $BA^{-1}B^T$ for $\widehat{S}$ and examine their effect on the non-unit eigenvalues. To investigate all three choices simultaneously, we let

$$\widehat{S} = \beta C + \gamma BA^{-1}B^T, \tag{9}$$

where β and γ are either 0 or 1. Note that since A^{-1} may be dense even when A is sparse, the choices S and $BA^{-1}B^T$ are not necessarily practical. However, they provide insight into the best theoretical choice and how different choices affect the quality of $\mathcal{P}_D$.

Since the fractions $v^TCv/v^T\widehat{S}v$ and $v^TBA^{-1}B^Tv/v^T\widehat{S}v$ appear in (7), for our choice of $\widehat{S}$ we shall find the generalised Rayleigh quotient

$$\mu(v) = \frac{v^TBA^{-1}B^Tv}{v^TCv}$$

useful for assessing the quality of our preconditioners. This ratio is bounded, for any $v \in \mathbb{R}^m$, $v \neq 0$, by [22, Theorems 4.2.11 and 7.7.6]

$$0 < \lambda_{\min}(C^{-1}BA^{-1}B^T) \leq \mu(v) \leq \lambda_{\max}(C^{-1}BA^{-1}B^T) < 1. \tag{10}$$

Substituting (9) for $\widehat{S}$ in (7) and simplifying gives

$$\lambda_{1,2} = \frac{1}{2}\left(1 + \frac{1}{\beta+\mu(v)\gamma}\right) \pm \sqrt{\frac{1}{4}\left(1 - \frac{1}{\beta+\mu(v)\gamma}\right)^2 + \frac{\mu(v)}{\beta+\mu(v)\gamma}}.$$

Table 1 shows the non-unit eigenvalues we expect for each choice of $\widehat{S}$. If $\mu(v) \ll 1$, both $\widehat{S} = C$ and $\widehat{S} = S$ are good choices, since the eigenvalues of $\mathcal{P}_D^{-1}\mathcal{A}$ are clustered near 1. However, $\widehat{S} = BA^{-1}B^T$ is not a good choice, since the eigenvalues of $\mathcal{P}_D^{-1}\mathcal{A}$ may be spread out. The limiting case $\mu(v) = 1$ shows that if $\mu(v)$ is close to 1 then $\mathcal{P}_D^{-1}\mathcal{A}$ will have very small eigenvalues and will, therefore, be ill-conditioned for all three choices. This suggests scaling $\widehat{S}$, i.e., choosing $\omega\widehat{S}$ for some positive scalar ω, to ensure that $\mu(v)$ is not too close to 1. Note that when $\widehat{S} = C$, the largest eigenvalue is always bounded by 2. However, the choice $\widehat{S} = S$ leads to large eigenvalues when $\mu(v)$ is close to 1.

Because the optimal choice of $\widehat{S}$ depends on $\mu(v)$, we compute the extreme values of $\mu(v)$, i.e, the extreme eigenvalues of $C^{-1}BA^{-1}B^T$, for the test problem

Table 1: Approximate values of non-unit eigenvalues of $\mathcal{P}_D^{-1}\mathcal{A}$ for different choices of $\mu(v)$. Note that S is singular when $\mu(v) = 1$.

$\widehat{S}$	$\lambda_{1,2}$	$\mu(v) \ll 1$	$\mu(v) = 1$
C	$1 \pm \sqrt{\mu(v)}$	$1 \pm \sqrt{\mu(v)}$	0, 2
$BA^{-1}B^T$	$\frac{1}{2}(1 + \frac{1}{\mu(v)}) \pm \sqrt{\frac{1}{4}(1 - \frac{1}{\mu(v)})^2 + 1}$	1, $\frac{1}{\mu(v)}$	0, 2
S	$\frac{1}{2}(1 + \frac{1}{1-\mu(v)}) \pm \sqrt{\frac{1}{4}(1 - \frac{1}{1-\mu(v)})^2 + \frac{\mu(v)}{1-\mu(v)}}$	1, $\frac{1}{1-\mu(v)}$	—

from Section 2.4 with both uniform and Halton points, with the results given in Tables 2 and 3. We see that for both point sets the smallest value of $\mu(v)$ for this problem is on the order of 10^{-2} or 10^{-3} and that the largest is near 1. The spread of eigenvalues means that the best choice appears to be $\widehat{S} = C$, since then non-unit eigenvalues of $\mathcal{P}_D^{-1}\mathcal{A}$ are near 1 when $\mu(v)$ is small and for any value of $\mu(v)$ the largest eigenvalue is bounded by 2. Our numerical results in Section 5.1 confirm this. Note that this gives an analogous preconditioner to that proposed by Le Gia and Tran [29]. However, we caution that if eigenvalues of $\mathcal{P}_D^{-1}\mathcal{A}$ become too small, the convergence rate of the conjugate gradient method may decrease.

Table 2: Extreme eigenvalues of $C^{-1}BA^{-1}B^T$ for different problem sizes and the uniform point set.

$n+m$	289	1 089	4 225	16 641
$\lambda_{\min}(C^{-1}BA^{-1}B^T)$	0.0011	0.0030	0.0065	0.012
$\lambda_{\max}(C^{-1}BA^{-1}B^T)$	0.77	0.88	0.94	0.97

Table 3: Extreme eigenvalues of $C^{-1}BA^{-1}B^T$ for different problem sizes and the Halton point set.

$n+m$	358	1 369	5 227	19 345
$\lambda_{\min}(C^{-1}BA^{-1}B^T)$	0.072	0.013	0.015	0.021
$\lambda_{\max}(C^{-1}BA^{-1}B^T)$	0.87	0.93	0.96	0.98

The preconditioner $\mathcal{P}_D$ utilizes blocks of the scaled matrix $\mathcal{A} = \mathcal{M}^{-1}\bar{\mathcal{A}}\mathcal{M}^{-1}$ in (5). These blocks are not available if we do not explicitly compute $\mathcal{A}$, but instead perform diagonal scalings of vectors within our Krylov subspace method as discussed at the end of Section 2.2. In this case diagonal scalings of vectors can also be used to solve systems with $\mathcal{P}_D$, as we now discuss. At each iteration of the preconditioned conjugate gradient method we must solve a system of the form $\mathcal{P}_D u = v$ for some vector $v \in \mathbb{R}^{n+m}$. We can write $\mathcal{P}_D = \mathcal{M}^{-1}\bar{\mathcal{P}}_D\mathcal{M}^{-1}$, where

$$\bar{\mathcal{P}}_D = \begin{bmatrix} \bar{A} & 0 \\ 0 & \bar{S} \end{bmatrix}$$

and $\bar{S}$ is formed from $\bar{A}$, $\bar{B}$ and $\bar{C}$, analogously to $\widehat{S}$. Accordingly, solving a system with $\mathcal{P}_D$ requires $2(n+m)$ multiplications in addition to the cost of solving a system with $\bar{\mathcal{P}}_D$.

3.2 Block triangular preconditioners

We now consider block triangular preconditioners for (5). Since $\mathcal{A}$ has the block decomposition

$$\begin{bmatrix} A & B^T \\ B & C \end{bmatrix} = \begin{bmatrix} A & 0 \\ B & S \end{bmatrix} \begin{bmatrix} I & A^{-1}B^T \\ 0 & I \end{bmatrix},$$

we choose the preconditioner

$$\mathcal{P}_T = \begin{bmatrix} A & 0 \\ B & \widehat{S} \end{bmatrix}. \tag{11}$$

It is easy to see that

$$\mathcal{P}_T^{-1}\mathcal{A} = \begin{bmatrix} I & A^{-1}B^T \\ 0 & \widehat{S}^{-1}S \end{bmatrix}$$

and so the ideal choice is $\widehat{S} = S$, since then all eigenvalues of $\mathcal{P}_T^{-1}\mathcal{A}$ are 1 and GMRES converges in at most two steps [23]. However, the Schur complement is prohibitively expensive to apply and we are unaware of a spectrally equivalent approximation. If $\widehat{S} \neq S$ then $\mathcal{P}_T^{-1}\mathcal{A}$ has n eigenvalues at 1 and the remainder are the eigenvalues of $\widehat{S}^{-1}S$. We again consider the choices $\widehat{S} = C$ and $\widehat{S} = BA^{-1}B^T$, which correspond to each term in the ideal choice S.

When $\widehat{S} = C$ the eigenvalues of $\widehat{S}^{-1}S$ are $1 - \eta$, where η is an eigenvalue of $C^{-1}BA^{-1}B^T$, and (10) shows the eigenvalues of $\widehat{S}^{-1}S$ lie in (0,1). Thus, the eigenvalues of $\mathcal{P}_T^{-1}\mathcal{A}$ are contained in an interval but $\eta \approx 1$ will result in small eigenvalues, which may cause slow convergence rates. When $\widehat{S} = BA^{-1}B^T$ the eigenvalues of $\widehat{S}^{-1}S$ are given by $1/\eta - 1$. If η is small the eigenvalues of $\mathcal{P}_T^{-1}\mathcal{A}$ may be spread out while if $\eta \approx 1$ some eigenvalues of $\mathcal{P}_T^{-1}\mathcal{A}$ may be very small; both situations can negatively affect the speed of convergence. Our numerical experiments in Section 5.1 verify that C is a better choice for $\widehat{S}$ than $BA^{-1}B^T$ for our problem.

Note that if we do not explicitly form $\mathcal{A}$ in (5) when using our iterative solver then we must somehow apply the appropriate diagonal scalings in $\mathcal{P}_T$. This scaling procedure, which is analogous to that for $\mathcal{P}_D$, requires $2(n+m)$ multiplications at each iteration.

4 Domain decomposition preconditioner

We see that preconditioners built from blocks A, B and C, and the Schur complement S, are effective at reducing the number of Krylov subspace iterations required to solve (5). However, these preconditioners are costly to apply. For

larger problems, A is typically much bigger than C, since there are usually many more interior than boundary points. Consequently, we examine a two-level restricted additive Schwarz (RAS) domain decomposition preconditioner [4] for A, although we note that the same procedure could easily be applied to C as well. Although the RAS preconditioner is nonsymmetric it requires less communication than the additive Schwarz (AS) method, making it better suited to parallel implementations. (Note that the block triangular discussed in the previous section is already nonsymmetric.) Moreover, the rate of convergence of RAS methods is typically similar to, or better than, that of AS methods [4, 16].

In the traditional overlapping AS method, we divide the domain $V = \Omega$ into a set of disjoint subdomains, so that $V = V_{1,0} + V_{2,0} + ... + V_{k,0}$. These subdomains are then extended by width θ to give $V = V_{1,\theta} + V_{2,\theta} + ... + V_{k,\theta}$ (see Figure 2a). The restriction of V to the ith domain $V_{i,\theta}$ is associated with the operator $R_{i,\theta}$ while the corresponding prolongation operator is $R_{i,\theta}^T$. The restriction of A to the ith domain is $A_{i,\theta} = R_{i,\theta} A R_{i,\theta}^T$ so that $A_{i,\theta}$ contains the rows and columns of A associated with the data sites in subdomain $V_{i,\theta}$. Then, the additive Schwarz (AS) preconditioner M of A is

$$M_{AS}^{-1} = \sum_{i=1}^{k} R_{i,\theta}^T A_{i,\theta}^{-1} R_{i,\theta}.$$

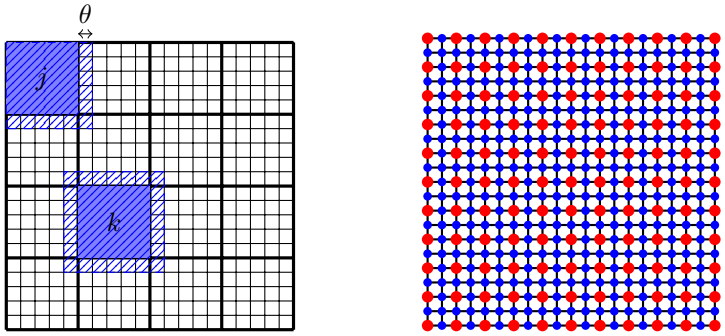
The restricted additive Schwarz preconditioner is a slight modification of the AS preconditioner that uses the overlap for computations but only projects information from the non-overlapping domain. In this case the prolongation operator becomes $R_{i,0}$, and the RAS preconditioner is

$$M_{RAS}^{-1} = \sum_{i=1}^{k} R_{i,0}^T A_{i,\theta}^{-1} R_{i,\theta}.$$

The subdomain solves in the AS and RAS methods could be replaced by inexact solves, and all subdomain solves can be computed in parallel, but we do not consider this here. As discussed for the ideal preconditioners in the previous section, we may wish to apply the diagonal scaling matrix $\mathcal{M}$ to form $\mathcal{A}$ in (5) implicitly in our Krylov sovler. However, applying the AS or RAS preconditioner requires the diagonally scaled matrix A in (5) to be computed explicitly, so that solves with the small matrices $A_{i,\theta}$ can be performed. This scaling is performed once at the start of computations and requires $2kn$ multiplications, where $k = k(n)$ is the number of nonzeros in $\bar{A}$. This scaling can also be performed in parallel. Since $n \gg m$ in most applications of interest, when using the AS or RAS preconditioners it may be more economical to explicitly compute $\mathcal{A}$ than to implicitly scale by $\mathcal{M}$ within the Krylov solver.

As the number of subdomains increases, the speed of convergence of the RAS scheme can be slow because information takes longer to propagate through all subdomains. However, this slow convergence can be remedied by incorporating a coarse-grid correction to give a two-level, or augmented, RAS method (ARASM). We consider only additive corrections and let $V_0 \subset V$ be the set

Figure 2: (a) Additive Schwarz subdomains on the boundary (j) and in the interior (k). The overlapping domain is shown with diagonal lines and the non-overlapping domain is shaded. The overlap is θ because the overlapping domain is obtained from the non-overlapping domain by extending by θ in each direction. (b) Coarse and fine grid points for uniform points with coarse grid points denoted by larger red circles.



(a) (b)

of coarse grid RBF centres and R_0 and R_0^T be the corresponding restriction and prolongation operators. The coarse grid representation of A is then $A_0 = R_0 A R_0^T$ and the two-level preconditioner is

$$M_{ARASM}^{-1} = R_0^T A_0^{-1} R_0 + \sum_{i=1}^{k} R_{i,0}^T A_{i,\theta}^{-1} R_{i,\theta}.$$

Selection of coarse grid points in the case of uniform points is fairly straightforward and in our numerical experiments our coarse grid contains every second node of the fine grid (see Figure 2b). In the case of scattered data, one option is to select a fixed number of points per subdomain [2, 29] as coarse grid points. Another is to start with the finest grid and thin this out, which can be achieved by a Delaunay triangulation [11]. However, in the context of meshless methods the introduction of a mesh may not be desirable. Alternatively, the points can be thinned out by using the furthest point algorithm [10]. Since we deal with a sequence of levels, it is also possible to use the points from a previous level to form the coarse grid. This is particularly appealing in the case of nested sequences of points and in our numerical experiments we use the RBF centres from level $k-1$ to form a coarse grid at level k.

To demonstrate the effectiveness of the ARASM preconditioner, and its sensitivity to the size of the problem, number of domains and overlap, we apply right-preconditioned GMRES with the ARASM preconditioner to the matrix A in the saddle point system (5) for the test problem defined in Section 2.4 with both point sets. The initial vector for GMRES is the zero vector and we stop when the relative residual $\|r_k\|_2/\|r_0\|_2$ falls below 10^{-8}, where $r_k = b^i - A\alpha_k^i$ is

Table 4: GMRES iteration counts for the diagonally preconditioned A in (5) (third column) and RAS preconditioner applied to A for uniform points. The number of subdomains is k and the overlap is θ.

n	η	A	$k=4$			$k=8$			$k=16$		
			$\theta=\eta$	$\theta=2\eta$	$\theta=4\eta$	$\theta=\eta$	$\theta=2\eta$	$\theta=4\eta$	$\theta=\eta$	$\theta=2\eta$	$\theta=4\eta$
225	1/16	25	14	11	7	17	13	8	18	12	8
961	1/32	51	19	15	11	22	19	15	27	18	18
3 969	1/64	106	24	21	18	26	23	22	29	22	26
16 129	1/128	223	28	26	24	29	26	26	33	26	29
65 025	1/256	470	30	29	27	31	28	28	33	28	28

Table 5: GMRES iteration counts for the diagonally preconditioned A in (5) (third column) and RAS preconditioner applied to A for the Halton points. The number of subdomains is k and the overlap is θ.

n	η	A	$k=4$			$k=8$			$k=16$		
			$\theta=\eta$	$\theta=2\eta$	$\theta=4\eta$	$\theta=\eta$	$\theta=2\eta$	$\theta=4\eta$	$\theta=\eta$	$\theta=2\eta$	$\theta=4\eta$
296	1/16	91	19	13	11	43	15	11	88	18	12
1247	1/32	202	24	16	14	37	19	15	187	23	16
4 979	1/64	427	27	22	18	41	24	19	242	30	21
18 848	1/128	903	32	26	23	41	27	24	93	31	23
65 025	1/256	2950	37	28	26	43	30	28	220	36	28

the residual at the kth iteration.

Tables 4 and 5 show the GMRES iteration counts. When just the diagonal preconditioning proposed by Fasshauer is applied, the iteration number grows rapidly with the problem size, particularly for the Halton point problem. In contrast, with the ARASM preconditioner the iteration number has a much more modest dependence on n. Moreover, this dependence is weaker when the overlap is larger. It seems for the matrices generated by Halton points that a larger overlap is needed than for uniform points; because the data is scattered, this is perhaps not surprising. We are not certain for the drop in iterations for 16 subdomains and an overlap of η for the Halton point problem with 18 848 unknowns, although perhaps it is caused by a favourable arrangement of points for our ARASM method.

When using the uniform grid, the resulting matrix A is highly structured. Consequently, for this grid we additionally tested the preconditioner

$$\widehat{M}^{-1}_{ARASM} = R_0^T A_0^{-1} R_0 + \sum_{i=1}^{k} R_{i,0}^T A_{*,\theta}^{-1} R_{i,\theta}.$$

that replaces the $A_{i,\theta}$ in M_{ARASM} with a fixed matrix $A_{*,\theta}$ and is, therefore, cheaper to apply. The matrix $A_{*,\theta}$ is simply the restriction of A to an interior domain and its presence allows us to apply $A_{*,\theta}$ for all interior domains by computing a single LU decomposition. Domains on the boundary do not have overlap on all the edges and so certain rows and columns of $A_{*,\theta}$ have to be removed to make the matrix suitable for these domains. However, this results

Table 6: Dimensions n and m and number of millions of nonzeros in $\mathcal{A}$, PCG iterations to reach a tolerance of 10^{-8} for (5) and condition number of $\mathcal{A}$. Note that the condition number for both level 4 matrices and the level 5 uniform point matrix are estimated using the Matlab function `condest` but it was not possible to estimate the level 5 Halton matrix the condition number using `condest` in a reasonable time.

	Level	1	2	3	4	5
Uniform	n	225	961	3 969	16 129	65 025
	m	64	128	256	512	1 024
	nnz	0.05	0.43	3.5	28	217
	CG Iterations	289	1 089	4 225	16 641	64 049
	Condition number	3.3×10^4	5.5×10^5	9.0×10^6	3.4×10^8	5.4×10^9
Halton	n	296	1 247	4 979	18 848	65 025
	m	62	122	248	497	1 024
	nnz	0.079	0.68	5.4	38	215
	CG Iterations	358	1 369	5 227	19 345	66 049
	Condition number	3.9×10^4	7.7×10^5	1.5×10^7	4.4×10^8	

in only eight additional LU decompositions, one for each of the edges and for each of the corners. Applying this strategy gives identical results to those in Table 4 but is significantly cheaper to apply.

In summary, it seems that the ARASM preconditioner is a good approximation of A in the block diagonal and block triangular preconditioners. We verify this in the following section.

5 Numerical results

In this section we apply our preconditioners to the test problem described in Section 2.4. Table 6 shows the problem sizes we consider, from which we see that, with the exception of the last level, the matrices generated with Halton points are larger and have more nonzeros than the matrices generated with uniform points. Note that when the Jacobi diagonal preconditioner is used, conjugate gradients terminates in $n + m$ steps (see Table 6). The condition numbers for the diagonally preconditioned coefficient matrix in (5) are also given in Table 6, from which we see that, at a given level, the matrices generated using Halton points have larger condition number than the corresponding uniform point matrices. This difference could be related to the larger dimension as well as ill-conditioning introduced by points not being as well-separated as in the uniform case. Throughout, we terminate computations when the relative residual $\|r_k\|_2/\|r_0\|_2$ falls below 10^{-8}.

Table 7: Iteration counts for PCG to reach a tolerance of 10^{-8} with the diagonally preconditioned matrix $\mathcal{A}$ in (5) and the ideal block diagonal preconditioner for levels $1, \ldots, 5$. We denote by DNF problems for which the tolerance was not reached within 10 hours.

	$\widehat{S}$	1	2	3	4	5
Uniform	C	27	38	46	56	66
	$BA^{-1}B^T$	80	114	129	149	168
	S	32	51	74	108	149
Halton	C	46	65	86	111	142
	$BA^{-1}B^T$	116	186	299	415	DNF
	S	62	112	180	281	DNF

5.1 Ideal preconditioner

The ideal preconditioners $\mathcal{P}_D$ and $\mathcal{P}_T$ described in Section 3 are examined first. We apply conjugate gradients when the block diagonal preconditioner $\mathcal{P}_D$ is used and right-preconditioned GMRES when the block triangular preconditioner $\mathcal{P}_T$ is used.

From Table 7 we see that, as predicted in Section 3.1, $\widehat{S} = C$ gives the fastest convergence rate for the block diagonal preconditioner $\mathcal{P}_D$. Although the convergence speed is mesh-dependent, the iteration growth is fairly modest, particularly for uniform points. The larger iteration counts for the Halton points are not unexpected, as the matrices $\mathcal{A}$ are of larger dimension, have more nonzeros, and become more ill-conditioned as the dimension grows than the matrices for uniform points (see Table 6). On the other hand, CG requires more iterations when $BA^{-1}B^T$ and S are used and the iteration growth is more rapid. Note that, as discussed in Section 3.1, when $\mu(v)$ is close to 1 the non-unit eigenvalues of $\mathcal{P}_D^{-1}\mathcal{A}$ may be large when $\widehat{S} = S$, while the largest eigenvalue is bounded by 2 when $\widehat{S} = C$. This explains the lower iteration counts for the less expensive choice C. Additionally, for the largest problem and the Halton points, the cost of solving with the exact matrix A in the preconditioner, combined with the high iteration counts, meant that we were unable to reach the tolerance within 10 hours when $\widehat{S} = BA^{-1}B^T$ or $\widehat{S} = S$. Additional computations, which for brevity are not reported, show that GMRES achieves similar iteration counts to CG when $\widehat{S} = C$.

The CG iteration counts are reflected in the condition numbers of $\mathcal{P}_D^{-1}\mathcal{A}$ (see Table 8) which are smallest when $\widehat{S} = C$. In addition, $\widehat{S} = C$ has smaller condition number growth than that for the exact Schur complement S. (Note that, because of their size and the cost of forming $\mathcal{P}_D^{-1}\mathcal{A}$, we were unable to use `condest` to estimate the condition numbers for the level 4 Halton point matrices.)

We consider now the ideal block triangular preconditioner $\mathcal{P}_T$ in (11). As predicted in Section 3.2, GMRES converges in 2 steps when $\widehat{S} = S$ (see Table 9). However, forming the Schur complement is prohibitively expensive for large

Table 8: Condition numbers of $\mathcal{P}_D^{-1}\mathcal{A}$ for levels 1–4. Note that the condition numbers for the level 4 uniform point matrix are estimated using the Matlab function `condest`. It was not possible to estimate the condition numbers for the level 4 Halton matrix and both level 5 matrices using `condest` in a reasonable time.

	$\widehat{S}$	1	2	3	4
Uniform	C	2.9×10^2	2.6×10^3	2.9×10^4	3.4×10^5
	$BA^{-1}B^T$	4.1×10^5	1.7×10^6	9.2×10^6	3.6×10^7
	S	1.3×10^3	2.5×10^4	5.4×10^5	7.3×10^6
Halton	C	1.1×10^3	1.4×10^4	1.0×10^5	
	$BA^{-1}B^T$	2.3×10^5	1.6×10^6	1.8×10^7	
	S	6.5×10^3	2.0×10^5	2.8×10^6	

Table 9: Iteration counts for GMRES to reach a tolerance of 10^{-8} with the ideal block triangular preconditioner for levels $1, \ldots, 5$. We denote by DNF problems for which the tolerance was not reached within 10 hours.

	$\widehat{S}$	1	2	3	4	5
Uniform	C	14	20	23	28	29
	$BA^{-1}B^T$	33	48	62	76	79
	S	2	2	2	2	2
Halton	C	24	34	43	56	71
	$BA^{-1}B^T$	45	68	105	170	DNF
	S	2	2	2	2	2

problems. The choice $\widehat{S} = C$ is a reasonable alternative and performs better than $BA^{-1}B^T$, as we might expect from the analysis in Section 3.2. An added bonus for uniform points is that when $\widehat{S} = C$ the growth in the iteration count appears to slow as the problem size increases. The condition number of $\mathcal{P}_T^{-1}\mathcal{A}$ is not as relevant to the convergence rate as that of $\mathcal{P}_D^{-1}\mathcal{A}$, because $\mathcal{P}_T^{-1}\mathcal{A}$ is nonsymmetric, and we do not report this information. However, for all problem sizes the condition numbers of $\mathcal{P}_T^{-1}\mathcal{A}$ are of similar magnitudes to those of $\mathcal{P}_D^{-1}\mathcal{A}$ for $\widehat{S} = C$ and $\widehat{S} = BA^{-1}B^T$ and are less than 100 for the Schur complement S.

5.2 Additive Schwarz preconditioners

Although Tables 7 and 9 show that we can develop preconditioners that significantly reduce the number of GMRES iterations, our ideal preconditioners require a linear solve with A at each iteration. As the level increases, both the dimension of A and the density of nonzeros grow, making this solve costly. Consequently, we see how the preconditioners are affected by replacing A by the ARASM preconditioner described in Section 4. The ARASM preconditioner is nonsymmetric and so for both the block diagonal preconditioner and block tri-

Table 10: Iteration counts for GMRES to reach a tolerance of 10^{-8} with the block preconditioners, using the additive Schwarz preconditioner for the (1,1) block A, for levels $2, \ldots, 5$.

		2	3	4	5
Uniform	Block diagonal	41	53	57	66
	Block triangular	28	34	46	51
Halton	Block diagonal	77	94	123	151
	Block triangular	53	61	78	99

Table 11: Time in seconds for GMRES to reach a tolerance of 10^{-8} for the Jacobi preconditioner and both block preconditioners applied to the Halton point matrices for levels 2,...,5. In the block preconditioners we use the additive Schwarz preconditioner for the (1,1) block A.

	2	3	4	5
Diagonal	1.4	44	1 388	39 876
Block diagonal	1.1	12	128	1 771
Block triangular	0.71	7.8	77	1 244

angular preconditioner we apply right-preconditioned GMRES and set $\widehat{S} = C$. If $n < 2000$ we use 4 subdomains and an overlap of $\theta = 2\eta$, where η is as in Tables 4 and 5. Otherwise, we use 8 subdomains and an overlap of $\theta = 4\eta$.

The ARASM block diagonal preconditioner gives higher iteration counts than the corresponding ideal preconditioner (see Table 7) for small problems but both perform similarly for larger problems. We caution that conjugate gradients is used with the ideal preconditioner and so the results are not directly comparable with those in Table 7 but, as noted in Section 5.1, GMRES behaves similarly to CG when the ideal block diagonal preconditioner (6) is applied with $\widehat{S} = C$. The number of iterations of the ARASM block triangular preconditioner is larger than for the ideal preconditioner, but the growth with dimension is still modest and iteration counts are lower than for the ARASM block diagonal preconditioner. Moreover, the time required to solve the system by the block preconditioned GMRES method is more than 20 times faster than applying CG to the Jacobi preconditioned system (5) for the largest matrix generated by Halton points (see Table 11).

6 Conclusions

We have developed block diagonal and block triangular preconditioners for the symmetric positive definite systems that arise in the RBF multiscale collocation method for PDEs and have described the spectra of the preconditioned matrices for different choices of (2,2) block $\widehat{S}$. We find, analytically and experimentally, that the block diagonal preconditioner with $\widehat{S} = C$ has the most favourable

spectrum for fast CG convergence. On the other hand, choosing $\widehat{S} = S = C - BA^{-1}B^T$ in the block triangular preconditioner guarantees convergence of GMRES in two iterations. However, the block triangular preconditioner is costly to form when $\widehat{S} = S$ and so a reasonable alternative is to choose $\widehat{S} = C$ in the block triangular preconditioner. Practical preconditioners that replace A by a restricted additive Schwarz method augmented with a coarse-grid correction (ARASM) were introduced and numerical experiments show that they are effective. Further speed-ups could be achieved by using inexact solves within the ARASM preconditioner and by exploiting parallelism. Additionally, a spectrally equivalent approximation to $\widehat{S}$ should improve the rate of convergence of the block triangular preconditioned system.

Acknowledgements We thank the referees for their helpful comments that improved the paper. This work was supported in part by award KUK- C1-013-04, made by King Abdullah University of Science and Technology (KAUST).

References

[1] R. K. Beatson, J. B. Cherrie, and C. T. Mouat. Fast fitting of radial basis functions: Methods based on preconditioned GMRES iteration. *Advances in Computational Mathematics*, 11:253–270, 1999.

[2] R. K. Beatson, W. A. Light, and S. Billings. Fast solution of the radial basis function interpolation equations: Domain decomposition methods. *SIAM Journal on Scientific Computing*, 22:1717–1740, 2000.

[3] M. D. Buhmann. *Radial Basis Functions: Theory and Implementations*, volume 12 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2003.

[4] X.-C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM Journal on Scientific Computing*, 21:792–797, 1999.

[5] A. Chernih and S. Hubbert. Closed form representations and properties of the generalised Wendland functions. *Journal of Approximation Theory*, 177:17–33, 2014.

[6] Q. Deng and T. A. Driscoll. A fast treecode with multiquadric interpolation with varying shape parameters. *SIAM Journal on Scientific Computing*, 34:A1126–A1140, 2012.

[7] P. Farrell and H. Wendland. RBF multiscale collocation for second order elliptic boundary value problems. *SIAM Journal on Numerical Analysis*, 51(4):2403–2425, 2013.

[8] G. E. Fasshauer. Solving differential equations with radial basis functions: Multilevel methods and smoothing. *Advances in Computational Mathematics*, 11:139–159, 1999.

[9] G. E. Fasshauer. *Meshfree Approximation Methods with MATLAB*, volume 6 of *Interdisciplinary Mathematical Sciences*. World Scientific Publishing, Hackensack, NJ, 2007.

[10] T. Feder and D. Greene. Optimal Algorithms for Approximate Clustering. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 434–444, New York, USA, 1988. ACM.

[11] M. S. Floater and A. Iske. Multistep scattered data interpolation using compactly supported radial basis functions. *Journal of Computational and Applied Mathematics*, 73:65–78, 1996.

[12] N. Flyer and G. B. Wright. Transport schemes on a sphere using radial basis functions. *Journal of Computational Physics*, 226:1059–1084, 2007.

[13] B. Fornberg and E. Lehto. Stabilization of RBF-generated finite difference methods for convective PDEs. *Journal of Computational Physics*, 230(6):2270–2285, 2011.

[14] C. Franke and R. Schaback. Convergence order estimates of meshless collocation methods using radial basis functions. *Advances in Computational Mathematics*, 8:381–399, 1998.

[15] R. W. Freund and N. M. Nachtigal. QMR: A quasi-minimal residual method for non-Hermitian linear systems. *Numerische Mathematik*, 60:315–339, 1991.

[16] A. Frommer and D. B. Szyld. An algebraic convergence theory for restricted additive Schwarz methods using weighted max norms. *SIAM Journal on Numerical Analysis*, 39:463–479, 2001.

[17] E. J. Fuselier. Sobolev-type approximation rates for divergence-free and curl-free RBF interpolants. *Mathematics of Computation*, 77(263):1407–1423, 2008.

[18] P. Giesl and H. Wendland. Meshless collocation: Error estimates with application to dynamical systems. *SIAM Journal on Numerical Analysis*, 45(4):1723–1741, 2007.

[19] A. Greenbaum. *Iterative Methods for Solving Linear Systems*. SIAM, Philadelphia, PA, 1997.

[20] J. Halton. On the Efficiency of Certain Quasi-Random Sequences of Points in Evaluating Multi-Dimensional Integrals. *Numer. Math.*, 2(1):84–90, 1960.

[21] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, 1952.

[22] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1990.

[23] I. C. F. Ipsen. A note on preconditioning nonsymmetric matrices. *SIAM Journal on Scientific Computing*, 23:1050–1051, 2001.

[24] A. Iske. *Multiresolution Methods in Scattered Data Modelling*, volume 37 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin, 2004.

[25] E. J. Kansa. Multiquadrics—A scattered data approximation scheme with applications to computational fluid-dynamics—I Surface approximations and partial derivative estimates. *Computers & Mathematics with Applications*, 19(8–9):127–145, 1990.

[26] Y. A. Kuznetsov. Efficient iterative solvers for elliptic finite element problems on nonmatching grids. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 10:187–211, 1995.

[27] Q. T. Le Gia, I. H. Sloan, and T. Tran. Overlapping additive Schwarz preconditioners for elliptic PDEs on the unit sphere. *Mathematics of Computation*, 78:79–101, 2009.

[28] Q. T. Le Gia, I. H. Sloan, and H. Wendland. Multiscale RBF collocation for solving PDEs on spheres. *Numerische Mathematik*, 121(1):99–125, 2012.

[29] Q. T. Le Gia and T. Tran. Fast iterative solvers for boundary value problems on a local spherical region. *ANZIAM Journal*, 54:C492–C507, 2013.

[30] L. Ling and E. J. Kansa. Preconditioning for radial basis functions with domain decomposition methods. *Mathematical and Computer Modelling*, 40:1413–1427, 2004.

[31] L. Ling and E. J. Kansa. A least-squares preconditioner for radial basis functions collocation methods. *Advances in Computational Mathematics*, 23:31–54, 2005.

[32] W. R. Madych and S. A. Nelson. Multivariate interpolation and conditionally positive definite functions. II. *Mathematics of Computation*, 54:211–230, 1990.

[33] M. F. Murphy, G. H. Golub, and A. J. Wathen. A note on preconditioning for indefinite linear systems. *SIAM Journal on Scientific Computing*, 21:1969–1972, 2000.

[34] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7:856–869, 1986.

[35] D. Schräder and H. Wendland. A high-order, analytically divergence-free discretization method for Darcy's problem. *Mathematics of Computation*, 80(273):263–277, 2011.

[36] H. A. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of BiCG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13:631–644, 1992.

[37] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4(4):389–396, 1995.

[38] H. Wendland. *Scattered Data Approximation*, volume 17 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2005.

[39] H. Wendland. Divergence-free kernel methods for approximating the Stokes problem. *SIAM Journal on Numerical Analysis*, 47(4):3158–3179, 2009.

[40] H. Wendland. Multiscale analysis in Sobolev spaces on bounded domains. *Numerische Mathematik*, 116(3):493–517, 2010.

[41] R. Yokota, L. Barba, and M. G. Knepley. PetRBF – A parallel $O(N)$ algorithm for radial basis function interpolation with Gaussians. *Computer Methods in Applied Mechanics and Engineering*, 199:1793–1804, 2010.