Radial basis function methods for optimal control of the convection-diffusion equation*

Pedro González Casanova†1 and Jorge Zavaleta1

1Instituto de Matemáticas, Universidad Nacional Autónoma de México, Ciudad Universitaria, Ciudad de México 04510, MEXICO

Abstract

PDE-constrained optimization problems have been barely solved by radial basis functions (RBFs) methods [5]. It is well known that RBF methods can attain an exponential rate of convergence when $C^\infty$ kernels are used, also, these techniques, which are truly scattered, are known to be flexible to discretize complex boundaries in several dimensions. On the other hand, exponential convergence implies an exponential growth of the condition number of the Gram matrix associated with these meshfree methods and global collocation techniques are known to be computationally expensive. In this paper, and in the context of optimal constrained optimization problems, we aim to explore a possible answer to both problems. Specifically, we introduce two local RBF methods: LAM-DQ based in the combination of an asymmetric local method (LAM), inspired in local Hermite interpolation (LHI), with the differential quadrature method (DQ), and LAM-LAM which use twice the local asymmetric method. The efficiency of these local methods against global collocation by solving several synthetic convection-diffusion control problems is analyzed. In this article, we also propose a preconditioning technique and treat the ill-conditioning problem by using extended arithmetic precision. We think that these local methods, which are highly parallelizable, shows a possible way to solve massive optimization control problems in an efficient way.

Keywords: Radial basis functions, PDE-constrained optimization problems, convection-diffusion control, RBF local method

1 Introduction

Several works have appeared in the literature which deals with the analysis and formulation of numerical methods for the solution of distributed control problems in two or three dimension [13]. These works have been formulated within the context of two general frames: the discretized-optimized and/or the optimize-discretized approaches. In particular, Galerkin methods have been proposed and analyzed within both frames, (see [13] and references therein). On the other hand, we could only find one article in the literature, which uses radial basis function methods for constrained optimization problems, see [5]. More precisely in [5] the author solves Poisson constrained optimization problems by using global RBFs symmetric and asymmetric collocation techniques.

*Preprint
†Corresponding author, email address: casanova@matem.unam.mx
It is well known that a major limitation of global RBFs collocation techniques is that as the number of nodes or the shape parameter increases, the condition number of the corresponding Gram matrix grows. In the case of infinitely differentiable RBFs, convergence can be exponential but the corresponding condition number also increases in an exponential way [8].

The current article is formulated within the context of the optimize-then-discretize approach and has the following objectives:

• Extend the application of RBFs meshfree methods to solve convection-diffusion constrained optimization problems.

• Use local RBFs methods, or more precisely, introduce a local asymmetric version, LAM, inspired on the local Hermite interpolation technique, LHI, [10].

• Show that the proposed local methods can attain the same accuracy but with the advantage of a considerable reduction of the computing time, giving the possibility to extend its application for massive problems.

• Introduce a simple but effective preconditioner to invert the local matrices of the LAM method.

We find that the local interpolation method, LHI, applied to the convection-diffusion control problem gives rise to a saddle point problem which is well known, in general, to be ill-posed. Although we manage to prove that under certain conditions the saddle point matrix is invertible, the result is not strong enough and we thus formulated a different approach. Specifically, we decouple the Euler Lagrange system of equations, corresponding to the control problem, by formulating a biharmonic problem which let us compute the state variable through LAM, and once this is done we use the state variable to calculate the control by DQ or using again LAM.

It is worth to mention that, an important alternative to the ill-conditioned problem of RBFs collocation methods, is that they can be solved by using domain decomposition methods, see, [2], [1], [12]. In this work, however, we are interested in comparing the results of the local methods with those obtained by solving the global asymmetric collocation (AC) method by using direct solvers with quad precision which is a current alternative that has been used by Kansa [3] and Sarra, [8] among others.

This paper is organized as follows. In section 2, we briefly state the continuous control problem and refer the reader to the proper references. Section 3, is devoted to formulate the LAM-LAM and LAM-DQ local methods for the solution of the convection-diffusion constrained optimization problems as well as the preconditioning technique. Section 4, we present numerical examples to show the capabilities of the local methods. In section 5, conclusions are presented.

2 The convection-diffusion control problem

Throughout this paper, we will be concerned with the solution of the following distributed control problem

\[
\min_{y,u} \frac{1}{2} \| y - \hat{y}\|_{L^2(\Omega)}^2 + \frac{\beta}{2} \| u\|_{L^2(\Omega)}^2
\]

s.t. \( \mathcal{E} y = u \) in \( \Omega \), \( B y = g \) on \( \partial\Omega \)

(1)

where, \( y \) is the state, \( u \) the control, \( \hat{y} \) the objective state, \( \beta > 0 \) the penalty constant, \( \mathcal{E} \) is a PDE stationary linear operator and \( B \) a Dirichlet, Neumann or Robin, boundary operator. Such problems were introduced by Lions in [4].
The distributed control problem (1) can be equivalently formulated as a functional that incorporates the PDE constraints by means of Lagrange multipliers, (see [6]), namely as

\[
\mathcal{L}(y,u,p_1,p_2) = \frac{1}{2} \|y - \hat{y}\|^2_{L^2(\Omega)} + \frac{\beta}{2} \|u\|^2_{L^2(\Omega)} + \int_{\Omega} (Ey - u) p_1 + \int_{\partial\Omega} (By - g) p_2. \tag{2}
\]

Taking the Frechet derivative of the functional (2) with respect to \(y, u\) and \(p\) it is possible to obtain the following Euler Lagrange equations in terms of the state \(y\) and the control variable \(u\),

\[
\begin{align*}
Ey &= u \quad \text{in } \Omega \\
By &= g \quad \text{on } \partial\Omega \\
\beta E^* u &= \hat{y} - y \quad \text{in } \Omega \\
u &= 0 \quad \text{on } \partial\Omega
\end{align*}
\tag{3}
\]

where we have eliminated \(p\) by using the equation \(p = \beta u\). Here, \(y\) and \(u\) satisfying (3) are known as the optimal state and optimal control, respectively.

3 Numerical schemes

The following schemes will be discretized by using multiquadric RBFs \(i.e. \Phi(x) = \sqrt{c + \|x\|^2}\), where \(c\) is the shape parameter. We first describe the global asymmetric collocation and local methods to solve the minimization problem.

3.1 Asymmetric collocation

In order to formulate the global asymmetric collocation scheme for the former coupled pair of equations (3) we first define the following ansatz

\[
y(x) = H(x)\lambda, \quad u(x) = H(x)\mu,
\]

where \(H\) is known as the reconstruction vector, taken here as usual as

\[
H(x) = \begin{bmatrix} \Phi(x - x_i) \\ p_\ell(x) \end{bmatrix} \in \mathbb{R}^{n+n_p},
\]

with \(n\) the number of nodes and \(n_p\) the number of polynomial terms. Taking the first \(n_b < n\) nodes to be the boundary nodes, the resulting system of linear equations is given by

\[
\begin{bmatrix} G^B & \beta E^* \\ -E & G \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} d \\ 0 \end{bmatrix},
\]

where

\[
E = \begin{bmatrix} \mathcal{E}\Phi_\Omega & 0 \\ 0 & \mathcal{E}P_\Omega \end{bmatrix}, \quad E_* = \begin{bmatrix} 0 & 0 \\ \mathcal{E}^*\Phi_\Omega & \mathcal{E}^*P_\Omega \end{bmatrix}, \quad G^B = \begin{bmatrix} \mathcal{B}\Phi_{\partial\Omega} & \mathcal{B}P_{\partial\Omega} \\ \Phi_\Omega & P_\Omega \end{bmatrix},
\]

are square matrices of size \((n+n_p) \times (n+n_p)\), and \((\mathcal{B}\Phi_{\partial\Omega})_{j,i} = \mathcal{B}\Phi(x_j - x_i), (\mathcal{B}P_{\partial\Omega})_{j,\ell} = \mathcal{B}p_\ell(x_j), (\mathcal{E}\Phi_\Omega)_{k,i} = \mathcal{E}\Phi(x_k - x_i), (\mathcal{E}P_\Omega)_{k,\ell} = \mathcal{E}p_\ell(x_k)\), for \(\mathcal{E} = \mathcal{E}^*, \mathcal{E}, I\), with \(I\) the identity operator, \(G := G^B\) is the standard Gram matrix for \(\mathcal{B} = I\), \(P^t = \begin{bmatrix} P_{\partial\Omega}^t \\ P_\Omega^t \end{bmatrix}\) and

\[
d = \begin{bmatrix} g(x_j) \\ \hat{y}(x_k) \end{bmatrix}_{1 \leq j \leq n_b} \begin{bmatrix} 0 \\ 1 \leq \ell \leq n_p \end{bmatrix}^t.
\]
If \( G^{B} = G \), i.e. taking \( B = I \), we solve this system through block LU factorization as follows:

\[
\begin{bmatrix}
  G & \beta E_* \\
  -E & G
\end{bmatrix}
= \begin{bmatrix}
  I_{n+n_p} & 0 \\
  -EG^{-1} & I_{n+n_p}
\end{bmatrix}
\begin{bmatrix}
  G & \beta E_* \\
  0 & R
\end{bmatrix},
\]

where \( R = G + \beta E^{-1}G \), the Schur complement of \( G \) and \( I_n \) is the identity matrix of size \( n \).

### 3.2 A local asymmetric scheme

The system of equations (3) can be easily shown to be equivalent to the following boundary value elliptic problem, assuming \( y \) is smooth enough

\[
\begin{align*}
  M y &= \hat{y} & \text{in } \Omega \\
  \mathcal{E} y &= 0 & \text{on } \partial \Omega \\
  B y &= g & \text{on } \partial \Omega
\end{align*}
\]

where the differential operator \( M \) is given by \( M = I + \beta \mathcal{E}^* \mathcal{E} \). Although system (3) can be directly discretized, it involves the solution of a saddle point problem, which is well known to be singular unless special conditions, e.g. inf-sup conditions, in the case of finite elements, are imposed. We thus find it more convenient to use system (4) to compute the numerical solution.

To formulate the LAM scheme of the system (4) we consider the following: Let \( X \subset \Omega \) be a set of \( n \) scattered nodes and let \( X_c \) be a subset of \( n_c \) nodes. Consider neighborhoods \( D_k \) (e.g. a disc of fixed radius) around the \( k \)-th point of \( X_c \) and label the nodes of \( D_k \cap X \) so that:

- There are \( n^{(k)} \) nodes in \( D_k \) i.e. \( n^{(k)} = \#(X \cap D_k) \).
- The first node, \( x^{(k)}_1 \) is the center of \( D_k \).
- The first \( n_c^{(k)} \) nodes are centers of other discs, i.e. \( n_c^{(k)} = \#(X_c \cap D_k) \).
- The following \( n_b^{(k)} \) nodes lie on \( \partial \Omega \), \( n_b^{(k)} = \#(\partial \Omega \cap (D_k \setminus X_c)) \).
- The remaining \( n_\iota^{(k)} \) nodes belong to the interior of \( \Omega \) (and none of them are centers of any disc), so that \( n^{(k)} = n_c^{(k)} + n_b^{(k)} + n_\iota^{(k)} \).

For each disk, the method forms a local system whose solutions are used to build a global sparse matrix. The solution of this global system gives the approximated values of the PDE system (4) at the centers \( X_c \subset \Omega \).

Choosing a conditionally positive definite radial basis function \( \Phi \) of order \( m \) and let \( n_p \) be dimension of the corresponding polynomial space, we define the reconstruction vector

\[
H^{(k)}(x) = \begin{bmatrix}
  \Phi \left( x - x^{(k)}_j \right) \\
  \frac{p_\ell(x)}{1 \leq j \leq n^{(k)}}
\end{bmatrix} \in \mathbb{R}^{n^{(k)} + n_p}.
\]

Defining the following ansatz

\[
y^{(k)}(x) = H^{(k)}(x) \lambda^{(k)},
\]
we obtain the local linear system

\[ A^{(k)} \lambda^{(k)} = \begin{bmatrix} \Phi & P \\ B\Phi & BP \\ \mathcal{E}\Phi & \mathcal{E}P \\ \mathcal{M}\Phi & \mathcal{MP} \\ P^t & 0 \end{bmatrix} \lambda^{(k)} = d^{(k)} \]

with the data vector

\[ d^{(k)} = \left[ y\left(x_j^{(k)}\right)_{1 \leq j \leq n_b} \, \, \, y\left(x_j^{(k)}\right)_{n_b < j \leq n_b + n_{b_1}} \, \, \, 0 \, \, \, y\left(x_j^{(k)}\right)_{n_b + n_{b_1} < j \leq n_b + n_{b_2}} \, \, \, 1 \right] \]

where \( n_b^{(k)} \) and \( n_{b_1}^{(k)} \) are the number of boundary points for each of the boundary conditions, so that \( n_b = n_{b_1}^{(k)} + n_{b_2}^{(k)} \). Solving for \( \lambda^{(k)} \) we obtain the local solution

\[ y^{(k)}(x) = H^{(k)}(x) \left(A^{(k)}\right)^{-1} d^{(k)} = W^{(k)}(x) d^{(k)}, \]  

(5)

where \( W^{(k)} \) is known as the vector of weights. Using this last expression it is possible to compute \( \mathcal{D}u^{(k)} \) for any differential operator \( \mathcal{D} \) through \( \mathcal{D}u^{(k)}(x) = (\mathcal{D}W^{(k)})(x) d^{(k)} \).

Denote by \( y_c = [y\left(x_j^{(k)}\right)]_{k=1}^{n_c} \in \mathbb{R}^{n_c} \) the vector of the values of \( y \) at each of the centers. Then for each \( k \), the unknown elements of \( d^{(k)} \) belong to \( y_c \).

Consider now the following system of equations

\[ \hat{y}\left(x_1^{(k)}\right) = \mathcal{M} y\left(x_1^{(k)}\right) = \mathcal{M} H^{(k)}(x_1^{(k)}) \left(A^{(k)}\right)^{-1} d^{(k)} = W^{(k)}_{\mathcal{M}} \left(x_1^{(k)}\right) d^{(k)} \]  

(6)

for \( k = 1, \ldots, n_c \) and \( W^{(k)}_{\mathcal{M}} = \mathcal{M} W^{(k)} \).

This is a linear system in \( y_c \), whose elements are the approximated solution of the PDE system (4), at the centers, and which can be written as \( S y_c = b \). Note that since in each \( d^{(k)} \) there are only a few number of centers, \( i.e. \, n_c^{(k)} \) is relatively small, the matrix \( S \) is sparse and thus standard preconditioning techniques can be used.

In order to build the matrix \( S \), we compute the weights by solving the following equation, (see equation (6)),

\[ W^{(k)}_{\mathcal{M}} \left(x_1^{(k)}\right) = \mathcal{M} H^{(k)}(x_1^{(k)}) \left(A^{(k)}\right)^{-1} \]  

(7)

Once the state \( y \) has been computed, we can obtain the control \( u \), through one of the following two algorithms:

1. **Local asymmetric method (LAM).** Solve the problem for \( u \) by means of,

\[ \beta \mathcal{E}^* u = \hat{y} - y \quad \text{in} \, \Omega \]
\[ u = 0 \quad \text{on} \, \partial \Omega \]  

(8)

using the computed values of \( y \).
2. **Differential quadrature (DQ).** Where we evaluate

\[ u = \mathcal{E}y \]

by discretizing the operator using the differential quadrature technique.

We shall denote the first scheme by LAM-LAM and by LAM-DQ to the second one. We omit the description of the LAM-LAM algorithm, as the second part, the system (8), has been essential already described. We thus briefly recall the differential quadrature method for this problem. The main point of the RBF differential quadrature method, see Shu [9], is to build a discrete operator \( \tilde{\mathcal{E}} \) which approximates the continuous linear differential operator \( \mathcal{E} \). Its construction can be summarized as follows. First, we solve the following system

\[
\mathcal{E}\Phi(x) \bigg|_{x=x_k} = \sum_{j=1}^{n_k} w_{k,j}^{\mathcal{E}} \Phi(x_{k,j}), \quad k = 1, 2, \ldots, N
\]

(9)

where the nodes \( \{x_{k,j}\}_{j=1}^{n_k} \subset \Omega \) are the \( n_k \) nearest points to \( x_k \in X \subset \Omega \). For simplicity, we have taken \( \Phi \) to be a strictly positive definite radial basis function, (the formulation also holds for conditional positive radial basis functions). It is well known, see [11], that the system (9) is invertible. Once the coefficients \( w_{k,j}^{\mathcal{E}} \) are computed, the approximated discretization of the operator \( \mathcal{E} \) of a smooth enough function \( u \) is given by

\[
\mathcal{E}u(x) \bigg|_{x=x_k} \approx \tilde{\mathcal{E}}u(x_k) = \sum_{j=1}^{n_k} w_{k,j}^{\mathcal{E}} u(x_{k,j}), \quad k = 1, 2, \ldots, N.
\]

Note that unlike LAM approach, see equation (7), the differential quadrature technique does not include the boundary operator \( B \) in the computation of the weights, equation (9), see [9].

4 **Numerical examples**

In this section, we will discuss different examples to illustrate our main contribution. Specifically, that the proposed local algorithms can attain errors which are comparable to the global asymmetric colocation technique but a much lower computational cost. The analysis of the numerical experiments for these techniques is not trivial due to the existence of three parameters that simultaneously controls the quality of the results. These parameters are the fill distance, the penalty constant, \( \beta \), and the shape parameter, \( c \).

The experiments were set up in the following way: given a total number of nodes \( n \), we vary the values of \( \beta \) and/or \( c \), showing that we obtain completely different results. In fact, although the error can be good the condition number can be close to the machine precision, which means that we have problems that are numerically ill-posed and the result may not be reliable. On the other hand, we can have a good condition number, which means that the scheme is stable, but the error can be very poor. The goal then is to find the appropriate parameters that guarantee both stability and good numerical errors. To do this, we look for values of \( \beta \) and \( c \) for which the error is minimal and the condition number of the Gram matrix which is within the used precision. The reason for this criterion is that the condition number tells us, approximately, how many digits of the error are reliable. In other words, for a condition number of \( 10^k \), up to \( k \) digits of accuracy may be lost within the floating-point arithmetic used. We remark that the computed condition number is only an approximation that serves as a bound
for the exact value of the maximum inaccuracy that may occur in the algorithms. In the case of the LAM-LAM and LAM-DQ methods, the restriction of the condition number is imposed on the local Gram matrices.

The numerical results obtained by each method are compared with respect to the value of $\beta$, although the values of $c$ do not necessarily have to coincide for both techniques since one method is global and the other local. The results are presented using multiquadric RBFs and quadruple precision to further investigate the performance of the methods as well as the effect of the condition number. Finally, independently of the values for $\beta$ and $c$, there are problems that requires a greater number of local nodes to obtain good numerical errors.

4.1 Problem 1

The first test problem that we would like to analyze is Poisson control given by:

$$\Delta y = u, \quad -\beta \Delta u = \hat{y} - y \text{ in } \Omega$$
$$y = g, \quad u = 0 \text{ on } \partial\Omega$$

$$\hat{y} = \sin \pi x_1 \sin \pi x_2$$
$$g = 0$$

with exact solution given by

$$y_\beta(x_1, x_2) = \frac{1}{1 + 4\beta(\pi)^4} \sin \pi x_1 \sin \pi x_2$$
$$u_\beta(x_1, x_2) = \frac{2\pi^2}{1 + 4\beta(\pi)^4} \sin \pi x_1 \sin \pi x_2.$$

Since we want to restrict the values of the condition numbers $\kappa$, corresponding to the local scheme, we shall use the value of $\kappa = \max_k \kappa(A^{(k)})$ to measure the numerical ill-posedness, meanwhile for the global method we use $k = \kappa(G)$.

Table 1 contains the values $\|y - \hat{y}\|_{L^2(\Omega)}$ for th state $y$ and $\|u\|_{L^2(\Omega)}$ for control $u$; the relative error, $RE_y = \|y - y_\beta\|_{L^2(\Omega)}/\|y_\beta\|_{L^2(\Omega)}$, $RE_u = \|u - u_\beta\|_{L^2(\Omega)}/\|u_\beta\|_{L^2(\Omega)}$ respectively and the Cost

$$= (\|y - \hat{y}\|_{L^2(\Omega)} + \beta \|u\|_{L^2(\Omega)})/2, \quad \text{where } \|f\|_{L^2(\Omega)}^2 = \sum_{k=1}^n |f(x_k)|^2.$$

From table 1 we can observe that for small values of $\beta$ the errors obtained by global collocation and LAM-DQ techniques are comparable. Moreover, for large values of $\beta$, it is possible to change the number of nodes in the local systems to improve the LAM-DQ error. It is important to note that for small values of $\beta$ it is possible to obtain similar errors for both methods, but using a relatively small number of nodes for the local systems, which has a considerable impact on the computing time. Even when more nodes are used in local systems for large values of $\beta$, the computing time is still lower than the one used for AC. In addition, as $\beta \to 0$, we have $\kappa(S) \to 1$, suggesting that the method is highly stable for these cases.

Figure 1 shows in detail the effect of the variation of $\beta$ and $c$ on the error and the condition number. We can see that as $\beta$ tends to zero and the value of $c$ increases the error decreases, so for both methods the results can be improved with respect to the error but they may be unreliable because of the conditioning when $c$ is increased. The results that are reported in the table 1 are far from the values of $\kappa$ for which the solutions are affected by rounding errors with respect to the precision used, still we obtain errors below $10^{-5}$. It is important to mention that as in our case, in [7] the authors observe that
The process of finding the optimal value of $\delta$ offers a much lower computation time. There is clearly more room to improve this part, especially in the same magnitude as for the global method, but with a lower condition number and still preserving values up to $\beta = 10^{-6}$, such so that $\hat{c} \approx 10$ orders of magnitude for $c = 9$ where the condition number is around $10^{35}$ and in the case of AC up to 16 orders of magnitude for $c = 8$ where the condition number reaches $10^{37}$, while the condition number for LAM-DQ Precond is around $10^{31}$ for both values of $c$.

Table 1 compares the performance of LAM-DQ Precond with the best values reported in table 1 for $\beta = 10^{-6}$. In particular, for $c = 5$ and $\hat{c} = 5.001$, values obtained for $RE_y$ and $RE_u$ are almost of the same magnitude as for the global method, but with a lower condition number and still preserving a much lower computation time. There is clearly more room to improve this part, especially in the process of finding the optimal value of $\delta$ and thus looking for more efficient preconditioners.

| $\beta$ | $10^{-4}$ | $10^{-6}$ | $10^{-10}$ |
|---------|-----------|-----------|------------|
| $c$     | 6.00e-01 | 1.00e00  | 1.00e00   |
| $RE_y$  | 4.30e-06 | 2.94e-07 | 2.59e-11 |
| $RE_u$  | 3.04e-04 | 8.65e-05 | 8.31e-07 |
| $\gamma$| 3.32e-01 | 3.45e-03 | 3.45e-07 |
| $\mu$   | 1.68e02  | 1.75e02  | 1.75e02  |
| Cost    | 1.47e00  | 1.53e-02 | 1.53e-06 |
| $\kappa(S)$ | 4.87e26 | 1.51e24  | 1.42e24  |
| Time    | 00m46s  | 00m09s   | 00m09s   |

Table 1: Results from problem 1. For LAM-DQ $n^{(k)} = 50$, except for * where $n^{(k)} = 100$. In both cases $n = 622$. 

as the value of $\beta$ decreases so does $\|y - \hat{y}\|_{L_2(\Omega)}$. In their case, it was only possible to explore this for values up to $\beta = 10^{-6}$, due to the limitation of their iterative methods designed for the finite element method.

We also analyzed the use of a preconditioner for LAM-DQ. Figure 2 shows a comparison of the methods for $\beta = 10^{-6}$. The point we want to emphasize here is that it is possible to reduce the conditioning of the local matrices $A^{(k)}$ in such a way that the results obtained for large values of $c$ are reliable. In this particular example, when using the preconditioner $P^{(k)} A^{(k)}$, with $P^{(k)} = (A^{(k)}_x)^{-1}$, where $A^{(k)}_x$ is obtained in the same way as $A^{(k)}$ just by using the shape parameter $\hat{c} \neq c$, where $\hat{c} = c + \delta$ with $\delta$ small. For example, for the case of the figure 2, given $c = m \times 10^a$ it was taken $\delta = 0.001 \times 10^a$, such so that $\hat{c} = (m + 0.001) \times 10^a$, obtaining an error of the same size as in the case of LAM-DQ, but with a lower condition number, even reaching a difference up to 14 orders of magnitude for $c = 9$ where the condition number is around $10^{35}$ and in the case of AC up to 16 orders of magnitude for $c = 8$ where the condition number reaches $10^{37}$, while the condition number for LAM-DQ Precond is around $10^{31}$ for both values of $c$.

Table 2 compares the performance of LAM-DQ Precond with the best values reported in table 1 for $\beta = 10^{-6}$. In particular, for $c = 5$ and $\hat{c} = 5.001$, values obtained for $RE_y$ and $RE_u$ are almost of the same magnitude as for the global method, but with a lower condition number and still preserving a much lower computation time. There is clearly more room to improve this part, especially in the process of finding the optimal value of $\delta$ and thus looking for more efficient preconditioners.
Figure 1: Comparison between the values of the relative error ($RE_y$) and the condition number ($\kappa$), by varying the shape parameter $c$. These calculations were obtained using quadruple precision, and different values of the penalty constant $\beta$. 

LAM-DQ, $n^{(k)} = 50$

AC

(a) $RE_y$

(b) $RE_y$

(c) $\kappa = \max_k \kappa(A^{(k)})$

(d) $\kappa = \kappa(G)$. 

9
4.2 Problem 2

The next test problem is a convection-diffusion control problem for which there is no exact solution, given by

\[-\epsilon \Delta + \omega \cdot \nabla) y = u, \quad \beta(-\epsilon \Delta - \omega \cdot \nabla) u = y - \hat{y} \quad \text{in} \ \Omega
\]

\[y = g, \quad u = 0 \quad \text{on} \ \partial \Omega\]

\[\hat{y} = 0\]

\[g = \begin{cases} 1 & \text{in} \ \{0\} \times \left[\frac{1}{2}, 1\right] \cup [0, 1] \times \{1\} \\ 0 & \text{elsewhere} \end{cases}\]

\[\omega = (\cos \theta, \sin \theta), \quad \text{with} \ \theta = \frac{\pi}{6}\]

\[\epsilon = \frac{1}{200}\]
This example corresponds to a boundary layer problem, which is of interest due to the sharp gradient attained at the boundary layer. Table 3 contains the values $\|y - \hat{y}\|_{L^2(\Omega)}$ for the state $y$ and $\|u\|_{L^2(\Omega)}$ for the control $u$.

From the results reported in Table 3 it can be seen that for any method it is possible to find $c$ in such a way that the minimum value of $\|y - \hat{y}\|_{L^2(\Omega)}$ is of the same order in magnitude. The difference is in $\|u\|_{L^2(\Omega)}$ since the norm obtained for the local scheme case is much smaller than for AC, which seems to affect in the same way the value of the cost functional.

Figure 3 only shows the solution for LAM-DQ using $n = 50000$, since AC takes too much time, in fact, although we did not complete the experiment, we estimate that it will take around two days to obtain the results. The high number of total nodes were used to show the capabilities of LAM-DQ to handle big problems and to show in detail the solution obtained for the state for this problem. We can see how the solution is very close to 0 in all the domain except very near of the boundary layer.
4.3 Problem 3

The last test problem is also a convection-diffusion control problem for which there is no exact solution, given by

\[
(-\epsilon \Delta + \omega \cdot \nabla)y = u, \quad \beta(-\epsilon \Delta - \omega \cdot \nabla)u = y - \hat{y} \quad \text{in } \Omega
\]

\[
y = g, \quad u = 0 \quad \text{on } \partial\Omega
\]

\[
\hat{y} = \begin{cases} 
(2x_1 - 1)^2(2x_2 - 1)^2 & \text{in } [0, \frac{1}{2}]^2 \cap \Omega \\
0 & \text{elsewhere}
\end{cases}
\]

\[
g = \begin{cases} 
(2x_1 - 1)^2(2x_2 - 1)^2 & \text{in } [0, \frac{1}{2}]^2 \cap \partial\Omega \\
0 & \text{elsewhere}
\end{cases}
\]

\[
\omega = (\cos \theta, \sin \theta), \quad \text{with } \theta = 2.4
\]

\[
\epsilon = \frac{1}{200}
\]

Table 4 contains the same values as the previous example: \(\|y - \hat{y}\|_{L_2(\Omega)}\) for the state \(y\) and \(\|u\|_{L_2(\Omega)}\) for control \(u\).

| \(\beta\) | LAM-DQ | AC |
| -- | ---- | ---- |
| \(10^{-2}\) | 4.00E-04 | 4.00E-05 |
| \(10^{-6}\) | 7.00E-03 | 6.00E-04 |
| \(10^{-10}\) | 7.00E-03 | 6.00E-04 |

Table 4: Results from problem 3. For LAM-DQ \(n^{(k)} = 50\). In both cases \(n = 622\)

The results reported in the table show again, that as in the previous example, for any method it is possible to find \(c\) in such a way that the minimum value of \(\|y - \hat{y}\|_{L_2(\Omega)}\) is of the same order in magnitude. Here, for the number of total nodes considered, there is no difference in the magnitude of \(\|u\|_{L_2(\Omega)}\), and therefore also for the value of the cost functional.

However, for the particular case for \(\beta = 10^{-10}\) shown in figure with \(n = 3021\), we have values for \(y\) of the same magnitude but the control norm is lower for LAM-DQ. In addition, the control calculated through LAM-DQ visually resembles the results calculated by finite element method in [6] and [7], which shows the consistency of the LAM-DQ solutions with respect to the finite element method.

Finally, we compare the computing time for both methods. The tests were carried out using our own routines programmed in C++ on a machine with an Intel Core i5 M540 processor (2.53GHz). The execution time of the algorithms seems only to be dependent on the total number of nodes, that is, no matter which value of \(c\) and \(\beta\) are taken or if it is a convection-diffusion or Poisson control. Table 5 shows the different calculation times by varying the total number of nodes, showing that for all cases LAM-DQ has a shorter execution time in all cases. Figure 5 shows in a more clear way the
Figure 4: Solution for problem 3, \( n = 3021 \), \( \beta = 10^{-10} \)

5 Conclusions

In this article, we solve control distributed problems for convection-diffusion linear PDEs problems by global and local radial basis functions methods. Inspired by the local Hermite interpolation method proposed by [10], we introduced two new techniques, LAM-DQ and LAM-LAM.

A saddle point problem is obtained if we discretize the primal and adjoint equations by using radial basis function methods. We proposed a solution to this problem by discretizing instead, a well-posed biharmonic problem for the state variable and then obtaining the control by a second decoupled equation.

An important contribution of this paper is that these local methods, in comparison to global colocation techniques, can attain similar precision errors for the same number of nodes, but with a considerable reduction of the computing, CPU, time.

While the condition number of the sparse global matrices in all our experiments, remains within an acceptable value, below the machine precision, the maximum condition number of the local matrices...
| No. Nodos | AC   | LAM-DQ | LAM-DQ Precond |
|----------|------|--------|----------------|
| 500      | 00m58s | 00m07s | 00m15s |
| 1000     | 07m34s | 00m18s | 00m34s |
| 1500     | 24m59s | 00m36s | 01m01s |
| 2000     | 59m09s | 01m05s | 01m58s |
| 2500     | 11m44s | 01m49s | 02m29s |
| 3000     | 197m50s | 02m51s | 03m40s |
| 3500     | 316m33s | 04m15s | 05m19s |
| 4000     | 467m52s | 06m06s | 07m12s |

(a) Table

Figure 5: Calculation time employed by the methods. For LAM-DQ, $n^{(k)} = 50$.

can grow up to the point where they are numerically singular as the fill distance tends to zero.

In this article, we deal with this problem by using quad precision and by proposing a simple but effective preconditioner. By doing this, we manage to solve problems having 50000 nodes and reduce the condition number of the local matrices up to 10 orders of magnitude. The ill-conditioning of the Gram local and global matrices is currently an active research area in the field of radial basis function theory.

Although many proposals have appeared in the literature to deal with this general problem, we believe that the methods and the analysis proposed in this article present a significant contribution which shows the way to solve large distributed control problems.

Acknowledgements

Funding: This work was supported by the National Autonomous University of México [grant: PAPIIT, IN102116]
References

[1] P.P. Chinchapatnam, K. Djidjeli, and P.B. Nair. Domain decomposition for time-dependent problems using radial based meshless methods. *Numerical Methods for Partial Differential Equations*, 23(1):38–59, 2007.

[2] P. González-Casanova, J.A. Mu noz Gómez, and G. Rodríguez-Gómez. Node adaptive domain decomposition method by radial basis functions. *Numerical Methods for Partial Differential Equations*, 25(6):1482–1501, 2009.

[3] E.J. Kansa and P. Holoborodko. On the ill-conditioned nature of $c^\infty$ rbf strong collocation. *Engineering Analysis with Boundary Elements*, 78:26–30, 2017.

[4] J.L. Lions. *Optimal control of systems*. Springer, New York, 1968.

[5] J.W. Pearson. A radial basis function method for solving pde-constrained optimization problems. *Numerical Algorithms*, 64(3):481–506, 2013.

[6] T. Rees. *Preconditioning iterative methods for PDE constrained optimization*. PhD thesis, University of Oxford, 2010.

[7] T. Rees, H.S. Dollar, and A.J. Wathen. Optimal solvers for pde-constrained optimization. *SIAM Journal on Scientific Computing*, 32(1):271–298, 2010.

[8] S.A. Sarra. Radial basis function approximation methods with extended precision floating point arithmetic. *Engineering Analysis with Boundary Elements*, 35(1):68–76, 2011.

[9] C. Shu, H. Ding, and K.S. Yeo. Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 192(7):941–954, 2003.

[10] D. Stevens, H. Power, M. Lees, and H. Morvan. A local-hermitian rbf meshless numerical method for the solution of multi-zone-problems. *Numerical Methods for Partial Differential Equations*, 27(5):1201–1230, 2010.

[11] H. Wendland. *Scattered data approximation*. Cambridge University Press, 2004.

[12] X. Zhou, Y.C. Hon, and J. Li. Overlapping domain decomposition method by radial basis functions. *Applied Numerical Mathematics*, 44(1):241–255, 2003.

[13] Z.J. Zhou and N.N. Yan. A survey of numerical methods for convection-diffusion optimal control problems. *Journal of Numerical Mathematics*, 22(1):61–85, 2014.