Radial basis function methods for optimal control of the convection–diffusion equation: A numerical study

Pedro González Casanova, Christian Gout, Jorge Zavaleta

Radial basis function methods for optimal control of the convection-diffusion equation: A numerical study

Pedro González Casanova[a], Christian Gout[b,c,*], Jorge Zavaleta[a]

[a] *Instituto de Matemáticas, UNAM, Cuidad Universitaria, Mexico D.F., CP. 04510, MEXICO.*
[b] *Normandie Université, INSA Rouen, LMI, Lab. of Mathematics of INSA Rouen, 76000 Rouen, France*
[c] *Magique 3D - Advanced 3D Numerical Modeling in Geophysics, INRIA Bordeaux Sud Ouest, France*

Abstract

In this paper, we perform a numerical study for the solution of optimal constrained optimization problems for linear convection-diffusion PDEs by local and global radial basis function techniques. To the best of our knowledge, these control problems have not been treated in the literature by RBFs methods. It is well-known that the algebraic system of RBFs methods presents a larger condition number and a higher numerical complexity as the number of nodes (or shape parameter), increases. In this work, and in the context of optimal constrained optimization problems, we explore a possible answer to both problems. Specifically, we introduce a local RBF method (denoted as LAM-DQ), based on the combination of an asymmetric RBFs local method (LAM), inspired in local Hermite interpolation (LHI), combined with the differential quadrature method (DQ). We also propose a preconditioning technique that in combination with extended arithmetic precision let us treat the ill-conditioning problem. We numerically prove that as the number of nodes increases, then for errors of the same order, the condition number remains tractable, in quad-precision, and the numerical complexity of the local method remains bounded.

Keywords: Local radial basis functions methods, PDE-constrained optimization problems, convection-diffusion control.
2010 MSC: 65N35, 49J20, 65K10.

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 (see, for instance, Zhou and Yan [1]). These works have been formulated within two general frames: the discretized-optimized or the optimize-discretized approaches. In particular, Galerkin methods have been proposed and ana-

*Corresponding author
Email address: christian.gout@insa-rouen.fr (Christian Gout)

lyzed within both frames, (see [1] and references therein). On the other hand, Pearson [2], which up to now seems to be the only reference on this subject, solves Poisson constrained optimization problems by using global RBFs symmetric and asymmetric collocation techniques.

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, see [3]. In the case of infinitely differentiable RBFs, convergence can be exponential, but the corresponding condition number also increases in an exponential way [4].

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

1. Perform a comparative numerical study of global and local RBFs meshfree methods for the solution of convection-diffusion constrained optimization problems.
2. Formulate a local asymmetric method (LAM) inspired in the local Hermite interpolation technique, (LHI), [5].
3. Show that the proposed local method can attain the same accuracy of global techniques but with the advantage of a considerable reduction of the computing time, and making it possible to extend its application to a large number of nodes.
4. Introduce a simple but effective preconditioner to improve the condition number of the local matrices of the LAM method.

We note that in point (3) above, although the number of centers can be very large, the value of the fill distance should be such that the condition number of the local matrices, in extended precision, are numerically well posed.

We find that the discretization of the primal and dual Euler Lagrange equations by the local Hermite interpolation method (LHI) gives rise to a saddle point problem, which is well known to be difficult to solve due to their indefiniteness and often poor spectral properties, see [6]. To avoid this problem, we use both the primal and dual equations to build a Biharmonic system for the state variable. This system is discretized by a local asymmetric interpolation method (LAM), introduced in this work, which let us compute the state variable. The resulting state variable is used to compute the control by the differential quadrature (DQ) method. The couple method is denoted as LAM-DQ. An alternative technique to this method, which we denote as LAM-LAM, is to discretize the second equation by using LAM technique again instead of DQ. This method, however, is more expensive than LAM-LAM and produces almost the same quality results. Note that, unlike LHI the LAM technique do not incorporates the convection diffusion operator, or in general the linear parabolic or elliptic operator, in the ansatz although the boundary operators are included. The resulting matrix is asymmetric but it is able to handle in an easer way the boundary conditions than the purely asymmetric scheme and the resulting global matrix is coupled using the same procedure as LHI. Moreover the numerical complexity of these local method is, consequently, lower than LHI techniques.

It is worth noting that according to Bayona et al. [7], the DQ method introduced by Shu [8], gave the foundations for the radial basis function-generated finite difference (RBF-DF) approximations. We also recall that in a different context, namely

for interpolation of vector fields, both global techniques and LHI methods have been investigated (see Cervantes et al. [9], [10]).

In this work, we are interested in comparing the results obtained from local methods with those obtained by solving the problem with global asymmetric collocation (AC), techniques by using direct solvers with quad precision. The extended precession approach is a current alternative to the bad conditioning problem that has been used and supported as a reliable alternative by Kansa [11] and Sarra [4], among others. We note that, although here we aim to investigate the control problems using the extended precision approach, several alternatives to the ill-conditioned problem of RBFs collocation methods have been recently formulated, see for example [12], [13]; Fornberg and Flyer [14] and references therein for a comprehensive review on this subject.

This paper is organized as follows. In section 2, we briefly state the continuous control problem and refer the reader to the proper references. In section 3, we formulated the LAM-DQ and LAM-LAM local methods for solving of the convection-diffusion constrained optimization problems. Section 4, presents numerical examples to show the capabilities and performance of the LAM-DQ local method as well as the preconditioning technique. 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

$$\begin{gathered} \min_{y,u} \tfrac{1}{2}\|y-\hat{y}\|^2_{L^2(\Omega)} + \tfrac{\beta}{2}\|u\|^2_{L^2(\Omega)} \\ \text{s.t. } \mathcal{E}y = u \text{ in } \Omega, \quad \mathcal{B}y = g \text{ on } \partial\Omega \end{gathered} \tag{1}$$

where, y is the state, u the control, $\hat{y}$ the objective state, $\beta > 0$ a penalty constant, $\mathcal{E}$ is a PDE stationary linear operator with variable coefficients and $\mathcal{B}$ a Dirichlet, Neumann or Robin, boundary operator. These problems were introduced and analyzed by L. J. Lions in [15].

The distributed control problem (1) can be equivalently formulated using a functional that incorporates the PDE constraints by means of Lagrange multipliers, (see [16] and [17]), namely as

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

where p_1 and p_2 are the Lagrange multipliers. Taking the Frechet derivative of functional (2) with respect to y, u and p_1 and p_2 it is possible to obtain the following Euler-Lagrange equations in terms of the state y and the control variable u,

$$\begin{array}{ll|ll} \mathcal{E}y = u & \text{in } \Omega & \beta\mathcal{E}^*u = \hat{y}-y & \text{in } \Omega \\ \mathcal{B}y = g & \text{on } \partial\Omega & u = 0 & \text{on } \partial\Omega \end{array} \tag{3}$$

where it is possible to show that $p_1 = p_2 = p$ and that p can be eliminated by using the equation $p = \beta u$. The variables y and u satisfying (3) are 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

$$y(x) = \mathrm{H}(x)\lambda, \quad u(x) = \mathrm{H}(x)\mu,$$

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

$$\mathrm{H}(x) = \left[\begin{array}{c|c} \Phi\left(x - \mathrm{x}_i\right) & p_\ell(x) \\ {\scriptstyle 1\le i\le n} & {\scriptstyle 1\le \ell\le n_p} \end{array} \right] \in \mathbb{R}^{n+n_p},$$

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

$$\begin{bmatrix} G^{\mathcal{B}} & \beta E_* \\ -E & G \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} \mathrm{d} \\ 0 \end{bmatrix},$$

where

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

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

$$\mathrm{d} = \left[\begin{array}{c|c|c} g(\mathrm{x}_j) & \hat{y}(\mathrm{x}_k) & 0 \\ {\scriptstyle 1\le j\le n_b} & {\scriptstyle n_b+1\le k\le n} & {\scriptstyle 1\le \ell\le n_p} \end{array} \right]^t.$$

If $G^{\mathcal{B}} = G$, *i.e.* taking $\mathcal{B} = I$, 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 $R = G + \beta EG^{-1}E_*$, 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

$$\begin{array}{rl} \mathcal{M}y = \hat{y} & \text{in } \Omega \\ \mathcal{E}y = 0 & \text{on } \partial\Omega \\ \mathcal{B}y = g & \text{on } \partial\Omega \end{array} \tag{4}$$

where the differential operator $\mathcal{M}$ is given by $\mathcal{M} = \mathrm{I} + \beta\mathcal{E}^*\mathcal{E}$. 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 notation: Let $\mathrm{X} \subset \bar{\Omega}$ 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 $\mathrm{D}_k \cap \mathrm{X}$ so that:

- There are $n^{(k)}$ nodes in D_k *i.e.* $n^{(k)} = \#(\mathrm{X} \cap \mathrm{D}_k)$.
- The first node, $\mathrm{x}_1^{(k)}$ is the center of D_k.
- The first $n_c^{(k)}$ nodes are centers of other discs, *i.e.* $n_c^{(k)} = \#(\mathrm{X}_c \cap \mathrm{D}_k)$.
- The following $n_b^{(k)}$ nodes lie on $\partial\Omega$, $n_b^{(k)} = \#(\partial\Omega \cap (\mathrm{D}_k \backslash \mathrm{X}_c))$.
- The remaining $n_\iota^{(k)}$ nodes belong to the interior of Ω (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 $\mathrm{X}_c \subset \Omega$.

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

$$\mathrm{H}^{(k)}(x) = \left[\begin{array}{c|c} \Phi\left(x - \mathrm{x}_j^{(k)}\right)_{1 \le j \le n^{(k)}} & p_\ell(x)_{1 \le \ell \le n_p} \end{array} \right] \in \mathbb{R}^{n^{(k)} + n_p}.$$

Defining the following ansatz

$$y^{(k)}(x) = \mathrm{H}^{(k)}(x)\lambda^{(k)},$$

we obtain the local linear system

$$A^{(k)}\lambda^{(k)} = \begin{bmatrix} \Phi & P \\ \mathcal{B}\Phi & \mathcal{B}P \\ \mathcal{E}\Phi & \mathcal{E}P \\ \mathcal{M}\Phi & \mathcal{M}P \\ P^t & 0 \end{bmatrix} \lambda^{(k)} = \mathrm{d}^{(k)} \tag{5}$$

with the data vector

$$\mathrm{d}^{(k)} = \left[\begin{array}{c|c|c|c|c} y\left(\mathrm{x}_j^{(k)}\right) & g\left(\mathrm{x}_j^{(k)}\right) & 0 & \hat{y}\left(\mathrm{x}_j^{(k)}\right) & 0 \\ {\scriptstyle 1\le j\le n_c^{(k)}} & {\scriptstyle n_c^{(k)}<j\le n_c^{(k)}+n_{b1}^{(k)}} & {\scriptstyle n_c^{(k)}+n_{b1}^{(k)}<j\le n_c^{(k)}+n_b^{(k)}} & {\scriptstyle n_c^{(k)}+n_b^{(k)}<j\le n^{(k)}} & {\scriptstyle 1\le \ell\le n_p} \end{array} \right]^t$$

where $n_{b1}^{(k)}$ and $n_{b2}^{(k)}$ are the number of boundary points for each of the boundary conditions, so that $n_b^{(k)} = n_{b1}^{(k)} + n_{b2}^{(k)}$. Solving for $\lambda^{(k)}$ we obtain the local solution

$$y^{(k)}(x) = \mathrm{H}^{(k)}(x)\left(A^{(k)}\right)^{-1}\mathrm{d}^{(k)} = W^k(x)\mathrm{d}^{(k)}, \tag{6}$$

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

Denote by $y_c = \left[y\left(\mathrm{x}_1^{(k)}\right)\right]_{k=1}^{n_c} \in \mathbb{R}^{n_c}$ the vector of the values of y at each of the centers. Then for each k, the unknown elements of $\mathrm{d}^{(k)}$ belong to y_c.

Consider now the following system of equations

$$\hat{y}\left(\mathrm{x}_1^{(k)}\right) = \mathcal{M}y\left(\mathrm{x}_1^{(k)}\right) = \mathcal{M}\mathrm{H}^{(k)}(\mathrm{x}_1^{(k)})\left(A^{(k)}\right)^{-1}\mathrm{d}^{(k)} = \mathrm{W}_{\mathcal{M}}^{(k)}\left(\mathrm{x}_1^{(k)}\right)\mathrm{d}^{(k)} \tag{7}$$

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

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 $Sy_c = b$. Note that since in each $\mathrm{d}^{(k)}$ 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 (7)),

$$\mathrm{W}_{\mathcal{M}}^{(k)}\left(\mathrm{x}_1^{(k)}\right) = \mathcal{M}\mathrm{H}^{(k)}(\mathrm{x}_1^{(k)})\left(A^{(k)}\right)^{-1} \tag{8}$$

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,

$$\begin{array}{rl} \beta\mathcal{E}^*u = \hat{y} - y & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega \end{array} \tag{9}$$

   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, since the second part of the algorithm, system (9), has been essential already described. Also, we omit numerical results for LAM-LAM since LAM-DQ presents the best performance between the two algorithms. We thus briefly recall the differential quadrature method for this problem.

The main point of the RBF differential quadrature method, see Shu [8], is to build a discrete operator $\tilde{\mathbf{E}}$ which approximates the continuous linear differential operator $\mathcal{E}$. Its construction can be summarized as follows. First, we solve the following system

$$\mathcal{E}\Phi(x)\Big|_{x=\mathrm{x}_k} = \sum_{j=1}^{n_k} w_{k,j}^{\mathcal{E}} \Phi(\mathrm{x}_{k,j}), \quad k = 1, 2, \ldots, N \tag{10}$$

where the nodes $\{\mathrm{x}_{k,j}\}_{j=1}^{n_k} \subset \Omega$ are the n_k nearest points to $\mathrm{x}_k \in \mathrm{X} \subset \Omega$. For simplicity, we have taken Φ to be a strictly positive definite radial basis function, (the formulation also holds for conditional positive radial basis functions). It is well known, see [18], that the system (10) 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

$$\mathcal{E}u(x)\Big|_{x=\mathrm{x}_k} \approx \tilde{\mathbf{E}}u(\mathrm{x}_k) = \sum_{j=1}^{n_k} w_{k,j}^{\mathcal{E}} u(\mathrm{x}_{k,j}), \quad k = 1, 2, \ldots, N.$$

Note that unlike LAM approach, see equation (8), the differential quadrature technique does not include the boundary operator $\mathcal{B}$ in the computation of the weights, equation (10), see [8].

4. Numerical examples

In this section, we will discuss different examples to illustrate our main contribution. Specifically, that the proposed local algorithm can attain errors which are comparable to the global asymmetric collocation 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, β, 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 β 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 β and c for which the error is minimal and the condition number of the Gram matrix lower than 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-DQ method, the restriction of the condition number is imposed on the local Gram matrices.

The numerical results obtained by each method are computed for different values of β, note that 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 β and c, there are problems that require a greater number of local nodes to obtain good numerical errors.

4.1. Problem 1

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

$$\begin{array}{lll} -\triangle y = u, & -\beta \triangle u = \hat{y} - y & \text{in } \Omega \\ y = g, & u = 0 & \text{on } \partial\Omega \end{array}$$

$$\begin{array}{c} \hat{y} = \sin \pi x_1 \sin \pi x_2 \\ g = 0 \end{array}$$

with exact solution given by

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

Since we want to restrict the values of the condition numbers κ, 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 $\kappa = \kappa(G)$.

Table 1 contains the values $\|y - \hat{y}\|_{L_2(\Omega)}$ for the state y; $\|u\|_{L_2(\Omega)}$ for control u and the relative errors, $RE_y = \|y - y_\beta\|_{L_2(\Omega)} / \|y_\beta\|_{L_2(\Omega)}$, $RE_u = \|u - u_\beta\|_{L_2(\Omega)} / \|u_\beta\|_{L_2(\Omega)}$, for the state and the control respectively; the Cost $= (\|y - \hat{y}\|^2_{L_2(\Omega)} + \beta \|u\|^2_{L_2(\Omega)})/2$, where $\|f\|^2_{L_2(\Omega)} = \sum_{k=1}^{n} |f(\mathrm{x}_k)|^2$.

From table 1 we can observe that for small values of β the errors obtained by global collocation and LAM-DQ techniques are comparable. Moreover, for large values of β, 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 β and small number of nodes for the local systems it is possible to obtain similar errors for both methods, which is property that has a considerable impact on the computing time. Even when more nodes are used in local systems for large values of β, 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.

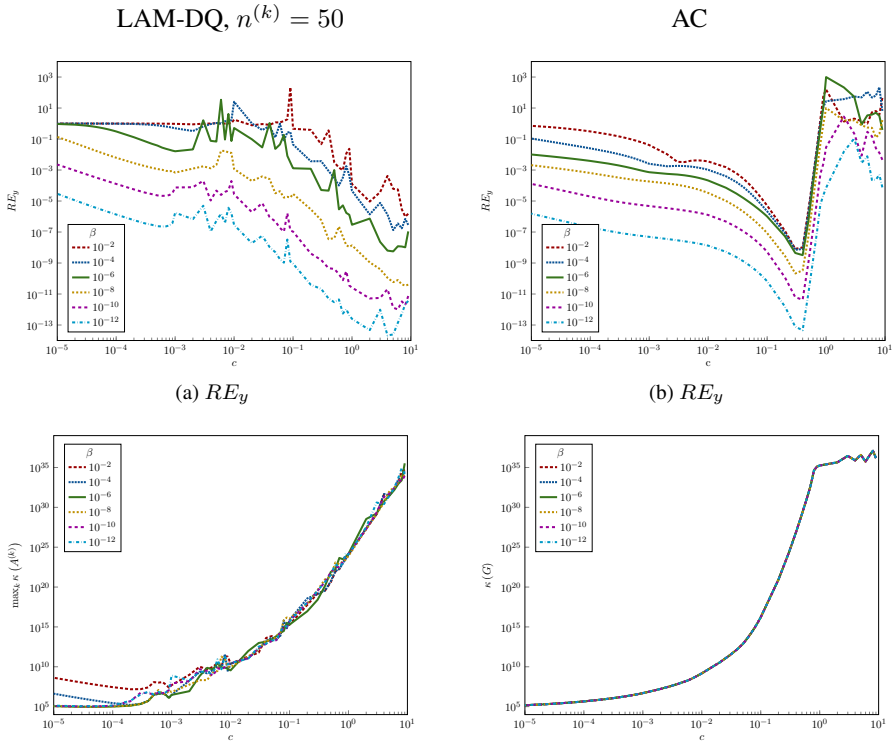
	LAM-DQ			AC		
β	10^{-4*}	10^{-6}	10^{-10}	10^{-4}	10^{-6}	10^{-10}
c	6.00×10^{-1}	1.00	1.00	3.00×10^{-1}	4.00×10^{-1}	4.00×10^{-1}
RE_y	4.30×10^{-6}	2.94×10^{-7}	2.59×10^{-11}	7.15×10^{-9}	3.23×10^{-9}	4.28×10^{-12}
RE_u	3.04×10^{-4}	8.65×10^{-5}	8.31×10^{-7}	6.33×10^{-9}	1.98×10^{-7}	6.59×10^{-8}
$\|y-\hat{y}\|$	3.32×10^{-1}	3.45×10^{-3}	3.45×10^{-7}	4.22×10^{-1}	4.39×10^{-3}	4.39×10^{-7}
$\|u\|$	1.68×10^{2}	1.75×10^{2}	1.75×10^{2}	2.14×10^{2}	2.22×10^{2}	2.23×10^{2}
Cost	1.47	1.53×10^{-2}	1.53×10^{-6}	2.38	2.47×10^{-2}	2.48×10^{-6}
κ	4.87×10^{26}	1.51×10^{24}	1.42×10^{24}	2.03×10^{24}	9.05×10^{26}	9.05×10^{26}
$\kappa(S)$	4.68×10^{7}	3.81×10^{4}	1.39			
Time	46sec	9sec	9sec	1min 51sec	1min 51sec	1min 51sec

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

Figure 1 shows in detail the variation of β and c on the error and the condition number. We can see that as β 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 κ 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 [19] the authors observe that as the value of β 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.

Table 2, shows that when the condition number of the local systems is bounded, the shape parameter decreases as $n^{(k)}$ increases. On the other hand, we performed numerical experiments, not reported here because the results are similar to those obtained in table 2, where we do not bound the size of the condition number. These computations show that the shape parameter slightly changes when the number of local nodes increases. Finally, we observe that the variation of the shape parameter is independent of the value of the penalty constant β.

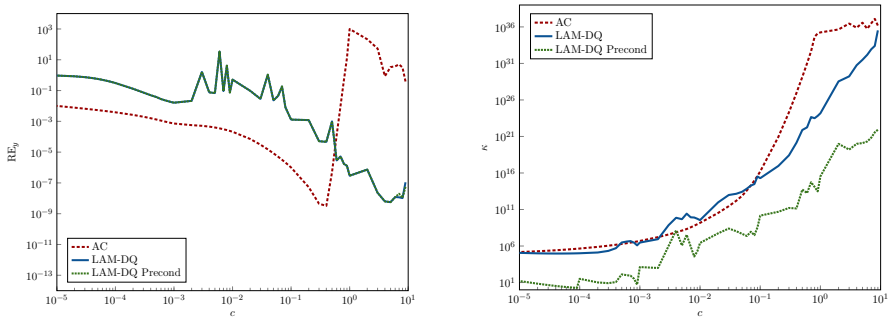
We also analyzed the use of a preconditioner for LAM-DQ, figure 2 shows a comparison of the methods for $\beta = 10^{-6}$. The point we want to emphasize here is that it is possible to reduce the conditioning of the local matrices $A^{(k)}$ in such a way that the results obtained for large values of c are reliable. In this particular example, when using the preconditioner $P^{(k)}A^{(k)}$, with $P^{(k)} = (A_*^{(k)})^{-1}$, where $A_*^{(k)}$ 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 δ small. For example, for the case of the figure 2, given $c = m \times 10^{\alpha}$ it was taken $\delta = 0.001 \times 10^{\alpha}$, such so that $\hat{c} = (m + 0.001) \times 10^{\alpha}$, 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



(a) RE_y (b) RE_y

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

Figure 1: Comparison between the values of the relative error (RE_y) and the condition number (κ), by varying the shape parameter c. These calculations were obtained using quadruple precision, and different values of the penalty constant β.



(a) Relative error (b) Condition number

Figure 2: Comparison of different methods for $\beta = 10^{-6}$

β	$n^{(k)}$	c	RE_y	$\kappa_{max}(A^{(k)})$	$\kappa(S)$
10^{-2}	25	2	7.73×10^{-2}	1.17×10^{20}	9.42×10^{8}
	50	1	1.24×10^{-4}	1.14×10^{24}	6.72×10^{7}
	75	7×10^{-1}	1.24×10^{-4}	6.92×10^{24}	1.71×10^{9}
	100	6×10^{-1}	2.68×10^{-5}	4.63×10^{26}	1.61×10^{8}
10^{-4}	25	7	4.26×10^{-4}	6.26×10^{23}	4.04×10^{8}
	50	1	4.55×10^{-5}	1.18×10^{24}	9.45×10^{6}
	75	7×10^{-1}	7.91×10^{-6}	7.17×10^{24}	1.97×10^{8}
	100	6×10^{-1}	4.30×10^{-6}	4.87×10^{26}	4.68×10^{7}
10^{-6}	25	9	1.23×10^{-5}	1.63×10^{26}	1.78×10^{6}
	50	1	2.94×10^{-7}	1.51×10^{24}	3.81×10^{4}
	75	9×10^{-1}	2.35×10^{-7}	6.66×10^{26}	1.20×10^{7}
	100	7×10^{-1}	5.77×10^{-8}	7.59×10^{26}	4.55×10^{6}
10^{-10}	25	9	5.06×10^{-10}	6.29×10^{25}	1.16
	50	1	2.59×10^{-11}	1.42×10^{24}	1.39
	75	7×10^{-1}	7.65×10^{-12}	1.07×10^{26}	2.65
	100	7×10^{-1}	5.43×10^{-12}	9.26×10^{27}	6.45

Table 2: Behavior of shape parameter against the number of local nodes ($n^{(k)}$) for different values of β. Here we used $n = 622$ and test problem 1.

condition number reaches 10^{37}, while the condition number for LAM-DQ Precond is around 10^{21} for both values of c.

Table 3 displays the performance of LAM-DQ with and without preconditioning. Computations were performed for $\beta = 10^{-6}$ and the best values reported in table 1. In particular, for $c = 5$ and $\hat{c} = 5.001$, the values of RE_y and RE_u for the local methods have nearly the same order of magnitude than the values corresponding to the global method, (AC), but with lower condition number. Moreover, the computing time for local methods is clearly much lower than the CPU time obtained for the global technique. There is clearly more room to improve this part, especially in the process of finding the optimal value of δ and thus looking for more efficient preconditioners.

4.2. Problem 2

The following problem was used by Pearson in [2] in a finite element context. It is an optimal control formulation of the double-glazing problem discussed in [20]. This is essentially a convection-diffusion control problem with variable coefficients for which there is no exact solution and is defined as follows.

	LAM-DQ	LAM-DQ Precond	AC
c	1.00	5.00	4.00×10^{-1}
RE_y	2.94×10^{-7}	5.63×10^{-9}	3.23×10^{-9}
RE_u	8.65×10^{-5}	6.09×10^{-6}	1.98×10^{-7}
$\|y-\hat{y}\|$	3.45×10^{-3}	3.45×10^{-3}	4.39×10^{-3}
$\|u\|$	1.75×10^{2}	1.75×10^{2}	2.22×10^{2}
Cost	1.53×10^{-2}	1.53×10^{-2}	2.47×10^{-2}
κ	1.51×10^{24}	1.25×10^{20}	9.05×10^{26}
$\kappa(S)$	3.81×10^{4}	1.91×10^{6}	
Time	9sec	20sec	1min 51sec

Table 3: Comparison of the methods LAM-DQ; LAM-DQ-Precond and AC for $\beta = 10^{-6}$; $n^{(k)} = 50$. Here $\kappa = \max_k \kappa(A^{(k)})$ is the condition number for local methods and $\kappa = \kappa(G)$ the condition number for the global AC technique. In all cases $n = 622$.

$$\begin{aligned} (-\epsilon \triangle + \omega \cdot \nabla) y = u, \quad \beta(-\epsilon \triangle - \omega \cdot \nabla) u = y - \hat{y} \quad & \text{in } \Omega = [-1,1]^2 \\ y = g, \qquad\qquad\qquad\qquad u = 0 \quad & \text{on } \partial\Omega \end{aligned}$$

$$\begin{aligned} \hat{y} &= 0 \\ g &= \begin{cases} 1 & \{1\} \times [-1,1] \\ 0 & \text{elsewhere} \end{cases} \\ \omega &= [2x_2(1-x_1^2), -2x_1(1-x_2^2)]^t \\ \epsilon &= \frac{1}{200} \end{aligned}$$

This example corresponds to a boundary layer problem, which is of interest due to the sharp gradient attained at the boundary layer. Table 4 contains the values $\|y - \hat{y}\|_{L_2(\Omega)}$ for the state y and $\|u\|_{L_2(\Omega)}$ for the control u.

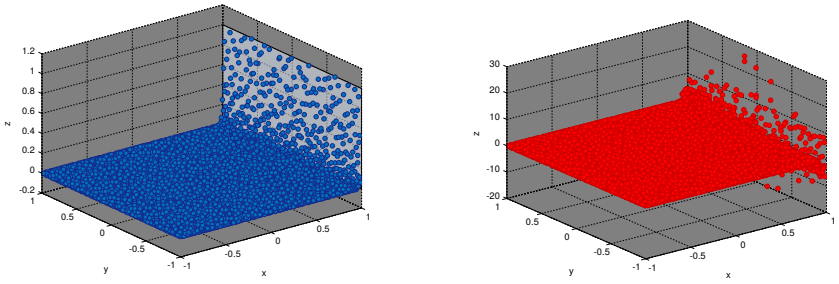
From the results reported in table 4 it can be seen that for any method it is possible to find c in such a way that the minimum value of $\|y - \hat{y}\|_{L_2(\Omega)}$ is of the same order in magnitude. The difference is in $\|u\|_{L_2(\Omega)}$ since the norm obtained for the local scheme case is much smaller than for AC, which seems to affect in the same way the value of the cost functional.

Figure 3 we only display the solution for LAM-DQ using $n = 50000$, this is due to the fact that AC takes to 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.

	LAM-DQ			AC		
β	10^{-2}	10^{-6}	10^{-10}	10^{-2}	10^{-6}	10^{-10}
c	8.00×10^{-6}	6.00×10^{-4}	6.00×10^{-4}	4.00×10^{-6}	1.00×10^{-5}	1.00×10^{-5}
$\|y-\hat{y}\|$	3.97	2.74×10^{-3}	2.74×10^{-7}	1.71	4.17×10^{-4}	4.17×10^{-8}
$\|u\|$	1.48×10^{1}	3.73×10^{-2}	3.74×10^{-6}	2.47×10^{1}	4.26×10^{1}	4.26×10^{1}
Cost	8.99	3.75×10^{-6}	3.76×10^{-14}	4.52	9.09×10^{-4}	9.09×10^{-8}
κ	4.73×10^{4}	1.04×10^{5}	1.09×10^{5}	9.53×10^{4}	1.08×10^{5}	1.08×10^{5}
$\kappa(S)$	2.58×10^{3}	1.03	1.00			
Time	10sec	10sec	10sec	1min 52sec	1min 52sec	1min 52sec

Table 4: Results from problem 2. For LAM-DQ $n^{(k)} = 50$ and $\kappa = \max_k \kappa(A^{(k)})$. For AC $\kappa = \kappa(G)$. In both cases $n = 622$.

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



(a) State, $c = 10^{-7}$

(b) Control, $c = 10^{-7}$

Figure 3: Solution for problem 2, $n = 50000$, $\beta = 10^{-6}$

4.3. Problem 3

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

$$\begin{aligned} (-\epsilon \triangle + \omega \cdot \nabla) y = u, \quad \beta(-\epsilon \triangle - \omega \cdot \nabla) u = y - \hat{y} \quad & \text{in } \Omega \\ y = g, \qquad\qquad\qquad\qquad\qquad\quad u = 0 \quad & \text{on } \partial\Omega \end{aligned}$$

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

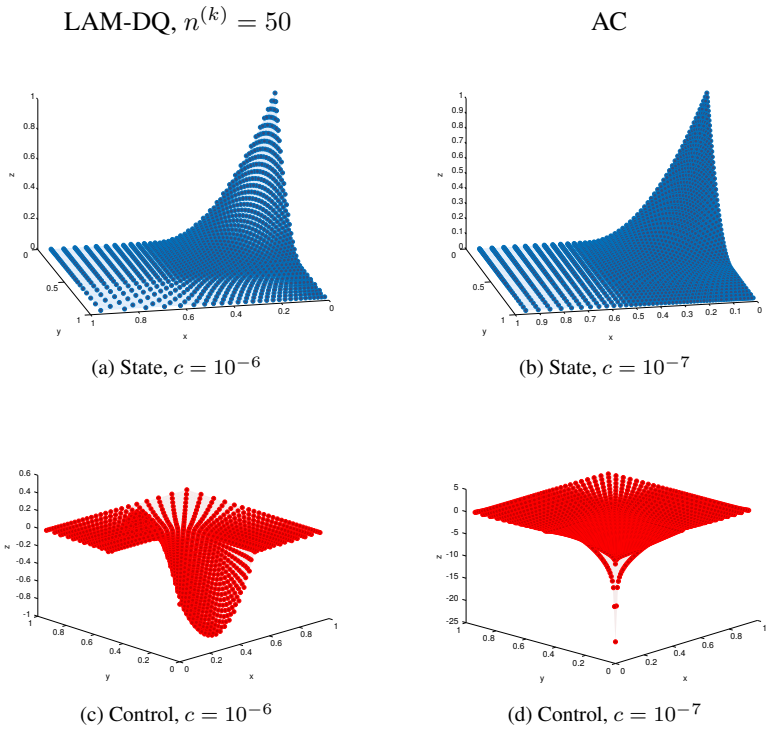
$$g = \begin{cases} (2x_1 - 1)^2 (2x_2 - 1)^2 & \text{in } \left[0, \frac{1}{2}\right]^2 \cap \partial\Omega \\ 0 & \text{elsewhere} \end{cases}$$

$$\omega = (\cos\theta, \sin\theta), \text{ with } \theta = 2.4$$

$$\epsilon = \frac{1}{200}$$

	LAM-DQ			AC		
β	10^{-2}	10^{-6}	10^{-10}	10^{-2}	10^{-6}	10^{-10}
c	4.00×10^{-4}	7.00×10^{-3}	7.00×10^{-3}	4.00×10^{-5}	6.00×10^{-4}	6.00×10^{-4}
$\|y-\hat{y}\|$	3.57×10^{-1}	1.49×10^{-4}	1.49×10^{-8}	1.47×10^{-1}	1.65×10^{-4}	1.66×10^{-8}
$\|u\|$	5.62	3.00	3.00	1.35	3.20	3.20
Cost	2.22×10^{-1}	4.51×10^{-6}	4.50×10^{-10}	2.00×10^{-2}	5.12×10^{-6}	5.11×10^{-10}
κ	6.52×10^{5}	1.25×10^{9}	6.52×10^{8}	2.63×10^{5}	2.41×10^{6}	2.41×10^{6}
$\kappa(S)$	5.25×10^{2}	2.27	1.00			
Time	10sec	10sec	10sec	1min 52sec	1min 52sec	1min 52sec

Table 5: Results from problem 3. For LAM-DQ $n^{(k)} = 50$ and $\kappa = \max_k \kappa(A^{(k)})$. For AC $\kappa = \kappa(G)$. In both cases $n = 622$



(a) State, $c = 10^{-6}$

(b) State, $c = 10^{-7}$

(c) Control, $c = 10^{-6}$

(d) Control, $c = 10^{-7}$

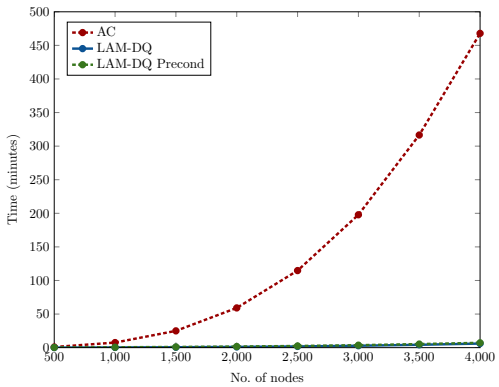
Figure 4: Solution for problem 3, $n = 3021$, $\beta = 10^{-10}$

Table 5 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.

The results reported in the table 5 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 $\|y-\hat{y}\|_{L_2(\Omega)}$

No. Nodos	AC	LAM-DQ	LAM-DQ Precond
500	58sec	7sec	15sec
1000	7min 34sec	18sec	34sec
1500	24min 59sec	36sec	1min 1sec
2000	59min 9sec	1min 5sec	1min 38sec
2500	114min 44sec	1min 49sec	2min 29sec
3000	197min 50sec	2min 51sec	3min 40sec
3500	316min 33sec	4min 15sec	5min 19sec
4000	467min 52sec	6min 6sec	7min 12sec

(a) Table



(b) Graph

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

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 4 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 [16] and [19], 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 β are taken or if it is a convection-diffusion or Poisson control. Table 5a shows the different calculation times by varying the total number of nodes, showing that for all cases LAM-DQ has a smaller execution time in all cases. Figure 5b shows in a more clear way the difference between the computing time of both methods, showing the advantage of LHI-DQ to solve massive problems.

It is worth to make some remarks on the computational complexity of these methods. There are two parts involved in the algorithm: the first consists of solving the

biharmonic problem through LAM technique to calculate the state. The second uses DQ to compute the control.

For the LAM algorithm, we first need to build the $n_c \times n_c$ global matrix S. Each row of S is composed of the weights obtained by solving a local system, so a total of n_c local systems need to be solved in order to build S. For each local system, we need to determine the $n^{(k)}$ nearest neighbors nodes to build it. This is done by a k-d tree technique which takes $\mathcal{O}(n^{(k)} \log n_c) = \mathcal{O}(\log n_c)$ operations, since $n^{(k)}$ is constant with respect to n_c. Then, each local system is solved by LU factorization which is well known to be of cubic order. $A^{(k)}$ is a $(n^{(k)} + n_p) \times (n^{(k)} + n_p)$ matrix, so solving this system is of constant order with respect to n_c. This implies that the overall process of building S is $\mathcal{O}(n)$.

The global sparse system $Sy_c = b$ is efficiently solved by the LU factorization with partial pivoting `cusolverSp` of `CUDA`, so the complexity can be estimated as follows. As we pointed out earlier, each row of S has $n_c^{(k)}$ nonzero entries, which is a constant equal to the number of centers in the local supports. It is well known (see [21]), that the computational complexity of LU for a band matrix with bandwidth k is $k^2 n$. Therefore, for S the bandwidth should be $k = \max(n_c^{(k)}) < n^{(k)}$ which is constant with respect to n. Thus we have that the complexity of LAM technique is of order $\mathcal{O}(n^2)$ when $k^2 \leq n$, namely it is quadratic.

Analogously for DQ we have that it has quadratic complexity since each row of the weight matrix has $n^{(k)}$ nonzero entries. Then, LAM-DQ has a complexity of $\mathcal{O}(n^2)$. Importantly, this can be verified numerically from figure 5b, where the behavior of number of nodes against time can easily be seen to be cubic for AC whereas it is quadratic for LAM-DQ, which verifies the analytical reasoning.

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 [5], we formulated two local techniques, LAM-DQ and LAM-LAM.

A saddle point problem is obtained if we discretize the primal and adjoint equations by using LHI. 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 collocation 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 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.

As we mentioned at the introduction, although here we performed our research using the extended precision approach, several alternatives to the ill-conditioned problem of RBFs collocation methods have been recently formulated, for example [12], [13], [14]. These techniques, which are currently an important active field of research, are of interest and will be considered in further works related to this problem.

Despite that these approaches are of interest and will be considered in further works, we believe that the methods based on extended precision and the analysis proposed in this article present a significant contribution which shows a way to solve large distributed control problems.

Acknowledgements

We wish to thank the two referees for their careful reading of the manuscript and their valuable comments that helped improve the paper. This work was supported by the National Autonomous University of México [grant: PAPIIT, IN102116] and in part by M2NUM project (co-financed by the European Union with the European regional development fund (ERDF, HN0002137) and by the Normandie Regional Council). P. Gonzalez Casanova and C. Gout thanks ECOS Nord project for supporting this work (M15M01).

References

References

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

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

[3] W. Chen, Z. J. Fu, C. S. Chen, Recent advances in radial basis function collocation methods, Springer-Verlag Berlin Heidelberg, 2014.

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

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

[6] M. Benzi, G. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numerica 14 (2005) 1–137.

[7] V. Bayona, M. Moscoso, M. Carretero, M. Kindelan, Rbf-fd formulas and convergence properties, Journal of Computational Physics 229 (22) (2010) 8281–8295.

[8] C. Shu, H. Ding, 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) (2003) 941–954.

[9] D. Cervantes, P. González-Casanova, C. Gout, L. Juárez, L. Reséndiz, Vector field approximation using radial basis functions, Journal of Computational and Applied Mathematics 240 (2013) 163–173.

[10] D. A. Cervantes, P. González-Casanova, C. Gout, M. A. Moreles, A line search algorithm for wind field adjustment with incomplete data and rbf approximation, Computational and Applied Mathematics 37 (3) (2018) 2519–2532.

[11] E. J. Kansa, P. Holoborodko, On the ill-conditioned nature of c^{∞} rbf strong collocation, Engineering Analysis with Boundary Elements 78 (2017) 26–30.

[12] E. Lehto, V. Shankar, G.-B. Wright, A radial basis function (RBF) compact finite difference (FD) scheme for reaction-diffusion equations on surfaces, SIAM J. Scientific Computing 39 (5).

[13] B. Fornberg, E. Lehto, C. Powell, Stable calculation of gaussian-based rbf-fd stencils, Computers and Mathematics with Applications 65 (4) (2013) 627–637.

[14] B. Fornberg, N. Flyer, A primer on radial basis functions with applications to the geosciences, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2015.

[15] J. L. Lions, Optimal control of systems governed by partial differential equations, Springer-Verlag, Berlin, New York, 1971.

[16] T. Rees, Preconditioning iterative methods for pde constrained optimization, Ph.D. thesis, University of Oxford (2010).

[17] J. W. Pearson, A. J. Wathen, Fast iterative solvers for convection-diffusion control problems, Electronic Transactions on Numerical Analysis 40 (2013) 294–310.

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

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

[20] H. C. Elman, D. J. Silvester, A. J. Wathen, Finite Elements and Fast Iterative Solvers : with Applications in Incompressible Fluid Dynamics, Numerical Mathematics and Scientific Computation, OUP Oxford, 2005.

[21] I. S. Duff, A. M. Erisman, J. K. Reid, Direct Methods for Sparse Matrices, Oxford University Press, Inc., New York, NY, USA, 1986.