

A Radial Basis Function Method for Solving PDE Constrained Optimization Problems

John W. Pearson

In this article, we apply the theory of meshfree methods to the problem of PDE constrained optimization. We derive new collocation-type methods to solve the distributed control problem with Dirichlet boundary conditions and the Neumann boundary control problem, both involving Poisson's equation. We prove results concerning invertibility of the matrix systems we generate, and discuss a modification to guarantee invertibility. We implement these methods using MATLAB, and produce numerical results to demonstrate the methods' capability. We also comment on the methods' effectiveness in comparison to the widely-used finite element formulation of the problem, and make some recommendations as to how this work may be extended.

Key words and phrases: Radial basis functions; PDE constrained optimization; Poisson control; collocation method.

Oxford University Mathematical Institute
Numerical Analysis Group
24-29 St Giles'
Oxford, England OX1 3LB
E-mail: john.pearson@maths.ox.ac.uk

July, 2011

1 Introduction

A fast-developing area of research is the application of meshfree methods, using radial basis functions (RBFs), to find the numerical solution of partial differential equations (PDEs) [3, 4, 5, 12]. A starting point for many such pieces of work is Poisson's equation [1, 6, 13]. In this article, we extend this research to an area of considerable recent attention, that of PDE constrained optimization (see for example [15, 16, 17, 18]). We investigate the applicability of RBF collocation methods to the problem of Poisson control (that is, optimizing a quantity with Poisson's equation as a constraint), which is probably the most commonly investigated PDE constrained optimization problem. When Poisson control is investigated however, the method used is generally that of applying a finite element method (FEM) to it. To our knowledge, the application of an RBF method to this problem has not been undertaken. We comment on the solution of a number of problems using this method with Gaussian kernels, as well as the effect of altering the RBF parameter, the number of RBFs used, and the penalty constant (which we will define in Section 3.1) in the control problem.

In Section 2, we give a brief overview of research that has been undertaken in the field of numerical solution of PDEs, in particular Poisson's equation, using RBF collocation methods. In Section 3, we then derive a method to solve problems involving the distributed control of Poisson's equation with Dirichlet boundary conditions and the Neumann boundary control of Poisson's equation. This involves using a Lagrange multiplier approach with a similar collocation strategy as one could use to solve the PDE itself. In Section 5, we test our method on a number of problems to illustrate its effectiveness. We will also comment on the method's performance when parameters and the penalty constant are varied.

2 Meshfree solution of Poisson's equation

Recently, the application of meshfree methods to solve PDE problems has become a topic of much research. Literature such as [1, 4, 5, 6] motivate and derive such methods, and apply them to Poisson's equation. In [13], three collocation strategies are discussed for this problem. The first is *straight collocation*, where the solution is taken to be a straightforward sum of RBFs multiplied by coefficients, which are found by solving a matrix system. The 'centres' of the RBFs are taken to be inside the domain where the solution is sought, with some on the boundary. The second method is *symmetric collocation*, where one modifies the basis function in the interpolant by including the operator involved in the PDE being solved. The third method, that of *direct collocation*, is similar to the first, but also uses the PDE on the boundary, and takes the centres of some of the RBFs to be outside the domain.

In [19], convergence proofs and error estimates are shown for meshless Galerkin methods for a class of PDEs of which Poisson's equation is an example, and it is shown that the error bounds obtained in the energy norm are the same as for a finite element approach. Another useful contribution is the discussion of Domain Decomposition Methods

for preconditioning such problems, as discussed in [14].

Meshfree methods have also been researched for solving other PDEs; for example in [3], a meshless method for solving the steady state Navier-Stokes equations has been investigated. In [12], amongst a number of other issues, a solution procedure for hyperbolic and parabolic PDEs is also discussed.

If one wishes to solve Poisson's equation within a domain Ω using straight collocation, one takes n RBFs with centres at the interior points $\boldsymbol{\xi}_j \in \Omega_{\text{interior}}$, $j = 1, \dots, n$ and n_∂ RBFs with centres at $\boldsymbol{\xi}_j \in \partial\Omega$, $j = n + 1, \dots, n + n_\partial$. We denote the RBF centred at $\boldsymbol{\xi}_j$ as ϕ_j . Then we approximate our solution variable y by $\sum_{i=1}^{n+n_\partial} Y_i \phi_i$, where Y_i are coefficients to be found. Each RBF takes the form $\phi = \phi(r)$, with $r = \|\mathbf{x} - \boldsymbol{\xi}\|$, where \mathbf{x} denotes the position vector and $\boldsymbol{\xi}$ denotes the centre of the RBF.

Now suppose that we wish to solve Poisson's equation with Dirichlet boundary conditions, that is

$$\begin{aligned} -\nabla^2 y &= g \quad \text{in } \Omega, \\ y &= f \quad \text{on } \partial\Omega, \end{aligned}$$

in this way. Then the matrix system that we obtain from straight collocation is

$$\begin{bmatrix} A \\ D_1 \end{bmatrix} [\mathbf{y}] = \begin{bmatrix} \mathbf{g} \\ \mathbf{f} \end{bmatrix}, \quad (2.1)$$

where \mathbf{y} denotes the vector of coefficients Y_i , $i = 1, \dots, n + n_\partial$, and A , D_1 , \mathbf{g} and \mathbf{f} are given by

$$\begin{aligned} A &= \{a_{ij}\}_{i=1, \dots, n, j=1, \dots, n+n_\partial}, & a_{ij} &= -\nabla^2 \phi_j|_{\mathbf{x}=\boldsymbol{\xi}_i}, \\ D_1 &= \{d_{1,ij}\}_{i=1, \dots, n_\partial, j=1, \dots, n+n_\partial}, & d_{1,ij} &= \phi_j|_{\mathbf{x}=\boldsymbol{\xi}_{n+i}}, \\ \mathbf{g} &= \{g_i\}_{i=1, \dots, n}, & g_i &= g(\boldsymbol{\xi}_i) \\ \mathbf{f} &= \{f_i\}_{i=1, \dots, n_\partial}, & f_i &= f(\boldsymbol{\xi}_{n+i}), \end{aligned}$$

with ∇^2 , as in the rest of this paper, denoting the Laplacian with respect to the position vector \mathbf{x} .

On the other hand, if we are solving Poisson's equation with Neumann boundary conditions, that is

$$\begin{aligned} -\nabla^2 y &= g \quad \text{in } \Omega, \\ \frac{\partial y}{\partial n} &= h \quad \text{on } \partial\Omega, \end{aligned}$$

in this way, then the resulting matrix system that is

$$\begin{bmatrix} A \\ N_1 \end{bmatrix} [\mathbf{y}] = \begin{bmatrix} \mathbf{g} \\ \mathbf{h} \end{bmatrix}, \quad (2.2)$$

where A and \mathbf{g} are as above, and N_1 and \mathbf{h} are given by

$$\begin{aligned} N_1 &= \{n_{1,ij}\}_{i=1, \dots, n_\partial, j=1, \dots, n+n_\partial}, & n_{1,ij} &= \left. \frac{\partial \phi_j}{\partial n} \right|_{\mathbf{x}=\boldsymbol{\xi}_{n+i}}, \\ \mathbf{h} &= \{h_i\}_{i=1, \dots, n_\partial}, & h_i &= h(\boldsymbol{\xi}_{n+i}). \end{aligned}$$

Using direct collocation to solve these systems will yield similar systems for the Dirichlet and Neumann problems. When using symmetric collocation, we instead approximate y as $\sum_{i=1}^n Y_i \nabla_{\boldsymbol{\xi}}^2 \phi_i + \sum_{i=n+1}^{n+n_{\partial}} Y_i \phi_i$ (where $\nabla_{\boldsymbol{\xi}}$ denotes the Laplacian with respect to $\boldsymbol{\xi}$). When solving the Dirichlet problem for instance, this will yield the following matrix system:

$$\begin{bmatrix} \tilde{A} \\ \tilde{D}_1 \end{bmatrix} [\mathbf{y}] = \begin{bmatrix} \mathbf{h} \\ \mathbf{f} \end{bmatrix}, \quad (2.3)$$

where \mathbf{h} and \mathbf{f} are as above, and

$$\begin{aligned} \tilde{A} &= \begin{bmatrix} \tilde{A}_1 & \tilde{A}_2 \end{bmatrix}, \\ \tilde{D}_1 &= \begin{bmatrix} \tilde{D}_{11} & \tilde{D}_{12} \end{bmatrix}, \end{aligned}$$

with

$$\begin{aligned} \tilde{A}_1 &= \{\tilde{a}_{1,ij}\}_{i=1,\dots,n, j=1,\dots,n}, & \tilde{a}_{1,ij} &= -\nabla_{\boldsymbol{\xi}}^2(\nabla^2 \phi_j)|_{\mathbf{x}=\boldsymbol{\xi}_i}, \\ \tilde{A}_2 &= \{\tilde{a}_{2,ij}\}_{i=1,\dots,n, j=1,\dots,n_{\partial}}, & \tilde{a}_{2,ij} &= -\nabla^2 \phi_j|_{\mathbf{x}=\boldsymbol{\xi}_{n+i}}, \\ \tilde{D}_{11} &= \{\tilde{d}_{11,ij}\}_{i=1,\dots,n_{\partial}, j=1,\dots,n}, & \tilde{d}_{11,ij} &= \nabla_{\boldsymbol{\xi}}^2 \phi_j|_{\mathbf{x}=\boldsymbol{\xi}_i}, \\ \tilde{D}_{12} &= \{\tilde{d}_{12,ij}\}_{i=1,\dots,n_{\partial}, j=1,\dots,n_{\partial}}, & \tilde{d}_{12,ij} &= \phi_j|_{\mathbf{x}=\boldsymbol{\xi}_{n+i}}. \end{aligned}$$

It should be noted that in general, more mesh points are needed when using symmetric collocation to obtain the same accuracy as when using straight collocation [8, 9]. However, when using the symmetric collocation method, invertibility is guaranteed for all configurations of (distinct) centres of RBFs [13, 20].

We will find that many of the matrices that appear when deriving the model for solving Poisson's equation using an RBF method will also appear when solving the Poisson control problems discussed in Section 3. The methods we derive in Section 3 will be based on the straight collocation and symmetric collocation methods.

3 Extension to Poisson control

In this Section, we explain how we extend the theory of solving Poisson's equation using a collocation method to solving the problem of *Poisson control*, that is minimizing a functional such that Poisson's equation holds (an example of a *PDE constrained optimization* problem).

3.1 Formulation of the distributed control problem

The problem that we examine is of the form

$$\min_{y,u} \frac{1}{2} \|y - \hat{y}\|_{L_2(\Omega)}^2 + \frac{\beta}{2} \|u\|_{L_2(\Omega)}^2 \quad (3.1)$$

such that

$$\begin{aligned} -\nabla^2 y &= u & \text{in } \Omega, \\ y &= f & \text{on } \partial\Omega, \end{aligned}$$

for some *penalty constant* $\beta > 0$. Throughout this work, we take $\Omega = [0, 1]^2$. The value of β is important; if β is small, then y being ‘far away’ from \hat{y} is penalized heavily, whereas if β is large, it is more important that u has small $L_2(\Omega)$ -norm. In this Section, we consider a method based on straight collocation as defined in Section 2; a method based on symmetric collocation is given in Section 3.2.

We define y here to be the *state* and u to be the *control*. The fact that we are controlling u inside the entire domain Ω makes this a *distributed control* problem. Our theory will be extended in Section 3.3 to a *boundary control problem*, that is a problem where we could control u only over the boundary of our domain $\partial\Omega$.

We first turn to interpreting this problem as a matrix system of equations. To do this, we discretize y , u as y_h , u_h , and introduce Lagrange multipliers λ , μ , χ , which we discretize as λ_h , μ_h and χ_h . We collocate our variables as follows:

$$\begin{aligned} y_h &= \sum_{i=1}^{n+n_\partial} Y_i \phi_i, & u_h &= \sum_{i=1}^{n+n_\partial} U_i \phi_i, \\ \lambda_h &= \sum_{i=1}^n \Lambda_i \phi_i, & \mu_h &= \sum_{i=n+1}^{n+n_\partial} M_i \phi_i, & \chi_h &= \sum_{i=n+1}^{n+n_\partial} X_i \phi_i. \end{aligned}$$

where ϕ_1, \dots, ϕ_n denote trial functions whose centres are located inside the domain, and $\phi_{n+1}, \dots, \phi_{n+n_\partial}$ denote trial functions whose centres are on the boundary. These ϕ_i are defined in terms of radial basis functions.

We now define the vectors \mathbf{y} , \mathbf{u} , $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$ and $\boldsymbol{\chi}$ as

$$\begin{aligned} \mathbf{y} &= \{y_i\}_{i=1, \dots, n+n_\partial}, & \mathbf{u} &= \{u_i\}_{i=1, \dots, n+n_\partial}, \\ \boldsymbol{\lambda} &= \{\lambda_i\}_{i=1, \dots, n}, & \boldsymbol{\mu} &= \{\mu_i\}_{i=1, \dots, n_\partial}, & \boldsymbol{\chi} &= \{\chi_i\}_{i=1, \dots, n_\partial}, \end{aligned}$$

at which point we can write our discretized problem as

$$\min_{y_h, u_h} \frac{1}{2} \|y_h - \hat{y}\|_{L_2(\Omega)}^2 + \frac{\beta}{2} \|u_h\|_{L_2(\Omega)}^2$$

such that

$$-\nabla^2 y_h = u_h \quad \text{in } \Omega,$$

with y_h attaining the same values as f at all the radial basis function centres on the boundary.

Now, in the L_2 -norm, the quantities $\frac{1}{2} \|y_h - \hat{y}\|_{L_2(\Omega)}^2$ and $\frac{\beta}{2} \|u_h\|_{L_2(\Omega)}^2$ can be written as

$$\begin{aligned} \frac{1}{2} \|y_h - \hat{y}\|_{L_2(\Omega)}^2 &= \frac{1}{2} \int_{\Omega} (y_h - \hat{y})^2 d\Omega \\ &= \frac{1}{2} \int_{\Omega} y_h^2 d\Omega - \int_{\Omega} y_h \hat{y} d\Omega + \frac{1}{2} \int_{\Omega} \hat{y}^2 d\Omega \\ &= \frac{1}{2} \mathbf{y}^T M_y \mathbf{y} - \mathbf{d}^T \mathbf{y} + C, \\ \frac{\beta}{2} \|u_h\|_{L_2(\Omega)}^2 &= \frac{\beta}{2} \int_{\Omega} u_h^2 d\Omega \\ &= \frac{\beta}{2} \mathbf{u}^T M_u \mathbf{u}, \end{aligned}$$

where M_y and M_u are *mass matrices* (which are equal due to the fact that y and u are discretized in the same way in our formulation) defined as

$$M_y = \{m_{ij}^y\}_{i,j=1,\dots,n+n_{\partial}}, \quad m_{ij}^y = \int_{\Omega} \phi_i \phi_j d\Omega,$$

\mathbf{d} is defined as

$$\mathbf{d} = \{d_i\}_{i=1,\dots,n+n_{\partial}}, \quad d_i = \int_{\Omega} \hat{y} \phi_i d\Omega,$$

and C is a constant, the value of which we will see to be unimportant.

We use a straight collocation method similar to the method used for solving the PDE itself (as discussed in Section 2), to obtain three equations as shown below. The first guarantees that Poisson's equation is solved at every point corresponding to a centre of an RBF, the second ensures that the Dirichlet boundary conditions are satisfied, and the third ensures that the control is set to zero on the boundary (as for these problems we only wish to "control" the interior of the domain Ω).

$$\begin{aligned} A\mathbf{y} + D_2\mathbf{u} &= \mathbf{0}, \\ D_1\mathbf{y} &= \mathbf{f}, \\ D_1\mathbf{u} &= \mathbf{0}, \end{aligned}$$

where

$$\begin{aligned} A &= \{a_{ij}\}_{i=1,\dots,n, j=1,\dots,n+n_{\partial}}, \quad a_{ij} = -\nabla^2 \phi_j|_{\mathbf{x}=\boldsymbol{\xi}_i}, \\ D_1 &= \{d_{1,ij}\}_{i=1,\dots,n_{\partial}, j=1,\dots,n+n_{\partial}}, \quad d_{1,ij} = \phi_j|_{\mathbf{x}=\boldsymbol{\xi}_{n+i}}, \\ D_2 &= \{d_{2,ij}\}_{i=1,\dots,n, j=1,\dots,n+n_{\partial}}, \quad d_{2,ij} = -\phi_j|_{\mathbf{x}=\boldsymbol{\xi}_i}, \\ \mathbf{f} &= \{f_i\}_{i=1,\dots,n_{\partial}}, \quad f_i = f(x_{n+i}). \end{aligned}$$

As before, $\boldsymbol{\xi}_i$ denotes the centre of the i -th RBF we are considering, and \mathbf{f} is the vector containing the boundary values (i.e. the specified values at nodes $\mathbf{x}_{n+1}, \dots, \mathbf{x}_{n+n_{\partial}}$).

We are therefore now left with the following Lagrangian $\mathcal{L} = \mathcal{L}(\mathbf{y}, \mathbf{u}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\chi})$, of which we wish to find the stationary point:

$$\mathcal{L} = \frac{1}{2}\mathbf{y}^T M_y \mathbf{y} - \mathbf{d}^T \mathbf{y} + C + \frac{\beta}{2}\mathbf{u}^T M_u \mathbf{u} + \boldsymbol{\lambda}^T (A\mathbf{y} + D_2\mathbf{u}) + \boldsymbol{\mu}^T (D_1\mathbf{y} - \mathbf{g}) + \boldsymbol{\chi}^T (D_1\mathbf{u}),$$

where $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$ and $\boldsymbol{\chi}$ denote the vectors of the relevant coefficients for our three *Lagrange multipliers*.

Now, differentiating \mathcal{L} with respect to \mathbf{y} , \mathbf{u} , $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$ and $\boldsymbol{\chi}$ and setting the partial derivatives to be zero gives us the following equations which must be satisfied:

$$\begin{aligned} \frac{\partial}{\partial \mathbf{y}} : \quad & M_y \mathbf{y} + A^T \boldsymbol{\lambda} + D_1^T \boldsymbol{\mu} = \mathbf{d}, \\ \frac{\partial}{\partial \mathbf{u}} : \quad & \beta M_u \mathbf{u} + D_2^T \boldsymbol{\lambda} + D_1^T \boldsymbol{\chi} = \mathbf{0}, \\ \frac{\partial}{\partial \boldsymbol{\lambda}} : \quad & A\mathbf{y} + D_2\mathbf{u} = \mathbf{0}, \\ \frac{\partial}{\partial \boldsymbol{\mu}} : \quad & D_1\mathbf{y} = \mathbf{f}, \\ \frac{\partial}{\partial \boldsymbol{\chi}} : \quad & D_1\mathbf{u} = \mathbf{0}. \end{aligned}$$

Finally, combining these 5 equations gives the system that we wish to solve:

$$\begin{bmatrix} M_y & 0 & A & D_1^T & 0 \\ 0 & \beta M_u & D_2^T & 0 & D_1^T \\ A & D_2 & 0 & 0 & 0 \\ D_1 & 0 & 0 & 0 & 0 \\ 0 & D_1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \boldsymbol{\lambda} \\ \boldsymbol{\mu} \\ \boldsymbol{\chi} \end{bmatrix} = \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{f} \\ \mathbf{0} \end{bmatrix}. \quad (3.2)$$

The system that we obtain is symmetric, and furthermore is of *saddle point* structure, which is guaranteed if we use the above ‘discretize-then-optimize’ formulation. This is useful for proving the conditions for invertibility of this system, as in Section 4.

3.2 An alternative expansion in terms of basis functions

In this Section, we discuss an alternative method for solving (3.1), motivated by the fact that when solving Poisson’s equation using symmetric collocation, one is guaranteed to be solving an invertible matrix system, whatever the configuration of RBF centres. The consequence of this with regard to the distributed control problem is given in Theorem 3, though this methodology could also be extended to Neumann boundary control problems.

To carry out this method, we take

$$y_h = \sum_{i=1}^n Y_i \nabla_{\xi}^2 \phi_i + \sum_{i=n+1}^{n+n_{\partial}} Y_i \phi_i, \quad u_h = \sum_{i=1}^n U_i \nabla_{\xi}^2 \phi_i + \sum_{i=n+1}^{n+n_{\partial}} U_i \phi_i,$$

$$\lambda_h = \sum_{i=1}^n \Lambda_i \nabla_{\xi}^2 \phi_i, \quad \mu_h = \sum_{i=n+1}^{n+n_{\partial}} M_i \phi_i, \quad \chi_h = \sum_{i=n+1}^{n+n_{\partial}} X_i \phi_i.$$

Using the same method as in Section 3.1, we obtain the matrix system

$$\begin{bmatrix} \tilde{M}_y & 0 & \tilde{A}^T & \tilde{D}_1^T & 0 \\ 0 & \beta \tilde{M}_u & \tilde{D}_2^T & 0 & \tilde{D}_1^T \\ \tilde{A} & \tilde{D}_2 & 0 & 0 & 0 \\ \tilde{D}_1 & 0 & 0 & 0 & 0 \\ 0 & \tilde{D}_1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \boldsymbol{\lambda} \\ \boldsymbol{\mu} \\ \boldsymbol{\chi} \end{bmatrix} = \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{g} \\ \mathbf{0} \end{bmatrix}, \quad (3.3)$$

where \mathbf{d} and \mathbf{f} are as defined in Section 3.1, and

$$\tilde{M}_y = \tilde{M}_u = \begin{bmatrix} \tilde{M}_{11} & \tilde{M}_{12} \\ \tilde{M}_{21} & \tilde{M}_{22} \end{bmatrix},$$

$$\tilde{A} = \begin{bmatrix} \tilde{A}_1 & \tilde{A}_2 \end{bmatrix},$$

$$\tilde{D}_1 = \begin{bmatrix} \tilde{D}_{11} & \tilde{D}_{12} \end{bmatrix},$$

$$\tilde{D}_2 = \begin{bmatrix} \tilde{D}_{21} & \tilde{D}_{22} \end{bmatrix},$$

with

$$\tilde{M}_{11} = \{\tilde{m}_{11,ij}\}_{i=1,\dots,n, j=1,\dots,n}, \quad \tilde{m}_{11,ij} = \int_{\Omega} \nabla^2 \phi_i \cdot \nabla^2 \phi_j d\Omega,$$

$$\tilde{M}_{12} = \{\tilde{m}_{12,ij}\}_{i=1,\dots,n, j=1,\dots,n_{\partial}}, \quad \tilde{m}_{12,ij} = \int_{\Omega} \nabla^2 \phi_i \cdot \phi_{n+j} d\Omega,$$

$$\tilde{M}_{21} = \{\tilde{m}_{21,ij}\}_{i=1,\dots,n_{\partial}, j=1,\dots,n}, \quad \tilde{m}_{21,ij} = \int_{\Omega} \phi_{n+i} \cdot \nabla^2 \phi_j d\Omega,$$

$$\tilde{M}_{22} = \{\tilde{m}_{22,ij}\}_{i=1,\dots,n_{\partial}, j=1,\dots,n_{\partial}}, \quad \tilde{m}_{22,ij} = \int_{\Omega} \phi_{n+i} \cdot \phi_{n+j} d\Omega,$$

$$\tilde{A}_1 = \{\tilde{a}_{1,ij}\}_{i=1,\dots,n, j=1,\dots,n}, \quad \tilde{a}_{1,ij} = -\nabla^2 (\nabla_{\xi}^2 \phi_j)|_{\mathbf{x}=\boldsymbol{\xi}_i},$$

$$\tilde{A}_2 = \{\tilde{a}_{2,ij}\}_{i=1,\dots,n, j=1,\dots,n_{\partial}}, \quad \tilde{a}_{2,ij} = -\nabla^2 \phi_j|_{\mathbf{x}=\boldsymbol{\xi}_{n+i}},$$

$$\tilde{D}_{11} = \{\tilde{d}_{11,ij}\}_{i=1,\dots,n_{\partial}, j=1,\dots,n}, \quad \tilde{d}_{11,ij} = \nabla_{\xi}^2 \phi_j|_{\mathbf{x}=\boldsymbol{\xi}_i},$$

$$\tilde{D}_{12} = \{\tilde{d}_{12,ij}\}_{i=1,\dots,n_{\partial}, j=1,\dots,n_{\partial}}, \quad \tilde{d}_{12,ij} = \phi_j|_{\mathbf{x}=\boldsymbol{\xi}_{n+i}},$$

$$\tilde{D}_{21} = \{\tilde{d}_{21,ij}\}_{i=1,\dots,n, j=1,\dots,n}, \quad \tilde{d}_{21,ij} = -\nabla_{\xi}^2 \phi_j|_{\mathbf{x}=\boldsymbol{\xi}_i},$$

$$\tilde{D}_{22} = \{\tilde{d}_{22,ij}\}_{i=1,\dots,n, j=1,\dots,n_{\partial}}, \quad \tilde{d}_{22,ij} = -\phi_j|_{\mathbf{x}=\boldsymbol{\xi}_{n+i}}.$$

For the remainder of this paper, we will be working with the simpler formulation derived in Section 3.1, along with equally-spaced RBF centres. However, the system derived in this Section is useful, as we prove in Section 4 that (3.3) is invertible for any configuration of RBF centres (provided the locations of the centres are distinct).

3.3 RBF solution of Neumann boundary control problem

In this Section, we describe a method (based on the straight collocation method for solving Poisson's equation with Neumann boundary conditions) to solve the Neumann boundary control problem

$$\min_{y,u} \frac{1}{2} \|y - \hat{y}\|_{L_2(\Omega)}^2 + \frac{\beta}{2} \|u\|_{L_2(\partial\Omega)}^2$$

such that

$$\begin{aligned} -\nabla^2 y &= g \quad \text{in } \Omega, \\ \frac{\partial y}{\partial n} &= u \quad \text{on } \partial\Omega. \end{aligned}$$

As well as using ϕ_i to denote a basis of RBFs defined within Ω , we now take ψ_i , $i = 1, \dots, n_\partial$, to be a new basis of RBFs defined solely on the boundary $\partial\Omega$. Then writing

$$y_h = \sum_{i=1}^{n+n_\partial} Y_i \phi_i, \quad u_h = \sum_{i=1}^{n_\partial} U_i \psi_i, \quad \lambda_h = \sum_{i=1}^n \Lambda_i \phi_i, \quad \mu_h = \sum_{i=1}^{n_\partial} M_i \psi_i,$$

and proceeding by the same method as in Section 3.2 yields the following matrix system:

$$\begin{bmatrix} M_y & 0 & A^T & N_1^T \\ 0 & \beta \hat{M}_u & 0 & N_2^T \\ A & 0 & 0 & 0 \\ N_1 & N_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \boldsymbol{\lambda} \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \\ \mathbf{h} \\ \mathbf{0} \end{bmatrix}, \quad (3.4)$$

where M_y , A and \mathbf{d} are as defined in Section 3.1, and

$$\begin{aligned} \hat{M}_u &= \{\hat{m}_{ij}^u\}_{i=1, \dots, n_\partial, j=1, \dots, n_\partial}, \quad \hat{m}_{ij}^u = \int_{\partial\Omega} \psi_i \psi_j ds, \\ N_1 &= \{n_{1,ij}\}_{i=1, \dots, n_\partial, j=1, \dots, n+n_\partial}, \quad n_{1,ij} = \left. \frac{\partial \phi_j}{\partial n} \right|_{\mathbf{x}=\boldsymbol{\xi}_{n+i}}, \\ N_2 &= \{n_{2,ij}\}_{i,j=1, \dots, n_\partial}, \quad n_{2,ij} = -\psi_j|_{\mathbf{x}=\boldsymbol{\xi}_{n+i}} = -d_{1,ij}, \\ \mathbf{g} &= \{g_i\}_{i=1, \dots, n_\partial}, \quad g_i = g(\boldsymbol{\xi}_{n+i}). \end{aligned}$$

Theorem 4 in Section 4 gives a result on the invertibility of this matrix system.

3.4 Implementation aspects

When implementing the methods we have derived in Sections 3.1 and 3.3, we need to consider a number of things. First of all, we need to choose what radial basis functions we will use in our methods. A number of widely used RBFs are discussed in literature such as [4, 7, 13]. Throughout this work, we will use Gaussians (which are basis functions

of the form $\phi = \phi(r) = e^{-cr^2}$, although our theory could be extended to any positive definite RBF. We will vary the value of the coefficient c in the Gaussians; as we will demonstrate in Section 5, it is not a trivial problem to choose the appropriate value of c to obtain an accurate solution to the matrix systems previously derived.

The choice of the so called ‘shape parameter’ c , when it appears, is likely to be important, as this feature is observed when solving Poisson’s equation itself [11, 13]. Large values of c result in a well-conditioned problem but poor convergence, whereas small values of c result in a good approximation, but bad conditioning. We will therefore vary the values of c in our numerical tests; all plots shown in Section 5 are produced with values of c seen to be appropriate for the particular problem.

We must also choose the centres of the RBFs (which will be needed to construct the matrices and vectors in (3.2)), and note their locations. We will proceed using randomly generated centres inside the domain and along the boundary for simplicity.

When constructing the matrices as defined in Sections 3.1 and 3.3, we note a couple of things. First, the matrices M_y (and hence M_u if we use the same number of basis functions to represent y and u) and \hat{M}_u are clearly symmetric. This will save some computational time when constructing the matrix system (3.2). When constructing the mass matrices, we will need an effective quadrature rule: we find that using Gauss-Legendre quadrature with around 50 quadrature points in each direction is sufficient when Gaussians are used (note that in this case, we may separate the terms along the individual axes, leaving effectively two 50 point quadrature rules, rather than one 2500 point rule, for a 2D problem). Finally, we note that \mathbf{d} will clearly depend on our choice of \hat{y} , and \mathbf{f} or \mathbf{g} will depend on what we choose the boundary terms to be – this will need to be taken into account when implementing our method. We solve the matrix system we have constructed using ‘`backslash`’ in MATLAB, and perform standard *a posteriori* tests to verify the accuracy of our solutions.

To test our method, we consider four specific problems, listed below. Problems 1, 2 and 3 are all distributed control problems with Dirichlet boundary conditions, which we will solve using the method derived in Section 3.1, and Problem 4 is a Neumann boundary control problem, which we will solve using the method discussed in Section 3.3. The problems are all in two dimensions on a domain $\Omega = [0, 1]^2$ for illustrative purposes, but our theory can be easily extended to solving three dimensional problems. In the problems as stated, $\mathbf{x} = [x_1, x_2]^T$ denote the coordinates we are working with.

- **Problem 1:** We wish to solve the distributed Poisson control problem

$$\min_{y,u} \frac{1}{2} \|y - \hat{y}\|_{L_2(\Omega)}^2 + \frac{\beta}{2} \|u\|_{L_2(\Omega)}^2$$

such that

$$\begin{aligned} -\nabla^2 y &= u, & \mathbf{x} \in \Omega, \\ y &= f, & \mathbf{x} \in \partial\Omega, \end{aligned}$$

where $\Omega = [0, 1]^2$, and

$$\begin{aligned}\hat{y} &= x_1(1 - x_1)x_2(1 - x_2), \quad \mathbf{x} \in \Omega, \\ f &= 0.\end{aligned}$$

- **Problem 2:** We consider the same formulation as Problem 1, except that

$$\begin{aligned}\hat{y} &= \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_1, \\ 0 & \text{if } \mathbf{x} \in \Omega_2, \end{cases} \\ f &= \begin{cases} 1 & \text{if } \mathbf{x} \in \partial\Omega_1 := \partial\Omega \cap \Omega_1, \\ 0 & \text{if } \mathbf{x} \in \partial\Omega_2 := \partial\Omega \cap \Omega_2, \end{cases} \\ \Omega_1 &= [0, \frac{1}{2}]^2, \quad \Omega_2 = \Omega \setminus \Omega_1.\end{aligned}$$

This problem was previously considered using an FEM method in [17].

- **Problem 3:** We again consider the same formulation as Problem 1, but with

$$\begin{aligned}\hat{y} &= \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_1, \\ 0 & \text{if } \mathbf{x} \in \Omega_2, \end{cases} \\ f &= 0, \\ \Omega_1 &= [0, \frac{1}{2}]^2, \quad \Omega_2 = \Omega \setminus \Omega_1.\end{aligned}$$

This problem was previously considered using an FEM method in [15].

- **Problem 4:** We wish to solve the Neumann boundary control problem

$$\min_{y,u} \frac{1}{2} \|y - \hat{y}\|_{L_2(\Omega)}^2 + \frac{\beta}{2} \|u\|_{L_2(\partial\Omega)}^2$$

such that

$$\begin{aligned}-\nabla^2 y &= g, \quad \mathbf{x} \in \Omega, \\ \frac{\partial y}{\partial n} &= u, \quad \mathbf{x} \in \partial\Omega,\end{aligned}$$

where $\Omega = [0, 1]^2$, and

$$\begin{aligned}\hat{y} &= x_1(1 - x_1)x_2(1 - x_2), \quad \mathbf{x} \in \Omega, \\ g &= g(x_1, x_2) = x_1x_2.\end{aligned}$$

Problems 1, 2 and 3 are variants on the Poisson distributed control problem, with different boundary conditions and target functions, and, as stated above, Problem 4 is an example of the Neumann boundary control problem. Problems 2 and 3 are interesting ones specifically because we cannot exactly attain our target solution, no matter how

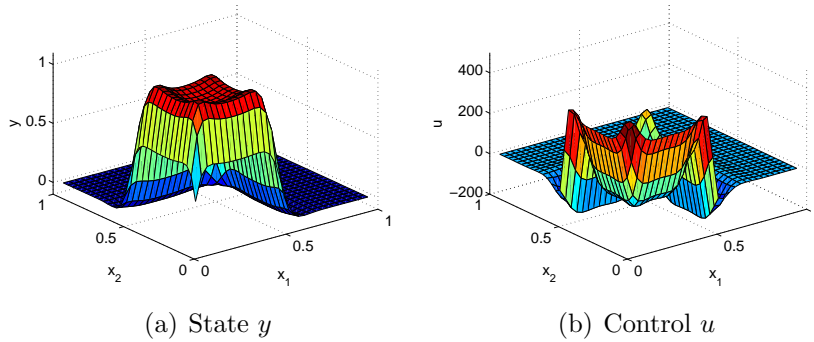


Figure 1: Plot of finite element solution (interior points only) to optimal control Problem 3 using **Q1-Q1** approximation, step size $h = 2^{-5}$ and $\beta = 10^{-6}$, and the control needed to obtain this solution.

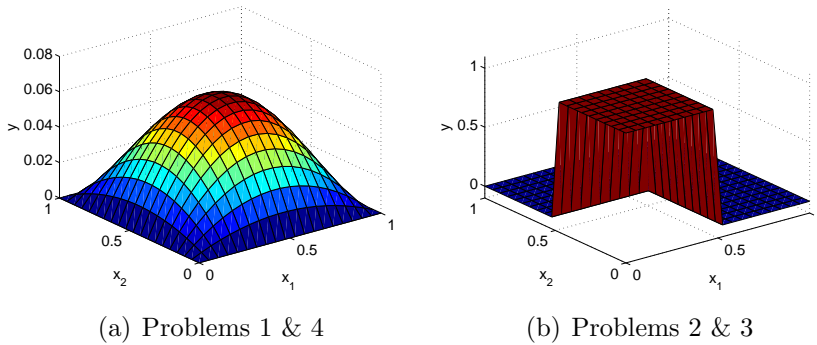


Figure 2: Plots of target solution $y = \hat{y}$ for Poisson control Problems 1 and 4 (left) and Problems 2 and 3 (right).

small we set β . The reason for this is that we have specified that y must be equal to 0 on all of $\partial\Omega$, and so the target $\hat{y} = 1$ on $[0, \frac{1}{2}]^2$ cannot be reached. Consequently, for small β , the control may exhibit a lack of smoothness as y gets closer to \hat{y} . Figure 1 illustrates a finite element solution we obtain for the state when $\beta = 10^{-6}$, and the non-smooth control that we require for this to happen. Figure 2 shows the targets \hat{y} that we hope the state to be as close to as possible, for each of our 4 problems.

4 Solvability of matrix systems

In order to guarantee the existence of solutions to PDE constrained optimization problems using our methods, it is necessary to obtain results on the invertibility of the matrix systems derived in Section 3. We therefore prove a number of such results in this Section.

Theorem 1 is a general result from saddle point theory, see [2]. We will use this result to prove conditions for invertibility of the matrix systems that we have derived in Section 3. Theorem 2 gives a result concerning the invertibility of the matrix system (3.2) for the solution of distributed control problems in terms of the invertibility of the

matrix system (2.1), that is the matrix system generated by solving Poisson's equation with Dirichlet boundary conditions using straight collocation.

Theorem 3 proves the unconditional solvability of the alternative matrix system (3.3) for distributed control problems, using the fact that the matrix system (2.3) is invertible, provided that all RBF centres are distinct [13, 20].

Theorem 4 gives a similar result as Theorem 2, this time concerning the invertibility of (3.4), the matrix system for Neumann boundary control problems, in terms of the invertibility of (2.2), the matrix system generated by solving Poisson's equation with Neumann boundary conditions.

Theorem 1 *Any saddle point system of the form $\begin{bmatrix} \Phi & \Psi^T \\ \Psi & 0 \end{bmatrix}$, where Φ is an $m \times m$ matrix and Ψ is an $n \times m$ matrix with $m \geq n$, is invertible provided that Φ is invertible and the Schur complement $\Psi\Phi^{-1}\Psi^T$ is invertible.*

Theorem 2 *The matrix \mathcal{A} given by*

$$\mathcal{A} = \begin{bmatrix} M_y & 0 & A^T & D_1^T & 0 \\ 0 & \beta M_u & D_2^T & 0 & D_1^T \\ A & D_2 & 0 & 0 & 0 \\ D_1 & 0 & 0 & 0 & 0 \\ 0 & D_1 & 0 & 0 & 0 \end{bmatrix}$$

(that is the matrix corresponding to the Poisson distributed control problem as described in Section 3.1) is invertible, provided $\begin{bmatrix} A \\ D_1 \end{bmatrix}$ is invertible.

Proof. The matrix \mathcal{A} may be written as a saddle point system $\mathcal{A} = \begin{bmatrix} C & E^T \\ E & 0 \end{bmatrix}$, where

$$C = \begin{bmatrix} M_y & 0 \\ 0 & \beta M_u \end{bmatrix} \text{ and } E = \begin{bmatrix} A & D_2 \\ D_1 & 0 \\ 0 & D_1 \end{bmatrix}. \text{ Now, because of Theorem 1, we simply}$$

need to show that C^{-1} and $(EC^{-1}E^T)^{-1}$ exist. The fact that mass matrices are invertible means that we may write $C^{-1} = \begin{bmatrix} M_y^{-1} & 0 \\ 0 & \frac{1}{\beta}M_u^{-1} \end{bmatrix}$ to prove the first part.

We prove that $EC^{-1}E^T$ is invertible by showing that it is positive definite. To do this, we note that, for any $\mathbf{v} \in \mathbb{R}^{n+2n_\partial}$,

$$\mathbf{v}^T EC^{-1}E^T \mathbf{v} = (C^{-1/2}E^T \mathbf{v})^T (C^{-1/2}E^T \mathbf{v}),$$

and so $\mathbf{v}^T EC^{-1}E^T \mathbf{v} > 0$ unless $C^{-1/2}E^T \mathbf{v} = \mathbf{0}$, which occurs if and only if $E^T \mathbf{v} = \mathbf{0}$ as $C^{-1/2}$ is invertible.

Now, writing $\mathbf{v} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \mathbf{v}_3]^T$, where $\mathbf{v}_1 \in \mathbb{R}^n$, $\mathbf{v}_2 \in \mathbb{R}^{n_\partial}$, $\mathbf{v}_3 \in \mathbb{R}^{n_\partial}$, we obtain

$$E^T \mathbf{v} = \mathbf{0} \Leftrightarrow \begin{bmatrix} A \\ D_1 \end{bmatrix}^T \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad D_2^T \mathbf{v}_1 + D_1^T \mathbf{v}_3 = \mathbf{0}.$$

By the invertibility of $\begin{bmatrix} A \\ D_1 \end{bmatrix}$ (and hence also the fact that D_1 has full rank), we have from the first of the above equations that $\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$, from which it follows from the second equation that $\mathbf{v}_3 = \mathbf{0}$. We therefore obtain that $E^T \mathbf{v} = \mathbf{0}$ if and only if $\mathbf{v} = \mathbf{0}$, and so $EC^{-1}E^T$ is invertible. \square

Theorem 3 *The matrix*

$$\begin{bmatrix} \tilde{M}_y & 0 & \tilde{A}^T & \tilde{D}_1^T & 0 \\ 0 & \beta\tilde{M}_u & \tilde{D}_2^T & 0 & \tilde{D}_1^T \\ \tilde{A} & \tilde{D}_2 & 0 & 0 & 0 \\ \tilde{D}_1 & 0 & 0 & 0 & 0 \\ 0 & \tilde{D}_1 & 0 & 0 & 0 \end{bmatrix}$$

is invertible.

Proof. This can be proved in the same way as Theorem 2, using Theorem 1, as well the fact that the matrix $\begin{bmatrix} \tilde{A} \\ \tilde{D}_1 \end{bmatrix}$ is always invertible in the case of symmetric collocation provided that all RBF centres are distinct. \square

Theorem 4 *The matrix*

$$\begin{bmatrix} M_y & 0 & A^T & N_1^T \\ 0 & \beta\hat{M}_u & 0 & N_2^T \\ A & 0 & 0 & 0 \\ N_1 & N_2 & 0 & 0 \end{bmatrix}$$

is invertible, provided $\begin{bmatrix} A \\ N_1 \end{bmatrix}$, the (straight) collocation matrix corresponding to Poisson's equation with Neumann boundary conditions, is invertible.

Proof. This can also be proved in a similar way as Theorem 2 using Theorem 1. \square

In this Section we have therefore shown that if one proves the invertibility of a matrix system for solving Poisson's equation (with Dirichlet or Neumann boundary conditions) using a particular array of nodes, one has automatically proved the invertibility of the equivalent PDE constrained optimization problem (that is the distributed control problem and the Neumann boundary control problem respectively). The invertibility of matrix systems for solving Poisson's equation will hold almost always, but counterexamples can be constructed [10, 13].

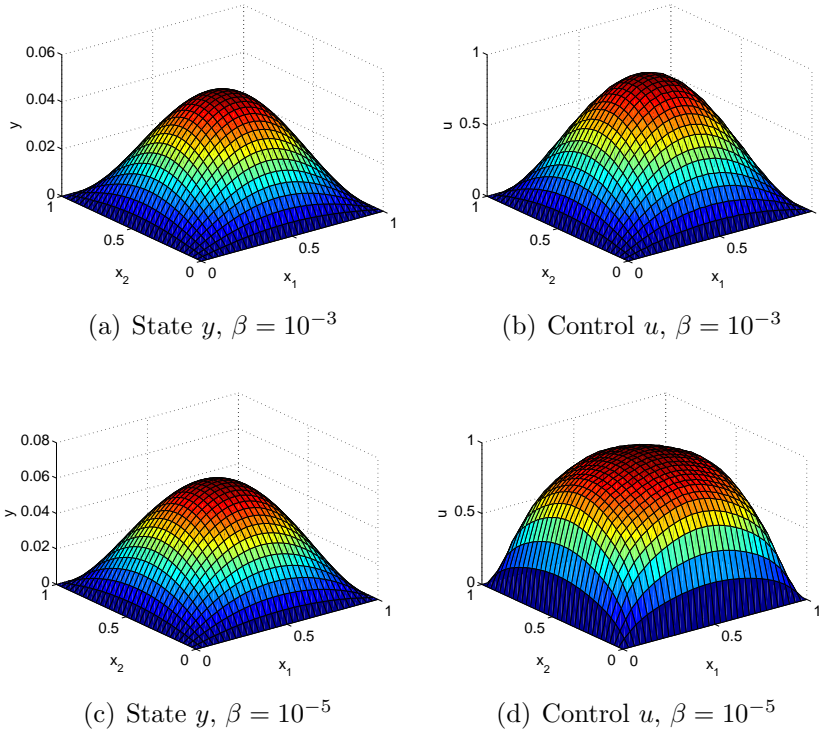


Figure 3: Diagrams of the RBF solutions to the distributed Poisson control Problem 1 for the state y and control u , for $\beta = 10^{-3}$ and 10^{-5} with $c = 150$ and 300 respectively. Here, we took $h = 2^{-5}$.

5 Numerical results

In this Section we present numerical results, all of which were generated using equally-spaced points.

Figures 3, 4, 5 and 6 show solutions obtained for state and control using our RBF methods with $\beta = 10^{-3}$ and $\beta = 10^{-5}$, for the distributed control problems (Figures 3–5) and Neumann boundary control problem (Figure 6) stated in Section 3.4. The figures exhibit many features which we would expect of solutions to a PDE constrained optimization problem. The state y is close to the target function \hat{y} for each problem (as shown in Figure 2), and it is even closer to the target function if we set β to be smaller. In Figure 5 in particular, our control u exhibits very similar non-smoothness near $\partial\Omega$ for small β , as we saw for the finite element solution in Figure 1.

We observe that the solutions we obtained seemed to depend on the value of c we used in our Gaussians. We therefore seek to analyze how the values of c , as well as β , affected the values of $\|y - \hat{y}\|_{L_2(\Omega)}$, $\|u\|_{L_2(\Omega)}$ and $\frac{1}{2}\|y - \hat{y}\|_{L_2(\Omega)}^2 + \frac{\beta}{2}\|u\|_{L_2(\Omega)}^2$ (the functional our method has attempted to minimize). As our method merely generates the values of the state and control at each RBF centre, it is difficult to evaluate any of these functionals exactly, as they represent integrals. We find that if we approximate $\|y - \hat{y}\|_{L_2(\Omega)}$ by the ℓ_2 -norm of the vector of the difference between the RBF solution for y and the target

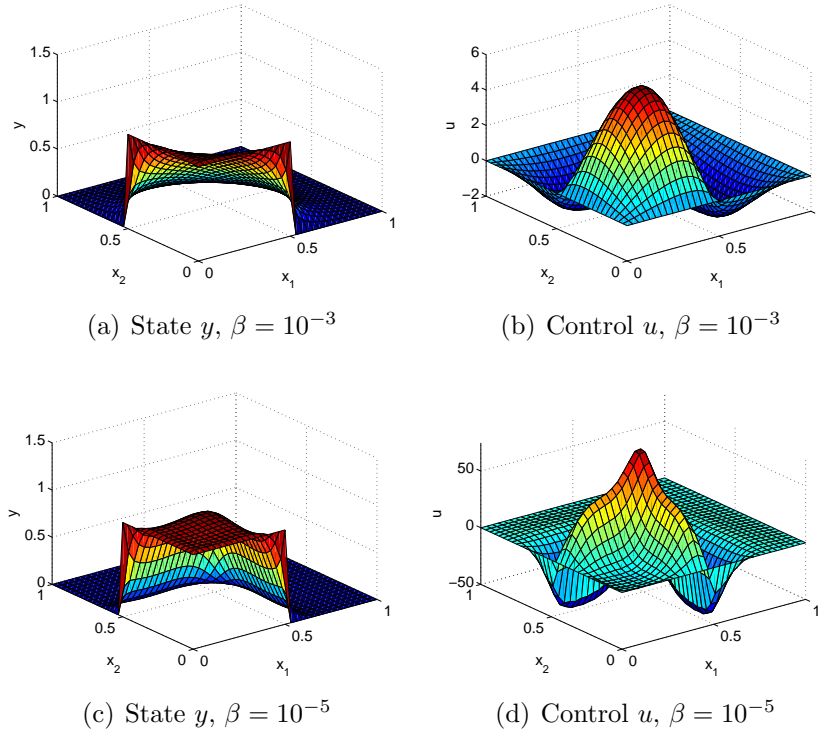


Figure 4: Diagrams of the RBF solutions to the distributed Poisson control Problem 2 for the state y and control u , for $\beta = 10^{-3}$ and 10^{-5} with $c = 250$ in each case. Here, we took $h = 2^{-5}$.

solution \hat{y} at each point (denoted by $\|y - \hat{y}\|_{\ell_2(\Omega)}$), and approximate $\|u\|_{L_2(\Omega)}$ by the ℓ_2 -norm of the vector of the RBF solution of the control u (denoted by $\|u\|_{\ell_2(\Omega)}$), these give an illustration of our success in minimizing $\frac{1}{2} \|y - \hat{y}\|_{L_2(\Omega)}^2 + \frac{\beta}{2} \|u\|_{L_2(\Omega)}^2$. Note that these values will clearly depend on h , but we will keep this value constant when we carry out this analysis.

In Table 1, we analyze the values of $\|y - \hat{y}\|_{\ell_2(\Omega)}$, $\|u\|_{\ell_2(\Omega)}$ and $\frac{1}{2} \|y - \hat{y}\|_{\ell_2(\Omega)}^2 + \frac{\beta}{2} \|u\|_{\ell_2(\Omega)}^2$ for varying β . As we would expect, at first, as β gets smaller, the state y gets rapidly closer to the target function \hat{y} , and although $\|u\|_{\ell_2(\Omega)}$ does become much larger, our approximation of the functional we wish to minimize, $\frac{1}{2} \|y - \hat{y}\|_{L_2(\Omega)}^2 + \frac{\beta}{2} \|u\|_{L_2(\Omega)}^2$, gets smaller and smaller.

In Table 2, we considered a specific case (namely Problem 3 with $h = 0.04$ and $\beta = 10^{-4}$), and again analyzed the values of $\|y - \hat{y}\|_{\ell_2(\Omega)}$, $\|u\|_{\ell_2(\Omega)}$ and $\frac{1}{2} \|y - \hat{y}\|_{\ell_2(\Omega)}^2 + \frac{\beta}{2} \|u\|_{\ell_2(\Omega)}^2$, but this time varying the value of the parameter c . The effects seemed to be similar as for Poisson's equation itself; taking a value of c that is too large results in convergence being a problem, but a value of c that is too small results in ill-conditioning. In Table 2, we observe that the 'middle ground' of values of c (for this case roughly $c \in [200, 500]$) generate very similar results — the results still vary slightly as h is still relatively large, and the solution may not have converged to many digits. We recommend

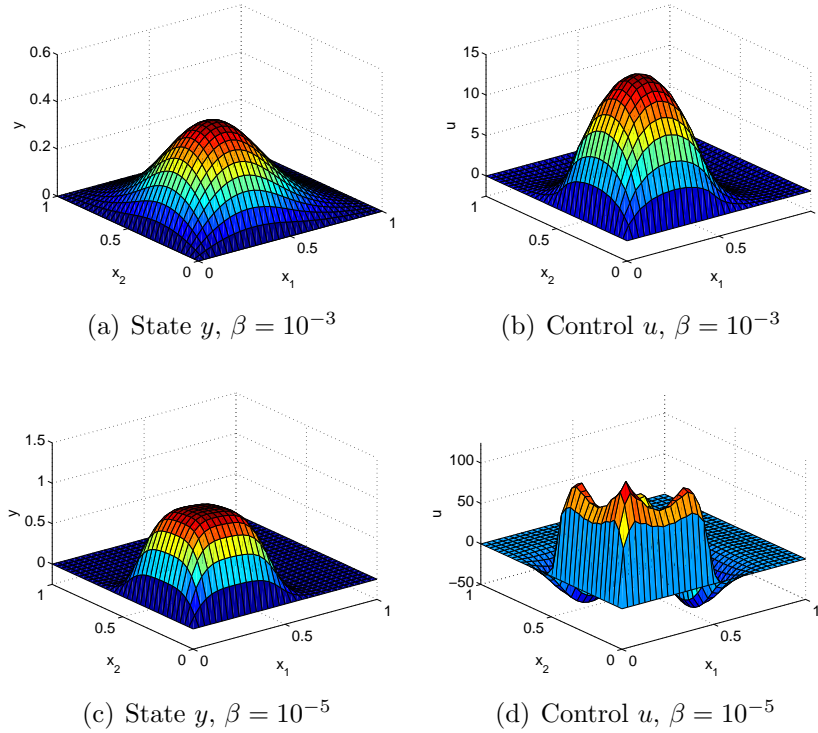


Figure 5: Diagrams of the RBF solutions to the distributed Poisson control Problem 3 for the state y and control u , for $\beta = 10^{-3}$ and 10^{-5} with $c = 250$ and 300 respectively. Here, we took $h = 2^{-5}$.

taking c to be roughly within this interval, although the ideal values may vary if h is changed, for example when considering the problems for which the solution is plotted in Figures 3, 4, 5 and 6, or if β is varied, as in Table 1. We conclude that choosing the value of c appropriately is important when considering our methods for solving Poisson control problems. In Table 3, we display the condition numbers for the matrix in (3.2) for $h = 0.04$ and a variety of values of c and β ; the results demonstrate the ill-conditioning for the problem when using small values of c . The case $h = 0.04$ is an interesting one, as it is a case for which the condition number can be above or below 10^{16} ; if the condition number is above this number, one cannot expect to obtain any digits of accuracy when implementing the method in double precision arithmetic. The conditioning also becomes more of a problem as h is decreased, so it would be desirable to find optimal choices of c for any h and β .

Providing the value of c is chosen appropriately, we can conclude that our methods are powerful ones for solving the Poisson control problems that we have considered. [For further details about Poisson control problems and the features of FEM solutions, see for example [15, 16, 17].] We compare the FEM method and the RBF collocation method we have presented here in Section 6.

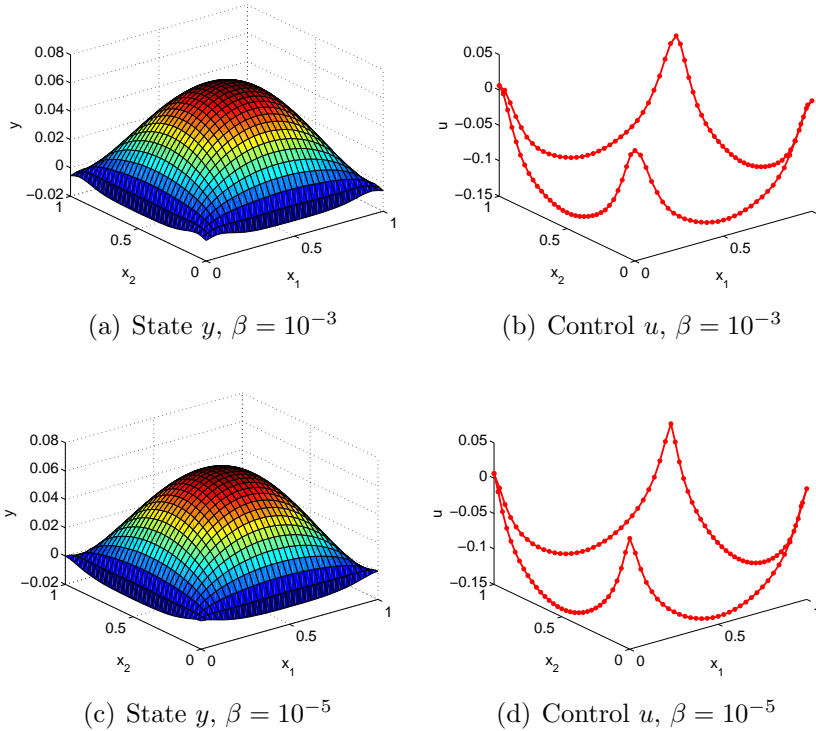


Figure 6: Diagrams of the RBF solutions to the Neumann boundary control Problem 4 for the state y and control u , for $\beta = 10^{-3}$ and 10^{-5} with $c = 500$ in each case. Here, we took $h = 2^{-5}$.

6 Concluding remarks

In this article, we first reviewed some research that has been undertaken to solve Poisson's equation using meshfree methods, and derived systems of equations that are commonly used for Dirichlet and Neumann boundary value problems. We then motivated, derived, implemented and tested a new radial basis function method for solving optimal control problems involving Poisson's equation, and proved a number of solvability results relating to the resulting matrix systems. The problems considered were the distributed control problem with Dirichlet boundary conditions and the Neumann boundary control problem. We found that this method was a powerful one for each of these formulations.

There are clear advantages to using an RBF method for such problems over the more commonly used finite element method. Most notably, it is not necessary to construct a mesh for this method, which saves a considerable amount of computational work and memory. If one did wish to select centres for the RBFs, there would be no restriction on where these centres could be (provided they did not coincide). There are also good direct methods for small matrix systems built into programs such as MATLAB, so if we wished to compute an RBF solution to this problem using a relatively small number of functions, this method would be ideal. In addition, the solutions obtained are infinitely smooth, provided appropriate RBFs are used in the method, so if a high degree of

smoothness were required, this would be a method of choice. However, problems arise when we wish to compute a (highly accurate) solution to the problem using many RBFs, resulting in a large matrix system. As shown in Figure 7, the resulting matrix system for such a meshfree method is extremely dense, which makes the sparse matrix system obtained when an FEM is used preferable, as there have been powerful iterative methods developed for such systems [15, 16, 17]. Computing the mass matrices in (3.2) is also an expensive procedure when creating the global matrix system, although we would not expect the computation time involved to be greater than the time taken to solve the resulting system. Furthermore, ill-conditioning of the matrix system is a significant drawback for large systems in particular. It is possible however that this issue, as well as the denseness of the system, could be resolved by using compactly supported radial basis functions.

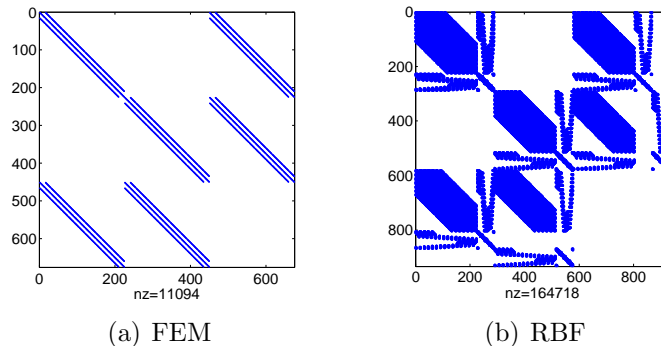


Figure 7: ‘spy’ plots in MATLAB for the matrix system needed to be solved for the distributed Poisson control problem using a **Q1-Q1** mixed FEM, and for the matrix system (excluding entries which have absolute value less than 10^{-15}) using the RBF method of Section 3.1 using Gaussians with $c = 250$ to solve the same problem. In each case $h = 2^{-4}$ and $\beta = 10^{-4}$.

There is much more investigation that could be undertaken based on this work. For example, other types of RBFs could be tested on these problems, in particular compactly supported RBFs. This methodology could also be extended to other optimal control problems: convection-diffusion control, Stokes control and Navier-Stokes control to name but a few. It is also desirable to provide a theoretical framework for this method, proving, for example, numerical stability and error bounds, as well as theoretical results concerning the optimal choices of parameters involved in the method (in particular the optimal choice of c for any given values of h and β to take when basing the method on Gaussians of the form e^{-cr^2}). Furthermore, it would be helpful to find effective solvers to the system of equations that this problem leads to, including efficient preconditioning techniques. Solutions to any of these problems would be highly desirable in order to develop this new and potentially very useful application of radial basis function theory.

Acknowledgements. The author would like to express his gratitude to Holger Wendland for invaluable help and advice while producing this manuscript. He would also like to thank Andy Wathen and Mark Richardson for helpful discussions about this

work. The author acknowledges financial support from the Engineering and Physical Sciences Research Council (EPSRC).

References

- [1] AMINATAEI, A., MAZAREI, M.M.: *Numerical Solution of Poisson's Equation Using Radial Basis Function Networks on the Polar Coordinate*. Computers and Mathematics with Applications 56, 2887–2895 (2008)
- [2] BENZI, M., GOLUB, G.H., LIESEN, J.: *Numerical Solution of Saddle Point Problems*. Acta Numerica 14, 1–137 (2005)
- [3] CHINCHAPATNAM, P.P., DJIDJELI, K., NAIR, P.B.: *Radial Basis Function Meshless Method for the Steady Incompressible Navier-Stokes Equations*. International Journal of Computer Mathematics 84, 1509–1526 (2007)
- [4] DUAN, Y.: *A Note on the Meshless Method Using Radial Basis Functions*. Computers and Mathematics with Applications 55, 66–75 (2008)
- [5] DUAN, Y., TAN, Y.: *A Meshless Galerkin Method for Dirichlet Problems Using Radial Basis Functions*. Journal of Computational and Applied Mathematics 196, 394–401 (2006)
- [6] ELANSARI, M., OUAZAR, D., CHENG, A.H.: *Boundary Solution of Poisson's Equation Using Radial Basis Function Collocated on Gaussian Quadrature Nodes*. Communications in Numerical Methods in Engineering 17, 455–464 (2001)
- [7] FASSHAUER, G.E.: *Meshfree Approximation Methods with MATLAB*. Interdisciplinary Mathematical Series – Vol. 6, World Scientific (2007)
- [8] FRANKE, C., SCHABACK, R.: *Convergence Order Estimates of Meshless Collocation Methods Using Radial Basis Functions*. Advances in Computational Mathematics 8, 381–399 (1998)
- [9] GIESL, P., WENDLAND, H.: *Meshless Collocation: Error Estimates with Application to Dynamical Systems*. SIAM Journal on Numerical Analysis 45, 1723–1741 (2007)
- [10] HON, Y.C., SCHABACK, R.: *On Unsymmetric Collocation by Radial Basis Functions*. Applied Mathematics and Computation 119, 177–186 (2001)
- [11] HUANG, C.-S., YEN, H.-D., CHENG, A.H.-D.: *On the Increasingly Flat Radial Basis Function and Optimal Shape Parameter for the Solution of Elliptic PDEs*. Engineering Analysis with Boundary Elements 34, 802–809 (2010)
- [12] KANSA, E.J.: *Motivation for Using Radial Basis Functions to Solve PDEs*. Technical Report, available at <http://www.rbf-pde.org/kansaweb.pdf> (1999)

- [13] LARSSON, E., FORNBERG, B.: *A Numerical Study of Some Radial Basis Function Based Solution Methods for Elliptic PDEs*. Computers and Mathematics with Applications 46, 891–902 (2003)
- [14] LING, L., KANSA, E.J.: *Preconditioning for Radial Basis Functions with Domain Decomposition Methods*. International Workshop on MeshFree Methods (2003)
- [15] PEARSON, J.W., WATHEN, A.J.: *A New Approximation of the Schur Complement in Preconditioners for PDE Constrained Optimization*. Submitted to Numerical Linear Algebra with Applications (2010)
- [16] REES, T.: *Preconditioning Iterative Methods for PDE Constrained Optimization*. DPhil Thesis, University of Oxford (2010)
- [17] REES, T., DOLLAR, H.S., WATHEN, A.J.: *Optimal solvers for PDE-Constrained Optimization*. SIAM Journal of Scientific Computing 32, 271–298 (2010)
- [18] TRÖLTZSCH, F.: *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*. Graduate Studies in Mathematics Vol. 112, American Mathematical Society (2010)
- [19] WENDLAND, H.: *Meshless Galerkin Methods Using Radial Basis Functions*. Mathematics of Computation 68, 1521–1531 (1999)
- [20] WU, Z.M.: *Hermite-Birkhoff Interpolation of Scattered Data by Radial Basis Functions*. Approximation Theory and its Applications 8, 1–10 (1992)

		PROBLEM 1		
		$\ y - \hat{y}\ _{\ell_2}$	$\ u\ _{\ell_2}$	$\frac{1}{2} \ y - \hat{y}\ _{\ell_2}^2 + \frac{\beta}{2} \ u\ _{\ell_2}^2$
β	1	0.5333	9.2368×10^{-4}	0.1422
	10^{-1}	0.5333	0.0092	0.1422
	10^{-2}	0.5332	0.0923	0.1422
	10^{-3}	0.5318	0.9212	0.1418
	10^{-4}	0.5187	8.9974	0.1385
	10^{-5}	0.4143	72.9537	0.1125
	10^{-6}	0.1218	252.3730	0.0393
	10^{-7}	0.0125	334.8599	0.0057
	10^{-8}	0.0309	346.2321	0.0011

		PROBLEM 2		
		$\ y - \hat{y}\ _{\ell_2}$	$\ u\ _{\ell_2}$	$\frac{1}{2} \ y - \hat{y}\ _{\ell_2}^2 + \frac{\beta}{2} \ u\ _{\ell_2}^2$
β	1	7.1264	0.0100	25.3930
	10^{-1}	7.1263	0.1004	25.3925
	10^{-2}	7.1250	1.0037	25.3878
	10^{-3}	7.1121	10.0161	25.3411
	10^{-4}	6.9860	98.1677	24.8843
	10^{-5}	5.9678	819.9132	21.1687
	10^{-6}	2.9779	3.1969×10^3	9.5439
	10^{-7}	1.9906	4.9581×10^3	3.2105
	10^{-8}	2.0520	5.5134×10^3	2.2573

		PROBLEM 3		
		$\ y - \hat{y}\ _{\ell_2}$	$\ u\ _{\ell_2}$	$\frac{1}{2} \ y - \hat{y}\ _{\ell_2}^2 + \frac{\beta}{2} \ u\ _{\ell_2}^2$
β	1	9.0000	0.0112	40.4999
	10^{-1}	8.9999	0.1124	40.4994
	10^{-2}	8.9986	1.1240	40.4938
	10^{-3}	8.9861	11.2182	40.4377
	10^{-4}	8.8639	110.0162	39.8894
	10^{-5}	7.8915	924.2282	35.4090
	10^{-6}	5.2858	3.7510×10^3	21.0047
	10^{-7}	4.5766	6.3976×10^3	12.5192
	10^{-8}	4.6519	7.4407×10^3	11.0969

		PROBLEM 4		
		$\ y - \hat{y}\ _{\ell_2}$	$\ u\ _{\ell_2}$	$\frac{1}{2} \ y - \hat{y}\ _{\ell_2}^2 + \frac{\beta}{2} \ u\ _{\ell_2}^2$
β	1	0.3018	0.0165	0.0457
	10^{-1}	0.2584	0.1220	0.0341
	10^{-2}	0.1227	0.4762	0.0087
	10^{-3}	0.0546	0.7115	0.0017
	10^{-4}	0.0506	0.7511	0.0013
	10^{-5}	0.0504	0.7553	0.0013
	10^{-6}	0.0504	0.7558	0.0013
	10^{-7}	0.0504	0.7558	0.0013
	10^{-8}	0.0504	0.7558	0.0013

Table 1: Table of solutions to Poisson control problem obtained for Problems 1, 2, 3 and 4 as detailed in Section 3.4, with RBFs evenly distributed with spacing $h = 2^{-4}$. Results are given for a variety of β , and we took $c = 600$ for each Problem except for Problem 4, for which we took $c = 200$.

		PROBLEM 3		
		$\ y - \hat{y}\ _{\ell_2}$	$\ u\ _{\ell_2}$	$\frac{1}{2} \ y - \hat{y}\ _{\ell_2}^2 + \frac{\beta}{2} \ u\ _{\ell_2}^2$
c	5	8.0592	422.4264	41.3978
	10	7.8923	437.0349	40.6940
	20	7.8652	488.1549	42.8456
	50	7.8795	461.8085	41.7069
	100	7.8902	437.9095	40.7158
	200	7.9374	438.3183	41.1073
	300	7.9633	437.8889	41.2945
	400	7.9816	437.6609	41.4300
	500	8.0088	439.1099	41.7113
	1000	11.9402	353.6145	77.5368
	2000	12.9930	33.2258	84.4645
	5000	12.9992	20.4689	84.5109
10000	12.9998	139.2565	85.2565	

Table 2: Table of solutions to Poisson control problem obtained for Problem 3, with RBFs evenly distributed with spacing $h = 0.04$, and $\beta = 10^{-4}$. Results are given for a variety of values of c .

		β				
		1	10^{-2}	10^{-4}	10^{-6}	10^{-8}
c	5	3.8134×10^{22}	1.8294×10^{23}	1.0683×10^{24}	5.9630×10^{24}	1.2993×10^{26}
	10	1.9574×10^{22}	2.5474×10^{23}	9.8843×10^{24}	2.7701×10^{25}	6.6535×10^{27}
	20	5.5563×10^{22}	6.0688×10^{23}	3.1042×10^{25}	1.5761×10^{26}	8.6136×10^{26}
	50	1.1631×10^{23}	3.6307×10^{24}	8.6211×10^{25}	4.1830×10^{27}	7.7395×10^{28}
	100	2.4438×10^{23}	2.8896×10^{24}	5.4541×10^{27}	1.3888×10^{28}	3.2053×10^{28}
	200	2.5909×10^{16}	2.5915×10^{18}	2.5929×10^{20}	2.5610×10^{22}	1.5051×10^{24}
	500	3.0687×10^9	3.0687×10^{11}	3.0685×10^{13}	3.0486×10^{15}	1.8480×10^{17}
	1000	3.7765×10^7	3.7765×10^9	3.7762×10^{11}	3.7511×10^{13}	2.2645×10^{15}
	2000	1.3916×10^7	1.3916×10^9	1.3915×10^{11}	1.3818×10^{13}	8.2187×10^{14}
	5000	1.2839×10^8	1.2839×10^{10}	1.2838×10^{12}	1.2807×10^{14}	1.0300×10^{16}
	10000	1.1905×10^{10}	1.1905×10^{12}	1.1905×10^{14}	1.1898×10^{16}	1.1205×10^{18}

Table 3: Approximate condition number of the matrix in (3.2) for the distributed Poisson control problem on $[0, 1]^2$, with RBFs evenly distributed with spacing $h = 0.04$, for a variety of values of c and β . Approximation is obtained by use of the MATLAB function `condst`.