On the Optimal Shape Parameter for Gaussian Radial Basis Function Finite Difference Approximation of the Poisson Equation

Oleg Davydov* and Dang Thi Oanh†‡

July 26, 2011

Abstract

We investigate the influence of the shape parameter in the meshless Gaussian RBF finite difference method with irregular centres on the quality of the approximation of the Dirichlet problem for the Poisson equation with smooth solution. Numerical experiments show that the optimal shape parameter strongly depends on the problem, but insignificantly on the density of the centres. Therefore, we suggest a multilevel algorithm that effectively finds near-optimal shape parameter, which helps to significantly reduce the error. Comparison to the finite element method and to the generalised finite differences obtained in the flat limits of the Gaussian RBF is provided.

1 Introduction

The quality of the approximation by Gaussian and other infinitely smooth radial basis functions (RBFs) is known to strongly depend on the choice of the shape (or scaling) parameter, see for example [4, Chapter 17] and references therein. In particular, this applies to the RBF-based meshless numerical methods for solving partial differential equations.

In this paper, we investigate the choice of the shape parameter for a generalised finite difference method (RBF-FD) that employs numerical differentiation stencils generated by Gaussian RBF interpolation on irregular centres. The RBF-FD methods are attracting growing attention in the literature, see for example [1, 3, 7, 11, 13, 14, 16]. Even though a theoretical justification for these methods has yet to be developed, the numerical results in the above papers show their exceptional promise. In contrast to the

*Department of Mathematics, University of Strathclyde, 26 Richmond Street, Glasgow G1 1XH, Scotland, oleg.davydov@strath.ac.uk

†Department of Computer Science, Faculty of Information Technology - Thai Nguyen University, Quyet Thang Commune, Thai Nguyen City, Viet Nam, dtoanhtn@gmail.com

‡The second author was supported in part by the National Foundation for Science and Technology Development (NAFOSTED) and a Natural Science Research Project of the Ministry of Education and Training.

more popular weak form based methods, generalised finite differences do not require numerical integration that may be computationally demanding for non-polynomial shape functions on non-standard domains. Moreover, one of their main advantages is high flexibility in the choice of stencil supports, which facilitates the development of adaptive methods [3] and potentially allows to handle problems with singularities in complicated 3D domains without meshing.

We consider the Dirichlet problem for the Poisson equation in 2D with a smooth solution. RBF-FD discretisation is obtained using the centres of several uniformly refined triangulations to allow direct comparison with the finite element method. The stencil supports are obtained by a meshless algorithm suggested in [3], leading to the system matrices with the density of non-zero entries close to the density of the stiffness matrices arising from the finite element method based on linear shape functions on the same triangulations. The RBF stencil weights are obtained by solving local interpolation problems. Because the standard interpolation matrices of the Gaussian RBF $\varphi(r) = e^{-\varepsilon^2 r^2}$ are severely ill-conditioned for small values of the shape parameter ε, special techniques are needed to allow the full range of ε [6, 8, 9, 16]. We rely on the RBF-QR method of [6] adapted to RBF interpolation with a constant term.

Our main goal is to investigate the dependence of the optimal shape parameter $\varepsilon_{\mathrm{opt}}$ on various factors such as the right hand side f of the Poisson equation, the domain, the density of the centres. The numerical experiments suggest that $\varepsilon_{\mathrm{opt}}$ strongly depends on f, but varies only slightly when the domain or density is changed. Based on these observations, we introduce and investigate a multilevel algorithm for the estimation of $\varepsilon_{\mathrm{opt}}$, where the shape parameter on a set of centres Ξ is optimised with respect to the error against a solution on a refined set of centres Ξ^{ref}. Such an algorithm can be practically useful if several refinement levels are available such that the computational cost of the approximate solutions on the coarse levels is negligible comparing to the cost of the final computation on the finest level, where highly optimised shape parameter leads to a significantly more accurate solution. This high accuracy, in addition to the meshless nature of the method, may further justify its practical use despite the relatively high computational cost of the system matrix assembly. As a by-product of our investigation we also observe that the polynomial type generalised finite difference method obtained in the flat limit case $\varepsilon = 0$ is a competitive and rather cheap option, but its results are often significantly worse than those obtained with $\varepsilon = \varepsilon_{\mathrm{opt}}$. Note that the Gibbs and Runge phenomena [5, 10] may be responsible for the sub-optimal behaviour in and close to the flat limit case, although they are not expected for the low order numerical differentiation stencils considered in this paper.

The paper is organised as follows. In Section 2 we describe RBF-FD discretisation methods for the Dirichlet problem. Section 3 is devoted to the QR method of computation of stencils for small ε. In Section 4 we provide the results of the numerical tests on the optimal shape parameter. Section 5 is devoted to our multilevel algorithm for the estimation of the optimal shape parameter. A conclusion and an outlook for the future work are provided in the final Section 6.

2 RBF-FD discretisation of Poisson equation

2.1 Discretisation on irregular centres

Let D be a linear differential operator, and $X = \{x_i\}_{i=1}^n$ a fixed irregular set of centres in $\mathbb{R}^d$. A linear *numerical differentiation formula* for the operator D,

$$Du(x) \approx \sum_{i=1}^{n} w_i(x) u(x_i), \tag{1}$$

is determined by the *weights* $w_i = w_i(x)$. The vector $w = [w_1, \ldots, w_n]^T$ is called *stencil*.

In the *finite difference method* stencils are used for the discretisation of partial differential equations. Consider the Dirichlet problem for the Poisson equation in a bounded domain $\Omega \subset \mathbb{R}^d$: given a function f defined on Ω, and a function g defined on $\partial\Omega$ find u such that

$$\begin{aligned} \Delta u &= f \text{ on } \Omega, & (2)\\ u|_{\partial\Omega} &= g. & (3) \end{aligned}$$

This problem can be discretised with the help of differentiation formulae (1) as follows.

Let $\Xi \subset \overline{\Omega}$ be a finite set of *discretisation centres*, $\partial\Xi := \Xi \cap \partial\Omega$ and $\Xi_{\text{int}} := \Xi \setminus \partial\Xi$. Assume that for each $\zeta \in \Xi_{\text{int}}$ a set $\Xi_\zeta \subset \Xi$ is chosen such that $\zeta \in \Xi_\zeta$ and

$$\Xi = \bigcup_{\zeta \in \Xi_{\text{int}}} \Xi_\zeta. \tag{4}$$

For each $\zeta \in \Xi_{\text{int}}$, choose a linear numerical differentiation formula for Laplace operator Δ,

$$\Delta u(\zeta) \approx \sum_{\xi \in \Xi_\zeta} w_{\zeta,\xi} u(\xi), \tag{5}$$

with stencil $[w_{\zeta,\xi}]_{\xi \in \Xi_\zeta}$, and replace (2)–(3) by the system of linear equations

$$\begin{aligned} \sum_{\xi \in \Xi_\zeta} w_{\zeta,\xi} \hat{u}(\xi) &= f(\zeta), \quad \zeta \in \Xi_{\text{int}}, & (6)\\ \hat{u}(\xi) &= g(\xi), \quad \xi \in \partial\Xi. & (7) \end{aligned}$$

If (6)–(7) is nonsingular, then its solution $\hat{u} : \Xi \to \mathbb{R}$ can be compared with the vector $u_{|\Xi} = [u(\xi)]_{\xi \in \Xi}$ of the discretised exact solution of (2)–(3).

A standard finite difference method is obtained from the above if we take $\Omega \subset \mathbb{R}^2$ to be a square domain, Ξ a uniformly spaced grid, and (5) the classical 5-point differentiation formula for the Laplacian.

The performance of meshless methods heavily depends on how the local sets Ξ_ζ (*stencil supports*) are chosen for each $\zeta \in \Xi_{\text{int}}$. There are many algorithms in the literature for choosing Ξ_ζ, see [3, Section 5] for an overview. In the numerical results of this paper we make use of the method described in [3, Algorithm 1].

2.2 RBF-FD method

Let $\Xi_\zeta = \{x_0, \ldots, x_n\} \subset \mathbb{R}^d$, $x_0 = \zeta$, be a local set of discretisation centres, as in Section 2.1. Given a positive definite function $\varphi : \mathbb{R}_+ \to \mathbb{R}$ and a continuous function $u : \mathbb{R}^d \to \mathbb{R}$, the *RBF interpolant with a constant term* [2, 4, 15] is sought in the form

$$s(x) = \sum_{j=0}^{n} a_j \varphi_j(x) + c, \qquad \varphi_j(x) = \Phi(x - x_j), \quad \Phi(x) := \varphi(\|x\|), \tag{8}$$

where $\|x\|$ is the Euclidean norm of x, and the coefficients a_j and c are chosen such that

$$s(x_i) = u(x_i), \quad i = 0, \ldots, n, \qquad \sum_{j=0}^{n} a_j = 0. \tag{9}$$

Thus, the coefficients are uniquely determined as the solution of the linear system

$$\sum_{j=0}^{n} a_j \Phi(x_i - x_j) + c = u(x_i), \quad i = 0, \ldots, n, \qquad \sum_{j=0}^{n} a_j = 0, \tag{10}$$

written in matrix form as

$$\begin{bmatrix} \Phi_X & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} a \\ c \end{bmatrix} = \begin{bmatrix} u_{|X} \\ 0 \end{bmatrix}, \qquad \Phi_X := [\Phi(x_i - x_j)]_{i,j=0}^{n}, \quad \mathbf{1} := [1 \ \cdots \ 1]^T.$$

The matrix Φ_X is symmetric and positive definite for any set X.

The interpolant s provides a good approximation of u at x if the function u is sufficiently smooth and the set of points $x_0, \ldots, x_n \in \mathbb{R}^d$ is sufficiently dense in a neighbourhood of x. Moreover, the derivatives of s are good approximations of the derivatives of u if φ is sufficiently smooth [15].

According to (5), we need to numerically differentiate the Laplacian of u at x_0. In *RBF-FD method*, an approximation of $\Delta u(x_0)$ is considered in the form

$$\Delta u(x_0) \approx \Delta s(x_0) = \sum_{j=0}^{n} a_j \Delta\varphi_j(x_0) = \sum_{i=0}^{n} w_i u(x_i), \tag{11}$$

where the constant c not present as it is annihilated by the Laplace operator. It is easy to see that the weights w_i can be found by solving the RBF interpolation problem (8)–(9) with the data given by $\Delta\varphi_i(x_0)$, $i = 1, \ldots, n$,

$$\begin{bmatrix} \Phi_X & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} = \begin{bmatrix} [\Delta\varphi_i(x_0)]_{i=0}^{n} \\ 0 \end{bmatrix}. \tag{12}$$

Indeed, if the weight vector w satisfies (12), then

$$\begin{aligned} \sum_{j=0}^{n} a_j \Delta\varphi_j(x_0) &= \begin{bmatrix} a \\ c \end{bmatrix}^T \begin{bmatrix} [\Delta\varphi_i(x_0)]_{i=0}^{n} \\ 0 \end{bmatrix} = \begin{bmatrix} a \\ c \end{bmatrix}^T \begin{bmatrix} \Phi_X & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} \\ &= \begin{bmatrix} u_{|X} \\ 0 \end{bmatrix}^T \begin{bmatrix} w \\ v \end{bmatrix} = \sum_{i=0}^{n} w_i u(x_i). \end{aligned}$$

In this paper we restrict our attention to the Gaussian RBF $\varphi(r) = e^{-(\varepsilon r)^2}$, which is positive definite for any value of the *shape parameter* $\varepsilon > 0$. For this function, the matrix Φ_X takes the form

$$\Phi_X = [e^{-\varepsilon^2\|x_i - x_j\|^2}]_{i,j=0}^n. \tag{13}$$

The Laplacians of the shifts of the Gaussian function $\Phi(x) = e^{-\varepsilon^2\|x\|^2}$ needed in (12) are given by

$$\Delta\varphi_i(x) = 2\varepsilon^2 e^{-\varepsilon^2\|x-x_i\|^2}(2\varepsilon^2\|x - x_i\|^2 - d). \tag{14}$$

Note that both the constant term c and the side condition $\sum_{j=0}^n a_j = 0$ can be removed in (8)–(9) because Φ_X is nonsingular for Gaussian. Respectively, in the linear system (12) the coefficient v and the last equation can be removed, leading to a simpler linear system $\Phi_X w = [\Delta\varphi_i(x_0)]_{i=0}^n$. However, in general, stencils obtained this way do not satisfy the highly desirable property $\sum_{j=0}^n w_j = 0$. and therefore we prefer to use RBF interpolants with a constant term.

Instead of the stencils derived by (12), discretisations of the Dirichlet problem may be obtained with the help of numerical differentiation of certain linear combinations of Laplacians, leading to multipoint RBF stencils considered in [3]. Another alternative is provided by the Hermite RBF stencils introduced in [16].

3 Stable computation for small ε

Since the matrix (13) is extremely ill-conditioned for small ε, alternative approaches for solving (12) are needed in this case. Several methods are available, see [6] and references therein. We follow the RBF-QR method of [6], and adapt it to the case of RBF interpolation with a constant term.

3.1 Polar-Chebyshev functions and their Laplacians

Following [6], we consider the following polar-Chebyshev expansion of $\varphi_k(x) = \varphi(\|x - x_k\|) = e^{-(c\|x-x_k\|)^2}$ with both x and x_k in the unit disk in 2D,

$$\varphi_k(x) = \sum_{j=0}^{\infty}\sum_{m=0}^{\lfloor j/2\rfloor} d_{j,m}c_{j,m}(x_k)T_{j,m}^c(x) + \sum_{j=0}^{\infty}\sum_{m=1-p}^{\lfloor j/2\rfloor} d_{j,m}s_{j,m}(x_k)T_{j,m}^s(x) \tag{15}$$

where $p = 0$ if j is even and $p = 1$ if j is odd. Here, the *polar-Chebyshev functions* $T_{j,m}^c(x), T_{j,m}^s(x)$ are defined in polar coordinates $x = r\cos\theta$, $y = r\sin\theta$ on the unit disk $-1 \le r \le 1$, $0 \le \theta < \pi$, by

$$\begin{aligned} T_{j,m}^c(x) &= T_{j,m}(r)\cos((2m+p)\theta), \\ T_{j,m}^s(x) &= T_{j,m}(r)\sin((2m+p)\theta), \quad 2m+p \neq 0, \end{aligned} \tag{16}$$

where

$$T_{j,m}(r) = e^{-\varepsilon^2 r^2} r^{2m} T_{j-2m}(r),$$

and $T_n(r) = \cos(n \arccos r)$ denotes the Chebyshev polynomial of degree n. The coefficients $d_{j,m}$, $c_{j,m}(x_k)$ and $s_{j,m}(x_k)$ are given by

$$d_{j,m} = \frac{\varepsilon^{2j}}{2^{j-2m-1}(\frac{j+2m+p}{2})!(\frac{j-2m-p}{2})!} = \frac{\varepsilon^{2j}}{2^{j-2m-1}(\lfloor\frac{j+1}{2}\rfloor + m)!(\lfloor\frac{j}{2}\rfloor - m)!} \tag{17}$$

and

$$\begin{aligned} c_{j,m}(x_k) &= \mu_{j,m} e^{-\varepsilon^2 r_k^2} r_k^j \cos((2m+p)\theta_k)\, {}_1F_2(\alpha, \beta_1, \beta_2, \varepsilon^4 r_k^2), \\ s_{j,m}(x_k) &= \mu_{j,m} e^{-\varepsilon^2 r_k^2} r_k^j \sin((2m+p)\theta_k)\, {}_1F_2(\alpha, \beta_1, \beta_2, \varepsilon^4 r_k^2), \\ \alpha &= \tfrac{j-2m+p+1}{2}, \quad \beta_1 = j-2m+1, \quad \beta_2 = \tfrac{j+2m+p+2}{2}, \end{aligned}$$

where $x_k = r_k \cos\theta_k$, $y_k = r_k \sin\theta_k$,

$$\mu_{j,m} = \begin{cases} 1/2 & \text{if } j = m = 0, \\ 1 & \text{if } j \geq 2 \text{ even and } m = 0 \text{ or } m = j/2, \\ 2 & \text{otherwise,} \end{cases}$$

and ${}_1F_2$ is the hypergeometric function given by the series

$${}_1F_2(\alpha, \beta_1, \beta_2, \varepsilon^4 r_k^2) = \sum_{\ell=0}^{\infty} \varepsilon^{4\ell} r_k^{2\ell} \prod_{q=0}^{\ell-1} \frac{\alpha+q}{(\beta_1+q)(\beta_2+q)}.$$

The series (15) is convergent for any fixed $\varepsilon \geq 0$ because $c_{j,m}(x_k)$ and $s_{j,m}(x_k)$ are $\mathcal{O}(1)$ as soon as x_k lies in the unit disk, and the scaling coefficients $d_{j,m}$ decay superexponentially as $j \to \infty$. A justification of the above formulas as well as further details on their practical implementation can be found in [6].

For the computation of numerical differentiation stencils, we will also need to evaluate the Laplacians of the functions (16). If $r \neq 0$, then we can use the formula $\Delta u = u_{rr} + \frac{1}{r}u_r + \frac{1}{r^2}u_{\theta\theta}$ to compute these Laplacians in polar coordinates as

$$\begin{aligned} \Delta T^c_{j,m}(x) &= V_{j,m}(r)\cos((2m+p)\theta), \\ \Delta T^s_{j,m}(x) &= V_{j,m}(r)\sin((2m+p)\theta), \quad 2m+p \neq 0, \end{aligned} \tag{18}$$

where

$$V_{j,m}(r) = T''_{j,m}(r) + \frac{1}{r}T'_{j,m}(r) - \frac{(2m+p)^2}{r^2}T_{j,m}(r).$$

Assuming that $0 < |r| < 1$, we can rewrite this expression in terms of the Chebyshev polynomial $T_{j-2m}(r) = \cos((j-2m)\arccos r)$ and its derivative $T'_{j-2m}(r) = \frac{j-2m}{\sqrt{1-r^2}}\sin((j-2m)\arccos r)$,

$$\begin{aligned} V_{j,m}(r) = e^{-\varepsilon^2 r^2} r^{2m-2} \Big\{ \Big(4(m-\varepsilon^2 r^2)^2 - 4\varepsilon^2 r^2 - \frac{(j-2m)^2 r^2}{1-r^2} - (2m+p)^2 \Big) T_{j-2m}(r) \\ + \Big(4(m-\varepsilon^2 r^2) + \frac{1}{1-r^2} \Big) r T'_{j-2m}(r) \Big\}, \quad 0 < |r| < 1. \end{aligned} \tag{19}$$

If $m \geq 1$, then (19) can also be used when $r = 0$, and it follows that $V_{j,m}(0) = 0$ in this case. Indeed, $V_{j,m}$ has a positive power of r as a factor if $m \geq 2$, and $V_{j,1} = -5pT_{j-2}(0) = 0$ since $T_{j-2}(0) = 0$ when j is odd and $p = 0$ when j is even.

For $m = 0$ we obtain from (19) for $0 < |r| < 1$,

$$V_{j,0}(r) = e^{-\varepsilon^2 r^2}\Big\{\Big(4\varepsilon^4 r^2 - 4\varepsilon^2 - \frac{j^2}{1-r^2} - \frac{p}{r^2}\Big)T_j(r) + \Big(4\varepsilon^2 r + \frac{r}{1-r^2} + \frac{1}{r}\Big)T_j'(r)\Big\},$$

which includes a potentially singular at $r = 0$ term $\frac{1}{r}T_j'(r) - \frac{p}{r^2}T_j(r)$. However,

$$\frac{T_j'(r)}{r} - \frac{pT_j(r)}{r^2} = \begin{cases} T_j'(r)/r & \text{if } j \text{ is even,} \\ (T_j(r)/r)', & \text{if } j \text{ is odd,} \end{cases}$$

which is the zero function if $j = 0$ or 1, and a polynomial of degree $j - 2$ for $j \geq 2$. This polynomial is odd if j is odd, and hence vanishes at $r = 0$. For an even $j = 2k$ it is not difficult to calculate that $\lim_{r\to 0} T_j'(r)/r = (-1)^{k+1}j^2$. Since $T_{2k}(0) = (-1)^k$, we conclude that $V_{j,m}(0) = 2(-1)^{k+1}(2\varepsilon^2 + j^2)$ if $m = 0$ and $j = 2k$ is even, and $V_{j,m}(0) = 0$ in all other cases. Therefore

$$\Delta T^c_{j,m}(0) = \begin{cases} 2(-1)^{k+1}(2\varepsilon^2 + j^2), & \text{if } m = 0 \text{ and } j = 2k \text{ even,} \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

$$\Delta T^s_{j,m}(0) = 0 \quad \text{in all cases.} \tag{21}$$

3.2 Gauss-QR basis functions

Given a set of points $\{x_0, \ldots, x_n\}$ in the unit disk, the interpolant (8) is a linear combination of the functions $\{\varphi_0, \ldots, \varphi_n\}$. Due to its ill-conditioning, the basis $\{\varphi_k\}_{k=0}^n$ is not suitable for the computation of the interpolant s if ε is small. Therefore, this basis has to be preconditioned. In the QR method of [6] a new basis $\{\psi_k\}_{k=0}^n$ is obtained as follows. In view of (15),

$$\begin{bmatrix} \varphi_0(x) \\ \vdots \\ \varphi_n(x) \end{bmatrix} = C \cdot D \cdot \begin{bmatrix} T^c_{0,0}(x) \\ T^c_{1,0}(x) \\ T^s_{1,0}(x) \\ T^c_{2,0}(x) \\ T^c_{2,1}(x) \\ T^s_{2,1}(x) \\ \vdots \end{bmatrix} = CD\,T(x),$$

where D is an infinite diagonal matrix containing the scaling coefficients $d_{j,m}$, and C is a semi-infinite matrix consisting of the coefficients $c_{j,m}(x_k)$ and $s_{j,m}(x_k)$ in the k-th row. Let C_1 be the $(n+1) \times (n+1)$ matrix consisting of the first $n+1$ columns of C, and let $C_1 = QR_1$ be its QR factorisation, where Q is an orthogonal and R_1 an upper triangular $(n+1) \times (n+1)$ matrix. Furthermore, let D_1 be the main minor of D of order n. Assuming that C_1 (or, equivalently, R_1) is non-singular, we set

$$\begin{bmatrix} \psi_0 \\ \vdots \\ \psi_n \end{bmatrix} = D_1^{-1}R_1^{-1}Q^T \begin{bmatrix} \varphi_0 \\ \vdots \\ \varphi_n \end{bmatrix}. \tag{22}$$

The basis $\{\psi_k\}_{k=0}^n$ performs remarkably well in numerical tests provided in [6] and in this paper. However, its numerical implementation does not use (22) directly, which would be numerically unstable for small ε. Instead, it is based on a truncation of the expansion (15) and subsequent cancellation of the powers of ε, as explained below. The truncation point $j_{\max}$ is determined such that

$$\frac{\max\limits_{i>M} D_{ii}}{\min\limits_{1\le i\le n+1} D_{ii}} \quad \text{is less than unit round-off,}$$

where $M = \frac{1}{2}(j_{\max}+1)(j_{\max}+2)$ is the number of terms in (15) with $0 \le j \le j_{\max}$. As a result of the truncation, the functions $\varphi_0, \ldots, \varphi_n$ are replaced by numerically identical functions $\tilde{\varphi}_0, \ldots, \tilde{\varphi}_n$ given by

$$\begin{bmatrix} \tilde{\varphi}_0(x) \\ \vdots \\ \tilde{\varphi}_n(x) \end{bmatrix} = \tilde{C}\tilde{D}\,\tilde{T}(x),$$

where $\tilde{C}$ consists of the first M columns of C, $\tilde{D}$ is the main minor of D of order M, and $\tilde{T}(x)$ consists of the first M components of $T(x)$.

Let

$$\tilde{D} = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}, \qquad \tilde{C} = \begin{bmatrix} C_1 & C_2 \end{bmatrix} = Q \cdot \begin{bmatrix} R_1 & R_2 \end{bmatrix}, \quad R_2 := Q^T C_2,$$

so that

$$\tilde{C}\tilde{D} = Q \cdot \begin{bmatrix} R_1 D_1 & R_2 D_2 \end{bmatrix}.$$

The functions $\psi_0, \ldots, \psi_n$ are numerically implemented as $\tilde{\psi}_0, \ldots, \tilde{\psi}_n$ given by

$$\begin{bmatrix} \tilde{\psi}_0(x) \\ \vdots \\ \tilde{\psi}_n(x) \end{bmatrix} = D_1^{-1} R_1^{-1} Q^T \begin{bmatrix} \tilde{\varphi}_0(x) \\ \vdots \\ \tilde{\varphi}_n(x) \end{bmatrix} = \begin{bmatrix} I & \tilde{R} \end{bmatrix} \cdot \tilde{T}(x), \tag{23}$$

where $\tilde{R}$ is defined by

$$\tilde{R} := D_1^{-1} R_1^{-1} R_2 D_2.$$

Moreover, $\tilde{R}$ is computed in a numerically stable way as explained below. Note that each $\tilde{\psi}_k$ is the sum of a polar-Chebyshev function $T^c_{j,m}(x)$ or $T^s_{j,m}(x)$ from the first n components of $\tilde{T}(x)$ plus a linear combination of components $n+2$ to M.

To ensure numerical stability, the matrix $\tilde{R}$ is computed as follows. First compute $A = R_1^{-1} R_2$. Then the product $D_1^{-1} A D_2$ is obtained by simultaneous scaling of the rows of A by the diagonal entries of D_1^{-1} and scaling of its columns by the diagonal entries of D_2. Therefore, a typical entry $r_{\mu\nu}$ of $\tilde{R}$ has the form $r_{\mu\nu} = d^{-1}_{j_1,m_1} d_{j_2,m_2} a_{\mu\nu}$, where $a_{\mu\nu}$ is the corresponding entry of A, and d_{j_1,m_1}, d_{j_2,m_2} are diagonal elements of D_1 and D_2, respectively. This implies that $j_1 \le j_2$, and hence by (17), $r_{\mu\nu}$ is given by

$$r_{\mu\nu} = \frac{\varepsilon^{2(j_2-j_1)} (\lfloor \frac{j_2+1}{2} \rfloor + m_2)! (\lfloor \frac{j_2}{2} \rfloor - m_2)!}{2^{j_2-j_1-2(m_2-m_1)} (\lfloor \frac{j_1+1}{2} \rfloor + m_1)! (\lfloor \frac{j_1}{2} \rfloor - m_1)!} a_{\mu\nu},$$

which does not include negative powers of ε, thus allowing stable computation for any $\varepsilon > 0$, and even for $\varepsilon = 0$ in the flat limit case, see [6].

3.3 Computation of Gaussian RBF-FD stencils by QR method

If ε is small, then the matrix of the linear system (12) is ill-conditioned. We now explain how Gauss-QR basis functions can be used to precondition this system. Since

$$\Phi_X = \begin{bmatrix} \varphi_0(x_0) & \cdots & \varphi_0(x_n) \\ \vdots & & \vdots \\ \varphi_n(x_0) & \cdots & \varphi_n(x_n) \end{bmatrix},$$

we have

$$\Psi_X := \begin{bmatrix} \psi_0(x_0) & \cdots & \psi_0(x_n) \\ \vdots & & \vdots \\ \psi_n(x_0) & \cdots & \psi_n(x_n) \end{bmatrix} = D_1^{-1} R_1^{-1} Q^T \Phi_X.$$

By left-multiplying both sides of (12) by

$$\begin{bmatrix} D_1^{-1} R_1^{-1} Q^T & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix},$$

where $\mathbf{0}$ denotes the zero matrices, we obtain the following preconditioned linear system

$$\begin{bmatrix} \Psi_X & h \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} = \begin{bmatrix} [\Delta\psi_i(x_0)]_{i=0}^n \\ 0 \end{bmatrix}, \qquad h := D_1^{-1} R_1^{-1} Q^T \mathbf{1},$$

which is replaced in the numerical implementation by

$$\begin{bmatrix} \widetilde{\Psi}_X & h \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} = \begin{bmatrix} [\Delta\tilde{\psi}_i(x_0)]_{i=0}^n \\ 0 \end{bmatrix}, \tag{24}$$

with

$$\widetilde{\Psi}_X := \begin{bmatrix} \tilde{\psi}_0(x_0) & \cdots & \tilde{\psi}_0(x_n) \\ \vdots & & \vdots \\ \tilde{\psi}_n(x_0) & \cdots & \tilde{\psi}_n(x_n) \end{bmatrix}.$$

The entries of $\widetilde{\Psi}_X$ are computed with the help of the polar-Chebyshev functions as in (23). However, for small ε, the vector h cannot be computed directly because D_1^{-1} involves negative powers of ε, which excessively magnifies the rounding errors. The following regularisation approach performs well in our experiments. We first replace h by $\tilde{h} := \tilde{D} R_1^{-1} Q^T$, where $\tilde{D}$ is obtained from D_1^{-1} by replacing by zero all entries that exceed the reciprocal of the unit round-off. By solving for w_0 in the last row of (24) and substituting w_0 elsewhere, we obtain the following linear system,

$$\sum_{j=1}^{n} (\tilde{\psi}_i(x_j) - \tilde{\psi}_i(x_0)) w_j + \tilde{h}_i v = \Delta\tilde{\psi}_i(x_0), \qquad i = 0, \ldots, n. \tag{25}$$

After normalising the last column of the matrix of this system with respect to the maximum norm, and then normalising the first row, we solve it for $w_1, \ldots, w_n$ and $\tilde{v} := v / \|\tilde{h}\|_\infty$, and finally compute w_0 as

$$w_0 = -\sum_{j=1}^{n} w_j.$$

The condition number of the normalised matrix of (25) was comparable to the condition number of $\widetilde{\Psi}_X$ in all numerical tests, which shows that this method successfully eliminates the problem arising from the bad scaling of h. It is easy to see that the above method is also applicable when $\varepsilon = 0$.

In view of (23), the values $\tilde{\psi}_i(x_j)$ are obtained by evaluating the polar-Chebyshev functions $T^c_{j,m}, T^s_{j,m}$, and $\Delta\tilde{\psi}_i(x_0)$ require $\Delta T^c_{j,m}(x_0)$ and $\Delta T^s_{j,m}(x_0)$ according to

$$[\Delta\tilde{\psi}_i(x_0)]_{i=0}^n = \left[\begin{array}{cc} I & \tilde{R} \end{array}\right] \cdot \Delta\tilde{T}(x_0).$$

To use the polar-Chebyshev expansion (15) we need to ensure that the centres $x_0, \ldots, x_n$ are in the unit disk. Moreover, the Laplacians of $T^c_{j,m}, T^s_{j,m}$ are faster to evaluate at the origin by using (20)–(21) than at any other points by (18)–(19). Therefore, in general, if the centres are not located in the unit disk, a linear transform of the independent variables, for example

$$y = (x - x_0)/\mu, \qquad \mu := 2 \max_{1\le i\le n} \|x_i - x_0\|,$$

is needed before applying the QR method. The new centres

$$y_j = (x_j - x_0)/\mu, \quad j = 0, \ldots, n,$$

are in the unit disk, and $y_0 = 0$. It is easy to see that the Gaussian numerical differentiation stencil w of (12), with the shape parameter ε, satisfies $w = \tilde{w}/\mu^2$, where $\tilde{w}$ is obtained by solving

$$\left[\begin{array}{cc} \Phi_Y & \mathbf{1} \\ \mathbf{1}^T & 0 \end{array}\right] \left[\begin{array}{c} \tilde{w} \\ \tilde{v} \end{array}\right] = \left[\begin{array}{c} [\Delta\varphi_i(0)]_{i=0}^n \\ 0 \end{array}\right], \tag{26}$$

with the shape parameter set to $\mu\varepsilon$. Hence, w can be computed by applying the QR method to (26) and then rescaling the stencil $\tilde{w}$.

Since the functions $\psi_0, \ldots \psi_n$ can only be generated by (22) if the matrix R_1 is non-singular, we always compute the condition number of R_1, and only proceed if it does not exceed a tolerance value of 10^{12}. However, in the numerical experiments described below in Sections 4 and 5 this value has never been exceeded. If R_1 is singular or ill-conditioned, then the QR factorisation $C_1 = QR_1$ has to be adjusted by using selective column pivoting as described in [6].

4 Optimal shape parameter

In this section we investigate numerically the performance of the Gaussian RBF-FD method depending on the choice of the shape parameter ε.

4.1 Test Problems

We consider the Dirichlet problem (2)-(3) on four domains listed below, with the right hand sides given by the functions f_1–f_8 and boundary conditions defined by the restriction of the corresponding exact solutions u_1–u_8, see Table 1. For each domain

Ω we consider five sets of discretisation centres $\Xi = \Xi^{(1)}, \ldots, \Xi^{(5)}$ generated as follows. First, an initial triangulation $\mathcal{T}^{(1)}$ is computed using MATLAB PDE Toolbox [12] with default mesh generation parameters. This triangulation is uniformly refined four times, which produced the triangulations $\mathcal{T}^{(2)}, \ldots, \mathcal{T}^{(5)}$. The sets of discretisation centres $\Xi^{(1)}, \ldots, \Xi^{(5)}$ consist of all vertices of corresponding triangulations. The number of interior centres for each $\Xi^{(i)}$ is shown in Table 2.

Domains: (a) the square $(-1, 1)^2$, (b) the unit disk $r < 1$, (c) the unit disk with a square hole $(-0.4, 0.4)^2$, and (d) a polygonal domain shown in Figure 3 (right). Some of the triangulations are illustrated in Figures 1–3.

exact solution	right hand side
$u_1(x,y) = \sin(\pi x)\sin(\pi y)$	$f_1(x,y) = -2\pi^2 \sin(\pi x)\sin(\pi y)$
$u_2(x,y) = e^{-(x-0.1)^2-0.5y^2}$	$f_2(x,y) = e^{-(x-0.1)^2-0.5y^2}(y^2 + (-2x+0.2)^2 - 3)$
$u_3(x,y) = e^x \cos y$	$f_3(x,y) = 0$
$u_4(r,\phi) = r^2(r-1)\sin(2\phi)$	$f_4(r,\phi) = 5r\sin(2\phi)$
$u_5(x,y) = \sin(2xy)$	$f_5(x,y) = -4\sin(2xy)(x^2+y^2)$
$u_6(x,y) = \sin(2\pi(x-y))$	$f_6(x,y) = -8\pi^2 \sin(2\pi(x-y))$
$u_7(x,y) = \sin(x^3y) + e^x - x/(1+y^2)$	$f_7(x,y) = -9\sin(x^3y)x^4y^2 + 6\cos(x^3y)xy + e^x - \sin(x^3y)x^6 - \frac{8xy^2}{(1+y^2)^3} + \frac{2x}{(1+y^2)^2}$
$u_8(x,y) = \frac{1}{2}u_1(x,y) + u_2(x,y)$	$f_8(x,y) = \frac{1}{2}f_1(x,y) + f_2(x,y)$

Table 1: Test functions $u_1, \ldots, u_8$ (exact solutions of the test problems) and their Laplacians (right hand sides for the test problems) $f_i = \Delta u_i$, $i = 1, \ldots, 8$. The functions u_4 and f_4 are given in polar coordinates.

	square	disk	disk with hole	polygon
$\Xi^{(1)}$	33	28	15	15
$\Xi^{(2)}$	149	125	90	83
$\Xi^{(3)}$	633	529	420	381
$\Xi^{(4)}$	2609	2177	1800	1625
$\Xi^{(5)}$	10593	8833	7440	6705

Table 2: Number of interior centres for each discretisation.

4.2 Numerical experiments

To assess the quality of a discrete solution $\hat{u}$ of the Dirichlet problem, defined on a set of discretisation centres $\Xi = \Xi^{(i)}$, we consider its root mean square (rms) error against

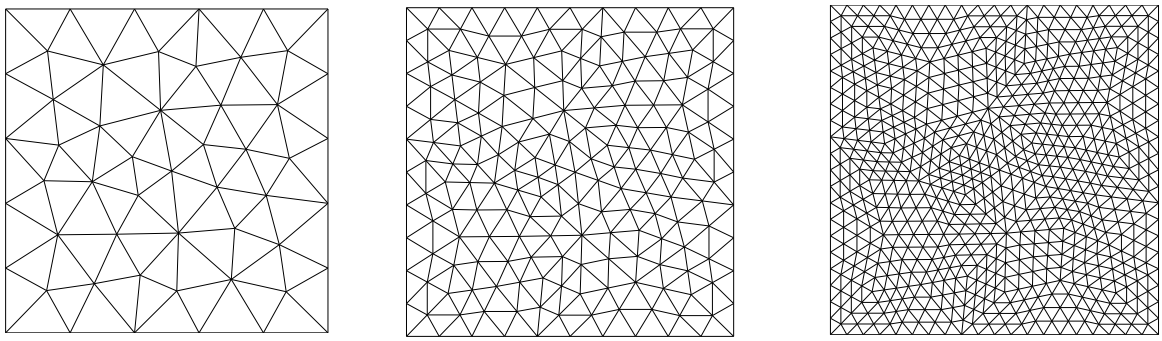


Figure 1: Initial triangulation $\mathcal{T}^{(1)}$ and its two uniform refinements $\mathcal{T}^{(2)}, \mathcal{T}^{(3)}$ for the square domain. The sets of discretisation centres $\Xi^{(i)}$ are given by the vertices of the respective triangulations.

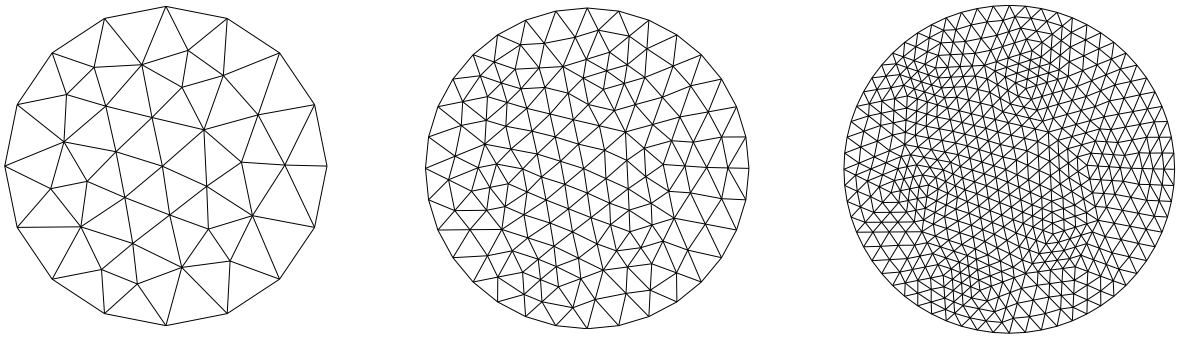


Figure 2: Initial triangulation $\mathcal{T}^{(1)}$ and its two uniform refinements $\mathcal{T}^{(2)}, \mathcal{T}^{(3)}$ for the disk.

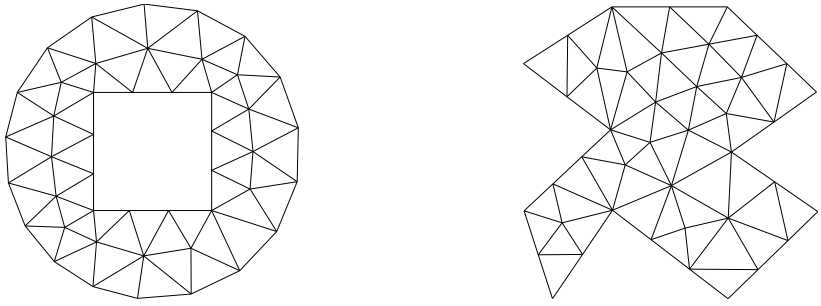


Figure 3: Initial triangulation $\mathcal{T}^{(1)}$ for the disk with a square hole and for the polygonal domain.

the values of the exact solution on Ξ_{int},

$$\texttt{rmse} := \Big(\frac{1}{\#\Xi_{\mathrm{int}}} \sum_{\xi\in\Xi_{\mathrm{int}}} \big(\hat{u}(\xi) - u(\xi)\big)^2\Big)^{1/2}. \tag{27}$$

Apart from the RBF-FD solutions, this formula applies to the standard linear finite element method with midpoint quadrature rule on the corresponding triangulation $\mathcal{T}^{(i)}$. We will use `rmse` of the finite element method as reference. For the RBF-FD method, we consider in addition the rms error of the numerical differentiation formula (5), given by

$$\texttt{rmsed} := \Big(\frac{1}{\#\Xi_{\mathrm{int}}} \sum_{\zeta\in\Xi_{\mathrm{int}}} r_\zeta^2\Big)^{1/2}, \quad r_\zeta = \Delta u(\zeta) - \sum_{\xi\in\Xi_\zeta} w_{\zeta,\xi} u(\xi). \tag{28}$$

For each $\zeta \in \Xi_{\mathrm{int}}$, we select the stencil supports Ξ_ζ by a meshless algorithm described in [3, Algorithm 1] with the target size of $\Xi_\zeta \setminus \{\zeta\}$ set to 6. This leads to Ξ_ζ consisting of either 7 or 6 points, depending on the local geometric constellation of Ξ around ζ. Since the triangulations $\mathcal{T}^{(i)}$ are quasi-uniform, Ξ_ζ obtained by this method are only rarely different from the set of vertices of all triangles sharing ζ as a vertex, that is the stencil supports of the linear finite element method. We do not provide matrix density plots similar to those in [3, Figure 11b] because the curves for the finite element stencil supports on one hand and meshless stencil supports on the other hand are not distinguishable on the triangulations considered in this paper. Therefore the comparison of the errors to those obtained by the linear finite element method is fair. In fact, from our experience, using finite element stencil supports leads to results very similar to the ones described below, but we prefer to use a meshless method for choosing Ξ_ζ.

Since the direct method of calculation of Gaussian RBF-FD stencils by solving (12) fails for small ε because of ill-conditioning, and because Gauss-QR method is more expensive for large ε, we choose for each set $\Xi^{(i)}$ a 'safe' value $\varepsilon_{\mathrm{dmin}}$ that guarantees that the condition number of the matrix of the system (12) does not exceed 10^{12} for any local set Ξ_ζ if $\varepsilon \geq \varepsilon_{\mathrm{dmin}}$. The values of $\varepsilon_{\mathrm{dmin}}$ are given in Table 3. In the experiments in this paper we always use the RBF-FD method directly if $\varepsilon \geq \varepsilon_{\mathrm{dmin}}$, and we use QR method if $\varepsilon < \varepsilon_{\mathrm{dmin}}$. To compute the Gauss-QR stencils we have adapted the MATLAB code provided in [6] and available for download from `http://user.it.uu.se/~bette/research.html`

Figures 4 and 5 and Tables 4 and 5 present the results for the test function u_1 on all domains and sets of centres. In particular, Figure 5 compares the optimal rms error of Gaussian RBF-FD with the error of the finite element method, the error obtained if choosing the 'safe' shape parameter, and the error in the 'flat limit' case of $\varepsilon = 0$. Further results, for the test functions u_2–u_8, are presented in Figures 6 and 7.

We can make the following observations from these numerical experiments.

- The optimal value of the shape parameter $\varepsilon_{\mathrm{opt}}$ depends on the test function. However, it does not vary much when the number of centres or even the domain is changed.

- The errors with $\varepsilon = \varepsilon_{\mathrm{opt}}$ or even $\varepsilon = 0$ are always comparable with the error of the finite element method and reduce with the same rate when the set of centres

	square	disk	disk with hole	polygon
$\Xi^{(1)}$	0.045	0.047	0.052	0.046
$\Xi^{(2)}$	0.096	0.099	0.112	0.100
$\Xi^{(3)}$	0.203	0.203	0.225	0.207
$\Xi^{(4)}$	0.401	0.418	0.455	0.418
$\Xi^{(5)}$	0.819	0.819	0.890	0.890

Table 3: 'Safe' shape parameter $\varepsilon_{\text{dmin}}$ for each discretisation.

	square		disk		disk with hole		polygon	
$\Xi^{(1)}$	1.36	[1.22,1.49]	1.40	[1.21,1.59]	1.33	[1.09,1.54]	1.39	[1.00,1.75]
$\Xi^{(2)}$	1.32	[1.14,1.47]	1.34	[1.21,1.45]	1.31	[1.15,1.45]	1.33	[0.88,1.67]
$\Xi^{(3)}$	1.31	[1.15,1.46]	1.31	[1.20,1.41]	1.31	[1.17,1.44]	1.31	[0.87,1.63]
$\Xi^{(4)}$	1.31	[1.16,1.45]	1.30	[1.20,1.39]	1.30	[1.17,1.42]	1.31	[0.92,1.61]
$\Xi^{(5)}$	1.32	[1.16,1.46]	1.29	[1.19,1.39]	1.30	[1.17,1.43]	1.30	[0.85, 1.63]

Table 4: Optimal shape parameters for the rms error of the solution $\hat{u}$ for the test function u_1. For each domain, the number in the first column is the optimal shape parameter, whereas the second column indicates the range of values of the shape parameter, for which the rms error is at most twice the optimal error.

	square	disk	disk with hole	polygon
$\Xi^{(1)}$	1.13	1.11	0.67	1.20
$\Xi^{(2)}$	1.25	1.26	1.14	1.24
$\Xi^{(3)}$	1.27	1.27	1.27	1.28
$\Xi^{(4)}$	1.28	1.28	1.28	1.27
$\Xi^{(5)}$	1.28	1.28	1.28	1.27

Table 5: Optimal shape parameters for the rms differentiation error for the test function u_1.

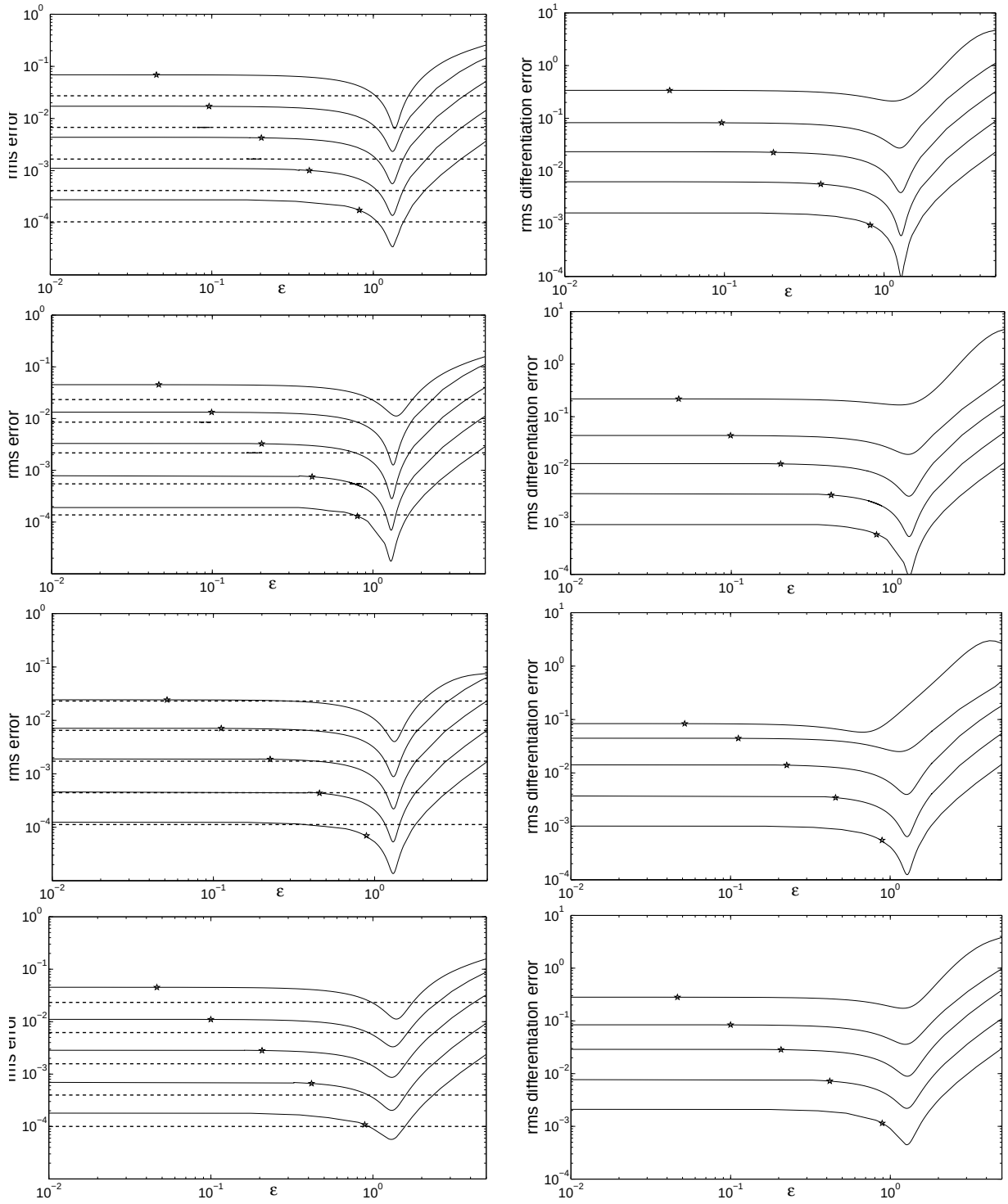
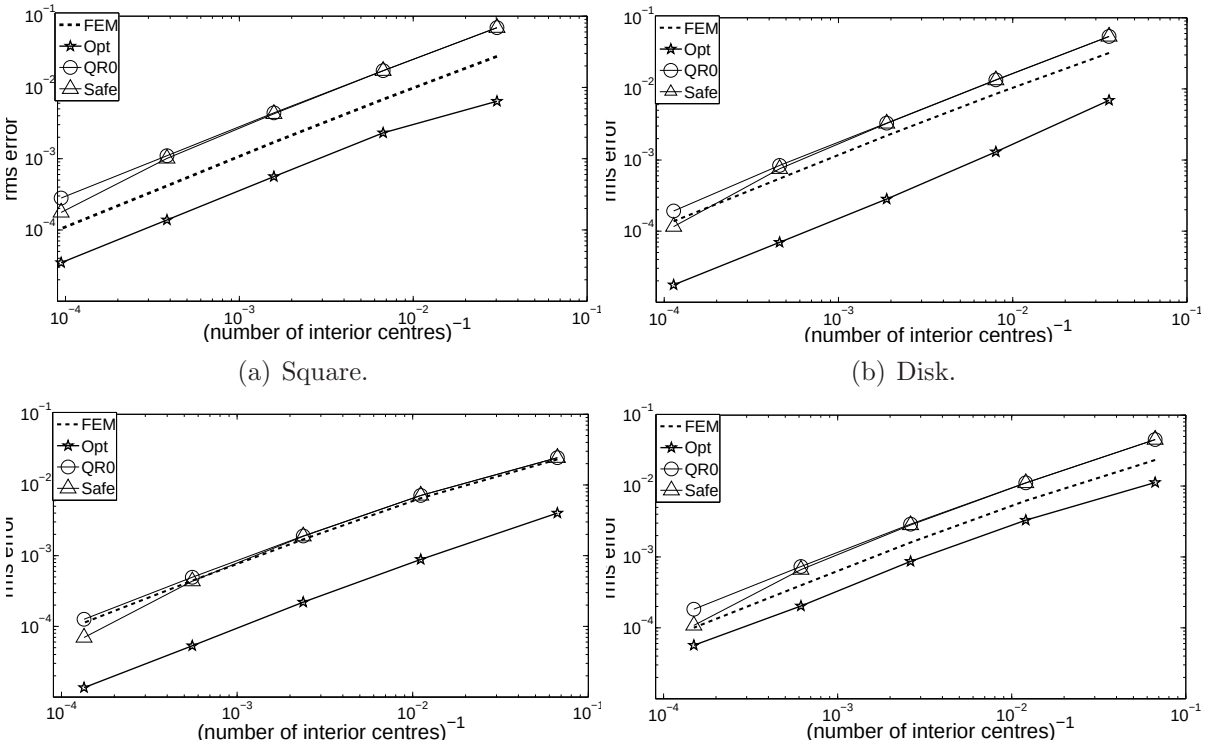


Figure 4: Left: The rms error of the Gaussian RBF-FD solutions for the test function u_1 on five sets of centres as a function of the shape parameter ε (solid lines) compared to the rms error of the FEM solutions (dashed lines). Right: The numerical differentiation error of the RBF-FD solutions. From top to bottom: square, disk, disk with hole and polygonal domain. In each subfigure the five solid curves present the error of the RBF-FD method on the five sets of centres, whereas the dashed constant curves show the error of the finite element method on the five triangulations for comparison. The stars indicate the value of $\varepsilon = \varepsilon_{\mathrm{dmin}}$.



(a) Square. (b) Disk.

(c) Disk with hole. (d) Polygonal domain.

Figure 5: The rms error of the Gaussian RBF-FD solutions for the test function u_1 on five sets of centres as function of the number of degrees of freedom, for three values of the shape parameter: `Safe` refers to $\varepsilon = \varepsilon_{\text{dmin}}$, as shown in Table 3, `QR0` refers to $\varepsilon = 0$, and `Opt` to the optimal values of ε shown in Table 4.

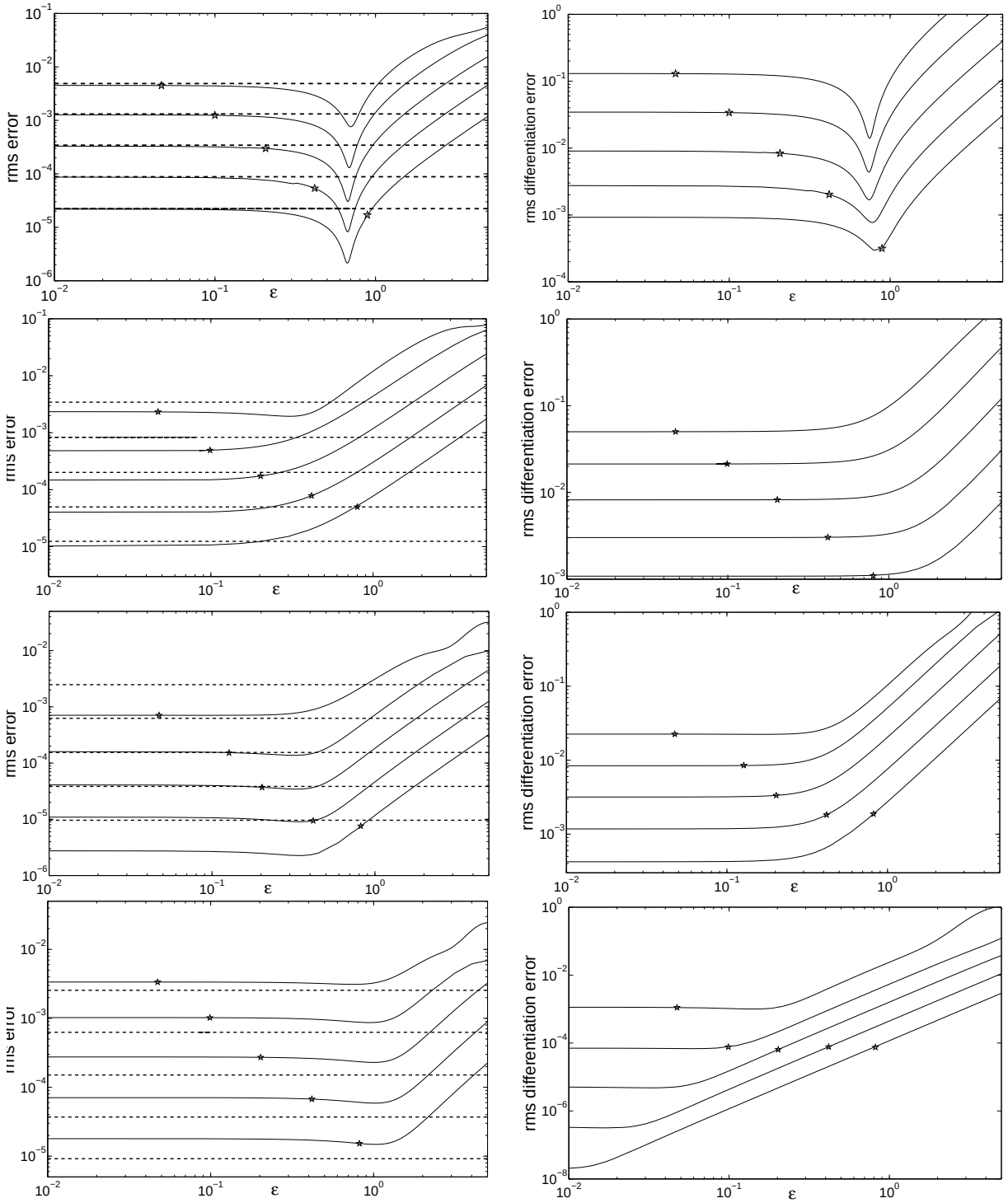
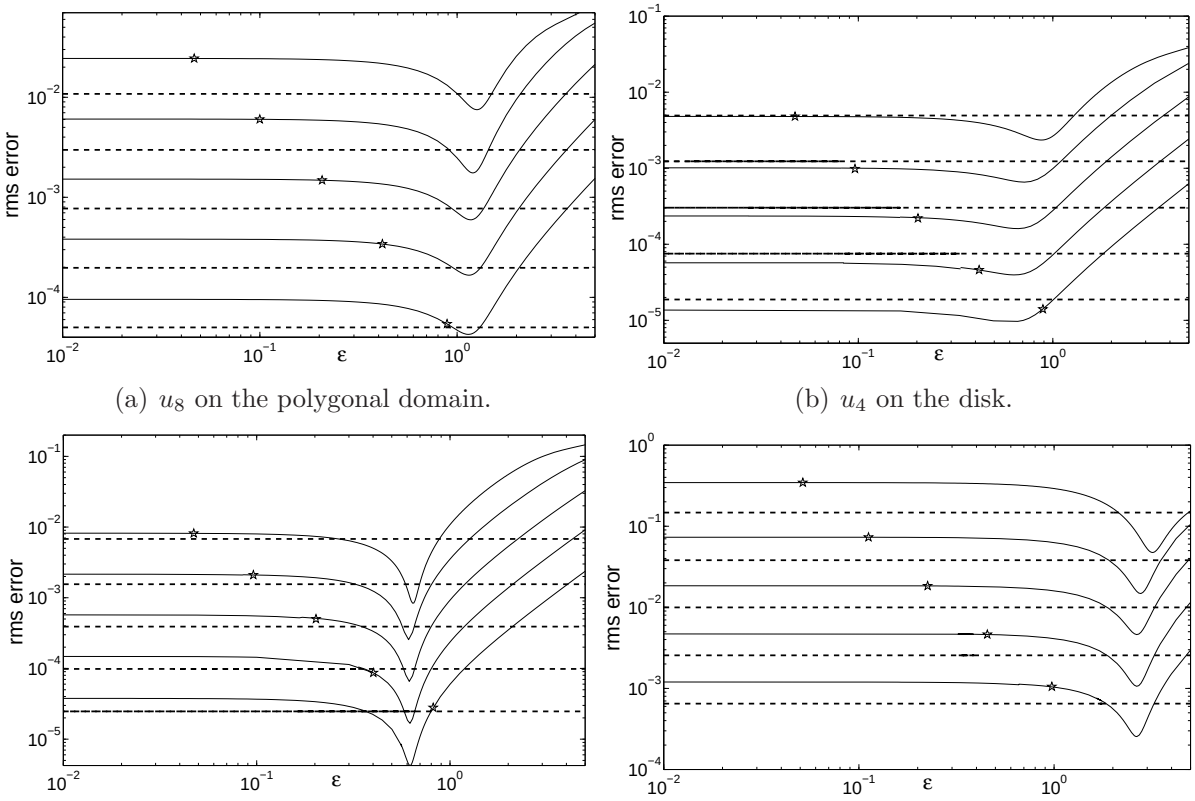


Figure 6: The rms error of the Gaussian RBF-FD solutions (left) and the rms differentiation error (right) as in Figure 4. From top to bottom: u_2 on the polygonal domain, u_7, u_3 and u_5 on the disk.



(a) u_8 on the polygonal domain.

(b) u_4 on the disk.

(c) u_2 on the square.

(d) u_6 on the disk with hole.

Figure 7: The rms error of the Gaussian RBF-FD solutions for the test functions u_8, u_4, u_2, u_6. The layout of the figures is the same as in Figure 4 (left).

is refined. If the error for ε_{opt} is significantly better than the error with $\varepsilon = 0$, then it is normally also significantly better than the FEM error.

- For certain test functions $\varepsilon = 0$ is optimal on some sets of centres, whereas a non-zero optimal value can be found on others. In these cases however, ε_{opt} does not perform significantly better than $\varepsilon = 0$, so that the latter is nevertheless near-optimal.

- The 'safe' shape parameter gives results close to $\varepsilon = 0$ on coarse sets of centres, but it becomes an increasingly dangerous strategy when the set of centres is refined, even though in some situations it happens by chance to be close to optimal.

- The value of ε optimal for the PDE error correlates well with the optimal value of ε for the numerical differentiation.

- It seems difficult to predict the value of the optimal shape parameter other than by numerical experiments. It is interesting to compare $\varepsilon_{\text{opt}} \approx 1.3$ for u_1 and 0.7 for u_2, with $\varepsilon_{\text{opt}} \approx 1.1$ for $u_8 = \frac{1}{2}u_1 + u_2$. Further experiments have shown that for the test problems with exact solution $au_1 + u_2$ with $0 \le a \le 1$ the optimal shape parameter lies between 0.7 and 1.3, for example $\varepsilon_{\text{opt}} \approx 0.7$ if $a = 0.05$, $\varepsilon_{\text{opt}} \approx 0.9$ if $a = 0.2$, $\varepsilon_{\text{opt}} \approx 1.3$ if $a = 1$.

5 Estimation of optimal shape parameter

Based on the observations at the end of the previous section, we suggest a multilevel algorithm for the estimation of the optimal shape parameter. It iteratively minimises a cost function defined with the help of either rms error between two solutions on a coarse and a fine set of centres, or the error of the numerical differentiation of a fine solution using the stencils generated on the coarse set of centres.

Let Ξ and Ξ^{ref} be two sets of centres such that $\Xi \subset \Xi^{\text{ref}}$, and $\varepsilon, \varepsilon_{\text{ref}}$ two values of the shape parameter. Denote by $\hat{u}$ (respectively, $\hat{u}_{\text{ref}}$) the Gaussian RBF solution of the Dirichlet problem (2)–(3) with the shape parameter ε on Ξ (respectively, ε_{ref} on Ξ^{ref}). As explained above, the stencils to set up the system can be computed either directly or by the QR method. Clearly, $\hat{u}_{\text{ref}}$ can be restricted to Ξ. Our first cost function is given by the root mean square distance between two approximate solutions on the set of centres Ξ,

$$\text{cost}(\varepsilon, \varepsilon_{\text{ref}}) := \Big(\frac{1}{\#\Xi_{\text{int}}} \sum_{\xi \in \Xi_{\text{int}}} \big(\hat{u}(\xi) - \hat{u}_{\text{ref}}(\xi)\big)^2 \Big)^{1/2}. \tag{29}$$

The second approach is to measure the accuracy of the RBF numerical differentiation formulae on the set Ξ, obtained with the shape parameter value ε,

$$\Delta u(\zeta) \approx \sum_{\xi \in \Xi_\zeta} w_{\zeta,\xi} u(\xi), \qquad \zeta \in \Xi_{\text{int}},$$

against the ones on the refined set of centres $\Xi^{\rm ref}$, obtained with the shape parameter value $\varepsilon_{\rm ref}$,

$$\Delta u(\zeta) \approx \sum_{\xi \in \Xi_\zeta^{\rm ref}} w_{\zeta,\xi}^{\rm ref} u(\xi), \qquad \zeta \in \Xi_{\rm int}^{\rm ref}.$$

The error between two approximate Laplacians of $\hat{u}_{\rm ref}$ at $\zeta \in \Xi_{\rm int}$ is given by

$$e_\zeta = \sum_{\xi \in \Xi_\zeta} w_{\zeta,\xi} \hat{u}_{\rm ref}(\xi) - \sum_{\xi \in \Xi_\zeta^{\rm ref}} w_{\zeta,\xi}^{\rm ref} \hat{u}_{\rm ref}(\xi),$$

and this leads to the second cost function in the form

$$\mathrm{cost}(\varepsilon, \varepsilon_{\rm ref}) := \Big(\frac{1}{\#\Xi_{\rm int}} \sum_{\zeta \in \Xi_{\rm int}} e_\zeta^2 \Big)^{1/2}. \tag{30}$$

Note that (30) is cheaper to compute than (29), especially if $\Xi^{\rm ref}$ is fixed and Ξ varies, since (30) does not require the knowledge of the approximate solution $\hat{u}$ of the Dirichlet problem on Ξ.

Underlying assumption is that the optimal shape parameters for two sets of centres are close together as observed in the numerical results of the previous section.

Algorithm 1. Estimation of optimal shape parameter using approximate solution on a refined set of centres. *Options* (referred to as **Algorithm 1a** and **Algorithm 1b**, respectively): **(a)** the cost function is defined by (29), and **(b)** the cost function is defined by (30). *Input:* two sets of centres $\Xi, \Xi^{\rm ref}$ such that $\Xi \subset \Xi^{\rm ref}$ and initial estimate of the optimal shape parameter $\varepsilon_{\rm ref}$. *Output:* estimated optimal shape parameter $\varepsilon_{\rm opt}$. *Parameters:* tolerances $\lambda > \delta > 0$, maximum number of iterations m, upper bound C for the shape parameter. In the numerical tests below the following parameter values have been used: $\delta = 0.01$, $\lambda = 0.1$, $m = 4$, and $C = 5$.

I. Compute Gaussian RBF solution $\hat{u}_{\rm ref}$ on $\Xi^{\rm ref}$ with shape parameter $\varepsilon_{\rm ref}$ and find $\varepsilon \in [\varepsilon_{\min}, \varepsilon_{\max}]$ such that $\mathrm{cost}(\varepsilon, \varepsilon_{\rm ref})$ is minimised, where $[\varepsilon_{\min}, \varepsilon_{\max}] = [0, C]$ if $\varepsilon_{\rm ref} = 0$ and $[\varepsilon_{\min}, \varepsilon_{\max}] = [\varepsilon_{\rm ref} - \lambda, \varepsilon_{\rm ref} + \lambda]$ otherwise.

II. *For* $i = 1, \ldots, m$:
   If $|\varepsilon - \varepsilon_{\rm ref}| < \delta$: STOP and return $\varepsilon_{\rm opt} = \varepsilon$.
   ElseIf $\varepsilon = \varepsilon_{\min}$ or $\varepsilon = \varepsilon_{\max}$: STOP and return $\varepsilon_{\rm opt} =$ `NaN`.
   Else: Set $\varepsilon_{\rm ref} = \varepsilon$ and repeat Step I.
   Return: $\varepsilon_{\rm opt} =$ `NaN`

Remarks

1. Algorithm 1 fails if it returns `NaN`. If this happens with input $\varepsilon_{\rm ref} > 0$, this indicates a wrong initial estimate of the shape parameter and a remedy is to rerun Algorithm 1 with $\varepsilon_{\rm ref} = 0$. If however `NaN` is returned with input $\varepsilon_{\rm ref} = 0$, then the likely reason is that the solution $\hat{u}_{\rm ref}$ is not sufficiently accurate, and we suggest to replace $\Xi^{\rm ref}$ by a finer set of centres in this case.

2. In our numerical tests with the sets $\Xi^{(1)}, \ldots, \Xi^{(5)}$, Algorithm 1 reliably computes nearly optimal shape parameter if $\Xi^{\text{ref}} = \Xi^{(n+1)}$ for $\Xi = \Xi^{(n)}$ when $\Xi^{(n)}$ is sufficiently fine. However, for a coarse set of centres this may be unreliable and a larger gap between Ξ and Ξ^{ref} is needed. Therefore, we apply Algorithm 1 as follows: First run it with $\Xi = \Xi^{(1)}$, $\Xi^{\text{ref}} = \Xi^{(3)}$, $\varepsilon_{\text{ref}} = 0$, to obtain a nearly optimal shape parameter ε_1 for $\Xi^{(1)}$. Then run it with $\varepsilon_{\text{ref}} = \varepsilon_1$, $\Xi = \Xi^{(2)}$, $\Xi^{\text{ref}} = \Xi^{(3)}$, to obtain a nearly optimal shape parameter ε_2 for $\Xi^{(2)}$. If Algorithm 1 fails and returns `NaN` (which happened extremely rare in our tests), then set $\varepsilon_{\text{ref}} = 0$ and rerun Algorithm 1. In our tests Algorithm 1 never failed with $\varepsilon_{\text{ref}} = 0$. The value ε_2 is also used on the sets $\Xi^{(3)}, \Xi^{(4)}, \Xi^{(5)}$. Therefore, multiple values of ε are only tested on $\Xi^{(1)}, \Xi^{(2)}, \Xi^{(3)}$, which is cheaper than the cost of the computation with a single ε on $\Xi^{(5)}$.

3. Optimisation with respect to ε in Step I is done using MATLAB function `fminbnd`. We set `MaxFunEvals` $= 9$ and `TolX`$= 10^{-2}$ to reduce computation cost.

4. Parameter m is an upper bound on the number of computations of the RBF solution on Ξ^{ref}. Setting m to a small value may help to reduce the cost. However, if m is too small it causes unnecessary failures of the algorithm, and costly reruns with refined Ξ^{ref}. In our experiments $m = 4$ was sufficiently large to ensure that the failures are extremely rare. Setting $C = 5$ is justified by the graphs in Section 4 where the optimal shape parameter is always less than this number.

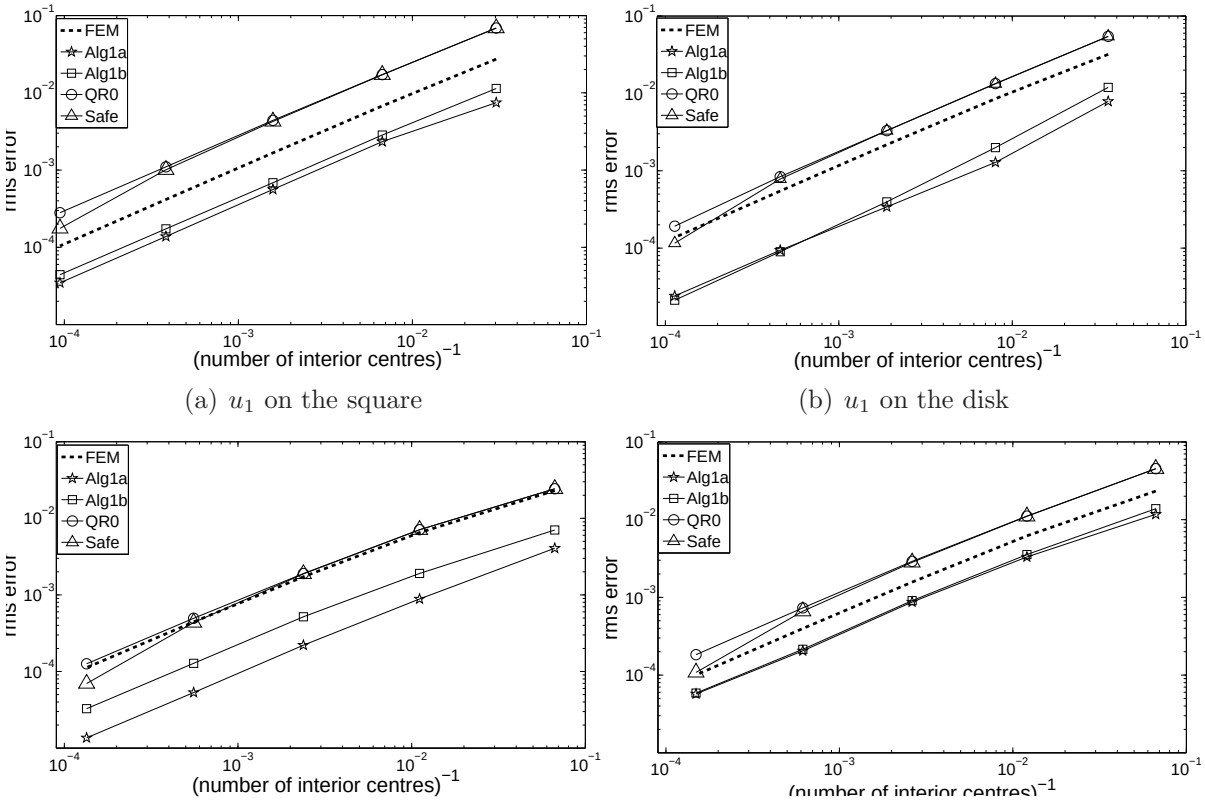
5. When computing Gaussian RBF solutions $\hat{u}$ or $\hat{u}_{\text{ref}}$ we either use Gauss-direct or Gauss-QR depending on whether the shape parameter ε is smaller than the smallest safe value $\varepsilon_{\text{dmin}}$ for which the condition number of the matrix of (12) does not exceed 10^{12}, see Table 3. If the interval $[\varepsilon_{\min}, \varepsilon_{\max}]$ includes $\varepsilon_{\text{dmin}}$, then we further reduce cost by first running `fminbnd` in the interval $[\varepsilon_{\text{dmin}}, \varepsilon_{\max}]$ using Gauss-direct, and then, only if $\varepsilon_{\text{dmin}}$ is optimal in this interval, we run `fminbnd` in the interval $[\varepsilon_{\min}, \varepsilon_{\text{dmin}}]$ using Gauss-QR.

Numerical experiments

Figures 8–10 and Tables 6–8 below illustrate the performance of Algorithm 1 for the test problems in Figures 4, 6 and 7. In these experiments Algorithm 1a effectively finds near-optimal shape parameter, whereas Algorithm 1b sometimes returns sub-optimal, albeit acceptable results. The tables also confirm that the number of iterations needed in Step II is small, typically just 2 or 3.

6 Conclusion and future work

In this paper we have demonstrated in numerical examples that for many problems the optimal shape parameter depends strongly on the right hand side of the Poisson equation, but only mildly on the density of the centres and on the domain Ω. We conjecture that this phenomenon extends to any problems where the exact solution is analytic in the domain and on its boundary. Based on this, we suggested in Section 5



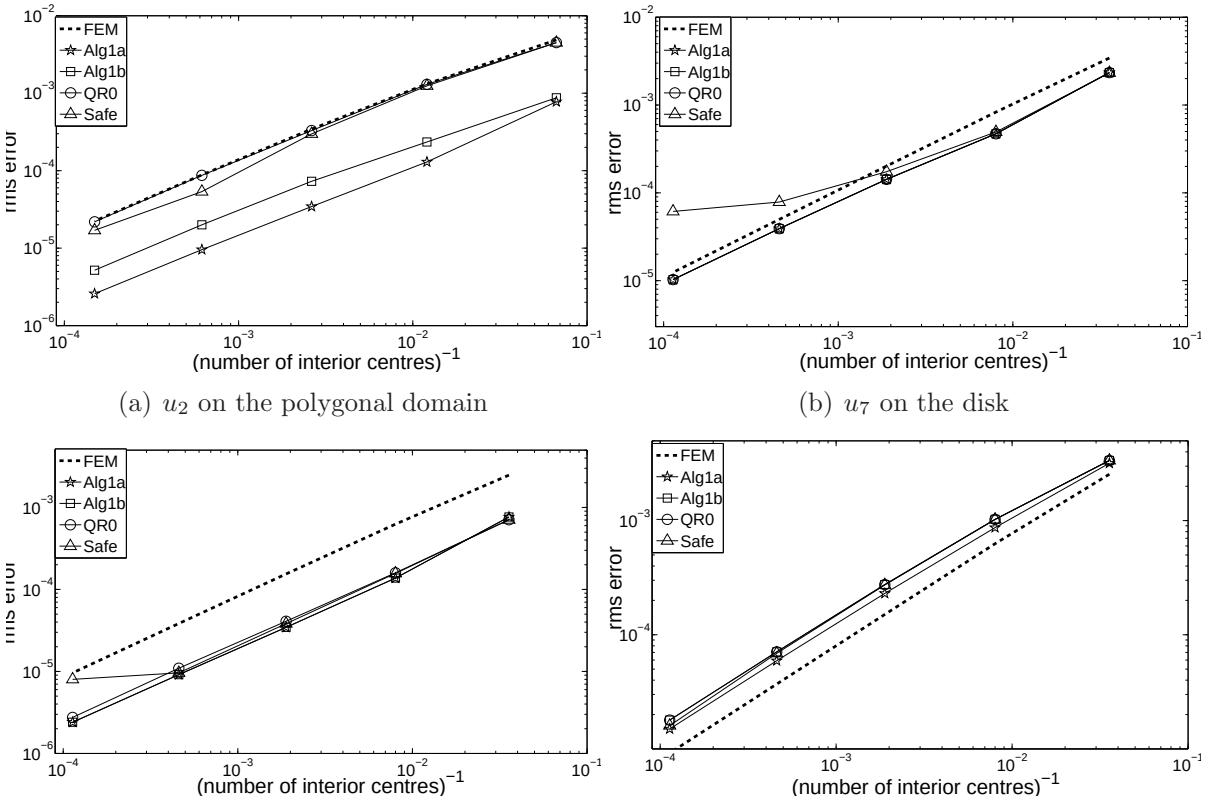
(a) u_1 on the square

(b) u_1 on the disk

(c) u_1 on the disk with hole

(d) u_1 on the polygonal domain

Figure 8: The rms error of the Gaussian RBF-FD solutions for the test function u_1 on five sets of centres as function of the number of degrees of freedom, with the shape parameter values produced by Algorithm 1: `Alg1a` and `Alg1b` refer to Algorithm 1a and 1b, respectively. For the sake of comparison, the error curves `FEM`, `Safe` and `QR0` with the same meaning as in Figure 5 are also included.



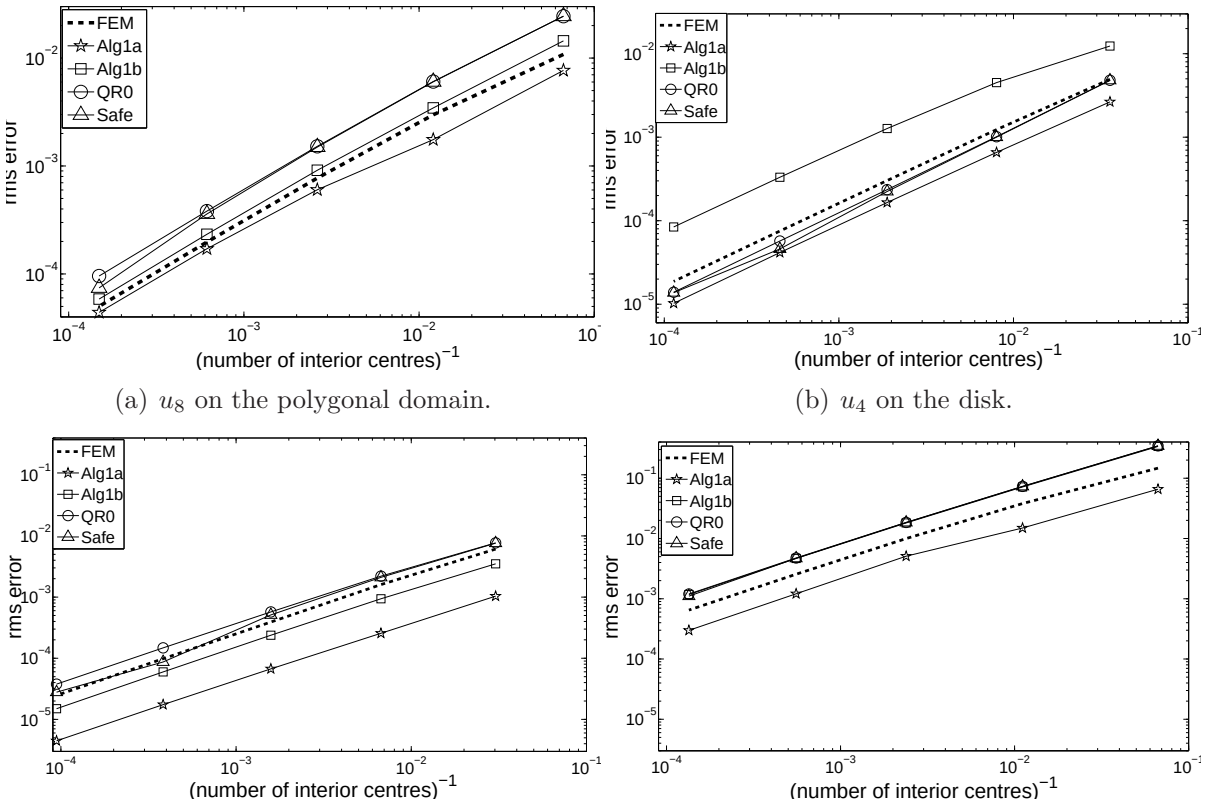
(a) u_2 on the polygonal domain

(b) u_7 on the disk

(c) u_3 on the disk

(d) u_5 on the disk

Figure 9: The rms error of the Gaussian RBF-FD solutions for the test functions and domains as in Figure 6 on five sets of centres, with the shape parameter values produced by Algorithm 1. The symbols in the legend are the same as in Figure 8.



(a) u_8 on the polygonal domain.

(b) u_4 on the disk.

(c) u_2 on the square.

(d) u_6 on the disk with hole.

Figure 10: The rms error of the Gaussian RBF-FD solutions for the test functions and domains as in Figure 7 on five sets of centres, with the shape parameter values produced by Algorithm 1. The symbols in the legend are the same as in Figure 8.

	square		disk		disk with hole		polygon	
	ε_{opt}	`nIter`	ε_{opt}	`nIter`	ε_{opt}	`nIter`	ε_{opt}	`nIter`
Algorithm 1a	1.31	2	1.35	2	1.31	2	1.33	2
Algorithm 1b	1.25	2	1.25	3	0.91	4	1.24	2

Table 6: The near optimal shape parameter ε_{opt} and the number of iterations `nIter` in Step II of Algorithm 1 when $\Xi = \Xi^{(2)}$ and $\Xi^{\text{ref}} = \Xi^{(3)}$ (see Remark 2 after Algorithm 1) for the test function u_1.

	u_2 polygonal		u_7 disk		u_3 disk		u_5 disk	
	ε_{opt}	`nIter`	ε_{opt}	`nIter`	ε_{opt}	`nIter`	ε_{opt}	`nIter`
Algorithm 1a	0.69	2	0	2	0.31	3	0.93	4
Algorithm 1b	0.74	2	0	1	0.17	2	0	1

Table 7: The near optimal shape parameter ε_{opt} and the number of iterations `nIter` as in Table 6 for the test functions and domains as in Figures 6 and 9.

an algorithm to estimate the optimal shape parameter by comparing RBF-FD solutions on two sets of centres and verified it numerically on the same test problems. Our tests with the full range of the shape parameters, including the flat limit at $\varepsilon = 0$ were possible thanks to the recent QR method [6] which we adapted to the interpolation with a constant term and computation of Gaussian RBF-FD stencils.

Apparently, it is difficult to explain this phenomenon theoretically and develop analytic methods to determine the optimal shape parameter for a given right hand side of the Poisson equation. Further work is needed to see whether this behaviour persists for other types of equations, non-Dirichlet boundary conditions or 3D problems, as well as for other radial basis functions.

The case $\varepsilon = 0$ seems of independent interest because, computed by QR method, it is effectively a polynomial rather than RBF method. It was sometimes optimal and in general competitive in our experiments. Its computational cost is the lowest of any ε requiring QR method [6].

Higher order stencils

In our experiments we used stencil supports generated by [3, Algorithm 1] which contain just 6 or 7 points as they are designed to compete with the finite element method based on linear shape functions. Therefore it is important to investigate whether the optimal shape parameter is still indifferent to domain shapes and densities of centres if larger stencils are employed. Figure 11 presents results of an initial test in this direction, confirming that this is likely to be the case. Here, the Poisson equation with the right hand side and Dirichlet boundary conditions derived from the exact solution u_1 of Table 1 was solved on the square domain with centres generated by the same triangulations $\mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(4)}$ considered before, see Figure 1 and Table 2. However, the

	u_8 polygonal		u_4 disk		u_2 square		u_6 disk with hole	
	ε_{opt}	`nIter`	ε_{opt}	`nIter`	ε_{opt}	`nIter`	ε_{opt}	`nIter`
Algorithm 1a	1.21	4	0.73	4	0.61	3	2.82	2
Algorithm 1b	0.84	3	1.91	4	0.74	2	0.38	4

Table 8: The near optimal shape parameter ε_{opt} and the number of iterations `nIter` as in Table 6 for the test functions and domains as in Figures 7 and 10.

new set of centres $\widetilde{\Xi}^{(i)}$ associated with $\mathcal{T}^{(i)}$ includes vertices and midpoints of edges of the triangulation $\mathcal{T}^{(i)}$, which implies $\widetilde{\Xi}^{(i)} = \Xi^{(i+1)}$ because $\mathcal{T}^{(i+1)}$ is obtained by the uniform refinement of $\mathcal{T}^{(i)}$. Obviously, $\widetilde{\Xi}^{(i)}$ is the set of centres corresponding to the finite element method with quadratic shape functions on $\mathcal{T}^{(i)}$, and the corresponding stencil support selection method includes into $\widetilde{\Xi}^{(i)}_\zeta$ all points of $\widetilde{\Xi}^{(i)}$ lying in the union of the triangles of $\mathcal{T}^{(i)}$ containing $\zeta \in \widetilde{\Xi}^{(i)}$. Thus, $\widetilde{\Xi}^{(i)}_\zeta$ consist of 9 points if ζ is the middle point of an edge of $\mathcal{T}^{(i)}$ and $3n+1$ points if ζ is an interior vertex connected to n other vertices of $\mathcal{T}^{(i)}$. We have solved the Dirichlet problem with Gaussian RBF-FD method using finite element stencil supports $\widetilde{\Xi}^{(i)}_\zeta$. Figure 11 provides the rms error of this solution and the rms differentiation error. The stars on the first three curves indicate the position of the 'safe' $\varepsilon = \varepsilon_{\text{dmin}}$, so that the QR method is used to the left of these points. Note that the values of $\varepsilon_{\text{dmin}}$ are now higher than those in Table 3 because larger stencils are used. The fourth curve (for $\widetilde{\Xi}^{(4)}$) is completely obtained by the QR method. We observe that the optimal shape parameter for the solution and numerical differentiation errors is about 1.3 for all $\widetilde{\Xi}^{(i)}$, $i = 1, \ldots, 4$, which is close to the values obtained for u_1 in Section 4, see Figure 4 and Table 4. We can also see that the errors are significantly better than those obtained with the same number of centres for the same problem in Section 4, as expected from larger stencils. However, Figure 11 does not seem to indicate a higher convergence order. Clearly, further research is needed on meshless stencil support selection algorithms leading to stencils of size comparable with higher order finite element methods and delivering comparably accurate RBF-FD solutions.

Adaptive centres

For practical applications it is important to determine good shape parameters for more complex right hand sides, where typically distributions of centres with spatially varying densities are needed. In [3] we tested RBF-FD methods on adaptive centres generated by adaptive refinement for the Dirichlet problem (2)–(3), where the domain Ω is the disk sector defined by the inequalities $r < 1$, $-3\pi/4 < \varphi < 3\pi/4$ in polar coordinates, the right hand side $f = 0$, the boundary conditions are defined by $g(r, \varphi) = \cos(2\varphi/3)$ along the arc, and $g(r, \varphi) = 0$ along the straight lines. The exact solution is $u(r, \varphi) = r^{2/3} \cos(2\varphi/3)$. Figure 12 and Table 9 present the results of new experiments for several sets of centres generated by adaptive refinement of an initial triangulation as described in [3, Section 4]. We use small stencil supports generated by [3, Algorithm 1] and apply QR method when the condition number of the linear system (12) exceeds 10^{12}. Figure 12

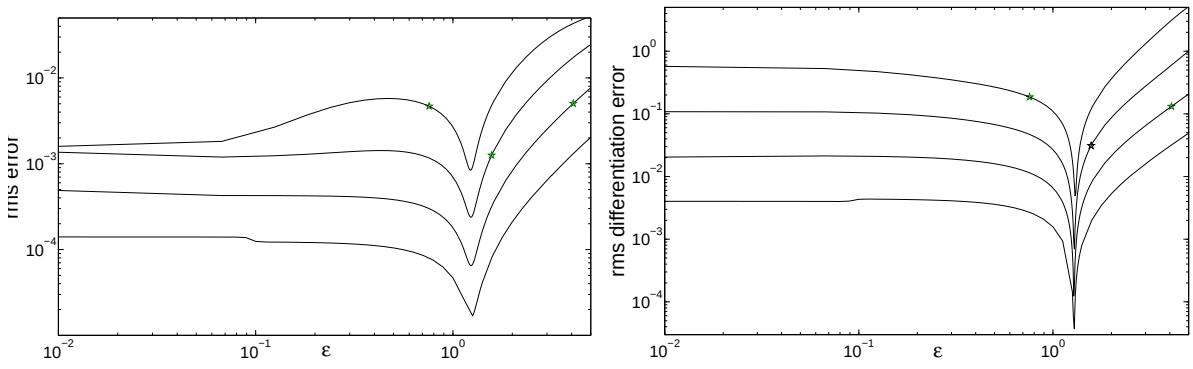


Figure 11: The rms error of the Gaussian RBF-FD solutions (left) and the rms numerical differentiation error (right) for the test function u_1 on the square using stencil supports of the quadratic finite element method.

depicts the rms error of the Gaussian RBF-FD solution against the exact solution for two versions of the shape parameter: $\varepsilon = 0$ and optimal $\varepsilon = \varepsilon_{\text{opt}}$ found by minimising the rms error. Recall that in [3] the shape parameter was chosen individually for each stencil as the smallest 'safe' ε with the property that the condition number of (12) does not exceed 10^{12}. Comparing Figure 12 with the curve for Gaussian RBF in [3, Figure 10a], we observe that the results are very close. In particular, the optimal shape parameter shows no significant advantage over $\varepsilon = 0$ for this test function, similar to what we found for u_3, u_5, u_7 on uniform refinements in Section 4. Table 9 gives more detailed information about the values of the optimal shape parameter and corresponding errors. Note that choosing a single value of the shape parameter everywhere in the domain may not be the right approach for functions with singularities or spatially varying smoothness. We hope nevertheless that the results of this paper will help develop effective shape parameter selection algorithms for more complicated problems in the future.

Acknowledgements

We are grateful to two anonymous referees for useful suggestions that have helped to improve the paper. The second author thanks Professor Hoang Xuan Phu for stimulating discussions.

References

[1] J. P. Boyd and L. Wang. Truncated Gaussian RBF differences are always inferior to finite differences of the same stencil width. *Commun. Comput. Phys.*, 5:42–60, 2009.

[2] M. D. Buhmann. *Radial Basis Functions*. Cambridge University Press, New York, NY, USA, 2003.

[3] O. Davydov and D. T. Oanh. Adaptive meshless centres and RBF stencils for Poisson equation. *J. Comput. Phys.*, 230:287–304, 2011.

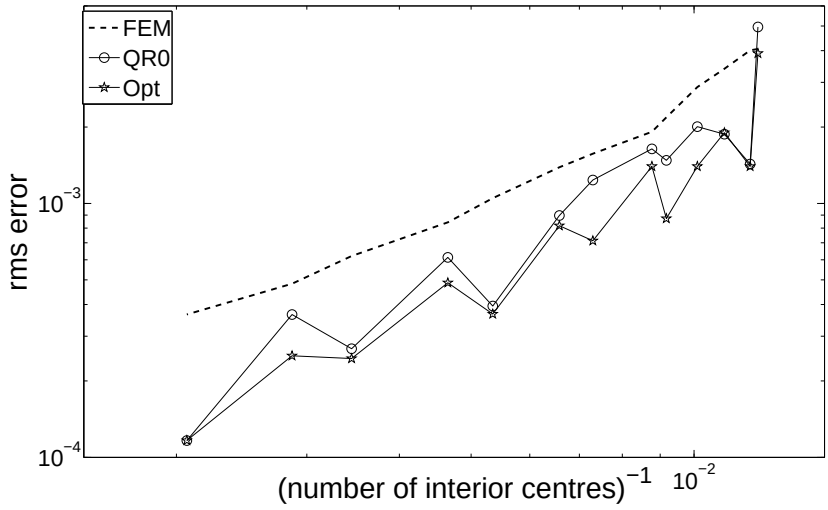


Figure 12: The rms error of the Gaussian RBF-FD solutions on adaptive centres for the test problem described in Section 6: The curve marked by `QR0` is obtained with $\varepsilon = 0$ and `Opt` with the optimal shape parameter, whereas `FEM` indicates the rms error of the finite element method with linear shape functions.

#centres	ε_{opt}	rms error for ε_{opt}	rms error for $\varepsilon = 0$	rms error of FEM
82	2.26	3.91e-03	4.96e-03	4.06e-03
84	0	1.43e-03	1.43e-03	4.01e-03
91	0	1.87e-03	1.87e-03	3.39e-03
99	1.03	1.44e-03	2.01e-03	2.87e-03
109	0.95	8.72e-04	1.48e-03	2.18e-03
114	0.74	1.39e-03	1.64e-03	1.92e-03
137	0.87	7.13e-04	1.24e-03	1.57e-03
152	0.52	8.17e-04	8.97e-04	1.39e-03
187	0.38	3.67e-04	3.95e-04	1.05e-03
215	0.58	4.87e-04	6.13e-04	8.43e-04
290	0.32	2.45e-04	2.68e-04	6.21e-04
349	0.59	2.51e-04	3.65e-04	4.83e-04
484	0	1.17e-04	1.17e-04	3.66e-04

Table 9: The sizes of the sets of centres, values of the optimal shape parameter and the rms errors for $\varepsilon = \varepsilon_{\text{opt}}$, $\varepsilon = 0$ and FEM for the experiments on adaptive centres described in Section 6.

[4] G. F. Fasshauer. *Meshfree Approximation Methods with MATLAB*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2007.

[5] B. Fornberg and N. Flyer. The Gibbs phenomenon for radial basis functions. In A. Jerri, editor, *The Gibbs Phenomenon in Various Representations and Applications*, pages 201–222. Sampling Publishing, Potsdam, NY, 2011.

[6] B. Fornberg, E. Larsson, and N. Flyer. Stable computations with Gaussian radial basis functions. *SIAM J. Sci. Comput.*, 33(2):869–892, 2011.

[7] B. Fornberg and E. Lehto. Stabilization of RBF-generated finite difference methods for convective PDEs. *J. Comput. Phys.*, 230:2270–2285, 2011.

[8] B. Fornberg and C. Piret. A stable algorithm for flat radial basis functions on a sphere. *SIAM J. Sci. Comput.*, 30:60–80, 2007.

[9] B. Fornberg and G. Wright. Stable computation of multiquadric interpolants for all values of the shape parameter. *Comput. Math. Appl.*, 48:853–867, 2004.

[10] B. Fornberg and J. Zuev. The Runge phenomenon and spatially variable shape parameters in RBF interpolation. *Comput. Math. Appl.*, 54:379–398, 2007.

[11] C. K. Lee, X. Liu, and S. C. Fan. Local multiquadric approximation for solving boundary value problems. *Comput. Mech.*, 30(5-6):396–409, 2003.

[12] *Partial Differential Equation Toolbox^{TM} User's Guide*. The MathWorks, Inc, 2009.

[13] C. Shu, H. Ding, and K. S. Yeo. Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Eng.*, 192(7-8):941–954, 2003.

[14] A. I. Tolstykh and D. A. Shirobokov. On using radial basis functions in a 'finite difference mode' with applications to elasticity problems. *Computational Mechanics*, 33(1):68–79, 2003.

[15] H. Wendland. *Scattered Data Approximation*. Cambridge University Press, 2005.

[16] G. B. Wright and B. Fornberg. Scattered node compact finite difference-type formulas generated from radial basis functions. *J. Comput. Phys.*, 212(1):99–123, 2006.