Using wavelets in the dual reciprocity method for the Poisson's equation

EINDHOVEN UNIVERSITY OF TECHNOLOGY
Department of Mathematics and Computing Science

RANA 00-17
December 2000

Using Wavelets in the Dual Reciprocity Method for the Poisson's equation

by

E. Perry-Debain and H.G. ter Morsche

Reports on Applied and Numerical Analysis
Department of Mathematics and Computing Science
Eindhoven University of Technology
P.O. Box 513
5600 MB Eindhoven. The Netherlands
ISSN: 0926-4507

Using Wavelets in the Dual Reciprocity Method for the Poisson's equation

E. Perrey-Debain, H.G. ter Morsche

December 2000

Abstract

Based on the idea of the Dual Reciprocity Method, a numerical method has been devised to interpolate the source term of the Poisson's equation by using B-spline approximation and then use them to approximate particular solutions. The advantage of such a construction is the possibility of using the Fast Wavelet Transform to minimize the number of coefficients in the wavelet expansion. Practical concerns in the implementation of the method are discussed. Three numerical examples are presented to validate the approach.

1 Introduction

In the past decade there has been considerable research into the problem of computing and/or eliminating the domain integrals which occur in the numerical solution of partial differential equations by integral equation methods. The domain integral typically arise in two ways: (a) when the differential equation is linear but inhomogeneous (cf.[2] [1]) or (b)when a non-linear or time-dependent equation is solved by a perturbation approach with the non-linearities and/or time-dependency viewed as an inhomogeneous perturbation of a linear and/or time-independent problem (cf. [3] [4]). In both cases, the accurate evaluation of these domain integrals may be the dominant computational task in terms of arithmetic operations.

The current approaches to this problem seem to fall into two categories: (1) methods such as the Dual and Multiple Reciprocity Methods which replace the domain integral by a series of boundary integrals due to the approximation of the inhomogeneous terms by means of polynomials, a finite series of Radial Basis Functions, a Fourier series (cf. [1] [5]) or by means of spline functions (cf.[6]) and (2) methods based on direct numerical integrations. This method requires domain discretization (see references in [2] and [8]).

At present the most widely used technique seems to be the Dual Reciprocity Method (DRM). This technique was initially developed by Nardini and Brebbia [7] in the context of two-dimensional elastodynamics and has been extended to deal with a variety of problems wherein the domain integral may account for linear/non-linear static/dynamic effects. A comprehensive description of the DRM, its applications and the computer implementation of the method was given by Partdrige et al.[2].

In this paper, we show how to replace the traditional Radial Basis Functions by B-spline wavelets [9] in order to get an *efficient* numerical evaluation of particular solutions for two-dimensional Poisson's equation. Section 2 contains a brief overview of the Dual Reciprocity Method with Radial Basis Functions. In section 3, we present the theoretical ingredients needed for our numerical applications (Wavelet theory, Fast Wavelet Transform, interpolation scheme and the numerical evaluation of particular solutions). Three numerical examples of increasing complexity are given in section 4 to illustrate the effectiveness of our proposed method. We also compare our results with previous methods.

2 The DRM with Radial Basis Function

2.1 Integral formulation and particular solution

Let $\Omega \subset \mathbb{R}^2$ be an open bounded simply connected set with Lipschitz boundary Γ. We consider the problem of solving the boundary value problem

$$\triangle u(\boldsymbol{x}) = f(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Omega \tag{1}$$

with boundary conditions

$$B_i(u) \mid_{\Gamma_i} = 0, \qquad i = 1, 2, ..., m \tag{2}$$

where Γ_i is a partition of Γ and $B_{i=1...m}$ are the usual boundary operators (Dirichlet, Neumann and Robin type). To solve eqn (1), it is convenient to first reduce it to an equivalent homogeneous equation, since the resulting boundary integral reformulation will then be free of domain integrals. To do this, let u_p be a particular solution of eqn (1) (i.e. a solution of 1 which does not necessarily satisfy the given boundary conditions) and let $u = v + u_p$. Then by linearity, v satisfies

$$\triangle v(\boldsymbol{x}) = 0, \qquad \boldsymbol{x} \in \Omega \tag{3}$$

with boundary conditions derived from that of u and u_p. The integral formulation for the previous equation reads

$$c(\boldsymbol{x})v(\boldsymbol{x}) + \int_\Gamma v(\boldsymbol{y}) \frac{\partial G(\boldsymbol{x}, \boldsymbol{y})}{\partial n} d\Gamma(\boldsymbol{y}) - \int_\Gamma G(\boldsymbol{x}, \boldsymbol{y}) \frac{\partial v(\boldsymbol{y})}{\partial n} d\Gamma(\boldsymbol{y}) = 0 \tag{4}$$

where n is the outward normal unit vector, G is the fundamental solution for the bidimensional Laplace equation $G(\boldsymbol{x}, \boldsymbol{y}) = -(2\pi)^{-1} \ln |\boldsymbol{x} - \boldsymbol{y}|$ and $c(\boldsymbol{x})$ is a geometric coefficient associated with the location of the point $\boldsymbol{x}$ with respect to the surface Γ. Using standard discretization techniques [10] leads to the following non sparse system

$$\mathbf{Hv} - \mathbf{G} \frac{\partial \mathbf{v}}{\partial n} = 0 \tag{5}$$

where $\mathbf{v}$ and $\partial \mathbf{v}/\partial n$ denote respectively the boundary values for v and its normal derivative, $\mathbf{G}$ and $\mathbf{H}$ are the coefficient matrices obtained by integrating G and its normal derivative over the boundary elements. By linearity, the previous equation is equivalent to

$$\mathbf{Hu} - \mathbf{G} \frac{\partial \mathbf{u}}{\partial n} = \mathbf{Hu}_p - \mathbf{G} \frac{\partial \mathbf{u}_p}{\partial n} \tag{6}$$

From the previous equation, it can be seen that the particular solution u_p transforms the domain integral into boundary integrals. Since u_p is not unique (it is only required that u_p and its normal derivative are integrable over the boundary), there are many methods for computing u_p. One of the most efficient techniques, called Dual Reciprocity Method, consists in expanding f as a finite series of Radial Basis Functions (RBFs).

2.2 The Radial Basis Functions approximation

Radial basis function interpolation to scattered data $(\boldsymbol{x}_i, f_i = f(\boldsymbol{x}_i))$ for the set of distinct points $A = \{\boldsymbol{x}_i \in \overline{\Omega} = \Omega \cup \Gamma \quad i = 1, ..., M\}$ uses a radial function $\phi : \mathbb{R}^+ \rightarrow \mathbb{R}$ and the space $\mathbb{P}_q$ of polynomials of $\mathbb{R}^2$ with total degree not exceeding q to construct the interpolant

$$f_I(\boldsymbol{x}) = \sum_{i=1}^{M} a_i \phi(|\boldsymbol{x} - \boldsymbol{x}_i|) + \sum_{i=1}^{Q} b_i p_i(\boldsymbol{x}) \tag{7}$$

via the linar system

$$\sum_{i=1}^{M} a_i \phi(|\boldsymbol{x}_j - \boldsymbol{x}_i|) + \sum_{i=1}^{Q} b_i p_i(\boldsymbol{x}_j) = f_j, \qquad 1 \le j \le M, \tag{8}$$

$$\sum_{j=1}^{M} a_j p_i(\boldsymbol{x}_j) = 0, \qquad 1 \le i \le Q = \begin{pmatrix} q+2 \\ 2 \end{pmatrix}, \tag{9}$$

where $p_1, ..., p_Q$ is a basis of $\mathbb{P}_q$. For a wide choice of functions ϕ and polynomial degree q, including the case $q = Q = 0$, the non-singularity of the $(M+Q) \times (M+Q)$ previous system, written as

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \tag{10}$$

in matrix notation, has been established by Micchelli [11] and Powell [12]. By linearity, an approximate solution u_p satisfying $\triangle u_p = f_I$ on Ω can be found by solving

$$\triangle \psi_i(\boldsymbol{x}) = \phi(|\boldsymbol{x} - \boldsymbol{x}_i|), \qquad 1 \le i \le M \tag{11}$$

$$\triangle \psi_i^p(\boldsymbol{x}) = p_i(\boldsymbol{x}), \qquad 1 \le i \le Q \tag{12}$$

for all $\boldsymbol{x} \in \mathbb{R}^2$. Then u_p is given by

$$u_p(\boldsymbol{x}) = \sum_{i=1}^{M} a_i \psi_i(\boldsymbol{x}) + \sum_{i=1}^{Q} b_i \psi_i^p(\boldsymbol{x}). \tag{13}$$

Since $\triangle$ is rotationally symmetric, then to solve eqn(11), it is natural to look for a radial solution $\psi(r)$. In this case

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r \frac{\partial \psi}{\partial r}\right)(r) = \phi(r) \tag{14}$$

and $\psi_i(\boldsymbol{x}) = \psi(|\boldsymbol{x} - \boldsymbol{x}_i|)$. Equation (14) can be integrated using standard variation of parameter formulas for ordinary differential equations. Equation (12) is usually solvable by the method of undeterminated coefficients. The form of eqn(14) suggests that one seeks for "good" functions ϕ that can be integrated analytically. The functions $\phi(r) = r$ (muti-conic function) and $\phi(r) = r^{2m-2} \ln r$ (m^{th}-order thin plate spline, $m = 2, 3, ...$) have these properties (the associated particular solutions are simply given by $\psi(r) = r^3/9$ and $\psi(r) = (r^m/2m)^2(\ln r - 1/m)$). These two radial functions have been widely used in the engineering community for their simplicity and their convergence properties.

With repect to the degree of approximation of the radial basis functions we introduce in the following definition the two notions: the *domain mesh size* h corresponding to a set of points $A \in \overline{\Omega}$ and the *approximation rate* λ which corresponds to a radial function ϕ. In this definition f_h stands for a function of the form 7, which interpolates a given function $f \in H^\lambda(\Omega)$ at the interpolation points A with mesh size h. Here $H^\lambda(\Omega)$ is the Sobolev space of order λ.

Definition 1

$$h = \sup_{\boldsymbol{x} \in \overline{\Omega}} \min\{|\boldsymbol{x} - \boldsymbol{x}_i|, \quad \boldsymbol{x}_i \in A\}$$

Definition 2 *A number λ is called the approximation rate if for all $f \in H^\lambda(\Omega)$ there exists a number $h_f > 0$ such that for all h with $0 < h < h_f$ and sets of interpolation points A with mesh size h, the existence and uniqueness of f_h is guaranteed. Moreover, one has*

$$\|f - f_h\|_{L_2(\Omega)} =\le C\, h^\lambda,$$

where C is independent of f_h.

Then the following theorem holds [13]

Theorem 1 *The approximation rate for the m^{th}-order spline with $q = m - 1$ (see eqn(7)) is m; the approximation rate for the multi-conic with $q = 0$ is $3/2$*

The next important results due to Powell [14] and Levesley [15] establishes the uniform convergence of the 2^{nd}-order spline (called Thin Plate Spline or TPS) and the multi-conic function

Theorem 2 *Let f_h be the TPS interpolant to f. Then for any $f \in C^2(\Omega)$ there exists a constant h such that sufficiently small h one has*

$$\max_{\boldsymbol{x}\in\overline{\Omega}} |f(\boldsymbol{x}) - f_h(\boldsymbol{x})| \leq c|\ln h|h$$

Theorem 3 *Let f_h be the multi-conic interpolant to f. Then for any $f \in C^2(\Omega)$ there exists a constant c such that for h sufficiently small h one has*

$$\max_{\boldsymbol{x}\in\overline{\Omega}} |f(\boldsymbol{x}) - f_h(\boldsymbol{x})| \leq ch^{1/2}$$

Some other frequently used radial basis functions for function approximation are the gaussian, the multiquadrics and the inverse multiquadrics (See [16] for convergence properties).

3 The DRM with Wavelets

3.1 Preliminaries

3.1.1 Stationary multiresolution on $\mathbb{R}^2$

The concept of multiresolution analysis plays a central role in the context of classical wavelets on $\mathbb{R}^n$. In this section, we shall briefly recall the basic setting of wavelet analysis as far as it is needed for our purposes. For more details we refer to Chui [17], Dahmen [18], Jawerth and Sweldens [19].

The scaling function and the subspaces V_j Let H be the Hilbert space $L_2(\mathbb{R}^2)$ with the usual inner product $< \cdot, \cdot >$ and associated norm $\| \cdot \| = < \cdot, \cdot >^{1/2}$. A multiresolution sequence $\mathcal{V} = \{V_j\}_{j\in\mathbb{Z}}$ consists of nested closed subspaces $V_j \subset H$ whose union is dense in H, i.e.

$$V_j \subset V_{j+1}, \qquad \text{clos}_H \left(\bigcup_{j\in\mathbb{Z}} V_j \right) = H. \tag{15}$$

In all cases of practical interest the spaces V_j have the form

$$V_j = \text{clos}_H < \phi_{j,k} : \quad k \in \mathbb{Z}^2 >$$

where functions $\phi_{j,k}$ form a Riesz basis, i.e. they satisfy the stability condition

$$A \sum_{k\in\mathbb{Z}^2} |c_k|^2 \leq \left\| \sum_{k\in\mathbb{Z}^2} c_k \phi_{j,k} \right\|_H^2 \leq B \sum_{k\in\mathbb{Z}^2} |c_k|^2 \tag{16}$$

for some positive constants A, B independent of the sequence $\{c_k\}_{k\in\mathbb{Z}^2}$. Nestedness of the spaces V_j combined with eqn (16) means that every $\phi_{j,k}$ possesses a unique expansion

$$\phi_{j,k} = \sum_{l\in\mathbb{Z}^2} m_{l,k}^j \phi_{j+1,l}$$

with a mask or filter sequence $\{m^j_{l,k}\}_{l\in\mathbb{Z}^2} \in l_2(\mathbb{Z}^2)$.
Now, we assume that functions $\phi_{j,k}$ stem from a unique function ϕ called scaling function or generator of the multiresolution sequence $\mathcal{V}$

$$\phi_{j,k}(\boldsymbol{x}) = 2^j\,\phi(2^j\,\boldsymbol{x} - k).$$

As a consequence of the inclusion $V_0 \subset V_1$, the function ϕ must be refinable, that is, there exist a mask $\boldsymbol{a} \in l_2(\mathbb{Z})$ such that

$$\phi(\boldsymbol{x}) = 2\sum_{kin\mathbb{Z}^2} a_l\,\phi(2\,\boldsymbol{x} - l). \tag{17}$$

Hence the functions $\phi_{j,k}$ automatically satisfy

$$\phi_{j,k} = \sum_{l\in\mathbb{Z}^2} a_{l-2\,k}\phi_{j+1,l}.$$

In this case, the mask is independent of the scale j and the spatial location k: the multiresolution is *stationary.*

The wavelet function and the detail spaces W_j We will use W_j to denote a space complementing V_j in V_{j+1}, i.e. a space that satisfies

$$V_{j+1} = V_j \oplus W_j \tag{18}$$

where the symbol $\oplus$ stands for direct sum. In other words, each element of V_{j+1} can be written, in a unique way, as the sum of an element of W_j and an element of V_j. We note that the spaces W_j themselves are not necessarily unique; there may be several ways to complement V_j in V_{j+1}. The spaces W_j contains the "detail" information at resolution $j+1$. Consequently,

$$H = ...W_{-1} \oplus W_0 \oplus W_1...$$

Now, for j fixed, the wavelet functions $\psi^e_{j,k}$ $e = 1,2,3; k \in \mathbb{Z}^2$ constitute Riesz basis of the detail space W_j. They are obtained by dilatations and integer translations of a finite number of mother functions ("mother wavelets"). In fact, one has

$$\psi^e_{j,k}(\boldsymbol{x}) = 2^j\psi^e(2^j\boldsymbol{x} - k), \qquad e = 1,2,3.$$

Equation (18) implies that there exist some matrices (b^e_k) such that

$$\psi^e(\boldsymbol{x}) = 2\sum_{k\in\mathbb{Z}^2} b^e_k\phi(2\boldsymbol{x} - k) \tag{19}$$

In the sequel, we shall restrict ourselves to refinable functions with *compact* support and we will always assume that $\mathrm{supp}(\boldsymbol{a}) = \{k \in \mathbb{Z}^2 | a_k \neq 0\}$ in (17) and $\mathrm{supp}(\boldsymbol{b}^e)$ in (19) are ***finite.***

Biorthogonal basis In practice, it is often very convenient to have access to a suitable biorthogonal wavelet basis. Here, a dual scaling function $\tilde{\phi}$ and dual wavelets $\tilde{\psi}^e$ generate a dual multiresolution of H with subspaces $\tilde{V}_j$ and $\tilde{W}_j$, such that they form a dual pair. With $< \cdot,\cdot >$ denoting the usual innerproduct in $L(\mathbb{R}^2)$ the duality means:

$$\begin{aligned} < \phi, \tilde{\phi}(\cdot - k) > &= \delta_{0,k} \\ < \psi^e, \tilde{\psi}^{e'}(\cdot - k) > &= \delta_{0,k}\delta_{e,e'}. \end{aligned}$$

Moreover, they must satisfy the biorthogonality conditions

$$< \phi, \tilde{\psi}^e(\cdot - k) >=< \tilde{\phi}, \psi^e(\cdot - k) >= 0.$$

By using scaling properties, we have the more general properties:

$$< \phi_{j,k}, \tilde{\phi}_{j',k'} >= \delta_{j,j'}\delta_{k,k'}$$

and

$$< \psi^e_{j,k}, \tilde{\psi}^{e'}_{j',k'} >= \delta_{j,j'}\delta_{k,k'}\delta_{e,e'}$$

where the definitions of $\tilde{\phi}_{j,k}$ and $\tilde{\psi}^e_{j,k}$ are similar to the ones for $\phi_{j,k}$ and $\psi^e_{j,k}$. Since the dual basis define a multiresolution, the dual functions must satisfy a refinement equation similar to (??)

$$\tilde{\phi}(\boldsymbol{x}) = 2 \sum_{k \in \mathbb{Z}^2} \tilde{a}_k \tilde{\phi}(2\boldsymbol{x} - k) \tag{20}$$

and

$$\tilde{\psi}^e(\boldsymbol{x}) = 2 \sum_{k \in \mathbb{Z}^2} \tilde{b}^e_k \tilde{\phi}(2\boldsymbol{x} - k). \tag{21}$$

For numerical reasons, it is attractive to work with compactly supported dual functions.

3.1.2 Multiscale transformation

Since $V_j = V_{j-1} \oplus W_{j-1}$, a function $v_j \in V_j$ can uniquely be written as the sum of a function $v_{j-1} \in V_{j-1}$ and a function $w_{j-1} \in W_{j-1}$, i.e.

$$\begin{aligned} v_j(\boldsymbol{x}) &= \sum_k c_{j,k}\phi_{j,k}(\boldsymbol{x}) = v_{j-1}(\boldsymbol{x}) + w_{j-1}(\boldsymbol{x}) \\ &= \sum_l c_{j-1,l}\phi_{j-1,l}(\boldsymbol{x}) + \sum_{e=1}^{3}\sum_l d^e_{j-1,l}\psi^e_{j-1,l}(\boldsymbol{x}). \end{aligned}$$

The following relations show how to pass between these two representations. By (20) and (21),

$$c_{j-1,l} =< v_j, \tilde{\phi}_{j-1,l} >=< v_j, \sum_k \tilde{a}_{k-2l}\tilde{\phi}_{j,k} >= \sum_k \tilde{a}_{k-2l}c_{j,k} \tag{22}$$

and, similarly

$$d^e_{j-1,l} = \sum_k \tilde{b}^e_{k-2l}c_{j,k}. \tag{23}$$

The opposite direction, from the $c_{j-1,l}$ and the $d^e_{j-1,l}$ to the $c_{j,k}$ is achieved by first writing $\tilde{\phi}(2\boldsymbol{x} - k) \in \tilde{V}_1$ in the bases of $\tilde{V}_0$ and $\tilde{W}_0$

$$2\tilde{\phi}(2\boldsymbol{x} - k) = \sum_l a_{k-2l}\tilde{\phi}(\boldsymbol{x} - l) + \sum_{e=1}^{3}\sum_l b^e_{k-2l}\tilde{\psi}^e(\boldsymbol{x} - l)$$

Then, by following the same process as in (22), we obtain

$$c_{j,k} = \sum_l a_{k-2l}c_{j-1,l} + \sum_{e=1}^{3}\sum_l b^e_{k-2l}d^e_{j-1,l}. \tag{24}$$

When applied recursively, these formulae define a Fast Wavelet Transform (FWT), the relations (22) and (23) define the forward transform, while (24) defines the inverse transform. The benefit of such a transformation is that by simply deleting information of very small magnitudes in each subspace W_j, much less data information has to be stored or transmitted.

3.2 B-Spline Wavelet

One of the basic methods for constructing wavelets involves the use of cardinal B-spline functions (or simply B-splines). These are probably the simplest functions with small supports that are most efficient for numerical implementation. Moreover, they give rise to an important class of dual pairs where both scaling functions have compact support. Because our B-splines as the wavelets stemming from them are simply products of the univariate B-splines and univariate wavelets, we first deal with the one-dimensional setting.

3.2.1 Scaling function, wavelet and filter sequences on the real line

We present here, briefly, the main results from the work of Cohen, Daubechies and Feauveau [9]. The two advantages of their construction are: (i) all functions including the dual ones have compact support (ii) the dual functions can be chosen according to some regularity properties.

Scaling function and its dual Let $\lfloor x \rfloor$ ($\lceil x \rceil$) denote the largest (smallest) integer less (greater) than or equal to x and define $N_d = N_1 * \cdots * N_1$ as the d-fold convolution of the characteristic function of the unit interval $[0, 1)$ (box function). Then the scaling function function

$$\varphi_d(x) = N_d(x + \lfloor d/2 \rfloor) \tag{25}$$

is called a B-spline of order d. It satisfies the following refinement equation:

$$\sum_{k=-\lfloor d/2 \rfloor}^{\lceil d/2 \rceil} 2^{1-d} \binom{d}{k + \lceil d/2 \rceil} \varphi_d(2x - k).$$

By writing this equation in the form

$$\varphi_d(x) = \sqrt{2} \sum_k h_k \varphi_d(2x - k), \tag{26}$$

we introduce a sequence (h_k) of coëfficiënts, called the filtersequence $\boldsymbol{h}$. The Fourier transform of the scaling function must then satisfy

$$\hat{\varphi}_d(\omega) = H(\omega/2)\hat{\varphi}_d(\omega/2)$$

where H, the so-called two-scale symbol, is defined by

$$H(\omega) = 2^{-1/2} \sum_k h_k z^k \qquad z = e^{-i\omega}.$$

In the same way, the refinement equation for the dual function reads

$$\tilde{\varphi}_{d,\tilde{d}}(x) = \sqrt{2} \sum_k \tilde{h}_k \tilde{\varphi}_{d,\tilde{d}}(2x - k) \tag{27}$$

and its associated two-scale symbol is defined by

$$\tilde{H}(\omega) = 2^{-1/2} \sum_k \tilde{h}_k$$

Cohen et al. showed that for every $d, \tilde{d} \in \mathbb{N}, \tilde{d} \geq d, d + \tilde{d}$ even, there exist a compactly supported

scaling function $\tilde{\varphi}_{d,\tilde{d}}$ such that $(\varphi_d, \tilde{\varphi}_{d,\tilde{d}})$ form a dual pair (the role of the parameters $d, \tilde{d}$ will be pointed out below). The dual two-scale symbol is given by

$$\tilde{H}(\omega) = e^{-i\kappa\omega/2}(\cos\omega/2)^{\tilde{d}}\left[\sum_{k=0}^{D-1}\begin{pmatrix} D-1+k \\ k \end{pmatrix}(\sin\omega/2)^{2k}\right]$$

where $\tilde{d} \geq 1, d+\tilde{d} = 2D$ is even, and $\kappa = 1$ if d is odd, 0 if d is even. In the previous refinement equations, it is understood that the filter sequences $\boldsymbol{h}$ and $\tilde{\boldsymbol{h}}$ depend on the parameters $d, \tilde{d}$.

Wavelet and its dual The wavelet function ψ and its dual must form a dual pair and verify the biorthogonality conditions. It is known that such candidates are given by

$$\begin{aligned}\psi_{d,\tilde{d}}(x) &= \sqrt{2}\sum_k (-1)^k \tilde{h}_{1-k}\varphi_d(2x-k),\\ \tilde{\psi}_{d,\tilde{d}}(x) &= \sqrt{2}\sum_k (-1)^k h_{1-k}\tilde{\varphi}_{d,\tilde{d}}(2x-k).\end{aligned}$$

which is quite similar to the orthonormal case. As a consequence, it can be shown that for any pair $d, \tilde{d}$ as stated above, we have the following properties

$$\begin{aligned}\int_{\mathbb{R}} x^r \psi_{d,\tilde{d}}(x)dx = 0, \qquad r = 0,\ldots,\tilde{d}-1,\\ \int_{\mathbb{R}} x^r \tilde{\psi}_{d,\tilde{d}}(x)dx = 0, \qquad r = 0,\ldots,d-1.\end{aligned}$$

The wavelets $\psi_{d,\tilde{d}}, \tilde{\psi}_{d,\tilde{d}}$ are said to have vanishing moments of order $\tilde{d}, d$, respectively. The order of vanishing moments governs the compression capacity of a wavelet and the approximation power of the space V_j.

Examples We list $H(\omega)$ and $\tilde{H}(\omega)$ for the two pairs $d = \tilde{d} = 1$ and $d = \tilde{d} = 2$.

- $d = \tilde{d} = 1$ (Haar basis)

$$H(\omega) = \tilde{H}(\omega) = \frac{1}{2}(1+z).$$

- $d = \tilde{d} = 2$ (Hat function)

$$\begin{aligned}H(\omega) &= \frac{1}{4}(z^{-1}+2+z).\\ \tilde{H}(\omega) &= \frac{1}{4}(-\frac{1}{2}z^{-2}+z^{-1}+3+z-\frac{1}{2}z^2)\end{aligned}$$

3.2.2 The bivariate case

The simplest way of generating othogonal or biorthogonal wavelets on $\mathbb{R}^2$ is via tensor products. Given a dual pair $(\phi, \tilde{\phi})$ of univariate scaling functions, the products

$$\begin{aligned}\phi(\boldsymbol{x}) &= \varphi_d(x_1)\varphi_d(x_2)\\ \tilde{\phi}(\boldsymbol{x}) &= \varphi_{d,\tilde{d}}(x_1)\varphi_{d,\tilde{d}}(x_2)\end{aligned}$$

where $\boldsymbol{x} = (x_1, x_2)$, form a dual pair in $L_2(\mathbb{R}^2)$. The three "mother wavelets" are obtained by the products

$$\begin{aligned}\psi^1(\boldsymbol{x}) &= \varphi_d(x_1)\psi_{d,\tilde{d}}(x_2),\\ \psi^2(\boldsymbol{x}) &= \psi_{d,\tilde{d}}(x_1)\varphi_d(x_2),\\ \psi^3(\boldsymbol{x}) &= \psi_{d,\tilde{d}}(x_1)\psi_{d,\tilde{d}}(x_2).\end{aligned}$$

The dual functions are defined analogously. Again, it is understood that the bivariate functions $\phi, \psi^{e=1..3}$ and their duals are depending on the parameter $d, \tilde{d}$. The corresponding masks are obtained from the univariate ones in a straightforward fashion. By (26), we obtain

$$\begin{aligned}\phi(\boldsymbol{x}) &= 2\sum_{k_1,k_2} h_{k_1}h_{k_2}\varphi_d(2x_1 - k_1)\varphi_d(2x_2 - k_2),\\ &= 2\sum_k a_k\phi(2\boldsymbol{x} - k), \qquad k = (k_1, k_2)\end{aligned} \tag{28}$$

and similarly for the other functions.

3.2.3 Interpolation scheme

Given a function $f \in L_2(\mathbb{R}^2)$, we would like to find a "good" approximation of f into the space of reference V_0 corresponding to the finest level of resolution (finest grid). The natural way is to use the biorthogonal projector P_0

$$P_0 f = \sum_l < f, \tilde{\phi}_{0,l} > \phi_{0,l}. \tag{29}$$

However, the computation of inner products with dual wavelets is not an easy task since one needs suitable quadrature rules and in many cases, the dual wavelets are not known explicitly but only via certain functional equations from which the function values have to be computed or approximated. Some of these problems are discussed in [20]. An alternative approach is to use the quasi-interpolation operator of Chui [17]. As shown below, this approach is very advantageous for computational reasons. We expose here the quasi-interpolation scheme on the real line.
For any sequence $\{f(j) = f_j\}$, we want to determine the coëfficiënts $\{c_k\}$ in

$$\sum_k c_k N_d(x + d/2 - k)|_{x=j} = f_j, \qquad j \in \mathbb{Z}. \tag{30}$$

Using the symbol notation

$$\begin{aligned}\tilde{N}_d(z) &= \sum_k N_d(k + d/2)z^k,\\ \tilde{C}(z) &= \sum_k c_k z^k,\\ \tilde{F}(z) &= \sum_k f_k z^k,\end{aligned}$$

we may write eqn (30) as

$$\tilde{C}(z)\,\tilde{N}_d(z) = \tilde{F}(z).$$

The introduction of $\tilde{D}(z) = 1 - \tilde{N}_d(z)$ allows to write formally

$$\tilde{C} = \tilde{F}(1 - \tilde{D})^{-1} = (1 + \tilde{D} + \tilde{D}^2 + \cdots)\tilde{F}.$$

In can be shown that this series converges. This last expression motivates the consideration of the finite sequence $\Lambda_k = \{\lambda_j^{(k)}\}$, defined by

$$\tilde{\Lambda}_k(z) = \sum_j \lambda_j^{(k)} z^j = (1 + \tilde{D} + \tilde{D}^2 + \cdots + \tilde{D}^k). \tag{31}$$

Then the following theorem holds (cf.[17])

Theorem 4 *Let $d \geq 1$. Then for each $k > (d-3)/2$, the linear operator Q_k defined by*

$$(Q_k f)(x) = \sum_l \sum_j \lambda_{l-j}^{(k)} f_j \, N_d(x + d/2 - l) \tag{32}$$

satisfies

$$(Q_k p)(x) = p(x), \tag{33}$$

where p is a polynomial of degree at most $d-1$.

Unfortunately, the operator Q_k is not interpolating because of the truncation in (31). The construction of an interpolating operator needs to define the following spline operator

$$(R_d f)(x) = \frac{1}{N_d(d/2)} \sum_l f(l) N_d(2^{\delta_d} x + d/2 - l), \tag{34}$$

where δ_d denotes the smallest integer bounded below by $\log_2 d - 1$. This operator is both local and interpolatory but is very bad representation of f, since even the constant data function is not reproduced. To construct a bounded linear local operator P that possesses both the polynomial-reproduction property of Q and the interpolatory property of R_d, we consider the blending operation

$$P = R_d + Q_k - R_d \circ Q_k. \tag{35}$$

Of course the operator P can be scaled to yield the operators

$$P^h = \sigma_h \circ P \circ \sigma_{h^{-1}}, \quad h > 0 \tag{36}$$

where we use the notation $(\sigma_h f)(x) = f(h^{-1}x)$ to describe the scaling process. Now, the approximation propertie of the operator P is given in the following theorem.

Theorem 5 *Let K be any compact set in $\mathbb{R}$, and Ω an open set containing K. Then for each bounded $f \in C^d(\Omega)$, there exists a positive constant C, depending only on f and K, such that*

$$\max_{x \in K} |(P^h f - f)(x)| \leq C h^d \tag{37}$$

for all sufficiently small $h > 0$.

The bivariate case is achieved via the usual tensor product.

Examples In the case where $d = 2$, we can choose $k = 1$. Then the lambda's are given by (31)

$$\lambda_j^{(1)} = 2\delta_{j,0} - N_2(1 + j)$$

and we obtain

$$(Q_1 f)(x) = \sum_l f(l) \, N_2(x + 1 - l).$$

In this particular case, it is easy to see that $P = R_2 = Q_1$ and the sequence $\{c_k\}$ is directly given by the values of f at the integer knots: N_2 is an *interpolating* function.

For $d = 3$, we may choose $k = 1$ and we obtain

$$\begin{aligned} \lambda_j^{(1)} &= 2\delta_{j,0} - N_3(3/2 + j) \\ (Q_1 f)(x) &= \sum_l \left\{ -\frac{1}{8} f(l-1) + \frac{5}{4} f(l) - \frac{1}{8} f(l+1) \right\} N_3(x + 3/2 - l) \\ (R_3 f)(x) &= \sum_l \left\{ \frac{4}{3} f(l) \right\} N_3(x + 3/2 - l), \end{aligned}$$

and by using (35)

$$(Pf)(x) = \sum_l \left\{ -\frac{1}{48} f(l \pm 2) - \frac{1}{8} f(l \pm 1) + \frac{79}{24} f(l) \right\} N_3(x + 3/2 - l).$$

3.3 The numerical evaluation of particular solutions

3.3.1 Fast Wavelet Transform and compression

Consider a bounded function of the form $f = \chi_{\Omega_0} f$ where χ_{Ω_0} is the characteristic function of a bounded domain $\Omega_0 \subset \mathbb{R}^2$. An approximation $f_{0;h}$ of f is found by using the operator P^h (see 36):

$$f_{0;h}(\boldsymbol{x}) = P^h f(\boldsymbol{x}) = \sum_k c_{0,k} \phi_{0,k}(h^{-1}\boldsymbol{x}) \tag{38}$$

where the finite set $\{c_{0,k} \neq 0\}$ is derived from the values of f on the grid $h\mathbb{Z}^2$. After applying the forward Fast Wavelet Transform J times, $f_{0;h}$ can be rewritten as

$$f_{0;h}(\boldsymbol{x}) = \sum_k c_{-J,k} \phi_{-J,k}(h^{-1}\boldsymbol{x}) + \sum_{j=-1}^{-J} \sum_{e=1}^{3} \sum_k d_{j,k}^e \psi_{j,k}^e(h^{-1}\boldsymbol{x}) \tag{39}$$

The basic reason why this new representation might be useful is that each wavelet contains information about $f_{0;h}$ essentially at location k and at the scale j. In part of the domain where $f_{0;h}$ has high frequency behaviors or discontinuities, a lot of wavelet coefficients are needed, and where $f_{0;h}$ is smooth, we can use fewer coefficients and still get a good quality of approximation. In other words, the FWT allows us to focus on the most relevant parts of the function. Now, let $\mathcal{I}_\mathcal{M}$ be the set of indices corresponding to the M non-zero wavelet coefficients of $f_{0;h}$. We want to find the "best" approximation of $f_{0;h}$ by using the representation

$$f_{0;h}^N(\boldsymbol{x}) = \sum_k c_{-J,k} \phi_{-J,k}(h^{-1}\boldsymbol{x}) + \sum_{\mathcal{I}_\mathcal{N}} d_{j,k}^e \psi_{j,k}^e(h^{-1}\boldsymbol{x}), \qquad \mathcal{I}_\mathcal{N} \subset \mathcal{I}_\mathcal{M}, \quad N = \#\mathcal{I}_\mathcal{N} \tag{40}$$

Because of the stability condition, we have the inequality

$$\|f_{0;h} - f_{0;h}^N\|_{L_2} \leq C\, h \left(\sum_{\mathcal{I}_\mathcal{M} \setminus \mathcal{I}_\mathcal{N}} (d_{j,k}^e)^2 \right)^{1/2}. \tag{41}$$

This reveals that the best way to pick N wavelet coefficients making the L_2 error as small as possible, is by simply picking the N coefficients with largest absolute value. In the following, $\mathcal{I}_\mathcal{N}$ stands for this set.

3.3.2 The Newton potential

The classical method for obtaining a particular solution $\triangle u_p = f^N_{0;h}$ is to construct the associated Newton potential (see [1]) given by the convolution integral

$$u_p(\boldsymbol{x}) = (2\pi)^{-1} \int_{\mathbb{R}^2} f^N_{0;h}(\boldsymbol{y}) \ln|\boldsymbol{x}-\boldsymbol{y}|\, d^2\boldsymbol{y}. \tag{42}$$

Now let $\{\phi_p, \psi_p^{e=1..3}\}$ be associated potentials of $\{\phi, \psi^{e=1..3}\}$. The particular solution u_p and its normal derivative are then given by

$$u_p(\boldsymbol{x}) = \sum_k c_{-J,k} 2^J h^2 \left(\phi_p(2^{-J}h^{-1}\boldsymbol{x}-k) + \frac{\ln(2^J h)}{2\pi}\right) + \sum_{\mathcal{I}_\mathcal{N}} d^e_{j,k} 2^{-j} h^2 \psi^e_p(2^j h^{-1}\boldsymbol{x}-k), \tag{43}$$

$$\frac{\partial u_p}{\partial n}(\boldsymbol{x}) = \sum_k c_{-J,k} h \frac{\partial \phi_p}{\partial n}(2^{-J}h^{-1}\boldsymbol{x}-k) + \sum_{\mathcal{I}_\mathcal{N}} d^e_{j,k} h \frac{\partial \psi^e_p}{\partial n}(2^j h^{-1}\boldsymbol{x}-k). \tag{44}$$

3.3.3 Numerical evaluation of ϕ_p and its derivative

The potential ϕ_p is given by the convolution integral

$$\phi_p(\boldsymbol{x}) = (2\pi)^{-1} \int_{\mathbb{R}^2} \phi(\boldsymbol{y}) \ln|\boldsymbol{x}-\boldsymbol{y}|\, d^2\boldsymbol{y} \tag{45}$$

and the regularity property can be derived from the following theorem (cf.[21])

Theorem 6
(a) Let ϕ be a compactly supported and bounded function having, then ϕ_p is in class $C^1(\mathbb{R}^2)$.
(b) If in (a) there is an open set D on which ϕ_p is continuous and satisfies a Hölder condition of the form

$$|\phi(\boldsymbol{x}) - \phi(\boldsymbol{y})| \le C|\boldsymbol{x}-\boldsymbol{y}|^\beta \qquad 0 < \beta \le 1, \quad (\boldsymbol{x},\boldsymbol{y}) \in D \times D$$

then $\phi_p|_D$ is in class $C^2(D)$ and satisfies the Poisson equation $\triangle\phi_p = \phi$ at all points of D.

Now, let $\alpha = (\alpha_1, \alpha_2)$ be a multi-index of non negative integers. We write $D^\alpha\phi$ to denote the usual mixed partial derivative of ϕ of order $|\alpha| = \alpha_1 + \alpha_2$. It is easy to see that the function $D^\alpha\phi$ $(0 \le |\alpha| \le d-1$ is bounded and compactly supported. It follows from (a) that $D^\alpha\phi_p$ belongs to $C^1(\mathbb{R}^2)$. Therefore, $\phi_p \in C^d(\mathbb{R}^2)$ and $\partial\phi_p/\partial n \in C^{d-1}(\mathbb{R}^2)$. Of course, a better statement can be achieved by condidering (b) in some open subset of $\mathbb{R}^2$.
The singularity of the integrand can be removed by using polar coordinates,

$$\phi_p(\boldsymbol{x}) = (2\pi)^{-1} \int_{\operatorname{supp}\phi} \phi(\boldsymbol{y}(r,\theta)) r \ln r\, dr d\theta,$$

where $\operatorname{supp}\phi = [-\lfloor d/2\rfloor, d - \lfloor d/2\rfloor]^2$ denotes the support of the generator ϕ. This last integral is regular and is therefore computed via standard gaussian quadrature formulas. By observing that the restrictions of N_d to any interval $[k, k+1)$ is in π_{d-1} (i.e. the collection of algebraic polynomials of degree at most $d-1$), it is judicious for better accuracy to divide the domain integral into d^2 domain integrals

$$\phi_p(\boldsymbol{x}) = (2\pi)^{-1} \sum_{k,k'=0}^{d-1} \int_{I_k \times I_{k'}} \phi(\boldsymbol{y}(r,\theta)) r \ln r\, dr d\theta, \tag{46}$$

where $I_k = [k - \lfloor d/2 \rfloor, k + 1 - \lfloor d/2 \rfloor]$, $0 \leq k \leq d-1$. For $d \geq 2$, the x_1-derivative is obtained from

$$\frac{\partial \phi_p}{\partial x_1}(\boldsymbol{x}) = (2\pi)^{-1} \sum_{k,k'=0}^{d-1} \int_{I_k \times I_{k'}} \frac{\partial \phi}{\partial y_1}(\boldsymbol{y}(r,\theta)) r \ln r \, dr d\theta \tag{47}$$

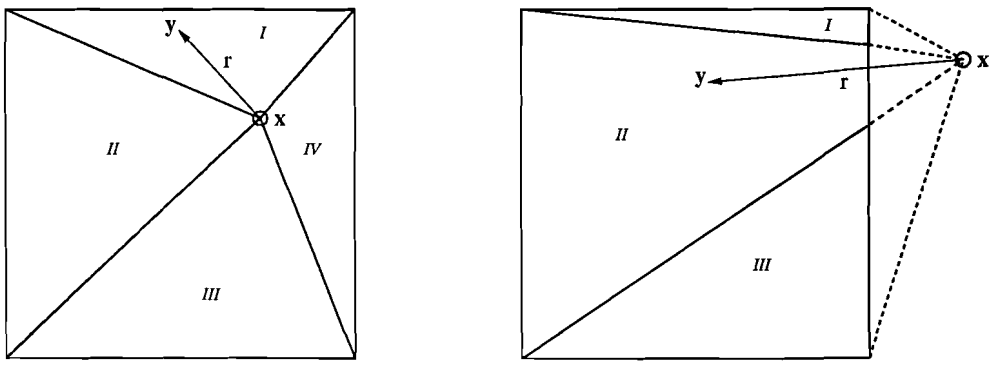
and the gradients of ϕ are calculated by using the following property for the univariate B-spline

$$\frac{dN_d}{dx} = N_{d-1}(x) - N_{d-1}(x-1).$$

The numerical evaluation of (46) and (47) involves integral the type

$$\int_{I_k \times I_{k'}} g(\boldsymbol{y}(r,\theta)) r \ln r \, dr d\theta,$$

where g stands for either ϕ or its derivative. Figure 1 illustrates the discretization of the unit square domain $I_k \times I_{k'}$ according to the position of the evaluation point $\boldsymbol{x}$ with respect to the domain of integration. In each zone, the integration is performed by using 30 nodes in both r and θ directions. As explained before a high accuracy is expected since the restriction of g to the square domain is a polynomial of order $d-1$.



x inside the square domain

x outside the square domain

Figure 1: Integration scheme

The other potentials $\psi_p^{e=1..3}$ and their derivatives are directly evaluated by using equations (19)

$$\psi_p^e(\boldsymbol{x}) = \frac{1}{2} \sum_k b_k^e \phi_p(2\boldsymbol{x} - k). \tag{48}$$

By using symmetry properties about the line $x_1 = x_2$, it is easy to show that

$$\begin{aligned} \phi_p(x_1, x_2) &= \phi_p(x_2, x_1), \\ \psi_p^2(x_1, x_2) &= \psi_p^1(-x_2, x_1), \\ \psi_p^3(x_1, x_2) &= \psi_p^3(x_2, x_1) \end{aligned}$$

and consequently

$$\begin{aligned}
\frac{\partial \phi_p}{\partial x_2}(x_1,x_2) &= \frac{\partial \phi_p}{\partial x_1}(x_2,x_1),\\
\frac{\partial \psi_p^2}{\partial x_1}(x_1,x_2) &= \frac{\partial \psi_p^1}{\partial x_2}(-x_2,x_1),\\
\frac{\partial \psi_p^2}{\partial x_2}(x_1,x_2) &= -\frac{\partial \psi_p^1}{\partial x_1}(-x_2,x_1),\\
\frac{\partial \psi_p^3}{\partial x_2}(x_1,x_2) &= \frac{\partial \psi_p^3}{\partial x_1}(x_2,x_1).
\end{aligned}$$

3.3.4 Storage

The evaluation of the previous integrals is of course time consuming. For this reason, it is advantageous to "store" the set $\mathcal{S} = \left\{\phi_p, \psi_p^1, \psi_p^3, \frac{\partial \phi_p}{\partial x_1}, \frac{\partial \psi_p^1}{\partial x_1}, \frac{\partial \psi_p^1}{\partial x_2}, \frac{\partial \psi_p^3}{\partial x_1}\right\}$ (the other functions are deduced by symmetry). Moreover, in our computations for points x far from the integration domain, we may use simple asympotic formulas as given in the next paragraph.

Asymptotic behavior In our analysis below, we apply arithmetic to vectors $\boldsymbol{x} \in \mathbb{R}^2$ by interpreting them as points in the complex plane. Thus, we have that $\ln \boldsymbol{x} = \ln|\boldsymbol{x}| + i \arg(\boldsymbol{x})$. So $\ln|\boldsymbol{x}|$ is the real part of $\boldsymbol{x}$, denoted as $\ln|\boldsymbol{x}| = \Re(\ln \boldsymbol{x})$.
Let's first observe that for any $|\boldsymbol{x}| > |\boldsymbol{y}|$, we have the following expansion

$$\begin{aligned}
\ln|\boldsymbol{x}-\boldsymbol{y}| &= \Re \ln(x-y)\\
&= \ln|\boldsymbol{x}| - \Re\left[\sum_{n=1}^{\infty} \frac{1}{n}\left(\frac{y}{x}\right)^n\right]
\end{aligned}$$

where $x = |\boldsymbol{x}|e^{i\theta_x}$ and $y = |\boldsymbol{y}|e^{i\theta_y}$. Substituting this last expression into the convolution integral yields

$$\phi_p(\boldsymbol{x}) = (2\pi)^{-1}\left(\ln|\boldsymbol{x}|P_0(\boldsymbol{x}) - \sum_{n=1}^{\infty} \frac{1}{n|\boldsymbol{x}|^{2n}} P_n(\boldsymbol{x})\right), \tag{49}$$

where P_n is the polynomial given by

$$P_n(\boldsymbol{x}) = \sum_{\substack{q,q'=0\\ q+q' \text{even}}}^{n} (-1)^{\frac{3q'+q}{2}} \binom{n}{q}\binom{n}{q'} x_1^{n-q}x_2^q \int_{\text{supp}\,\phi} y_1^{n-q'}y_2^{q'}\phi(\boldsymbol{y})\, d^2\boldsymbol{y}$$

The expansion for the first derivative can be derived by differentiating (49). Yet, to obtain a better formulae, it is convenient to split the derivative of ϕ into two terms

$$\frac{\partial \phi}{\partial y_1}(\boldsymbol{y}) = \phi^+(\boldsymbol{y}) - \phi^-(\boldsymbol{y}) \tag{50}$$

where

$$\begin{aligned}
\phi^+(\boldsymbol{y}) &= N_{d-1}(y_1 + \lfloor d/2 \rfloor)N_d(y_2 + \lfloor d/2 \rfloor)\\
\phi^-(\boldsymbol{y}) &= N_{d-1}(y_1 - 1 + \lfloor d/2 \rfloor)N_d(y_2 + \lfloor d/2 \rfloor)
\end{aligned}$$

and then apply the expansion formulae separately. The formulaes for the other functions are obtained by (49).

Storage with finite-element approximation By using simple geometric transformations (translation, symmetry and rotation), it can be seen that the functions of $\mathcal{S}$ only need to be stored in a square domain $D_a = [0, a] \times [0, a]$ where the parameter a depends on the function to be approximated. Good storage can be realized if the following criteria are satisfied:
(i) The access to the approximated values must be efficient in terms of arithmetic operations.
Owing to the domain of storage and in view of avoiding complicated geometric transformations, the use of quadrilateral Lagragian-type elements is suitable.
(ii) The stored function give a good approximation of the original one.
The order of the element must be chosen according to the regularity of the function in the domain of storage. Nevertheless, this consideration is only "asymptotically" true (i.e. when the characteristic length size of the element tend to zero) and in practice only low order (linear, quadratic and cubic) are worth consideration since higher order penalises (i).
(iii) The storage must be economic.
The parameter a must be as small as possible such that for any $|x| \geq a$, the asymptotic expressions of the type (49) can replace the original functions.

4 Practical implementation for $d = \tilde{d} = 2$

We present a practical realization of a 2D Poisson's equation solver using the ingredients from the previous section for the particular case $d = \tilde{d} = 2$. Let's recall that the parameter d governs the compression capacity of the wavelet and the approximation degree of the B-spline (see 37) whereas $\tilde{d}$ governs the regularity of the dual functions. As this last point is not crucial for us (we do not use the biorthogonal projector (29)), it is advantageous for computational reasons to choose $\tilde{d}$ as small as possible since the length of the filter sequence $\tilde{h}$ grows as $\tilde{d}$ increases (this is true also for d and h, see [9]). In Figure 2, we plot the scaling function φ_2 and the wavelet $\psi_{2,2}$. Note that the supports are respectively $[-1, 1]$ and $[-1, 2]$ and the wavelet is symmetric about $x = 1/2$.

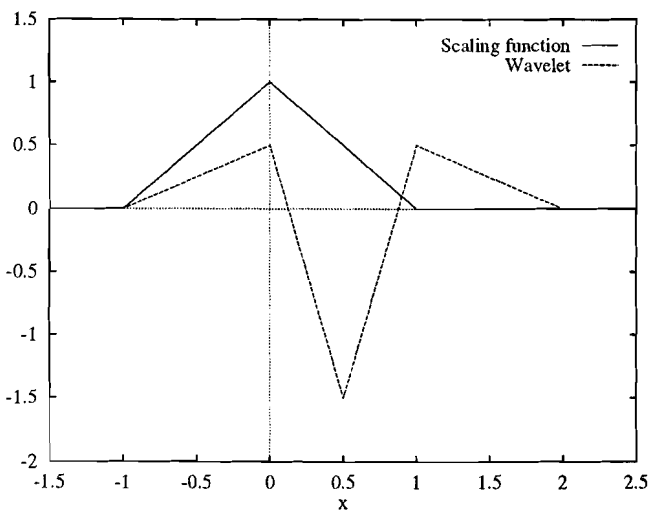


Figure 2: Scaling function and wavelet for $d = \tilde{d} = 2$

4.1 Numerical evaluation and storage of $\mathcal{S}$

Figure 3 shows the graph of ϕ_p, $\frac{\partial \phi_p}{\partial x_1}$, ψ_p^1 and ψ_p^3. Note that functions of $\mathcal{S}$ excepted ϕ_p all tend to zero as $|x|$ goes to infinity.

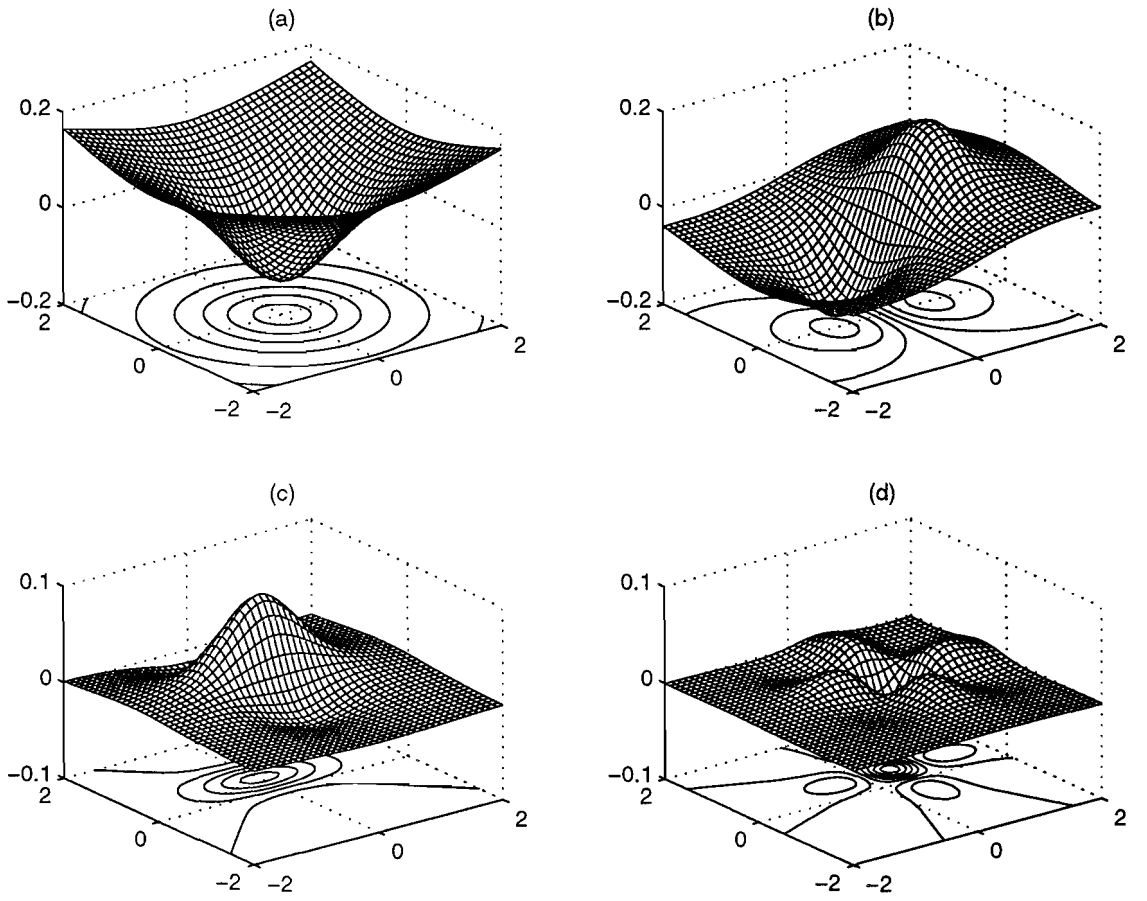


Figure 3: Particular solutions (a) ϕ_p, (b) $\frac{\partial \phi_p}{\partial x_1}$, (c) ψ_p^1, (d) ψ_p^3

By applying (49) and (50) with $d = 2$, we can define (for $|x|$ sufficiently large) the following truncated series

$$
\begin{aligned}
(\phi_p)_N(x) &= (2\pi)^{-1}\left(\ln|x| - \sum_{n=1}^{N} \frac{1}{2n|x|^{4n}} \sum_{q=0}^{n} A_{n,q}\, x_1^{2(n-q)} x_2^{2q}\right), \\
A_{n,q} &= \sum_{q'=0}^{n} \binom{2n}{2q} \binom{2n}{2q'} \frac{(-1)^{q+q'}}{(n-q'+1)(2(n-q')+1)(q'+1)(2q'+1)}
\end{aligned}
$$

and

$$
\begin{aligned}
\left(\frac{\partial \phi_p}{\partial x_1}\right)_N (x) &= (2\pi)^{-1}\left(\ln|x+(s,0)| - \sum_{n=1}^{N} \frac{1}{2n|x+(s,0)|^{4n}} \sum_{q=0}^{n} (x_1+s)^{2(n-q)} x_2^{2q} B_{n,q}\right) \\
&- (2\pi)^{-1}\left(\ln|x-(s,0)| - \sum_{n=1}^{N} \frac{1}{2n|x-(s,0)|^{4n}} \sum_{q=0}^{n} (x_1-s)^{2(n-q)} x_2^{2q} B_{n,q}\right),
\end{aligned}
$$

$$B_{n,q} = \sum_{q'=0}^{n} \begin{pmatrix} 2n \\ 2q \end{pmatrix} \begin{pmatrix} 2n \\ 2q' \end{pmatrix} \frac{(-1)^{q+q'}}{(2(n-q')+1)(q'+1)(2q'+1)4^{n-q'}},$$

where s stands for the shift $s = 1/2$. In order to examine the accuracy of the previous series, we evaluate numerically (see Figure 4) the following auxiliary functions:

$$\begin{aligned} E1(r,N) &= \log_{10}\left(\max_{\theta\in[0,2\pi]} |(\phi_p)_N(r,\theta) - \phi_p(r,\theta)|\right), \\ E2(r,N) &= \log_{10}\left(\max_{\theta\in[0,2\pi]} \left|\left(\frac{\partial\phi_p}{\partial x_1}\right)_N(r,\theta) - \frac{\partial\phi_p}{\partial x_1}(r,\theta).\right|\right). \end{aligned}$$

The minimum distance $r = 1.5$ is taken just above the theoretical distance $\min\{r > |y|, y \in \text{supp}\,\phi\} = \sqrt{2}$. It is clear that the accuracy improves as N increases but this gives rise to more computation since the number of terms in the series grows as N^2. To keep a reasonable number of terms (let's say $N \leq 5$), we impose a threshold of accuracy: $10^{-5.5}$ for the potential and 10^{-5} for the derivative. The number of terms (i.e. N) is then found such that the two parameters previously defined are less than the threshold.

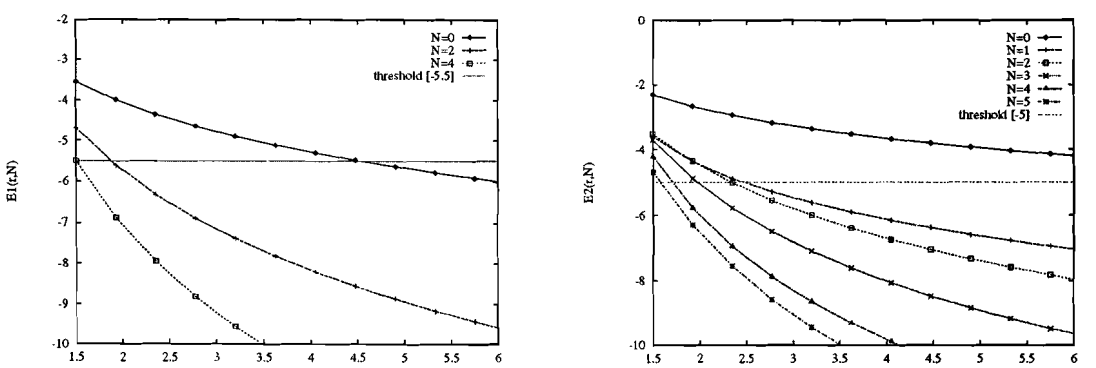


Figure 4: Asymptotic behavior for $(\phi_p)_N$ and $(\partial\phi_p/\partial x_1)_N$

The regularity of the two functions (we have *at least* $\phi_p \in C^2(D_{3/2})$ and $\partial\phi_p/\partial x_1 \in C^1(D_{3/2})$) suggests the use a piecewise bilinear Lagrange polynomial for the storage in the domain $D_{3/2} = [0,3/2] \times [0,3/2]$ (for more details, see chapter 5 in [22]). However, numerical tests showed that the biquadratic Lagrange polynomial gives a better approximation (in fact, the use of the linear approximation has been shown to be unrealistic). The biquadratic shape function will be considered here for a mesh spacing (i.e. distance between two nodes) $h_s = 0.01$.
The asymptotic series for the potentials ψ_p^1 and ψ_p^3 have a "slow" convergence and the leading order terms ($N = 1$)

$$\begin{aligned} (\psi_p^1)_1(x) &= -\frac{3}{32\pi|x-(0,s)|^4}((x_2-s)^2 - x_1^2), \\ (\psi_p^3)_1(x) &= \frac{27}{256\pi|x-(s,s)|^8}((x_1-s)^4 - 6(x_1-s)^2(x_2-s)^2 + (x_2-s)^4) \end{aligned}$$

are sufficiently accurate for $|x-(0,s)| \geq 3$ and $|x-(s,s)| \geq 3$ (the minimal theoretical distances

are $\sqrt{3.25} = 1.803...$ for ψ_p^1 and $\sqrt{4.5} = 2.121...$ for ψ_p^3). These functions are stored in the domain $D_3 = [0,3] \times [0,3]$ using the same shape functions as introduced above for mesh spacing $h_s = 0.01$. The corresponding derivatives are obtained by differentiating the previous formulas and stored in the same manner. The total cost of storage is $(2 * 150^2 + 5 * 300^2) \times 4$ bytes ≈ 2 Megabytes.

4.2 Domain extension

As the function $f = \chi_{\Omega_0} f$ contains artificial singularities at the boundary, a lot of non negligible wavelet coefficients are likely to increase in a certain neighbourhood of the boundary and the benefit of the FWT may be drastically reduced or even completely deleted. To avoid this major drawback, we restrict ourselves to a rectangular domain $\Omega_0 = [-A, A] \times [-B, B]$ and consider $f = \chi_{\Omega_0^{ext}} f$ where Ω_0^{ext} is the extended domain $\Omega_0^{ext} = [-(A + A'), A + A'] \times [-(B + A'), B + A']$. The original domain Ω must fit in Ω_0 as shown in Figure 5.

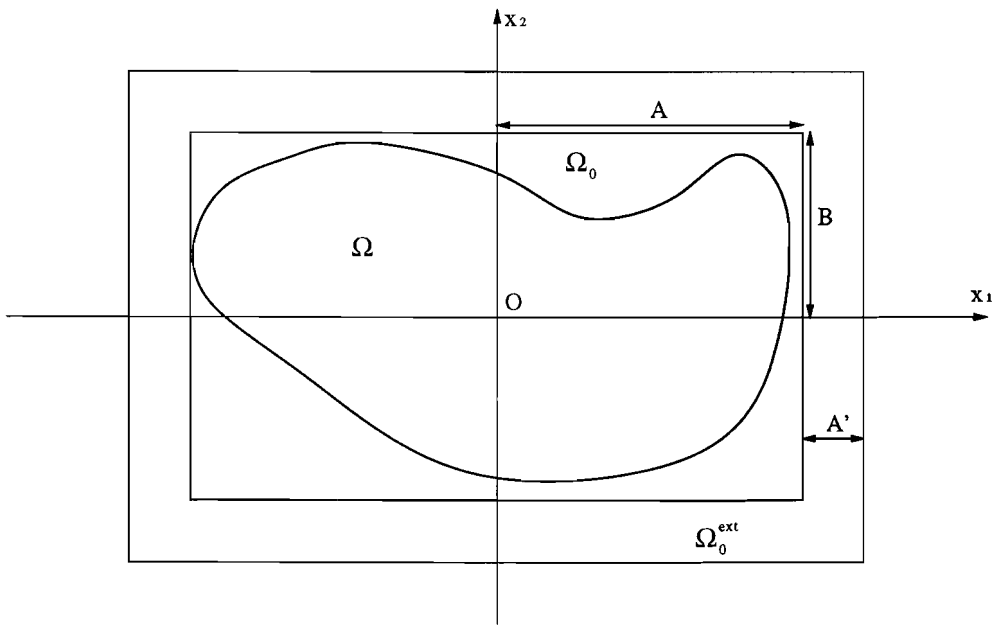


Figure 5: Domain extension

The parameter A' depends on the coarsest scale $-J$ and the domain mesh size h and is found such that the "wavelet" representation (39) of $f_{0;h}|_{\Omega_0}$ (i.e. the restriction of $f_{0;h}$ in Ω_0) is not altered by the artificial singularities at the boundary of Ω_0^{ext}. We can show that A' is simply given by the following recursion formula

$$\begin{aligned} A'_{(J)} &= 2A'_{(J-1)} + 2h, \\ A'_{(0)} &= h. \end{aligned}$$

Finally, the particular solution u_p is the associated potential of

$$f_{0;h}^{n_d}|_{\Omega_0}(\boldsymbol{x}) = \sum_{\mathcal{J}^0} c_{-J,k}\phi_{-J,k}(h^{-1}\boldsymbol{x}) + \sum_{\mathcal{I}^0_{n_d} \subset \mathcal{I}^0} d^e_{j,k}\psi^e_{j,k}(h^{-1}\boldsymbol{x}), \tag{51}$$

where $\mathcal{J}^0$ and $\mathcal{I}^0$ are the sets of indices

$$\mathcal{J}^0 = \{k \mid \operatorname{supp}\phi_{-J,k} \cap \Omega_0 \neq \emptyset\}, \quad n_c = \#\mathcal{J}^0,$$

x_1	x_2	computed	exact
1.5	0.00	0.3500	0.3500
1.2	0.35	0.4140	0.4140
0.6	0.45	0.5660	0.5660
0.0	0.45	0.6380	0.6380
0.9	0.00	0.6380	0.6380
0.3	0.00	0.7820	0.7820
0.0	0.00	0.8000	0.8000

Table 1: Solutions of $\triangle u = -2$ on Ω

x_1	x_2	computed	exact
1.5	0.00	0.1875	0.1875
1.2	0.35	0.1774	0.1774
0.6	0.45	0.1213	0.1213
0.0	0.45	0.0000	0.0000
0.9	0.00	0.2051	0.2051
0.3	0.00	0.0838	0.0838
0.0	0.00	0.0000	0.0000

Table 2: Solutions of $\triangle u = -x_1$ on Ω

$$\mathcal{I}^0 \quad = \quad \{(e,j,k) \mid \operatorname{supp} \psi^e_{j,k} \cap \Omega_0 \neq \emptyset\}$$

and $\mathcal{I}^0_{n_d}$ contains the n_d largest $|d^e_{j,k}|$ such that $(e,j,k) \in \mathcal{I}^0$. We note that n_c is not an arbitrary parameter and depends on J, h and the domain Ω_0 whereas n_d is to be assigned by the user.

4.3 Numerical examples

Three problems of increasing complexity are considered. Computations were performed on a SUN WorkStation 300 MHz. The algorithm for the computation of the particular solution was implemented in single precision, the algorithm for the BEM Laplace's solver was implemented in double precision.

Example 1 To test the effectiveness of the method we solve the Poisson's equation on the ellipse $\Omega = \{(x,y)|x^2/4 + y^2 \leq 1\}$ with the boundary conditions $u|_\Gamma = 0$. For f, we chose the three values $f = -2, -x$ and $-x^2$. These problems have been solved by Golberg [23][24] Partdridge, Brebbia and Wrobel [2] and Alessandri and Tralli [6]. Numerical results comparing these four different approaches are given in Tables 8.4-8.8 in [25].
In our calculations, we use 20 quadratic elements for the boundary. Solutions for the two first cases are given in Tables 1 - 2. These results have been obtained with $h = 1, J = 1$, $(n_c = 15)$ and $n_d = 0$. For convenience, all numbers have been rounded to four decimal places. The excellent accuracy obtained is not surprising since the 2^{nd} order B-spline approximates exactly any constant or linear profile.
For the quadratic profile, three tests have been carried out: in all cases, $h = 0.01$ and $n_d = 0$ and we chose (a) $J = 7$ $(n_c = 15)$, (b) $J = 6$ $(n_c = 45)$, (c) $J = 5$ $(n_c = 135)$. Solutions are given in Table 3. Note that the test (a) gives reasonable results with very few coefficients. This is because these coefficients (obtained after 7 forward FWT) keep the average value of the function on the

x_1	x_2	comp.(a)	comp.(b)	comp.(c)	exact
1.5	0.00	0.2547	0.2602	0.2599	0.2598
1.2	0.35	0.2137	0.2198	0.2201	0.2201
0.6	0.45	0.1494	0.1432	0.1437	0.1437
0.0	0.45	0.0963	0.1031	0.1036	0.1037
0.9	0.00	0.2422	0.2406	0.2402	0.2402
0.3	0.00	0.1509	0.1517	0.1513	0.1514
0.0	0.00	0.1289	0.1360	0.1365	0.1366

Table 3: Solutions of $\triangle u = -x_1^2$ on Ω

finest grid $f_{0;h}|_{\Omega_0}$. For example, by choosing the parameters as in the two first cases, the solution would be disatrous even if the computational cost is roughly the same.

Example 2 We consider the problem

$$\begin{aligned} \triangle u(\boldsymbol{x}) &= 2e^{x_1-x_2}, \qquad \boldsymbol{x} \in \Omega, \\ u(\boldsymbol{x}) &= e^{x_1-x_2} + e^{x_1}\cos x_2, \qquad \boldsymbol{x} \in \Gamma, \end{aligned}$$

where Ω is the same ellipse as introduced in the previous section. The solution in Ω is simply given by $u(\boldsymbol{x}) = e^{x_1-x_2} + e^{x_1}\cos x_2$. This problem has been chosen so we could compare our results with previously published BEM results in [24]. The boundary is discretized with 40 quadratic elements. To compute the particular solution, we chose $h = 0.01$ and considered the four following tests: (a) $J = 7$ ($n_c = 15$), (b) $J = 6$ ($n_c = 45$), (c) $J = 5$ ($n_c = 135$), (d) $J = 4$ ($n_c = 405$). In all cases, the wavelet coefficients are neglected ($n_d = 0$). In Figure we show graphs of the absolute value of the error (in decimal logarithmic scale) on the x_1-axis. Here again, reasonable results (maximum 5 % of relative error) are obtained with very few coefficients (case (a)). The accuracy obtained with test(d) is almost comparable with Golberg's results (see Table 6 in [24]) using the Multiquadrics interpolation. Note that the computational cost (in term of CPU time) for all these tests are similar and doesn't exceed 10 seconds (some more detailed results are presented in the next example).

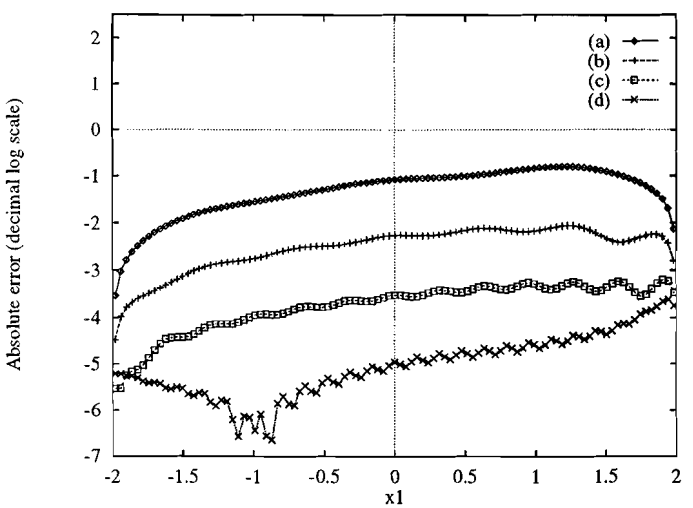


Figure 6: Absolute errors along the x_1-axis for Example 2

x_1	x_2	comp.(a)	comp.(b)	comp.(c)	Maple V
0.00	0.04	3.670	4.007	3.990	3.989
0.76	-0.44	0.762	0.825	0.814	0.814
0.28	0.00	3.403	3.901	3.889	3.889
0.32	-0.68	1.577	1.714	1.703	1.703
0.52	0.48	1.775	2.045	2.020	2.020
0.96	0.24	0.066	0.067	0.068	0.068
–	CPU	10.4 s	11.4 s	19.9 s	–

Table 4: Solutions ($\times 10^{-2}$) for Example 3

Example 3 In the previous examples, the source terms f are infinitely smooth and best results have been obtained by only considering the coefficients for the scaling function at different levels of compression. To see the benefit of the wavelet coefficients we consider the following problem

$$\begin{aligned} \triangle u(\boldsymbol{x}) &= f(r) = -\left(1 + 50|r - 1/2|\right)^{-1}, \qquad \boldsymbol{x} \in \Omega \\ u(\boldsymbol{x}) &= 0, \qquad \boldsymbol{x} \in \Gamma, \end{aligned}$$

where $r = (x_1^2 + x_2^2)^{1/2}$ and Ω is the unit circle. For this particular problem, the solution u is radial and is given by the following integral

$$u(r) = \ln r \int_0^r f(t)t\,dt + \int_r^1 f(t)t \ln t\,dt, \qquad r < 1. \tag{52}$$

The calculations were done using 40 quadratic elements for the boundary. Internal values are computed at approximately 2000 nodes. Three tests have been carried out by fixing the parameters $h = 0.01$ and $J = 6 (n_c = 25)$ and by choosing respectively (a) $n_d = 10$, (b) $n_d = 100$, (c) $n_d = 1000$. These different values for n_d are estimated after examination of the decreasing rearangement of the wavelet coefficients. In Table 4, we show computed values for u at selected points. Results in the last column are obtained from (52) by using MAPLE V. The last row illustrates the efficiency of the method (the cost for the 6 forward FWT is about 1 second). A plot of u is shown in Figure 7. Note the symmetry of the solution, as expected.

5 Conclusion

In this paper, we have shown how to replace the traditional Radial Basis Functions by B-spline wavelets [9] to solve 2D Poisson's equation. As there is no analytical form for the associated particular solutions, these last are stored in a square domain by using finite-element approximation. Asymptotic expressions are then available outside of the domain of "storage". The main advantages of such a construction are (i) the computational cost to interpolate the source term is negligible even for high degree of freedom, this is not the case when using the RBFs since the interpolation matrix (10) has to be inverted, (ii) the Fast Wavelet Transform permits to obtain high level of compression, especially for smooth function, and a good accuracy (4 digits) can be obtained even for "singular" functions, (iii) the global cost, in terms of CPU time, is relatively cheap: 5 to 30 seconds according to the complexity on a Sun Worstation 300 Mhz.
Although we have only considered linear approximation ($d = 2$) in the previous applications, the use of higher order might be fruitful especially when dealing with large scale problems and smooth functions.

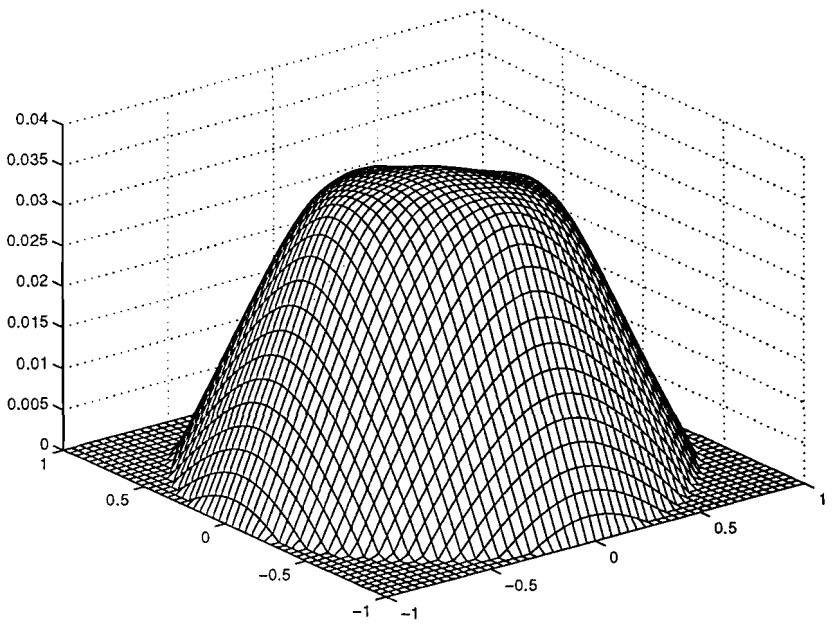


Figure 7: Solution for Example 3 (c)

References

[1] Atkinson KE. The numerical evaluation of particular solutions for Poisson's equation. IMA J. Numerical Analysis 1985;5:319-338.

[2] Partridge PW, Brebbia CA, Wrobel LC. The Dual Boundary Element Method. Computational Mechanics Publications, Elsevier 1992.

[3] Ramachandran PA. Boundary Element Methods in Transport Phenomena. Computational Mechanics Publications, Elsevier 1994.

[4] Zhu SP. Time-dependent reaction-diffusion problems and the LTDRM approach in Boundary Integral Methods: Numerical and Mathematical aspects. Computational Mechanics Publications/WIT Press 1999.

[5] Kassab AJ, Nordlund RS. Efficient implementation of the Fourier dual reciprocity boundary element method using two-dimensional fast Fourier transforms. Engineering Analysis with Boundary Elements 1993;12:93-102.

[6] Alessandri CA and Tralli A. A spline based-approach for avoiding domain integrations in BEM. Computers and Structures 1991;41:859-868.

[7] Nardini D and Brebbia CA. A new approach to free vibration analysis using boundary elements. Computational Mechanics Publications, Southampton 1982.

[8] Golberg MA. The numerical evaluation of particular solutions in the BEM - A Review. Boundary Elements Communications 1995;6:99-106.

[9] Cohen A, Daubechies I, Feauveau J. Bi-orthogonal bases of compactly supported wavelets. Comm. Pure and Applied Math 1992;45:485-560.

[10] Brebbia CA and Dominguez J. Boundary Elements - An Introductory Course. Computational Mechanics Publications, Mc Graw-Hill 1989.

[11] Micchelli CA. Interpolation of scaterred data: distance matrices and conditionally positive definite functions. Constructive Approximation 1986;2:11-22.

[12] Powell MJD. Radial basis functions for multivariable interpolation: a review. Numerical Analysis 1987:223-241. Griffiths DF and Watson GA editors, Longman Scientific & Technical (Harlow)

[13] Jumarhon B, Amini S, Chen K. On the boundary element dual reciprocity method. Engineering Analysis with Boundary Elements 1997;20(3):387-96.

[14] Powell MJD. The uniform convergence of thin plate spline interpolation in two dimensions. Numerische Mathematik 1994;68:228-255.

[15] Levesley J. Pointwise estimates for multivariate interpolation using conditionally positive definite functions. University of Leicester 1994.

[16] Wu Z, Schaback R. Local error estimates for radial basis function interpolation of scattered data. IMA J. Numerical Analysis 1993;13:13-27.

[17] Chui CK. An introduction to wavelets. Academic Press 1992.

[18] Dahmen W. Wavelet and multiscale methods for operator equation. Acta Numerica 1997;6:55-228.

[19] Jawerth B, Sweldens W. An overview of wavelet based multiresolution analyses. SIAM rev. 1994;36(3):377-412.

[20] Barinka A, Barsch T, Dahlke S, Konik M. Some remarks on quadrature formulas for refinable functions and wavelets. Preprint.

[21] Doob JL. Classical potential theory and its probalistic counterpart. Springer-Verlag, New-York, 1984.

[22] Prenter PM. Splines and variational methods. John Wiley & Sons editors, 1975.

[23] Golberg MA. The method of fundamental solutions for Poisson's equation. Engineering Analysis with Boundary Elements 1995;16:205-213.

[24] Golberg MA, Chen CS, Karur SR. Improved multiquadric for approximations for partial differential equations. Engineering Analysis with Boundary Elements 1996;18:9-17.

[25] Golberg MA, Chen CS. Discrete projection methods for integral equations. Computational Mechanics Publications 1997.