The numerical evaluation of particular solutions for Poisson’s equation – a revisit

C.S. Chen¹, A.S. Muleshkov¹, M.A. Golberg²
¹Department of Mathematical Sciences, University of Nevada, Las Vegas, NV 89154, U.S.A.
E-Mail: chen@nevada.edu, muleshko@nevada.edu
²2025 University Circle, Las Vegas, NV 89119, U.S.A.
E-Mail: mag741@aol.com

Abstract

An analytic particular solution for Poisson’s equation in 2D has been constructed for polynomial forcing terms. When the forcing terms contain non-polynomials, Taylor series expansion is used to approximate the forcing terms. A symbolic computational algorithm using Mathematica has been implemented. No matrix inversion is required in evaluating particular solutions. The numerical results are highly accurate and efficient.

1 Introduction

The traditional method for evaluating a particular solution of Poisson’s equation is to construct the associated Newton potential through domain integration [1]. However, singularities in the integrand and irregular shapes of the boundary make it difficult, if not impossible, to evaluate the domain integral. Moreover, 70%-80% of the computational time is consumed in the numerical integration. Various numerical schemes have been proposed to alleviate this problem. In the BEM literature, the dual reciprocity method (DRM) [7] has been widely applied to avoid domain integration in solving various kinds of partial differential equations. In recent years, radial basis functions (RBFs) have become standard basis functions for interpolating the forcing term of the governing equation with excellent results. However, with large numbers of interpolation points, the matrix induced by RBFs using collocation becomes ill-conditioned unless the matrix is suitably preconditioned. Techniques for preconditioning are available but not easy to implement for scattered multivariable data.
It is the purpose of this paper to completely eliminate the problem of matrix inversion by using polynomial approximation in evaluating particular solutions. First let us consider

\[ \Delta u(x, y) = f(x, y), \quad (x, y) \in D, \tag{1} \]
\[ u(x, y) = g(x, y), \quad (x, y) \in \partial D, \tag{2} \]

where \( f \) is a polynomial. In his paper, Atkinson [1] had obtained the particular solution of (1) in near closed form. Our goal is to construct an analytical particular solution of (1). Based on this, we then extend our algorithm to \( f \) when \( f \) is not a polynomial. In recent years, symbolic computational software, such as Mathematica and Maple, have advanced in such a rapid pace that it becomes feasible, sometimes desirable, in numerical computation. They are available in various platforms including supercomputers. Our improvement over Atkinson’s algorithm allows the implementation of symbolic languages, due to the ease of manipulation of polynomials symbolically. Another advantage of our approach, as we shall see, is the proposed algorithm is highly accurate and efficient. To demonstrate the effectiveness of the proposed method, we give an example of Poisson’s equation in 2D using symbolic computation.

2 The Governing Equation

Let us consider Poisson’s equation in (1)-(2) where \( D \subseteq \mathbb{R}^2 \) is a bounded open nonempty domain with sufficiently regular boundary \( \partial D \). (1)-(2) can be converted to a homogeneous equation using the substitution \( v = u - u_p \) where \( u_p \) is a particular solution of the nonhomogeneous equation

\[ \Delta u_p(x, y) = f(x, y). \tag{3} \]

Note that \( u_p \) does not necessarily satisfy the boundary condition (2). Then the function \( v \) satisfies

\[ \Delta v(x, y) = 0, \quad (x, y) \in D, \tag{4} \]
\[ v(x, y) = g(x, y) - u_p(x, y), \quad (x, y) \in \partial D. \tag{5} \]

Since \( v \) satisfies Laplace’s equation, the integral formulation for \( v \) contains no domain integration. This method works well, if \( u_p \) in (3) can be determined analytically which is a major task for a general function \( f(x, y) \). If \( f(x, y) \) can be approximated by a polynomial using Taylor series expansion, then \( u_p \) can be evaluated as accurately as possible. Once the particular solution is determined, (4)-(5) can be solved efficiently by traditional numerical methods. In the next section, an analytic particular solution is derived in a closed form when \( f \) is a polynomial.
3 Derivation of Analytic Particular Solutions

In this section, we first assume that \( f_n(x, y) \) be a homogeneous polynomial of degree \( n \); i.e.,

\[
f_n(x, y) = \sum_{k=0}^{n} A_k x^{n-k} y^k.
\]  

(6)

Then a particular solution of

\[
\Delta u_{p,n} = f_n(x, y)
\]  

(7)

may be of the form

\[
u_{p,n} = \sum_{k=0}^{n+2} P_k x^{n-k+2} y^k
\]  

(8)

that is a polynomial of degree \( n+2 \). With given \( \{A_k\}_{0}^{n} \), we need to determine the unknowns \( \{P_k\}_{0}^{n+2} \). By a direct substitution of (8) into (7), we obtain

\[
(n-k+2)(n-k+1)P_k + (k+1)(k+2)P_{k+2} = A_k, \quad k = 0, 1, 2, \ldots, n.
\]  

(9)

This is a linear system of \( n+1 \) algebraic equations with \( n+3 \) unknowns \( \{P_k\}_{0}^{n+2} \). Hence, two of the unknowns can be chosen freely. For convenience, we choose \( P_{n+2} = -A_{n+1} = 0 \). For \( k = n, n-1 \) in (9), we get

\[
P_n = \frac{A_n}{(1)(2)}, \quad P_{n-1} = \frac{A_{n-1}}{(2)(3)}.
\]  

(10)

To produce a general formula for \( P_k \), we make the following observation

\[
P_{n-2} = \frac{A_{n-2}}{(3)(4)} - \frac{(n-1)(n)}{1 \cdot 2 \cdot 3 \cdot 4} A_n = \frac{2!}{4!} A_{n-2} - \frac{(n-1)(n)}{4!} A_n
\]  

(11)

\[
P_{n-3} = \frac{3!}{5!} A_{n-3} - \frac{(n-2)(n-1)}{5!} A_{n-1} = \frac{3!(n-3)!A_{n-3} - 1!(n-1)!A_{n-1}}{5!(n-3)!}
\]  

(12)

Furthermore,

\[
P_{n-2} = \frac{A_{n-4}}{5 \cdot 6} - \frac{(n-3)(n-2)}{6 \cdot 5} P_{n-2}
\]  

\[
= \frac{A_{n-4}}{5 \cdot 6} - \frac{(n-3)(n-2)}{6 \cdot 5} \left[ \frac{2!}{4!} A_{n-2} - \frac{(n-1)(n)}{4!} A_n \right]
\]  

\[
= \frac{4!}{6!} A_{n-4} - \frac{2!(n-2)!}{6!(n-4)!} A_{n-2} + \frac{0!}{6!(n-4)!} A_n
\]  

\[
= \frac{4!(n-4)!A_{n-4} - 2!(n-2)!A_{n-2} + 0!n!A_n}{6!(n-4)!}
\]  

(14)

Based on these observations, one states the following theorem.
Theorem 1  A particular solution of (7) is given by (8) where

\[ P_{n+1} = P_{n+2} = 0, \]  

\[ P_k = \sum_{m=0}^{[\frac{n-k}{2}]} \frac{(-1)^m(k+2m)!(n-k-2m)!}{k!(n-k+2)!} A_{k+2m}, \quad 1 \leq k \leq n. \]  

Here \([n-k]/2\] denotes the integer \(M\) satisfying \(M \leq (n-k)/2 < M + 1\).

**Proof:** The theorem is proved by Mathematical Induction. Since (16) holds for \(k = n\) and \(n - 1\), it is sufficient to show that, if (16) is valid for \(k = j\), then it is valid for \(k = j - 2\).

Indeed, let us assume that (16) is true for \(k = j\); i.e.,

\[ P_j = \sum_{m=0}^{[\frac{n-k}{2}]} \frac{(-1)^m(j+2m)!(n-j-2m)!}{j!(n-j+2)!} A_{j+2m}. \]

From (9), we have

\[(n-j+4)(n-j+3)P_{j-2} + j(j-1)P_j = A_{j-2}\]

which implies

\[ P_{j-2} = \frac{A_{j-2}}{(n-j+4)(n-j+3)} - \frac{j(j-1)P_j}{(n-j+4)(n-j+3)}. \]  

Substituting \(P_j\) in (17) by (16), we have

\[ P_{j-2} = \frac{A_{j-2}}{(n-j+4)(n-j+3)} \]

\[ - \frac{j(j-1)}{(n-j+4)(n-j+3)} \sum_{m=0}^{[\frac{n-j}{2}]} \frac{(-1)^m(j+2m)!(n-j-2m)!}{j!(n-j+2)!} A_{j+2m} \]

\[ = \frac{A_{j-2}}{(n-j+4)(n-j+3)} + \sum_{m=0}^{[\frac{n-j}{2}]} \frac{(-1)^{m+1}(j+2m)!(n-j-2m)!}{(j-2)!(n-j+4)!} A_{j+2m} \]

\[ = \sum_{m=-1}^{[\frac{n-j}{2}]} \frac{(-1)^{m+1}(j+2m)!(n-j-2m)!}{(j-2)!(n-(j-2)+2)!} A_{j+2m} \]

\[ = \sum_{m=0}^{[\frac{n-(j-2)}{2}]} \frac{(-1)^m(j+2m-2)!(n-j-2m+2)!}{(j-2)!(n-(j-2)+2)!} A_{j-2+2m}. \]
This completes the proof by Induction.

From Theorem 1, the closed form of a particular solution in (8) becomes

\[
    u_p(x, y) = \sum_{k=0}^{n} \left[ \sum_{m=0}^{\left\lfloor \frac{n-k}{2} \right\rfloor} \frac{(-1)^m(k + 2m)! (n - k - 2m)!}{k! (n - k + 2)!} A_{k+2m} \right] x^{n-k+2} y^k. \tag{18}
\]

In recent years, the advancement of symbolic computational technology allows one to implement (18) visually as close to its original mathematical formula as possible. The following examples demonstrate the ease of calculating (18) using a modern PC with symbolic computation. Using Mathematica version 3.0, (18) can be coded as

\[
    \Phi[n_] := \sum_{k=0}^{n} \sum_{m=0}^{\left\lfloor \frac{n-k}{2} \right\rfloor} \frac{(-1)^m(k + 2m)! (n-k-2m)!}{k! (n-k+2)!} \text{Part}[a[k+2m+1]] x^{n-k+2} y^k. \tag{19}
\]

Here \( n \) is the degree of the homogeneous polynomial. Note that in the above Mathematica code \( a[k+2m+1] \) is used instead of \( a[k+2m] \) because the initial index of the array \( a \) is 1 instead of 0.

**Example 1.** Let \( \Delta \Phi = (x^4 - 4x^3 y + 6x^2 y^2 - 4xy^3 + y^4) / 12 \). Using (19), we obtain the following

\[
    \text{In} [1] := a = \{1/12, -1/3, 1/2, -1/3, 1/12\}; \; \Phi[4]
\]

\[
    \text{Out} [1] := \frac{x^6}{360} - \frac{x^3 y^3}{18} + \frac{x^2 y^4}{24}
\]

A particular solution of \( \Phi \) is given in \( \text{Out} [1] \).

**Example 2.** In this example, we show the case when \( f \) is a monomial. Let \( \Delta \Phi = x^2 y^2 \). Similar to Example 1, we obtain

\[
    \text{In} [1] := a = \{0, 0, 1, 0, 0\}; \; \Phi[4]
\]

\[
    \text{Out} [1] := \frac{x^6}{180} + \frac{x^4 y^2}{12}
\]

A particular solution \( \Phi \) is given in \( \text{Out} [1] \). When \( f \) is a nonhomogeneous polynomial, we can always split them into different groups of homogeneous terms and apply Example 1 and 2 repeatedly to obtain particular solutions for each group and then sum them up by the principle of linear superposition. This can be easily handled using symbolic computation.

## 4 The MFS for Laplace’s Equation

In this section we apply the MFS to solve the Laplace equation in (4)-(5). In the MFS we embed the boundary of the domain into an auxiliary boundary \( \partial D_A \supset D \). In general, we choose \( \partial D_A \) as a circle in 2D and a sphere in 3D [3, 5]. We then place the source points on \( \partial D_A \). In general the source
points are evenly distributed on a sphere containing the domain $D$. The purpose of moving the source point outside of the domain $D$ is to avoid the singularities of the fundamental solutions of the Laplacian.

Let $\{x_j\}_{j=1}^m$ be a set of source points lying on the auxiliary boundary $\partial D_A$. We approximate the solution of $v(x)$ of (4) by a function of the form [2, 5]

$$v_m(x) = \sum_{j=1}^m c_j G(x, x_j) + c, \quad x_j \in \partial D_A,$$  \hspace{1cm} (20)

where $G(x, x_j)$ is a fundamental solution given by

$$G(x, x_j) = \begin{cases} \frac{1}{2\pi} \log ||x - x_j||, & x, x_j \in \mathbb{R}^2, \\ \frac{1}{4\pi ||x - x_j||}, & x, x_j \in \mathbb{R}^3, \end{cases}$$

and $\|\cdot\|$ is the Euclidean norm. For collocation, we need to choose a set of points $\{x_k\}_{k=1}^m$ on $\partial D$. Applying the boundary conditions of (5) to (20), we obtain the following system of equations

$$\sum_{j=1}^m c_j G(x_k, x_j) + c = g(x_k) - u_p(x_k), \quad \text{for } k = 1, 2, ..., m. \hspace{1cm} (21)$$

Notice that due to the extra constant term $c$ in (21), one more collocation point on the physical boundary is required. The above linear system of equations can be solved for $\{c_j\}_{j=1}^m \cup \{c\}$ by a linear solver. Bogomolny [2] showed that the auxiliary boundary $\partial D_A$ can be taken as a circle in 2D (sphere in 3D) and $\{x_j\}_{j=1}^m$ equally distributed. As indicated by Bogomolny [2], the larger the radius of the source circle (sphere), the better the approximation to be expected. In this case the resulting matrix in (21) becomes extremely ill-conditioned. However, the numerical result seems insensitive to the ill-conditioning in the MFS.

An approximation $u_m$ to $u$ is then given by

$$u_m(x) = \sum_{j=1}^m c_j G(x, x_j) + c + u_p, \quad x \in \overline{D}. \hspace{1cm} (22)$$

## 5 Numerical Example

To demonstrate the effectiveness of our proposed method, we solve the following problem using Mathematica version 3.0 on a PC. Only 15 lines of coding is needed for the complete program.

Let us consider the following problem:

$$\begin{align*}
\Delta u(x, y) &= 2e^{x-y}, \quad (x, y) \in D \\
u(x, y) &= e^{x-y} + e^x \cos y, \quad (x, y) \in \partial D
\end{align*} \hspace{1cm} (23, 24)$$
Define $R(\theta) = \sqrt{\cos(2\theta) + \sqrt{1.1 - \sin^2(2\theta)}}$. Let

$$\partial D = \{R(\theta)(\cos \theta, \sin \theta) : 0 \leq \theta \leq 2\pi\}. \quad (25)$$

The physical domain has the shape of a peanut. The picture of (25), often called the Oval of Cassini in the mathematical literature, is shown in Figure 1.

Figure 1: Oval of Cassini.

The analytical solution of (23)-(24) is given by

$$u(x, y) = e^{x-y} + e^x \cos y, \quad (x, y) \in D \cup \partial D.$$ 

The right hand side of (23) can be approximated by the first $N$ terms of its Taylor series

$$2e^{x-y} \simeq \sum_{n=0}^{N} \frac{2(x-y)^n}{n!}. \quad (26)$$

Define

$$A_k^{(n)} = \frac{2(-1)^k}{k!(n-k)!}, \quad 0 \leq k \leq n,$$

and

$$Q_n(x, y) = \frac{2(x-y)^n}{n!}.$$ 

Then,

$$Q_n(x, y) = \frac{2}{n!} \sum_{k=0}^{n} \binom{n}{k} (-1)^k x^{n-k} y^k = \sum_{k=0}^{n} A_k^{(n)} x^{n-k} y^k. \quad (27)$$

From (23), $P_k^{(n)}$ is given by

$$P_k^{(n)} = \sum_{m=0}^{\left[\frac{n-k}{2}\right]} \frac{(-1)^m (k + 2m)! (n - k - 2m)!}{k!(n-k+2)!} A_{k+2m}^{(n)}$$

$$= \frac{2(-1)^k}{k!(n-k+2)!} \sum_{m=0}^{\left[\frac{n-k}{2}\right]} (-1)^m, \quad 0 \leq k \leq n. \quad (28)$$
Notice that
\[
\sum_{m=0}^{s} (-1)^m = \begin{cases} 
1, & \text{if } s \text{ is even}, \\
0, & \text{if } s \text{ is odd},
\end{cases} = \frac{(-1)^s + 1}{2}.
\]

Then \( P_k^{(n)} \) in (28) can be rewritten as
\[
P_k^{(n)} = \frac{2(-1)^k}{k!(n-k+2)!} \cdot \frac{(-1)^{\left\lfloor \frac{n-k}{2} \right\rfloor} + 1}{2}.
\]

Thus the particular solution \( \phi_n \) with respect to the forcing term \( Q_n \) in (27) is given by
\[
\phi_n(x,y) = \sum_{k=0}^{n} \frac{(-1)^k \left( (-1)^{\left\lfloor \frac{n-k}{2} \right\rfloor} + 1 \right)}{k!(n-k+2)!} x^{n-k+2} y^k.
\]

Let us denote \( u_p^N \) the particular solution corresponding to the first \( N + 1 \) terms of the Taylor Series expansion; it is given by
\[
u_p^N(x,y) = \sum_{n=0}^{N} \phi_n(x,y) = \sum_{n=0}^{N} \sum_{k=0}^{n} \frac{(-1)^k \left( (-1)^{\left\lfloor \frac{n-k}{2} \right\rfloor} + 1 \right)}{k!(n-k+2)!} x^{n-k+2} y^k. \quad (29)
\]

The particular solution \( u_p \) in (3) can be approximated by \( u_p^N \) in (29) as accurately as possible by taking \( N \) sufficiently large. The evaluation of \( u_p^N \) can be implemented easily using Mathematica 3.0 as it has been shown in Section 3.

Once we evaluate the particular solution, we can proceed to compute the solution by the MFS as shown in Section 4. We choose 34 collocation points on the physical boundary and 35 source points on the fictitious boundary, a circle with a center at \((0,0)\) and radius 10. Errors in approximation to \( u(x,y) \) at eight selected points are shown in Table 1. As we see in Table 1, the accuracy improves dramatically as \( N \) increases. There is a \( 6 \sim 7 \) order of magnitude of improvement in error using \( N = 3 \) and 11. For \( N = 3, 6, 9, 11 \), \( u_p^N(x,y) \) in (29) contains 7, 18, 33, 45 terms respectively. However, the polynomial can be efficiently evaluated by properly writing it in the nested multiplication form similar to Horner’s scheme. Furthermore, there is no need for matrix inversion in evaluating the particular solutions. Hence, the computational cost should not be an issue. The advantage of using more terms outweighs the computational cost.

The overall accuracy depends not only on the particular solution but also on the homogeneous solution \([4, 6]\). The MFS has been proved to be a highly accurate numerical method for solving homogeneous elliptic differential equations \([3, 5]\). The implementation of the MFS is straightforward.
Since the solution is not affected by the ill-conditioning of the matrix in the MFS [2], we usually ignore the warning given by the linear solver.

Table 1. Errors in approximation to \( u(x,y) \)

<table>
<thead>
<tr>
<th>( x )</th>
<th>( y )</th>
<th>( N = 3 )</th>
<th>( N = 6 )</th>
<th>( N = 9 )</th>
<th>( N = 11 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-3.986E-4</td>
<td>-1.043E-6</td>
<td>-6.621E-9</td>
<td>-8.913E-11</td>
</tr>
<tr>
<td>0.6262</td>
<td>0</td>
<td>-4.459E-3</td>
<td>-2.929E-5</td>
<td>-7.734E-8</td>
<td>-1.016E-9</td>
</tr>
<tr>
<td>1.3419</td>
<td>0</td>
<td>-8.530E-3</td>
<td>-1.449E-4</td>
<td>-1.812E-7</td>
<td>2.353E-10</td>
</tr>
<tr>
<td>0.174</td>
<td>0.174</td>
<td>-4.054E-4</td>
<td>-2.045E-6</td>
<td>-6.503E-9</td>
<td>-8.535E-11</td>
</tr>
<tr>
<td>0.3728</td>
<td>0.3728</td>
<td>-1.081E-4</td>
<td>-1.721E-6</td>
<td>-1.634E-9</td>
<td>-2.643E-11</td>
</tr>
<tr>
<td>0</td>
<td>0.2071</td>
<td>-4.007E-5</td>
<td>1.258E-6</td>
<td>-4.129E-10</td>
<td>-1.348E-11</td>
</tr>
<tr>
<td>-0.3728</td>
<td>0.3728</td>
<td>-1.457E-4</td>
<td>1.475E-7</td>
<td>-2.006E-9</td>
<td>-1.479E-11</td>
</tr>
<tr>
<td>-1.3419</td>
<td>0</td>
<td>1.939E-2</td>
<td>1.122E-3</td>
<td>2.580E-7</td>
<td>2.799E-8</td>
</tr>
</tbody>
</table>

6 Conclusions

Evaluation of particular solution for Poisson’s equation in 2D is proposed using polynomial approximation. The closed form of the approximate particular solution has been explicitly formulated. The attractiveness of this approach is that the particular solution can be evaluated as accurately as possible and symbolic computation can be easily implemented. In contrast to other interpolation schemes such as radial basis functions, no matrix inversion is required in this approach. Hence there is no inherited ill-conditioned problem in the process of reconstructing nonhomogeneous terms of the given differential equation.

Following this line of reasoning, the proposed method can be extended to other linear differential operators and 3D cases. In addition, polynomial approximation is better understood than radial basis functions in the mathematics literature and thus detailed mathematical analysis is possible. This will be the subjects of our future research.

Acknowledgments: The first author acknowledges the support of a NATO grant under reference CRG 970593.

References


