

# Multiresolution Discrete Finite Difference Masks for Rapid Solution Approximation of the Poisson's Equation

Ravi Kumar Jha

Faculty of Computing  
Universiti Teknologi Malaysia  
Johor Bahru, Malaysia  
ravijha070@gmail.com

Hassan Ugail

Centre for Visual Computing  
University of Bradford  
Bradford, United Kingdom  
h.ugail@bradford.ac.uk

Habibollah Haron

Faculty of Computing  
Universiti Teknologi Malaysia  
Johor Bahru, Malaysia  
habib@utm.my

Andres Iglesias

Dept. of Applied Mathematics  
University of Cantabria  
Santander, Spain  
iglesias@unican.es

**Abstract**—The Poisson's equation is an essential entity of applied mathematics for modelling many phenomena of importance. They include the theory of gravitation, electromagnetism, fluid flows and geometric design. In this regard, finding efficient solution methods for the Poisson's equation is a significant problem that requires addressing. In this paper, we show how it is possible to generate approximate solutions of the Poisson's equation subject to various boundary conditions. We make use of the discrete finite difference operator, which, in many ways, is similar to the standard finite difference method for numerically solving partial differential equations. Our approach is based upon the Laplacian averaging operator which, as we show, can be elegantly applied over many folds in a computationally efficient manner to obtain a close approximation to the solution of the equation at hand. We compare our method by way of examples with the solutions arising from the analytic variants as well as the numerical variants of the Poisson's equation subject to a given set of boundary conditions. Thus, we show that our method, though simple to implement yet computationally very efficient, is powerful enough to generate approximate solutions of the Poisson's equation.

**Keywords:** Poisson's Equation, Laplace operator, Averaging, Finite difference scheme.

## I. INTRODUCTION

The Poisson's equation is represented as  $\nabla^2\varphi = -f$ , where  $\nabla$  is the standard elliptic Laplacian operator,  $\varphi$  and  $f$  are real valued functions on a manifold. In this case, given  $f$ ,  $\varphi$  is usually sought subject to a certain set of boundary conditions. This equation is due to the French mathematician, S. D. Poisson and is widely applicable to practical problems in engineering and physics. Examples include, modelling gravitational and electrostatic fields [4], surface reconstruction [9], image processing [11] as well as other applications in geometric design [5], [1], [13].

For instance, the problem of surface reconstruction seeks a continuous function  $\varphi$  which can adequately interpolate a set of discrete points  $p_i$  where the function sought provides both the surface as well as normal information. Poisson's equation in this instance can be utilised to obtain a solution to this problem [9]. Similarly, variants of the Poisson's equation can be utilised for geometric modelling where efficient ways of

creating and manipulating geometry that represent complex objects are sought after [13].

Due to their immense applicability in engineering, physics and science in general, formulation of partial differential equations (PDEs) and their solutions date back well over 300 years. Much credit must go to the mathematicians such as Euler, Leibniz, Hermann and Newton whose contributions to the field are highly notable. Thus, as mentioned earlier, given the practical applicability of the Poisson's equation, efficient solution mechanisms to this equation is highly sought after even to date.

Many techniques for finding the solution of PDEs have been proposed. These include analytic methods such as the separation of variables, Green's function, Fourier analysis [2] and many notable numerical schemes such as the finite element method (FEM) [8], the finite difference method (FDM) [8], [12] and the finite volume method (FVM) [3]. Of these numerical methods, the FDM is particularly interesting for its elegant formulation and ease of implementation.

Just like other numerical methods for solving PDEs, the FDM consists of partitioning the domain of computation into smaller grids to approximate the differential operator concerned. The operator can usually be approximated using the corresponding differential quotients. In the case of the FDM, the solution hinges on the approximation power of the Taylor series which can usually approximate the function satisfying the relevant PDE difference operator. The accuracy of the given FDM scheme can be computed by comparing the difference between the exact solution for the differential operator and the numerical solution using the difference operator. Aside from the solution being approximate, one major drawback with numerical schemes for solving PDEs is that the computation often involves large matrix manipulations which can be computationally intensive.

Our aim in this work is to develop a solution scheme,

specifically for the Poisson's equation, using the idea behind the FDM. The fundamental difference in what we are proposing here, however, is that we use the discrete finite difference mask to iteratively approximate the domain of solution. Thus, our method is computationally very efficient and computations in our case are linear averaging over the computational domain and is free from complex matrix operations. The fundamental idea behind this solution scheme is based on the fact that the Laplacian operator essentially signifies an averaging process. This is due to the very definition of the Laplacian operator and we take advantage of this to formulate a simple yet elegant way of obtaining a satisfactory approximate solution of the Poisson's equation subject to the given set of boundary conditions.

The rest of this paper is structured as follows. In Section II, we discuss our methodology in detail. We show the idea of FDM can be used to implement multiresolution discrete masks which can be applied continuously to approximate the Poisson's operator in the solution space. In Section III, we discuss how we can implement a solution scheme and provide details of our algorithm. In Section IV, we discuss some relevant examples to demonstrate the computational accuracy as well as the ease at which our solution scheme can be implemented. Finally, in Section V, we conclude this paper and present possible future direction in which this study can be taken forward.

## II. METHODOLOGY

As discussed earlier, there exist a vast number of techniques for solving elliptic PDEs such as the Poisson equation [6], [10]. Of those methods, the FDM is one of the compact methods and hence is a popular solution scheme. For this very reason, we hinge our method on this technique, and we implement our solution scheme to simplify the solution process further.

To discuss our methodology in detail, we take the elliptic PDE of the form,

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = -f(x, y), \quad (1)$$

in the bounded domain  $\Omega = [0, 1] \subset \mathfrak{R}$ . The above PDE is nothing but the 2-dimensional form of the Poisson's equation in the standard Euclidean space with the source term  $f(x, y)$ . We can now take the associated Laplacian operator such that,

$$\nabla^2 = \frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2}. \quad (2)$$

The first step in deriving a finite difference approximation of the Equation (1) is to partition the unit interval into a finite number of discretely smaller intervals. Thus, it is important to highlight the fact that the numerical solution, in this case,

is not defined on the whole domain but at a finite number of points in  $\Omega$  [6], [15].

We introduce the equally distributed grid points in each of the domain directions such that  $\varphi_i$  is  $0 \leq i \leq M + 1$  and  $\varphi_j$  is  $0 \leq n \leq N + 1$ . Here  $\varphi_i = ih$  while  $\varphi_j = jk$  with integer spacing  $h$  is given by  $h = 1/(M + 1)$  and  $k$  is given by  $k = 1/(N + 1)$ . Thus, at each of the grid points we are looking for a numerical value  $\varphi_{ij}$  where  $\varphi_0 = \alpha$  and  $\varphi_1 = \beta$  are imposed at the boundaries of  $\Omega$ .

Using the above formulation, we can write the discrete PDE operator for the Laplacian (2) such that,

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{k^2} = 0, \quad (3)$$

where  $u_{i,j}$  is the operator parameter for  $(i, j)$  node points and  $h$  and  $k$  are the spatial discretisations. For simplicity, we take equal spacing across both directions of the domain to have,

$$\frac{u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}}{4} = u_{i,j}. \quad (4)$$

For further details of this difference operator, the reader is referred to [12].

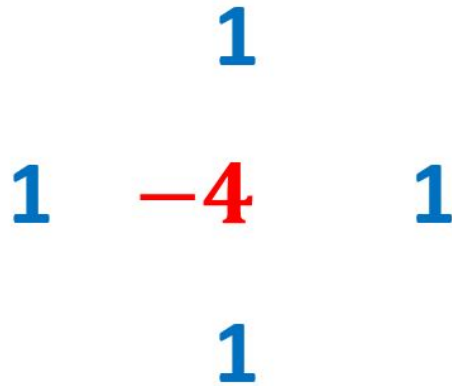


Fig. 1. Illustration of the averaging property of the finite difference operator.

Now, from the format of the Equation (4), and as shown in Figure 1, one can see that the form of  $u$  is actually the mean of its four neighbour points from the north, west, south and the east. Thus, the discrete Laplacian operator can be approximated by considering the standard averaging of the neighbouring values. It is this idea that we extend further in order to formulate our proposed approximate solution.

Considering the above discussions, we propose to consider the finite difference scheme in (4) as the mask which can be applied as a linear sequence of operative iterations to formulate the approximate solution.

One thought which might arise to the mind of a curious reader might be to understand the behaviour of the approximate solution for the given finite difference scheme in (4) while applying a linear sequence of iterations over the solution domain. In other words, can one write some form of a priori estimates on the stability as well as the boundedness of the approximate solution? i.e. does the consistent scheme we have proposed here successfully approximate the discrete problem. Here, we try to demonstrate this indeed is the case.

**Theorem.** Suppose  $u$  is a continuous differentiable function of order at least 4 in the interval  $[x - h_0, x + h_0]$ , for  $h_0 > 0$  then,  $\frac{u(x+h) - 2u(x) + u(x-h)}{h^2}$  is a continuous approximation of the second derivative  $u''$  of  $u$  at a point  $x$ .

**Proof.** We assume there exist a constant  $C > 0$  such that,

$$\left| \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} - u''_x \right| \leq Ch^2. \quad (5)$$

Using Taylor series expansion up to 4th order we obtain,

$$u(x+h) = u(x) + hu'(x) + \frac{h^2}{2}u''(x) + \frac{h^3}{6}u'''(x) + \frac{h^4}{24}u''''(\delta^+),$$

and,

$$u(x-h) = u(x) - hu'(x) - \frac{h^2}{2}u''(x) - \frac{h^3}{6}u'''(x) - \frac{h^4}{24}u''''(\delta^-),$$

where,  $\delta^+ \in [x, x+h]$  and  $\delta^- \in [x-h, x]$ .

By using the intermediate value theorem, we can express,

$$\frac{u(x+h) - 2u(x) + u(x-h)}{h^2} = u''_x + \frac{h^2}{12}u''''(\delta), \quad (6)$$

where  $\delta \in [x-h, x+h]$ .

Hence, given  $y \in [x-h_0, x+h_0]$ , we can deduce that  $C = \sup \frac{|u''''(y)|}{12}$ .

### III. DISCRETE FINITE DIFFERENCE MASKS

In this section, we show the above method can be utilised to formulate an algorithm which can produce approximate solutions of the Poisson's equation subject to a given set of boundary conditions. The idea behind is to utilise the mask given in Equation (4) to successively produce an approximate solution of the chosen PDE. Thus, given four discrete points

each in the direction of north, west, south and east with respect to the boundaries, we can utilise the simple averaging to produce a new point in the solution domain. Figure 2 further illustrates this idea, where the central point shown in the domain is obtained by applying the Laplacian operator - which essentially the average of the four neighbouring points as shown. Further, we note that the values on boundaries are known [6].

Starting from the above formulation, we can now add further approximate solution points into the domain by repeating the application of the Laplacian mask on the finer grid generated from the previous iteration. Figure 3 illustrates this procedure.

The process described above can be applied to the solution domain in a linear fashion whereby each iteration brings out a higher resolution in the solution domain. Figures 4 and 5 further illustrates this procedure.

Now, based on the formulation and description, we show how the algorithm for the above procedure can be implemented.

#### Algorithm - Approximate Solution using the Laplacian mask.

Set up the PDE problem.

Fix the boundary conditions.

Fix the value  $h$ . (this will determine the number of iterations,  $M$ )

For  $i = 1$  to  $M$  do,

Group the solution domain. (by finding the 4 point neighbours)

Compute the Solution <sub>$i$</sub> . (using the Laplacian mask)

Update the grid points

Return the final solution.

### IV. EXAMPLES

The discussions in this section focus on how the above described methodology can be validated by way of examples. In particular, we take two examples whereby in each case we have compared our proposed method with the exact solution as well as the standard finite difference numerical method. We show the variation in the error rates as well as the computational efficiencies in each case.

**Example 1.** In our first example, we take a linear 2-dimensional Poisson problem with the source term,  $6xy(1-y) - 2x^3$ . This problem also has the exact solution  $u(x, y) = y(1-y)x^3$  on the unit square with the dirichlet boundary conditions,

$$u(0, y) = 0, \quad u(x, 0) = 0,$$

$$u(1, y) = y(1-y), \quad u(x, 1) = 0.$$

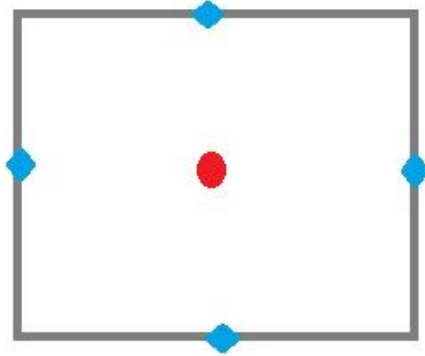


Fig. 2. Illustration of the four point averaging based on the Laplacian mask.

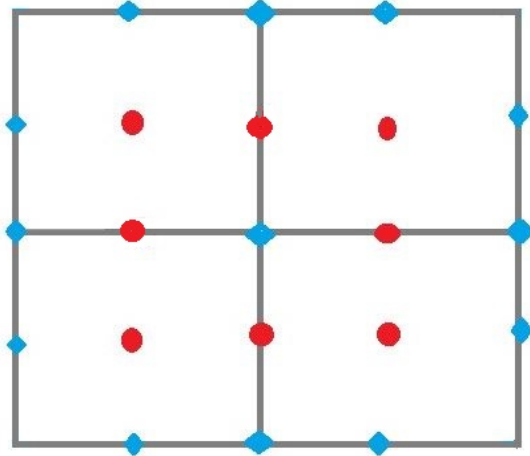


Fig. 3. Illustration of how iterative subdivision can be applied using the Laplacian operator.

Now we solve this problem using the standard finite difference numerical method to try and verify the results against our proposed method. Some typical results are shown in Tables I and II based on the experiments. Figures 6, 7 and 8 show the solution plots for exact, numerical solution using the FDM and our proposed solution respectively.

**Example 2.** As a second example we use the Poisson's problem with source term  $f = 34\pi^2 \sin(5\pi x) \sin(3\pi y)$ . Again, the exact solution for this problem is known to be  $u = \sin(5\pi x) \sin(3\pi y)$  on the unit square with the homogeneous Dirichlet boundary conditions,

$$u(0, y) = 0, \quad u(x, 0) = 0,$$

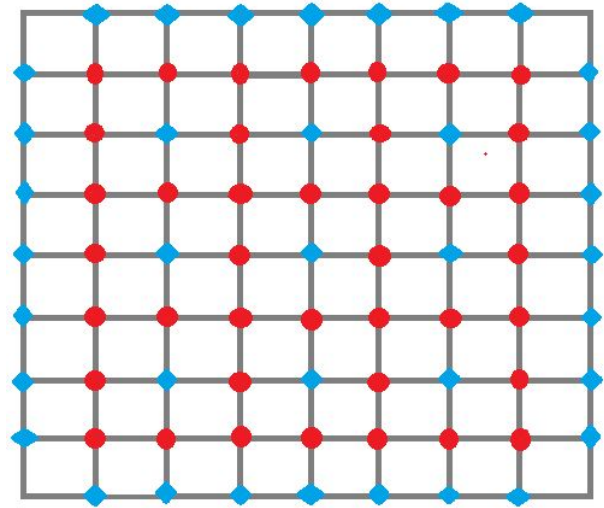


Fig. 4. Illustration of how iterative application of the Laplacian on the solution domain.

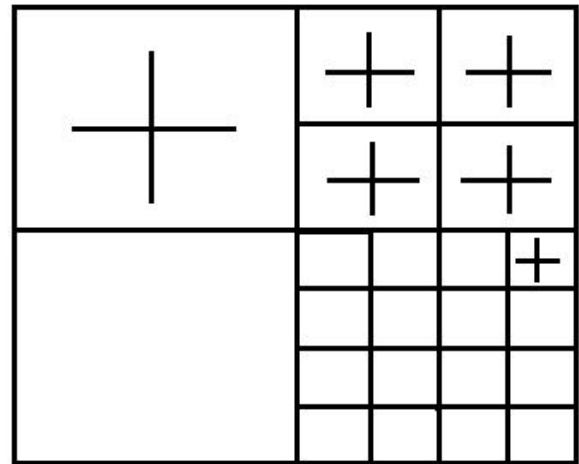


Fig. 5. General description how the Laplacian operator can be repeatedly applied in the solution domain to obtain more accurate approximations.

$$u(1, y) = y(1 - y), \quad u(x, 1) = 0.$$

Again, we solve this problem using the standard finite difference numerical method to try and verify the results against our proposed method. Some typical results are shown in Tables III and IV based on the experiments. Figures 9, 10 and 11 show the solution plots for exact, numerical solution using the FDM and our proposed solution respectively.

Note, for both the examples discussed above we have performed the computation using Matlab on a standard Windows PC.

Looking at the results from both examples, it can be seen that our proposed method approximates the solution of the Poisson's adequately. Though in the first example the

TABLE I

$L^2$  NORM ERROR VALUES AGAINST THE ANALYTIC SOLUTION, COMPARING THE FINITE DIFFERENCE NUMERICAL SOLUTION AGAINST OUR PROPOSED METHOD FOR THE POISSON'S PROBLEM DISCUSSED IN EXAMPLE 1.

Mesh size	Numerical FDM	Our proposed method
$17 \times 17$	0.0113	0.0308
$65 \times 65$	0.0112	0.0312
$257 \times 257$	0.0056	0.0312
$513 \times 513$	0.0008	0.0312
$1025 \times 1025$	0.0002	0.0312

TABLE II

RUN TIME IN SECONDS FOR THE FINITE DIFFERENCE NUMERICAL SOLUTION AGAINST OUR PROPOSED METHOD FOR THE POISSON'S PROBLEM DISCUSSED IN EXAMPLE 1.

Mesh size	Numerical FDM	Our proposed method
$17 \times 17$	0.0738	0.0067
$65 \times 65$	0.3638	0.0072
$257 \times 257$	4.6331	0.0353
$513 \times 513$	18.6956	0.1582
$1025 \times 1025$	74.9183	1.049

error rate is lower, we can see the error for our method is relatively high in the second example. It is noteworthy, the solution in the second case is noisy with lots of peaks and the averaging method as discussed here is likely to be less accurate in such cases. Nevertheless the solution from our method for example 2 can still be acceptable for situations such as those arising in geometric design. Therefore, we believe there is much merit in our proposed solution scheme, especially given that the implementation of this solution is relatively straightforward and comparatively speaking it is computationally very lightweight.

TABLE III

$L^2$  NORM ERROR VALUES AGAINST THE ANALYTIC SOLUTION, COMPARING THE FINITE DIFFERENCE NUMERICAL SOLUTION AGAINST OUR PROPOSED METHOD FOR THE POISSON'S PROBLEM DISCUSSED IN EXAMPLE 2.

Mesh size	Numerical FDM	Our proposed method
$17 \times 17$	0.0679	0.6997
$65 \times 65$	0.0042	0.9524
$257 \times 257$	0.0002	0.9946
$513 \times 513$	0.0225	0.9983
$1025 \times 1025$	0.3832	0.9995

TABLE IV

RUN TIME IN SECONDS COMPARING THE FINITE DIFFERENCE NUMERICAL SOLUTION AGAINST OUR PROPOSED METHOD FOR THE POISSON'S PROBLEM DISCUSSED IN EXAMPLE 2.

Mesh size	Numerical FDM	Our proposed method
$17 \times 17$	0.0736	0.0006
$65 \times 65$	0.4094	0.0015
$357 \times 257$	4.7123	0.0196
$513 \times 513$	18.8158	0.1398
$1025 \times 1025$	74.7506	0.9213

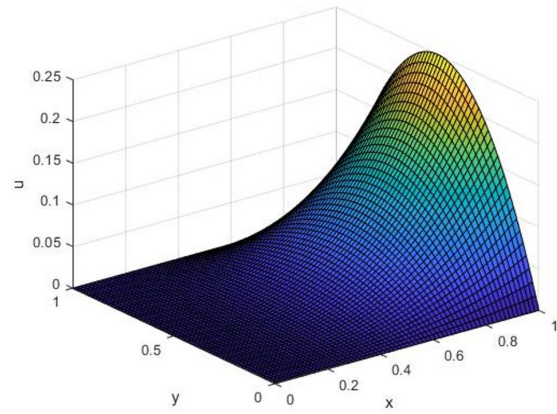


Fig. 6. Plot of the exact solution for the Poisson's problem discussed in Example 1.

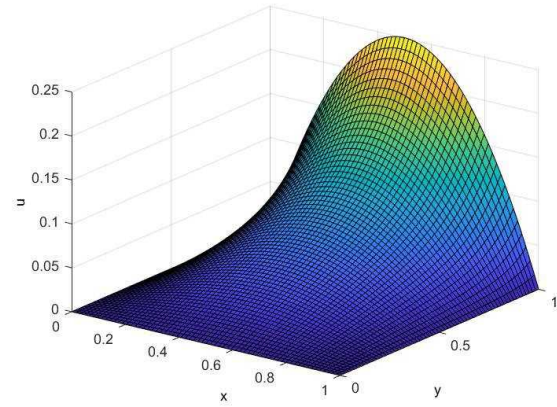


Fig. 7. Plot of the numerical finite difference solution for the Poisson's problem discussed in Example 1.

## V. CONCLUSION

In this paper, we have discussed a fast approximate solution method for the Poisson's equation subject to various boundary conditions. The solution technique we have discussed here makes use of the discrete finite difference Laplacian operator arising from the finite difference numerical solutions of PDEs. We have shown the existence as well as the stability of the discrete Laplacian operator for the approximate solution method we discuss here. We have discussed examples which validate both the accuracy as well as the computational efficiency of our method. The results indicate our method has merit, especially where pseudo approximate solutions are called. We note this might be very applicable in real time geometric design scenarios, for example.

The work discussed here is preliminary in our opinion. Hence there is much room for improvement. For example, we have taken the standard Laplacian masks and applied them iteratively in a linear form in the solution space. It would



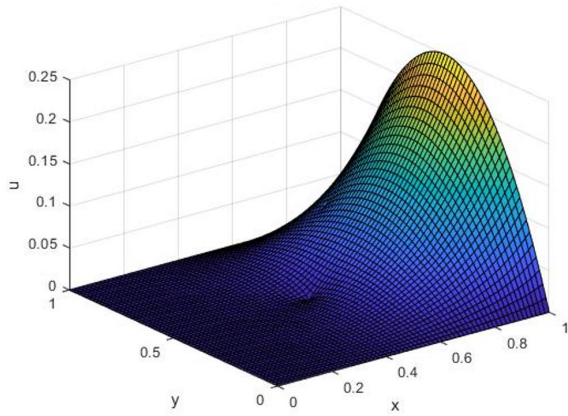


Fig. 8. Plot of the proposed solution for the Poisson's problem discussed in Example 1.

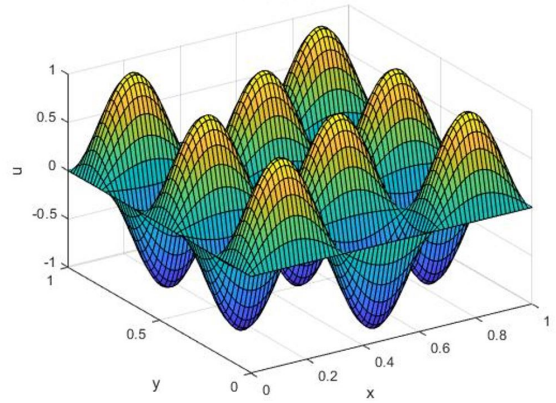


Fig. 10. Plot of the numerical FDM solution for the Poisson's problem discussed in Example 2.

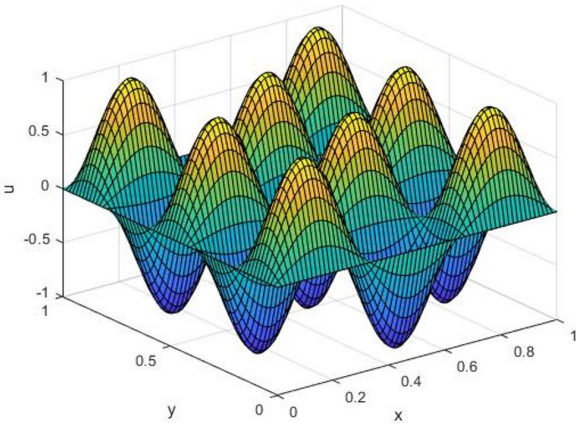


Fig. 9. Plot of the exact solution for the Poisson's problem discussed in Example 2.

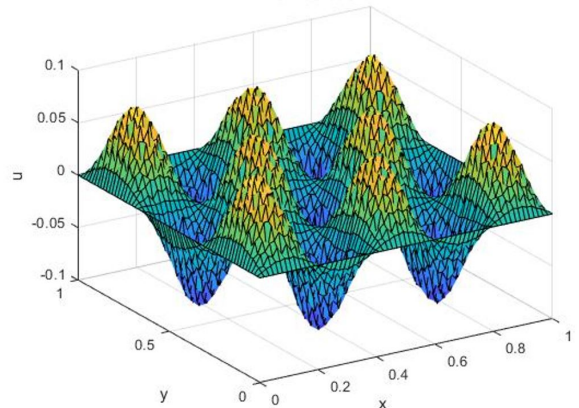


Fig. 11. Plot of the proposed solution for the Poisson's problem discussed in Example 2.

be interesting to introduce dynamic and non-linear parameters into the Laplacian operator to locally adjust the solution as the iteration progress. For example, for the problem discussed in Example 2, this could be very applicable. A further line of research would be to look into adopting the method presented to find approximate solutions of higher order elliptic PDEs such as the Biharmonic equation [16], [7], [14].

## VI. ACKNOWLEDGEMENT

This work was supported by the European Union's Horizon 2020 Programme H2020-MSCA-RISE-2017, under the project PDE-GIR with grant number 778035.

## REFERENCES

[1] N. Ahmat, G. Gonzalez-Castro and H. Ugail, Automatic Shape Optimisation of Pharmaceutical Tablets using Partial Differential Equations, *Computers and Structures*, 130 19, (2013).

[2] M. Andersson, R. Sigurdsson and M. Passare, Analytic Solutions to Partial Differential Equations, in *Complex Convexity and Analytic Functionals*, Progress in Mathematics, 225, 129-150, (2004).

[3] E.F. Anley, Numerical Solutions of Elliptic Partial Differential Equations by Using Finite Volume Method, *Pure and Applied Mathematics Journal*, 5(4), 120-129, (2015).

[4] R.J. Blakely, *Potential Theory in Gravity and Magnetic Applications*, Cambridge University Press, (2010).

[5] E. Chaudhry, S. J. Bian, H. Ugail, X. Jin and L. You, Dynamic Skin Deformation using Finite Difference Solutions for Character Animation, *Computers and Graphics*, 46(1), 294-305, (2015).

[6] F.W. Dorr, The Direct Solution of the Discrete Poisson Equation on a Rectangle, *SIAM Review*, 12(2), (1970).

[7] G. Gonzalez, R. Spares, H. Ugail, B.R. Whiteside and J. Sweeney, Towards the Analytic Characterization of Micro and Nano Surface Features using the Biharmonic Equation, *Applied Mathematical Modelling*, 36 (3), 1161-1172, (2012).

[8] A. Iserles, *A First Course in the Numerical Analysis of Differential Equations*, Cambridge University Press, (2008).

[9] M. Kazhdan, M. Bolitho and H. Hoppe, Poisson Surface Reconstruction, in *Eurographics Symposium on Geometry Processing*, (2006).

[10] M. Meyer, M. Desbrun, P. Schroder, and A.H. Barr, Discrete Differential-Geometry Operators for Triangulated 2-Manifolds, in *Hege*

- HC., Polthier K. (eds.) Visualization and Mathematics III, Mathematics and Visualization, Springer, Berlin, Heidelberg, (2003).
- [11] P. Pérez, M. Gangnet and A. Blake, Poisson Image Editing, ACM Transactions on Graphics, 22(3), 313-318, (2003).
  - [12] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, Springer-Verlag, (1999).
  - [13] H. Ugail, Spine Based Shape Parameterisation for PDE Surfaces, Journal of Computing, 72 195-206, (2004).
  - [14] H. Ugail, M. Márquez and A. Yilmaz, On Bézier Surfaces in Three-Dimensional Minkowski Space, Computers and Mathematics with Applications, 62 2899-2912, (2011).
  - [15] A.M. Winslow, Numerical Solution of the Quasilinear Poisson Equation in a Nonuniform Triangle Mesh, Journal of Computational Physics, 2, 149-172, (1967).
  - [16] L. You, H. Ugail, B.P. Tang, X. Jin and X. You, Blending using ODE Swept Surfaces with Shape Control and C1 Continuity, The Visual Computer, 30(6-8), 625-636. (2014).