Minireview Article

Solution of Laplace equation by modified differential transform method

.

ABSTRACT

In this paper, we applied themodified two-dimensional differential transform method to solve Laplace equation. Laplace equation is one of Elliptic partial differential equations. These kinds of differential equations have specific applications models of physics and engineering. We consider four models with two Dirichlet and two Neumann boundary conditions. The simplicity of this method compared to other iteration methods is shown here. It is worth mentioning that here only a few number of iterations are required to reach the closed form solutions as series expansions of some known functions.

Keywords: differential transform method, Elliptic partial differential equations, Laplace equation, Iteration

1. INTRODUCTION

Most of the problems in physics and engineering often use partial differential equations.A linear second order partial differential equation can be written as $Au_{xx} + Bu_{xy} + Cu_{yy} = F(x, y, u, u_x, u_y)$ where A, B and C may be functions of x and y. (x, y), denotes the independent variables and $u(x, y)$, the dependent variable, or solution of the PDE. If $B^2 - 4AC < 0$, then the equation is called Elliptic.
Laplace equation is one of Elliptic partial differential equations. The Laplace equation, named after the French mathematician Pierre-Simon Laplace. It is a fundamental equation in classical field theories and plays a crucial role in various scientific and engineering applications. The Laplace equation is a second-order partial differential equation. In two dimensions (2D): $\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$. Here, u is the unknown function of the variables x, y and ∇^2 is the Laplacian operator, which is the divergence of the gradient of u.
DTM was first introduced in 1988 by Zhou. This method can be used to solve linear and non-linear ordinary differential equations. Chen and Ho developed this method as a two-dimensional differential transformation method for PDEs and obtains closure by rank solutions for linear and nonlinear initial value problems. This method reduces the size of the computational domain and is easily applicable to many problems.
Dirichlet condition means the value of the function is prescribed, when $u(x, y) = g(x, y)$ on the boundary $\partial\Omega$.
Neumann condition means the value of the derivative normal to the boundary is prescribed, when $\frac{\partial u}{\partial n} = v(x, y)$ on the boundary $\partial\Omega$.

2. METHODOLOGY

In this study, the two dimensional differential transform method is used to find solutions of elliptic equations.
One dimensional differential transform function $u(x, y)$ can be represented as,

$$u(x, y) = \sum_{m=0}^{\infty} F(m)x^m \sum_{n=0}^{\infty} G(n)y^n = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} U(m, n)x^m y^n \qquad (1)$$

Where $U(m,n) = F(m)G(m)$ is called spectrum of $u(x,y)$. $u(x,y) = f(x)g(y)$.
The basic definitions and operations of two-dimensional differential transform method:
Definition 1: If a function $u(x,y)$ is analytic and differentiable with respect to time t in the domain of interest,

$$U(m,n) = \frac{1}{m!n!}\left[\frac{\partial^{m+n}u(x,y)}{\partial x^m \partial y^n}\right]_{x=x_0,y=y_0} \quad (2)$$

Where the spectrum $U(m,n)$ is the transformed function. Then the differential inverse transform of $U(m,n)$ is defined as follows

$$u(x,y) = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} U(m,n)(x-x_0)^m (y-y_0)^n \quad (3)$$

$u(x,y)$ - original function, $U(m,n)$ - transform function
Combining equation (2) and (3), it can be obtained that

$$u(x,y) = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} \frac{1}{m!n!}\left[\frac{\partial^{m+n}u(x,y)}{\partial x^m \partial y^n}\right]_{x=x_0,y=y_0} (x-x_0)^m (y-y_0)^n \quad (4)$$

Theorem:

1. If $u(x,y) = v(x,y) \pm w(x,y)$, then

$$U(m,n) = V(m,n) \pm W(m,n)$$

2. If $u(x,y) = av(x,y)$, then

$$U(m,n) = aV(m,n)$$

3. If $u(x,y) = v(x,y)w(x,y)$, then

$$U(m,n) = \sum_{k=0}^{m}\sum_{l=0}^{n} V(k,n-l)W(m-k,l)$$

4. If $u(x,y) = \frac{\partial^{r+s}v(x,y)}{\partial x^r \partial y^s}$, then

$$U(m,n) = \frac{(m+r)!}{m!}\frac{(n+s)!}{n!}V(m+r,n+s)$$

5. If $u(x,y) = e^{av(x,y)}$, then

$$U(m,n) = \begin{cases} e^{av(0,0)}, & m = n = 0 \\ a\sum_{k=0}^{m-1}\sum_{l=0}^{n}\frac{m-k}{m} V(m-k,l)U(k,n-l) & m \geq 1 \\ a\sum_{k=0}^{m}\sum_{l=0}^{n-1}\frac{n-l}{n}V(k,n-l)U(m-k,n), & n \geq 1 \end{cases}$$

6. If $u(x,y) = x^k y^h$, then

$$U(m,n) = \begin{cases} \delta(m-k,n-h), & m = k, n = h \\ 0, & otherwise \end{cases}$$

7. If $u(x,y) = x^k e^{ay}$, then

$$U(m,n) = \delta(m-k)\frac{a^n}{n!}$$

3. RESULTS

Solutions of Laplace equation

Consider the second order Laplace equation, given as
uxx + uyy = 0, $0 < x, y < \pi$

Dirichlet boundary condition (01)
u(x, 0) = sinh x, u(x, π) = −sinh x,
u(0, y) = 0, u(π, y) = sinh(π) cos y.
Taking the differential transform
(m + 1)(m + 2)U(m + 2, n) + (n + 1)(n + 2)U(m, n + 2) = 0
u(x, 0) = $\sum_{m=0}^{\infty} U(m,0)x^m = sinhx = \sum_{m=0}^{\infty} \frac{x^m}{m!}$; m-odd
which, on comparing the both sides yield

$$U(m,0) = \begin{cases} \frac{1}{m!}, & if m - odd \\ 0, & otherwise \end{cases}$$

$$u(0,y) = \sum_{n=0}^{\infty} U(0,n)y^n$$

$$U(0,n) = 0$$

$$U(m,n) = \begin{cases} \frac{(-1)^{n/2}}{m!\,n!}, & if\ m - odd, n - even \\ 0, & otherwise \end{cases}$$

$$u(x,y) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} U(m,n)x^m y^n$$

$$u(x,y) = \sum_{m=1,3,5..}^{\infty} \sum_{n=0,2,4,..}^{\infty} \frac{(-1)^{\frac{n}{2}}}{m!\,n!} x^m y^n$$

$$u(x,y) = sinhx\ cosy$$

<u>Dirichlet boundary condition (02)</u>

u(x, 0) = 0. u(x, π) = 0,

u(0, y) = sin y, u(π, y) = cosh(π) sin y.

Taking the differential transform

(m + 1)(m + 2)U(m + 2, n) + (n + 1)(n + 2)U(m, n + 2) = 0

$$u(x,0) = \sum_{m=0}^{\infty} x(m,0) = 0$$

$$u(m,0) = 0$$

$$u(o,y) = \sum_{n=0}^{\infty} x(m,0)x^m = siny = \sum_{n=0}^{\infty} \frac{(-1)^{\frac{n-1}{2}}}{n!} y^n$$

$$U(0,n) = \begin{cases} \frac{(-1)^{n-1/2}}{n!}, & if\ n-odd \\ 0, & otherwise \end{cases}$$

$$U(m,n) = \begin{cases} \frac{(-1)^{n-\frac{1}{2}}}{m!\,n!}, & if\ m-even, n-odd \\ 0, & otherwise \end{cases}$$

$$u(x,y) = \sum_{m=0.2,4.6,..}^{\infty} \sum_{n=1,3,5,..}^{\infty} \frac{(-1)^{n-1/2}}{n!} x^m y^n$$

$$u(x,y) = \sum_{m=0,2,4,..}^{\infty} \frac{x^m}{m!} \sum_{n=1,3,5}^{\infty} \frac{(-1)^{n-\frac{1}{2}}}{n!} y^n$$

$$u(x,y) = coshxsiny$$

<u>Neumann boundary condition (01)</u>

$u_y(x, 0) = 0$, $u_y(x, \pi) = 2 \cos 2x \sinh 2\pi$

$u_x(0, y) = 0$, $u_x(\pi, y) = 0$

Taking the differential transform

$(m + 1)(m + 2)U(m + 2, n) + (n + 1)(n + 2)U(m, n + 2) = 0$

$$u_y(x,\pi) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} n\pi^{n-1} U(m,n) x^m$$

$$u_y(x,\pi) = 2cos2xsinh(2\pi)$$

$$u_y(x,\pi) = 2\sum_{m=0}^{\infty} \frac{(-1)^{\frac{m}{2}}}{m!} (2x)^m \sum_{n=0}^{\infty} \frac{(2\pi)^n}{n!}$$

By changing the index n , and comparison, we have

$$U(m,n+1) = \frac{(-1)^{m/2} 2^{m+n+1}}{(n+1)m!\,n!}$$

$$U(m,n) = \begin{cases} \frac{(-1)^{m/2} n^{m+n}}{m!\,n!}, & if\ m-even, n-odd \\ 0, & otherwise \end{cases}$$

$$u(x,y) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{(-1)^{\frac{m}{2}} 2^{m+n}}{m!\,n!} x^m y^n$$

$$u(x.y) = \sum_{m=0,2,4,...}^{\infty} \frac{(-1)^{m/2}(2x)^m}{m!} \sum_{n=0,2,4,...}^{\infty} \frac{(2y)^n}{n!}$$

$$u(x,y) = cos2xcosh2y$$

Neumann boundary condition (02)

uy(x, 0) = cos x,   uy(x, π) = cosh π cos x,

ux(0, y) = 0,   ux(π, y) = 0

Taking the differential transform

(m + 1)(m + 2)U(m + 2, n) + (n + 1)(n + 2)U(m, n + 2) = 0

$$u_y(x,0) = \sum_{n=0}^{\infty} U(m,1)x^m = cosx = \sum_{m=0}^{\infty} \frac{(-1)^{\frac{m}{2}} x^m}{m!}$$

$$U(m,1) = \begin{cases} \frac{(-1)^{m/2}}{m!}, & if\ m-even \\ 0, & otherwise \end{cases}$$

$$\text{ux}(0,\text{y}) = \sum_{n=0}^{\infty} U(1,n)y^n = 0$$

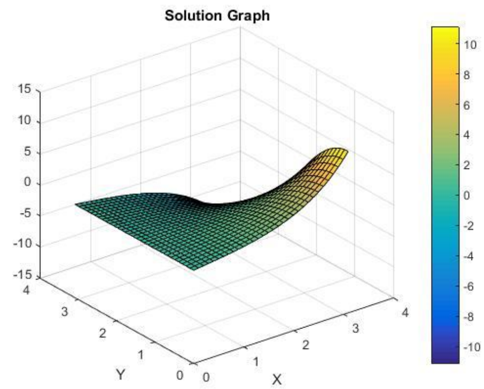
$$U(1,n) = 0$$

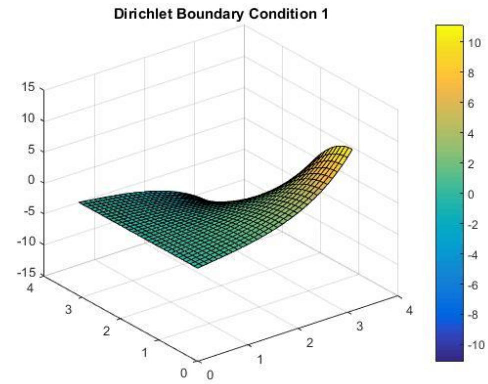
$$U(m,n) = \begin{cases} \frac{(-1)^{m/2}}{m!\,n!}, & if\ m-even, n-odd \\ 0, & otherwise \end{cases}$$

$$u(x,y) = \sum_{m=0,2,4,\ldots}^{\infty} \frac{(-1)^{\frac{m}{2}} x^m}{m!} \sum_{n=1,3,5,..}^{\infty} \frac{y^n}{n!}$$

$$u(x,y) = cosx. sinhy$$

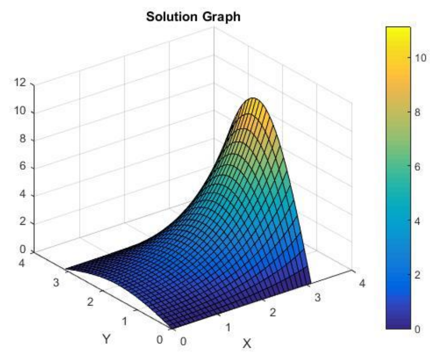
Example 01(Dirichlet Boundary Condition 01):

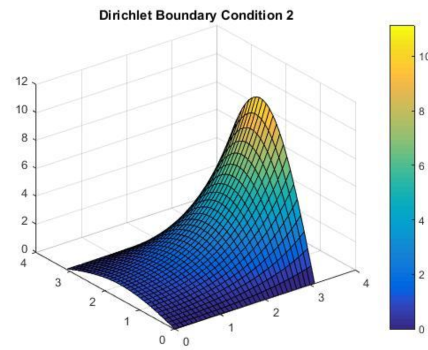




Approximate solution Exact Solution

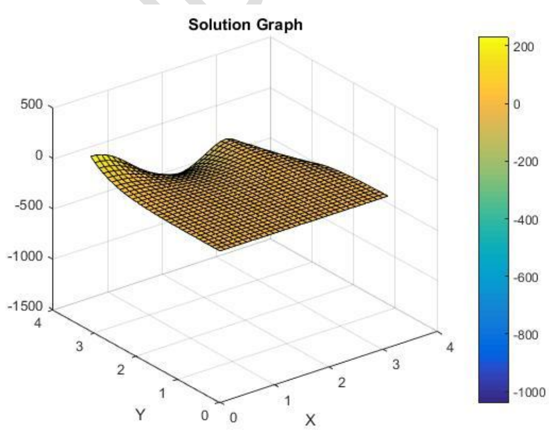
Example 02(Dirichlet Boundary Condition 02):

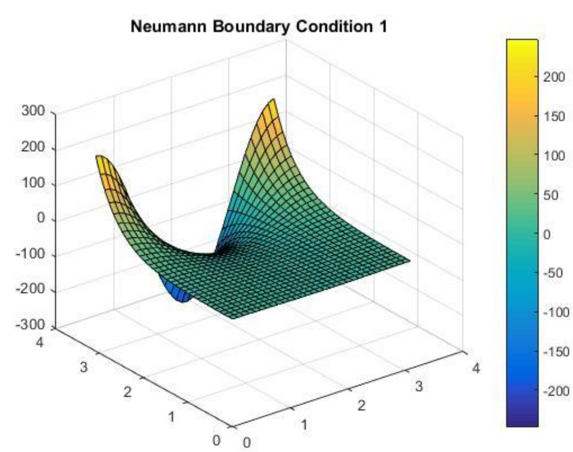




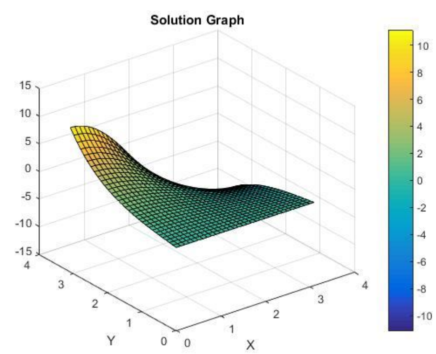
Approximate solution Exact Solution

Example 03(Neumann Boundary Condition 01):

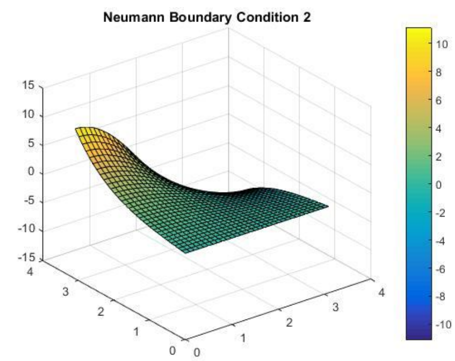




Approximate solution Exact Solution



Example 04(Neumann Boundary Condition 02):



Approximate solution Exact Solution

4. CONCLUSION

In this paper, we have considered 4 cases of Laplace equation and presented solutions. We have successfully developed the DTM to obtain the exact solutions of Laplace equation. A computer program (MATLAB) is used in making the computation within some seconds provided that the program is well posed. In obtaining the Approximate solution, more accurate values can be obtained when using larger values of m and n.

REFERENCES

Wazwaz, A.M., 2007. The variational iteration method for exact solutions of Laplace equation. *Physics Letters A*, *363*(4), pp.260-262.

Amir, M.J., Yaseen, M. and Iqbal, R., 2013. Exact solutions of Laplace equation by differential transform method. *arXiv preprint arXiv:1312.7277*.Ayaz, F. (2003).

Ayaz, F., 2003. On the two-dimensional differential transform method. *Applied Mathematics and computation*, *143*(2-3), pp.361-374. Mitra, A.K., (2010). *Finite difference method for the solution of Laplace equation.* Department of aerospace engineering Iowa state University.

Shortley, G.H. and Weller, R., 1938. The numerical solution of Laplace's equation. *Journal of Applied Physics*, *9*(5), pp.334-348.

Sohail, M. and Mohyud-Din, S.T., 2012. Reduced differential transform method for laplace equations. *Int. J. Modern Theo. Physics*, *1*(1), pp.6-12.

Rebenda, J. and Šmarda, Z., 2019, July. Boundary value problem for elliptic equations: A differential transformation approach. In *AIP Conference Proceedings* (Vol. 2116, No. 1). AIP Publishing.

Reimer, A.S. and Cheviakov, A.F., 2013. A matlab-based finite-difference solver for the poisson problem with mixed dirichlet–neumann boundary conditions. *Computer Physics Communications*, *184*(3), pp.783-798.

APPENDIX

MATLAB program to find approximation solution of Laplace equation with Dirichlet and Neumann boundary conditions.

```
% Define the range and gap for X and Y
X = 0:0.1:pi;
Y = 0:0.1:pi;

u = zeros(length(X), length(Y));

%input U(m,n)
Um = input('Enter the expression of U(m,n) part with m: ');
Un = input('Enter the expression of U(m,n) part with n: ');
m1 = input('states of m value(1-even, 0-odd): ');
n1 = input('states of n value(1-even, 0-odd): ');
mt = input('Enter the max terms : ');

% Convert the input expressions to symbolic variables
syms m
Um1 = sym(Um);
syms n
Un1 = sym(Un);

if m1==0 && n1==0
   for i = 1:length(X)
      for j = 1:length(Y)
         x = X(i);
         y = Y(j);

         % Compute the sum for m
sum_m = 0;
         for p = 1:2:max_terms
term_m = subs(Um1,m,p) * x^p;
sum_m = sum_m + term_m;
            if abs(term_m) < eps % Check for convergence
               break;
            end
         end

         % Compute the sum for n
sum_n = 0;
         for q = 1:2:max_terms
term_n = subs(Un1,n,q) * y^q;
sum_n = sum_n + term_n;
            if abs(term_n) < eps % Check for convergence
               break;
            end
         end
```

```
          % Compute the overall u value
u(i, j) = sum_m * sum_n;

      end
   end

elseif m1==0 && n1==1
   for i = 1:length(X)
      for j = 1:length(Y)
         x = X(i);
         y = Y(j);

         % Compute the sum for m
sum_m = 0;
         for p = 1:2:max_terms
term_m = subs(Um1,m,p) * x^p;
sum_m = sum_m + term_m;
            if abs(term_m) < eps % Check for convergence
               break;
            end
         end

         % Compute the sum for n
sum_n = 0;
         for q = 0:2:max_terms
term_n = subs(Un1,n,q) * y^q;
sum_n = sum_n + term_n;
            if abs(term_n) < eps % Check for convergence
               break;
            end
         end

         % Compute the overall u value
u(i, j) = sum_m * sum_n;

      end
   end

elseif m1==1 && n1==0
   for i = 1:length(X)
      for j = 1:length(Y)
         x = X(i);
         y = Y(j);

         % Compute the sum for m
sum_m = 0;
         for p = 0:2:max_terms
term_m = subs(Um1,m,p) * x^p;
sum_m = sum_m + term_m;
            if abs(term_m) < eps % Check for convergence
               break;
            end
```

```
            end

            % Compute the sum for n
sum_n = 0;
            for q = 1:2:max_terms
term_n = subs(Un1,n,q) * y^q;
sum_n = sum_n + term_n;
                if abs(term_n) < eps % Check for convergence
                    break;
                end
            end

            % Compute the overall u value
u(i, j) = sum_m * sum_n;

        end
    end
else
    for i = 1:length(X)
        for j = 1:length(Y)
            x = X(i);
            y = Y(j);

            % Compute the sum for m
sum_m = 0;
            for p = 0:2:max_terms
term_m = subs(Um1,m,p) * x^p;
sum_m = sum_m + term_m;
                if abs(term_m) < eps % Check for convergence
                    break;
                end
            end

            % Compute the sum for n
sum_n = 0;
            for q = 1:2:max_terms
term_n = subs(Un1,n,q) * y^q;
sum_n = sum_n + term_n;
                if abs(term_n) < eps % Check for convergence
                    break;
                end
            end

            % Compute the overall u value
u(i, j) = sum_m * sum_n;

        end
    end
end

u1 = real(u);

% Plot the graph
```

```
figure;
surf(X, Y, u1');
title('Solution Graph');
xlabel('X');
ylabel('Y');
colorbar;
```

MATLAB programs to get solution graph for exact solutions.

Example 01(Dirichlet Boundary Condition 01):

```
[X,Y] = meshgrid(0:0.1:pi,0:0.1:pi);
u = sinh(X).* cos(Y);
surf(X,Y,u)
title('Dirichlet Boundary Condition 1');
colorbar;
```

Example 02(Dirichlet Boundary Condition 02)

```
[X,Y] = meshgrid(0:0.1:pi,0:0.1:pi);
u = cosh(X).* sin(Y);
surf(X,Y,u)
title('Dirichlet Boundary Condition 2');
colorbar;
```

Example 03(Neumann Boundary Condition 01)

```
[X,Y] = meshgrid(0:0.1:pi,0:0.1:pi);
u = cos(2*X).* cosh(2*Y);
surf(X,Y,u)
title('Neumann Boundary Condition 1');
colorbar;
```

Example 04(Neumann Boundary Condition 02)

```
[X,Y] = meshgrid(0:0.1:pi,0:0.1:pi);
u = cos(X).*sinh(Y);
surf(X,Y,u)
title('Neumann Boundary Condition 2');
colorbar;
```