# Advanced Mathematics and Mechanics Applications Using MATLAB®

Third Edition

Howard B. Wilson
*University of Alabama*

Louis H. Turcotte
*Rose-Hulman Institute of Technology*

David Halpern
*University of Alabama*

CHAPMAN & HALL/CRC

A CRC Press Company
Boca Raton London New York Washington, D.C.

# Advanced Mathematics and Mechanics Applications Using MATLAB®

## Third Edition

Howard B. Wilson
*University of Alabama*
Louis H. Turcotte
*Rose-Hulman Institute of Technology*
David Halpern
*University of Alabama*

CHAPMAN & HALL/CRC
A CRC Press Company
Boca Raton London New York Washington, D.C.

**Library of Congress Cataloging-in-Publication Data**

Wilson, H.B.
Advanced mathematics and mechanics applications using MATLAB / Howard B. Wilson, Louis H. Turcotte, David Halpern.—3rd ed.
p. cm.
ISBN 1-58488-262-X
1. MATLAB. 2. Engineering mathematics—Data processing. 3. Mechanics, Applied—Data processing. I. Turcotte, Louis H. II. Halpern, David. III. Title.

TA345 . W55 2002
620′.00151—dc21 2002071267



Direct all inquiries to CRC Press LLC, 2000 N.W. Corporate Blvd., Boca Raton, Florida 33431.


International Standard Book Number 1-58488-262-X
Library of Congress Card Number 2002071267
Printed in the United States of America 1 2 3 4 5 6 7 8 9 0
Printed on acid-free paper

© 2003 by Chapman & Hall/CRC

# *Contents*

# Chapter 9

# Boundary Value Problems for Partial Differential Equations

## 9.1 Several Important Partial Differential Equations

Many physical phenomena are characterized by linear partial differential equations. Such equations are attractive to study because (a) principles of superposition apply in the sense that linear combinations of component solutions can often be used to build more general solutions and (b) finite difference or finite element approximations lead to systems of linear equations amenable to solution by matrix methods. The accompanying table lists several frequently encountered equations and some applications. We only show one- or two-dimensional forms, although some of these equations have relevant applications in three dimensions.

In most practical applications the differential equations must be solved within a finite region of space while simultaneously prescribing boundary conditions on the function and its derivatives. Furthermore, initial conditions may exist. In dealing with the initial value problem, we are trying to predict future system behavior when initial conditions, boundary conditions, and a governing physical process are known. Solutions to such problems are seldom obtainable in a closed finite form. Even when series solutions are developed, an infinite number of terms may be needed to provide generality. For example, the problem of transient heat conduction in a circular cylinder leads to an infinite series of Bessel functions employing characteristic values which can only be computed approximately. Hence, the notion of an “exact” solution expressed as an infinite series of transcendental functions is deceiving. At best, we can hope to produce results containing insignificantly small computation errors.

The present chapter applies eigenfunction series to solve nine problems. Examples involving the Laplace, wave, beam, and heat equations are given. Nonhomogeneous boundary conditions are dealt with in several instances. Animation is also provided whenever it is helpful to illustrate the nature of the solutions.

| Equation | Equation Name | Applications |
|---|---|---|
| $u_{xx} + u_{yy} = \alpha u_t$ | Heat | Transient heat conduction |
| $u_{xx} + u_{yy} = \alpha u_{tt}$ | Wave | Transverse vibrations of membranes and other wave type phenomena |
| $u_{xx} + u_{yy} = 0$ | Laplace | Steady-state heat conduction and electrostatics |
| $u_{xx} + u_{yy} = f(x, y)$ | Poisson | Stress analysis of linearly elastic bodies |
| $u_{xx} + u_{yy} + \omega^2 u = 0$ | Helmholtz | Steady-state harmonic vibration problems |
| $EIy_{xxxx} = -A\rho y_{tt} + f(x, t)$ | Beam | Transverse flexural vibrations of elastic beams |

## 9.2 Solving the Laplace Equation inside a Rectangular Region

Functions which satisfy Laplace's equation are encountered often in practice. Such functions are called harmonic; and the problem of determining a harmonic function subject to given boundary values is known as the Dirichlet problem [119]. In a few cases with simple geometries, the Dirichlet problem can be solved explicitly. One instance is a rectangular region with the boundary values of the function being expandable in a Fourier sine series. The following program employs the FFT to construct a solution for boundary values represented by piecewise linear interpolation. Surface and contour plots of the resulting field values are also presented.

The problem of interest satisfies the differential equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad , \quad 0 < x < a \quad , \quad 0 < y < b$$

with the boundary conditions of the form

$$\begin{aligned} u(x,0) &= F(x) \quad , \quad 0 < x < a \,, \\ u(x,b) &= G(x) \quad , \quad 0 < x < a \,, \\ u(0,y) &= P(y) \quad , \quad 0 < y < b \,, \\ u(a,y) &= Q(y) \quad , \quad 0 < y < b \,. \end{aligned}$$

The series solution can be represented as

$$u(x,y) = \sum_{n=1}^{\infty} f_n a_n(x,y) + g_n a_n(x, b-y) + p_n b_n(x,y) + q_n b_n(a-x, y)$$

where

$$a_n(x,y) = \sin\left[\frac{n\pi x}{a}\right] \sinh\left[\frac{n\pi(b-y)}{a}\right] / \sinh\left[\frac{n\pi b}{a}\right],$$

$$b_n(x,y) = \sinh\left[\frac{n\pi(a-x)}{b}\right] \sin\left[\frac{n\pi y}{b}\right] / \sinh\left[\frac{n\pi a}{b}\right],$$

and the constants $f_m$, $g_m$, $p_n$, and $q_n$ are coefficients in the Fourier sine expansions of the boundary value functions. This implies that

$$F(x) = \sum_{n=1}^{\infty} f_n \sin\left[\frac{n\pi x}{a}\right] \ , \ G(x) = \sum_{n=1}^{\infty} g_n \sin\left[\frac{n\pi x}{a}\right],$$

$$P(y) = \sum_{n=1}^{\infty} p_n \sin\left[\frac{n\pi y}{b}\right] \ , \ Q(y) = \sum_{n=1}^{\infty} q_n \sin\left[\frac{n\pi y}{b}\right].$$

The coefficients in the series can be computed by integration as

$$f_n = \frac{2}{a}\int_0^a F(x) \sin\left[\frac{n\pi x}{a}\right] \, dx \ , \ g_n = \frac{2}{a}\int_0^a G(x) \sin\left[\frac{n\pi x}{a}\right] \, dx,$$

$$p_n = \frac{2}{a}\int_0^b P(y) \sin\left[\frac{n\pi y}{b}\right] \, dy \ , \ q_n = \frac{2}{a}\int_0^b Q(y) \sin\left[\frac{n\pi y}{b}\right] \, dy,$$

or approximate coefficients can be obtained using the FFT. The latter approach is chosen here and the solution is evaluated for an arbitrary number of terms in the series.

The chosen problem solution has the disadvantage of employing eigenfunctions that vanish at the ends of the expansion intervals. Consequently, it is desirable to combine the series with an additional term allowing exact satisfaction of the corner conditions for cases where the boundary value functions for adjacent sides agree. This implies requirements such as $F(a) = Q(0)$ and three other similar conditions. It is evident that the function

$$u_p(x,y) = c_1 + c_2 x + c_3 y + c_4 xy$$

is harmonic and varies linearly along each side of the rectangle. Constants $c_1, \cdots, c_4$ can be computed to satisfy the corner values and the total solution is represented as $u_p$ plus a series solution involving modified boundary conditions.

The following program **laplarec** solves the Dirichlet problem for the rectangle. Function values and gradient components are computed and plotted. Functions used in this program are described below. The example data set defined in the driver program was chosen to produce interesting surface and contour plots. Different boundary conditions can be handled by slight modifications of the input data. In this example 100 term series are used. Figure 9.1 through Figure 9.4 show function and gradient components, as well as a contour plot of function values. Readers may find it instructive to run the program and view these figures from different angles

| | |
|---|---|
| **laplarec** | inputs data, calls computation modules, and plots results |
| **datafunc** | defines an example datacase |
| **ulinbc** | particular solution for linearly varying boundary conditions |
| **recseris** | sums the series for function and gradient values |
| **sincof** | generates coefficients in a Fourier sine series |
| **lintrp** | piecewise linear interpolation function allowing jump discontinuities |

using the interactive figure rotating capability provided in MATLAB. Note that the figure showing the function gradient in the $x$ direction used view([225,20]) to show clearly the jump discontinuity in this quantity.

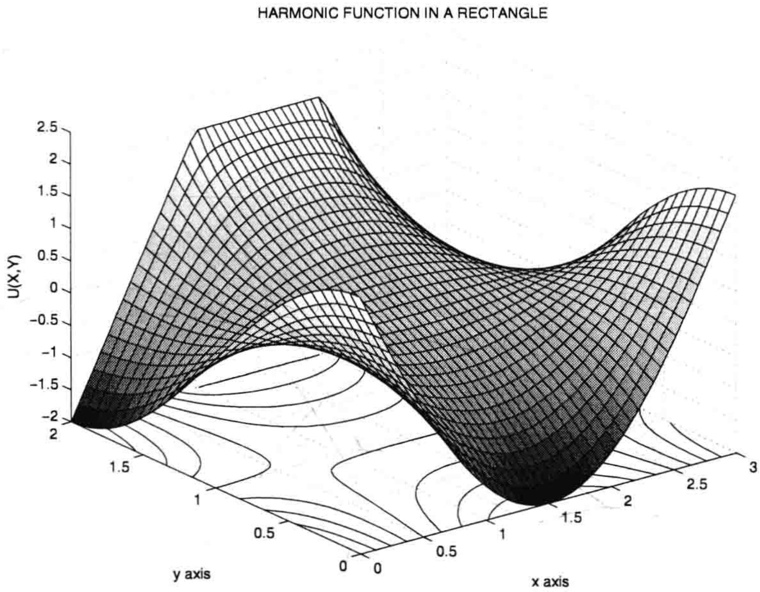


**Figure 9.1: Surface Plot of Function Values**

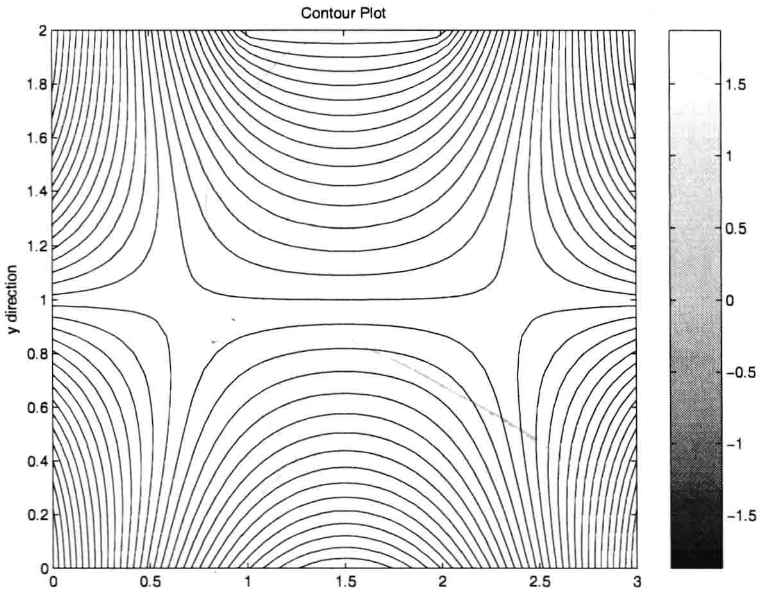


**Figure 9.2: Contour Plot of Function Values**

 

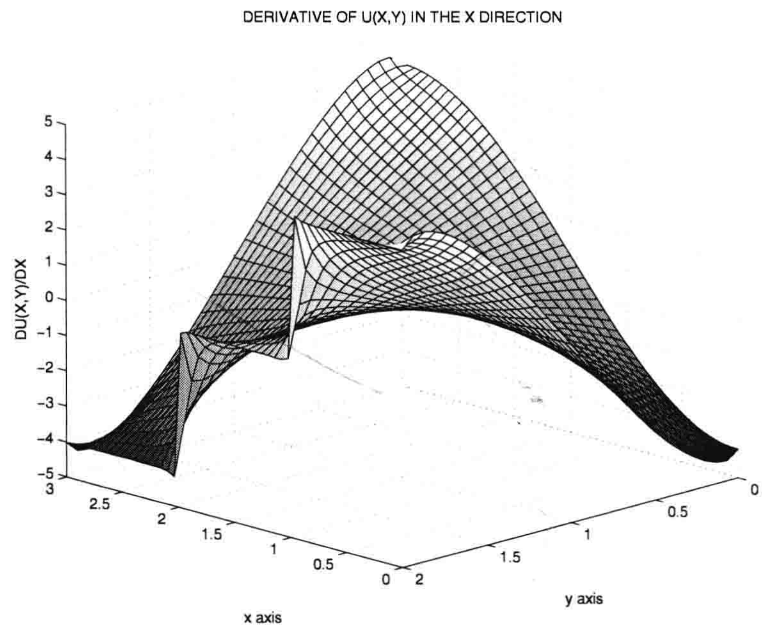


**Figure 9.3: Function Derivative in the x Direction**

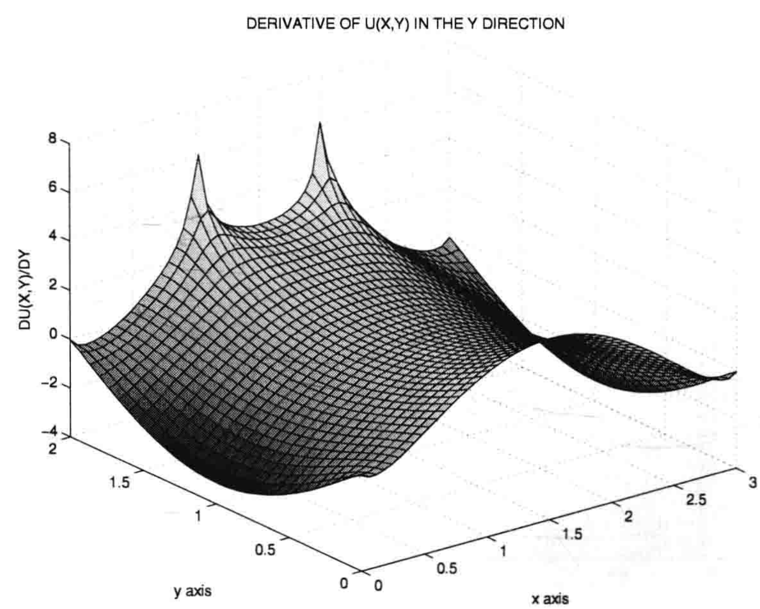


**Figure 9.4: Function Derivative in the y Direction**

## MATLAB Example

### Program laplarec

```
function [u,ux,uy,X,Y]=laplarec(...
                    ubot,utop,ulft,urht,a,b,nx,ny,N)
%
% [u,ux,uy,X,Y]=laplarec(...
%               ubot,utop,ulft,urht,a,b,nx,ny,N)
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% This program evaluates a harmonic function and its
% first partial derivatives in a rectangular region.
% The method employs a Fourier series expansion.
% ubot    - defines the boundary values on the bottom
%           side. This can be an array in which
%           ubot(:,1) is x coordinates and ubot(:,2)
%           is function values. Values at intermediate
%           points are obtained by piecewise linear
%           interpolation. A character string giving
%           the name of a function can also be used.
%           Then the function is evualuated using 200
%           points along a side to convert ubot to an
%           array. Similar comments apply for utop,
%           ulft, and urht introduced below.
% utop    - boundary value definition on the top side
% ulft    - boundary value definition on the left side
% urht    - boundary value definition on the right side
% a,b     - rectangle dimensions in x and y directions
% nx,ny   - number of x and y values for which the
%           solution is evaluated
% N       - number of terms used in the Fourier series
% u       - function value for the solution
% ux,uy   - first partial derivatives of the solution
% X,Y     - coordinate point arrays where the solution
%           is evaluated
%
% User m functions used: datafunc ulinbc
%                        recseris ftsincof

disp(' ')
disp('SOLVING THE LAPLACE EQUATION IN A RECTANGLE')
disp(' ')

if nargin==0
```

```
    disp(...
      'Give the name of a function defining the data')
    datfun=input(...
      '(try datafunc as an example): > ? ','s');
    [ubot,utop,ulft,urht,a,b,nx,ny,N]=feval(datfun);
end

% Create a grid to evaluate the solution
x=linspace(0,a,nx); y=linspace(0,b,ny);
[X,Y]=meshgrid(x,y); d=(a+b)/1e6;
xd=linspace(0,a,201)'; yd=linspace(0,b,201)';

% Check whether boundary values are given using
% external functions. Convert these to arrays

if isstr(ubot)
   ud=feval(ubot,xd); ubot=[xd,ud(:)];
end
if isstr(utop)
   ud=feval(utop,xd); utop=[xd,ud(:)];
end
if isstr(ulft)
   ud=feval(ulft,yd); ulft=[yd,ud(:)];
end
if isstr(urht)
   ud=feval(urht,yd); urht=[yd,ud(:)];
end

% Determine function values at the corners
ub=interp1(ubot(:,1),ubot(:,2),[d,a-d]);
ut=interp1(utop(:,1),utop(:,2),[d,a-d]);
ul=interp1(ulft(:,1),ulft(:,2),[d,b-d]);
ur=interp1(urht(:,1),urht(:,2),[d,b-d]);
U=[ul(1)+ub(1),ub(2)+ur(1),ur(2)+ut(2),...
       ut(1)+ul(2)]/2;

% Obtain a solution satisfying the corner
% values and varying linearly along the sides

[v,vx,vy]=ulinbc(U,a,b,X,Y);

% Reduce the corner values to zero to improve
% behavior of the Fourier series solution
% near the corners

```

 

```
f=inline('u0+(u1-u0)/L*x','x','u0','u1','L');
ubot(:,2)=ubot(:,2)-f(ubot(:,1),U(1),U(2),a);
utop(:,2)=utop(:,2)-f(utop(:,1),U(4),U(3),a);
ulft(:,2)=ulft(:,2)-f(ulft(:,1),U(1),U(4),b);
urht(:,2)=urht(:,2)-f(urht(:,1),U(2),U(3),b);

% Evaluate the series and combine results
% for the various component solutions

[ub,ubx,uby]=recseris(ubot,a,b,1,x,y,N);
[ut,utx,uty]=recseris(utop,a,b,2,x,y,N);
[ul,ulx,uly]=recseris(ulft,a,b,3,x,y,N);
[ur,urx,ury]=recseris(urht,a,b,4,x,y,N);
u=v+ub+ut+ul+ur; ux=vx+ubx+utx+ulx+urx;
uy=vy+uby+uty+uly+ury; close

% Show results graphically

surfc(X,Y,u), xlabel('x axis'), ylabel('y axis')
zlabel('U(X,Y)')
title('HARMONIC FUNCTION IN A RECTANGLE')
shg, pause
% print -deps laprecsr

contour(X,Y,u,30); title('Contour Plot');
xlabel('x direction'); ylabel('y direction');
colorbar, shg, pause
% print -deps laprecnt

surf(X,Y,ux), xlabel('x axis'), ylabel('y axis')
zlabel('DU(X,Y)/DX')
title('DERIVATIVE OF U(X,Y) IN THE X DIRECTION')
shg, pause
% print -deps laprecdx

surf(X,Y,uy), xlabel('x axis'), ylabel('y axis')
zlabel('DU(X,Y)/DY')
title('DERIVATIVE OF U(X,Y) IN THE Y DIRECTION')
% print -deps laprecdy
shg

%==============================================

function [ubot,utop,ulft,urht,a,b,...
                                nx,ny,N]=datafunc
```