

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

This book contains information obtained from authentic and highly regarded sources. Reprinted material is quoted with permission, and sources are indicated. A wide variety of references are listed. Reasonable efforts have been made to publish reliable data and information, but the author and the publisher cannot assume responsibility for the validity of all materials or for the consequences of their use.

Neither this book nor any part may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, microfilming, and recording, or by any information storage or retrieval system, without prior permission in writing from the publisher.

The consent of CRC Press LLC does not extend to copying for general distribution, for promotion, for creating new works, or for resale. Specific permission must be obtained in writing from CRC Press LLC for such copying.

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

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation, without intent to infringe.

Visit the CRC Press Web site at www.crcpress.com

© 2003 by Chapman & Hall/CRC

No claim to original U.S. Government works

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

Contents

1 Introduction

- 1.1 MATLAB: A Tool for Engineering Analysis
- 1.2 MATLAB Commands and Related Reference Materials
- 1.3 Example Problem on Financial Analysis
- 1.4 Computer Code and Results
 - 1.4.1 Computer Output
 - 1.4.2 Discussion of the MATLAB Code
 - 1.4.3 Code for Financial Problem

2 Elementary Aspects of MATLAB Graphics

- 2.1 Introduction
- 2.2 Overview of Graphics
- 2.3 Example Comparing Polynomial and Spline Interpolation
- 2.4 Conformal Mapping Example
- 2.5 Nonlinear Motion of a Damped Pendulum
- 2.6 A Linear Vibration Model
- 2.7 Example of Waves in an Elastic String
- 2.8 Properties of Curves and Surfaces
 - 2.8.1 Curve Properties
 - 2.8.2 Surface Properties
 - 2.8.3 Program Output and Code

3 Summary of Concepts from Linear Algebra

- 3.1 Introduction
- 3.2 Vectors, Norms, Linear Independence, and Rank
- 3.3 Systems of Linear Equations, Consistency, and Least Squares Approximation
- 3.4 Applications of Least Squares Approximation
 - 3.4.1 A Membrane Deflection Problem
 - 3.4.2 Mixed Boundary Value Problem for a Function Harmonic Inside a Circular Disk
 - 3.4.3 Using Rational Functions to Conformally Map a Circular Disk onto a Square
- 3.5 Eigenvalue Problems
 - 3.5.1 Statement of the Problem
 - 3.5.2 Application to Solution of Matrix Differential Equations

- 3.5.3 The Structural Dynamics Equation
- 3.6 Computing Natural Frequencies for a Rectangular Membrane
- 3.7 Column Space, Null Space, Orthonormal Bases, and SVD
- 3.8 Computation Time to Run a MATLAB Program
- 4 Methods for Interpolation and Numerical Differentiation**
 - 4.1 Concepts of Interpolation
 - 4.2 Interpolation, Differentiation, and Integration by Cubic Splines
 - 4.2.1 Computing the Length and Area Bounded by a Curve
 - 4.2.2 Example: Length and Enclosed Area for a Spline Curve
 - 4.2.3 Generalizing the Intrinsic Spline Function in MATLAB
 - 4.2.4 Example: A Spline Curve with Several Parts and Corners
 - 4.3 Numerical Differentiation Using Finite Differences
 - 4.3.1 Example: Program to Derive Difference Formulas
- 5 Gauss Integration with Geometric Property Applications**
 - 5.1 Fundamental Concepts and Intrinsic Integration Tools in MATLAB
 - 5.2 Concepts of Gauss Integration
 - 5.3 Comparing Results from Gauss Integration and Function QUADL
 - 5.4 Geometrical Properties of Areas and Volumes
 - 5.4.1 Area Property Program
 - 5.4.2 Program Analyzing Volumes of Revolution
 - 5.5 Computing Solid Properties Using Triangular Surface Elements and Using Symbolic Math
 - 5.6 Numerical and Symbolic Results for the Example
 - 5.7 Geometrical Properties of a Polyhedron
 - 5.8 Evaluating Integrals Having Square Root Type Singularities
 - 5.8.1 Program Listing
 - 5.9 Gauss Integration of a Multiple Integral
 - 5.9.1 Example: Evaluating a Multiple Integral
- 6 Fourier Series and the Fast Fourier Transform**
 - 6.1 Definitions and Computation of Fourier Coefficients
 - 6.1.1 Trigonometric Interpolation and the Fast Fourier Transform
 - 6.2 Some Applications
 - 6.2.1 Using the FFT to Compute Integer Order Bessel Functions
 - 6.2.2 Dynamic Response of a Mass on an Oscillating Foundation
 - 6.2.3 General Program to Plot Fourier Expansions
- 7 Dynamic Response of Linear Second Order Systems**
 - 7.1 Solving the Structural Dynamics Equations for Periodic Forces
 - 7.1.1 Application to Oscillations of a Vertically Suspended Cable
 - 7.2 Direct Integration Methods
 - 7.2.1 Example on Cable Response by Direct Integration

8 Integration of Nonlinear Initial Value Problems

- 8.1 General Concepts on Numerical Integration of Nonlinear Matrix Differential Equations
- 8.2 Runge-Kutta Methods and the ODE45 Integrator Provided in MATLAB
- 8.3 Step-size Limits Necessary to Maintain Numerical Stability
- 8.4 Discussion of Procedures to Maintain Accuracy by Varying Integration Step-size
- 8.5 Example on Forced Oscillations of an Inverted Pendulum
- 8.6 Dynamics of a Spinning Top
- 8.7 Motion of a Projectile
- 8.8 Example on Dynamics of a Chain with Specified End Motion
- 8.9 Dynamics of an Elastic Chain

9 Boundary Value Problems for Partial Differential Equations

- 9.1 Several Important Partial Differential Equations
- 9.2 Solving the Laplace Equation inside a Rectangular Region
- 9.3 The Vibrating String
- 9.4 Force Moving on an Elastic String
 - 9.4.1 Computer Analysis
- 9.5 Waves in Rectangular or Circular Membranes
 - 9.5.1 Computer Formulation
 - 9.5.2 Input Data for Program membwave
- 9.6 Wave Propagation in a Beam with an Impact Moment Applied to One End
- 9.7 Forced Vibration of a Pile Embedded in an Elastic Medium
- 9.8 Transient Heat Conduction in a One-Dimensional Slab
- 9.9 Transient Heat Conduction in a Circular Cylinder with Spatially Varying Boundary Temperature
 - 9.9.1 Problem Formulation
 - 9.9.2 Computer Formulation
- 9.10 Torsional Stresses in a Beam of Rectangular Cross Section

10 Eigenvalue Problems and Applications

- 10.1 Introduction
- 10.2 Approximation Accuracy in a Simple Eigenvalue Problem
- 10.3 Stress Transformation and Principal Coordinates
 - 10.3.1 Principal Stress Program
 - 10.3.2 Principal Axes of the Inertia Tensor
- 10.4 Vibration of Truss Structures
 - 10.4.1 Truss Vibration Program
- 10.5 Buckling of Axially Loaded Columns
 - 10.5.1 Example for a Linearly Tapered Circular Cross Section
 - 10.5.2 Numerical Results

- 10.6 Accuracy Comparison for Euler Beam Natural Frequencies by Finite Element and Finite Difference Methods
 - 10.6.1 Mathematical Formulation
 - 10.6.2 Discussion of the Code
 - 10.6.3 Numerical Results
- 10.7 Vibration Modes of an Elliptic Membrane
 - 10.7.1 Analytical Formulation
 - 10.7.2 Computer Formulation

11 Bending Analysis of Beams of General Cross Section

- 11.1 Introduction
 - 11.1.1 Analytical Formulation
 - 11.1.2 Program to Analyze Beams of General Cross Section
 - 11.1.3 Program Output and Code

12 Applications of Analytic Functions

- 12.1 Properties of Analytic Functions
- 12.2 Definition of Analyticity
- 12.3 Series Expansions
- 12.4 Integral Properties
 - 12.4.1 Cauchy Integral Formula
 - 12.4.2 Residue Theorem
- 12.5 Physical Problems Leading to Analytic Functions
 - 12.5.1 Steady-State Heat Conduction
 - 12.5.2 Incompressible Inviscid Fluid Flow
 - 12.5.3 Torsion and Flexure of Elastic Beams
 - 12.5.4 Plane Elastostatics
 - 12.5.5 Electric Field Intensity
- 12.6 Branch Points and Multivalued Behavior
- 12.7 Conformal Mapping and Harmonic Functions
- 12.8 Mapping onto the Exterior or the Interior of an Ellipse
 - 12.8.1 Program Output and Code
- 12.9 Linear Fractional Transformations
 - 12.9.1 Program Output and Code
- 12.10 Schwarz-Christoffel Mapping onto a Square
 - 12.10.1 Program Output and Code
- 12.11 Determining Harmonic Functions in a Circular Disk
 - 12.11.1 Numerical Results
 - 12.11.2 Program Output and Code
- 12.12 Inviscid Fluid Flow around an Elliptic Cylinder
 - 12.12.1 Program Output and Code
- 12.13 Torsional Stresses in a Beam Mapped onto a Unit Disk
 - 12.13.1 Program Output and Code
- 12.14 Stress Analysis by the Kolosov-Muskhelishvili Method
 - 12.14.1 Program Output and Code

- 12.14.2 Stressed Plate with an Elliptic Hole
- 12.14.3 Program Output and Code

13 Nonlinear Optimization Applications

- 13.1 Basic Concepts
- 13.2 Initial Angle for a Projectile
- 13.3 Fitting Nonlinear Equations to Data
- 13.4 Nonlinear Deflections of a Cable
- 13.5 Quickest Time Descent Curve (the Brachistochrone)
- 13.6 Determining the Closest Points on Two Surfaces
 - 13.6.1 Discussion of the Computer Code

A List of MATLAB Routines with Descriptions

B Selected Utility and Application Functions

References

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
$EI y_{xxxx} = -A \rho y_{tt} + f(x, t)$	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

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

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

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

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

The coefficients in the series can be computed by integration as

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

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

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

$$u_p(x, y) = c_1 + c_2x + c_3y + c_4xy$$

is harmonic and varies linearly along each side of the rectangle. Constants c_1, \dots, 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 dataset
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.

HARMONIC FUNCTION IN A RECTANGLE

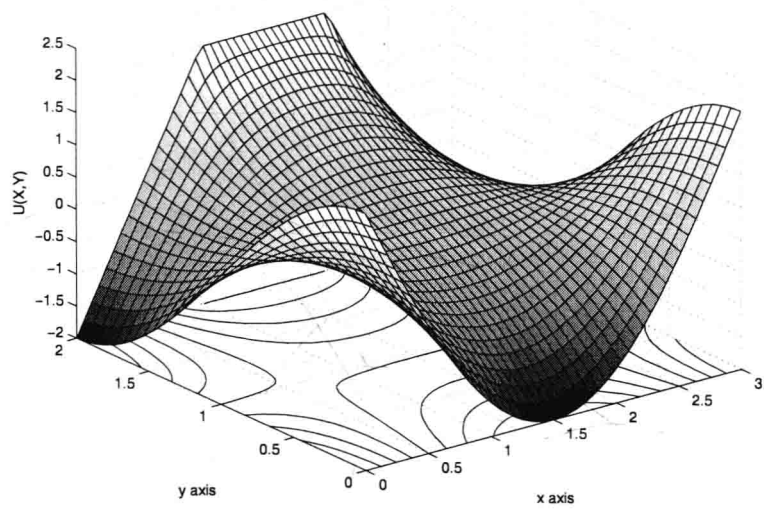


Figure 9.1: Surface Plot of Function Values

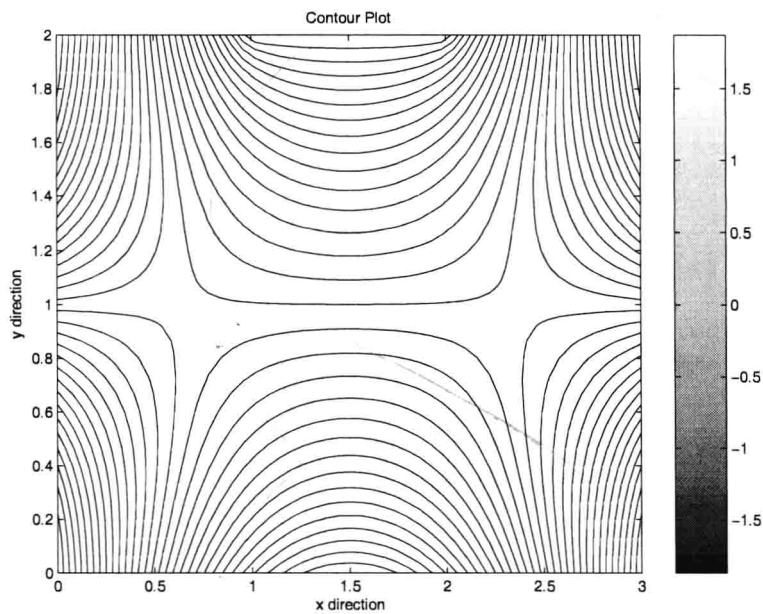


Figure 9.2: Contour Plot of Function Values

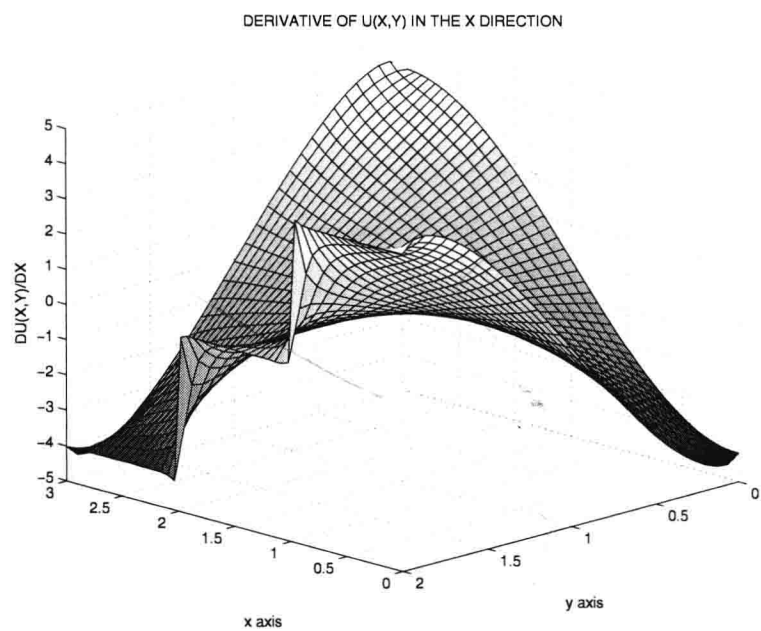


Figure 9.3: Function Derivative in the x Direction

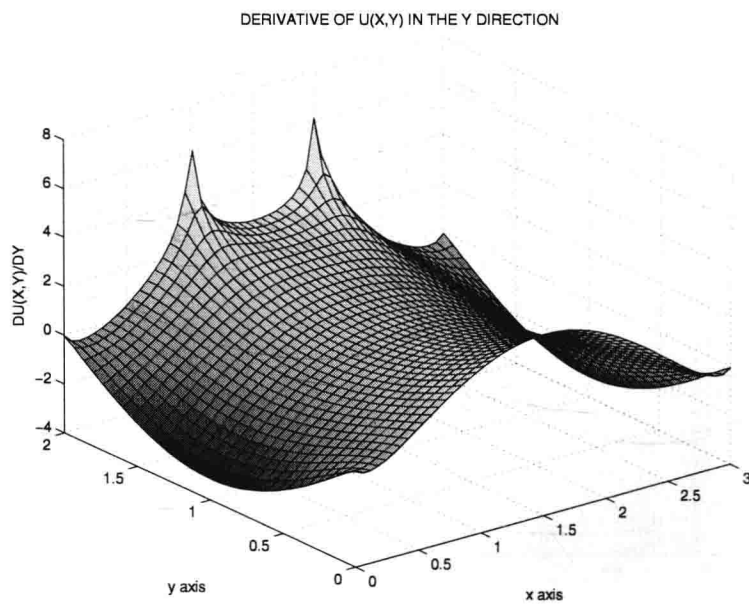


Figure 9.4: Function Derivative in the y Direction

MATLAB Example

Program laplarec

```
1: function [u,ux,uy,X,Y]=laplarec(...
2:         ubot,utop,ulft,urht,a,b,nx,ny,N)
3: %
4: % [u,ux,uy,X,Y]=laplarec(...
5: %         ubot,utop,ulft,urht,a,b,nx,ny,N)
6: % ~~~~~
7: % This program evaluates a harmonic function and its
8: % first partial derivatives in a rectangular region.
9: % The method employs a Fourier series expansion.
10: % ubot - defines the boundary values on the bottom
11: % side. This can be an array in which
12: % ubot(:,1) is x coordinates and ubot(:,2)
13: % is function values. Values at intermediate
14: % points are obtained by piecewise linear
15: % interpolation. A character string giving
16: % the name of a function can also be used.
17: % Then the function is evaluated using 200
18: % points along a side to convert ubot to an
19: % array. Similar comments apply for utop,
20: % ulft, and urht introduced below.
21: % utop - boundary value definition on the top side
22: % ulft - boundary value definition on the left side
23: % urht - boundary value definition on the right side
24: % a,b - rectangle dimensions in x and y directions
25: % nx,ny - number of x and y values for which the
26: % solution is evaluated
27: % N - number of terms used in the Fourier series
28: % u - function value for the solution
29: % ux,uy - first partial derivatives of the solution
30: % X,Y - coordinate point arrays where the solution
31: % is evaluated
32: %
33: % User m functions used: datafunc ulinbc
34: % recseris ftsincf
35:
36: disp(' ')
37: disp('SOLVING THE LAPLACE EQUATION IN A RECTANGLE')
38: disp(' ')
39:
40: if nargin==0
```

```

41:     disp(...)
42:         'Give the name of a function defining the data')
43:     datfun=input(...)
44:         '(try datafunc as an example): > ? ','s');
45:     [ubot,utop,ulft,urht,a,b,nx,ny,N]=feval(datfun);
46: end
47:
48: % Create a grid to evaluate the solution
49: x=linspace(0,a,nx); y=linspace(0,b,ny);
50: [X,Y]=meshgrid(x,y); d=(a+b)/1e6;
51: xd=linspace(0,a,201)'; yd=linspace(0,b,201)';
52:
53: % Check whether boundary values are given using
54: % external functions. Convert these to arrays
55:
56: if isstr(ubot)
57:     ud=feval(ubot,xd); ubot=[xd,ud(:)];
58: end
59: if isstr(utop)
60:     ud=feval(utop,xd); utop=[xd,ud(:)];
61: end
62: if isstr(ulft)
63:     ud=feval(ulft,yd); ulft=[yd,ud(:)];
64: end
65: if isstr(urht)
66:     ud=feval(urht,yd); urht=[yd,ud(:)];
67: end
68:
69: % Determine function values at the corners
70: ub=interp1(ubot(:,1),ubot(:,2),[d,a-d]);
71: ut=interp1(utop(:,1),utop(:,2),[d,a-d]);
72: ul=interp1(ulft(:,1),ulft(:,2),[d,b-d]);
73: ur=interp1(urht(:,1),urht(:,2),[d,b-d]);
74: U=[ul(1)+ub(1),ub(2)+ur(1),ur(2)+ut(2),...
75:     ut(1)+ul(2)]/2;
76:
77: % Obtain a solution satisfying the corner
78: % values and varying linearly along the sides
79:
80: [v,vx,vy]=ulinbc(U,a,b,X,Y);
81:
82: % Reduce the corner values to zero to improve
83: % behavior of the Fourier series solution
84: % near the corners
85:

```

```

86: f=inline('u0+(u1-u0)/L*x','x','u0','u1','L');
87: ubot(:,2)=ubot(:,2)-f(ubot(:,1),U(1),U(2),a);
88: utop(:,2)=utop(:,2)-f(utop(:,1),U(4),U(3),a);
89: ulft(:,2)=ulft(:,2)-f(ulft(:,1),U(1),U(4),b);
90: urht(:,2)=urht(:,2)-f(urht(:,1),U(2),U(3),b);
91:
92: % Evaluate the series and combine results
93: % for the various component solutions
94:
95: [ub,ubx,uby]=recseris(ubot,a,b,1,x,y,N);
96: [ut,utx,uty]=recseris(utop,a,b,2,x,y,N);
97: [ul,ulx,uly]=recseris(ulft,a,b,3,x,y,N);
98: [ur,urx,ury]=recseris(urht,a,b,4,x,y,N);
99: u=v+ub+ut+ul+ur; ux=vx+ubx+utx+ulx+urx;
100: uy=vy+uby+uty+uly+ury; close
101:
102: % Show results graphically
103:
104: surf(X,Y,u), xlabel('x axis'), ylabel('y axis')
105: zlabel('U(X,Y)')
106: title('HARMONIC FUNCTION IN A RECTANGLE')
107: shg, pause
108: % print -deps laprecsr
109:
110: contour(X,Y,u,30); title('Contour Plot');
111: xlabel('x direction'); ylabel('y direction');
112: colorbar, shg, pause
113: % print -deps laprecnt
114:
115: surf(X,Y,ux), xlabel('x axis'), ylabel('y axis')
116: zlabel('DU(X,Y)/DX')
117: title('DERIVATIVE OF U(X,Y) IN THE X DIRECTION')
118: shg, pause
119: % print -deps laprecdx
120:
121: surf(X,Y,uy), xlabel('x axis'), ylabel('y axis')
122: zlabel('DU(X,Y)/DY')
123: title('DERIVATIVE OF U(X,Y) IN THE Y DIRECTION')
124: % print -deps laprecdy
125: shg
126:
127: %=====
128:
129: function [ubot,utop,ulft,urht,a,b,...
130:          nx,ny,N]=datafunc

```