ELLIPTICAL PARTIAL DIFFERENTIAL EQUATIONS:

POISSON AND LAPLACE EQUATIONS

Charles R. O'Neill

School of Mechanical and Aerospace Engineering

Oklahoma State University

Stillwater, OK 74078

2nd Computer Project in MAE 3403

Computer Methods in Analysis and Design

April 1999

ABSTRACT

A formula for solving elliptic partial differential equations using finite differences and iteration was derived. A computer program was made to iteratively calculate the solutions of Laplacian and Poisson elliptic partial differential equations. The results show that the finite differences method does find an approximate solution but can be sensitive to boundary types and step size. Implications of these results are discussed.

INTRODUCTION

Physical processes commonly can be related to the change in the properties of the substance undergoing the process. Those processes that depend on more than two variables are called partial differential equations. Laplace and Poisson's equations are examples of elliptic partial differential equations and are used to model the steady state time-invariant response of physical systems. Slicing the system into small segments of equal length yields a set of finite difference equations which can theoretically be solved using any matrix solving method. The potential for large matrices, which are mostly filled with zeros, makes the iterative method of solving a matrix attractive. A common example of the elliptic partial differential equation is the heated plate, which will be the physical model used even though these equations can be used on many other problems.

The objective is to use an iterative technique to solve partial differential equations of the Poisson and Laplacian form. A computer program will use the derived iterative methods to calculate a steady state response to a physical model specified in a data file. Overrelaxation will be used to decrease the number of computer calculations. Three cases of the Laplace and Poisson equations will be used to verify the validity of the method and computer program. A further three cases will be analyized to determine the solutions to the class project problems given.

THEORY

Partial differential equations (PDEs) depend on the changing of two or more variables. In particular, the solutions to elliptic PDEs are the steady state responses of the system's boundary conditions. Poisson's Equation is

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = f(x, y)$$

The Laplace Equation is a special case of the Poisson equation where $f(x, y) = 0$.

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

Using a 3 point centered formula for the 2nd numerical derivative with respect to x and to y yields the relationship,

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_W + T_E - 2T_P}{\Delta x^2}$$
$$\frac{\partial^2 T}{\partial y^2} = \frac{T_N + T_S - 2T_P}{\Delta y^2}$$

When the x and y spacing are equal,$\Delta x = \Delta y = \Delta$, Poisson's equation becomes,

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = \frac{T_N + T_S + T_W + T_E - 4T_P}{\Delta^2}$$

which can be further simplified to

$$\frac{T_N + T_S + T_W + T_E - 4T_P}{\Delta^2} = f(x, y)$$

Solving for T_P yields,

$$T_P = \frac{T_N + T_S + T_W + T_E - \Delta^2 \cdot f(x, y)}{4}$$

Thus T_P can now be solved iterativly to approximate the solution of the partial differential equation.

Overrelaxation is the scaling of an iterated solution more than the iteration specifies in an attempt to converge the solution faster. By applying the principle of overrelaxation, the number of iterations to converge on the correct solution can be reduced significantly. Overrelaxation can be included by adding and subtracting T_P on the right side of the equation while also specifying the degree of overrelaxation ω. This yields,

$$T_P = T_P + \frac{\omega}{4} \cdot (T_N + T_S + T_W + T_E - 4T_P - \Delta^2 \cdot f(x, y))$$

Where ω is subject to the condition $1 \leq \omega < 2$ For matrices over $[10 \times 10]$ and where p and q are the number of x and y intervals respectivly, ω can be approximated by

$$\omega \approx 2 - \pi \cdot \sqrt{\frac{2}{p^2} + \frac{2}{q^2}}$$

METHOD OF CALCULATION

A computer program written in FORTRAN 77 was developed to calculate the solution to Poisson's and the Laplacian equation by iteration. The program first initializes a matrix T to store the temperature values at all the x and y values. A data file is used to input the necessary boundaries, boundary conditions, spacing and internal heat generation values. When the east temperature is given as -999 or smaller, the program assumes an adiabatic east boundary. All temperature values are implicitly declared floating point single precision. All index values are implicitly declared as integers. Next, a series of DO loops initialize the boundary values with the values obtained from the data file. Three nested DO loops iteratively calculate the value of each point inside of the external edge points using the method derived in the theory section. When the program uses an adiabatic boundary on the east boundary, a fictional point just outside of the adiabatic boundary is made to reflect the temperature seen at the point just inside the adiabatic boundary. The matrix of temperatures is printed at specified intervals and the end iteration while inside the outer DO loop. When an adiabatic boundary is specified, the fictional points are also printed. The program then exits.

RESULTS AND DISCUSSION

Calculations have been performed as described above on three test cases and three project problems. The three test cases consist of one trivial case, one internal heat generation case and one adiabatic boundary case. The project problems are those given in the MAE 3403 class and consist of one internal heat generation problem, a boundary temperature case and an adiabatic boundary case.

For Case 1, a trivial solution case, a square plate was heated from an initial temperature of zero degrees to steady state with 100 degree boundaries at the edges. The solution of 100 degrees throughout the plate was found as expected by the boundary conditions (Appendix B. Case 1). The contour plot is given in Figure C1 (Appendix C).

For Case 2, a square plate was internally heated with boundary temperatures of zero degrees. The internal heat generation should create a rounded rise in the plate temperature and the data (Appendix B. Case 2) confirms this. A contour plot is given in Figure C2.

For Case 3. a square plate was heated at the north, south and west boundary while the east boundary had a adiabatic surface. This case is solved as example 29.3 in Chapra(1998). The data (Appendix B. Case 3) gives the fictional temperatures past the adiabatic surface boundary; however, all other data conforms with Chapra's solution. A contour plot is given in Figure C3.

Problem 7.1 considers a plate with an internally generated heat source with boundary temperatures of zero degrees. The temperatures and boundaries were as followed: T(0,y)=0, T(x,0)=0, T(3,y)=0 and T(x,1)=0. The function to be solved is

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = -21$$

The computer program returned final answers of 2.56 degrees at the center (x=1.5, y=.5) of the plate (Appendix B. Problem 7.1). The values are similar to the test Case 2 in that the temperature reaches a rounded peak maximum value at the center of the plate (Figure P1). The peak is symmetric when divided along the center of the plate for both the x and y axis. A physical type of solution such as a long-uninsulated bar carrying electricity could be modeled with this type of equation and boundary conditions.

Problem 7.2a is a square plate with boundary temperatures of different temperatures as followed: North=100, South=300 and East=West=192. The data file is given (Appendix B. Problem 7.2a). The function to be solved is

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

Two calculations were performed with different grid sizes. The middle temperature is the same between the fine and coarse grid at 196 degrees. Extrapolating to the limit at the quarter temperature (y=0.25, x=0.25) yields

$$T_{exact} \approx T_{fine} + \frac{1}{3}(T_{fine} - T_{coarse}) = \frac{4}{3} \cdot 194.94 - \frac{1}{3} \cdot 195.0 = 194.92$$

This problem's contours (Figure P2) would be typical of a plate subjected to different temperatures.

Problem 7.2b reduces the computation of 7.2a by considering the plate as having an adiabatic surface at the symmetry line of 7.2a. This symmetry line is at x values of 0.5 after which the plate can be truncated. The contour plot (Figure P4.) gives the same solution as the left side of Figure P2. The solution (Appendix B. Problem 7.2b) shows the same data as in Problem 7.2a up to one value past the adiabatic line.

The iterative solution to Poisson and Laplace equations does work. Depending on the accuracy, the method can provide a rough, poor solution or a solution that approaches the precision of the computer. Results from the test cases did show a stable and repeatable method. This method is adaptable to many types of problems and can easily be modified for different conditions and boundary types. Iteration does have problems. Most serious is that the method uses finite segments. This prevents the exact answers from occurring and limits high quality answers to those calculations with many iterations. A second problem, slow convergence, is inherited from the first problem due to the number of points required.

This slow convergence can be offset to some degree using overrelaxation techniques, but high quality answers with large systems still require many calculations. Using properties of symmetry can help the computational load as shown in Problem 7.2b but only for larger systems. Not calculating the extra half of the plate did allow for a slight shorter computation; however, when the plate is modeled with a few number of points as in the coarse grid, the overhead of calculating the adiabatic boundary almost equals the calculation of the entire plate. A final problem is that the modeling method breaks at corners due to the presence of two different temperatures with no separating distance and maintaining the same step value. This can be seen in the corners of the contour plots in many of the Appendix C figures.

CONCLUSION

Iteration when combined with finite difference derivative and properly used provides good quality solutions to partial differential equations. The method is computationally better than solving plate temperatures by elimination matrix methods for sparse matrices; however, it still has faults that can create problems. Particularly, the method is dependent upon finite differences which creates many computations and no exact answers. When large computing power is available and the problem is sufficently formulated, the finite difference equations combined with iteration is powerful and adaptable.

REFERENCES

Chapra, Steven C., Raymond P. Canale (1998)

Numerical methods for engineers: with programming and software applications,

WCB/McGraw-Hill.

APPENDIX A

Computer Program

APPENDIX B

Input and Output

APPENDIX C

Figures

TEST CASES:

Figure C1. Contour plot of test case 1

Figure C2. Contour plot of test case 2

Figure C3. Contour plot of test case 3

PROJECT SOLUTIONS

Figure P1. Contour plot of 7.1

Figure P2. Contour plot of 7.2a coarse grid

Figure P3. Contour plot of 7.2a fine grid

Figure P4. Contour plot of 7.2b coarse grid

Figure P5. Contour plot of 7.2b fine grid