1983

Algorithm GENCOL: Collocation on General Domains with Bicubic Hermite Polynomials

Elias N. Houstis
Purdue University, enh@cs.purdue.edu

W. F. Mitchell

John R. Rice
Purdue University, jrr@cs.purdue.edu

Report Number:
83-444

ALGORITHM GENCOL: COLLOCATION ON GENERAL DOMAINS WITH BICUBIC HERMITE POLYNOMIALS

E.N. Houstis
Applied Mathematics
University of Thessaloniki

W.F. Mitchell
J.R. Rice
Computer Science
Purdue University

CSD-TR 444, June 30, 1983

ABSTRACT

This paper presents an algorithm for the collocation method described in the companion paper Collocation Software for Second Order Elliptic Partial Differential Equations by Houstis, Mitchell and Rice. The problem solved has a general elliptic linear operator with variable coefficients, general linear boundary conditions and a general two dimensional domain. This algorithm uses the output of the domain processor [Rice, 1982] (which can be generated "by hand" for simple domains). The basis functions used are bicubic Hermite polynomials defined on a tensor product grid covering the problem domain. This paper describes the driver, use of test problems and the files of the algorithm. The input and output of the algorithms themselves are documented in the initial comments of the algorithm. This documentation is reproduced here.

ALGORITHM GENCOL: COLLOCATION ON GENERAL DOMAINS WITH BICUBIC HERMITE POLYNOMIALS

E.N. Houstis, W.F. Mitchell and J.R. Rice

The algorithm GENCOL presented here implements the collocation method as described in [Houstis, et al, 1983a]. This algorithm is a descendent of the programs in [Houstis and Rice, 1977] and [Houstis et al, 1978], the present algorithm has been completely rewritten and many substantial changes made in the methods. Algorithms for the much simpler case of rectangular domains are presented in [Houstis et al, 1983b]; they share the same framework as GENCOL. The general domain is assumed to be processed by the algorithm of [Rice, 1982], the input expected from this program is described in the comments and the files include actual data for two examples. This input can be prepared by hand for simple domains, but it becomes quite tedious for more complex ones or for fine meshes. The driver provided also includes a program to format the linear system as a band matrix and a program to solve the system by Gauss elimination with scaled partial pivoting.

This algorithm is included in the ELLPACK system [Rice and Boisvert, 1983] as COLLOCATION.

A code skeleton of the used of this algorithm follows:

Main Program
```
DEFINE PROBLEM TO BE SOLVED
PROCESS THE GRID AND DOMAIN INFORMATION
INVOKE COLLOCATION ALGORITHM
FORMAT LINEAR SYSTEM AS BAND MATRIX
SOLVE LINEAR SYSTEM
PRODUCE REQUESTED OUTPUT
```

Subprograms
```
FUNCTIONS FOR DOMAIN DEFINITION
FUNCTIONS FOR OPERATOR AND BOUNDARY CONDITIONS
```

If the domain processor [Rice, 1982] is not used, then the functions for the domain definition are replaced by a file of input data to be read by the main program.

In order to make this algorithm self contained, we include three programs somewhat unrelated to it:

```
REGION: the domain processor (it has many subprograms)
SETUP : to create the band matrix
BANDGE: to solve the linear system
```

The software distributed with the algorithm consists of the following files:

file 1: driver
MAIN PROGRAM
SUBPROGRAM REGION (and all related subprograms)
SUBROUTINE SETUP
SUBROUTINE BANDGE
SUBROUTINE OUTPUT

file 2: example 1
FUNCTION BCOORD
FUNCTIONS PDE,BCOND

file 3: algorithm GENCOL
SUBPROGRAMS UNIQUE TO GENGOL
SUBPROGRAMS COMMON TO GENCOL, INTCOL AND HERMCOL

file 4: example 2
FUNCTION BCOORD
FUNCTIONS PDE,BCOND

file 5: example 1 domain data
SUBROUTINE REGION (to replace REGION above)
DATA

file 6: example 2 domain data
SUBROUTINE REGION (to replace REGION above)
DATA

The first three files can be compiled and executed to run GENCOL for the test case of Example 2 (non-rectangular domain) of the companion paper. If file 4 is substituted for file 2 and appropriate changes made in the driver (as indicated by comments), then the test case of Example 4 of the companion paper is ran. If one does not want to compile the domain processor (it is quite large), one may replace in file 1 the SUBROUTINE REGION and all related subroutines by the SUBROUTINE REGION in file 5, which reads the data in file 5 for the first test cases. Similarly, file 6 allows one to replace the domain processor for the second test case.

Variations on the two test cases can be obtained by changing the following variables in the driver (provided the domain processor is used):

AX,BX = x-grid limits
AY,BY = y-grid limits
NGRIDX,NGRIDY = number of x,y grid lines
NOUT = number of outputs selected
OUTFNC(I), (I=1 to NOUT) = functions for output

OUTTYP(I), (I=1 to NOUT) = types of output

REFERENCES

E.N. Houstis, R.E. Lynch and J.R. Rice (1978), Evaluations of numerical methods for elliptic partial differential equations, J. Comp. Physics, 27, 323-350.

E.N. Houstis, W.F. Mitchell and J.R. Rice (1983a), Collocation software for second order elliptic partial differential equations, CSD-TR 446, Computer Science Department, Purdue University.

E.N. Houstis, W.F. Mitchell and J.R. Rice (1983b), Algorithms INTCOL and HERMCOL: Collocation on rectangular domains with bicubic Hermite polynomials, CSD-TR 445, Computer Science Department, Purdue University.

E.N. Houstis and J.R. Rice (1977), Software for linear elliptic problems on general two dimensional domains, in Advances in Computer Methods for Partial Differential Equations II, (Vishnevetsky, ed.), IMACS, Rutgers University, New Brunswick, N.J.

J.R. Rice (1982), Algorithm: A two-dimensional domain processor, CSD-TR 417, Computer Science Department, Purdue University.

J.R. Rice and R.F. Boisvert (1983), Solving Elliptic Problems using ELLPACK, Springer-Verlag, New York.