1975

Development, Evaluation & Selection of Methods for Elliptic Partial Differential Equations

E. N. Hnoustis

Robert E. Lynch
Purdue University, rel@cs.purdue.edu

T. S. Papatheodorou

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

Report Number:
75-134

DEVELOPMENT, EVALUATION AND SELECTION OF METHODS FOR ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS

E. N. Houstis, R. E. Lynch,
T. S. Papatheodorou and J. R. Rice
Mathematical Sciences, Purdue University

January, 1975

CSD-TR 134

DEVELOPMENT, EVALUATION AND SELECTION OF METHODS FOR ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS

E. N. Houstis, R. E. Lynch,
T. S. Papatheodorou and J. R. Rice*
Mathematical Sciences, Purdue University

Abstract

We present a framework within which to evaluate and compare computational methods to solve elliptic partial differential equations. We then report on the results of comparisons of some classical methods as well as a new one presented here. Our main motivation is the belief that the standard finite difference methods are almost always inferior for solving elliptic problems and our results are strong evidence that this is true. The superior methods are higher order (fourth or more instead of second) and we describe a new collocation finite element method which we believe is more efficient and flexible than the other well known methods, e.g., fourth order finite differences, fourth order finite element methods of Galerkin, Rayleigh-Ritz or least squares type.

Our comparisons are in the context of the relatively complicated problems that arise in realistic applications. Our conclusion does not hold for simple model problems (e.g., Laplaces equation on a rectangle) where very specialized methods are superior to the generally applicable methods that we consider. The accurate and relatively simple treatment of boundary conditions involving curves and derivations is a feature of our collocation method.

Keywords: Algorithm selection, Collocation, Comparisons of algorithms, Elliptic partial differential equations, Finite differences, Finite elements

* Work partially supported by the National Science Foundation grant GP 32940X

1. INTRODUCTION. The first goal of this paper is to outline a framework in which to compare computational methods to solve elliptic partial differential equations. We believe that finite element methods are significantly superior to the standard finite difference methods and we expect that an algorithm comparison procedure as described here to provide solid and precise evidence to this effect. We assume the reader is familiar with these computational methods and we refer the reader to [1] for an up-to-date presentation of finite element methods. We note that they mention the comparison of these two methods and indicate (page 12) the need for planned experiments on the effectiveness of various computational methods.

The second goal of this paper is to present some concrete computational comparisons. These results have neither the range nor the precision that we expect to obtain in the future. However, they do indicate the general nature of the results to be obtained and provide strong evidence for prefering finite element methods over standard finite difference methods.

The next section briefly describes the algorithm selection framework and problem domain. The third section describes the metalgorithms and computational components for finite difference and finite element methods. We report on a count of the possible choices of metalgorithm components and note that there are tens of thousands of distinctly different computational methods of these types. We clearly cannot consider them all. One of the major open questions in the area of algorithm selection is to decide how to judge whether significantly better algorithms have been overlooked. In Section 4 we describe our comparison of methods. This comparison is only between two particular algorithms, one for each of finite difference and finite elements. We take what we think of as the most straightforward versions of these two methods. The resulting collocation method for finite elements is new, but we feel that it is an attractive choice. The final section contains the computational results.

A comparative evaluation of the finite difference and finite element method for elliptic partial differential equations has been attempted by Schultz [7], Eisenstat and Schultz [2], Rice [4], Birkhoff and Fix [1]. However, we should notice that their results are based only on the asymptotic arithmetic operation count and thus their applicability is very restricted.

2. ALGORITHM SELECTION AND THE PROBLEM SPACE. Rice [6] presents an abstract framework for considering the selection of algorithms. His basic model includes a problem space, an algorithm space, a subset of algorithms from which the selection is to be made and measures of the performance of algorithms. Within this framework, he identifies several types of problems and we are concerned here with "Best selection from a subclass of mappings (algorithms) and problems." The subclass of mappings that we want to consider are described as metalgorithms in the next section. In our actual comparisons, the subclass is further reduced to just two specific algorithms.

The problem space we would like to consider is as follows. Assume $u = (u^1,\ldots,u^n)$ satisfies the system of partial differential equations

$$G^i(x^1,\ldots,x^m,u^1,\ldots,u^1_{x^1},\ldots,u^n_{x^m},u^1_{x^1x^1},\ldots) = 0,\ i = 1,\ldots,n$$

in a given domain and certain auxiliary conditions on the boundary of the domain. The problem is: Given $\varepsilon > 0$ then estimate u within ε. We assume that the system of partial differential equations is elliptic. For a reasonable level of generality, we assume that the given domain Ω, is open, connected subset of R^m and that its boundary $\partial\Omega$ consists of a finite number of piece-wise smooth curves. We assume that the auxiliary conditions take the form of Dirichlet, Neumann or mixed boundary conditions.

This problem class is extemely large and must be reduced considerably in any currently practical study of computational methods. The comparisons made here specializes this space to linear problems in two variables and it chooses a small, but a hopefully representative, sample from this subset. Each problem in the problem space is determined by the following four attributes: (i) the geometry of the domain of definition, (ii) the differential operator, (iii) the auxiliary conditions, and (iv) the specified accuracy.

3. METALGORITHMS FOR FINITE DIFFERENCE AND FINITE ELEMENT METHODS. According to Rice [6] the word metalgorithm means a framework or theory to study algorithms. A metalgorithm consists of a set of blocks or components (possibly in flowchart form) and it represents a class of algorithms,

each of which has the form and attributes specified by the metalgorithm. Two metalgorithms are outlined here and an analysis of metalgorithm component choices is summarized in section 3.3.

3.1 Metalgorithm for finite difference methods. This metalgorithm consists of five components:

(i) a grid of points over the domain Ω that we call pivots if they lie in Ω or on $\partial\Omega$. We distinguish interior and boundary pivots according to whether their surrounding grid points are pivots or not.

(ii) a processor that generates a set of algebraic equations from the operator equations.

(iii) a processor that generates a set of algebraic equations from the auxiliary conditions.

(iv) an equation solver for the system equations of (ii) and (iii).

(v) measurement of the results and termination of the algorithm.

The computations represented by this metalgorithm consist of the determination and/or execution of these 5 components in the sequence listed.

3.2 Metalgorithm for finite element methods. This metalgorithm consists of six components:

(i) a partition of the domain Ω into a set of finite elements.

(ii) a choice of approximation functions associated with the finite elements (e.g., continuous and linear, Hermite cubics).

(iii) a processor that generates a set of algebraic equations from the operator equations.

(iv) a processor that generates a set of algebraic equations from the auxiliary conditions.

(v) an equation solver for the system of equations generated by components (iii) and (iv).

(vi) measurement of results and termination of the algorithm.

The computations represented by this metalgorithm consist of the determination and/or execution of these 6 components in the sequence listed.

3.3 Metalgorithm Component Choices. It is clear that there are several choices for any particular component of these two metalgorithms. We have made a detailed list of the possible choices and conservatively estimate their numbers as follows. For the finite difference metalgorithm there are, respectively, 4, 4, 4 and 5 choices for the first four components. For the finite element metalgorithm there are, respectively, 5, 8, 4, 6 and 3 choices for the first five components. One must realize that there are, in fact, further interesting possibilities not counted here and that there are a variety of significant variations in each of our listed choices. One concludes then that the 4*4*4*5 = 320 basic finite difference combinations becomes at least 10,000. Likewise, one concludes that the 5*8*4*6*3 = 2880 basic finite element possibilities lead to at least 100,000 significantly distinct choices for a finite element computational method (computer program). Note that these estimates do not count seemingly trivial variations in program implementation which do, in fact, affect the computational performance significantly.

It is clear that we cannot consider all these possible computational algorithms now and perhaps no one ever can.

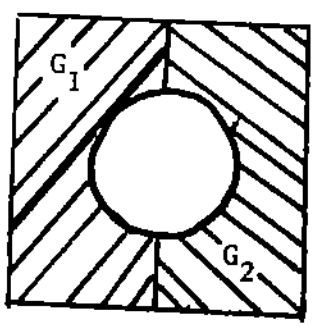
4. SPECIFIC CHOICES FOR A COMPARITIVE EVALUATION.

4.1 The Problem Subset. We consider eight equations which are tabulated in Table 1. We have several problems for each equation by having different values of specified accuracy ranging from one to four significant digits. In some cases an unspecified function t or g appears; they are determined so that the solution is as listed in the table.

Problems 2 and 3 require further explanation. They arise from the analysis of the torsion of a hollow bimetalic shaft. The geometry of the shaft is illustrated on the left in Figure 1 and the domain of the partial differential equation is shown on the right. The shear modulus G is a step function with the ratio = 3 for the two values. Symmetry is used (via Neumann boundary conditions) to restrict consideration to the upper half of the domain. The physical problem has the boundary condition that the stress u be constant on the inside boundary. This constant is

Table 1. The eight equations used in the comparisons reported in this paper.

No.	Operator Equation	Domain	Boundary Conditions	Solution
1.	$(e^{xy}u_x)_x + (e^{-xy}u_y)_y - \frac{u}{1+x+y} = f$	Unit Square	$u=0$	$e^{xy}\sin\pi x \sin\pi y$
2/3	$(\frac{1}{G}u_x)_x + (\frac{1}{G}u_y)_y = -2\theta$	Figure 1	see text	unknown
4.	$u_{xx} + u_{yy} = f$	Ellipse	$u=g$	$(e^x+e^y)/(1+xy)$
5.	$u_{xx} + u_{yy} = 0$	Circle	$u=g$	$\tan^{-1}\frac{y}{x}$
6.	$u_{xx} + (1+y^2)u_{yy} - u_x - (1+y^2)u_y = f$	Unit Square	$u-u_N=0$	$e^{x+y} + (x^2-x)^2 \log(1+y^2)$
7.	$u_{xx} + u_{yy} = -6xy\, e^x e^y (xy+x+y-3)$	Unit Square	$u=0$	$3e^x e^y (x-x^2)(y-y^2)$
8.	$u_{xx} + u_{yy} = f$	Unit Square	$u=g$	$x^{5/2}\, y^{5/2}$



$G_1/G_2 = 3$

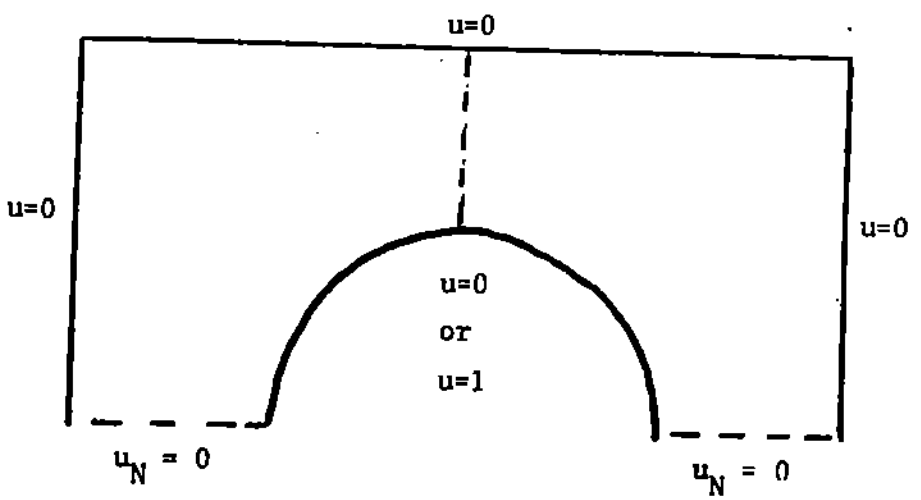


Figure 1. Illustration of the problem and domain for equations 2 and 3.

unknown, but there is a standard approach [3] to determine it by solving two problems. The first (we call it number 2) is with θ zero and $u = 1$ on the inside boundary. The second (we call it number 3) is with θ not zero and $u = 0$ on the inside boundary. The error in the computed solutions to these problems were measured by comparing with a more accurate finite element method. This approximate solution agrees with previously published ones, except that it is the most accurate.

4.2 The Finite Difference Method. Our specific choices for the components of the finite difference metalgorithm are:

(i) Grid: uniform grid

(ii) Interior pivot equations: the 5-point star difference approximation

(iii) Boundary pivot equations: For Dirichlet boundary conditions we use the same method as for interior pivot equations, i. e., a 5-point star with possibly non-uniform spacing. For Neumann boundary conditions we first estimate u_x and u_y by second degree Lagrange interpolations formula in the grid directions. One then uses $u_N = u_x \cos\alpha + u_y \sin\alpha$ where α is the angle the normal makes with the x-axis. For curved boundaries, one of the required derivatives is usually not directly estimable and we estimate it by further interpolation across grid lines.

(iv) Equation solver: we choose Profile Gaussian elimination. We realize that various iterative methods may be superior to Gaussian elimination. They are, however, more difficult to use in the variety of problems we consider because of unknown rates of convergence and relaxation parameters.

(v) Measures of performance: execution time and memory used.

4.3 The Finite Element Method

Our specific choices for the components of the finite element method are:

(i) Elements: a tensor produce of intervals to give rectangles, some of which may overlap the boundary.

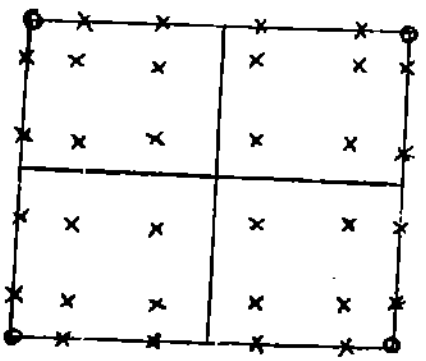
(ii) Approximation space: the bicubic Hermite rectangular elements.

(iii) Operator approximation equations: Collocation, we determine the approximate solution so that it satisfies the operator equation at

four Gaussian points inside each element.

(iv) Auxiliary conditions equations: Collocation, the rest of the degrees of freedom of the approximate solution ($4S_b + 4$ where S_b = number of boundary sides) are determined by requiring the approximate solution to satisfy the auxiliary conditions. For curved boundaries our procedure is, roughly speaking, to trim the element along the boundary, move the interior collocation points to remain interior and to distribute the boundary collocation points from the approximate rectangular boundary to the curved boundary. There are several details and special cases to be considered, but the scheme is both accurate and uniformally applicable provided the grid of elements is fine relative to the oscillations in the domain boundary.

The collocation points are illustrated below for the simple case of 4 elements in a rectangle



(v) Equation solver: profile Gaussian elimination.

(vi) Measures of performance: Execution time and memory used.

PRELIMINARY COMPARISONS. We summarize our comparison results in Figures 2 and 3. The computer programs for the finite difference and finite element methods are polished to an equal extent; each is what one expects from a straightforward Fortran implementation by a knowledgeable person. The execution times and memory used are for a CDC 6500.

We give a single graph for each equation which shows accuracy in solving the equation plotted against computer time and computer memory used for the two methods. The cross-over point is where the two methods are of approximately equal efficiency (fortunately, both measures of performance give the same cross-over points).

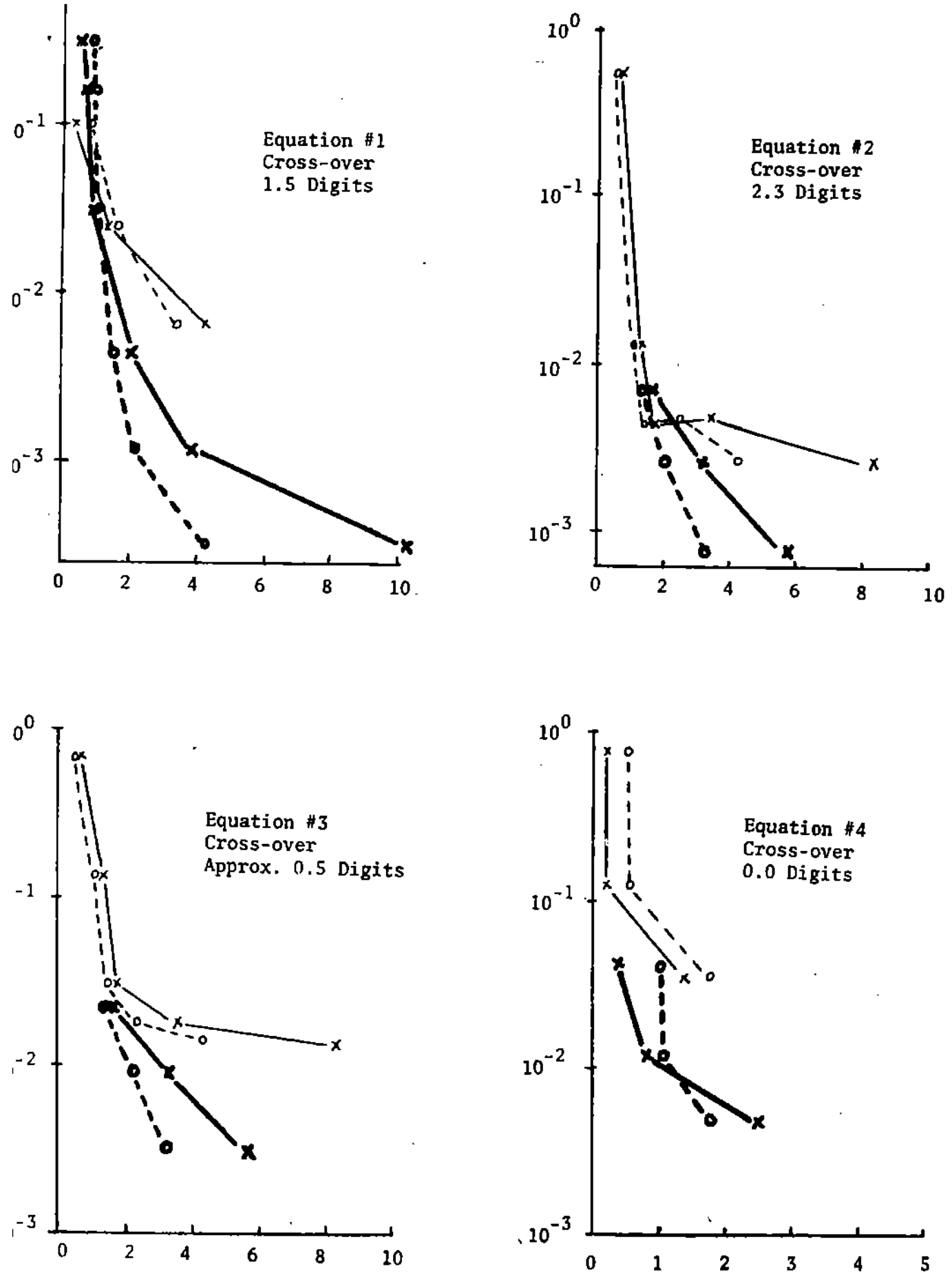
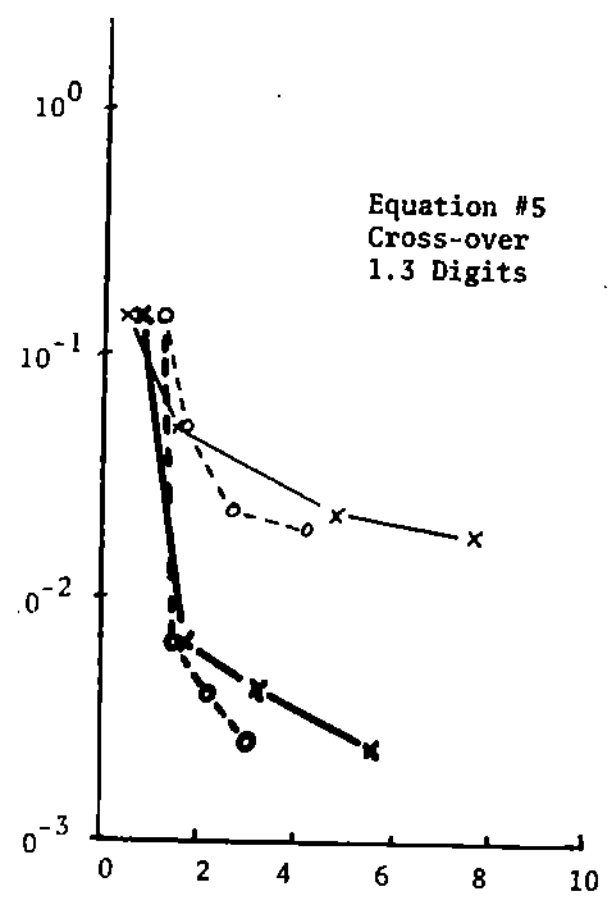
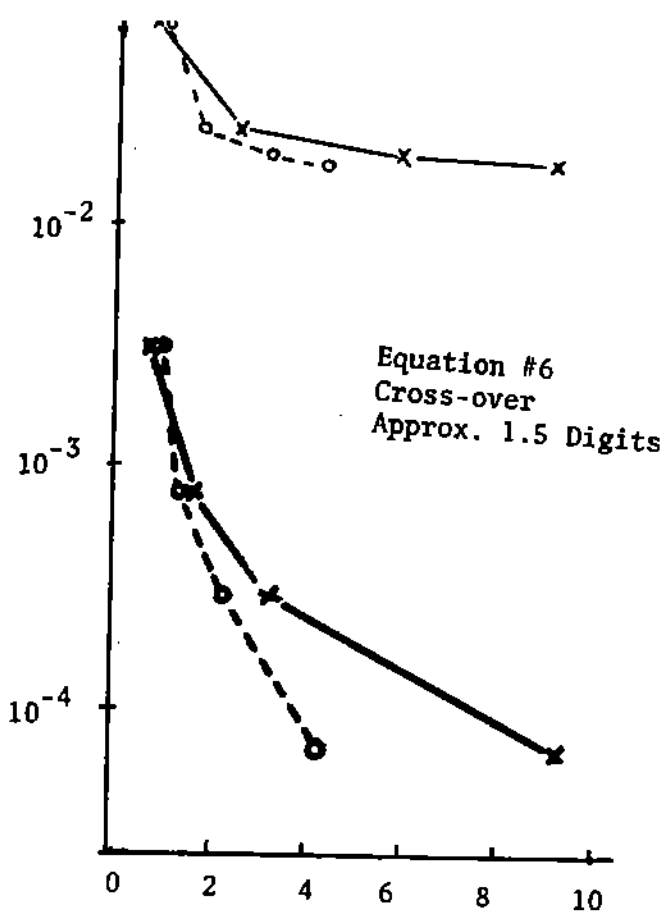
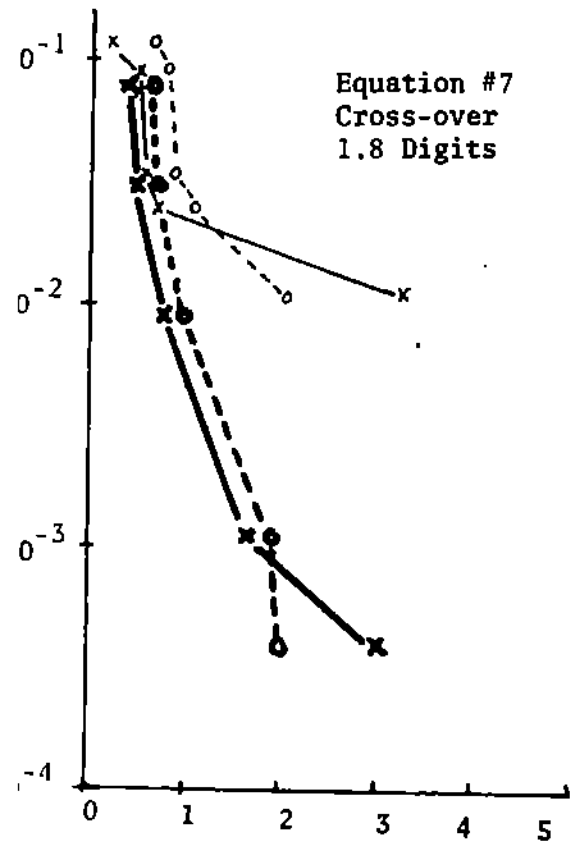


Figure 2. Comparisons for equations 1 to 4. The vertical logarithmic scale is the maximum error in the approximate solution. The horizontal scale is execution time in seconds (CDC 6500) and memory in units of 8,000 words. Heavy lines are for finite elements, light for finite differences; solid for execution time, dashed for memory.







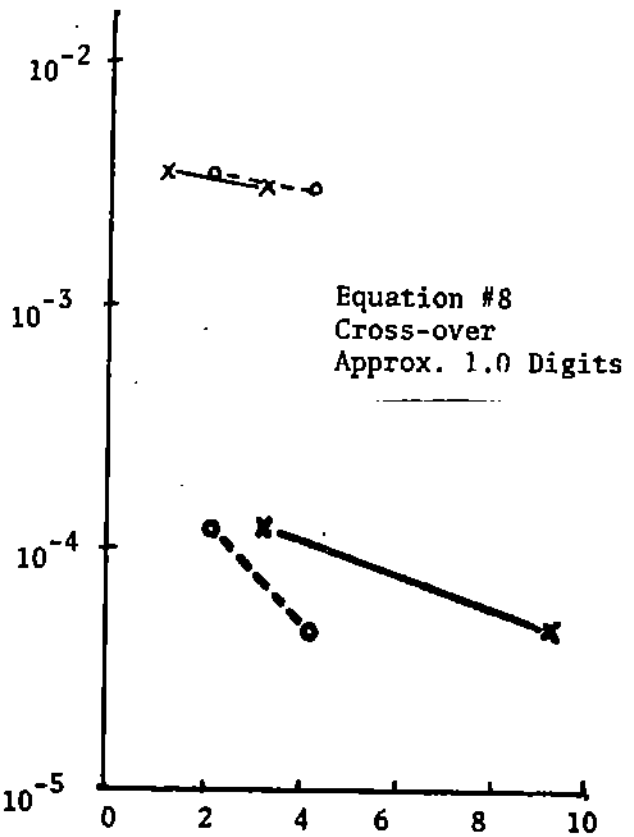


Figure 3. Comparisons for equations 5 to 8. The vertical logarithmic scale is the maximum error in the approximate solution. The horizontal scale is execution time in seconds (CDC 6500) and memory in units of 8,000 words. Heavy lines are for finite elements, light for finite differences: solid

These comparisons strongly support the contention that finite element methods are almost uniformly superior to the classical finite difference methods. As one would expect, the exceptions occur for very low accuracies, cruder than one expects to need in most scientific applications. If even moderate accuracy, say 3 digits, is needed, then the finite element method has a dramatic advantage. There are many features of these results than may be explained by known asymptotic error estimates. Some others (e.g., results for equations 4, 6 and 8) are not, which merely reflects that asymptotic estimates are frequently misleading when applied in non-asymptotic situations. Space precludes a detailed discussion of these comparison on an equation by equation basis.

We also give a comparison of the error (but not the efficiency) for three different finite element methods in Table 2 and for equation 1.

Number of cubic polynomial elements	Rayleigh-Ritz Galerkin [2]	Least Squares [2]	Collocation
1	---	---	3.0E-1
2	---	---	1.4E-1
4	1.5E-2	4.3E-2	3.2E-2
6	---	---	4.8E-3
9	1.9E-3	2.3E-3	1.3E-3
16	3.4E-4	4.4E-4	2.7E-4
25	1.0E-4	1.3E-4	---

Table 2. The maximum error versus the number of elements for solving Problem 1 by three different finite element methods, all of which use cubic polynomial basis functions.

REFERENCES

[1] Garrett Birkhoff and G. J. Fix, [1973], Higher-order linear finite element methods. AEC and ONR report, 33 pages.

[2] S. C. Eisenstat and M. H. Schultz, [1973], Complexity of partial differential equations, in "Complexity of Sequential and Parallel Numerical Algorithms". (Ed., J. F. Traub), Academic Press, pp. 271-282.

[3] J. F. Ely and O. C. Zienkiewicz, [1960], Torsion of compound bars - a relaxation solution. Int. J. Mech. Sci. 1(1975) pp. 356-365.

[4] J. R. Rice, [1972], On approximation theoretic methods for operator equations. Unpublished manuscript, 48 pages.

[5] J. R. Rice, [1974], The algorithm selection problem: Abstract Models, Computer Sciences Dept. CSD-TR 116, Purdue University, May 1974, 21 pages.

[6] J. R. Rice, [1975], A metalgorithm for adaptive quadrature. J. Assoc. Comp. Mech., 22(1975) pp. 61-82.

[7] H. Schultz, [1972], The Computational Complexity of Elliptic Partial Differential Equations, in "Complexity of Computer Computations" (Ed., R. E. Miller and J. W. Thatcher), Plenum Press, pp. 73-84.