1981

Performance Analysis of 13 Methods to Solve the Galerkin Method Equations

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

Report Number:
81-369

Rice, John R., "Performance Analysis of 13 Methods to Solve the Galerkin Method Equations" (1981).
Department of Computer Science Technical Reports. Paper 297.
<https://docs.lib.purdue.edu/cstech/297>

PERFORMANCE ANALYSIS OF 13 METHODS TO SOLVE THE
GALERKIN METHOD EQUATIONS

John R. Rice
Mathematical Sciences
Purdue University

CSD-TR 369
May 25, 1981

Abstract

We report on an experimental study of the effectiveness of 13 methods to solve the systems of linear equations obtained using the Galerkin method with bicubic Hermite polynomial basis functions applied to two dimensional elliptic partial differential equations. The study concludes that, within 99% confidence levels, the iteration methods considered provide no advantage over the usual Gauss elimination methods. The cross-over point for iteration methods becoming most efficient is usually for about an 11 by 11 grid (observed range: 7 by 7 to 17 by 17). These results support the conjecture that iteration with optimal parameter is as effective for finite element method systems of equations as it is known to be for finite difference method equations. These results are not in agreement with theoretical expectations about the asymptotic behavior of sparse matrix methods, some possible sources of the discrepancy are listed.

*This work supported in part by National Science Foundation grants MCS-76-10225, MCS-77-01408.

PERFORMANCE ANALYSIS OF 13 METHODS TO SOLVE THE GALERKIN METHOD EQUATIONS

John R. Rice*

I. THE EXPERIMENT

The Galerkin method applied to a linear elliptic partial differential equation generates a large system of linear equations to be solved. These systems are traditionally solved by some form of Gauss elimination. In this paper we study the effectiveness of iteration, elimination and sparse matrix methods for such systems. We present evidence that strongly supports the hypothesis that iteration methods are the most effective and sparse matrix methods provide little or no advantage over ordinary Gauss elimination for large systems of these equations. A more detailed hypothesis is stated in the final section. We also discuss which method of each class is the most efficient, where the cross-over points are (the iteration methods are relatively less efficient for smaller systems, say up to 500 equations) and several other observations are made.

The study is experimental of the following nature: A set of 13 partial differential equations (PDEs) are chosen from the population of [Rice et al, 1981], their numbers are 1-1, 4-1, 5-1, 5-4, 6-1, 7-1, 10-2, 10-3, 28-3, 41-1, 44-1, 44-2, 44-3. Each of these PDEs is self-adjoint on a rectangular domain with homogeneous boundary conditions, the actual operators are listed in the appendix. The Galerkin method equations are generated by the program P3C1 GALERKIN with the ELLPACK system [Rice, 1981] and are in the natural ordering (see Figure 2(a) for a typical pattern of non-zero elements).

These equations are then solved by modules in the ELLPACK system:

- LINPACK : A program for symmetric, positive definite band matrices [Dongara et al, 1979].
- YALE ENVELOPE : A program for symmetric matrices in envelope form [Eisenstat et al, 1977].
- YALE SPARSE : A LDL^T factorization for symmetric matrices in a sparse matrix representations [Eisenstat et al, 1977].
- SPARSE GE : A program for Gauss elimination with pivoting for matrices in a sparse matrix representation [Sherman, 1978].
- SOR : A program for SOR iteration [Kincaid et al, 1979].
- JACOBI CG : A program for the Jacobi method accelerated by a conjugate gradient technique [Kincaid et al, 1979].

All the above modules apply to the Galerkin equation in the natural order.

Two other orderings in ELLPACK were used:

- RCM : Reverse Cuthill-McKee ordering [Eisenstat et al, 1977].
- MD : Minimum degree ordering [Eisenstat et al, 1977].
- ND9 : Nested dissection for 9-point star [Eisenstat et al, 1977]

The solution using one of these orderings is indicated by prefixing the module name by RCM or MD (e.g. RCM + YALE SPARSE, MD + YALE SPARSE).

II. DISCUSSION OF THE PERFORMANCE DATA

The criterion of performance is the computer time $TIME_3$ required to solve the linear system. All PDEs were solved on a uniform, square N by N grid and the resulting linear system is of order $4(N-1)^2$. As expected, $\log(TIME_3)$ increases linearly with $\log N$ for all the problems so the slope of $TIME_3$ versus N (on a log-log scale) is taken as the primary measure of performance. The method with the smallest slope is the most efficient asymptotically as N increases. The computations were made on a VAX without floating point

acceleration, see [Rice, 1981a] for a discussion of the probable variations for another computing environment.

The 13 methods used are shown in Table 1. The slope is measured for each of the methods for a particular PDE. These slopes are ranked 1 to 13 (1 is the best) and the ranks are then averaged over the 13 PDEs. The average ranks are given along with sample run times for one PDE and a coarse 9x9 grid (256 equations) and a fine grid (3136 equations).

Table 2 gives the pairwise confidence levels for the rankings in Table 1. If the i, j entry in Table 2 is L , then the confidence level for the hypothesis that method i outperforms method j is greater than L . The data support the following hypothesis with confidence level exceeding 99%.

Hypothesis: The efficiencies of the 13 methods rank in groups is:

1. JACOBI CG, RCM + JACOBI CG, RCM + SOR, SOR
2. LINPACK, ND9 + YALE SPARSE
3. YALE ENVELOPE
4. MD + YALE SPARSE
5. RCM + YALE ENVELOPE, YALE SPARSE
6. SPARSE GE
7. RCM + YALE SPARSE
8. MD + YALE ENVELOPE

See [Hollander and Wolf, 1973] for the statistical procedure used. Table 2 distinguishes between some of the grouped methods with confidence levels less than 99%. The methods clearly group into iteration first, three elimination methods second (LINPACK, YALE ENVELOPE and ND9 + YALE SPARSE) and other sparse matrix methods last.

Performance rankings are asymptotic and do not hold for small systems. Thus there is interest in the cross-over points, points where the asymptotic rankings become valid. The cross-over points have been analyzed for the five most interesting methods: JACOBI CG for iteration,

Method	Average Rank	Median Slope	Sample run times for Problem 1-1	
			9x9 grid	29x29 grid
JACOBI CG	1.77	2.9	5.0	192
RCM + JACOBI CG	2.15	2.9	4.7	209
RCM + SOR	3.00	3.1	5.1	212
SOR	3.08	3.1	4.6	221
ND9 + YALE SPARSE	5.31	3.7	8.0	537
LINPACK	5.77	3.7	6.7	472
YALE ENVELOPE	7.00	3.9	4.4	457
MD + YALE SPARSE	7.92	4.0	9.1	800
RCM + YALE ENVELOPE	9.46	4.2	6.2	781
YALE SPARSE	9.62	4.2	9.5	1136
SPARCE GE	11.00	4.3	19.6	2411
RCM + YALE SPARSE	11.92	4.5	13.0	2016
MD + YALE ENVELOPE	13.00	5.8	20.2	BIG

Table 1. Average rankings of 13 methods for solving the Galerkin method equations. The basis of the ranks is the slope (on a log-log plot) of the time to solve the equations versus the grid size. The ranks are averaged over 13 problems, the median slope S is given for each method and they indicate that the asymptotic rate of time to solve the equations for an $N \times N$ grid is N^S . The times for indexing are not included, they are substantial for the minimum degree (MD) indexing. Sample times are given for two typical cases, a 9x9 grid gives a 256 equations, a 29x29 grid gives 3136 equations. The times are problem independent only for the LINPACK, YALE ENVELOPE and ND9 + YALE SPARSE methods.

1.	JACOBI CG	-												
2.	RCM + JACOBI CG	80	-											
3.	RCM + SOR	95	95	-										
4.	SOR	95	95	<80	-									
5.	ND9 + YALE SPARSE	99	99	99	99	-								
6.	LINPACK	99	99	99	99	80	-							
7.	YALE ENVELOPE	99	99	99	99	99	99	-						
8.	MD + YALE SPARSE	99	99	99	99	99	99	99	-					
9.	RCM + YALE ENVELOPE	99	99	99	99	99	99	99	99	-				
10.	YALE SPARSE	99	99	99	99	99	99	99	99	<80	-			
11.	SPARSE GE	99	99	99	99	99	99	99	99	99	99	-		
12.	RCM + YALE SPARSE	99	99	99	99	99	99	99	99	99	99	99	-	
13.	MD + YALE ENVELOPE	99	99	99	99	99	99	99	99	99	99	99	99	-
		1	2	3	4	5	6	7	8	9	10	11	12	13

Table 2. Pairwise confidence levels for rankings. If the i,j entry is L, then the confidence level for the hypothesis that method i is faster than method j is greater than L. There is modest uncertainty for the ranks among the iteration methods, for RCM + YALE ENVELOPE vs. YALE SPARSE and for ND9 + YALE SPARSE vs. LINPACK.

LINPACK and YALE ENVELOPE for elimination and YALE SPARSE with MD or ND9 for sparse matrix methods. Some details are given in Table 3, the principal observations are: (a) iteration methods cross-over for N between 7 and 17 (typically 400 equations) and become the most efficient, (b) LINPACK crosses YALE ENVELOPE at about 3000 equations, (c) the cross-over points of ND9 + YALE SPARSE with LINPACK and YALE ENVELOPE can only be estimated roughly by extrapolation. We estimate that ND9 + YALE SPARSE crosses YALE ENVELOPE at about 10,000 equations and crosses LINPACK at about 25,000 equations (if at all). All the cross-over points not given in Table 3 are for small system and are of little interest. Figure 1 shows a plot of TIME3 versus N for a typical PDE, 5-4. The data for ND9 + YALE SPARSE is omitted; it lies just above the LINPACK data and is almost exactly parallel to it.

One would expect the time of all the methods except iteration to be the same for a given size grid. In fact, only the LINPACK, YALE ENVELOPE and ND9+YALE SPARSE method times are PDE independent. The reason is that a few elements in the Galerkin equations which might be non-zero are actually zero for particular PDEs. These few elements make large changes in the minimal degree ordering for YALE SPARSE. For two PDEs (5-4 and 5-1), the original pattern of zeros and the resulting pattern resulting from the minimal degree ordering are shown in Figure 2 and 3.

One does not expect the Reverse Cuthill-McKee algorithm to make a significant reduction in the bandwidth of the Galerkin equations and hence it should not make a significant improvement in the efficiency of the elimination methods. In fact, it doubles the solution time because it increases the bandwidth substantially as seen in Figure 4 for PDE 5-4.

One possible source of observed efficiency of the iteration methods might be that they terminate prematurely. The iteration may safely terminate when the error in solving the linear system is smaller than the discretization error of the Galerkin method for the PDE. Comparing the

	Crossover Point			Comment
	Min	Median	Max	
LINPACK over YALE ENVELOPE	29	29	29	YALE ENVELOPE is more efficient for up to 3000 equations.
JACOBI CG over LINPACK	5	7	13	The LINPACK symmetric positive definite program is more efficient for up to about 250 equations.
JACOBI CG over YALE ENVELOPE	7	11	17	YALE ENVELOPE is more efficient for up to about 500 equations.
JACOBI CG over MD+YALE SPARSE	3	5	11	The minimum degree indexing with YALE SPARSE is more efficient for small problems, say < 150 equations.
JACOBI CG over ND9 + YALE SPARSE	3	5	8	The nested dissection indexing with YALE SPARSE is more efficient for small problems, say < 150 equations.

Table 3. Crossover points for representative, better methods of iteration (JACOBI CG), elimination (LINPACK and YALE ENVELOPE) and sparse matrix methods (YALE SPARSE with MD or ND9). The asymptotically faster methods ran as much as 10 times as long as the "slower" methods on the smallest grid, however, the total execution time was less than 1 second for such problems.

Figure 1. The solution time versus N for several methods: iteration (JACOBI CG), elimination (LINPACK and YALE ENVELOPE) and sparse matrix methods (MD + YALE SPARSE). The problem is $4u_{xx} + u_{yy} - 10u = f$.

The Galerkin Equations (Problem 5-4)

(a)

Figure 2. Pattern of non-zero elements in the Galerkin method equations for (a) the natural ordering and (b) the minimum degree ordering. The problem is $4u_{xx} + u_{yy} - 10u = f$ with a 7×7 grid.

Yale Min Degree matrix (Problem 5-4)

(b)

Figure 2 continued

The Galerkin Equations (Problem 5-1)

(a)

Figure 3. Pattern of non-zero elements in the Galerkin method equations for (a) the natural ordering and (b) the minimum degree ordering. The problem is $4u_{xx} + u_{yy} = f$ with a 7×7 grid.

Yale Min Degree matrix (Problem 5-1)

(b)

Figure 3 continued

Yale RCM matrix (Problem 5-1)

Figure 4. Pattern of non-zero elements in the Galerkin method equations for the reverse Cuthill-McKee (RCM) ordering. The problem is $4u_{xx} + u_{yy} = f$ with a 7×7 grid.

numerical solution of the PDE with the true solution does not always provide satisfactory information because the discretization error is beyond the round-off level of the machine used. The accuracy can be checked also by observing the behavior of the residual in the Galerkin method as it decreases more slowly than the discretization error. Figure 5 shows plots of the error and residual for several PDEs. Such examinations show that the iteration methods never terminate prematurely in the experiments reported here.

III. CONCLUSIONS AND COMPARISON WITH THEORY

The principal conclusion of this study is in establishing the hypothesis on the ranks stated earlier. Other, less extensive, data, not analyzed here, from other computing environments corroborates these results. Thus there is very strong evidence that iteration methods are the most effective for solving the Galerkin equations. There is still some uncertainty as to which of the iteration methods is best.

The LINPACK, YALE ENVELOPE and ND9+YALE SPARSE programs are close competitors. The statistical significance of the differences in their performance is illusory since all are constant. The small uncertainty in the timing mechanism produces the statistical uncertainty between LINPACK and ND9+YALE SPARSE. The observed performances are close enough that the ranking could well reverse in another computing environment.

The time for carrying out the minimum degree ordering algorithm has not been considered in this study because it does not affect the overall conclusion. This time is substantial (110-160 seconds for the 29 by 29 grid or about 15% of the solution time TIME3).

Other conclusions are: (a) the automatic termination tests of the iteration methods are effective for the Galerkin equations, (b) small differences in the zero pattern of the matrix cause large differences in the minimal degree ordering and lead to substantial (about 30%) variations in TIME3 about its average, (c) the reverse Cuthill-McKee ordering is

Figure 5. The least squares residual (solid) and maximum error (open) versus N for the Galerkin method applied to problems 1-1(circles,left), 6-1(stars,left), 4-1(circles,right), and 44-1(stars,right).

ineffective even though relatively inexpensive (less than 1% of TIME3 for five grids).

These results are only partly consistent with theoretical results on the behavior of the underlying algorithms. For the most interesting methods, iteration, the only theoretical results are that the methods converge. The closest theoretical analysis on the rates of convergence [Parter and Boley, 1979] do not directly apply here, but are compatible in that the fast convergence is predicted. The slope for the iteration methods is nearly 3, this means the work is the order of N^3 which is the same rate as has been established for SOR (with optimum relaxation parameter) applied to the finite difference method equations in model problems. Thus the efficiency observed in these experiments is the best one can reasonably expect.

The LINPACK and YALE ENVELOPE results are as one expects. The inherent advantage of the envelope method is small because the matrix is so close to a band matrix. Thus the higher efficiency of the LINPACK implementation of elimination compensates for this advantage.

Theory suggests that sparse matrix elimination methods should outperform the LINPACK and YALE ENVELOPE elimination methods. We note that MD + YALE ENVELOPE is expected to do badly, the work should be about order N^6 which is as observed. The SPARSE GE program is not expected to be competitive here because (a) it cannot take advantage of the symmetry of these problems and (b) it does pivoting which is not needed for these problems. We list four possible reasons for the discrepancy between the theory for sparse matrix methods and the observed results, they are listed in the order of most to least likely (in our opinion).

1. Certain types of computer work needed for the methods has not been included in the theory (e.g. manipulation of indexes and pointers).

2. These programs do not implement the underlying algorithms properly or as efficiently as they should.
3. The asymptotic results require a value of N much larger than used in this study. If so, this would still eliminate these methods from consideration for most computations.
4. Some idiosyncrasy in the computing environment affects the sparse matrix methods in an adverse way. There is a small amount of evidence in [Rice, 1981a] that this environment favors sparse matrix methods.

REFERENCES

1. J.J. Dongarra, J.R. Bunch, C.B. Moler and C.W. Stewart [1979], LINPACK User's Guide, Soc. Indust. Appl. Math., Philadelphia.
2. Eisenstat, S.C., M.C. Gursky, M.H. Schultz and A.H. Sherman, Yale Sparse Matrix Package: I-The Symmetric Codes, Rpt. 112, II-The Non Symmetric Codes, Rpt. 114, Computer Science, Yale University [1977].
3. M. Hollander and D.A. Wolfe [1973], Non-parametric Statistical Methods, John Wiley, New York, Chapter 7.
4. D. Kincaid, R.G. Grimes and D.M. Young [1979], The use of iterative methods for solving large sparse PDE-related linear systems, in Advances in Computer Methods for Partial Differential Equations, III, (R. Vichnevetsky and R. Stepleman, eds.), IMACS, New Brunswick, N.J. pp. 29-32.
5. S.V. Parter and D.L. Boley, Block relaxation techniques for finite-element elliptic equations: An example. LA-7870-MS, Los Alamos Scientific Lab., (1979), 17 pages.
6. J.R. Rice [1981], ELLPACK: Progress and Plans, in Elliptic Problem Solvers (M. Schultz, ed.), Academic Press, pp.
7. J.R. Rice [1981a], Machine and compiler effects on the performance of elliptic PDE software, CSD-TR 359, Computer Science Dept., Purdue University, 9 pages.
8. J.R. Rice, E.N. Houstis and W.R. Dyksen [1981], A population of linear, second order, elliptic partial differential equations on rectangular domains Mathematics of Computation, 35 (1981) pp. 475-484.

APPENDIX: THE PDEs

The partial differential equation problems used for this study are listed below. The domain for each problem is the unit square $0 \leq x, y \leq 1$ and the boundary conditions are all homogeneous i.e. $U(x, y) = 0$ on the boundary. In each case the forcing term $f(x, y)$ is determined to produce a particular true situation.

$$1-1 \quad (e^{xy} u_x)_x + (e^{-xy} u_y)_y - u/(1+x+y) = f$$

$$4-1 \quad u_{xx} + u_{yy} = f$$

$$5-1 \quad 4u_{xx} + u_{yy} = f$$

$$5-4 \quad 4u_{xx} + u_{yy} - 10u = f$$

$$6-1 \quad u_{xx} + u_{yy} - (100 + \cos(3\pi x) + \sin(2\pi y))u = f$$

$$7-1 \quad u_{xx} + u_{yy} = 1$$

$$10-2 \quad u_{xx} + u_{yy} = f$$

$$10-3 \quad u_{xx} + u_{yy} = f$$

$$28-3 \quad (wu_x)_x + (wu_y)_y = 1 \text{ where } w = 100 \text{ for } 0 \leq x, y \leq .5 \\ = 1 \text{ otherwise}$$

$$41-1 \quad u_{xx} + u_{yy} + 10u = f$$

$$44-1 \quad u_{xx} + u_{yy} + wu = w, \quad w = -2.030625 e^{(r/(1+r/2))}, \\ r(x, y) \text{ tabulated}$$

$$44-2 \quad u_{xx} + u_{yy} + wu = w, \quad w = -100 e^{(r/(1+r/2))}, \\ r(x, y) \text{ tabulated}$$

$$44-3 \quad u_{xx} + u_{yy} + wu = w, \quad w = -2.030625(1-r)e^{(r/(1+r/25))}, \\ r(x, y) \text{ tabulated}$$