1978

A System for Performance Evaluation of Partial Differential Equations Software

R. F. Boisvert

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

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

Report Number:
78-278

A SYSTEM FOR PERFORMANCE EVALUATION OF PARTIAL DIFFERENTIAL EQUATIONS SOFTWARE

R. F. Boisvert, E. N. Houstis, and J. R. Rice

Mathematical Sciences

Purdue University

CSD-TR 278

July 28, 1978

ABSTRACT

This paper describes a system to systematically compare the performance of various methods (software modules) for the numerical solution of partial differential equations. We discuss the general nature and large size of this performance evaluation problem and the data one obtains. The system meets certain design objectives that ensure a valid experiment: (1) precise definition of a particular measurement; (2) uniformity in definition of variables entering the experiment; (3) reproducibility of results. The ease of use of the system makes it possible to make the large sets of measurements necessary to obtain confidence in the results and its portability allows others to check or extend the measurements. The system has four parts: (1) semi-automatic generation of problems for experimental input; (2) the ELLPACK system for actually solving the equation; (3) a data management system to organize and access the experimental data; and (4) data analysis programs to extract graphical and statistical summaries from the data.

A System for Performance Evaluation of Partial Differential Equations Software

I. The Performance Evaluation Problem

This paper concerns partial differential equations (PDEs) and certain terminology from that area inevitably enters. However, we attempt to describe things so that the reader need not be familiar with PDE methods or exactly understand the meaning of the terms used. In this section we outline the general problem of evaluating software for solving PDEs.

A PDE problem consists of three parts:

Operator -- Domain -- Boundary Conditions

which are illustrated by the classical Poisson problem.

Operator: $$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 1$$

Domain: Unit Square $0 \leq x,y \leq 1$

Boundary Conditions: $$u(x,0) = 0, \quad u(x,1) = 0$$
$$u(0,y) = \sin(\pi y), \quad u(1,y) = 0$$

This example is as simple as possible and most PDE problems involve somewhat more complicated functions and domains.

There is a large variety of numerical methods for these problems, but those of current interest seem to be one of two kinds:

A. One Stage Methods. The method has little or no modularity in terms of common, high level mathematical procedures. The Fast Fourier Transform methods are of this type.

B. Four Stage Methods. The four stages are:

1. Discretize the domain. One puts a rectangular grid over it, cuts it up into triangles, etc.

2. Approximate the operator. One obtains an ordinary system of equations for a large number of unknowns by applying an approximation such as finite differences or finite elements of various kinds.

3. Reorder these equations. The original or natural order of these equations might not be suitable or as efficient for certain equation solution methods.

4. Solve these equations. Various kinds of direct methods (Gauss elimination) or iterative methods (overrelaxation, Newton's method) may be used to solve the equations for the approximation to the solution of the PDE.

The modularity of the four stage methods allows one to generate an enormous number of distinct numerical methods for PDEs, literally well into the thousands, even millions.

PDE problems absorb a substantial fraction of the scientific computing power, and it is very desirable to know which of these thousands of methods perform best. The goal of the system described here is to provide a partial answer to this question. We first indicate certain essential topics that are not discussed here:

1. The evaluation methodology. We work within the general framework outlined by [Rice, 1976]. A more specific discussion for mathematical

software is presented in [Rice, 1979] and PDE evaluation is explicitly discussed in [Houstis, Lynch, Papatheodorou and Rice, 1975 and 1978], [Houstis and Papatheodorou, 1977 and 1979], and [Houstis and Rice, 1979].

2. The PDE problem population. A large population of linear problems is given by [Houstis and Rice, 1979] along with a discussion of various characteristics and features of the population.

3. Numerical methods. The possible methods are described in many texts and their incorporation into the ELLPACK system (which is part of the system described here) is discussed in [Rice, 1977].

Thus, we assume that we know what we want to do: namely, solve each of a large number of PDEs by a large number of numerical methods. The question is how to get it done. The essential difficulty here is in the size and complexity of the problems, the software and the resulting data. Earlier attempts, including ours, involved a new program for each method and each PDE. These programs typically ran from 500 to 2000 lines of Fortran. Even though one can change the PDE, keeping the same method, without a complete rewrite, it is clear that this approach allows large scale experimentation only by an enormous investment in software. Furthermore, it was noted that these rewrites would affect the method at times, and it was not always clear that the last PDE was solved by the same method as the first.

Once this initial problem is solved, one then discovers two other substantial, but less formidable, difficulties. First, it is onerous to specify a hundred or two PDE problems and methods as required for a typical modest experiment. Second, the performance of a method for a PDE is not easily summarized in one number, and thus one obtains a large amount of data.

Further, this data should be accumulated as more and more experiments are made so that various performance criteria or population characteristics can be studied without redoing the measurements.

II. System Design Objectives

The system described here facilitates a large scale performance evaluation. The system incorporates the following specialization of the PDE problems and methods:

A. The operators are linear and elliptic.

B. The domains are two or three dimensional.

C. The domain discretization is with rectangular grids, not necessarily uniform.

Within this context the design objectives of the system are:

A. Provide a high level and precise definition for each measurement made.

B. Preserve uniformity over time for the PDE problems, methods and measurement procedures.

C. Make it practical to design and implement large scale experiments and to collect the performance data.

D. Allow reproducibility of results by others. This implies a high degree of portability in the system.

E. Provide an open ended experimental framework for the problem population and methods to be evaluated. This implies that others must be able to incorporate their own problems and methods within the experimental framework.

F. Accumulate the experimental results systematically as the number of experiments and sizes of the problem and method populations grow.

III. System Design and Operation

An overall view of the system is shown in Figure 1. There are four basic parts:

A. The experimental input system

B. The ELLPACK system

C. The experimental data management system

D. Data analysis programs

The largest part is the ELLPACK system; we describe it only briefly here since it has been discussed elsewhere [Rice, 1977]. A simple example of an ELLPACK run is shown in Figure 2; the upper part is the user input and the lower part is some of the printed output produced.

The ELLPACK system consists principly of a Fortran preprocessor (about 6000 statements) and a library of modules for implementing numerical methods and producing output. The library currently contains about 25,000 lines of code in subprograms as follows:

		lines of code
1	domain processor	1800
10	operator approximation modules	9000
4	equation reordering modules	500
15	equation solution modules	12000
8	output and auxilary programs	1800

The ELLPACK system is still growing, and it is anticipated that the number of methods (or modules for constructing methods) will increase substantially.

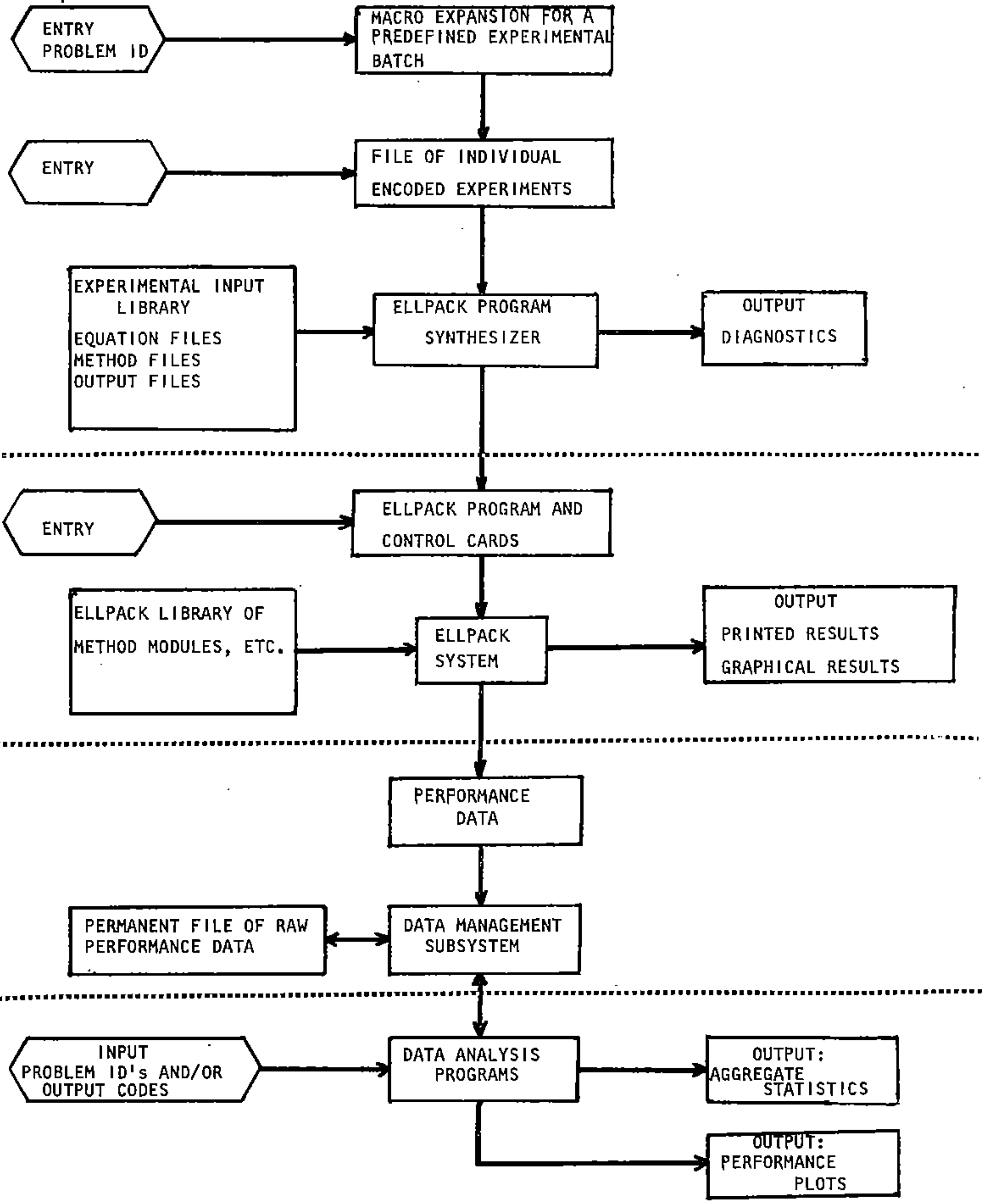
ENTRY
PROBLEM ID
MACRO EXPANSION FOR A PREDEFINED EXPERIMENTAL BATCH
ENTRY
FILE OF INDIVIDUAL ENCODED EXPERIMENTS
EXPERIMENTAL INPUT LIBRARY
EQUATION FILES
METHOD FILES
OUTPUT FILES
ELLPACK PROGRAM SYNTHESIZER
OUTPUT DIAGNOSTICS
ENTRY
ELLPACK PROGRAM AND CONTROL CARDS
ELLPACK LIBRARY OF METHOD MODULES, ETC.
ELLPACK SYSTEM
OUTPUT
PRINTED RESULTS
GRAPHICAL RESULTS
PERFORMANCE DATA
PERMANENT FILE OF RAW PERFORMANCE DATA
DATA MANAGEMENT SUBSYSTEM
INPUT
PROBLEM ID's AND/OR OUTPUT CODES
DATA ANALYSIS PROGRAMS
OUTPUT: AGGREGATE STATISTICS
OUTPUT: PERFORMANCE PLOTS

Figure 1. Block diagram of the system for performance evaluation of PDE software.

```
 *
 *   -------------------------
 *   PDE METHOD COMPARISON TEST
 *   -------------------------
 *
 *     ENCODED PROGRAM IS
 *          3//4,4/2/10/16//$TEST-P3-C1-COLLOCATION/
 *
 *   PROBLEM    3
 *   PROBLEM TYPE =      000.02    004.00    000.00    006.00
 *                       2020221002002
EQUATION.
          POISSON   $   CONSTANT COEFFICIENTS   $   TWO DIMENSIONS
          UXX$ + UYY$  =  6.*X*Y*EXP(X+Y)*(X*Y+X+Y-3.)
BOUNDARY.
          HOMOGENEOUS   $   DIRICHLET
          X=0.,U=0.
          X=1.,U=0.
          Y=0.,U=0.
          Y=1.,U=0.
FORTRAN.
      FUNCTION TRUE(X,Y)
      TRUE = 3.*EXP(X+Y)*X*Y*(1.-X)*(1.-Y)
      RETURN
      END
OPTIONS.  TIME   $   MEMORY   $   LEVEL=0
GRID.     UNIFORM X =  4   $   UNIFORM Y =  4
*
DISCRETIZATION.
.               P3-C1 COLLOCATION
INDEXING.
.               COLLOCATE BAND
SOLUTION.
.               BAND SOLVE
*
OUTPUT.    MAX-U   $   MAX(20,20)-ERROR
*
SEQUENCE.
               DISCRETIZATION
               INDEXING
               SOLUTION
               OUTPUT
               TEST
 *
 *----------------------------------------------------------------------
 *
 *   ELLPACK PROGRAM GENERATION SUMMARY
 *
 *           PROBLEM RECORD          =   3        (TYPE=202022100200222)
 *           OPTIONS RECORD          =   0
 *           DISCRETIZATION MODULE =   2        (TYPE=201111011111111)
 *           INDEXING MODULE         =  10
 *           SOLUTION MODULE         =  16
 *           OUTPUT RECORD           .   0
 *
 *       PROBLEM/DIS MODULE ARE COMPATIBLE
 *
 *       MODULE SEQUENCE IS LEGAL
 *
 *----------------------------------------------------------------------
END.
                         MEMORY REQUIREMENTS ARE ABOUT
        WORKSPACE         =     4     GRID LINES        =      9
        LIN EQ COEFS      =  1088     LIN EQ ID-S       =   1088
        INDEXES OF VARS   =   192     AMATRX,BVECTR     =   3136
        PROGRAM + MISC    =  2764     TOTAL MEMORY      =   8281
          - - - - - ELLPACK PREPROCESSOR TIME =    .57 SECONDS
```

Figure 2 (Part 1). Example of ELLPACK input.

```
-------------------
ELLPACK 77 OUTPUT
-------------------

         ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
         +                                                                                  +
         +  MAX ABSOLUTE VALUE OF SOLUTION ON   4 BY   4 BY   1 GRID IS    5.61576079E-01  +
         +                                                                                  +
         ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


-------------------
ELLPACK 77 OUTPUT
-------------------

         ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
         +                                                                                  +
         +  MAX ABSOLUTE VALUE OF ERROR    ON  20 BY  20 BY   1 GRID IS    1.64790284E-03  +
         +                                                                                  +
         ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


 EXECUTION TIME FOR MODULES  (SEC.)
 ------------------------

    1 TIME =    .10          2 TIME =    .03          3 TIME =    .18          4 TIME =   1.59

    5 TIME =    .10          6 TIME =   2.01



   3  07/26/78 11.21.25.  $TEST-P3-C1-COLLOCATION
202022100200222      000.02 004.00 000.00 006.00
2/10/16/
    4   4   3.33E-01    36   4.49E-04 1.65E-03 3.93E-05 3.19E-01 8.21E-02 6.88E-02 5.62E-01   0  8281    .31    .10    .03    .18
```

Figure 2 (Part 2). Output from an ELLPACK run. The data at the bottom is saved in a permanent file.

The experimental input system is built around a set of files containing pieces of ELLPACK programs. Specifically, we have

EQNFIL: file of PDE problems. The records in this file contain the PDE operator, domain definitions, boundary conditions and any supporting Fortran code to define various functions.

MACFIL: more than half of the PDE problems are parameterized and EQNFIL in this case only contains a set of parameters and specifies a record of MACFIL where these parameters are to be substituted.

Figure 3 contains a sample from both these files illustrating their nature. The current EQNFIL has about 100 PDE problems and it is open ended.

OUTFIL: file of ELLPACK system output commands

OPTFIL: file of ELLPACK system option commands

GRDFIL: file of non-uniform rectangular grid definitions. Uniform grids are defined in the input by a simple pair of integers.

The ELLPACK program synthesizer takes input of the form:

```
3//4,4/2/10/16//$TEST-P3-C1-COLLOCATION/
3//5,5/2/10/16//
3//6,6/2/10/16//
```

This is a group of three runs which differ only in the grids used (4x4, 5x5, and 6x6). The numbers between the slashes are either parameters (such as 4x4 for the grid) or record numbers from one of the files. Specifically, the first line of this group says to use:

record 3 from EQNFIL

default record from OPTFIL

4x4 grid (and nothing from GRDFIL)

operator approximation #2 (that is P3C1-COLLOCATION)

equation ordering module #10 (form a band matrix)

equation solution module #16 (use profile Gauss elimination)

default record from OUTFIL

comment for the group

The results of this group of experiments are automatically saved in the permanent file of performance data as a group; the common information about the three runs is not repeated. This is all done with portable Fortran programs.

The ELLPACK program synthesizer does considerable checking of the compatibility of PDE problems and numerical methods. Fifteen characteristics are defined (e.g., two dimensional, constant coefficients, homogeneous boundary conditions, uniform grid) for each PDE problem. Then each method has a corresponding vector for checking. The meanings of the values in these lists are as follows:

value	EQNFIL meaning	Method meaning
0	item not present	item must not be present
1	always matches	always matches
2	item present	item must be present

The output from the program synthesizer is a standard ELLPACK program as in Figure 2. A special Fortran subroutine is included which, at the end of the ELLPACK run, makes several measurements not automatically in the ELLPACK system (for example, computing the least squares residual) and places the resulting data in the permanent file. The ELLPACK system has a command to call a subroutine named TEST at the end of its normal execution.

Typically, one wants to test several methods on a subpopulation of problems so there is a mechanism to construct a predefined batch of experiments. This is implemented as an operating system control card procedure (strictly non-portable). Once a batch is defined, usually with 20 to 30 runs, one just enters the record number from EQNFIL and this macroprocessor generates all the input for the ELLPACK program synthesizer.

If an invalid combination is met, various diagnostics are issued but the experimental batch is not aborted. This obviously allows one to avoid nonsense experiments such as attempting two dimensional finite differences on a three dimensional problem. It also allows large batches to be run which contain a few incompatible cases and these cases are automatically skipped.

The experimental data generated is automatically put into a permanent data base by the experimental data system. This is implemented by a simple data management system designed specifically for our purposes which is based on a locally written random access disk file manager. The data resides on two files, one containing information about each problem, including which PDE methods have been tested on each problem and the second containing tabular information on the outcome of each test (see Figure 4). The random access keys are problem numbers.

```
RECORD 3
*         000.02    004.00    000.00    006.00
*         2020221002002
1         POISSON   $   CONSTANT COEFFICIENTS   $   TWO DIMENSIONS
1         UXX$ + UYY$  =  6.*X*Y*EXP(X+Y)*(X*Y+X+Y-3.)
2         HOMOGENEOUS   $   DIRICHLET
2         X=0.,U=0.
2         X=1.,U=0.
2         Y=0.,U=0.
2         Y=1.,U=0.
3      FUNCTION TRUE(X,Y)
3      TRUE = 3.*EXP(X+Y)*X*Y*(1.-X)*(1.-Y)
3      RETURN
3      END
-----------------------------------------
RECORD 4
*         000.44    090.90    000.00    080.25
*         2000002002002
1         SELF - ADJOINT $ TWO DIMENSIONS
1         COEF1(X,Y)UXX $ + UYY = F(X,Y)
2         DIRICHLET
2         X =0. ,U = TRUE(0.,Y)
2         X =1. ,U = TRUE(1.,Y)
2         Y =0. ,U = TRUE(X,0.)
2         Y =1. ,U = TRUE(X,1.)
3      FUNCTION COEF1(X,Y)
3      COEF1 = 2.
3      IF( X.LT..4 ) COEF1 =1.

3      FUNCTION TRUE(X,Y)
3      FX2 = .7 + .5*(X-.4) + (X-.4)**2/(1+X*X)
3      TRUE =AMIN1(X+.3,FX2)*(1.+(Y-1.)**2*EXP(-Y))
3      RETURN
3      END
-----------------------------------------------------
RECORD 5
*         000.38    090.60    000.00    070.40
EXPAND 1/1.5/
-----------------------------------------
RECORD 6
*         000.31    080.50    000.00    060.20
EXPAND 1/2.5/
```

```
-----------------------------------------
MACRO 2
*         2000221002002
1         CONSTANT COEFFICIENTS   $   TWO DIMENSIONS
1         4. UXX$ + UYY$ -^A U  =  F(X,Y)
2         HOMOGENEOUS   $   DIRICHLET
2         X=0.,U=0.
2         X=1.,U=0.
2         Y=0.,U=0.
2         Y=1.,U=0.
3      FUNCTION TRUE(X,Y)
3      TRUE = 2.*X*(X-1.)*(COS(6.28318530717958*Y)-1.)
3      RETURN
3      END
3      FUNCTION F(X,Y)
3      DATA TWOPI/6.28318530717958/
3      CTPY = COS(TWOPI*Y)
3      F = 2.*X*(X-1.)*(^A*(1.-CTPY)-TWOPI*TWOPI*CTPY)
3     $        +16.*(CTPY-1.)
3      RETURN
3      END
```

Figure 3. A segment of EQNFIL (top) and MACFIL (bottom). The symbols ^A are replaced by values of parameter in MACFIL.

Table 1. Definition of the data elements that are placed in the permanent data base.

```
   3  07/26/78 12.05.48.  $TEST-4TH-ORDER-HODIE
202022100200222     000.02 004.00 000.00 006.00
5,IORDER=41/9/16/
    4   4   3.33E-01      4   3.51E-03 9.05E-02 3.19E-04 5.93E+00 4.14E+00 1.13E+00 5.66E-01

                                                                  0  6862    .12    .11    .01    .00


        LINE 1   = EQNFIL RECORD NUMBER, DATE, TIME, COMMENT
        LINE 2   = PROBLEM/METHOD COMPATABILITY INFORMATION
                   AND PROBLEM CHARACTERISTICS
        LINE 3   = ENCODED METHOD DESCRIPTION
        LINE 4,..= TABLE OF EXPERIMENTAL RESULTS, EACH LINE CONTAINING
                    1. NUMBER OF GRID LINES IN X DIRECTION
                    2. NUMBER OF GRID LINES IN Y DIRECTION
                    3. MAXIMUM GRID SPACING IN ANY DIRECTION
                    4. NUMBER OF UNKNOWNS
                    5. MAXIMUM ERROR AT NODES
                    6. MAXIMUM ERROR ON A FIXED GRID
                    7. DISCRETE L2 ERROR AT NODES
                    8. MAXIMUM RESIDUAL AT MIDPOINTS OF SUBRECTANGLES
                    9. MAXIMUM RELATIVE RESIDUAL AT MIDPOINTS
                   10. DISCRETE L2 RESIDUAL AT MIDPOINTS
                   11. MAXIMUM OF COMPUTED SOLUTION AT NODES
                   12. NUMBER OF ITERATIONS USED BY ITERATIVE METHOD
                   13. MEMORY USED
                   14. TOTAL TIME (SEC.)
                   15. LIST OF MODULE TIMES
```

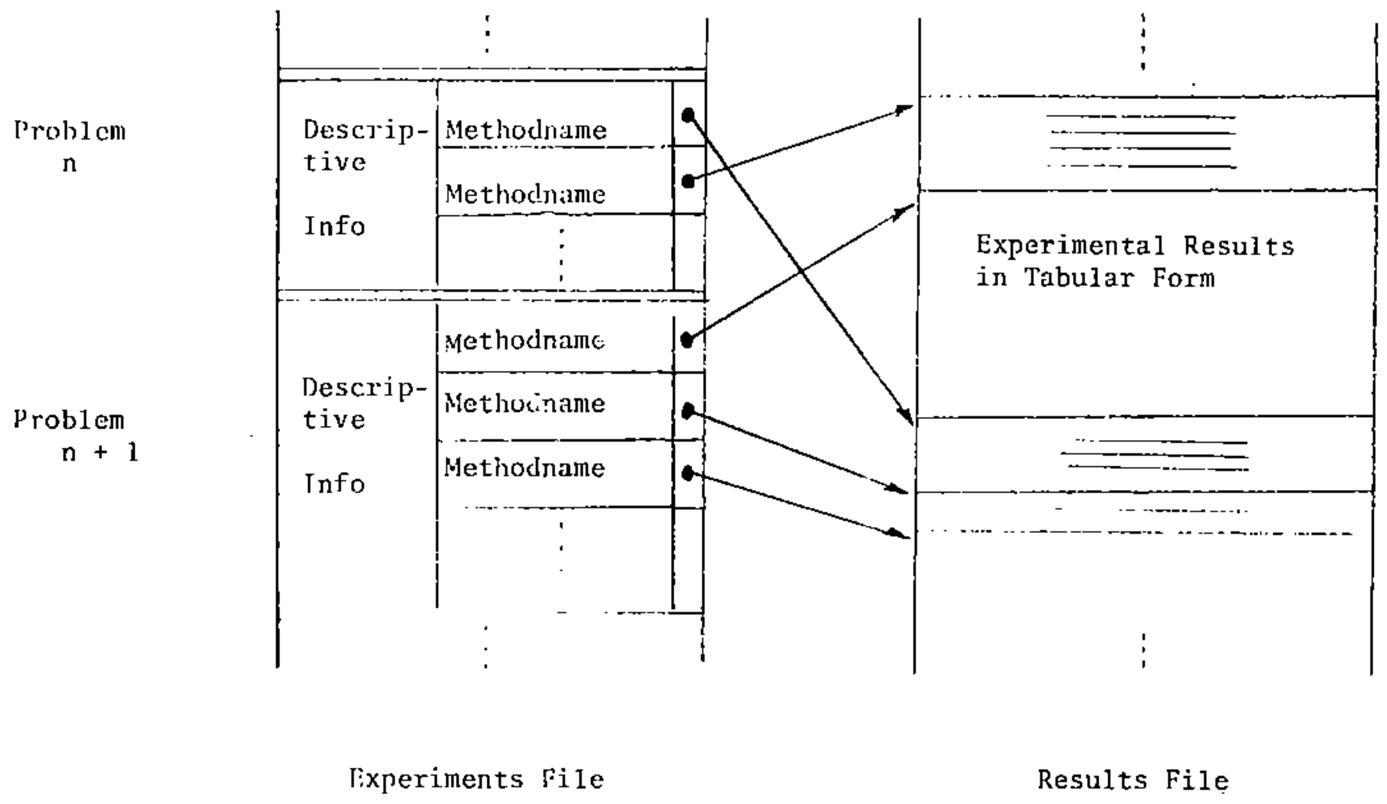

Figure 4. Logical organization of permanent data base

A single ELLPACK run generates two types of data: identification of the run made and the measured data as shown in the top of Figure 5. The nature of the PDE problem is to make several domain discretizations so as to investigate the convergence behavior of a method. To avoid some redundancy, the experimental data system automatically "collects" together the data for a particular PDE problem. Thus the data can be retrieved in the form shown in the bottom of Figure 5.

The precise definition of the data elements from an ELLPACK run is given in Table 1. Briefly, this data includes

(a) identification of the PDE problem, including parameters

(b) values of logical variables (switches) associated with the run, e.g., Dirichlet = .TRUE., self-adjoint = .FALSE.

(c) description of the rectangular grid

(d) measured values of the error and/or residual, execution times (total and of various modules), memory usage and method parameters (matrix size, number of iterations)

If one takes 100 PDE's and applies 25 methods with 6 different domain specifications, one sees that 15,000 partial differential equations are to be solved. That is a very substantial computation and various strategies are used to reduce the number of measurements. Nevertheless, the experimental data promises to become quite voluminous so that manual manipulation and analysis of the data would become very burdensome.

The final portion of the system is a set of three programs to display and analyze the experimental data. The first of these produces

```
  3  07/26/78 12.09.00.  $TEST-P3-C1-COLLOCATION
202022100200222      000.02 004.00 000.00 006.00
2/10/16/
   4   4   3.33E-01    36   4.49E-04 1.65E-03 3.93E-05 3.19E-01 8.21E-02 6.88E-02 5.62E-01   0  8281    .30    .10    .03    .18
   5   5   2.50E-01    64   1.36E-04 5.32E-04 1.10E-05 1.77E-01 5.98E-02 2.81E-02 5.10E-01   0 12216    .67    .17    .05    .44
```

```
REQUEST = PRINT PROB     ELLPACK     INFORMATION     RETRIEVAL     SYSTEM     00.37.31.     07/26/78.     PAGE   1
*****************************************************************************************************************


RECORD =     79
PROBLEM =     39    (A=0.5,B=3.0,C=10.0              )
PROBLEM INFO =     200020000200022          000.30     040.35     000.00     020.20


      METHOD = 5,IORDER=4/9/16/

-----------------------------------------------------------------------------------------------------------------------
 DATE     TIME   NX  NY   HMAX    NRUNK  ERRMAX  ERRMXF  ERRL2    RESMAX  RESMXR  RESL2    SOLMAX NIT   MEM  TOTLT TIME1 TIME2 TIME3
-----------------------------------------------------------------------------------------------------------------------

04/21/78 10.29   3    3 5.00E-01     1 5.2E-02        0 5.8E-03 3.3E+01 1.3E+00 8.9E+00 1.4E+01   0  6762    .15   .15     0     0
04/21/78 10.29   5    5 2.50E-01     9 1.4E-02        0 7.1E-04 2.2E+01 3.4E+00 2.5E+00 1.4E+01   0  7042   1.22  1.20   .01   .01
04/21/78 10.29   7    7 1.67E-01    25 6.8E-03        0 2.3E-04 1.5E+01 6.3E+00 1.0E+00 1.4E+01   0  7690   3.49  3.39   .05   .04
04/21/78 10.29   9    9 1.25E-01    49 4.6E-03        0 1.2E-04 1.1E+01 6.8E+00 6.9E-01 1.4E+01   0  8802   6.72  6.48   .10   .14
04/21/78 10.29  11   11 1.00E-01    81 3.4E-03        0 6.5E-05 9.5E+00 1.4E+00 4.1E-01 1.4E+01   0 10474  11.35 10.84   .18   .33


      METHOD = 1/9/16/

-----------------------------------------------------------------------------------------------------------------------
 DATE     TIME   NX  NY   HMAX    NRUNK  ERRMAX  ERRMXF  ERRL2    RESMAX  RESMXR  RESL2    SOLMAX NIT   MEM  TOTLT TIME1 TIME2 TIME3
-----------------------------------------------------------------------------------------------------------------------

04/21/78 10.40   5    5 2.50E-01     9 8.7E-03        0 6.0E-04 2.2E+01 3.5E+00 2.5E+00 1.4E+01   0  3259    .07   .06   .01   .01
04/21/78 10.40   9    9 1.25E-01    49 2.0E-03        0 1.0E-04 1.1E+01 8.4E+00 6.9E-01 1.4E+01   0  5010    .46   .28   .06   .12
04/21/78 10.40  13   13 8.33E-02   121 8.9E-04        0 3.2E-05 8.0E+00 5.7E+00 2.9E-01 1.4E+01   0  8794   1.42   .67   .16   .59
04/21/78 10.40  17   17 6.25E-02   225 5.3E-04        0 1.4E-05 6.0E+00 9.7E+00 1.9E-01 1.4E+01   0 15362   3.41  1.23   .33  1.86
```

Figure 5. (Top) The permanent experimental data recorded from a group of ELLPACK runs (see Table 1 for definitions).
(Bottom) A sample of the data as retrieved from the data base.

RECORD 34

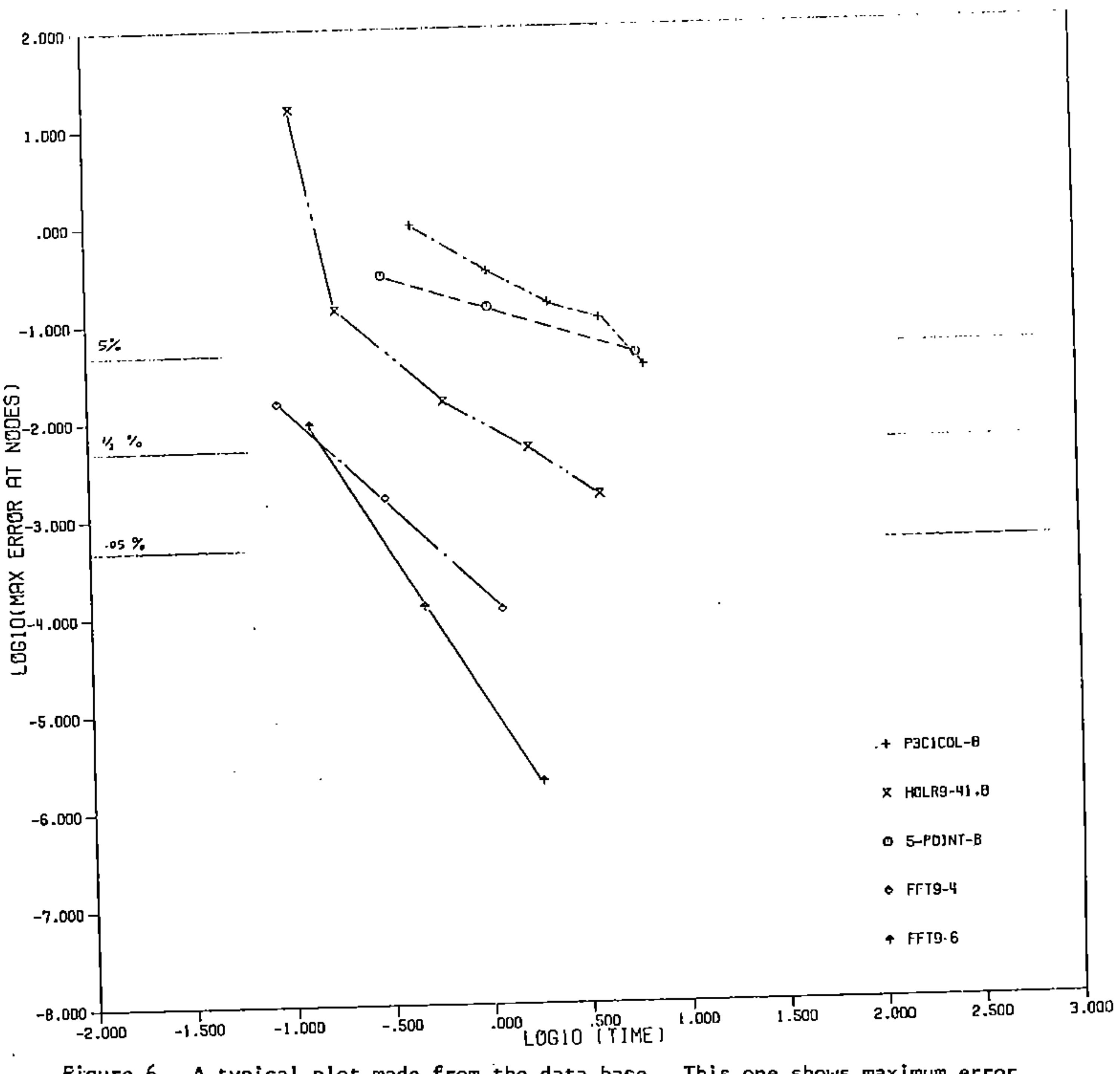

Figure 6. A typical plot made from the data base. This one shows maximum error versus execution time for a number of methods identified crytically in the lower right corner.

graphical output of two selected variables from an experiment (e.g., for PDE #6 and method #8 plot maximum error versus mesh size). A sample is shown in Figure 6, and it is seen that a number of methods may be plotted for one PDE on the same graph. The plot shown is by a quick but crude routine; if publication quality plots are desired, then a different graphical routine is used.

The second program produces aggregate statistics of various kinds. The data items desired are indicated by input of the form:

```
      RECORDS:  5, 7, 21, 3, 18
      METHODS:
                2/6/4
                4, IORDER = 4/3/16
   STATISTICS:
                5 PERCENT TIME
                .5 PERCENT MEMORY
                CONVERGENCE RATE
        PLOTS:
                H VS. MAX-ERROR
                TIME VS. MAX-RESIDUAL
```

The number after RECORDS are simply PDE problems. The items after METHODS are indices of numerical method modules that collectively define a method. Note that some methods are parameterized as indicated in the second item here. The STATISTICS items are derived from the raw data. Thus "5 PERCENT TIME" means the execution time required to achieve a maximum error of 5%. This time is obtained by interpolation as it is very unlikely that any grid achieved exactly a 5% error. A number of basic statistics (mean, median, range, standard deviation, etc.) are produced for each item named in STATISTICS. The items after PLOTS are just pairs of names of raw data measurements to be plotted. A standard scale is associated with each item (e.g., errors, times, and residuals are on a log scale, memory is on a linear scale).

The third data analysis program does not operate directly with the raw experimental data. As usual, in performance evaluation, there are multiple criteria of performance that are not easily compared. The goal is, nevertheless, to rank the numerical methods for various subpopulations, and we proceed as follows. First we examine the output of the preceding programs and make a subjective ranking of how each method under consideration performs on each problem. This gives rankings to the methods (ties are allowed if one cannot decide between two methods). We then apply the Friedman, Kendall and Babington-Smith test to determine an overall ranking [Hollander and Wolfe, 1973]. While this approach does not eliminate subjectivity from the determination of the "best" method; it does isolate it to a rather specific, limited context and eliminates the common conclusion of "we have examined all the data and conclude that method x is best." If a group of methods are of approximately equal rank over the subpopulation of problems, this will show up in the Friedman, Kendall and Babington-Smith test.

IV. Conclusions

We find that this approach has substantially increased the efficiency of the performance evaluation and the quality of the results. The , straightforward approach used earlier by [Houstis, Lynch, Papatheodorou and Rice, 1977] was so cumbersome that we felt it unreasonable to use it to make the quantity of measurements needed for firm (statistically speaking) conclusions. There is a very substantial capital investment involved in our system, which we estimate in Table 2. The effort for the numerical methods module is not considered as it is not part of the investment required for the system. It would be needed no matter how the performance evaluation were done.

Table 2. Components of the performance evaluation system with their sizes and estimated development efforts

Component	Program Statements	Lines	Estimated Man-months
Experimental Input Programs	600	1400	1/2
Experimental Input Library	---	2000	1/4
Subtotal Input	600	3400	1
ELLPACK Preprocessor	3000	6500	5-6
Domain Processor	1000	1800	3
Output Modules	700	1700	*4
Subtotal ELLPACK	4700	10000	13
Data Management	700	1300	*1/2
Data Analysis Programs	600	1000	2
System Integration			3-4
Total	6000	15200	20

*This part of the system makes substantial use of existing software for things like contour plotting and data base management.

With this system now operational we find that one person can define a new experiment (e.g. how well do higher order numerical methods perform on problems with singularities), generate the data and analyze it within a week or two. This is at least an order of magnitude less effort than required previously. There is a substantial computer cost involved in such an experiment; partial differential equations take some time to solve and the preprocessing input and a posteriori error analysis takes from 10 to 60 seconds (on our CDC 6500) per measurement.

REFERENCES

1. M. Hollander and D. A. Wolfe, Nonparametric Statistical Methods, John Wiley (1973), Chapter 7.

2. E. N. Houstis, R. E. Lynch, T. S. Papatheodorou, and J. R. Rice, Development, evaluation and selection of methods for elliptic partial differential equations, Ann. Assoc. Calcul. Analog., 11, (1975), 98-105.

3. ___________, Evaluation of numerical methods for elliptic partial differential equations, J. Comp. Physics, 27, (1978), 323-350.

4. E. N. Houstis and T. S. Papatheodorou, Comparison of fast direct methods for elliptic problems in Advances in Computer Methods for Partial Differential Equations (R. Vishnevetsky, ed.), IMACS, New Brunswick, NJ, (1977), 46-52.

5. ___________, Algorithm XXX, FFT9: Fast solution of Helmholtz type partial differential equations, ACM Trans. Math. Software, (1979), to appear.

6. E. N. Houstis and J. R. Rice, A population of elliptic partial differential equations in two variables, (1979), to appear.

7. ___________, The performance of high order methods on elliptic partial differential equations with singularities, (1979), to appear.

8. J. R. Rice, The algorithm selection problem, in Advances in Computers, Vol. 15 (Rubicoff and Yovits, eds.), Academic Press, (1976), 65-118.

9. ___________, ELLPACK: A research tool for elliptic partial differential equations software, in Mathematical Software III (J. Rice, ed.), Academic Press, (1977), 319-342.

10. ___________, Methodology for the Algorithm Selection Problem, Proc. IFIP Conf. Numerical Software Performance Evaluation, (1979).