Expansion of the Performance Evaluation Capabilities of ELLPACK

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

Report Number:
83-451
EXPANSION OF THE PERFORMANCE EVALUATION CAPABILITIES OF ELLPACK

John R. Rice

Department of Computer Sciences
Purdue University

CSD-TR 451

September 4, 1983

ABSTRACT

The ELLPACK system is embedded in a system for the performance evaluation of PDE software. The implementation of the present system is tailored to linear, two dimensional problems on rectangular domains. In this note we outline how this system is to be expanded to enlarge the class of problems for which performance evaluations can be carried out.
EXPANSION OF THE PERFORMANCE EVALUATION CAPABILITIES OF ELLPACK

John R. Rice

1. SYNOPSIS OF THE PRESENT SYSTEM

The present system has the following components [Boisvert et al., 1979].

A. PDE Population: A collection of 60+ linear, second order, parameterized partial differential equations on rectangular domains. Most have known exact solutions and the rest have highly accurate numerical solutions. This population is in a large file organized by Problem Number—case so that 17-3 refers to problem 17 with the 3rd set of parameters. Each PDE problem is encoded as a partial ELLPACK program (the grid, solution method, etc. is not present).

B. Problem Solving Module Files: The ELLPACK modules are numbered so that a solution method can be described cryptically. Thus 3/11/22 means "use discretization module #3, indexing module #11 and solution module #22". These are also output files and option "modules" for insertion into programs, but they are rarely used.

C. GENPROGRAM: A Fortran program to synthesize ELLPACK Programs: This program takes component input such as

17-6//7,7/3/11/22///

and creates the ELLPACK program that uses prob. 17-6
with default options
on a grid 7 by 7
with modules #3, #11, #22
with default output and no comments.

GENPROGRAM automatically includes code which collects the standard performance measurements and which writes them out to a specified file.

D. Data Base: The performance measurements are entered into a data base for future reference. This data base presently has the results from about 10,000 numerical solutions of elliptic problems (linear on two dimensional rectangles).
E. **Data Analysis Programs:** A set of programs to retrieve one or many results from the data base to (a) print out the data, (b) make one of several kinds of plots of the data or (c) apply some non-parametric statistical tests to the set of results.

2. **OBJECTIVES OF AN EXPANDED SYSTEM**

An expanded system is needed which

(a) allows other populations of elliptic problems to be used

(b) provides an easy mechanism to test the ELLPACK system itself

(c) uses as much of the existing system and framework as possible.

Objective (a) has an obvious motivation; it has been a long term objective of the ELLPACK project to carry out performance evaluations of PDE software on a wide variety of elliptic problems. Objective (b) arises from the fact that the new ELLPACK system is much more flexible than the previous one so that much more complex ELLPACK programs are possible. All the programs created by the existing system are of one simple structure and thus serve only to test the ELLPACK modules and not the ELLPACK system itself. Objective (c) is a natural one; implicit in its statement is the fact that some things must be changed.

An expanded system must provide for, at least, the following classes of problems (the number ranges are for problem identification)

- **100-199:** linear, two dimensional, non-rectangular domains
- **200-299:** linear, three dimensional, rectangular domains
- **300-399:** linear, three dimensional, non-rectangular domains
- **400-499:** nonlinear problems
- **500-599:** simultaneous equations
- **600-699:** initial value problems
- **700-799:** mutant problems
Mutant refers to problems where substantial Fortran code must be used to "mutate" ELLPACK to handle it; for example in imposing conditions along an interior interface.

The first question is: does the data base allow for such a variety of problems? The data essentially provides for grid, error and timing information for each PDE solution. That seems sufficient for this broader class of problems even though there is other information that would be useful sometimes (e.g. "time" step for initial value problems or further detailed breakdown of timing and errors for simultaneous equations). Plans for systematic, large scale evaluations of the more complex problems are so nebulous at this time that there seems little point in discarding the existing data base facility now. This decision means that the analysis programs can be used without modification.

```plaintext
*COMMENT

TEMPLATE FOR A PROGRAM TO SOLVE 2 COUPLED EQUATIONS
MUST CHANGE DEFAULT TEMPLATE PROCESSOR CHARACTERS

*ENDCOM
*OPTION (CCHR = '#')
*OPTION (CNUM = '@')
*OPTION (ICOLO = 80)
*IF (CASE = 1)
##SET
 NAX = 5
 NY = 5
INIT = 'SET U BY BICUBICS'
ITER = 5
DIS1 = 'HERMITE COLLOCATION'
SOL1 = 'AND GE'
DIS2 = 'HERMITE COLLOCATION'
SOL2 = 'BAND GE'
##END SET
*ENDIF
*IF (CASE = 2)
##SET
 IA = 21
 IM = 21
INIT = 'SET (U=GUESSU)'
ITER = 5
DIS1 = 'HODIE HELMHOLTZ'
SOL1 = 'BAND GE'
DIS2 = '5-POINT STAR'
SOL2 = 'LINPACK BAND'
##END SET
*ENDIF
```

Figure 1. Sample heading for an elliptic problem with two cases.

The next question is: can GENPROGRAM handle such a variety of problems? The change required in GENPROGRAM to
handle this variety of problems is equivalent to a complete rewrite. Thus we will continue to use GENPROGRAM on the existing class of problems and use a different approach for the more complex problems. The new approach uses the TOOLPACK template processor [Rice and Ward, 1982] and is described in more detail in the next section. GENPROGRAM will also be used for the linear, three dimensional, rectangular domain problems.

Recall that GENPROGRAM not only creates an ELLPACK program but also automatically provides the code to measure performance. For the simple problems of the present PDE population, this is accomplished by a standard subroutine. This simple approach is not feasible for more complex problems and special code has to be developed for each class of problems which is then inserted in each ELLPACK program of the class.

3. ELLPACK TEMPLATES FOR MORE COMPLEX ELLIPTIC PROBLEMS

The new approach for more complex problems is to identify the parameters in each problem and then set them to template variables at the top. Thus reading the top of the template clearly identifies these parameters and their values for each parameter set. See Figure 1 for an example template heading and Figure 2 for the essence of the corresponding ELLPACK program. Note that parameters can be numbers or arbitrary character strings (such as module names). This approach gives the flexibility that is needed to accommodate the variety of program structures that appear in complex programs.

One of the strengths of the current system is that a large number of PDE problems can be specified by minimum input. The format such as

17-6/5,5/8/11/22///

will remain except that the only "fixed-value" parameters are the first two sets (problem ID and grid size: e.g. 17-6 and 5,5). The other numerical values that appear are interpreted on a template by template basis. Thus 8 might be the number of iterations for one template and the domain index in another.

One may interpret

17-6/5,5/8/11/-12.5///

as being transformed into one of the following template commands.
#SET
NX = '5'
NY = '5'
ITER = '8'
EGRID = '11'
TOL = '-12.5'
PAR3 = '
PAR4 = '
#END SET

#SET
NX = '5'
NY = '5'
DOMAIN = '8'
A = '11'
B = '12.5'
PAR3 = '
PAR4 = '
#END SET

The 17-6 information is used to select the template and then these variables would be set before the specific variables of the subcases of the problem (e.g. subcase 6 would set further variables).

GRID.
FORTRAN.
@NX X POINTS & @NY POINTS
TIME1 = 0
TIME3 = 0
KEQN = 2
TRIPLE.
FORTRAN.
@INIT
CALL SAVE (UNKNV)
DO 100 I = 1, @ITER
KEQN = 1
DISCRETE.
FORTRAN.
@DIS1
TIME1 = TIME1 + ROTIME
SOLUTION.
FORTRAN.
TIME3 = TIME3 + ROTIME
CALL SAVE (UNKNV)
KEQN = 2
DISCRETE.
FORTRAN.
TIME1 = TIME1 + ROTIME
SOLUTION.
FORTRAN.
TIME3 = TIME3 + ROTIME
CALL SAVE (UNKNV)
100 CONTINUE

Figure 2. The essentials of the ELLPACK program template to solve two equations. Statements to accumulate timing information use the ELLPACK internal variable ROTIME.

Figure 3 shows how a set of non-rectangular domains can be incorporated into an ELLPACK template. Once L is set, the call on SETRNG sets the ranges of the parameter for each of the pieces. This mechanism relies on the fact that the ELLPACK domain processor will terminate as soon as the specified boundary closes. Thus no complaint is made when 5 boundary pieces are specified and only 2 used. The ELLPACK
project already has a large set of parameterized domains to be incorporated into a PDE population for elliptic problems on non-rectangular domains [Rice, 1982]. A few of these are too complex to be reasonable as domains for elliptic problems; the complex ones will be replaced by simpler ones.

**DECLARATION.**

```fortran
COMMON/PASSEL/L, PARAM1, PARAM2
REAL A(5), B(5)

BOUNDARY.
X = XCOORD (T, 1) , Y = YCOORD (T, 1) FOR T = A(1) TO B(1)
X = XCOORD (T, 2) , Y = YCOORD (T, 2) FOR T = A(2) TO B(2)
X = XCOORD (T, 3) , Y = YCOORD (T, 3) FOR T = A(3) TO B(3)
X = XCOORD (T, 4) , Y = YCOORD (T, 4) FOR T = A(4) TO B(4)
X = XCOORD (T, 5) , Y = YCOORD (T, 5) FOR T = A(5) TO B(5)
```

**FORTRAN.**

```fortran
L = @DOMAIN
CALL SETRNG (A, B)
PARAM1 = @PARAM1
PARAM2 = @PARAM2
```

Figure 3. ELLPACK template to access a parameterized set of domains with 1 to 5 boundary pieces. The template variable @DOMAIN is set at the top of the template just as NX or ITER are set in Figure 1. PARAM1 and PARAM2 are parameters that change the domain.

4. ELLPACK SYSTEM TEST MODE

When testing the correct execution of the ELLPACK system, one needs a set of programs that exercises the facilities in many ways. These programs need not involve substantial computations, that is the grid, number of iterations, etc. can be quite small. The whole population will have a "test" mode to facilitate testing the ELLPACK system. For each problem, a particular easy set of input will be identified and this set will be used automatically in the "test" mode. The correct values of L₁, L₂ and L₃ norms (call them ETRUE(I), I=1,2,3) of the error will be made available for this input and the computed results (call them E(I), I=1,2,3) compared to these values. The quantity

```
CHECK = max ((ETRUE(I)-E(I))/ETRUE(I))
```

is computed and if CHECK is larger than .01, a message is printed saying the test failed. Otherwise, no output occurs.

5. SUMMARY

The set of test problems is divided into 8 subsets,
number 0 through 7 as indicated in Section 2. A problem identification line such as

17-6/5,5/8/17/36///

is processed by a top level which merely

(a) reads the problem ID (17-6 here)
(b) selects, as appropriate, the template processor in GEN-

PROGRAM

The selected subprogram will complete the creation of the ELLPACK program.

The data base system will continue unchanged. The results for each class of problems will be kept in separate data bases just to reduce the size of the data bases; it is unlikely that there will be much interaction between them.

A test mode of operation will be installed in the system where the correct results for short computations will be compared automatically with actual results to test the system.

REFERENCES

