# SOFTWARE RELEASE NOTICE

01. SRN Number: PA-SRN-012

02. Project Title:
TPA - Total-System Performance Computer Code Executive Module.

Project No.

03. SRN Title: TPA Version 1.1

04. Originator/Requester:
Thomas J. Ratchford

Date:
01/20/94

05. Summary of Actions

- ■ Release of new code admitted to CM System
- □ Release of modified code:
  - □ Enhancements made
  - □ Corrections made
- □ Change of access code

06. Persons Authorized Access

| Name | RO/RW | A/C/D |
|---|---|---|
| | | |

07. Element Manager Approval: Date:

08. Remarks:

CNWRA Form TOP-6 (08/93)

# SOFTWARE RELEASE NOTICE

01. SRN Number: PA-SRN-012

02. Project Title:
TPA - Total-System Performance Computer Code Executive Module.

Project No.

03. SRN Title: TPA Version 1.1

04. Originator/Requester:
Thomas J. Ratchford

Date:
01/20/94

05. Summary of Actions

- ■ Release of new code admitted to CM System
- □ Release of modified code:
  - □ Enhancements made
  - □ Corrections made
- ■ Change of access code

06. Persons Authorized Access

| Name | RO/RW | A/C/D |
|---|---|---|
| | | |

07. Element Manager Approval: Date:

08. Remarks:

CNWRA Form TOP-6 (08/93)

# SOFTWARE SUMMARY FORM

| 01.Summary Date: 01/20/94 | 02. Summary prepared by(Name and Phone) T.J. Ratchford 522-3083 | 03. Summary Action: New |
|---|---|---|
| 04. Software Date: 8/15/93 | 05. Short Title: TPA | |
| 06. Software Title: TPA - Total-System Performance Assessment Computer Code Executive Module. | | 07. Internal Software ID: NONE |

| 08. Software Type: | 09.Processing Mode: | 10. APPLICATION AREA |
|---|---|---|
| ☐ Automated Data System | ☐ Interactive | A. General: ☐ Scientific/Engineering ☐ Auxiliary Analyses ☐ Total System PA |
| ■ Computer Program | ☐ Batch | ■ Subsystem PA ☐ Other |
| ☐ Subroutine/Module | ■ Combination | b. Specific: |

| 11. Submitting Organization and Address: | 12. Technical Contact(s) and Phone: |
|---|---|
| CNWRA, SwRI, San Antonio, Texas | R. Janetzke, (210) 522-3318 |

13. Narrative:
TPA Executive Module computer code controls the flow of information between several independent modules and does the final processing to produce the Complementary Comulative Distribution Function (CCDF) of the geologic repository.

| 14. Computer Platform | 15. Computer Operating System: | 16. Programming Language(s): | 17. Number of Source Program Statements: |
|---|---|---|---|
| CRAY/XMP | UNIX | FORTRAN | 37,982 lines of code |
| 18. Computer Memory Requirements: UNKNOWN | 19. Tape Drives: NONE | 20. Disk/Drum Units: N/A | 21. Graphics: UNKNOWN |

22. Other Operational Requirements
NONE

| 23. Software Availability: | 24. Documentation Availability: |
|---|---|
| ■ Available ☐ Limited ☐ In-House ONLY | ■ Available ☐ Inadequate ☐ In-House ONLY |

25. Submission Package Status:

Acceptance Criteria: Met ■ Not Met ☐ Software QA Assessment: Successful ■ Unsuccessful ☐

Code Custodian: Thomas J Ratchford Date: 1/20/94

CNWRA Form TOP-4-1 (08/93)

CRAY DIRECTORY FOR TPA EXEC SOURCE CODE LISTING

```
-rwx------   1 tjr1     tjr1         5122 Jan 20 07:46 airexc.F*
-rwx------   1 tjr1     tjr1         3774 Jan 20 07:46 blocks.F*
-rwx------   1 tjr1     tjr1         3483 Jan 20 07:46 c14ctl.H*
-rwx------   1 tjr1     tjr1         5365 Jan 20 07:46 c14exc.F*
-rwx------   1 tjr1     tjr1         3544 Jan 20 07:46 c14rd.F*
-rwx------   1 tjr1     tjr1        12338 Jan 20 07:46 calcrd.F*
-rwx------   1 tjr1     tjr1         1620 Jan 20 07:46 calgbl.H*
-rwx------   1 tjr1     tjr1          729 Jan 20 07:46 cdate1.H*
-rwx------   1 tjr1     tjr1         2013 Jan 20 07:46 center.F*
-rwx------   1 tjr1     tjr1         8160 Jan 20 07:46 cnsqnc.F*
-rwx------   1 tjr1     tjr1         1364 Jan 20 07:46 combin.F*
-rwx------   1 tjr1     tjr1         1716 Jan 20 07:46 comprd.F*
-rwx------   1 tjr1     tjr1         1694 Jan 20 07:46 compre.F*
-rwx------   1 tjr1     tjr1         2374 Jan 20 07:46 comprs.F*
-rwx------   1 tjr1     tjr1          405 Jan 20 07:46 ctime1.H*
-rwx------   1 tjr1     tjr1         2916 Jan 20 07:46 cumrel.H*
-rwx------   1 tjr1     tjr1         7715 Jan 20 07:46 diaexc.F*
-rwx------   1 tjr1     tjr1          324 Jan 20 07:46 dirsh1.H*
-rwx------   1 tjr1     tjr1          324 Jan 20 07:46 dirspn1.H*
-rwx------   1 tjr1     tjr1         5832 Jan 20 07:46 ditctl.H*
-rwx------   1 tjr1     tjr1         1033 Jan 20 07:46 dittpa.dat*
-rwx------   1 tjr1     tjr1         7716 Jan 20 07:46 diwexc.F*
-rwx------   1 tjr1     tjr1         3552 Jan 20 07:46 dmpfil.F*
-rwx------   1 tjr1     tjr1         1863 Jan 20 07:46 dosgbl.H*
-rwx------   1 tjr1     tjr1         2567 Jan 20 07:46 dosrd.F*
-rwx------   1 tjr1     tjr1          810 Jan 20 07:46 dr1ctl.H*
-rwx------   1 tjr1     tjr1         4413 Jan 20 07:46 dr1exc.F*
-rwx------   1 tjr1     tjr1         1296 Jan 20 07:46 dr2ctl.H*
-rwx------   1 tjr1     tjr1         4636 Jan 20 07:46 dr2exc.F*
-rwx------   1 tjr1     tjr1         3661 Jan 20 07:46 dr2rd.F*
-rwx------   1 tjr1     tjr1         1297 Jan 20 07:46 dtcomp.F*
-rwx------   1 tjr1     tjr1         3468 Jan 20 07:46 epalim.F*
-rwx------   1 tjr1     tjr1         1134 Jan 20 07:46 epalim.H*
-rwx------   1 tjr1     tjr1         1031 Jan 20 07:46 epanrm.F*
-rwx------   1 tjr1     tjr1         1445 Jan 20 07:46 epaprb.F*
-rwx------   1 tjr1     tjr1          792 Jan 20 07:46 errmsg.F*
-rwx------   1 tjr1     tjr1         1808 Jan 20 07:46 esort.F*
-rwx------   1 tjr1     tjr1         2630 Jan 20 07:46 exist.F*
-rwx------   1 tjr1     tjr1         3483 Jan 20 07:46 filnam.H*
-rwx------   1 tjr1     tjr1         7695 Jan 20 07:46 floctl.H*
-rwx------   1 tjr1     tjr1         6177 Jan 20 07:46 floexc.F*
-rwx------   1 tjr1     tjr1         3652 Jan 20 07:46 flotpa.dat*
-rwx------   1 tjr1     tjr1         1458 Jan 20 07:46 hdrdat.H*
-rwx------   1 tjr1     tjr1          706 Jan 20 07:46 init.F*
-rwx------   1 tjr1     tjr1         1782 Jan 20 07:46 iounit.H*
-rwx------   1 tjr1     tjr1          755 Jan 20 07:46 lastr.F*
-rwx------   1 tjr1     tjr1          752 Jan 20 07:46 lcmnt.F*
-rwx------   1 tjr1     tjr1          787 Jan 20 07:46 ldigt.F*
-rwx------   1 tjr1     tjr1          789 Jan 20 07:46 lexin.F*
-rwx------   1 tjr1     tjr1         1220 Jan 20 07:46 lflnm.F*
-rwx------   1 tjr1     tjr1          486 Jan 20 07:46 lhsctl.H*
-rwx------   1 tjr1     tjr1         3000 Jan 20 07:46 lhsexc.F*
-rwx------   1 tjr1     tjr1          793 Jan 20 07:46 lkwin.F*
-rwx------   1 tjr1     tjr1         1063 Jan 20 07:46 lkywd.F*
-rwx------   1 tjr1     tjr1         1064 Jan 20 07:46 lmdfr.F*
-rwx------   1 tjr1     tjr1          794 Jan 20 07:46 lmdin.F*
-rwx------   1 tjr1     tjr1          872 Jan 20 07:46 lnmbr.F*
-rwx------   1 tjr1     tjr1          826 Jan 20 07:46 logmsg.F*
-rwx------   1 tjr1     tjr1          731 Jan 20 07:46 lperd.F*
-rwx------   1 tjr1     tjr1          849 Jan 20 07:46 lquote.F*
-rwx------   1 tjr1     tjr1         1262 Jan 20 07:46 lsepr.F*
-rwx------   1 tjr1     tjr1          783 Jan 20 07:46 lsign.F*
-rwx------   1 tjr1     tjr1          763 Jan 20 07:46 lvbar.F*
```

MJR 1/20/94

```
-rwx------   1 tjr1     tjr1         5300 Jan 20 07:46 nefexc.F*
-rwx------   1 tjr1     tjr1         5327 Jan 20 07:46 nefrd.F*
-rwx------   1 tjr1     tjr1        31936 Jan 20 07:46 neftpa.dat*
-rwx------   1 tjr1     tjr1         9043 Jan 20 07:46 opnfil.F*
-rwx------   1 tjr1     tjr1         1695 Jan 20 07:46 opninp.F*
-rwx------   1 tjr1     tjr1         1371 Jan 20 07:46 opnlog.F*
-rwx------   1 tjr1     tjr1         1409 Jan 20 07:46 opnout.F*
-rwx------   1 tjr1     tjr1         6678 Jan 20 07:46 outfmt.F*
-rwx------   1 tjr1     tjr1         6836 Jan 20 07:46 output.F*
-rwx------   1 tjr1     tjr1          973 Jan 20 07:46 paghdr.F*
-rwx------   1 tjr1     tjr1          711 Jan 20 07:46 pagttl.F*
-rwx------   1 tjr1     tjr1         2208 Jan 20 07:46 parinf.F*
-rwx------   1 tjr1     tjr1         7128 Jan 20 07:46 rdfcom.H*
-rwx------   1 tjr1     tjr1        17676 Jan 20 07:46 rdfree.F*
-rwx------   1 tjr1     tjr1         6279 Jan 20 07:46 rdinhd.F*
-rwx------   1 tjr1     tjr1        30269 Jan 20 07:46 rdtok.F*
-rwx------   1 tjr1     tjr1          454 Jan 20 07:46 runits.F*
-rwx------   1 tjr1     tjr1         2578 Jan 20 07:46 scdfwr.F*
-rwx------   1 tjr1     tjr1        11583 Jan 20 07:46 scngbl.H*
-rwx------   1 tjr1     tjr1        76037 Jan 20 07:46 scninp.F*
-rwx------   1 tjr1     tjr1         4019 Jan 20 07:46 scumrd.F*
-rwx------   1 tjr1     tjr1         3816 Jan 20 07:46 scumwr.F*
-rwx------   1 tjr1     tjr1          972 Jan 20 07:46 seictl.H*
-rwx------   1 tjr1     tjr1         3846 Jan 20 07:46 seiexc.F*
-rwx------   1 tjr1     tjr1         1340 Jan 20 07:46 shsort.F*
-rwx------   1 tjr1     tjr1         2106 Jan 20 07:46 size.H*
-rwx------   1 tjr1     tjr1         1296 Jan 20 07:46 sotctl.H*
-rwx------   1 tjr1     tjr1         5376 Jan 20 07:46 sotexc.F*
-rwx------   1 tjr1     tjr1          309 Jan 20 07:46 split.F*
-rwx------   1 tjr1     tjr1         3555 Jan 20 07:46 sums.F*
-rwx------   1 tjr1     tjr1         1283 Jan 20 07:46 tcompr.F*
-rwx------   1 tjr1     tjr1         1662 Jan 20 07:46 temexc.F*
-rwx------   1 tjr1     tjr1         1577 Jan 20 07:46 titpag.F*
-rwx------   1 tjr1     tjr1        12634 Jan 20 07:46 tpa.F*
-rwx------   1 tjr1     tjr1       348326 Jan 20 07:46 tpa.cpp*
-rwx------   1 tjr1     tjr1         7945 Jan 20 07:46 tpa.inp*
-rwx------   1 tjr1     tjr1       369713 Jan 20 07:46 tpa.pre*
-rwx------   1 tjr1     tjr1         1369 Jan 20 07:46 tsdf.F*
-rwx------   1 tjr1     tjr1         1698 Jan 20 07:46 tsort.F*
-rwx------   1 tjr1     tjr1         2902 Jan 20 07:46 type.F*
-rwx------   1 tjr1     tjr1          900 Jan 20 07:46 upcase.F*
-rwx------   1 tjr1     tjr1          729 Jan 20 07:46 vdate1.H*
-rwx------   1 tjr1     tjr1         1620 Jan 20 07:46 volctl.H*
-rwx------   1 tjr1     tjr1         5346 Jan 20 07:46 volexc.F*
-rwx------   1 tjr1     tjr1         3669 Jan 20 07:46 volrd.F*
-rwx------   1 tjr1     tjr1          405 Jan 20 07:46 vtime1.H*
-rwx------   1 tjr1     tjr1          164 Jan 20 07:46 x.tpa.covr*
-rwx------   1 tjr1     tjr1          675 Jan 20 07:46 x.tpa.test*
```

NJR 1/20/94

# TPA Fortran Program
# Static and Dynamic Analysis

June 29, 1993

Earl S. Marwil
John E. Tolli
Scientific Computing Unit
Idaho National Engineering Laboratory

## 1. Introduction

This analysis was performed on the Cray version of the software as provided by Southwest Research Institute (SwRI).

One sample problem was supplied along with the source code. The program was analyzed using the Craft (Cross Reference Analysis of Fortran) tool, FORWARN, the Fortran 77 analyzer, and PC-Metric. These tools provide static analysis, coverage analysis, and complexity analysis.

## 2. References

[1] N.H. Marshall and E.S. Marwil, Cross Reference Analysis of Fortran (CRAFT), EG&G-CATT-9198, EG&G Idaho, Inc., July 1991.
[2] Fortran 77 Analyzer User's Manual, National Bureau of Standards, NBS GCR 81-359, 1981
[3] FORWARN User's Guide, Quibus Enterprises, Inc., July 1991.
[4] PC-Metric User's Guide, SET Laboratories, Inc., 1987.

## 3. Functions

The TPA program contains 78 Fortran routines.

TPA has no alternate entry points.

TPA has 2 extraneous routines: "init" and "lnmbr".

## 4. Common Block Irregularities

There are 27 common blocks in the TPA program.

All common block declarations are consistent.

Common block "lhsctl", containing the single variable "lhsgbl", is declared only in program module "lhsexc"; this common block could be removed and "lhsgbl" made local to "lhsctl".

There are 90 instances of a common block being declared in a routine in which none of its elements are otherwise referenced.

There are 15 instances of a common block variable being defined but unused.

There are 11 instances of a common block variable being undefined and unused.

## 5. Interface Irregularities

Main program module "tpa" calls "opnfil" with a real variable in argument number 6; "opnfil" expects a character string in this argument.

Program module "cnsqnc" calls "sotexc" with too many arguments. Program module "nefrd" calls "errmsg" with too many arguments.

## 6. Local Variable Irregularities

There are 200 instances of a parameter not being used in a module in which it is declared. This is mainly due to several parameters being declared via comdecks.

There are 32 instances of a local variable being undefined and unused in a module in which it is declared.

There are 13 instances of a local variable being defined and unused in a module in which it is declared.

There is 1 instance of a local variable being used but undefined in a module in which it is declared (character variable "null" in module "rdfree").

## 7. Fortran Extensions

All program modules except "shsort" contain some lower case alphabetic characters in their active Fortran.

Program module "blocks" contains an entity name ("letters") which is longer than 6 characters.

Program modules "calcrd", "rdinhd", and "scninp" have data statements which precede specification statements.

Program modules "c14rd", "dosrd", "dr2rd", "rdfree", "rdtok", and "volrd" contain character assignment statements in which there are potential overlaps.

Program module "blocks" contains a statement which has more than 19 continuation lines.

Program modules "parinf" and "scumwr" contain format statement fields which are not separated by a comma.

There are 134 instances of format statements which use the X descriptor without a repeat count.

## 8. Optimization

The following table summarizes the performance data gathered from execution of the sample problem. Only those routines exercised by the sample problem are shown (see "Coverage Analysis" for a list of routines not exercised by the sample problem, i.e., coverage = 0%). The table lists all program modules in descending order according to CPU time. To optimize code execution time, emphasis should be placed on those modules which appear highest in the listing.

The performance data show that a high percentage of the overall execution time (87.461%) is spent in the first routine listed (OUTFMT). Inspection of the source code for this routine reveals that it does very few calculations and a substantial amount of I/O (mostly output) so that the CPU time engendered by the routine is primarily due to execution of the instructions associated with the I/O functions. Vectorization is therefore not a viable option. Since memory references are not a problem (MC/MR < 1, which includes I/O conflicts) and the instruction buffer rate is not unusually high (IBFR < 1), there does not appear to be much to be done to improve this routine's performance with regard to execution time. A detailed optimization analysis might focus on the possibility of using asynchronous I/O to reduce wall clock time during program execution.

PERFORMANCE DATA FOR TPA

| ROUTINE NAME | Time | %ExTime | %AccumT | %Vflops | IFact | MC/MR | IBFR |
|---|---|---|---|---|---|---|---|
| OUTFMT | 9.247 | 87.461 | 87.461 | 0.00000 | 0.00 | 0.315 | 0.910 |
| RDTOK | 0.466 | 4.412 | 91.873 | 0.00000 | 0.00 | 0.015 | 2.076 |
| SCUMWR | 0.248 | 2.343 | 94.216 | 0.00000 | 0.00 | 0.305 | 0.954 |
| NEFRD | 0.144 | 1.359 | 95.575 | 18.37346 | 0.00 | 0.101 | 0.951 |
| SUMS | 0.073 | 0.695 | 96.270 | 70.30520 | 0.00 | 0.060 | 0.000 |
| LSEPR | 0.050 | 0.474 | 96.744 | 0.00000 | 108.53 | 0.073 | 1.516 |
| DMPFIL | 0.040 | 0.382 | 97.126 | 0.00000 | 0.00 | 0.225 | 0.708 |
| UPCASE | 0.035 | 0.329 | 97.455 | 0.00000 | 59.75 | 0.063 | 0.899 |
| RDFREE | 0.028 | 0.267 | 97.722 | 0.00000 | 0.00 | 0.119 | 0.775 |
| LSIGN | 0.027 | 0.258 | 97.980 | 0.00000 | 48.40 | 0.049 | 0.915 |
| LVBAR | 0.027 | 0.257 | 98.237 | 0.00000 | 71.56 | 0.063 | 1.115 |
| LCMNT | 0.027 | 0.253 | 98.490 | 0.00000 | 72.11 | 0.075 | 1.126 |
| LDIGT | 0.024 | 0.226 | 98.716 | 0.00000 | 64.53 | 0.099 | 1.128 |
| LMDIN | 0.022 | 0.208 | 98.924 | 0.00000 | 60.66 | 0.087 | 1.141 |
| LQUOTE | 0.020 | 0.187 | 99.111 | 0.00000 | 71.17 | 0.067 | 0.001 |
| LPERD | 0.019 | 0.180 | 99.291 | 0.00000 | 68.70 | 0.072 | 0.001 |
| LOGMSG | 0.014 | 0.131 | 99.422 | 0.00000 | 0.00 | 0.257 | 0.764 |
| LASTR | 0.010 | 0.096 | 99.518 | 0.00000 | 36.70 | 0.069 | 0.001 |
| TYPE | 0.008 | 0.071 | 99.590 | 0.00000 | 0.35 | 0.088 | 0.443 |
| EPANRM | 0.006 | 0.056 | 99.646 | 0.00000 | 20.29 | 1.293 | 0.000 |
| EXIST | 0.005 | 0.049 | 99.696 | 0.00000 | 0.10 | 0.115 | 0.191 |
| BLOCKS | 0.004 | 0.041 | 99.737 | 0.00000 | 0.00 | 0.121 | 0.915 |
| PAGHDR | 0.004 | 0.038 | 99.775 | 0.00000 | 0.00 | 0.244 | 0.615 |
| COMPRS | 0.003 | 0.029 | 99.804 | 0.00000 | 0.00 | 0.064 | 0.044 |
| PAGTTL | 0.002 | 0.024 | 99.828 | 0.00000 | 0.00 | 0.245 | 0.444 |

| ROUTINE NAME | Time | %ExTime | %AccumT | %Vflops | IFact | MC/MR | IBFR |
|---|---|---|---|---|---|---|---|
| CENTER | 0.002 | 0.018 | 99.846 | 0.00000 | 0.00 | 0.040 | 0.119 |
| LMDFR | 0.002 | 0.018 | 99.864 | 0.00000 | 1.56 | 0.027 | 0.617 |
| LFLNM | 0.002 | 0.018 | 99.882 | 0.00000 | 2.17 | 0.019 | 0.733 |
| OUTPUT | 0.002 | 0.018 | 99.900 | 0.00000 | 0.00 | 0.238 | 0.416 |
| TPA | 0.001 | 0.014 | 99.914 | 0.00000 | 0.00 | 0.323 | 0.940 |
| PARINF | 0.001 | 0.010 | 99.923 | 0.00000 | 0.00 | 0.306 | 0.810 |
| OPNINP | 0.001 | 0.009 | 99.933 | 0.00000 | 0.00 | 0.406 | 0.572 |
| SCDFWR | 0.001 | 0.009 | 99.942 | 0.00000 | 0.00 | 0.368 | 0.713 |
| OPNLOG | 0.001 | 0.008 | 99.950 | 0.00000 | 0.00 | 0.325 | 0.656 |
| SCNINP | 0.001 | 0.008 | 99.957 | 97.64466 | 0.00 | 0.135 | 1.563 |
| OPNFIL | 0.001 | 0.007 | 99.964 | 0.00000 | 0.00 | 0.369 | 0.535 |
| LEXIN | 0.001 | 0.007 | 99.971 | 0.00000 | 1.69 | 0.117 | 0.000 |
| TITPAG | 0.001 | 0.006 | 99.977 | 0.00000 | 0.00 | 0.279 | 0.842 |
| VOLEXC | 0.001 | 0.005 | 99.981 | 0.00000 | 0.00 | 0.485 | 0.473 |
| OPNOUT | 0.000 | 0.005 | 99.986 | 0.00000 | 0.00 | 0.328 | 0.571 |
| ESORT | 0.000 | 0.004 | 99.990 | 0.00000 | 0.00 | 0.078 | 0.031 |
| LKYWD | 0.000 | 0.003 | 99.992 | 0.00000 | 0.21 | 0.111 | 0.892 |
| CNSQNC | 0.000 | 0.002 | 99.994 | 0.00000 | 0.00 | 0.145 | 1.429 |
| TSORT | 0.000 | 0.001 | 99.996 | 0.00000 | 0.00 | 0.023 | 0.021 |
| ERRMSG | 0.000 | 0.001 | 99.997 | 0.00000 | 0.00 | 0.224 | 0.821 |
| CALCRD | 0.000 | 0.001 | 99.998 | 0.00000 | 0.00 | 0.125 | 1.388 |
| COMPRD | 0.000 | 0.000 | 99.998 | 0.00000 | 0.00 | 0.008 | 0.048 |
| COMPRE | 0.000 | 0.000 | 99.999 | 0.00000 | 0.00 | 0.019 | 0.048 |
| DTCOMP | 0.000 | 0.000 | 99.999 | 0.00000 | 0.00 | 0.042 | 0.061 |
| TCOMPR | 0.000 | 0.000 | 99.999 | 0.00000 | 0.00 | 0.031 | 0.061 |
| LKWIN | 0.000 | 0.000 | 100.000 | 0.00000 | 0.06 | 0.000 | 1.005 |
| RDINHD | 0.000 | 0.000 | 100.000 | 0.00000 | 0.00 | 0.335 | 1.165 |
| EPAPRB | 0.000 | 0.000 | 100.000 | 0.00000 | 0.00 | 0.052 | 0.908 |
| SHSORT | 0.000 | 0.000 | 100.000 | 0.00000 | 0.00 | 0.000 | 0.606 |
| TSDF | 0.000 | 0.000 | 100.000 | 99.00990 | 0.00 | 0.072 | 0.422 |
| COMBIN | 0.000 | 0.000 | 100.000 | 0.00000 | 0.00 | 0.773 | 0.731 |
| Totals (All Traced Routines) | 10.573 | 100.000 | 100.000 | 46.07836 | 17.62 | 0.288 | 0.954 |

Key:

- %AccumT = accumulated percentage of total CPU time
- %ExTime = percentage of total CPU time
- %Vflops = percentage of floating point operations due to vector floating point operations
- IBFR = Instruction Buffer Fetch Rate (megafetches/sec)
- IFact = Inline Factor (total calls to routine / average time spent in routine for each call)
- MC = number of memory conflicts
- MR = number of memory references
- Time = total CPU time (sec)

# 9. Coverage Analysis

One sample problem was supplied. A coverage analysis shows that this problem yielded a 58% segment coverage of TPA. Sample problems provided with simulation programs typically achieve 35% to 50% coverage. A statement of software quality cannot be made for routines that have low coverage, i.e., large portions of the code are untested.

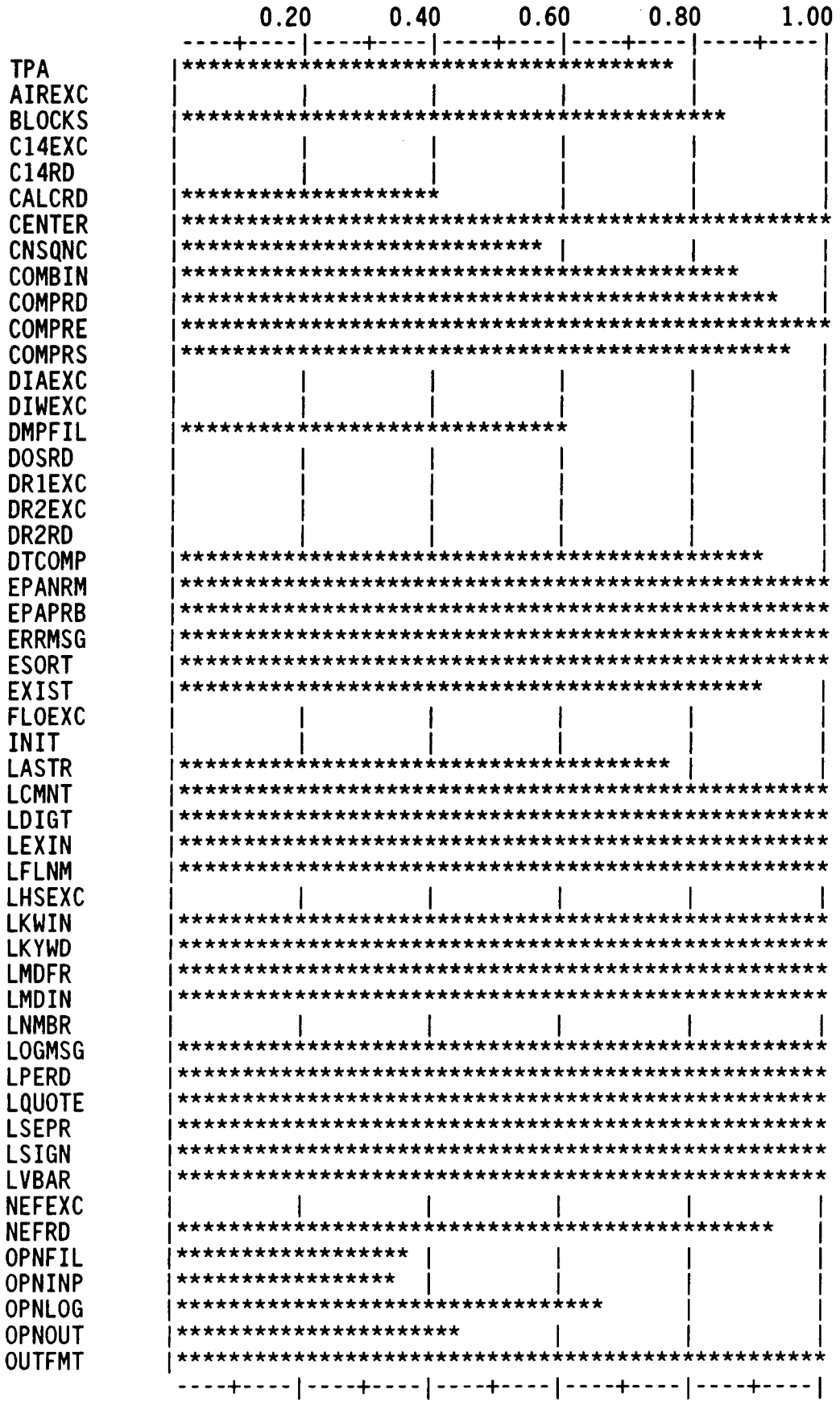
Note that 20 routines have 0% coverage. These routines are not tested with the supplied sample problem.

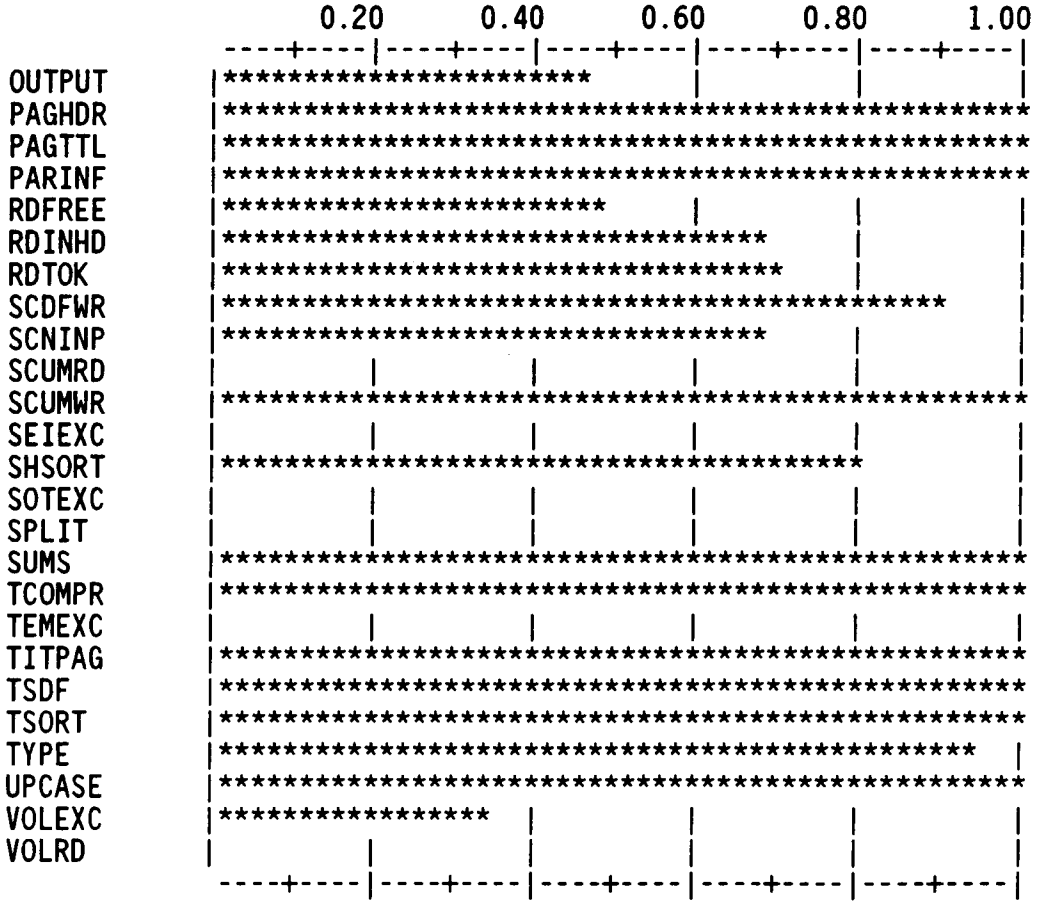
Four routines achieve 20%-39% coverage, 5 routines achieve 40%-59% coverage, 7 routines achieve 60%-79% coverage, 9 routines achieve 80%-99% coverage, and 31 routines achieve 100% coverage.

The following table shows the percent coverage for each routine. Note that "epalim" and "runits" do not appear in the table, since they are non-executable block data routines.

| Module Name | Number of Segments in module | Number of Segments Executed | Percent Segment Coverage |
|---|---|---|---|
| TPA | 42 | 32 | 76.2 |
| AIREXC | 20 | 0 | 0.0 |
| BLOCKS | 6 | 5 | 83.3 |
| C14EXC | 14 | 0 | 0.0 |
| C14RD | 17 | 0 | 0.0 |
| CALCRD | 82 | 32 | 39.0 |
| CENTER | 11 | 11 | 100.0 |
| CNSQNC | 94 | 52 | 55.3 |
| COMBIN | 7 | 6 | 85.7 |
| COMPRD | 13 | 12 | 92.3 |
| COMPRE | 13 | 13 | 100.0 |
| COMPRS | 16 | 15 | 93.7 |
| DIAEXC | 38 | 0 | 0.0 |
| DIWEXC | 36 | 0 | 0.0 |
| DMPFIL | 10 | 6 | 60.0 |
| DOSRD | 18 | 0 | 0.0 |
| DR1EXC | 10 | 0 | 0.0 |
| DR2EXC | 8 | 0 | 0.0 |
| DR2RD | 20 | 0 | 0.0 |
| DTCOMP | 10 | 9 | 90.0 |
| EPANRM | 1 | 1 | 100.0 |
| EPAPRB | 8 | 8 | 100.0 |
| ERRMSG | 1 | 1 | 100.0 |
| ESORT | 10 | 10 | 100.0 |
| EXIST | 10 | 9 | 90.0 |
| FLOEXC | 16 | 0 | 0.0 |
| INIT | 13 | 0 | 0.0 |
| LASTR | 4 | 3 | 75.0 |
| LCMNT | 4 | 4 | 100.0 |
| LDIGT | 4 | 4 | 100.0 |

| Module Name | Number of Segments in module | Number of Segments Executed | Percent Segment Coverage |
|---|---|---|---|
| LEXIN | 4 | 4 | 100.0 |
| LFLNM | 4 | 4 | 100.0 |
| LHSEXC | 5 | 0 | 0.0 |
| LKWIN | 4 | 4 | 100.0 |
| LKYWD | 4 | 4 | 100.0 |
| LMDFR | 4 | 4 | 100.0 |
| LMDIN | 4 | 4 | 100.0 |
| LNMBR | 4 | 0 | 0.0 |
| LOGMSG | 1 | 1 | 100.0 |
| LPERD | 4 | 4 | 100.0 |
| LQUOTE | 4 | 4 | 100.0 |
| LSEPR | 4 | 4 | 100.0 |
| LSIGN | 4 | 4 | 100.0 |
| LVBAR | 4 | 4 | 100.0 |
| NEFEXC | 12 | 0 | 0.0 |
| NEFRD | 25 | 23 | 92.0 |
| OPNFIL | 28 | 10 | 35.7 |
| OPNINP | 12 | 4 | 33.3 |
| OPNLOG | 3 | 2 | 66.7 |
| OPNOUT | 9 | 4 | 44.4 |
| OUTFMT | 40 | 40 | 100.0 |
| OUTPUT | 80 | 36 | 45.0 |
| PAGHDR | 1 | 1 | 100.0 |
| PAGTTL | 1 | 1 | 100.0 |
| PARINF | 1 | 1 | 100.0 |
| RDFREE | 101 | 48 | 47.5 |
| RDINHD | 16 | 11 | 68.7 |
| RDTOK | 92 | 65 | 70.7 |
| SCDFWR | 10 | 9 | 90.0 |
| SCNINP | 621 | 427 | 68.8 |
| SCUMRD | 17 | 0 | 0.0 |
| SCUMWR | 3 | 3 | 100.0 |
| SEIEXC | 6 | 0 | 0.0 |
| SHSORT | 10 | 8 | 80.0 |
| SOTEXC | 14 | 0 | 0.0 |
| SPLIT | 3 | 0 | 0.0 |
| SUMS | 51 | 51 | 100.0 |
| TCOMPR | 10 | 10 | 100.0 |
| TEMEXC | 3 | 0 | 0.0 |
| TITPAG | 1 | 1 | 100.0 |
| TSDF | 5 | 5 | 100.0 |
| TSORT | 10 | 10 | 100.0 |
| TYPE | 15 | 14 | 93.3 |
| UPCASE | 4 | 4 | 100.0 |
| VOLEXC | 12 | 4 | 33.3 |
| VOLRD | 20 | 0 | 0.0 |
| Totals | 1846 | 1070 | 58.0 |


0.20
0.40
0.60
0.80
1.00
TPA
AIREXC
BLOCKS
C14EXC
C14RD
CALCRD
CENTER
CNSQNC
COMBIN
COMPRD
COMPRE
COMPRS
DIAEXC
DIWEXC
DMPFIL
DOSRD
DR1EXC
DR2EXC
DR2RD
DTCOMP
EPANRM
EPAPRB
ERRMSG
ESORT
EXIST
FLOEXC
INIT
LASTR
LCMNT
LDIGT
LEXIN
LFLNM
LHSEXC
LKWIN
LKYWD
LMDFR
LMDIN
LNMBR
LOGMSG
LPERD
LQUOTE
LSEPR
LSIGN
LVBAR
NEFEXC
NEFRD
OPNFIL
OPNINP
OPNLOG
OPNOUT
OUTFMT




| | | | | | |
|---|---|---|---|---|---|
| coverage = 0. | AIREXC | C14EXC | C14RD | DIAEXC | DIWEXC |
| | DOSRD | DR1EXC | DR2EXC | DR2RD | FLOEXC |
| | INIT | LHSEXC | LNMBR | NEFEXC | SCUMRD |
| | SEIEXC | SOTEXC | SPLIT | TEMEXC | VOLRD |
| 0.20 <= coverage < 0.40 | CALCRD | OPNFIL | OPNINP | VOLEXC | |
| 0.40 <= coverage < 0.60 | CNSQNC | DMPFIL | OPNOUT | OUTPUT | RDFREE |
| 0.60 <= coverage < 0.80 | TPA | LASTR | OPNLOG | RDINHD | RDTOK |
| | SCNINP | SHSORT | | | |
| 0.80 <= coverage < 0.85 | BLOCKS | | | | |
| 0.85 <= coverage < 0.90 | COMBIN | | | | |
| 0.90 <= coverage < 0.95 | COMPRD | COMPRS | DTCOMP | EXIST | NEFRD |
| | SCDFWR | TYPE | | | |

| | | | | | |
|---|---|---|---|---|---|
| coverage = 1.00 | CENTER | COMPRE | EPANRM | EPAPRB | ERRMSG |
| | ESORT | LCMNT | LDIGT | LEXIN | LFLNM |
| | LKWIN | LKYWD | LMDFR | LMDIN | LOGMSG |
| | LPERD | LQUOTE | LSEPR | LSIGN | LVBAR |
| | OUTFMT | PAGHDR | PAGTTL | PARINF | SCUMWR |
| | SUMS | TCOMPR | TITPAG | TSDF | TSORT |
| | UPCASE | | | | |

Program coverage for this run =0.58

## 10. Complexity Analysis

Some key metrics are the number of executable statements (sloc), the number of non-blank comments (ncomt), McCabe's extended cyclomatic complexity (vg2), the number of branching statements (cgoto, ugoto, bIF, and lIF), and Halstead's predicted number of errors in (re)writing the code (bhat). Measures are normalized per 100 executable statements for ease of comparison and are listed in the table below.

The branching measures for this code indicate few unconditional GO TO statements and logical IFs for most program modules. This code appears to be well structured.

Except for "split", all routines have a good ratio of non-blank comments to source code.

McCabe's extended cyclomatic complexity (vg2), normalized per 100 lines of source code, indicates moderate to high values. Generally, the routines with the highest complexity are those most likely to have defects. As a guideline, normalized measures of 15 or greater should be considered complex. A software maintenance program should focus on those routines with the highest measures.

## Complexity Report by Subprogram for TPA

| Name | loc | sloc | cmnt | ncomt | ncomt /sloc | vg2 /sloc | cgoto | cgoto /sloc | ugoto | ugoto /sloc | bif bIF | bif /sloc | lif lIF | lif /sloc | Bhat |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| tpa | 835 | 148 | 481 | 463 | 312.8 | 13.5 | 0 | 0.0 | 2 | 1.4 | 9 | 6.1 | 3 | 2.0 | 2 |
| airexc | 485 | 63 | 246 | 240 | 381.0 | 15.9 | 0 | 0.0 | 4 | 6.3 | 9 | 14.3 | 0 | 0.0 | 1 |
| blocks | 117 | 10 | 32 | 29 | 290.0 | 30.0 | 0 | 0.0 | 1 | 10.0 | 0 | 0.0 | 1 | 10.0 | 0 |
| c14exc | 380 | 51 | 212 | 204 | 400.0 | 13.7 | 0 | 0.0 | 3 | 5.9 | 6 | 11.8 | 0 | 0.0 | 1 |
| c14rd | 400 | 33 | 216 | 210 | 636.4 | 27.3 | 0 | 0.0 | 0 | 0.0 | 1 | 3.0 | 1 | 3.0 | 1 |
| calcrd | 747 | 138 | 344 | 333 | 241.3 | 24.6 | 0 | 0.0 | 4 | 2.9 | 30 | 21.7 | 1 | 0.7 | 2 |
| center | 73 | 20 | 44 | 40 | 200.0 | 30.0 | 0 | 0.0 | 0 | 0.0 | 3 | 15.0 | 0 | 0.0 | 0 |
| cnsqnc | 427 | 158 | 187 | 181 | 114.6 | 30.4 | 0 | 0.0 | 0 | 0.0 | 35 | 22.2 | 0 | 0.0 | 2 |
| combin | 219 | 15 | 123 | 118 | 786.7 | 20.0 | 0 | 0.0 | 0 | 0.0 | 1 | 6.7 | 0 | 0.0 | 0 |
| comprd | 103 | 20 | 57 | 54 | 270.0 | 30.0 | 0 | 0.0 | 0 | 0.0 | 2 | 10.0 | 0 | 0.0 | 0 |
| compre | 116 | 20 | 61 | 57 | 285.0 | 30.0 | 0 | 0.0 | 0 | 0.0 | 2 | 10.0 | 0 | 0.0 | 0 |
| comprs | 81 | 29 | 44 | 37 | 127.6 | 24.1 | 0 | 0.0 | 2 | 6.9 | 3 | 10.3 | 1 | 3.4 | 0 |
| diaexc | 597 | 111 | 294 | 284 | 255.9 | 15.3 | 0 | 0.0 | 5 | 4.5 | 12 | 10.8 | 0 | 0.0 | 1 |
| diwexc | 596 | 103 | 303 | 292 | 283.5 | 15.5 | 0 | 0.0 | 4 | 3.9 | 11 | 10.7 | 0 | 0.0 | 2 |
| dmpfil | 141 | 37 | 84 | 76 | 205.4 | 2.7 | 0 | 0.0 | 1 | 2.7 | 0 | 0.0 | 0 | 0.0 | 1 |
| dosrd | 384 | 40 | 194 | 184 | 460.0 | 25.0 | 0 | 0.0 | 1 | 2.5 | 4 | 10.0 | 1 | 2.5 | 1 |
| dr1exc | 410 | 39 | 225 | 219 | 561.5 | 12.8 | 0 | 0.0 | 2 | 5.1 | 4 | 10.3 | 0 | 0.0 | 1 |
| dr2exc | 433 | 45 | 234 | 228 | 506.7 | 8.9 | 0 | 0.0 | 2 | 4.4 | 3 | 6.7 | 0 | 0.0 | 1 |
| dr2rd | 377 | 38 | 202 | 196 | 515.8 | 26.3 | 0 | 0.0 | 1 | 2.6 | 1 | 2.6 | 1 | 2.6 | 1 |
| dtcomp | 95 | 18 | 52 | 49 | 272.2 | 27.8 | 0 | 0.0 | 0 | 0.0 | 2 | 11.1 | 0 | 0.0 | 0 |
| epanrm | 70 | 3 | 47 | 43 | 1433.3 | 33.3 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 |
| epaprb | 281 | 16 | 153 | 147 | 918.8 | 25.0 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 |
| errmsg | 135 | 3 | 92 | 84 | 2800.0 | 33.3 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 |
| esort | 59 | 24 | 28 | 23 | 95.8 | 20.8 | 0 | 0.0 | 1 | 4.2 | 1 | 4.2 | 1 | 4.2 | 0 |
| exist | 72 | 19 | 44 | 39 | 205.3 | 26.3 | 0 | 0.0 | 1 | 5.3 | 3 | 15.8 | 0 | 0.0 | 0 |
| floexc | 443 | 79 | 216 | 210 | 265.8 | 10.1 | 0 | 0.0 | 4 | 5.1 | 6 | 7.6 | 0 | 0.0 | 1 |
| init | 231 | 16 | 111 | 108 | 675.0 | 43.8 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 |
| lastr | 34 | 7 | 26 | 19 | 271.4 | 28.6 | 0 | 0.0 | 0 | 0.0 | 1 | 14.3 | 0 | 0.0 | 0 |
| lcmnt | 34 | 7 | 26 | 19 | 271.4 | 28.6 | 0 | 0.0 | 0 | 0.0 | 1 | 14.3 | 0 | 0.0 | 0 |
| ldigt | 34 | 7 | 26 | 19 | 271.4 | 42.9 | 0 | 0.0 | 0 | 0.0 | 1 | 14.3 | 0 | 0.0 | 0 |
| lexin | 35 | 7 | 27 | 20 | 285.7 | 42.9 | 0 | 0.0 | 0 | 0.0 | 1 | 14.3 | 0 | 0.0 | 0 |
| lflnm | 41 | 7 | 31 | 24 | 342.9 | 71.4 | 0 | 0.0 | 0 | 0.0 | 1 | 14.3 | 0 | 0.0 | 0 |
| lhsexc | 275 | 24 | 157 | 152 | 633.3 | 12.5 | 0 | 0.0 | 0 | 0.0 | 2 | 8.3 | 0 | 0.0 | 0 |
| lkwin | 34 | 7 | 26 | 19 | 271.4 | 42.9 | 0 | 0.0 | 0 | 0.0 | 1 | 14.3 | 0 | 0.0 | 0 |
| lkywd | 39 | 7 | 28 | 21 | 300.0 | 100.0 | 0 | 0.0 | 0 | 0.0 | 1 | 14.3 | 0 | 0.0 | 0 |

| Name | loc | sloc | cmnt | ncomt | ncomt /sloc | vg2 /sloc | cgoto | cgoto /sloc | ugoto | ugoto /sloc | bif bIF | bif /sloc | lif lIF | lif /sloc | Bhat |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| lmdfr | 39 | 7 | 28 | 21 | 300.0 | 100.0 | 0 | 0.0 | 0 | 0.0 | 1 | 14.3 | 0 | 0.0 | 0 |
| lmdin | 34 | 7 | 26 | 19 | 271.4 | 42.9 | 0 | 0.0 | 0 | 0.0 | 1 | 14.3 | 0 | 0.0 | 0 |
| lnmbr | 36 | 7 | 25 | 18 | 257.1 | 71.4 | 0 | 0.0 | 0 | 0.0 | 1 | 14.3 | 0 | 0.0 | 0 |
| logmsg | 50 | 4 | 35 | 32 | 800.0 | 25.0 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 |
| lperd | 34 | 7 | 26 | 19 | 271.4 | 28.6 | 0 | 0.0 | 0 | 0.0 | 1 | 14.3 | 0 | 0.0 | 0 |
| lquote | 35 | 7 | 27 | 20 | 285.7 | 28.6 | 0 | 0.0 | 0 | 0.0 | 1 | 14.3 | 0 | 0.0 | 0 |
| lsepr | 43 | 7 | 33 | 26 | 371.4 | 114.3 | 0 | 0.0 | 0 | 0.0 | 1 | 14.3 | 0 | 0.0 | 0 |
| lsign | 35 | 7 | 27 | 20 | 285.7 | 42.9 | 0 | 0.0 | 0 | 0.0 | 1 | 14.3 | 0 | 0.0 | 0 |
| lvbar | 35 | 7 | 26 | 19 | 271.4 | 28.6 | 0 | 0.0 | 0 | 0.0 | 1 | 14.3 | 0 | 0.0 | 0 |
| nefexc | 390 | 46 | 222 | 212 | 460.9 | 13.0 | 0 | 0.0 | 2 | 4.3 | 5 | 10.9 | 0 | 0.0 | 0 |
| nefrd | 395 | 54 | 210 | 204 | 377.8 | 24.1 | 0 | 0.0 | 3 | 5.6 | 4 | 7.4 | 2 | 3.7 | 1 |
| opnfil | 208 | 59 | 132 | 119 | 201.7 | 27.1 | 0 | 0.0 | 0 | 0.0 | 8 | 13.6 | 0 | 0.0 | 1 |
| opninp | 209 | 31 | 116 | 108 | 348.4 | 12.9 | 0 | 0.0 | 4 | 12.9 | 3 | 9.7 | 0 | 0.0 | 0 |
| opnlog | 213 | 10 | 135 | 1271 | 270.0 | 10.0 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 |
| opnout | 201 | 24 | 115 | 107 | 445.8 | 12.5 | 0 | 0.0 | 4 | 16.7 | 2 | 8.3 | 0 | 0.0 | 0 |
| outfmt | 466 | 88 | 228 | 219 | 248.9 | 26.1 | 0 | 0.0 | 0 | 0.0 | 9 | 10.2 | 0 | 0.0 | 2 |
| output | 585 | 123 | 270 | 259 | 210.6 | 34.1 | 0 | 0.0 | 0 | 0.0 | 15 | 12.2 | 12 | 9.8 | 2 |
| paghdr | 381 | 9 | 216 | 2062 | 288.9 | 11.1 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 |
| pagttl | 135 | 4 | 91 | 832 | 075.0 | 25.0 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 |
| parinf | 229 | 5 | 124 | 1152 | 300.0 | 20.0 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 |
| rdfree | 542 | 219 | 218 | 205 | 93.6 | 27.9 | 0 | 0.0 | 2 | 0.9 | 28 | 12.8 | 1 | 0.5 | 3 |
| rdinhd | 532 | 30 | 259 | 247 | 823.3 | 23.3 | 0 | 0.0 | 5 | 16.7 | 3 | 10.0 | 1 | 3.3 | 0 |
| rdtok | 958 | 560 | 325 | 306 | 54.6 | 12.7 | 0 | 0.0 | 30 | 5.4 | 39 | 7.0 | 0 | 0.0 | 7 |
| scdfwr | 426 | 27 | 247 | 235 | 870.4 | 18.5 | 0 | 0.0 | 0 | 0.0 | 2 | 7.4 | 0 | 0.0 | 1 |
| scninp | 2893 | 1136 | 1261 | 1248 | 109.9 | 23.9 | 0 | 0.0 | 6 | 0.5 | 224 | 19.7 | 3 | 0.3 | 18 |
| scumrd | 425 | 62 | 208 | 199 | 321.0 | 22.6 | 0 | 0.0 | 0 | 0.0 | 6 | 9.7 | 1 | 1.6 | 1 |
| scumwr | 513 | 25 | 299 | 2831 | 132.0 | 8.0 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 1 |
| seiexc | 317 | 31 | 180 | 174 | 561.3 | 9.7 | 0 | 0.0 | 1 | 3.2 | 2 | 6.5 | 0 | 0.0 | 1 |
| SHSORT | 50 | 21 | 22 | 17 | 81.0 | 23.8 | 0 | 0.0 | 1 | 4.8 | 1 | 4.8 | 1 | 4.8 | 0 |
| sotexc | 454 | 66 | 235 | 229 | 347.0 | 10.6 | 0 | 0.0 | 3 | 4.5 | 5 | 7.6 | 0 | 0.0 | 1 |
| split | 11 | 6 | 3 | 1 | 16.7 | 33.3 | 0 | 0.0 | 0 | 0.0 | 1 | 16.7 | 0 | 0.0 | 0 |
| sums | 212 | 64 | 99 | 94 | 146.9 | 40.6 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 |
| tcompr | 108 | 18 | 56 | 52 | 288.9 | 27.8 | 0 | 0.0 | 0 | 0.0 | 2 | 11.1 | 0 | 0.0 | 0 |
| temexc | 240 | 12 | 136 | 1311 | 091.7 | 16.7 | 0 | 0.0 | 0 | 0.0 | 1 | 8.3 | 0 | 0.0 | 0 |
| titpag | 221 | 15 | 132 | 116 | 773.3 | 6.7 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 |
| tsdf | 269 | 11 | 145 | 1401 | 272.7 | 27.3 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 | 0.0 | 0 |
| tsort | 117 | 25 | 55 | 51 | 204.0 | 20.0 | 0 | 0.0 | 1 | 4.0 | 1 | 4.0 | 1 | 4.0 | 0 |
| type | 89 | 22 | 42 | 37 | 168.2 | 59.1 | 0 | 0.0 | 0 | 0.0 | 4 | 18.2 | 0 | 0.0 | 0 |
| upcase | 35 | 7 | 26 | 19 | 271.4 | 42.9 | 0 | 0.0 | 0 | 0.0 | 1 | 14.3 | 0 | 0.0 | 0 |

| Name | loc | sloc | cmnt | ncomt | ncomt /sloc | vg2 /sloc | cgoto | cgoto /sloc | ugoto | ugoto /sloc | bIF | bif /sloc | lIF | lif /sloc | Bhat |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| volexc | 456 | 57 | 244 | 238 | 417.5 | 10.5 | 0 | 0.0 | 2 | 3.5 | 4 | 7.0 | 0 | 0.0 | 1 |
| volrd | 381 | 38 | 204 | 198 | 521.1 | 26.3 | 0 | 0.0 | 1 | 2.6 | 1 | 2.6 | 1 | 2.6 | 1 |

Legend of Metrics in Report

loc -- lines of code
sloc -- number of executable statements
cmnt -- total number of commnts
ncomt -- number of non-blank COMMENT statements
100*ncomt/sloc -- percent, nonblank comments to number of executable statements
100*vg2/sloc -- percent, extended complexity of number of executable statements
cgoto -- number of COMPUTED GO TO statements
100*cgoto/sloc -- percent, computed GOTO's to number of executable statements
ugoto -- number of UNCONDITIONAL GO TO statements
100*ugoto/sloc -- percent, unconditional GOTO's to number of executable statements
bIF -- number of BLOCK IF statements
100*bif/sloc -- percent, Block IF statements to number of executable statements
lIF -- number of LOGICAL IF statements
100*lif/sloc -- percent, logical IF statements to number of executable statements
Bhat -- Halstead's predicted number of errors in writing code