SOFTWARE RELEASE NOTICE

01. SRN Number: PA-SRN-003

02. Project Title: Finite Element Computer Code for Saturated Flow, CNWRA Version 1.1

Project No.

03. SRN Title: Flash V3.0

04. Originator/Requester: Thomas J. Ratchford

Date: 12/14/93

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-108

02. Project Title: Finite Element Computer Code for Saturated Flow, CNWRA Version 1.1	Project No. 20-5702-065
03. SRN Title: FLASH, V3.0	
04. Originator/Requestor: Budhi Sagar	Date: 01/22/96

05. Summary of Actions

- ☐ Release of new software
- ☐ Release of modified software:
  - ☐ Enhancements made
  - ☐ Corrections made
- ☐ Change of access software
- ■ Software Retirement

06. Persons Authorized Access

Name	RO/RW	A/C/D
N/A		

07. Element Manager Approval: [signature] Date: 1/26/96

08. Remarks:

Not considered important to regulatory reviews in revised FY96 OPS Plans.

December 14, 1993

MEMORANDUM

TO: R. Baca

FROM: T.J. Ratchford

SUBJECT: Software Release Notice (SRN)

TOP-18 requires the Element Manager to be notified when a software code has been placed under configuration management. The attached Software Release Notices are to advise you of the placement of FLASH, BIGFLOW, and PORFLOW under configuration management and to request a listing of persons authorized access to FLASH, BIGFLOW, and PORFLOW. Please complete the form for each program and return it to me at your conveniences. If there are any questions please feel free to call me at 3083.

Block 06, Persons Authorized Access.

Under "Name":

Persons name and cray User Name.

Under "RO/RW":

Assign access permission for each person.

RO = Read only access RW = Read and Write access

Block 07: Element Manager Approval and date.

Thank You

Thomas J. Ratchford

Code Custodian

SOFTWARE SUMMARY FORM

01. Summary Date:	02. Summary prepared by (Name and phone)	03. Summary Action:
8/11/93	Mark V. Muller, 522-3222	New
04. Software Date:	05. Short Title:	
June 1992	FLASH, V3.0	

06. Software Title:	07. Internal Software ID:
FLASH - A Finite Element Computer Code for Variably Saturated Flow, CNWRA version 1.1	none

08. Software Type:	09. Processing Mode:	10. APPLICATION AREA
☐ Automated Data System	☐ Interactive	a. General: ■ Scientific/Engineering ☐ Bibliographic/Textual ☐ Process Control ☐ Management Business ☐ Computer Systems Support/Utility ☐ Other
■ Computer Program	■ Batch	
☐ Subroutine/Module	☐ Combination	b. Specific: hydrologic contaminant transport

11. Submitting Organization and Address:	12. Technical Contact(s) and Phone:
CNWRA	Bob Baca, (210) 522-3805

13. Narrative:

FLASH effects a numerical model designed to simulate two-dimensional, variably saturated, single-phase flow and heat conduction in the vadose zone; these processes are modeled for an arbitrary, vertical cross-section.

14. Keywords:

subsurface fluid movement, saturated flow, single phase flow, vadose zone

15. Computer Model and Manufacturer	16. Computer Operating System:	17. Programming Language(s):	18. Number of Source Program Statements:
Cray/XMP	UNICOS	FORTRAN	30,000 (estimated)
19. Computer Memory Requirements:	20. Tape Drives:	21. Disk/Drum Units:	22. Terminals:
100,000 words	N/A	N/A	N/A

23. Other Operational Requirements

None

24. Software Availability:	25. Documentation Availability:
■ Available ☐ Limited ☐ In-House ONLY ☐ Active ☐ Inactive	■ Available ☐ Inadequate ☐ In-House ONLY

26. Submission Package Status:

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

Code Custodian: Mark V. Muller Date: 8/11/93

CNWRA Form TOP-4-1 (11/91)

FLASH Software, Version 1.1
MS-DOS 5.0 File Format

Cray Version

Directory of B:\

```
BT1      IGI    228778 08-06-93   3:18p
BT3      IGI     99506 08-06-93   3:18p
FLASH    X     1058976 08-06-93   3:23p
        3 file(s)    1387260 bytes
                       69632 bytes free
```

Disk 01 of 03
Created 08/09/93

FLASH Software, Version 1.1
MS-DOS 5.0 File Format

Cray Version

Directory of B:\

```
DITCH    IGI    759803 08-06-93   3:19p
FEMESH   IGI      7238 08-06-93   3:19p
FLASH    IGI      7238 08-06-93   3:20p
MESH     IGI      7238 08-06-93   3:23p
T-3      IGI    633221 08-06-93   3:24p
        5 file(s)    1414738 bytes
                       41472 bytes free
```

Disk 02 of 03
Created 08/09/93

FLASH Software, Version 1.1
MS-DOS 5.0 File Format

Cray Version

Directory of B:\

BATCH		147	08-06-93	3:17p
CPL		68	08-06-93	3:18p
R		1012	08-06-93	3:23p
BT1	CRD	5876	08-06-93	3:17p
BT3	CRD	3751	08-06-93	3:18p
FLASH	CRD	5664	08-06-93	3:19p
GWITT	CRD	5664	08-06-93	3:23p
T-3	CRD	15266	08-06-93	3:23p
TEMP	CRD	14598	08-06-93	3:24p
VT-1C	CRD	20337	08-06-93	3:24p
FLASH	DAT	20554	08-06-93	3:19p
FLASH	F	265038	08-06-93	3:20p
TIMCHK	F	1129	08-06-93	3:24p
TIMER	F	525	08-06-93	3:24p
FLASH	ICI	295368	08-06-93	3:20p
FLASH	ICS	2511	08-06-93	3:20p
BT1	ISP	1974	08-06-93	3:18p
BT3	ISP	3780	08-06-93	3:18p
FLASH	ISP	19288	08-06-93	3:20p
FLASH	ISV	57344	08-06-93	3:20p
FLASH	IVO	4096	08-06-93	3:20p
LIB	O	4144	08-06-93	3:23p
TIMCHK	O	608	08-06-93	3:24p
FLASH	OLD	262206	08-06-93	3:21p
FLASH	RPT	44706	08-06-93	3:21p
25 file(s)		1055654 bytes		
		395264 bytes free		

Disk 03 of 03
Created 08/09/93

```
gemstone.7 ~/flash/ver11 =>
gemstone.7 ~/flash/ver11 =>
gemstone.7 ~/flash/ver11 =>
gemstone.7 ~/flash/ver11 =>
gemstone.7 ~/flash/ver11 =>
gemstone.7 ~/flash/ver11 =>
gemstone.7 ~/flash/ver11 =>
gemstone.7 ~/flash/ver11 =>
gemstone.7 ~/flash/ver11 =>
gemstone.7 ~/flash/ver11 =>
gemstone.7 ~/flash/ver11 =>
gemstone.7 ~/flash/ver11 =>
gemstone.7 ~/flash/ver11 =>
gemstone.7 ~/flash/ver11 => pwd
/u1/mmv/flash/ver11
gemstone.8 ~/flash/ver11 => ls -F
s.batch         s.bt1.crd       s.bt1.igi       s.bt1.isp       s.bt3.crd
s.bt3.igi       s.bt3.isp       s.cpl           s.ditch.igi     s.femesh.igi
s.flash.crd     s.flash.dat     s.flash.f       s.flash.ici     s.flash.ics
s.flash.igi     s.flash.isp     s.flash.isv     s.flash.ivo     s.flash.old
s.flash.rpt     s.flash.x       s.gwitt.crd     s.lib.o         s.mesh.igi
s.r+            s.t-3.crd       s.t-3.igi       s.temp.crd      s.timchk.f
s.timchk.o      s.timer.f       s.vt-1c.crd
gemstone.9 ~/flash/ver11 =>
                                                    134.20.1.1  16:25:00
```

I have reviewed this and will make suitable changes to code in the near-term
RG Bacca 2/15/94

FLASH 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). The file "lib.o" from directory */u1/rgb/flash* was needed to resolve some external references in FLASH, as well as the subroutine "timchk" obtained from the same directory. The source for those external references in lib.o were not available and thus not analyzed with the FLASH code.

The FLASH program contains 51 Fortran routines. Access to the source code was provided by SwRI on the INEL Cray. There were apparently no different machine versions embedded in the source code. It appears that the COMMON blocks in the code were at one time run through an include preprocessor as evidenced by the comment lines containing include statements preceding the block.

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.

FLASH aborts when loaded with a core preset of *indefinite*; a preset of *zero* was used to achieve successful execution on the sample problem. Because the program runs much slower with performance tracing turned on, the input file "flash.crd" was modified to run a shorter problem. The number of time intervals was reduced from 1000 down to 3 to achieve an execution time of approximately 2 CPU minutes for the instrumented code.

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

FLASH has 9 alternate entry points ("abort1", "abort2", "abort3", and "abort4" in "abortx"; "insert2" and "insert3" in "insert"; "splnh", "splnm", and "splnt" in "splnr").

FLASH has 2 extraneous subroutines: "darcy" and "isppin".

4. Common Block Irregularities

There are 31 common blocks in the FLASH program. All common block declarations are consistent.

Common block "save2" is declared only once (in "front"). Since it is zeroed out each time "front" is called, it could be eliminated and its contents made local to "front".

There are many instances of a common block being declared in a routine in which none of its elements are otherwise referenced There are several common block variables which are never used. These are too numerous to list in this analysis. The CRAFT analysis report contains complete information on these common blocks and variables.

5. Interface Irregularities

No exceptions to report.

6. Local Variable Irregularities

Parameter "lcn" is assigned inconsistent values in several modules. This causes parameter "nleq" to also have inconsistent values.

There are several instances of a parameter not being used in a module in which it is declared. These also are numerous and not listed here.

Local integer variable "ios" is defined but unused in module "wrt".

Local character variable "vers" is unused in module "resolf", except for an assignment statement in which its own value is assigned to itself.

7. Fortran Extensions

Fortran 77 requires that entity names be no longer than 6 characters. There are 37 instances of entity names which are 7 characters or longer. Fortran 77 requires use of only the uppercase alphabetic characters. All program modules contain some lower case alphabetic characters. These are extensions to the language which are recognized by most compilers. No changes need be made to these names.

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 (82.609%) is spent in the first 8 routines listed (STIFQ, RHSQ, SPLINT, BAND, GAMMA, SPLNM, SPLNH, DATAIN2). This is due primarily to the following (applies to some or all of the 8 routines):

1) a low percentage of floating point operations which are performed in vector mode (%Vflops is small)

2) a high overhead factor for calls to the routines (IFact > 1).

3) a high level of memory conflicts (MC/MR > 1)

4) a high rate of instruction buffer fetches (IBFR > 1)

A detailed optimization analysis effort should focus on these 4 areas.

PERFORMANCE DATA FOR FLASH

ROUTINE NAME	Time	%ExTime	%AccumT	%Vflops	IFact	MC/MR	IBFR
STIFQ	14.238	27.702	27.702	46.87251	0.20	0.809	0.436
RHSQ	7.581	14.750	42.452	74.01616	0.52	0.742	0.816
SPLINT	6.978	13.577	56.029	0.00000	813.58	1.991	0.469
BAND	3.241	6.306	62.335	83.40503	0.00	0.379	0.073
GAMMA	2.973	5.784	68.119	0.00000	411.67	1.946	0.767
SPLNM	2.762	5.375	73.494	0.00000	561.25	1.279	1.084
SPLNH	2.611	5.080	78.574	0.00000	495.61	1.268	1.047
DATAIN2	2.074	4.036	82.609	0.00000	0.00	0.135	1.043
PROCESS	1.890	3.678	86.287	44.52584	0.00	0.299	0.880
RECOVRG	1.290	2.509	88.796	0.00000	949.03	3.577	1.178
PRINT	0.881	1.713	90.510	48.04668	0.00	0.375	0.988
FLASH	0.657	1.277	91.787	44.09481	0.00	0.692	0.962
COPYUC	0.525	1.021	92.808	0.00000	0.01	0.180	0.283
STIFL	0.462	0.898	93.706	66.46841	0.15	0.735	0.726
NEWTON	0.454	0.883	94.589	90.57139	0.00	0.637	0.001
RESOLB	0.406	0.789	95.379	90.71507	0.00	0.685	0.477
RECOVRF	0.402	0.782	96.161	0.00000	159.71	2.238	1.082
SAVIT	0.370	0.720	96.881	0.00000	0.00	0.917	1.341
DJACOB	0.357	0.696	97.577	90.44619	0.00	0.072	0.397
RHSL	0.329	0.639	98.216	74.72393	0.29	0.901	1.005
PECLET	0.262	0.511	98.726	71.03779	0.00	0.541	0.735
SETUP	0.249	0.485	99.211	0.00000	0.00	0.796	0.001
DATAIN1	0.096	0.187	99.398	0.00000	0.00	0.964	0.654
FUZZY	0.076	0.148	99.546	0.00000	1.50	0.278	0.247
STIFT	0.069	0.134	99.680	55.69478	0.00	0.666	0.529
TRANSF	0.049	0.096	99.776	0.00000	17.36	0.152	0.815
RHST	0.045	0.087	99.863	75.77598	0.01	0.653	0.853
NUMERIC	0.025	0.048	99.911	0.00000	0.00	0.043	0.540
INSERT3	0.018	0.035	99.946	0.00000	0.12	0.165	0.084
LISTBC	0.009	0.018	99.964	0.00000	0.00	0.229	0.074
DATES	0.007	0.014	99.978	0.00000	0.00	0.390	0.907
FILSET	0.003	0.006	99.984	0.00000	0.00	0.749	0.542
TRANSK	0.003	0.005	99.990	0.00000	0.75	0.115	0.888
SPLINE	0.002	0.003	99.993	0.00000	0.00	0.182	0.032
SHAPEQ	0.001	0.003	99.995	100.00000	0.00	0.107	0.216
NFIELDS	0.001	0.002	99.997	0.00000	0.00	0.002	0.042
WHEADR	0.001	0.001	99.998	0.00000	0.00	0.857	0.953
STARS	0.000	0.001	99.999	0.00000	0.44	2.071	1.694
SHAPET	0.000	0.001	100.000	100.00000	0.00	0.156	0.227
SHAPEL	0.000	0.000	100.000	100.00000	0.00	0.384	0.384
FIND	0.000	0.000	100.000	0.00000	0.00	0.464	0.396
SHAPEF	0.000	0.000	100.000	0.00000	0.00	0.538	1.101
TIMCHK	0.000	0.000	100.000	51.61291	0.00	0.137	1.087
Totals (All Traced Routines)	51.396	100.000	100.000	48.80621	1069.00	0.838	0.647

```
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

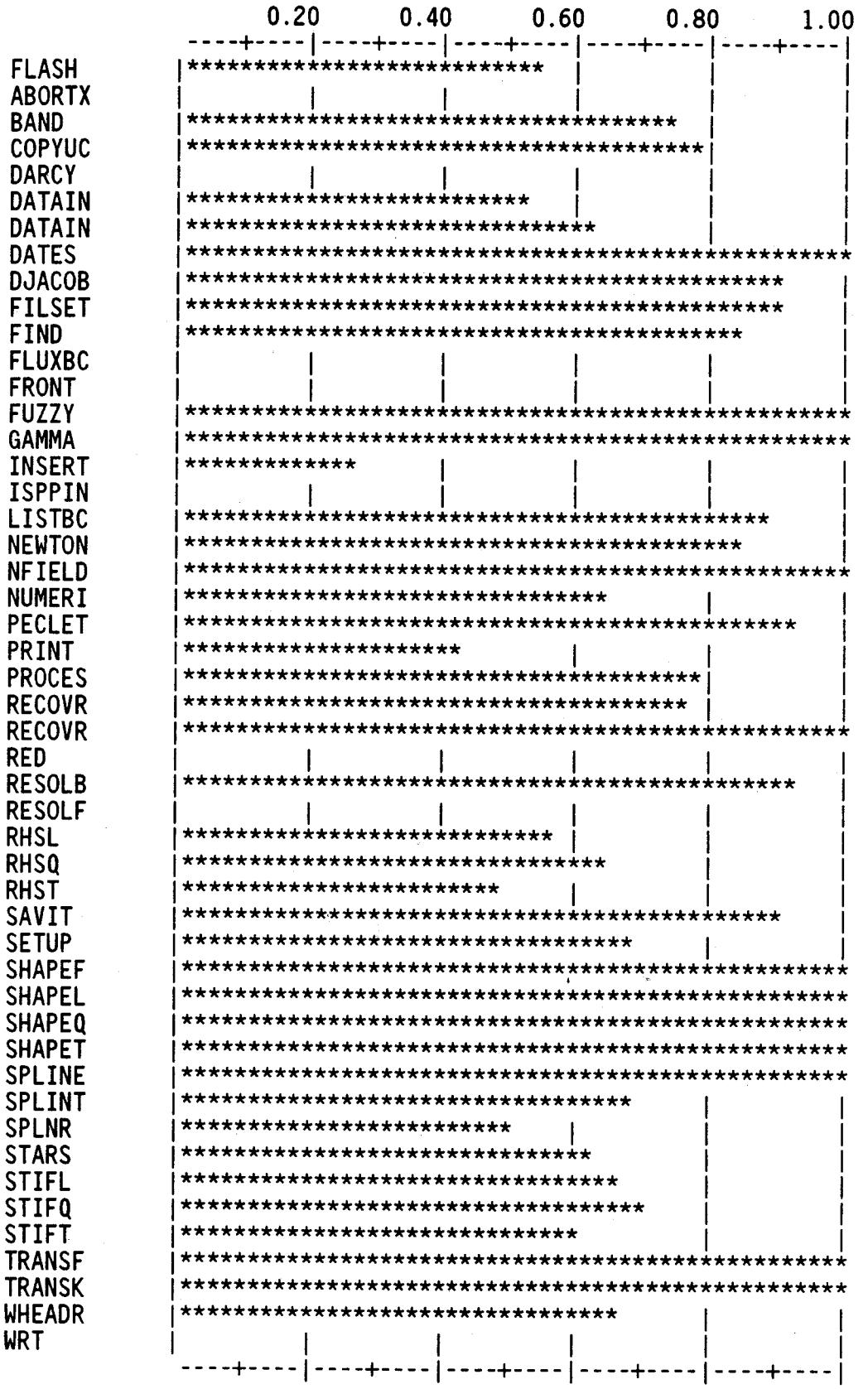
One sample problem was supplied. A coverage analysis shows that this problem yielded a 55% segment coverage of FLASH. Sample problems provided with simulation programs typically achieve only 35% to 50% coverage. A statement of software quality cannot be made for routines that have low coverage, i.e., large portions of the code are untested.

Note that 8 routines have 0% coverage. These routines are not tested with the supplied sample problem. Subroutine "insert" has under 30% coverage. All other routines exceed 40% coverage, with 12 routines achieving 100% coverage.

The following table shows the percent coverage for each routine, except for "catblk" and "shape" which are non-executable block data routines.

Please note that the names in the tables are truncated to 6 characters. Thus, subroutines "datain1" and "datain2" both appear with the name DATAIN, and subroutines "recovrf" and "recovrg" both appear with the name RECOVR.

Module Name	Number of Segments in module	Number of Segments Executed	Percent Segment Coverage
FLASH	121	65	53.7
ABORTX	8	0	0.0
BAND	62	46	74.2
COPYUC	19	15	78.9
DARCY	72	0	0.0
DATAIN	338	179	53.0
DATAIN	77	48	62.3
DATES	7	7	100.0
DJACOB	21	19	90.5
FILSET	11	10	90.9
FIND	6	5	83.3
FLUXBC	35	0	0.0
FRONT	108	0	0.0
FUZZY	6	6	100.0
GAMMA	3	3	100.0
INSERT	27	7	25.9
ISPPIN	37	0	0.0
LISTBC	24	21	87.5
NEWTON	71	60	84.5
NFIELD	7	7	100.0
NUMERI	14	9	64.3
PECLET	14	13	92.9
PRINT	80	34	42.5
PROCES	151	119	78.8
RECOVR	4	3	75.0
RECOVR	1	1	100.0
RED	8	0	0.0
RESOLB	27	25	92.6
RESOLF	44	0	0.0
RHSL	70	39	55.7
RHSQ	90	58	64.4
RHST	90	44	48.9
SAVIT	21	19	90.5
SETUP	77	52	67.5
SHAPEF	1	1	100.0
SHAPEL	5	5	100.0
SHAPEQ	5	5	100.0
SHAPET	5	5	100.0
SPLINE	5	5	100.0
SPLINT	16	11	68.7
SPLNR	10	5	50.0
STARS	8	5	62.5
STIFL	106	70	66.0
STIFQ	126	89	70.6
STIFT	126	75	59.5
TRANSF	3	3	100.0
TRANSK	1	1	100.0
WHEADR	12	8	66.7
WRT	7	0	0.0
Totals	2187	1202	55.0


0.20
0.40
0.60
0.80
1.00
FLASH
ABORTX
BAND
COPYUC
DARCY
DATAIN
DATAIN
DATES
DJACOB
FILSET
FIND
FLUXBC
FRONT
FUZZY
GAMMA
INSERT
ISPPIN
LISTBC
NEWTON
NFIELD
NUMERI
PECLET
PRINT
PROCES
RECOVR
RECOVR
RED
RESOLB
RESOLF
RHSL
RHSQ
RHST
SAVIT
SETUP
SHAPEF
SHAPEL
SHAPEQ
SHAPET
SPLINE
SPLINT
SPLNR
STARS
STIFL
STIFQ
STIFT
TRANSF
TRANSK
WHEADR
WRT

coverage = 0.	ABORTX	DARCY	FLUXBC	FRONT	ISPPIN
	RED	RESOLF	WRT		
0.20 <= coverage < 0.40	INSERT				
0.40 <= coverage < 0.60	FLASH	DATAIN	PRINT	RHSL	RHST
	SPLNR	STIFT			
0.60 <= coverage < 0.80	BAND	COPYUC	DATAIN	NUMERI	PROCES
	RECOVR	RHSQ	SETUP	SPLINT	STARS
	STIFL	STIFQ	WHEADR		
0.80 <= coverage < 0.85	FIND	NEWTON			
0.85 <= coverage < 0.90	LISTBC				
0.90 <= coverage < 0.95	DJACOB	FILSET	PECLET	RESOLB	SAVIT
coverage = 1.00	DATES	FUZZY	GAMMA	NFIELD	RECOVR
	SHAPEF	SHAPEL	SHAPEQ	SHAPET	SPLINE
	TRANSF	TRANSK			

Program coverage for this run =0.55

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.

Most 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 FLASH

Name	loc	sloc	cmnt	ncomt	ncomt /sloc	vg2 /sloc	cgoto	cgoto /sloc	ugoto	ugoto /sloc	bIF	bif /sloc	lIF	lif /sloc	Bhat
Flash	476	221	198	92	41.6	29.0	0	0.0	14	6.3	24	10.9	18	8.1	4
ABORTx	110	37	44	20	54.1	8.1	0	0.0	0	0.0	2	5.4	0	0.0	1
BAND	205	115	55	22	19.1	26.1	0	0.0	1	0.9	12	10.4	1	0.9	2
COPYUC	52	31	17	7	22.6	29.0	0	0.0	0	0.0	2	6.5	5	16.1	0
DARCY	238	129	77	32	24.8	27.1	0	0.0	4	3.1	4	3.1	3	2.3	2
DATAIN1	954	578	265	92	15.9	33.6	1	0.2	52	9.0	81	14.0	41	7.1	9
DATAIN2	339	164	92	36	22.0	30.5	0	0.0	10	6.1	17	10.4	16	9.8	4
DATES	46	17	15	8	47.1	23.5	0	0.0	1	5.9	1	5.9	1	5.9	0
DJACOB	125	52	33	15	28.8	19.2	0	0.0	1	1.9	3	5.8	1	1.9	1
FILSET	79	27	21	9	33.3	55.6	0	0.0	0	0.0	3	11.1	0	0.0	0
FIND	25	8	12	7	87.5	37.5	0	0.0	0	0.0	1	12.5	0	0.0	0
FLUXBC	123	73	35	14	19.2	21.9	1	1.4	9	12.3	0	0.0	11	15.1	1
FRONT	434	221	167	81	36.7	24.4	0	0.0	4	1.8	20	9.0	7	3.2	3
FUZZY	27	11	13	9	81.8	36.4	0	0.0	0	0.0	2	18.2	0	0.0	0
GAMMA	23	9	12	7	77.8	22.2	0	0.0	0	0.0	1	11.1	0	0.0	0
INSERT	115	61	48	29	47.5	16.4	0	0.0	3	4.9	9	14.8	0	0.0	0
ISPPIN	106	68	32	18	26.5	22.1	0	0.0	8	11.8	10	14.7	0	0.0	1
LISTBC	61	38	17	8	21.1	31.6	0	0.0	1	2.6	6	15.8	2	5.3	0
NEWTON	197	109	62	22	20.2	38.5	0	0.0	8	7.3	9	8.3	13	11.9	1
NFIELDS	30	16	12	6	37.5	31.3	0	0.0	1	6.3	1	6.3	2	12.5	0
NUMERIC	47	27	19	9	33.3	37.0	0	0.0	1	3.7	0	0.0	6	22.2	0
PECLET	99	36	35	15	41.7	19.4	0	0.0	1	2.8	2	5.6	1	2.8	0
PRINT	239	140	56	23	16.4	25.0	0	0.0	11	7.9	12	8.6	13	9.3	2
PROCESS	548	266	140	38	14.3	33.5	0	0.0	6	2.3	24	9.0	25	9.4	5
RECOVRF	22	8	11	7	87.5	25.0	0	0.0	0	0.0	1	12.5	0	0.0	0
RECOVRG	14	3	10	7	233.3	33.3	0	0.0	0	0.0	0	0.0	0	0.0	0
RED	62	17	25	13	76.5	23.5	0	0.0	2	11.8	1	5.9	2	11.8	0
RESOLB	123	51	40	18	35.3	25.5	0	0.0	1	2.0	3	5.9	1	2.0	1
RESOLF	192	90	61	26	28.9	24.4	0	0.0	1	1.1	5	5.6	2	2.2	1
RHSL	342	163	128	39	23.9	23.3	0	0.0	1	0.6	18	11.0	1	0.6	3
RHSQ	429	231	138	42	18.2	20.8	0	0.0	3	1.3	23	10.0	3	1.3	5
RHST	429	231	138	42	18.2	20.8	0	0.0	3	1.3	23	10.0	3	1.3	5
SAVIT	89	35	29	11	31.4	31.4	0	0.0	2	5.7	3	8.6	3	8.6	1
SETUP	234	123	70	31	25.2	32.5	0	0.0	1	0.8	16	13.0	0	0.0	2
SHAPEF	25	5	19	11	220.0	20.0	0	0.0	0	0.0	0	0.0	0	0.0	0
SHAPEL	93	38	47	23	60.5	7.9	0	0.0	0	0.0	0	0.0	0	0.0	1

Name	loc	sloc	cmnt	ncomt	ncomt /sloc	vg2 /sloc	cgoto	cgoto /sloc	ugoto	ugoto /sloc	bIF	bif /sloc	lIF	lif /sloc	Bhat
SHAPEQ	223	152	62	31	20.4	2.0	0	0.0	0	0.0	0	0.0	0	0.0	3
SHAPET	196	119	67	34	28.6	2.5	0	0.0	0	0.0	0	0.0	0	0.0	2
SPLINE	43	16	22	14	87.5	18.8	0	0.0	0	0.0	0	0.0	0	0.0	1
SPLINT	69	36	28	15	41.7	22.2	0	0.0	1	2.8	2	5.6	5	13.9	0
SPLNR	110	30	44	16	53.3	13.3	0	0.0	0	0.0	3	10.0	0	0.0	0
STARS	27	11	12	7	63.6	36.4	0	0.0	0	0.0	1	9.1	0	0.0	0
STIFL	449	222	170	55	24.8	26.6	0	0.0	3	1.4	25	11.3	3	1.4	3
STIFQ	534	294	171	53	18.0	23.5	0	0.0	5	1.7	30	10.2	5	1.7	7
STIFT	535	294	202	77	26.2	23.5	0	0.0	5	1.7	30	10.2	5	1.7	7
TRANSF	23	7	13	7	100.0	28.6	0	0.0	0	0.0	1	14.3	0	0.0	0
TRANSK	17	4	11	7	175.0	25.0	0	0.0	0	0.0	0	0.0	0	0.0	0
WHEADR	99	41	24	9	22.0	19.5	0	0.0	0	0.0	4	9.8	2	4.9	0
WRT	44	10	22	12	120.0	40.0	0	0.0	0	0.0	1	10.0	0	0.0	0

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