SOFTWARE RELEASE NOTICE

01. SRN Number: GHGC-SRN-107

02. Project Title: FLAME - A Finite Element Computer Code for Contaminant Transport In Variably-Satuated Media	Project No. 20-5704-192
03. SRN Title: FLAME	
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: P. C. Pe[illegible] Date: 1/26/96

08. Remarks:

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

CNWRA Form TOP-6 (06/95)

SOFTWARE SUMMARY FORM

01. Summary Date: 04/06/94	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: FLAME	
06. Software Title: FLAME - A Finite Element Computer Code For Contaminant Transport In Variably-Saturated Media.		07. Internal Software ID: NONE

08. Software Type:

- ☐ Automated Data System
- ■ Computer Program
- ☐ Subroutine/Module

09. Processing Mode:

- ☐ Interactive
- ☐ Batch
- ■ Combination

10. APPLICATION AREA

A. General:

- ■ Scientific/Engineering
- ☐ Auxiliary Analyses
- ☐ Total System PA
- ☐ Subsystem PA
- ☐ Other

b. Specific:

11. Submitting Organization and Address:

CNWRA, SwRI, San Antonio, Texas

12. Technical Contact(s) and Phone:

R. Baca, (210) 522-3805

13. Narrative:

FLAME - The Flame code is based on the Galerkin finite element method (Fletcher, 1984) in conjunction with the modified partial differential equation for mass transport.

14. Computer Platform: CRAY/XMP	15. Computer Operating System: UNIX	16. Programming Language(s): FORTRAN	17. Number of Source Program Statements: 42,232 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:

■ Available ☐ Limited ☐ In-House ONLY

24. Documentation Availability:

■ Available ☐ Inadequate ☐ In-House ONLY

25. Submission Package Status:

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

Code Custodian: N. J. Ratchford Date: 4/6/94

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

```
gemstone.6 ~/FLAME/WKDIR => ls
Makefile        abortx.F        catblk.F        copyuc.F        datain1.F
datain2.F       dates.F         fandsl.F        find.F          flame.F
flame.cpp       flame.crd       flame.f         flame.ivi*      flametar
fluxbc.F        fuzzy.F         gamma.F         insert.F        listbc.F
masflx.F        mfline.F        mfluxl.F        mfluxq.F        mfluxt.F
newton.F        nfields.F       numeric.F       print.F         process.F
recovrf.F       recovrg.F       resolv.F        rhsl.F          rhsq.F
rhst.F          rsav.F          rvel.F          savit.F         setfils.F
setup.F         shapef.F        shapel.F        shapeq.F        shapet.F
shapfq.F        shapft.F        spline.F        splint.F        splnh.F
splnm.F         stars.F         stifl.F         stifq.F         stift.F
timchk.F        transf.F        transk.F        wheadr.F        x.flame.covr
x.flame.test
gemstone.7 ~/FLAME/WKDIR =>
```

4/6/94 [signature]

134.20.1.1 13:06:06

FLAME Fortran Program Static and Dynamic Analysis

March 10, 1994

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 FLAME program contains 53 Fortran routines.

There are 6 alternate entry points:

```
Module    Alternate Entry Points
------    ----------------------
abortx    abort1, abort2, abort3, abort4
insert    insert2, insert3
```

There are no extraneous subroutines.

4. Common Block Irregularities

There are 24 common blocks in the FLAME program.

All common block declarations are consistent.

Variable exceptions are noted as follows:

Block name	Variable	Exception
/bcys/	kflux	undefined and unused
/catnum/	numrec	defined but unused
/cntrl/	iprop	undefined and unused
/gaussl/	xnl4	defined but unused
/indices/	iendel	undefined and unused

There are several instances of common blocks not being used by modules in which they are declared:

Block name	Modules not using
/gaussl/	rhsq
/sols/	fluxbc
/trandt/	datain2, fandsl, flame, savit, setup

5. Interface Irregularities

No exceptions to report.

6. Local Variable Irregularities

Parameter "itmax" is assigned inconsistent values in different modules.

There are several instances of a parameter not being used in a module in which it is declared:

Parameter	Modules not using
itmax	datain2, print
lbmax	flame, setfils
maxe	print, rsav, rvel, savit, splnh, splnm, wheadr
maxmat	fluxbc, masflx, mfline, newton, print, resolv, wheadr
maxp	splnh, splnm, wheadr
mcn	flame, fluxbc, newton, print, savit
mflls	process
mflns	process
mtt	datain2
ndf	masflx, mfline
npt	rsav
nrmax	flame, setfils
prg	rsav, rvel
vernum	rsav, rvel

Variable exceptions are noted as follows:

Module	Variable	Exception
datainl	mlines	undefined
timchk	iticks	defined but unused
	jticks	undefined and unused
	kind	undefined and unused
wheadr	irunid	used but undefined

```
usrnam      used but undefined
```

7. Fortran Extensions

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

The following routines contain entity names which are longer than 6 characters:

```
flame, datainl, datain2, find, insert, mfluxl, mfluxq, mfluxt,
nfields, numeric, process, process, recovrf, recovrg, rhsl, rhsq,
rhst, rsav, rvel, setfils, stifl, stifq, stift
```

Module "datain1" contains a transfer of control into an "else" block.

Format statement 310 in module "flame" has no comma between the format fields "0P" and "I5".

8. Optimization

FLAME aborts when loaded with a core preset of indefinites.

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.

In order to obtain meaningful statistics for performance evaluation, the program should execute for a reasonable amount of time. Note that the execution time for this sample problem is short (<< 10 sec) and that the resulting statistics may therefore not accurately reflect program performance for more typical (possibly longer) runs.

The performance data show that a high percentage of the overall execution time (90.301%) is spent in the first 6 routines listed. This is due primarily to the following (applies to some or all of the 6 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).

A detailed optimization analysis effort should focus on these 3 areas.

PERFORMANCE DATA FOR FLAME

ROUTINE NAME	Time	%ExTime	%AccumT	%Vflops	IFact	MC/MR	IBFR
FANDSL	0.319	29.025	29.025	55.88275	0.00	1.866	0.520
COPYUC	0.214	19.515	48.540	0.00000	0.00	1.284	0.475
STIFQ	0.196	17.873	66.413	89.24627	0.01	1.108	0.257

DATAIN1	0.109	9.959	76.372	87.43241	0.00	1.332	0.966
FUZZY	0.091	8.272	84.644	0.00000	1.78	3.297	0.245
NUMERIC	0.062	5.658	90.301	0.00000	0.01	1.738	0.465
PROCESS	0.035	3.189	93.490	78.38173	0.00	1.074	0.845
PRINT	0.027	2.434	95.924	82.53780	0.00	0.764	0.903
NFIELDS	0.025	2.286	98.210	0.00000	0.01	1.474	0.042
SETUP	0.006	0.525	98.735	99.58595	0.00	1.137	0.021
FLAME	0.005	0.498	99.233	96.17691	0.00	0.860	0.796
RVEL	0.003	0.228	99.462	58.16528	0.00	1.004	1.012
SHAPFQ	0.001	0.104	99.566	100.00000	0.00	0.433	0.193
LISTBC	0.001	0.081	99.647	100.00000	0.00	1.125	0.130
SHAPEQ	0.001	0.081	99.728	99.83045	0.00	0.579	0.255
NEWTON	0.001	0.067	99.795	98.68825	0.00	0.438	0.215
SETFILS	0.001	0.048	99.843	94.93246	0.00	0.917	0.682
DATES	0.000	0.032	99.875	64.53903	0.00	1.135	0.875
SHAPFT	0.000	0.026	99.901	100.00000	0.00	0.419	0.190
SHAPET	0.000	0.021	99.921	99.76162	0.00	0.679	0.248
SHAPEL	0.000	0.017	99.939	99.56928	0.00	0.806	0.447
RECOVRF	0.000	0.014	99.953	0.00000	0.05	2.627	0.960
GAMMA	0.000	0.012	99.965	0.00000	0.06	3.383	0.672
RECOVRG	0.000	0.010	99.975	0.00000	0.08	2.551	0.861
SPLINE	0.000	0.009	99.985	0.00000	0.00	0.840	0.079
MASFLX	0.000	0.005	99.989	100.00000	0.02	4.523	1.201
FLUXBC	0.000	0.004	99.993	0.00000	0.03	0.526	0.269
STARS	0.000	0.003	99.996	0.00000	0.03	14.409	1.464
SHAPEF	0.000	0.002	99.998	100.00000	0.00	2.736	1.084
TIMCHK	0.000	0.001	100.000	68.08512	0.00	4.493	0.967
DATAIN2	0.000	0.000	100.000	0.00000	0.00	17.500	2.204
MFLINE	0.000	0.000	100.000	0.00000	0.01	21.000	0.645
Totals (All Traced Routines)	1.098	100.000	100.000	79.61590	0.25	1.426	0.491

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

A coverage analysis shows that the sample problem yielded a 48% segment coverage of FLAME. 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 20 routines have 0% coverage. These routines are not tested with the supplied sample problem.

Four routines achieve 1%-19% coverage, 1 routine achieves 20%-39% coverage, 3 routines achieve 40%-59% coverage, 5 routines achieve 60%-79% coverage, 6 routines achieve 80%-99% coverage, and 13 routines achieve 100% coverage.

The following table shows the percent coverage for each routine, except for "catblk" which is a non-executable block data routine.

Module Name	Number of Segments in module	Number of Segments Executed	Percent Segment Coverage
FLAME	106	66	62.3
ABORTX	8	0	0.0
COPYUC	19	14	73.7
DATAIN1	303	164	54.1
DATAIN2	57	8	14.0
DATES	7	7	100.0
FANDSL	60	48	80.0
FIND	6	0	0.0
FLUXBC	21	4	19.0
FUZZY	6	5	83.3
GAMMA	3	3	100.0
INSERT	27	0	0.0
LISTBC	24	11	45.8
MASFLX	41	4	9.8
MFLINE	26	2	7.7
MFLUXL	12	0	0.0
MFLUXQ	29	0	0.0
MFLUXT	29	0	0.0
NEWTON	28	25	89.3
NFIELDS	7	7	100.0
NUMERIC	14	9	64.3
PRINT	22	18	81.8
PROCESS	125	104	83.2
RECOVRF	4	4	100.0
RECOVRG	1	1	100.0
RESOLV	26	0	0.0
RHSL	20	0	0.0
RHSQ	30	0	0.0
RHST	30	0	0.0
RSAV	41	0	0.0
RVEL	39	26	66.7
SAVIT	3	0	0.0
SETFILS	21	12	57.1

SETUP	67	58	86.6
SHAPEF	1	1	100.0
SHAPEL	20	20	100.0
SHAPEQ	7	7	100.0
SHAPET	7	7	100.0
SHAPFQ	10	10	100.0
SHAPFT	10	10	100.0
SPLINE	5	5	100.0
SPLINT	12	0	0.0
SPLNH	3	0	0.0
SPLNM	3	0	0.0
STARS	8	3	37.5
STIFL	39	0	0.0
STIFQ	52	47	90.4
STIFT	52	0	0.0
TIMCHK	4	4	100.0
TRANSF	3	0	0.0
TRANSK	1	0	0.0
WHEADR	3	0	0.0
Totals	1502	714	47.5

```
                0.20      0.40      0.60      0.80      1.00
          ----+----|----+----|----+----|----+----|----+----|
FLAME    |*******************************        |         |
ABORTX   |         |         |         |         |         |
COPYUC   |*************************************  |         |
DATAIN   |***************************  |         |         |
DATAIN   |*******  |         |         |         |         |
DATES    |**************************************************
FANDSL   |****************************************         |
FIND     |         |         |         |         |         |
FLUXBC   |**********         |         |         |         |
FUZZY    |******************************************       |
GAMMA    |**************************************************
INSERT   |         |         |         |         |         |
LISTBC   |***********************      |         |         |
MASFLX   |*****    |         |         |         |         |
MFLINE   |****     |         |         |         |         |
MFLUXL   |         |         |         |         |         |
MFLUXQ   |         |         |         |         |         |
MFLUXT   |         |         |         |         |         |
NEWTON   |*********************************************    |
NFIELD   |**************************************************
NUMERI   |********************************       |         |
PRINT    |*****************************************        |
PROCES   |******************************************       |
RECOVR   |**************************************************
RECOVR   |**************************************************
RESOLV   |         |         |         |         |         |
RHSL     |         |         |         |         |         |
RHSQ     |         |         |         |         |         |
```

```
RHST        |         |         |         |         |         |
RSAV        |         |         |         |         |         |
RVEL        |*********************************      |         |
SAVIT       |         |         |         |         |         |
SETFIL      |*****************************|         |         |
SETUP       |*******************************************      |
SHAPEF      |**************************************************
SHAPEL      |**************************************************
SHAPEQ      |**************************************************
SHAPET      |**************************************************
SHAPFQ      |**************************************************
SHAPFT      |**************************************************
SPLINE      |**************************************************
SPLINT      |         |         |         |         |         |
SPLNH       |         |         |         |         |         |
SPLNM       |         |         |         |         |         |
STARS       |*******************|         |         |         |
STIFL       |         |         |         |         |         |
STIFQ       |*********************************************    |
STIFT       |         |         |         |         |         |
TIMCHK      |**************************************************
TRANSF      |         |         |         |         |         |
TRANSK      |         |         |         |         |         |
WHEADR      |         |         |         |         |         |
             ----+----|----+----|----+----|----+----|----+----|
```

coverage = 0.	ABORTX	FIND	INSERT	MFLUXL	MFLUXQ
	MFLUXT	RESOLV	RHSL	RHSQ	RHST
	RSAV	SAVIT	SPLINT	SPLNH	SPLNM
	STIFL	STIFT	TRANSF	TRANSK	WHEADR
0.01 <= coverage < 0.20	DATAIN	FLUXBC	MASFLX	MFLINE	
0.20 <= coverage < 0.40	STARS				
0.40 <= coverage < 0.60	DATAIN	LISTBC	SETFIL		
0.60 <= coverage < 0.80	FLAME	COPYUC	FANDSL	NUMERI	RVEL
0.80 <= coverage < 0.85	FUZZY	PRINT	PROCES		
0.85 <= coverage < 0.90	NEWTON	SETUP			
0.90 <= coverage < 0.95	STIFQ				
coverage = 1.00	DATES	GAMMA	NFIELD	RECOVR	RECOVR
	SHAPEF	SHAPEL	SHAPEQ	SHAPET	SHAPFQ
	SHAPFT	SPLINE	TIMCHK		

Program coverage for this run =0.48

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 fairly 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 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 FLAME

Name	loc	sloc	cmnt	ncomt	ncomt /sloc	vg2 /sloc	cgoto	cgoto /sloc	ugoto	ugoto /sloc	bIF	bif /sloc	lIF	lif /sloc	Bhat
FLAME	424	197	177	72	36.5	26.4	0	0.0	13	6.6	22	11.2	17	8.6	3
ABORTx	79	25	38	15	60.0	12.0	0	0.0	0	0.0	2	8.0	0	0.0	1
COPYUC	47	31	12	2	6.5	29.0	0	0.0	0	0.0	2	6.5	5	16.1	0
DATAIN1	903	545	268	77	14.1	29.5	1	0.2	44	8.1	62	11.4	37	6.8	6
DATAIN2	223	99	62	18	18.2	33.3	0	0.0	1	1.0	10	10.1	11	11.1	2
DATES	41	17	10	3	17.6	23.5	0	0.0	1	5.9	1	5.9	1	5.9	0
FANDSL	162	92	45	13	14.1	28.3	0	0.0	5	5.4	1	1.1	7	7.6	1
FIND	20	8	7	2	25.0	37.5	0	0.0	0	0.0	1	12.5	0	0.0	0
FLUXBC	90	36	28	6	16.7	27.8	0	0.0	2	5.6	0	0.0	5	13.9	0
FUZZY	22	11	8	4	36.4	36.4	0	0.0	0	0.0	2	18.2	0	0.0	0
GAMMA	16	8	7	2	25.0	25.0	0	0.0	0	0.0	1	12.5	0	0.0	0
INSERT	110	61	43	24	39.3	16.4	0	0.0	3	4.9	9	14.8	0	0.0	0
LISTBC	56	38	12	3	7.9	31.6	0	0.0	1	2.6	6	15.8	2	5.3	0
MASFLX	127	72	43	10	13.9	29.2	0	0.0	5	6.9	6	8.3	6	8.3	1
MFLINE	78	45	19	2	4.4	35.6	0	0.0	3	6.7	2	4.4	5	11.1	1
MFLUXL	134	60	44	6	10.0	11.7	0	0.0	0	0.0	2	3.3	0	0.0	1
MFLUXQ	230	119	69	13	10.9	12.6	0	0.0	0	0.0	2	1.7	1	0.8	2
MFLUXT	230	119	69	13	10.9	12.6	0	0.0	0	0.0	2	1.7	1	0.8	3
NEWTON	104	48	34	9	18.8	27.1	0	0.0	3	6.3	3	6.3	3	6.3	1
NFIELDS	25	16	7	1	6.3	31.3	0	0.0	1	6.3	1	6.3	2	12.5	0
NUMERIC	42	27	14	4	14.8	37.0	0	0.0	1	3.7	0	0.0	6	22.2	0
PRINT	95	41	24	7	17.1	22.0	0	0.0	3	7.3	3	7.3	3	7.3	0
PROCESS	465	218	114	24	11.0	31.7	0	0.0	3	1.4	20	9.2	19	8.7	3
RECOVRF	17	10	6	2	20.0	20.0	0	0.0	0	0.0	1	10.0	0	0.0	0
RECOVRG	9	3	5	2	66.7	33.3	0	0.0	0	0.0	0	0.0	0	0.0	0
RESOLV	101	47	31	9	19.1	25.5	0	0.0	2	4.3	2	4.3	2	4.3	0
RHSL	166	71	57	12	16.9	15.5	0	0.0	0	0.0	2	2.8	0	0.0	1
RHSQ	216	102	65	14	13.7	15.7	0	0.0	0	0.0	3	2.9	0	0.0	2
RHST	212	102	64	14	13.7	15.7	0	0.0	0	0.0	3	2.9	0	0.0	2
RSAV	165	77	49	9	11.7	26.0	0	0.0	1	1.3	6	7.8	7	9.1	1
RVEL	162	72	46	8	11.1	27.8	0	0.0	1	1.4	6	8.3	7	9.7	1
SAVIT	38	6	11	2	33.3	33.3	0	0.0	0	0.0	1	16.7	0	0.0	0
SETFILS	92	51	17	1	2.0	23.5	0	0.0	0	0.0	10	19.6	0	0.0	0
SETUP	140	93	28	8	8.6	36.6	0	0.0	6	6.5	4	4.3	10	10.8	1
SHAPEF	20	5	14	6	120.0	20.0	0	0.0	0	0.0	0	0.0	0	0.0	0
SHAPEL	91	50	33	14	28.0	22.0	0	0.0	0	0.0	2	4.0	0	0.0	1

SHAPEQ	120	70	42	24	34.3	5.7	0	0.0	0	0.0	0	0.0	0	0.0	2
SHAPET	112	58	46	26	44.8	6.9	0	0.0	0	0.0	0	0.0	0	0.0	1
SHAPFQ	126	80	39	20	25.0	7.5	0	0.0	0	0.0	0	0.0	1	1.3	2
SHAPFT	109	62	40	20	32.3	9.7	0	0.0	0	0.0	0	0.0	1	1.6	1
SPLINE	36	16	16	8	50.0	18.8	0	0.0	0	0.0	0	0.0	0	0.0	0
SPLINT	52	28	20	8	28.6	21.4	0	0.0	1	3.6	2	7.1	3	10.7	0
SPLNH	41	9	17	7	77.8	22.2	0	0.0	0	0.0	1	11.1	0	0.0	0
SPLNM	41	9	16	7	77.8	22.2	0	0.0	0	0.0	1	11.1	0	0.0	0
STARS	22	11	7	2	18.2	36.4	0	0.0	0	0.0	1	9.1	0	0.0	0
STIFL	213	99	73	19	19.2	20.2	0	0.0	0	0.0	4	4.0	0	0.0	1
STIFQ	270	136	83	23	16.9	19.9	0	0.0	0	0.0	6	4.4	0	0.0	3
STIFT	270	136	83	23	16.9	19.9	0	0.0	0	0.0	6	4.4	0	0.0	3
TIMCHK	44	11	26	22	200.0	18.2	0	0.0	0	0.0	1	9.1	0	0.0	0
TRANSF	20	10	8	2	20.0	20.0	0	0.0	0	0.0	1	10.0	0	0.0	0
TRANSK	12	5	6	2	40.0	20.0	0	0.0	0	0.0	0	0.0	0	0.0	0
WHEADR	54	13	18	5	38.5	15.4	0	0.0	0	0.0	1	7.7	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