U.S. DEPARTMENT OF THE INTERIOR
U.S. GEOLOGICAL SURVEY

# Fortran Benchmark Programs WetC3D and 3DModel4

## User's Guide

Lawrence M. Baker

OPEN-FILE REPORT 91-328



Menlo Park, California

1991

June 1991

Open File Report No. 91-328

To obtain a copy of the distribution tape described in Section 2, send a blank 2400-foot reel of 1/2-inch tape to:

Lawrence M. Baker
U.S. Geological Survey
345 Middlefield Road MS977
Menlo Park, CA 94025

# FORTRAN BENCHMARK PROGRAMS WETC3D AND 3DMODEL4 USER'S GUIDE

## 1 INTRODUCTION

This report describes the procedures used to compile and execute two USGS benchmark programs, WetC3D and 3DModel4. They are both written in ANSI-standard Fortran-77, except for the use of the VAX Fortran `INCLUDE` statement, which is used to specify the problem parameters for each run and the system-dependent routines, which obtain execution environment and timing information from the operating system.

WetC3D is a finite-difference code that computes the motion of a planar crack expanding under excess fluid pressure in an elastic solid. It is a homogeneous model with all parameters that define the problem specified at compile-time (via Fortran `PARAMETER` statements in an include file, `WetC3D.dat` — note the mixed-case spelling). There are six different benchmark versions of the include file. They vary the ratio of the grid spacing in the fluid grid to the solid grid from 4:1 to 40:1. Their names are of the form WETC3D_n.DAT, where "n" is the ratio: 4, 8, 12, 20, 32, and 40 (see Section B.1).

3DModel4 is a finite-difference code that computes the motion in an elastic solid with arbitrary structure (boundaries) and physical properties (heterogeneities) from a double-couple point source. It is a heterogeneous model with parameters that define the problem specified either at compile-time (via Fortran `PARAMETER` statements in an include file, `3DModel4.inc` — note the mixed-case spelling), or at run-time (via a file whose name is specified in the include file instead). There are two different benchmark versions of the include file: SMALL.INC, which uses a small ($80 \times 80 \times 40$) grid, and LARGE.INC, which uses a large ($220 \times 300 \times 60$) grid (see Section B.2).

Both codes have been run on scalar and vector architectures, and on serial and parallel architectures. (Special effort has been made to make sure that both codes vectorize well on CRAY supercomputers.) Because WetC3D is a homogeneous model, and all coefficients are specified at compile-time, the memory reference pattern is primarily unit-stride. WetC3D's array storage requirements range from 350 thousand (single-precision floating-point) words to 2.6 million words.

3DModel4 allows the specification of an arbitrarily heterogeneous model. However, for most applications, a substantial number of points will share the same media properties, and therefore, the same coefficients. To conserve memory, a single copy of each unique set of coefficients is stored in a separate table. A lookup is performed at run-time for each point in the grid to obtain the proper set of coefficients every time a polynomial is evaluated, generating an indirect (random) memory reference pattern. The small model

Program WetC3D:

| | |
|---|---|
| 1. WETC3D.FOR | Main program and 3-D elastic solid calculations |
| 2. GETHYD.FOR | 2-D fluid-flow calculations |
| 3. GETSPL.FOR | Fluid/solid grid interpolation |
| 4. WETC3D_4.DAT | WetC3D.inc for a 4:1 fluid-to-solid grid ratio |
| 5. WETC3D_8.DAT | WetC3D.inc for a 8:1 fluid-to-solid grid ratio |
| 6. WETC3D_12.DAT | WetC3D.inc for a 12:1 fluid-to-solid grid ratio |
| 7. WETC3D_20.DAT | WetC3D.inc for a 20:1 fluid-to-solid grid ratio |
| 8. WETC3D_32.DAT | WetC3D.inc for a 32:1 fluid-to-solid grid ratio |
| 9. WETC3D_40.DAT | WetC3D.inc for a 40:1 fluid-to-solid grid ratio |

Program 3DModel4:

| | |
|---|---|
| 10. 3DMODEL4.FOR | Main program and most subroutines |
| 11. CHOOSE.FOR | Subroutines to choose which versions of INNER4/INNER2 to use for the run |
| 12. INNER4.FOR | One scalar and eight vector versions of the fourth-order equation solver |
| 13. INNER2.FOR | One scalar and six vector versions of the second-order equation solver |
| 14. ABSORB.FOR | Absorbing boundary and free surface calculations |
| 15. SMALL.INC | 3DModel4.inc for the small problem (2MW) |
| 16. LARGE.INC | 3DModel4.inc for the large problem (30MW) |

System-dependent routines:

| | |
|---|---|
| 17. CPUSUBS.VMS | VAX/VMS version |
| 18. CPUSUBS.SUNOS | Sun/SunOS version (except PFAULT) |
| 19. PFAULT.SUNOS | Sun/SunOS version of PFAULT (written in C) |
| 20. CPUSUBS.UNICOS | CRAY/UNICOS version |

Figure 1. Contents of the Distribution Tape.

version of 3DModel4 requires approximately 2 million words for its array storage; the large model version requires approximately 30 million words.

Both programs perform a relatively small number of operations per memory access.

## 2 DISTRIBUTION TAPE

The distribution tape which accompanies this report is a single 2400-foot reel of 1/2-inch tape, recorded at 1600 bpi. It contains 20 files (see Figure 1), written in ANSI "F" format, using unblocked 80-byte records (card-image format). The volume identifier is "USGS$_{\sqcup\sqcup}$," where "$\sqcup$" represents the ASCII space character ($32_{10}$). Systems that are unable to process the ANSI labels may simply skip them and read only the data portion of each file.

The versions of WetC3D and 3DModel4 that are distributed are for benchmark purposes only. Those who wish to obtain a copy of the current production versions should contact the appropriate author given in Section 4.

# 3 SYSTEM-DEPENDENT ROUTINES

Appendix A contains listings of the system-dependent routines on the distribution tape. Six entry points are required, with specifications as follows.

## 3.1 CPUINI and CPUEND

Subroutine CPUINI (no arguments) writes a description of the execution environment (operating system, CPU model number, memory allocation, date and time) to the standard output unit ("*" in Fortran-77). For example, the VAX/VMS version of CPUINI (see Section A.1.1) prints:

```
VAX/VMS V5.4-1     Host: ISDMNL            CPU type: 11    13-JUN-1991 04:00:02.18

Working Set (pages):  Maximum Authorized:          4096
                      Maximum Authorized Extent:  16400
```

CPUINI also initializes the execution environment, if necessary. For example, the call to `abrupt_underflow` at line 58 in the Sun/SunOS version (see Section A.2.1) disables full IEEE exception handling for floating-point underflow, since the hardware fixup (set the result to zero) is acceptable.

Finally, CPUINI saves initial values for the elapsed time, the CPU time, and the number of page faults for use by CPUEND.

Subroutine CPUEND (no arguments) writes a summary of the run-time statistics to the standard output unit. For example, the VAX/VMS version of CPUEND prints:

```
                    VAX/VMS Execution Statistics
                    ----------------------------

ELAPSED:    0 00:01:34.88  CPU: 0:01:33.55  BUFIO: 5  DIRIO: 12  FAULTS: 6827
```

## 3.2 PERFON and PERFOF

Subroutines PERFON and PERFOF (no arguments) are called to enable and disable the hardware performance monitor on a CRAY supercomputer, respectively. They are not otherwise used by the benchmarks. PERFON and PERFOF need not be implemented (return immediately).

## 3.3 GETTIM

Subroutine GETTIM returns the execution time for the run in its only argument, in floating-point seconds. It is important that the most accurate clock available on the system be used to calculate the execution time, which *must* include fractional parts of a second. (The standard UNIX `time` function is *not* suitable, since it does not return fractional seconds.)

On a multi-processor system, GETTIM should return elapsed time. On a uni-processor system, either elapsed time or CPU time (system plus user) may be used. (Of course, elapsed times will vary with the load on the system, so it is best to run on a dedicated or otherwise idle machine when performing the benchmarks.)

The origin time is arbitrary; both programs always use differences in time to compute performance.

VAX/VMS systems:

```
$ Copy     WetC3D_n.dat WetC3D.dat
$ Fortran WetC3D+GetHyd+GetSpl+CPUSubs.vms
$ Link     WetC3D
$ Assign/User_Mode WetC3D_n.log Sys$Output
$ Run      WetC3D
```

Sun/SunOS systems:

```
% cp WETC3D_n.DAT WetC3D.dat
% cat WETC3D.FOR GETHYD.FOR GETSPL.FOR CPUSUBS.SUNOS > wetc3d.f
% cp PFAULT.SUNOS pfault.c
% f77 -O wetc3d.f pfault.c -o wetc3d
% wetc3d > wetc3d_n.log
```

Notes: The include file is spelled using mixed case characters!
Ignore the messages from f77 about the length mismatches in common scrach.

CRAY/UNICOS systems:

```
% cp WETC3D_n.DAT WetC3D.dat
% cat WETC3D.FOR GETHYD.FOR GETSPL.FOR CPUSUBS.UNICOS > wetc3d.f
% cf77 -o wetc3d -Zv -- wetc3d.f
% wetc3d > wetc3d_n.log
```

Notes: The include file is spelled using mixed case characters!
Ignore the messages from CF77 about the increases in the length of Common 'SCRACH'.

Figure 2. Compilation and Execution Instructions for WetC3D.

### 3.4 PFAULT

Subroutine PFAULT returns the integer count of page faults for the run in its only argument. PFAULT need not be implemented (return zero).

## 4 COMPILATION AND EXECUTION INSTRUCTIONS

No interaction with either program is required at run-time, and neither program reads any input data files. Other than the diagnostics printed on the standard output unit, there is only one file opened for write by each program, and its name is specified in the include file.

Each program prints the execution environment, problem specification, and execution summary on the standard output unit (see Appendix C). Overall performance is expressed in *Useful MFLOPS*, which is calculated by dividing the total number of floating-point operations *which contributed to the solution of the problem* by the execution time.

### 4.1 WetC3D

The WetC3D include files have no user-selectable parameters.

VAX/VMS systems:

```
$ Copy     Small.inc 3DModel4.inc
$ Fortran 3DModel4+Choose+Inner4+Inner2+Absorb+CPUSubs.vms
$ Link     3DModel4
$ Assign/User_Mode Small.log Sys$Output
$ Run      3DModel4
```

Notes: SPREAD should be set to 1.

Sun/SunOS systems:

```
% cp SMALL.INC 3DModel4.inc
% cat 3DMODEL4.FOR CHOOSE.FOR INNER4.FOR INNER2.FOR ABSORB.FOR CPUSUBS.SUNOS
      > 3dmodel4.f
% cp PFAULT.SUNOS pfault.c
% f77 -N150 -O 3dmodel4.f pfault.c -o 3dmodel4
% 3dmodel4 > small.log
```

Notes: The include file is spelled using mixed case characters!
SPREAD should be set to 1.
Ignore the messages from f77 about the length mismatches in common scrach and mesh.

CRAY/UNICOS systems:

```
% cp SMALL.INC 3DModel4.inc
% cat 3DMODEL4.FOR CHOOSE.FOR INNER4.FOR INNER2.FOR ABSORB.FOR CPUSUBS.UNICOS
      > 3dmodel4.f
% cf77 -o 3dmodel4 -Zv -- 3dmodel4.f
% 3dmodel4 > small.log
```

Notes: The include file is spelled using mixed case characters!
Set INNER and SPREAD to 0 to determine the optimal values.
Ignore the messages from CF77 about the increases in the length of Common 'SCRACH' and 'MESH'.

Figure 3. Compilation and Execution Instructions for 3DModel4.

To compile and execute the WetC3D benchmarks, repeat the commands given in Figure 2 for each of the six versions of WETC3D_n.DAT, replacing "n" with the number of the version of being run (4, 8, 12, 20, 32, or 40). The data file specified by ARRAY parameter in the include file is not useful and may be deleted after each run.

A sample run for the problem specified in WETC3D_4.DAT is given in Section C.1.

### 4.2 3DModel4

The 3DModel4 include files have two user-selectable parameters: INNER and SPREAD.

INNER specifies the versions of INNER4/INNER2 to use (they are paired INNR4S with INNR2S, and INR4Vn with INR2Vn, for as many vector versions as available). INNER=1 selects the scalar versions of INNER4/INNER2. INNER=2 to INNER=9 selects one of the eight vector versions of INNER4/INNER2. (They vary in the amount of loop unrolling

in the source code.) `INNER=0` requests CHOOSE.FOR to choose the fastest version at run-time.

When the coefficient table is initally set up, it contains a single copy of each unique set of coefficients. If one of the vector versions of INNER4/INNER2 has been chosen, CHOOSE.FOR will also spread out the coefficients table by making replicas, if requested. It makes enough replicas so that no set of coefficients is reused until at least `SPREAD` values further along the Y axis of the grid, or until the space in the coefficients table is exhausted, whichever comes first. If `SPREAD=0`, CHOOSE.FOR will automatically increase `SPREAD` until performance ceases to improve (or space runs out). If `INNER=0`, CHOOSE.FOR re-runs all the different versions of INNER4/INNER2 each time `SPREAD` is increased, in case one or more versions were previously rate-limited by memory contention.

`SPREAD` should normally be set to `1` for a scalar machine, such as a VAX or a SUN. It should be set to `0` on a vector machine with interleaved memory banks, such as a CRAY, to reduce CPU stalls caused by conflicting memory bank accesses. `INNER` should initially be set to `0` to determine which versions of INNER4/INNER2 run the fastest.

3DModel4 contains compiler directives for the Cray CF77 Compiling System, which are used to control parallelization/vectorization on a CRAY supercomputer. These statements all begin with `CFPP$`, so they should be safely treated as comments by other Fortran compilers.

To compile and execute the small 3DModel4 benchmark, use the commands given in Figure 3. For the large run, replace "`SMALL`" with "`LARGE`," "`Small`" with "`Large`," and "`small`" with "`large`," respectively, and reissue the commands. The data file specified by `SEISFI` parameter in the include file is not useful and may be deleted after each run.

A sample run for the problem specified in SMALL.INC is given in Section C.2.

## 5 CONTACTS

Benchmark versions of WetC3D and 3DModel4:

Lawrence M. Baker
U.S. Geological Survey
345 Middlefield Road MS977
Menlo Park, CA 94025
(415) 329-5608 or FTS 459-5608

Production version of WetC3D:

Dr. Bernard Chouet
U.S. Geological Survey
345 Middlefield Road MS977
Menlo Park, CA 94025
(415) 329-4796 or FTS 459-4796

Production version of 3DModel4:

Dr. Arthur D. Frankel
U.S. Geological Survey
National Center
12201 Sunrise Valley Drive MS922
Retson, VA 22092
(703) 648-4119 or FTS 959-4119

# APPENDIX A
# SYSTEM-DEPENDENT ROUTINES

## A.1 VAX/VMS VERSION

### A.1.1 CPUSUBS.VMS

```
C-------------------------------------------------------------------------
C     SUBROUTINE CPUINI -- PERFORM CPU INITIALIZATION
C-------------------------------------------------------------------------
C
C                       THIS IS THE VAX/VMS VERSION
C
C     ENTRY POINT CPUINI:
C
C          PRINT THE SYSTEM/CPU ENVIRONMENT USING LIB$GETSYI
C          PRINT THE PROCESS MEMORY ENVIRONMENT USING LIB$GETJPI
C          INITIALIZE EXECUTION STATISTICS USING LIB$INIT_TIMER
C
C     ENTRY POINT CPUEND:
C
C          PRINT EXECUTION STATISTICS USING LIB$SHOW_TIMER
C
C     PERFORMANCE MONITOR ENTRY POINTS PERFON AND PERFOF:
C
C          RETURN
C
C     ENTRY POINT GETTIM:
C
C          RETURN JOB'S CPU TIME USING LIB$GETJPI
C
C     ENTRY POINT PFAULT:
C
C          RETURN JOB'S PAGE FAULTS USING LIB$GETJPI
C
C-------------------------------------------------------------------------
C
C                               Disclaimer
C
C Although this program has been tested by the Geological Survey, United
C States Department of the Interior, no warranty, expressed or implied,
C is made by the Geological Survey, as to the accuracy and functioning
C of the program and related program material, nor shall the fact of
C distribution constitute any such warranty, and no responsibility is
C assumed by the Geological Survey in connection therewith.
C
C-------------------------------------------------------------------------
C
      Subroutine CPUINI
```

```
C
      Include    '($JPIDef)/NoList'
      Include    '($LNMDef)/NoList'
      Include    '($SYIDef)/NoList'
C
      Integer*4  LIB$GETSYI, SYS$TRNLNM, LIB$DATE_TIME, LIB$GETJPI
      Integer*4  LIB$INIT_TIMER, LIB$SHOW_TIMER
      External   LPRINT
C
      Integer*4  status, bufadd, handle, cputim, wsauth, wsauthext
      Integer*2  itmlst(8), buflen, item
      Character  host*16, version*8, cpuid*4, daytim*24
      Save       handle
      Equivalence  (itmlst(1), buflen)
      Equivalence  (itmlst(2), item  )
      Equivalence  (itmlst(3), bufadd)
      Equivalence  (itmlst(5), lenadd)
      Data         itmlst/8*0/
C
      status = LIB$GETSYI (SYI$_VERSION,,version)
      status = LIB$GETSYI (SYI$_NODENAME,,host)
      If (status .and. (host .ne. ' ')) Then
         nodlen = LEN(host) + 2
      Else
         buflen = LEN(host)
         item   = LNM$_STRING
         bufadd = %LOC(host)
         lenadd = %LOC(nodlen)
         status = SYS$TRNLNM (,'LNM$SYSTEM_TABLE','SYS$NODE',,itmlst)
         If (.not. status) Then
            Call LIB$SIGNAL (%VAL(status))
         End If
      End If
      status = LIB$GETSYI (SYI$_CPU,,cpuid)
      status = LIB$DATE_TIME (daytim)
      status = LIB$GETJPI (JPI$_WSAUTH,,,wsauth)
      status = LIB$GETJPI (JPI$_WSAUTHEXT,,,wsauthext)
      Write (*,601) version, host(1:nodlen-2), cpuid, daytim, wsauth,
     1              wsauthext
  601 Format (/' VAX/VMS ', A, '   Host: ', A, '  CPU type: ', A,
     1        '  ', A//
     2        ' Working Set (pages):  Maximum Authorized:', T52, I6/
     3        ' ', T24 'Maximum Authorized Extent:', T52, I6/)
      handle = 0
      status = LIB$INIT_TIMER (handle)
      Return
C
      Entry CPUEND
C
      status = LIB$SHOW_TIMER (handle,,LPRINT,0)
      Return
C
      Entry PERFON
C
      Entry PERFOF
C
```

```
      Return
C
      Entry GETTIM (time)
C
      status = LIB$GETJPI (JPI$_CPUTIM,,,cputim)
      time = cputim / 100.0
      Return
C
      Entry PFAULT (nfault)
C
      status = LIB$GETJPI (JPI$_PAGEFLTS,,,nfault)
      Return
C
      End
      Integer*4 Function LPRINT (string, arg)
C
      Include    '($SSDef)/NoList'
C
      Integer*4  arg
      Character  string*(*)
C
      Write (*,101) string
  101 Format (/' ', T26, 'VAX/VMS Execution Statistics'/
     1         ' ', T26, '----------------------------'//A/)
      LPRINT = SS$_NORMAL
      Return
C
      End
```

## A.2 SUN/SUNOS VERSION

### A.2.1 CPUSUBS.SUNOS

```
C-----------------------------------------------------------------------
C     SUBROUTINE CPUINI -- PERFORM CPU INITIALIZATION
C-----------------------------------------------------------------------
C
C                      THIS IS THE SUN/SUNOS VERSION
C
C     ENTRY POINT CPUINI:
C
C          DISABLE FLOATING-POINT UNDERFLOW TRAPS AS MUCH AS POSSIBLE
C             USING abrupt_underflow ()
C          INITIALIZE EXECUTION STATISTICS USING ETIME AND PFAULT
C
C     ENTRY POINT CPUEND:
C
C          PRINT EXECUTION STATISTICS USING ETIME AND PFAULT
C
C     PERFORMANCE MONITOR ENTRY POINTS PERFON AND PERFOF:
C
C          RETURN
C
C     ENTRY POINT GETTIM:
C
C          RETURN JOB'S CPU TIME USING ETIME
```

```
C
C     ENTRY POINT PFAULT:
C
C          C ROUTINE PFAULT.SUNOS
C
C-----------------------------------------------------------------------
C
C                                Disclaimer
C
C Although this program has been tested by the Geological Survey, United
C States Department of the Interior, no warranty, expressed or implied,
C is made by the Geological Survey, as to the accuracy and functioning
C of the program and related program material, nor shall the fact of
C distribution constitute any such warranty, and no responsibility is
C assumed by the Geological Survey in connection therewith.
C
C-----------------------------------------------------------------------
C
      Subroutine CPUINI
C
      Integer    TIME
C
      Real       tarray(2), cputim
      Integer    cpuhr, cpumin, cpusec, cpuhun
      Character  versn*4, host*16, cpuid*5, daytim*24
      Save       lapsec, cputim, nfault
C
      versn   = '4.0'
      Call hostnm (host)
      cpuid   = 'SPARC'
      Call fdate (daytim)
      Write (*,601) versn, host, cpuid, daytim
  601 Format (/' SUN/SUNOS ', A, '   Host: ', A, '  CPU type: ', A,
     1        ' ', A/)
      Call abrupt_underflow ()
      lapsec = TIME ()
      Call ETIME (tarray)
      cputim = tarray(1) + tarray(2)
      Call PFAULT (nfault)
      Return
C
      Entry CPUEND
C
      Call ETIME (tarray)
      lapsec = TIME() - lapsec
      laphr  = lapsec / 3600
      lapsec = lapsec - (laphr*3600)
      lapmin = lapsec / 60
      lapsec = lapsec - (lapmin*60)
      laphun = 0
      cputim = tarray(1) + tarray(2) - cputim
      cpuhun = NINT (cputim*100.)
      cpuhr  = cpuhun / (3600*100)
      cpuhun = cpuhun - (cpuhr*3600*100)
      cpumin = cpuhun / (60*100)
      cpuhun = cpuhun - (cpumin*60*100)
```

```
      cpusec = cpuhun / 100
      cpuhun = cpuhun - (cpusec*100)
      Call PFAULT (n)
      nfault = n - nfault
      Write (*,602) laphr, lapmin, lapsec, laphun,
     1              cpuhr, cpumin, cpusec, cpuhun, nfault
  602 Format (/' ', T27, 'SUN/SUNOS Execution Statistics'/
     1        ' ', T27, '-----------------------------'//
     2        ' Elapsed: ', I4, 2(':', I2.2), '.', I2.2,
     3        '    CPU: ',  I4, 2(':', I2.2), '.', I2.2,
     4        '    Faults: ', I8/)
      Return
C
      Entry PERFON
C
      Entry PERFOF
C
      Return
C
      Entry GETTIM (secs)
C
      Call ETIME (tarray)
C     TARRAY(1) : USER TIME
C     TARRAY(2) : SYSTEM TIME ON BEHALF OF USER
      secs = tarray(1) + tarray(2)
      Return
C
      End
```

## A.2.2 PFAULT.SUNOS

```
/*
 * F77-callable subroutine to report number of page faults so far
 */

/*
                                Disclaimer

 Although this program has been tested by the Geological Survey, United
 States Department of the Interior, no warranty, expressed or implied,
 is  made  by the Geological Survey, as to the accuracy and functioning
 of the program and related program material, nor  shall  the  fact  of
 distribution  constitute  any  such warranty, and no responsibility is
 assumed by the Geological Survey in connection therewith.

*/

#include <sys/time.h>
#include <sys/resource.h>
#include <local/f77types.h>
#define RUSAGE_SELF 0

void pfault_(n)
F77INT *n;

{
struct rusage s;
```

```
27
28 getrusage(RUSAGE_SELF, &s);
29 *n = s.ru_minflt + s.ru_majflt;
30 }
```

## A.3 CRAY/UNICOS VERSION

### A.3.1 CPUSUBS.UNICOS

```
 1 C-----------------------------------------------------------------------
 2 C     SUBROUTINE CPUINI -- PERFORM CPU INITIALIZATION
 3 C-----------------------------------------------------------------------
 4 C
 5 C                       THIS IS THE CRAY/UNICOS VERSION
 6 C
 7 C     ENTRY POINT CPUINI:
 8 C
 9 C          CALL GETHMC TO OBTAIN MACHINE CHARACTERISTICS
10 C          CALL SECOND AND TIMEF TO INITIALIZE CPU AND ELAPSED TIME
11 C
12 C     ENTRY POINT CPUEND:
13 C
14 C          PRINT CPU AND ELAPSED TIME STATISTICS
15 C
16 C     PERFORMANCE MONITOR ENTRY POINTS PERFON AND PERFOF:
17 C
18 C          RETURN
19 C
20 C     ENTRY POINT GETTIM:
21 C
22 C          RETURN TASK'S CPU TIME USING TSECND
23 C
24 C     ENTRY POINT PFAULT:
25 C
26 C          RETURN
27 C
28 C-----------------------------------------------------------------------
29 C
30 C                                Disclaimer
31 C
32 C Although this program has been tested by the Geological Survey, United
33 C States Department of the Interior, no warranty, expressed or implied,
34 C is  made  by the Geological Survey, as to the accuracy and functioning
35 C of the program and related program material, nor  shall  the  fact  of
36 C distribution  constitute  any  such warranty, and no responsibility is
37 C assumed by the Geological Survey in connection therewith.
38 C
39 C-----------------------------------------------------------------------
40 C
41 CFPP$ SKIP R
42 C
43       Subroutine CPUINI
44 C
45       Real       lapms
46       Integer    cpuhr, cpumin, cpusec, cpuhun
47       Integer    mctabl(0:127)
```

```
      Integer    cputyp, mbanks, ncpus, memsiz, mcycle
      Integer    cppsec, mbusy
      Logical    hasgs, hasbdm
C...  Hand translated from /usr/include/sys/table.h
      Equivalence  (mctabl( 0), cputyp)
      Equivalence  (mctabl( 1), mbanks)
      Equivalence  (mctabl( 2), ncpus )
      Equivalence  (mctabl( 4), memsiz)
      Equivalence  (mctabl( 5), mcycle)
      Equivalence  (mctabl( 6), cppsec)
      Equivalence  (mctabl( 8), mbusy )
      Equivalence  (mctabl(65), hasgs )
      Equivalence  (mctabl(72), hasbdm)
      Save         lapms, cputim
C
      Call GETHMC (mctabl)
      Call DATE   (idate )
      Call CLOCK  (itime )
      Write (*,601) cputyp, idate, itime, ncpus, memsiz/(1024*1024),
     1              cppsec, mbanks, hasgs, mcycle, hasbdm, mbusy
  601 Format (/' ', T19, 'CRAY/UNICOS  ', A8, ' System  ', A8, ' ', A8//
     1        ' ', T6 , I8, ' Processors',
     2             T41, I6, 'MW Main memory size'/
     3        ' ', T6 , I8, ' Clock period (pico secs)',
     4             T41, I8, ' Memory banks'/
     5        ' ', T6 , L8, ' Gather/scatter hardware',
     6             T41, I8, ' Memory read time (cycles)'/
     7        ' ', T6 , L8, ' Bidirectional memory',
     8             T41, I8, ' Memory bank busy (cycles)'/)
*     Write (*,603) (i, mctabl(i), mctabl(i), i = 0,127)
* 603 Format (/' Machine characteristics table:'//
*    1       (' ', I3, ':  ', I10, ' ', A8))
      lapms  = TIMEF ()
      cputim = SECOND ()
C
      Return
C
      Entry CPUEND
C
      cputim = SECOND() - cputim
      lapms  = TIMEF()  - lapms
      laphun = NINT (lapms*0.1)
      laphr  = laphun / (3600*100)
      laphun = laphun - (laphr*3600*100)
      lapmin = laphun / (60*100)
      laphun = laphun - (lapmin*60*100)
      lapsec = laphun / 100
      laphun = laphun - (lapsec*100)
      cpuhun = NINT (cputim*100)
      cpuhr  = cpuhun / (3600*100)
      cpuhun = cpuhun - (cpuhr*3600*100)
      cpumin = cpuhun / (60*100)
      cpuhun = cpuhun - (cpumin*60*100)
      cpusec = cpuhun / 100
      cpuhun = cpuhun - (cpusec*100)
      Write (*,602) laphr, lapmin, lapsec, laphun,
```

```
104      1              cpuhr, cpumin, cpusec, cpuhun
105    602 Format (/' ', T25, 'CRAY/UNICOS Execution Statistics'/
106      1         ' ', T25, '-------------------------------'//
107      2         ' ', T19, 'Elapsed: ',   I4, 2(':', I2.2), '.', I2.2,
108      3         '    CPU: ',    I4, 2(':', I2.2), '.', I2.2/)
109 C
110       Return
111 C
112       Entry PERFON
113 C
114       Entry PERFOF
115 C
116       Return
117 C
118       Entry GETTIM (time)
119 C
120 C...   SECOND is cumulative across all CPUs (including rendesvous overhead)
121 C      time = SECOND ()
122 C...   TSECND is unique to each CPU (task)
123       time = TSECND ()
124       Return
125 C
126       Entry PFAULT (nfault)
127 C
128       nfault = 0
129       Return
130 C
131       End
```

# APPENDIX B
# INCLUDE FILES

## B.1 WETC3D

### B.1.1 WETC3D_4.DAT

```
1       Parameter  (NX     =     25)
2       Parameter  (NY     =     70)
3       Parameter  (NZ     =     20)
4       Parameter  (NT     =    101)
5       Parameter  (NL     =      4)
6       Parameter  (NI     =      4)
7       Parameter  (IW     =      4)
8       Parameter  (NTIPF  =     10)
9       Parameter  (NBARF  =      1)
10      Parameter  (NTIPL  =     11)
11      Parameter  (NTIPR  =     50)
12      Parameter  (NBARR  =     31)
13      Parameter  (H      =   0.25)
14      Parameter  (C      =  0.025)
15      Parameter  (CF1    =  0.075)
16      Parameter  (PAC    =     2.)
17      Parameter  (BMU    =    0.5)
18      Parameter  (STIFF  =   200.)
19      Character  ARRAY*(*)
20      Parameter  (ARRAY  = 'wex160.dat')
```

### B.1.2 WETC3D_8.DAT

```
1       Parameter  (NX     =     25)
2       Parameter  (NY     =     70)
3       Parameter  (NZ     =     20)
4       Parameter  (NT     =    101)
5       Parameter  (NL     =      4)
6       Parameter  (NI     =      8)
7       Parameter  (IW     =      4)
8       Parameter  (NTIPF  =     10)
9       Parameter  (NBARF  =      1)
10      Parameter  (NTIPL  =     11)
11      Parameter  (NTIPR  =     50)
12      Parameter  (NBARR  =     31)
13      Parameter  (H      =   0.25)
14      Parameter  (C      =  0.025)
15      Parameter  (CF1    =  0.075)
16      Parameter  (PAC    =     4.)
17      Parameter  (BMU    =    0.1)
18      Parameter  (STIFF  =   200.)
19      Character  ARRAY*(*)
20      Parameter  (ARRAY  = 'wex070.dat')
```

### B.1.3 WETC3D_12.DAT

```
1       Parameter  (NX     =     25)
2       Parameter  (NY     =     70)
3       Parameter  (NZ     =     20)
4       Parameter  (NT     =    101)
5       Parameter  (NL     =      4)
6       Parameter  (NI     =     12)
7       Parameter  (IW     =      4)
8       Parameter  (NTIPF  =     10)
9       Parameter  (NBARF  =      1)
10      Parameter  (NTIPL  =     11)
11      Parameter  (NTIPR  =     50)
12      Parameter  (NBARR  =     31)
13      Parameter  (H      =   0.25)
14      Parameter  (C      =  0.025)
15      Parameter  (CF1    =  0.075)
16      Parameter  (PAC    =     6.)
17      Parameter  (BMU    = 0.0308)
18      Parameter  (STIFF  =   200.)
19      Character  ARRAY*(*)
20      Parameter  (ARRAY  = 'wex076.dat')
```

### B.1.4 WETC3D_20.DAT

```
1       Parameter  (NX     =     25)
2       Parameter  (NY     =     70)
3       Parameter  (NZ     =     20)
4       Parameter  (NT     =    101)
5       Parameter  (NL     =      4)
6       Parameter  (NI     =     20)
7       Parameter  (IW     =      4)
8       Parameter  (NTIPF  =     10)
9       Parameter  (NBARF  =      1)
10      Parameter  (NTIPL  =     11)
11      Parameter  (NTIPR  =     50)
12      Parameter  (NBARR  =     31)
13      Parameter  (H      =   0.25)
14      Parameter  (C      =  0.025)
15      Parameter  (CF1    =  0.075)
16      Parameter  (PAC    =    10.)
17      Parameter  (BMU    =  0.005)
18      Parameter  (STIFF  =   200.)
19      Character  ARRAY*(*)
20      Parameter  (ARRAY  = 'wex077.dat')
```

### B.1.5 WETC3D_32.DAT

```
1       Parameter  (NX     =     25)
2       Parameter  (NY     =     70)
3       Parameter  (NZ     =     20)
4       Parameter  (NT     =    101)
5       Parameter  (NL     =      4)
6       Parameter  (NI     =     32)
7       Parameter  (IW     =      4)
8       Parameter  (NTIPF  =     10)
9       Parameter  (NBARF  =      1)
10      Parameter  (NTIPL  =     11)
```

```
      Parameter  (NTIPR  =     50)
      Parameter  (NBARR  =     31)
      Parameter  (H      =   0.25)
      Parameter  (C      =  0.025)
      Parameter  (CF1    =  0.075)
      Parameter  (PAC    =    16.)
      Parameter  (BMU    =  0.001)
      Parameter  (STIFF  =   200.)
      Character  ARRAY*(*)
      Parameter  (ARRAY  = 'wex079.dat')
```

### B.1.6 WETC3D_40.DAT

```
      Parameter  (NX     =     25)
      Parameter  (NY     =     70)
      Parameter  (NZ     =     20)
      Parameter  (NT     =    101)
      Parameter  (NL     =      4)
      Parameter  (NI     =     40)
      Parameter  (IW     =      4)
      Parameter  (NTIPF  =     10)
      Parameter  (NBARF  =      1)
      Parameter  (NTIPL  =     11)
      Parameter  (NTIPR  =     50)
      Parameter  (NBARR  =     31)
      Parameter  (H      =   0.25)
      Parameter  (C      =  0.025)
      Parameter  (CF1    =  0.075)
      Parameter  (PAC    =    20.)
      Parameter  (BMU    =  0.001)
      Parameter  (STIFF  =   200.)
      Character  ARRAY*(*)
      Parameter  (ARRAY  = 'wex080.dat')
```

## B.2 3DMODEL4

### B.2.1 SMALL.INC

```
C...  Grid
      Integer    NX, NY, NZ
      Real       H
      Parameter  (NX     =    80)
      Parameter  (NY     =    80)
      Parameter  (NZ     =    40)
      Parameter  (H      =  200.)
C...  Time
      Integer    NT, IDT
      Real       DT
      Parameter  (NT     =    10)
      Parameter  (DT     =  0.03)
      Parameter  (IDT    =     1)
C...  Fault
      Integer    YLEN, ZLEN
      Parameter  (YLEN   =     0)
      Parameter  (ZLEN   =     0)
C...  Medium
      Integer    NCO
```

```
      Parameter  (NCO    = 65536)
      Character  SVELOC*(*), PVELOC*(*), DENSTY*(*)
      Parameter  (SVELOC = '2000.')
      Parameter  (PVELOC = '3500.')
      Parameter  (DENSTY = '2700.')
C...  Source/receivers
      Integer    XS, YS, ZS, SOURCE, ZG, SP
      Real       SVEL, PVEL, WIDTH, RAMP
      Parameter  (XS     =    55)
      Parameter  (YS     =    40)
      Parameter  (ZS     =    10)
      Parameter  (SOURCE =     2)
      Parameter  (SVEL   = 2000.)
      Parameter  (PVEL   = 3500.)
      Parameter  (WIDTH  =   7.5)
      Parameter  (RAMP   =    .7)
      Parameter  (ZG     =     1)
      Parameter  (SP     =     1)
C...  Options
      Integer    STYLE, FRESRF, RDSRC, NTEND, IVEL, OSTYLE, SRFMAX
      Integer    INNER, SPREAD, TSLICE, XSLICE, YSLICE
      Integer    NOFF, IOFFOX, IOFFOY, IDOFFX, IDOFFY
      Parameter  (STYLE  =     1)
      Parameter  (FRESRF =     1)
      Parameter  (RDSRC  =     2)
      Parameter  (NTEND  =   620)
      Parameter  (IVEL   =     1)
      Parameter  (OSTYLE =     0)
      Parameter  (SRFMAX =     0)
      Parameter  (INNER  =     0)
      Parameter  (SPREAD =     0)
      Parameter  (TSLICE =     0)
      Parameter  (XSLICE =    40)
      Parameter  (YSLICE =    40)
      Parameter  (NOFF   =    16)
      Parameter  (IOFFOX =     4)
      Parameter  (IOFFOY =    40)
      Parameter  (IDOFFX =     5)
      Parameter  (IDOFFY =     0)
C...  Files
      Character  SORCE1*(*), TOPOFI*(*), SMFILE*(*), SEISFI*(*)
      Character  SURFFI*(*), XYFILE*(*), XZFILE*(*), YZFILE*(*)
      Character  XFILE*(*),  YFILE*(*)
      Parameter  (SORCE1 = ' ')
      Parameter  (TOPOFI = ' ')
      Parameter  (SMFILE = ' ')
      Parameter  (SEISFI = 'small.out')
      Parameter  (SURFFI = 'surftest.dat')
      Parameter  (XYFILE = 'xytest.dat')
      Parameter  (XZFILE = 'xztest.dat')
      Parameter  (YZFILE = 'yztest.dat')
      Parameter  (XFILE  = 'xtest.dat')
      Parameter  (YFILE  = 'ytest.dat')
```

## B.2.2 LARGE.INC

```
C...  Grid
      Integer    NX, NY, NZ
      Real       H
      Parameter  (NX     =   220)
      Parameter  (NY     =   300)
      Parameter  (NZ     =    60)
      Parameter  (H      =  200.)
C...  Time
      Integer    NT, IDT
      Real       DT
      Parameter  (NT     =    10)
      Parameter  (DT     =  0.03)
      Parameter  (IDT    =     1)
C...  Fault
      Integer    YLEN, ZLEN
      Parameter  (YLEN   =     0)
      Parameter  (ZLEN   =     0)
C...  Medium
      Integer    NCO
      Parameter  (NCO    = 65536)
      Character  SVELOC*(*), PVELOC*(*), DENSTY*(*)
      Parameter  (SVELOC = '2000.')
      Parameter  (PVELOC = '3500.')
      Parameter  (DENSTY = '2700.')
C...  Source/receivers
      Integer    XS, YS, ZS, SOURCE, ZG, SP
      Real       SVEL, PVEL, WIDTH, RAMP
      Parameter  (XS     =    55)
      Parameter  (YS     =    40)
      Parameter  (ZS     =    10)
      Parameter  (SOURCE =     2)
      Parameter  (SVEL   = 2000.)
      Parameter  (PVEL   = 3500.)
      Parameter  (WIDTH  =   7.5)
      Parameter  (RAMP   =    .7)
      Parameter  (ZG     =     1)
      Parameter  (SP     =     1)
C...  Options
      Integer    STYLE, FRESRF, RDSRC, NTEND, IVEL, OSTYLE, SRFMAX
      Integer    INNER, SPREAD, TSLICE, XSLICE, YSLICE
      Integer    NOFF, IOFFOX, IOFFOY, IDOFFX, IDOFFY
      Parameter  (STYLE  =     1)
      Parameter  (FRESRF =     1)
      Parameter  (RDSRC  =     2)
      Parameter  (NTEND  =   620)
      Parameter  (IVEL   =     1)
      Parameter  (OSTYLE =     0)
      Parameter  (SRFMAX =     0)
      Parameter  (INNER  =     0)
      Parameter  (SPREAD =     0)
      Parameter  (TSLICE =     0)
      Parameter  (XSLICE =    40)
      Parameter  (YSLICE =    40)
      Parameter  (NOFF   =    16)
      Parameter  (IOFFOX =     4)
```

```
      Parameter  (IOFFOY =    40)
      Parameter  (IDOFFX =     5)
      Parameter  (IDOFFY =     0)
C...  Files
      Character  SORCE1*(*), TOPOFI*(*), SMFILE*(*), SEISFI*(*)
      Character  SURFFI*(*), XYFILE*(*), XZFILE*(*), YZFILE*(*)
      Character  XFILE*(*),  YFILE*(*)
      Parameter  (SORCE1 = ' ')
      Parameter  (TOPOFI = ' ')
      Parameter  (SMFILE = ' ')
      Parameter  (SEISFI = 'large.out')
      Parameter  (SURFFI = 'surftest.dat')
      Parameter  (XYFILE = 'xytest.dat')
      Parameter  (XZFILE = 'xztest.dat')
      Parameter  (YZFILE = 'yztest.dat')
      Parameter  (XFILE  = 'xtest.dat')
      Parameter  (YFILE  = 'ytest.dat')
```

# APPENDIX C
# SAMPLE RUNS

## C.1 WETC3D

The following is a listing of the batch log file generated by the problem specified in WETC3D_4.DAT, run on a VAX 4000 model 300. (Since the batch log file also contains records written to Sys$Output, there is no need to redirect Sys$Output to a separate file.)

```
$ SET NOVERIFY
$ Set Default [.Temp]
$
$ Copy    WetC3D_4.dat WetC3D.dat
$ Fortran Wetc3D+GetHyd+GetSpl+CPUSubs.vms
$ Link    WetC3D
$
$ Run     WetC3D

Output data is in GD:[HBUNDOCK.TEMP]WEX160.DAT;1

VAX/VMS V5.4-1     Host: ISDMNL            CPU type: 11    17-JUN-1991 11:05:05.96

Working Set (pages):  Maximum Authorized:           4096
                      Maximum Authorized Extent:   16400


   WetC3d Parameters
----------------------

NX      =          25
NY      =          70
NZ      =          20
NT      =         101
NL      =           4
NI      =           4
IW      =           4
NTIPF   =          10
NTIPL   =          11
NTIPR   =          50
NBARF   =           1
NBARR   =          31
H       = 0.25000000
C       = 2.50000004E-02
CF1     = 7.50000030E-02
PAC     =  2.0000000
BMU     = 0.50000000
STIFF   =  200.00000
ARRAY   = 'wex160.dat'
```

```
40
41 Paging flag = F (    0:    0, non-paging:paging)
42
43 NK =   26
44
45
46     Floating-Point Operations Per Time Step
47     ---------------------------------------
48
49 Elastic solid   1812131     723487  Velocities
50                           1084204  Stresses
51                              4440  Displacement
52 Gridding          66583
53 Fluid flow       154131
54 Total           2032845
55
56
57                          WetC3D Execution Summary
58                          ------------------------
59
60                        Time        CPU                    CPU Secs/    Useful
61 Execution Phase        Steps     Minutes     Percent      Time Step    MFLOPS
62 ------------------------------------------------------------------------------
63 Elastic solid            100       1.389         89%        0.833     2.175
64 Gridding                 100       0.064          4%        0.038     1.743
65 Fluid flow               100       0.112          7%        0.067     2.304
66 Total                    100       1.564        100%        0.938     2.167
67
68
69                        VAX/VMS Execution Statistics
70                        ----------------------------
71
72 ELAPSED:    0 00:01:49.78  CPU: 0:01:34.27  BUFIO: 5  DIRIO: 16  FAULTS: 9433
73
74 FORTRAN STOP
75 $
76   HBUNDOCK     job terminated at 17-JUN-1991 11:06:56.18
77
78   Accounting information:
79   Buffered I/O count:             181        Peak working set size:    3013
80   Direct I/O count:               201        Peak page file size:      6197
81   Page faults:                  13168        Mounted volumes:             0
82   Charged CPU time:       0 00:01:44.26      Elapsed time:     0 00:02:07.88
```

## C.2 3DMODEL4

The following is a listing of the batch log file generated by the problem specified in SMALL.INC, run on a VAX 4000 model 300. (Since the batch log file also contains records written to Sys$Output, there is no need to redirect Sys$Output to a separate file.)

```
$ SET NOVERIFY
$ Set Default [.Temp]
$
$ Copy    Small.inc 3DModel4.inc
$ Fortran 3DModel4+Choose+Inner4+Inner2+Absorb+CPUSubs.vms
$ Link    3DModel4
$
$ Run     3DModel4

VAX/VMS V5.4-1     Host: ISDMNL            CPU type: 11    17-JUN-1991 14:24:13.47

Working Set (pages):  Maximum Authorized:          4096
                      Maximum Authorized Extent:   16400


  3DModel4 Parameters
-----------------------

NX      =          80
NY      =          80
NZ      =          40
H       =  200.00000
NT      =          10
DT      = 2.99999993E-02
IDT     =           1
YLEN    =           0
ZLEN    =           0
NCO     =       65536
SVELOC = '2000.'
PVELOC = '3500.'
DENSTY = '2700.'
XS      =          55
YS      =          40
ZS      =          10
SOURCE =           2
SVEL    =  2000.0000
PVEL    =  3500.0000
WIDTH   =  7.5000000
RAMP    = 0.69999999
ZG      =           1
SP      =           1
STYLE   =           1
FRESRF  =           1
RDSRC   =           2
NTEND   =         620
IVEL    =           1
OSTYLE  =           0
SRFMAX  =           0
INNER   =           0
SPREAD  =           0
TSLICE  =           0
XSLICE  =          40
YSLICE  =          40
NOFF    =          16
IOFFOX  =           4
IOFFOY  =          40
```

```
 57 IDOFFX =           5
 58 IDOFFY =           0
 59 SORCE1 = ' '
 60 TOPOFI = ' '
 61 SMFILE = ' '
 62 SEISFI = 'small.out'
 63 SURFFI = 'surftest.dat'
 64 XYFILE = 'xytest.dat'
 65 XZFILE = 'xztest.dat'
 66 YZFILE = 'yztest.dat'
 67 XFILE  = 'xtest.dat'
 68 YFILE  = 'ytest.dat'
 69 dt=    0.030 h=  200.000 vpmax= 3500.000 vsmax= 2000.000
 70 fac= 0.86, stable
 71
 72 len4o=      1 (       135 entries)
 73 len2o=      1 (        45 entries)
 74 lena =      1 (         2 entries)
 75 Total=      3 (       182 entries)
 76
 77 Choosing fastest versions of INNER4/INNER2:
 78
 79 INNR4S/INNR2S took   29.97
 80 INR4V1/INR2V1 took   47.58
 81 INR4V2/INR2V2 took   39.47
 82 INR4V3/INR2V3 took   36.65
 83 INR4V4/INR2V4 took   35.65
 84 INR4V5/INR2V4 took   35.05
 85 INR4V6/INR2V4 took   33.85
 86 INR4V7/INR2V5 took   34.12
 87 INR4V8/INR2V6 took   64.53
 88
 89 INNR4S/INNR2S chosen.
 90
 91 Initialization took  488.32
 92
 93 Step     2 took   29.93
 94 Step     3 took   29.91
 95 Step     4 took   30.78
 96 Step     5 took   31.11
 97 Step     6 took   29.55
 98 Step     7 took   28.46
 99 Step     8 took   28.75
100 Step     9 took   28.45
101
102
103     Floating-Point Operations Per Time Step
104     ---------------------------------------
105
106     Fourth-order         58437015
107     Second-order          2248740
108     Boundary               219024
109     Total                60904779
110
111
112                          3DModel4 Execution Summary
```

```
                          ------------------------

                         Time        CPU                    CPU Secs/    Useful
Execution Phase          Steps     Minutes     Percent     Time Step    MFLOPS
-------------------------------------------------------------------------------
Fourth-order                 8       3.746         95%        28.091     2.080
Second-order                 8       0.166          4%         1.247     1.803
Boundary                     8       0.031          1%         0.230     0.952
Total                        8       3.942        100%        29.569     2.060


                         VAX/VMS Execution Statistics
                         ----------------------------

ELAPSED:    0 00:24:15.20  CPU: 0:12:05.34  BUFIO: 17  DIRIO: 40  FAULTS: 47887

FORTRAN STOP
$
  HBUNDOCK     job terminated at 17-JUN-1991 14:48:33.33

  Accounting information:
  Buffered I/O count:              364         Peak working set size:    15186
  Direct I/O count:                540         Peak page file size:      18997
  Page faults:                   51516         Mounted volumes:              0
  Charged CPU time:          0 00:12:42.63     Elapsed time:      0 00:26:25.82
```