UNIVERSITY OF HELSINKI
DEPARTMENT OF PHYSICS

TOOLS FOR HIGH PERFORMANCE COMPUTING

Exercise 3

Student: Caike Crepaldi

Professor: Antti Kuronen

September 2015

Note

This exercise was done using a ssh remote login through my pangolin (pangolin.it.helsinki.fi) university linux account while using my macbook pro with OS X. To do such a thing, the following command was run in the OS X Yosemite terminal.

```
Macbook-Pro:~crepaldi $ ssh username@pangolin.it.helsinki.fi
```

The editor I used to edit files and write codes was **emacs**. All files (and Makefiles, when applicable) created by the student and used in this exercise were packed together with this PDF documentation and shall be available to the assistant for further analysis in its own problem folder.

Problem 1

The Fortran source code of the program created for the first problem (calculation of Ackermann recursive function) is shown bellow.

ex3p1.f90

```
1  program ackermann
2    implicit none
3    integer :: m,n,ack
4
5    write(6,*) "Insert a value for the integer m:"
6    read(5,*) m
7    write(6,*) "Insert a value for the integer n:"
8    read(5,*) n
9
10   write(*,*) "A(m,n) =", ack(m,n)
11
12 end program ackermann
13
14 recursive function ack(m,n) result(a)
15   integer, intent(in) :: m,n ! variable can enter but cannot be changed
16   integer :: a
17   if (m==0) then
18      a=n+1
19   else if (n==0) then
20      a=ack(m-1,1)
```

```
21    else
22        a=ack(m-1,ack(m,n-1))
23    end if
24 end function ack
```

We can compile the function with the command ***gfortran*** *-g -O0 ex3p1.f90* and run it in the **gdb** debugger.

Command 1

```
username@tktl-pangolin:~/Tools-for-HPC/exercise-3/$ gfortran -g -O0 ex3p1.f90
username@tktl-pangolin:~/Tools-for-HPC/exercise-3/$ gdb a.out
(gdb) run
 Insert a value for the integer m:
4
 Insert a value for the integer n:
1
^C
Program received signal SIGINT, Interrupt.
0x0000000000400923 in ack (m=0, n=4421) at ex3p1.f90:17
17          if (m==0) then
(gdb) where
#0  0x0000000000400923 in ack (m=0, n=4421) at ex3p1.f90:17
#1  0x00000000004009a2 in ack (m=1, n=4420) at ex3p1.f90:22
#2  0x000000000040098c in ack (m=1, n=4421) at ex3p1.f90:22
#3  0x000000000040098c in ack (m=1, n=4422) at ex3p1.f90:22
#4  0x000000000040098c in ack (m=1, n=4423) at ex3p1.f90:22
#5  0x000000000040098c in ack (m=1, n=4424) at ex3p1.f90:22
#6  0x000000000040098c in ack (m=1, n=4425) at ex3p1.f90:22
#7  0x000000000040098c in ack (m=1, n=4426) at ex3p1.f90:22
#8  0x000000000040098c in ack (m=1, n=4427) at ex3p1.f90:22
#9  0x000000000040098c in ack (m=1, n=4428) at ex3p1.f90:22
#10 0x000000000040098c in ack (m=1, n=4429) at ex3p1.f90:22
#11 0x000000000040098c in ack (m=1, n=4430) at ex3p1.f90:22
#12 0x000000000040098c in ack (m=1, n=4431) at ex3p1.f90:22
#13 0x000000000040098c in ack (m=1, n=4432) at ex3p1.f90:22
#14 0x000000000040098c in ack (m=1, n=4433) at ex3p1.f90:22
#15 0x000000000040098c in ack (m=1, n=4434) at ex3p1.f90:22
#16 0x000000000040098c in ack (m=1, n=4435) at ex3p1.f90:22
#17 0x000000000040098c in ack (m=1, n=4436) at ex3p1.f90:22
#18 0x000000000040098c in ack (m=1, n=4437) at ex3p1.f90:22
#19 0x000000000040098c in ack (m=1, n=4438) at ex3p1.f90:22
#20 0x000000000040098c in ack (m=1, n=4439) at ex3p1.f90:22
#21 0x000000000040098c in ack (m=1, n=4440) at ex3p1.f90:22
#22 0x000000000040098c in ack (m=1, n=4441) at ex3p1.f90:22
---Type <return> to continue, or q <return> to quit---
```

We can see that the values of the integers growing thanks to the recursive function. It takes some time until the program completly evaluates A(4,1). Also, we can see the program spending his execution time evaluating the variable a in the if-else control blocks.

Problem 2

See the commands bellow.

Command 2

```
username@tktl-pangolin:~/Tools-for-HPC/exercise-3/mmc/src$ make compiler=gnuprof
gfortran -p -g -ffree-line-length-none -c defs.f90
gfortran -p -g -ffree-line-length-none -c latticedata.f90
gfortran -p -g -ffree-line-length-none -c calcdr.f90
gfortran -p -g -ffree-line-length-none -c potentialparameters.f90
gfortran -p -g -ffree-line-length-none -c interfaces.f90
gfortran -p -g -ffree-line-length-none -c properties.f90
gfortran -p -g -ffree-line-length-none -c calcgr.f90
```

```
gfortran -p -g -ffree-line-length-none -c mtfort90.f90
gfortran -p -g -ffree-line-length-none -c eam_modules.f90
gfortran -p -g -ffree-line-length-none -c createlattice.f90
gfortran -p -g -ffree-line-length-none -c displace.f90
gfortran -p -g -ffree-line-length-none -c nlist.f90
gfortran -p -g -ffree-line-length-none -c energy.f90
gfortran -p -g -ffree-line-length-none -c getarguments.f90
gfortran -p -g -ffree-line-length-none -c graphicssubs.f90
gfortran -p -g -ffree-line-length-none -c miscsubs.f90
gfortran -p -g -ffree-line-length-none -c mmc.f90
gfortran -p -g -ffree-line-length-none -c nlistsubs.f90
gfortran -p -g -ffree-line-length-none -c prdist.f90
gfortran -p -g -ffree-line-length-none -c readxyz.f90
gfortran -p -g -ffree-line-length-none -c SANDIAeam_eamal.f90
gfortran -p -g -ffree-line-length-none -c potential_interface.f90
gfortran -p -g -ffree-line-length-none -c EAMforces_eamal.f90
gfortran -p -g -ffree-line-length-none -c md.f90
gfortran -p -g -ffree-line-length-none -c tersoff_compound.f90
gfortran -p -g -ffree-line-length-none -c init_tersoff.f90
gfortran -p -g -ffree-line-length-none -c splinereppot.f90
gfortran -p -g -o mmc calcdr.o calcgr.o createlattice.o defs.o displace.o energy.o getarguments.
    o graphicssubs.o interfaces.o latticedata.o miscsubs.o mmc.o mtfort90.o nlist.o nlistsubs.o
    potentialparameters.o prdist.o properties.o readxyz.o eam_modules.o SANDIAeam_eamal.o
    potential_interface.o EAMforces_eamal.o md.o tersoff_compound.o init_tersoff.o splinereppot.
    o
```

```
username@tktl-pangolin:~/Tools-for-HPC/exercise-3/mmc/src$ cd ..
username@tktl-pangolin:~/Tools-for-HPC/exercise-3/mmc$ cd run
username@tktl-pangolin:~/Tools-for-HPC/exercise-3/mmc/run$ ../src/mmc -maxtype 2 -element Fe Cr -eamnbands 2
-bh -xyzinfile mdlat.in.xyz -maxmc 300 -tprob1 0.5 -tprob2 0.5 -exactconc -iswap 20
```

(skipping most of the program's output)

```
Atomic moves attempted and accepted:       16200        6885
Volume moves attempted and accepted:           0           0
Swap   moves attempted and accepted:         405          10

CPU time (sec total,sec/atom/step)      35.64405800         0.2200250494E-02

username@tktl-pangolin:~/Tools-for-HPC/exercise-3/mmc/run$ gprof -l ../src/mmc

Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  5.50      1.97     1.97                             calc_p_ (EAMforces_eamal.f90:69 @ 4420eb)
  4.73      3.66     1.69                             calc_force_ (EAMforces_eamal.f90:407 @
     4441a1)
  4.67      5.33     1.67                             calc_p_ (EAMforces_eamal.f90:88 @ 4423c3)
  3.66      6.64     1.31                             calc_p_ (EAMforces_eamal.f90:85 @ 442331)
  3.55      7.91     1.27                             calc_p_ (EAMforces_eamal.f90:68 @ 4420cb)
  2.96      8.97     1.06                             calc_p_ (EAMforces_eamal.f90:70 @ 44211c)
  2.80      9.97     1.00                             calc_p_ (EAMforces_eamal.f90:110 @ 4426ae)
  2.71     10.94     0.97                             calc_p_ (EAMforces_eamal.f90:74 @ 442147)
  2.07     11.68     0.74                             calc_p_ (EAMforces_eamal.f90:102 @ 4424e2)
  2.04     12.41     0.73                             calc_p_ (EAMforces_eamal.f90:87 @ 44239e)
  1.99     13.12     0.71                             calc_force_ (EAMforces_eamal.f90:424 @
     4444a9)
  1.99     13.83     0.71                             calc_p_ (EAMforces_eamal.f90:104 @ 4424fa)
  1.85     14.49     0.66                             calc_p_ (EAMforces_eamal.f90:105 @ 442567)
  1.85     15.15     0.66                             calc_p_ (EAMforces_eamal.f90:101 @ 4424ca)
  1.76     15.78     0.63                             calc_p_ (EAMforces_eamal.f90:64 @ 442064)
  1.76     16.41     0.63                             calc_p_ (EAMforces_eamal.f90:100 @ 4424b0)
  1.71     17.02     0.61                             calc_p_ (EAMforces_eamal.f90:67 @ 4420ab)
  1.68     17.62     0.60                             calc_p_ (EAMforces_eamal.f90:114 @ 442752)
  1.65     18.21     0.59                             calc_p_ (EAMforces_eamal.f90:97 @ 442455)
  1.54     18.76     0.55                             calc_p_ (EAMforces_eamal.f90:99 @ 442476)
  1.52     19.30     0.55                             calc_force_ (EAMforces_eamal.f90:382 @ 443
     ea8)
```

```
  1.51     19.84     0.54                             calc_force_ (EAMforces_eamal.f90:402 @
     44411c)
  1.48     20.37     0.53                             calc_p_ (EAMforces_eamal.f90:96 @ 44241b)
  1.45     20.89     0.52                             calc_force_ (EAMforces_eamal.f90:380 @ 443
     dce)
  1.45     21.41     0.52                             calc_p_ (EAMforces_eamal.f90:108 @ 442641)
  1.43     21.92     0.51                             calc_p_ (EAMforces_eamal.f90:98 @ 442467)
  1.37     22.41     0.49                             calc_p_ (EAMforces_eamal.f90:107 @ 4425d4)
  1.34     22.89     0.48                             calc_force_ (EAMforces_eamal.f90:432 @
     4445c8)
  1.26     23.34     0.45                             calc_force_ (EAMforces_eamal.f90:400 @
     444042)
  1.23     23.78     0.44                             calc_p_ (EAMforces_eamal.f90:82 @ 4422c9)
  1.20     24.21     0.43                             calc_force_ (EAMforces_eamal.f90:425 @
     444548)
  1.17     24.63     0.42                             calc_p_ (EAMforces_eamal.f90:79 @ 44222c)
  1.12     25.03     0.40                             calc_p_ (EAMforces_eamal.f90:83 @ 4422fb)
  1.12     25.43     0.40                             calc_p_ (EAMforces_eamal.f90:76 @ 4421bf)
  1.03     25.80     0.37                             calc_force_ (EAMforces_eamal.f90:348 @ 443
     c4f)
  1.01     26.16     0.36                             calc_force_ (EAMforces_eamal.f90:401 @
     4440af)
```

(skipping part of the output)

```
                     Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) for 0.03% of 35.76 seconds

index % time    self  children    called     name
                0.00    0.00  877716/877716      createvlist_ (nlistsubs.f90:190 @ 42b404) [582]
[153]    0.0    0.00    0.00  877716         dists_ (miscsubs.f90:52 @ 41e1a6) [153]
-----------------------------------------------
                0.00    0.00       1/856167      mmc (mmc.f90:132 @ 42204a) [4263]
                0.00    0.00  856166/856167      createvlist_ (nlistsubs.f90:192 @ 42b54e) [584]
[154]    0.0    0.00    0.00  856167         createmaps_ (nlistsubs.f90:48 @ 42ab5a) [154]
-----------------------------------------------
                0.00    0.00   33244/33244       eam_energy_ (potential_interface.f90:220 @ 4417
                   c4) [973]
[155]    0.0    0.00    0.00   33244         calc_force_ (EAMforces_eamal.f90:245 @ 4435bd)
    [155]
-----------------------------------------------
                0.00    0.00   33244/33244       eam_energy_ (potential_interface.f90:219 @
                   441793) [972]
[156]    0.0    0.00    0.00   33244         calc_fp_ (EAMforces_eamal.f90:158 @ 442bdd) [156]
-----------------------------------------------
                0.00    0.00   33244/33244       eam_energy_ (potential_interface.f90:218 @
                   44174e) [971]
[157]    0.0    0.00    0.00   33244         calc_p_ (EAMforces_eamal.f90:10 @ 441f0b) [157]
-----------------------------------------------
                0.00    0.00   33244/33244       energy_ (energy.f90:42 @ 409856) [1010]
[158]    0.0    0.00    0.00   33244         eam_energy_ (potential_interface.f90:115 @ 4406c9)
    [158]
                0.00    0.00   33244/132942      __defs_MOD_dbf (defs.f90:82 @ 40717a) [5636]
-----------------------------------------------
                0.00    0.00       1/33244       mmc (mmc.f90:182 @ 4221bc) [4289]
                0.00    0.00       3/33244       mmc (mmc.f90:524 @ 426bc6) [4487]
                0.00    0.00      30/33244       mmc (mmc.f90:500 @ 426674) [4473]
                0.00    0.00     405/33244       mmc (mmc.f90:311 @ 424183) [4372]
                0.00    0.00     405/33244       mmc (mmc.f90:337 @ 4248f4) [4388]
                0.00    0.00   16200/33244       mmc (mmc.f90:420 @ 4255d8) [4428]
                0.00    0.00   16200/33244       mmc (mmc.f90:423 @ 4257f5) [4431]
[159]    0.0    0.00    0.00   33244         energy_ (energy.f90:3 @ 409699) [159]
-----------------------------------------------
```

(skipping part of the output)

```
username@tktl-pangolin:~/Tools-for-HPC/exercise-3/mmc/run$ gprof ../src/mmc
```

Flat profile:

Each sample counts as 0.01 seconds.

% time	cumulative seconds	self seconds	calls	self s/call	total s/call	name
61.58	22.02	22.02	33244	0.00	0.00	calc_p_
36.47	35.06	13.04	33244	0.00	0.00	calc_force_
1.03	35.43	0.37	33244	0.00	0.00	calc_fp_
0.39	35.57	0.14	33244	0.00	0.00	eam_energy_
0.25	35.66	0.09	33210	0.00	0.00	sort_
0.06	35.68	0.02	877716	0.00	0.00	dists_
0.06	35.70	0.02	77981	0.00	0.00	__mtmod_MOD_grnd
0.06	35.72	0.02	6	0.00	0.00	spl2b2_y4d_
0.03	35.73	0.01	25515	0.00	0.00	displace_
0.03	35.74	0.01	301	0.00	0.00	createvlist_
0.03	35.75	0.01	4	0.00	0.01	init_pot_
0.03	35.76	0.01	4	0.00	0.00	spl1b2_
0.00	35.76	0.00	856167	0.00	0.00	createmaps_
0.00	35.76	0.00	132942	0.00	0.00	__defs_MOD_dbf
0.00	35.76	0.00	95001	0.00	0.00	__mtmod_MOD_igrnd
0.00	35.76	0.00	33244	0.00	0.00	energy_
0.00	35.76	0.00	1610	0.00	0.00	l2i.1880
0.00	35.76	0.00	405	0.00	0.00	print_swap_debug_info.1876
0.00	35.76	0.00	333	0.00	0.00	word_next_read_
0.00	35.76	0.00	301	0.00	0.00	createllist_
0.00	35.76	0.00	135	0.00	0.00	upcase_
0.00	35.76	0.00	55	0.00	0.00	getwords_
0.00	35.76	0.00	54	0.00	0.00	prgr_
0.00	35.76	0.00	31	0.00	0.00	printout_
0.00	35.76	0.00	28	0.00	0.00	allocatenlist_
0.00	35.76	0.00	27	0.00	0.00	icc_
0.00	35.76	0.00	24	0.00	0.00	dcc_
0.00	35.76	0.00	15	0.00	0.00	adjustdr_
0.00	35.76	0.00	4	0.00	0.00	spl2b2_y3d_
0.00	35.76	0.00	4	0.00	0.00	xyzdump_
0.00	35.76	0.00	1	0.00	35.76	MAIN__
0.00	35.76	0.00	1	0.00	0.00	__mtmod_MOD_sgrnd
0.00	35.76	0.00	1	0.00	0.00	getarguments_
0.00	35.76	0.00	1	0.00	0.00	getmasses_
0.00	35.76	0.00	1	0.00	0.04	initialize_potential_
0.00	35.76	0.00	1	0.00	0.00	printinfo_
0.00	35.76	0.00	1	0.00	0.00	readxyz_
0.00	35.76	0.00	1	0.00	0.00	toomany_

% time — the percentage of the total running time of the program used by this function.

cumulative seconds — a running sum of the number of seconds accounted for by this function and those listed above it.

self seconds — the number of seconds accounted for by this function alone. This is the major sort for this listing.

calls — the number of times this function was invoked, if this function is profiled, else blank.

self ms/call — the average number of milliseconds spent in this function per call, if this function is profiled, else blank.

total ms/call — the average number of milliseconds spent in this function and its descendents per call, if this function is profiled, else blank.

name — the name of the function. This is the minor sort for this listing. The index shows the location of the function in the gprof listing. If the index is in parenthesis it shows where it would appear in

```
          the gprof listing if it were to be printed.


Copyright (C) 2012 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.


                     Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) for 0.03% of 35.76 seconds

index % time    self  children    called     name
                                        1             MAIN__ [1]
                0.00   35.76       1/1           main [2]
[1]    100.0    0.00   35.76       1+1       MAIN__ [1]
                0.00   35.66   33244/33244       energy_ [4]
                0.00    0.04       1/1           initialize_potential_ [10]
                0.01    0.02     301/301         createvlist_ [11]
                0.01    0.01   25515/25515       displace_ [12]
                0.00    0.00   16210/77981       __mtmod_MOD_igrnd <cycle 1> [71]
                0.00    0.00   12361/77981       __mtmod_MOD_grnd <cycle 1> [16]
                0.00    0.00    1610/1610        l2i.1880 [18]
                0.00    0.00     405/405         print_swap_debug_info.1876 [20]
                0.00    0.00      31/31          printout_ [26]
                0.00    0.00      15/15          adjustdr_ [30]
                0.00    0.00       4/4           xyzdump_ [32]
                0.00    0.00       1/1           getarguments_ [33]
                0.00    0.00       1/1           __mtmod_MOD_sgrnd [72]
                0.00    0.00       1/1           readxyz_ [36]
                0.00    0.00       1/28          allocatenlist_ [27]
                0.00    0.00       1/856167      createmaps_ [19]
                                   1             MAIN__ [1]
-----------------------------------------------
                                                 <spontaneous>
[2]    100.0    0.00   35.76                 main [2]
                0.00   35.76       1/1           MAIN__ [1]
-----------------------------------------------
                0.14   35.52   33244/33244       energy_ [4]
[3]     99.7    0.14   35.52   33244         eam_energy_ [3]
               22.02    0.00   33244/33244       calc_p_ [5]
               13.04    0.00   33244/33244       calc_force_ [6]
                0.37    0.00   33244/33244       calc_fp_ [7]
                0.09    0.00   33210/33210       sort_ [8]
                0.00    0.00  132942/132942      __defs_MOD_dbf [70]
-----------------------------------------------
                0.00   35.66   33244/33244       MAIN__ [1]
[4]     99.7    0.00   35.66   33244         energy_ [4]
                0.14   35.52   33244/33244       eam_energy_ [3]
-----------------------------------------------
               22.02    0.00   33244/33244       eam_energy_ [3]
[5]     61.6   22.02    0.00   33244         calc_p_ [5]
-----------------------------------------------
               13.04    0.00   33244/33244       eam_energy_ [3]
[6]     36.5   13.04    0.00   33244         calc_force_ [6]
-----------------------------------------------
                0.37    0.00   33244/33244       eam_energy_ [3]
[7]      1.0    0.37    0.00   33244         calc_fp_ [7]
-----------------------------------------------
                0.09    0.00   33210/33210       eam_energy_ [3]
[8]      0.3    0.09    0.00   33210         sort_ [8]
-----------------------------------------------
                0.01    0.03       4/4           initialize_potential_ [10]
[9]      0.1    0.01    0.03       4         init_pot_ [9]
                0.02    0.00       6/6           spl2b2_y4d_ [15]
                0.01    0.00       4/4           spl1b2_ [17]
```

```
                0.00    0.00       4/4           spl2b2_y3d_ [31]
-----------------------------------------------------------------

                0.00    0.04       1/1           MAIN__ [1]
[10]     0.1    0.00    0.04       1         initialize_potential_ [10]
                0.01    0.03       4/4           init_pot_ [9]
-----------------------------------------------------------------
```

(skipping part of the output)

(a) calc_p_ (EAMforces_eamal.f90:69 @ 4420eb) → Most of the time is spent in line 69 (inside of function/subroutine calc_p) in the file EAMforces_eamal.f90.

(b) Most of the time is spent in the function/subroutine **energy**.

Problem 3

To do such task, I modified the ex3p3.f90 file in order to calculate the cputime spent in the do-loop. See the code below.

ex3p3.f90

```
1  program ex3p3
2    implicit none
3    integer,parameter :: N=15000
4    integer :: a(N,N)
5    integer :: i,j
6    real(kind=10) :: t1,t2
7
8    call cpu_time(t1)! Begin measurement
9    do j=1,N
10      do i=1,N
11          a(i,j)=(i+j)/2
12      end do
13   end do
14   call cpu_time(t2)! End measurement
15   print *, "Time spent =", t2-t1
16   !print '(10i8)',a(1:N:10000,1:N:10000)
17
18 end program ex3p3
```

See the commands below for the compilation command and CPUtime outputs.

Command 3

```
username@tktl-pangolin:~/Tools-for-HPC/exercise-3/$ gfortran -O0 -o ex3p3_0 ex3p3.f90
username@tktl-pangolin:~/Tools-for-HPC/exercise-3/$ gfortran -O1 -o ex3p3_1 ex3p3.f90
username@tktl-pangolin:~/Tools-for-HPC/exercise-3/$ gfortran -O2 -o ex3p3_2 ex3p3.f90
username@tktl-pangolin:~/Tools-for-HPC/exercise-3/$ gfortran -O3 -o ex3p3_3 ex3p3.f90
username@tktl-pangolin:~/Tools-for-HPC/exercise-3/$ ./ex3p3_0
Time spent = 1.33183300000000000003
username@tktl-pangolin:~/Tools-for-HPC/exercise-3/$ ./ex3p3_1
Time spent = 0.642623000000000000047
username@tktl-pangolin:~/Tools-for-HPC/exercise-3/$ ./ex3p3_2
Time spent = 0.452621999999999999998
username@tktl-pangolin:~/Tools-for-HPC/exercise-3/$ ./ex3p3_3
Time spent = 0.383699000000000000027
```

Problem 4

The source code for the program is shown below.

ex3p4.f90

```
1  program precision_speed
2    implicit none
3    integer :: n
4
5    write(*,*) "Please insert a value for the size of the array:"
6    read(*,*) n
7
8    call arrays(n)
9
10 end program precision_speed
11
12 subroutine arrays (n)
13   implicit none
14   integer :: i,j
15   integer, intent(in) :: n
16   integer, parameter :: sp = selected_real_kind(5,10) ! single precision
17   integer, parameter :: dp = selected_real_kind(10,40) ! double precision
18   integer, parameter :: qp = selected_real_kind(20,1000) ! quadruple precision
19   real(sp), dimension (n) :: a
20   real(dp), dimension(n) :: b
21   real(qp), dimension(n) :: c
22   real :: sum
23   real :: t1,t2,t3,t4,t5,t6,tcpu
24
25   call cpu_time(t1)
26
27   sum = 0.0
28   do i=1,(n-1)
29      do j=(i+1),n
30         sum = sum + (a(j)-a(i))
31      end do
32   end do
33
34   call cpu_time(t2)
35   tcpu = t2-t1
36   write(*,*) "CPUtime for single-precision array =", tcpu
37
38   call cpu_time(t3)
39   sum =0.0
40   do i=1,(n-1)
41      do j=(i+1),n
42         sum = sum + (b(j)-b(i))
43      end do
44        end do
45   call cpu_time(t4)
46   tcpu = t4-t3
47   write(*,*) "CPUtime for double-precision array =", tcpu
48
49   call cpu_time(t5)
50   sum =0.0
51   do i=1,(n-1)
52      do j=(1+1),n
```

```
53            sum = sum + (c(j)-c(i))
54          end do
55            end do
56       call cpu_time(t6)
57       tcpu = t6-t5
58       write(*,*) "CPUtime for quadruple-precision array =", tcpu
59
60    end subroutine arrays
```

The program above creates 3 arrays (single, double and quadruple precision) in function of the integer n (size of the array) chosen by the user and then does the calculations. In the end of each double do-loop, the program prints the CPUtime spent by the calculation using each real kind.

See the commands below for the compilation command and CPUtime outputs.

Command 4

```
username@tktl-pangolin:~/Tools-for-HPC/exercise-3/$ gfortran ex3p4.f90
username@tktl-pangolin:~/Tools-for-HPC/exercise-3/$ ./a.out
 Please insert a value for the size of the array:
10
 CPUtime for single-precision array =   3.00002284E-06
 CPUtime for double-precision array =   1.00000761E-06
 CPUtime for quadruple-precision array =   4.40001022E-05
username@tktl-pangolin:~/Tools-for-HPC/exercise-3/$ ./a.out
 Please insert a value for the size of the array:
100
 CPUtime for single-precision array =   2.89999880E-05
 CPUtime for double-precision array =   4.39998694E-05
 CPUtime for quadruple-precision array =   8.30000034E-04
username@tktl-pangolin:~/Tools-for-HPC/exercise-3/$ ./a.out
 Please insert a value for the size of the array:
1000
 CPUtime for single-precision array =   2.66800006E-03
 CPUtime for double-precision array =   3.49799963E-03
 CPUtime for quadruple-precision array =   4.43109982E-02
```

We can easily see that bigger the precision of the array, bigger the time that the CPU spents in the calculations. By chosing bigger values of n, the difference between the times spent in the calculation using each real kind gets more noticeable.