PHYS-4007/5007: Computational Physics

Determining Machine Precision of Real Numbers in Fortran

1 Prep Work

Log into your account on the Linux side of the computers in Brown Hall 264. Open a terminal window, then check to make sure that you have a subdirectory called fortran in your login directory using

```
> ls
```

— remember that the '>' above represents the Linux system prompt. **If you do not have such a subdirectory**, make one using

```
> mkdir fortran
```

then change to this subdirectory using

```
> cd fortran
```

Now create a new file called 'limit.f' with the command

```
> emacs limit.f &
```

(remember that the '&' sign put this job in background so that you can still access Linux in the terminal window).

2 Determining the Precision of Single- and Double-Precision Numbers

For this tutorial, you will be asked to write code to determine the machine precision for single- and double-precision numbers using the Fortran programming language in Linux. The code that you will write is based on pseudocode from *Computational Physics* (2007) of Landau, Paez, and Bordeiana on Page 23.

In the emacs GUI (Graphic User Interface) window, type in the following lines (except for the first sequence of number lines, don't worry about typing in the "comment" lines either, enter them later when you have more time):

```
123456789T123456...
      PROGRAM LIMIT
C
C This code will determine the machine precision for the machine on which it
C runs.  It is based on pseudocode from Computational Physics (1997) of Landau
C and Paez on Page 25.
C
C At the Unix prompt, use the command:
C     gfortran -o limit.exe limit.f        (Linux Workstations)
C to create the executable version of this code.
C Note that the executable must not be called "limit" by itself,
C since there is a 'limit' command in Unix.
C
      INTEGER I, NITER
      REAL SEPS, SONE
      DOUBLE PRECISION DEPS, DONE
      CHARACTER ASK
C
C Set an initial guess for how many iterations this will take.
C
      DATA NITER / 200 /
C
C Determine the machine precision.  Ask the user to continue every 100th
C iteration.
C
      SEPS = 1.0
      DEPS = 1.0D0
C
C Single precision calculation.
C
      PRINT *, 'In the tabel below:'
      PRINT *, '  ITER:  The iteration number.'
      PRINT *, '  ONE:   The value for the REAL variable ONE.'
      PRINT *, '  EPS:   The difference of ONE and the integer 1.'
      PRINT *, 'When the value printed for ONE is exactly 1, the'
      PRINT *, 'value for EPS will be the machine precision for a'
      PRINT *, 'single precision variable.'
      PRINT *, '  '
      PRINT *, ' ITER       ONE             EPS'
```

```
      DO 100 I = 1, NITER
         SEPS = SEPS / 2.0
         SONE = 1.0 + SEPS
         WRITE (*,50) I, SONE, SEPS
 50      FORMAT(I4, 2X, F15.12, 2X, 1PE14.7)
         IF (MOD(I, 100) .EQ. 0) THEN
            PRINT *, 'Continue with the single precision ',
     1         'calculations (y/n)? [y]'
            READ (*, '(A)') ASK
            IF ((ASK .EQ. 'N') .OR. (ASK .EQ. 'n')) GOTO 200
         ENDIF
 100  CONTINUE
C
C Double precision calculation.
C
 200  PRINT *, ' '
      PRINT *, 'In the tabel below:'
      PRINT *, '  ITER:  The iteration number.'
      PRINT *, '  ONE:   The value for the REAL*8 variable ONE.'
      PRINT *, '  EPS:   The difference of ONE and the integer 1.'
      PRINT *, 'When the value printed for ONE is exactly 1, the'
      PRINT *, 'value for EPS will be the machine precision for a'
      PRINT *, 'double precision variable.'
      PRINT *, '  '
      PRINT *, ' ITER          ONE                        EPS'
      DO 300 I = 1, NITER
         DEPS = DEPS / 2.0D0
         DONE = 1.0D0 + DEPS
         WRITE (*,250) I, DONE, DEPS
 250     FORMAT(I4, 2X, F23.20, 2X, 1PE23.15)
         IF (MOD(I, 100) .EQ. 0) THEN
            PRINT *, 'Continue with the double precision ',
     1         'calculations (y/n)? [y]'
            READ (*, '(A)') ASK
            IF ((ASK .EQ. 'N') .OR. (ASK .EQ. 'n')) STOP
         ENDIF
 300  CONTINUE
C
      STOP
      END
```

Now save your file, but don't exit the emacs editor yet. Go to the terminal window and enter the following command:

> gfortran -o limit.exe limit.f

If you have errors upon compiling, correct these until you no longer see any error messages. Once you have successfully compiled your code, run it with

> ./limit.exe

You will get a line-by-line output for the first hundred iterations. Look at the 'ONE' column (*i.e.*, the second column of numbers) of data for the first line that is exactly equal to '1' (*i.e.*, 1.000000000000) — this should occur around iteration 24. Now, look at the line just above this line (*i.e.*, iteration 23). Write down the value you get for 'EPS' (*i.e.*, the 3rd column of numbers) here (six significant digits is sufficient for this number):

Single-Precision Machine Error: _____

This represents the machine error for single-precision numbers. Now answer 'n' (minus the quotes) to the question *Continue with the single precision calculations (y/n)? [y].*

You will now see another 3-column tabular list of data for the double precision machine error. Once again, look at the 'ONE' column (*i.e.*, the second column of numbers) of data for the first line that is exactly equal to '1' (*i.e.*, 1.00000000000000000000) — this should occur around iteration 53. Now, look at the line just above this line (*i.e.*, iteration 52). Write down the value you get for 'EPS' (*i.e.*, the 3rd column of numbers) here (ten significant digits is sufficient for this number):

Double-Precision Machine Error: _____

This represents the machine error for double-precision numbers. Now answer 'n' (minus the quotes) to the question *Continue with the double precision calculations (y/n)? [y].*

Save this information so that you can make use of these values when writing future codes for this course, or any future code you write for your own research.