More on Logbooks

Some questions have come up as to what we are looking for in your log books. Below is a sample created with the change-log facility in EMACS:

Thu Aug 28 11:13:03 1997 Andrew C. Pineda <acpineda@gila.arc.unm.edu>

* testprec.f90: Wrote a program that investigates the effect of machine precision in calculations. The program I wrote is as follows:

A brief description of the exercise.

```
program testprec
!real x,y,zz
double precision  x,y,zz
x=1.1
print '(A,E20.12)','x= ',x
y=1.0; z=3.0; zz=y/z
print '(A,E20.12)','The result of the expression 1.0-3.0*zz is',1.0-3.0*zz
print *, 'The epsilon (machine precision) for the type used to represent x is', epsilon(x)
end program testprec
```

When I ran the program with the variables declared as real, the following output was obtained:

```
x=    .110000002384E+01
The result of the expression 1.0-3.0*zz is  -.298023223877E-07
 The epsilon (machine precision) for the type used to represent x is 0.1192092896E-06
```

Note that the value assigned to x is not exactly 1.1. This is because 1.1 is not exactly representable in binary (base-2). Hence what is stored is the nearest machine representable approximation to 1.1. When this is converted back to base-10 for output, we see a number that agrees with 1.1 to the precision allowed by the real data type.

Some discussion of observations and conclusions.

We see this effect again in the computation of 1.0-3.0*zz. 1.0 and 3.0 are both exactly representable by the real data type, but zz=1.0/3.0 is not, so we see a difference at the point where the machine representation differs from the exact result, i.e. in the seventh decimal place. The difference is comparable in magnitude with the machine precision returned by the Fortran intrinsic function EPSILON.

To verify these conclusions, I ran the program again with the variables declared as double precision, and obtained the following result.

```
x=    .110000002384E+01
The result of the expression 1.0-3.0*zz is   .555111512313E-16
 The epsilon (machine precision) for the type used to represent x is 0.22204460492503130E-15
```

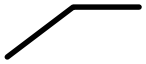
The results I see from the calculation of 1.0-3.0*zz are as I

expected for a double precision calculation. Since double precision maintains about 15 digits of accuracy, the result of 1.0-3.0*zz is smaller, as expected. It is also in rough agreement with the value reported by the EPSILON function.

The output for x is puzzling. It appears that the 1.1 that appears in the assignment x=1.1 is a single precision 1.1.

Questions to be answered:

1) Is the precision reported by EPSILON an absolute or relative value?
2) How do I enter 1.1 in the program so that it agrees with 1.1 to double precision accuracy?



Questions brought up by the exercise…