ACM40660 Practical 5

ICHEC

September 23, 2020

1 Overview

This week we shall start to use pointers and to look at user input to programs.

2 User Input in C

- So far we have looked at printing information to the screen, *printf*.
- Programs can also accept user input from the screen using, *scanf*.

```c
#include <stdio.h>
int main(void) {
  int i; double a;
// Enter information from user
  printf("Enter an int and double\n");
  scanf("%d %lf",&i,&a)
}
```

- The first thing to notice is that for *scanf* the addresses of the variables are passed because we want to bring information out of the function.
- The formatting descriptors are the same as those for *printf*, *e.g.* %d.
- However in this case the descriptors define how much space is reserved for the variable, thus we need to use of *%lf* in the format for long float or double.
- The *scanf* function returns the number of values it has read in, this can be tested.
- It is good practice to print the request for information.

3 User Input, FORTRAN

- The equivalent function in FORTRAN is below.

```
program uinput
   integer (kind=4) :: i,ierr=0
   real (kind=8) :: a

   write(6,*) ' Enter an int and double '
   do while (ierr .ne. 0)
     read(5,*,iostat=ierr) i,a
   end do
   if (ierr .ne. 0) write(6,*) ' Problem with input '
end program uinput
```

- In fortran read and write statements are assigned to a unit. Unit 5 is input from screen and 6 output to screen.
- The error is returned via an argument.
- Formatting is the same as that for write but the "*" format will accept anything.

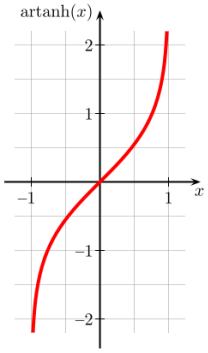
4 Exercises

- Create a program that displays the Fibonacci series upto n.
- The formulae for the Fibonacci series are:

$$F_n = F_{n-1} + F_{n-2} \quad (1)$$
$$F_0 = 0 \qquad F_1 = 1 \quad (2)$$

  1. Get the user to enter the value of n.
  2. Have a function that that has two arguments. On input the arguments have F_{n-1} and F_{n-2}. On exit they have F_n and F_{n-1}.
  3. Use a loop to find the whole series upto n.
  4. Print out the series.

- The Inverse Hyperbolic Tangent function can be calculated on a computer in several ways. Different methods behave differently and therefore differ in terms of speed, accuracy, precision and range of input accepted. Assume that both methods accept only real values whose absolute value is less than 1.



- The hyperbolic arc tangent can be expressed as a Maclaurin series, which is depicted below:

$$\operatorname{arctanh}(x) = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{2n+1}, x \in \Re : |x| < 1$$

- Another way to calculate the hyperbolic arc tangent is by using natural logarithms:

$$\operatorname{arctanh}(x) = \frac{1}{2}[\ln(1+x) - \ln(1-x)], x \in \Re : |x| < 1$$

  1. Get the user in enter a real positive number *delta*.
  2. Implement a function *artanh1(x)* as an approximation to this Maclaurin series. As the series is infinite, we will set the function to stop when the element in the series is smaller than a given precision, *delta*.

3. Implement function *artanh2(x)* using approximations to natural logarithms. Remember that natural logarithms are already implemented in the language's standard math libraries.
4. Write a *main* that will calculate $\text{arctanh}(x)$ where $x \in [-0.9, 0.9]$ and sampled every 0.01, using both methods. Store the results for both methods in separate arrays.
5. Compare the accuracy of each method with each other (to 10 significant figures).