Physics 410

Assignment #1: Due Friday, September 13, 2013

Attached is a simple program in C that computes the sum

$$S_N = \sum_{n=1}^{2N} (-1)^n \frac{(n-1)}{n}$$

by four different algorithms: (1) by generating each term and adding it sequentially from smallest to largest n then (2) from largest to smallest n, by first combining terms in the series analytically to obtain

$$S_N^{eq} = \sum_{n=1}^{N} \frac{1}{2n(2n-1)}$$

then (3) generating each term and adding it sequentially, first from smallest to largest n, then (4) from largest to smallest n.

1) Type in this program, compile and execute it using the IDE that you plan to use for the course. When successful, you will obtain a data file containing the sums. Attach a printout of the file for $N = 2$ to $N = 2^{20}$ by powers of 2 for credit for this part.

Note: The program takes two parameters that set the range of N from command line in a terminal window. It writes the output to both a terminal window and a file. You may have to find where your IDE writes the file by default, or modify the code to write it in a convenient location.

2) Discuss your results. Why do the answers not agree to all digits? Discuss the reason that each of the 4 algorithms gives a result different from the correct answer. Which of the 4 algorithms is most precise and why?

3) Discuss the declaration of the variables T, *countup* and *countdown*. What would happen if they were declared as int rather than float and why?

4) Now run the code for $N = 10$ to $N = 10^{10}$ by powers of 10. What happens? Why? How could you fix this?

Code for Alternating Sum

```
//
//  main.c
//  testingxcode
//
//  Created by Kristin Schleich on 2013-09-01.
//  Copyright (c) 2013 Kristin Schleich. All rights reserved.
//

#include <stdio.h>
#include <stdlib.h>
#include <math.h> // math functions


void output(FILE *, float, float, float,float, float);

// Program to calculate the sum of (-1)^n (n-1)/n by four different algorithms

int main()

{
float up, down, addup, adddown;
int  N,S,count;
float counterup, counterdown,T;


FILE *ofile;

up = 0;
down = 0;

ofile=fopen("subtractresults", "w");

// print to screen
printf ("Enter N, the first terminating number of the sum \n");
// read from screen
scanf("%d", &N);

printf ("Enter S, the power for N^S, the last terminating number \n");
// read from screen
scanf("%d", &S);

T=1.0;

fprintf(ofile,"#Results for subtraction error for N=%d, S=%d \n",N,S);
fprintf(ofile,"#Columns are terminating number, alg. 1 up, alg. 1 down, alg. 2 up, alg. 2 down,abs. dif


for(count=0;count<S;count++)
{
//reinitialize variables every iteration
      up = 0;
      down = 0;
      addup= 0;
      adddown= 0;

      T*=N;
      counterdown=2*T;
      counterup=1.0;

     while(counterup<2*T+1.0)
     {
          up-= (counterup-1.0)/counterup;
          up*=-1.0;
          down-=(counterdown-1.0)/counterdown;
          down*=-1.0;

          counterup+=1.0;
          counterdown-=1.0;
     }
     down*=-1.0;
     counterup=1.0;
     counterdown=T;

     while(counterup<T+1.0)
     {
          addup+=1/(2.0*counterup-1.0)/(2.0*counterup);
          adddown+=1/(2.0*counterdown-1.0)/(2.0*counterdown);

          counterup+=1.0;
          counterdown-=1.0;
      }
      // print to screen
      printf("T= %12.0f \n",T);

      printf("S(1)= %12.8f \n",up);
      printf("S(2)= %12.8f\n",down);
      printf("S(3)= %12.8f\n",addup);
      printf("S(4)= %12.8f\n",adddown);
      printf("\n");
      // print to file
      output(ofile,T, up, down,addup, adddown);
}

fclose(ofile);
return 0;
}

void output(FILE *ofile, float T, float up, float down,float addup, float adddown)
{
fprintf(ofile,"%12.0f %12.8f %12.8f %12.8f %12.8f %12.8f %12.8f\n",T,up,down,addup,adddown,
     fabs(down/up - 1.0), fabs(up/addup-1.0));
}
```