Examen - Errores Numéricos en la Computación

William Oquendo, woquendo@gmail.com

Contents

Abstract

Solamente se recibe y califica lo enviado a través de github. Debe estar presente en el salón al realizar la prueba. Siga las instrucciones de nombres que especifica cada ejercicio. Por defecto, para cada ejercicio cree un archivo con el numero de ejercicio y la extensión apropiada. Por ejemplo, si en el problema 1 le piden hacer un programa, entonces la solución sería un archivo llamado 1.cpp . Pero si en el ejercicio 5 le piden responder o hacer un análisis, entonces la solución se escribiría en un archivo 5.txt . Cada archivo de respuesta debe estar incluido en el repositorio pero en commits diferentes. Se evaluará lo enviado al repositorio remoto asignado hasta la hora de finalización del examen. A menos que se indique lo contrario, **ningún** programa deberá pedir entrada de usuario. Haga uso de lo visto hasta ahora en el curso, además de lo que se pide explícitamente en cada punto. Todos los códigos que usted envíe serán chequeados con los sanitizers y con valgrind y no deben presentar errores.

1 (2.5/5.0) Substracción cancelativa

1. La función $f(x) = (1-x)^8$ puede usarse como otro ejemplo de la substracción cancelativa cuando $x \to 1$. En este ejercicio va a comparar graficamente diversas aproximaciones para calcular esta función. Para esto se van a definir tres formas de calcular la función:

   - usando la función pow de c++, `std::pow(1-x,8)` . A esta la llamaremos $p(x)$.
   - La segunda forma, usando la expansion del polinomio,
   $$q(x) = 1 - 8x + 28x^2 - 56x^3 + 70x^4 - 56x^5 + 28x^6 - 8x^7 + x^8, \tag{1}$$
   en donde las multiplicaciones se evalúan término a término (por ejemplo, $x^3 = x \cdot x \cdot x$), sin usar la función `std::pow`.
   - Y la tercera, $r(x)$, que es similar a $q(x)$ pero en donde las potencias se evalúan cada una usando la función std::pow en lugar de multiplicar.

   Implemente cada una de las anteriores funciones como una función de c++ en un código de prueba que debe imprimir el valor de la función evaluado en $x = 0.01$. Guarde este código en un archivo llamado `1.cpp` .

2. Extienda su programa del punto anterior para que imprima una tabla de cuatro columnas con la siguiente información

$$\begin{array}{cccc} x & p(x) & q(x) & r(x) \\ \vdots & \vdots & \vdots & \vdots \end{array}$$

   Para esto $x \in [0.992, 1.008]$ y $\Delta x = 0.0001$. Guarde la solución en un archivo llamado `2.cpp` .

3. Haga una gráfica de la tabla anterior, con las características apropiadas de calidad y presentación, e indique cuál es la mejor aproximación para el cálculo que se está realizando y porqué. Guarde la figura en archivo `3.pdf` y el análisis en el archivo `3.txt` .

2 (2.5/5.0) Precisión de la máquina

La serie

$$f(N) = \sum_{n=1}^{N} \frac{1}{n^2} \tag{2}$$

tiende a $\pi^2/2$ cuando $N \to \infty$. Matemáticamente, esta serie también puede escribirse como

$$g(N) = \sum_{n=N}^{1} \frac{1}{n^2}. \tag{3}$$

4. Escriba un programa que calcule la diferencia porcentual entre el valor exacto y las series $f(N)$ y $g(N)$ como función del valor de N. Su programa debe imprimir una tabla como

$$\begin{array}{ccc} N & \Delta f & \Delta g \\ \vdots & \vdots & \vdots \end{array}$$

   donde Δ es la diferencia porcentual con el valor exacto. Llame a este programa `4.cpp` .

5. Grafique los datos obtenidos anteriormente en la figura `5.pdf`.

6. Explique porqué difieren estas series y cuál es la más precisa y porqué. Escriba este análisis en el archivo `6.txt`.