Facultad de Matemática y Computación (UH)
Ciencia de la Computación
Matemática Numérica

Clase Práctica 8:
Derivar puede cualquiera, integrar también (si sabe numérica y usa `Sympy`)

Curso 2024

Lo más sencillo

Ejercicio 1: Lo básico de las integrales (20000 créditos)

Dada una función integrable $f : \mathbb{R} \to \mathbb{R}$, y un intervalo $[a, b]$, implemente una función en Python que calcule el valor de $\displaystyle\int_a^b f(x)dx$ usando:

a) El método compuesto del rectángulo.

b) El método compuesto de los trapecios.

c) El método compuesto del punto medio.

d) El método compuesto de Simpson.

Ejercicio 2: Y una demostración sencilla (30000 créditos)

Demuestra que si un método simple que permite aproximar la integral $\displaystyle\int_a^b f(x)dx$ tiene un error $O((b-a)^n)$, entonces si se usa el méttodo compuesto correspondiente con N intervalos, se obtiene una aproximación con un error $O(h^{n-1})$, donde $h = \frac{b-a}{N}$.

Ejercicio 3: Mejor error conocido y acotado... (40000 créditos)

Para cada uno de los métodos compuestos implementados por usted en el ejercicio 1 para calcular $\displaystyle\int_a^b f(x)dx$, diga en cuántos subintervalos es necesario dividir el intervalo $[a, b]$ para obtener una aproximación de la integral con un error que sea menor que una tolerancia predeterminada.

Ejercicio 4: Lo mínimo de las derivadas aproximadas (60000 créditos)

Dada una función $f \in C^{\infty}[a,b]$, un punto $\bar{x} \in [a,b]$ y un valor de $h \in \mathbb{R}_+$, seleccione un método para aproximar derivadas usando diferencias finitas[1] e implemente una función en Python que:

a) Devuelva el valor de la derivada de $f(\bar{x})$, evaluada en el punto $\bar{x}$.

b) Grafique las funciones $f(x)$ y $f'(x)$ en el intervalo $[a,b]$.

c) Para un valor fijo de $\bar{x}$ grafique el error que se comete al aproximar $f'(\bar{x})$ usando distintos valores de h si se usa el método seleccionado por usted.

Ejercicios integradores

Ejercicio 5: De coraje, manos sucias y errores (80000 créditos)

Hasta ahora se sabe que la primera derivada de una función $f(x)$ se puede aproximar como

$$f'(x) = \frac{f(x+h) - f(x)}{h} \text{ y } f'(x) = \frac{f(x+h) - f(x-h)}{2h},$$

y que la primera aproximación tiene un error $O(h)$ y la segunda $O(h^2)$.

Sin embargo, esas aproximaciones también se pueden obtener interpolando la función y derivando el polinomio de interpolación[2].

a) Demuestre que si la función se aproxima por un polinomio de interpolación que pasa por los puntos $(x_0, f(x_0))$ y $(x_0+h, f(x_0+h))$, se obtiene la primera aproxmación, con su correspondiente error.

b) Demuestre que si la función se aproxima por un polinomio de interpolación que pasa por los puntos $(x_0, f(x_0))$, $(x_0-h, f(x_0-h))$, $(x_0+h, f(x_0+h))$, se deriva, y se evalúa la derivada en el punto x_0 se obtiene la segunda aproxmación, con su correspondiente error.

c) ¿Qué expresiones se obtienen si la función se interpola en tres puntos, se deriva el polinomio de interpolación y se evalúa en los puntos x_0 y x_2?

d) ¿Qué error se comete en cada caso?

Ejercicio 6: Órdenes de precisión y sistemas de ecuaciones lineales (100000 créditos)

Dada una función $f(x)$ cualquiera se puede aproximar la integral $\int_a^b f(x)dx$ como una suma: $w_1 f(x_1) + w_2 f(x_2) + \ldots + w_n f(x_n)$, donde w_i son números reales seleccionados convenientemente.

[1] ¡Taratatáaan! Acabas de encontrar una pregunta secreta por un valor de 20000 créditos: ¿qué son diferencias finitas? `:-/`

[2] Al menos eso fue lo que dijo el profe en la conferencia, pero a los profesores no siempre se les puede creer todo... ¡Hay que demostrar lo que ellos dicen!

a) Diga qué valores deben tener los números w_i de forma que las integrales $\int_a^b f_i(x)dx$ se puedan calcular de manera exacta para todas las funciones $f_i(x) = x^i$, con $i = \overline{0, n}$.

b) Diseñe e implemente un algoritmo que reciba el valor de n y devuelva los valores de las constantes w_i, con $i = \overline{1, n}$.

Y dos ejercicios para responderlos todos

Ejercicio 7: Todas las derivadas por el precio de un `Sympy`[3] (200000 créditos)

Utilizando las facilidades que brinda `Sympy` y sus[4] conocimientos de interpolación y su uso en la aproximación de derivadas numéricas:

a) Implemente una función que reciba dos enteros `n` y `p` y devuelva una función y un número $h_{best} \in \mathbb{R}$ donde:

   a) La función recibe tres argumentos: una función y dos números donde

      1) la función es la implementación en Python de una función $f \in C^{\infty}[-\infty, +\infty]$,
      2) el primer número es un punto $x \in \mathbb{R}$ donde se desea aproximar la derivada n-ésima de $f(x)$, y
      3) el segundo número es un valor $h \in \mathbb{R}$ con el que se desea aproximar la derivada.

   y que devuelva: una aproximación de la derivada $f^{(n)}(x)$ que tenga un error $O(h^p)$.

   b) El valor h_{best} es una recomendación de qué valor de h es el más apropiado[5] para aproximar la derivada de orden `n` con un error $O(h^p)$.

Ejercicio 8: $\int_a^b f(x)dx$ + `Sympy` = `:-D` (300000 créditos)

Una forma de aproximar la integral de una función $f(x)$ en un intervalo $[a, b]$ es interpolar la función en n puntos e integrar el polinomio de interpolación.

a) Diseñe un algoritmo que reciba un número n y devuelva una fórmula para hallar la integral de una función $f(x)$ en un intervalo $[a, b]$ utilizando un polinomio de interpolación con n puntos.

b) ¿Qué error se comete al aproximar la integral por la expresión propuesta por usted en el inciso anterior?

c) Utilice su algoritmo para hallar fórmulas de integración basada en 1, 2, y 3 puntos, y los errores cometidos en cada caso.

[3] Este ejercicio puede estar difícil de hacer, pero si se hace, los demás deberían poderse hacer muy rápido.

[4] "sus conocimientos" se refiere a los conocimientos de la persona que lee estas líneas, y no a los de `Sympy`.

[5] ¡Taratatáaan! Se acerca una pregunta secreta por un valor de 20000 créditos: ¿Por qué hace falta un valor de h que sea "el más apropiado" y no simplemente coger uno bien pequeño?

Bibliografía recomendada

- *Elementary Numerical Analysis, An algorithmic approach. 3rd Edition.* S. D. Conte y Carl de Boor. McGraw-Hill Book Company. 1980.
- *Numerical Methods. 9th Edition.* J. D. Faires y R. L. Burden. Brooks Cole Publishing, 2011.
- *Numerical Computing with Matlab.* C. Moler. 2004.
- *Matemática Numérica II. Notas de Clase* A. León Mecías. Facultad de Matemática y Computación, UH. 2014.