

Tarea 6 - Problema 2 - Programación avanzada

Nicolás Aylwin

Para este problema se creó el archivo de funciones template llamado “cuadratura.h”. En este se incluyen los tres métodos de cuadratura. Comentaré lo no mencionado en los comentarios sobre las funciones.

En la del trapezoide, fijé una cantidad alta inicial porque noté que convergía mejor con la integral especificada. Además una cantidad máxima de trapecios pues a veces le cuesta mucho converger dado el error, y puede iterar por demasiado rato. Sobre las otras funciones todo está aclarado en los comentarios.

El programa en sí que resuelve el problema, en este caso “problema2.cc”, se ejecuta y a través de la terminal va solicitando los valores en que evaluar las funciones (e integrales que componen estas). Es autoexplicativo y fácil de usar.

Sobre la función Gamma, para la segunda y tercera integral, se prohíbe el ingresar valores de s tal que $s < 1$, pues en estos casos la integral diverge. De entregar un valor así se solicita que se entregue uno distinto hasta que se logre (por un ciclo `while(true)`).

Para el cálculo de la tercera integral (asumiendo $1 \leq s$), se tuvo mucho cuidado en lo que significaba de cero hasta el infinito. La función es acampanada y en sus bordes tiene a cero, ya que la ubicación de esta depende del parámetro s , contextualicé usando derivadas. En particular:

$$(x^{s-1}e^{-x})' = e^{-x}((s-1)x^{(s-2)} - x^{(s-1)})$$

Así se tiene que el máximo de la función (único punto crítico):

$$(s-1) = x$$

Además se tiene que:

$$(x^{s-1}e^{-x})'' = e^{-x}(x^{(s-1)} - 2(s-1)x^{(s-2)} + (s-1)(s-2)x^{(s-3)})$$

Donde los punto de inflexión serán (resolviendo la cuadrática):

$$x = (s-1) \pm \sqrt{s-1}$$

Teniendo estos resultados, podemos saber donde está la parte de la función que más aporta a la integral. Primero tomemos una unidad de medida acorde al caso, la diferencia entre el punto de inflexión mayor y el máximo de la función será:

$$(s-1) + \sqrt{s-1} - (s-1) = +\sqrt{s-1}$$

Este justamente es la variable tipo double “paso” en el while que evalúa romberg (en problema2.cc). También así definí el punto de inflexión mayor “inflex”. Luego, integro hasta el punto de inflexión

más 3 veces el paso (inicialmente) y voy aumentando el paso hasta que los valores obtenidos estén muy cerca. La gracia de esto es situarme en una zona donde la campana de la función esta a punto de terminar, por lo que estoy considerando la parte que más afecta la integral desde el primer instante.

Cabe destacar que además se incluye un if para los resultados obtenidos, esto porque en algunos casos romberg divergía luego de no conseguir el error, y entregaba valores muy pequeños sin sentido, de esta manera, si llega a diverger entrego el último resultado con sentido.

Finalmente se entrega el valor de la última integral.