

PROYECTO FINAL INGENIERIA DE SOFTWARE

JUAN CAMILO RAMIREZ

JUAN FELIPE CASTILLO GOMEZ

SANTIAGO PRADO

JUAN DIEGO LORA

UNIVERSIDAD ICESI

FACULTAD DE INGENIERÍA, DISEÑO Y CIENCIAS APLICADAS

INGENIERÍA DE SISTEMAS

SANTIAGO DE CALI

2024

**Introducción**

El presente documento busca definir la mejor estrategia de Integración para funciones de una sola variable (x), asegurando el mejor resultado en precisión (Accuracy) y desempeño (Performance) que se pueda obtener.

**Alcance**

Para este proyecto hemos limitado el alcance a funciones de una sola variable (x). Por lo tanto, la elección de estrategia y distribución de la solución será limitada por dicho alcance.

**Métodos de integración**

* **Sumas de Riemann**

Este método consiste en dividir un intervalo de integración en N rectángulos. De cada rectángulo se obtendrá su área individual, para al final, sumar dichas áreas y conseguir un valor de integración muy aproximado al real.

**Definición del método matemático:**

Donde,

N: Se define como el número de particiones o rectángulos que se crearan.

Se define como la altura individual de cada rectángulo

Se define como la base individual de cada rectángulo

Hace referencia a la función que se va a integrar

Se define como (Se tiene presente el extremo derecho de cada rectángulo)

Se define como el rango inferior y superior de integración respectivamente

Hace referencia al índice de la sumatoria

Por tanto, la definición completa del método de Riemann se detalla como:

La ventaja de este método reside en el número de rectángulos entre el intervalo de integración, mientras más rectángulos existan (tiendan a ser infinitos) más cercano al valor real de integración se logrará conseguir.

* **Monte Carlo**

Este método consiste en la generación de múltiples puntos aleatorios dentro del rango de integración. De estos puntos se obtiene el promedio, y se multiplica por la diferencia entre los rangos de integración.

**Definición del método matemático:**

Donde,

N: Se define como el número de puntos aleatorios dentro del rango de integración.

Se define como la altura promedio

Se define como la base

Hace referencia a la función que se va a integrar

Se define como el punto sub(i) generado aleatoriamente

Se define como el rango inferior y superior de integración respectivamente

Hace referencia al índice de la sumatoria

Se define como

Por tanto, la definición completa del método de Monte Carlo se detalla como:

La ventaja de este método está en el número de puntos que se generan aleatoriamente, la idea con estos puntos es calcular una altura promedia a la altura real de la función, al generar múltiples puntos aleatorios se tiende a cumplir la Ley de los grandes números (Que dice que el promedio de los resultados obtenidos en múltiples muestras aleatorias estará muy cerca del valor real)

**Selección**

Para la selección, buscamos las soluciones que más nos aportaran en el Perfomance y el Accuracy.

**Experimentos**

En el presente estudio, se realizaron experimentos comparativos para evaluar la eficacia de los métodos de Montecarlo y Riemann en la aproximación de la integral , cuyo resultado teórico esperado es . A continuación, se detallan los procedimientos ejecutados y los resultados obtenidos de cada método:

* **Método de Montecarlo**

Este método involucró la utilización de distintas cantidades de puntos aleatorios para evaluar la integral. Los resultados obtenidos muestran una convergencia progresiva hacia el valor de (3.1415926535897932384) conforme se incrementa el número de puntos:

1,000 puntos:

* Resultado: 3.1981427147040593
* Latencia: 0.054 segundos (0 minutos, 0.054 segundos)
* Decimales de precisión: 1

10,000 puntos:

* Resultado: 3.138421463314007
* Latencia: 0.147 segundos (0 minutos, 0.147 segundos)
* Decimales de precisión: 1

100,000 puntos:

* Resultado: 3.1433764408243734
* Latencia: 0.456 segundos (0 minutos, 0.456 segundos)
* Decimales de precisión: 2

1,000,000 puntos:

* Resultado: 3.1423180626463925
* Latencia: 1.667 segundos (0 minutos, 1.667 segundos)
* Decimales de precisión: 2

10,000,000 puntos:

* Resultado: 3.141669142483554
* Latencia: 14.55 segundos (0 minutos, 14.55 segundos)
* Decimales de precisión: 3

100,000,000 puntos:

* Resultado: 3.1414386984801967
* Latencia: 141.43 segundos (2 minutos, 21.43 segundos)
* Decimales de precisión: 3

1,000,000,000 puntos:

* Resultado: 3.1415895821397997
* Latencia: 1599.36 segundos (26 minutos, 39.36 segundos)
* Decimales de precisión: 4

Además, se llevó a cabo otro experimento adicional en este método de Montecarlo para aproximar el valor de la integral de en el intervalo de 0 a 1. Los resultados obtenidos mostraron que, al incrementar progresivamente la cantidad de puntos aleatorios utilizados, la estimación se aproximaba cada vez más al valor teórico de 0.3333. A continuación, se presentan una serie de histogramas que ilustran los resultados obtenidos con 1,000, 10,000, 100,000 y 1,000,000 de puntos. En las siguientes gráficas, se observa que a medida que aumenta el número de puntos, la precisión del método de Montecarlo mejora significativamente, evidenciando una mayor concentración de resultados en torno al valor teórico esperado.

Gráfico, Histograma

Descripción generada automáticamente

Gráfico, Histograma

Descripción generada automáticamente

Gráfico, Histograma

Descripción generada automáticamente

Gráfico, Histograma

Descripción generada automáticamente

* **Método de Riemann**

Este método, que consiste en la división del intervalo de integración en particiones, presentó una latencia mayor en comparación con el método de Montecarlo, pero logró una precisión superior en la aproximación del valor de (3.1415926535897932384). La mejora en la precisión fue notable a medida que se incrementaba el número de particiones utilizadas en la aproximación:

1,000 particiones:

* Resultado: 3.1414874770021406
* Latencia: 0.058 segundos (0 minutos, 0.058 segundos)
* Decimales de precisión: 3

10,000 particiones:

* Resultado: 3.1415893274305815
* Latencia: 0.149 segundos (0 minutos, 0.149 segundos)
* Decimales de precisión: 4

100,000 particiones:

* Resultado: 3.1415925484068232
* Latencia: 0.533 segundos (0 minutos, 0.533 segundos)
* Decimales de precisión: 6

1,000,000 particiones:

* Resultado: 3.141592650263614
* Latencia: 3.062 segundos (0 minutos, 3.062 segundos)
* Decimales de precisión: 7

10,000,000 particiones:

* Resultado: 3.14159265348461
* Latencia: 30.63 segundos (0 minutos, 30.63 segundos)
* Decimales de precisión: 9

100,000,000 particiones:

* Resultado: 3.141592653586467
* Latencia: 325.17 segundos (5 minutos, 25.17 segundos)
* Decimales de precisión: 10

1,000,000,000 particiones:

* Resultado: 3.141592653589688
* Latencia: 3510.99 segundos (58 minutos, 30.99 segundos)
* Decimales de precisión: 12

**Conclusión general sobre el método de Montecarlo y sumas Riemann**

Los experimentos realizados proporcionaron una comparativa valiosa entre los métodos de Montecarlo y Riemann para la aproximación de integrales. Aunque el método de Montecarlo es más rápido en términos de latencia para cantidades menores de puntos, el método de Riemann ofrece una mayor precisión al incrementar el número de particiones, alcanzando una aproximación casi exacta a con particiones suficientemente grandes, pero con un perfomance muy ineficaz.

**Determinación del punto de inicio para la distribución basada en el análisis de rendimiento**

Los resultados de nuestros experimentos destacan diferencias significativas en rendimiento y precisión entre los métodos de Montecarlo y Riemann, subrayando la importancia de adoptar estrategias distributivas para manejar cálculos extensivos, particularmente con el método de Riemann debido a su intensiva demanda de recursos computacionales.

* **Método de Montecarlo:**

Este método demuestra una convergencia rápida hacia el valor esperado de las integrales, con un aumento notable en la precisión a medida que se incrementa el número de puntos. Observamos una mejora sustancial en precisión a partir de 100,000 puntos, pero con un incremento asociado en la latencia. Aunque el método de Montecarlo es eficiente para cantidades menores de puntos, para simulaciones a gran escala (es decir, más de 1,000,000,000 puntos), donde la latencia alcanza 1599.36 segundos (aproximadamente 26 minutos y 39 segundos), se hace imprescindible la implementación de una estrategia distribuida para gestionar eficientemente el aumento en la carga computacional.

* **Método de Riemann:**

A pesar de su alta precisión, este método presenta ineficiencias significativas en términos de latencia, especialmente al incrementar el número de particiones. La latencia crece exponencialmente con el número de particiones, llegando a ser impracticable en simulaciones detalladas; por ejemplo, con 1,000,000,000 de particiones, la ejecución puede tardar casi una hora. Es crucial, por lo tanto, considerar el uso de técnicas de computación distribuida a partir de 10,000,000 de particiones para optimizar el rendimiento sin comprometer la precisión.

**Impacto de la distribución en el rendimiento**

La estrategia de distribución tiene un efecto significativo en el rendimiento de ambos métodos. En el método de Montecarlo, permite gestionar un mayor número de puntos sin afectar considerablemente la latencia, mejorando notablemente el rendimiento. Para el método de Riemann, la distribución es casi indispensable para manejar el alto costo computacional y mejorar el performance.

En conclusión, la implementación de estrategias de distribución es fundamental para manejar la carga computacional en simulaciones extensivas. Este enfoque no solo mejora la eficiencia de procesamiento, sino que también asegura la precisión en los resultados, haciendo viables los métodos de Montecarlo y Riemann para aplicaciones prácticas en escenarios de alta complejidad.

**Selección final (Dejarla pendiente)**