

REPORTE DE ALGORITMOS

Regla de Simpson 3/8

|  |  |
| --- | --- |
| Nombre | Expediente |
| Zuñiga Fragoso Diego Joel | 317684 |

Asignatura: Método Numéricos 2023-2

Docente: Vargas Vázquez Damián

1. **Antecedentes teóricos**

La regla de Simpson 3/8 es una extensión del método de Simpson 1/3, una técnica numérica para aproximar el valor de una integral definida. Al igual que en Simpson 1/3, la regla de Simpson 3/8 divide el intervalo de integración en subintervalos y utiliza polinomios cúbicos para aproximar la función en cada uno de ellos.

En comparación con la regla de Simpson 1/3, la regla de Simpson 3/8 utiliza cuatro puntos de la función en cada subintervalo, permitiendo una aproximación más precisa al emplear polinomios de mayor grado. Esto se traduce en una mayor flexibilidad para adaptarse a curvas más complejas.

Es importante señalar que la regla de Simpson 3/8 se aplica cuando el número total de subintervalos es divisible por 3. En casos donde esto no es posible, se pueden utilizar segmentos finales con la regla de Simpson 1/3.

En resumen, la regla de Simpson 3/8 es una extensión que mejora la capacidad de aproximación del método original, ofreciendo una alternativa robusta y precisa para calcular integrales definidas en diversos contextos numéricos y científicos.

1. **Algoritmos y sus resultados**

Cada algoritmo esta seccionado e incluye descripciones de lo que sucede. Además de contar con capturas de sus resultados

|  |
| --- |
| **Código**  #include <iostream>  #include <stdio.h>  #include <math.h>  using namespace std;  class function  {  int degree;  double\* coefficients;  public:  function()  {  cout << "Ingrese el grado de la funcion:\t\t";  cin >> degree;  coefficients = new double[degree + 1];  for (int exponent = degree; exponent >= 0; exponent--)  {  if (exponent > 0)  cout << "\nIngrese el coeficiente de x^" << exponent << ":\t\t";  else  cout << "\nIngrese el coeficiente sin x:\t\t";  cin >> coefficients[exponent];  }  cout << "\n\nLa funcion ingresada es:\t"; this->print();  }  ~function()  {  delete[] coefficients;  }  void print()  {  cout << "f(x) = ";  for (int exponent = degree; exponent >= 0; exponent--)  {  if (exponent > 0)  printf("(%g)x^%d + ", coefficients[exponent], exponent);  else  printf("(%g)", coefficients[exponent]);  }  }  double evaluate(double x)  {  double result = 0.0;  for (int exponent = degree; exponent >= 0; exponent--)  result += pow(x, exponent) \* coefficients[exponent];  return result;  }  };  int main()  {  cout << "Programa para realizar regla de Simpson 3/8 Simple y multiple";  cout << endl << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";  cout << "\n\nCREACION DE FUNCION:\n\n";  // Creamos una nueva funcion  function fx;  cout << endl << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";  cout << "\n\nSIMPSOM 3/8:\n\n";  // Llenado de datos para simpsom 3/8 multiple o simple  double a, b;  cout << "Ingrese el limite inferior de la integral =\t\t"; cin >> a;  cout << "\nIngrese el limite superior de la integral =\t\t"; cin >> b;  int n;  cout << "\nIngrese el numero de intervalos =\t\t\t"; cin >> n;  double delta = (b - a) / n;  // Aplicacion del metodo de simpsom 3/8  double aux = 0.0;  for (int i = 1; i < n; i++) // Desde el segundo segmento hasta el penultimo  aux += fx.evaluate(a + (delta \* i));  double integral = (3 \* delta / 8) \* (fx.evaluate(a) + (3 \* aux) + fx.evaluate(b));  cout << "\n\nAproximacion de la integral de (" << a << ") -> (" << b << "), de la funcion: \n"; fx.print();  cout << " = " << integral << endl << endl;  system("pause");  return 0;  } |
| **Resultado** |

1. **Conclusiones**

La regla de Simpson 3/8 se presenta como una extensión valiosa y más precisa del método de Simpson 1/3 en la aproximación numérica de integrales definidas. Esta técnica, al utilizar polinomios cúbicos en lugar de parábolas, ofrece una mayor flexibilidad para adaptarse a funciones más complejas, permitiendo una aproximación más precisa de las áreas bajo las curvas.