

REPORTE DE ALGORITMOS

interpolación de sPLINES

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

Asignatura: Método Numéricos 2023-2

Docente: Vargas Vázquez Damián

1. **Antecedentes teóricos**

Los splines son una técnica en matemáticas y computación gráfica utilizada para aproximar curvas o superficies. La idea principal detrás de los splines es dividir una curva en segmentos más pequeños y representar cada segmento mediante un polinomio de bajo grado.

1. Orígenes Matemáticos:

El término "spline" proviene de la palabra alemana "splint", que significa "tabla" o "listón". La técnica de los splines se originó en la década de 1940 en el campo de la interpolación numérica y fue desarrollada para aproximar curvas suaves que pasan a través de un conjunto dado de puntos.

1. Interpolación y Ajuste de Curvas

Los splines se destacan en problemas de interpolación y ajuste de curvas. A diferencia de algunos métodos que utilizan un solo polinomio para toda la curva, los splines dividen la curva en segmentos y utilizan polinomios más simples para cada segmento. Esto ayuda a evitar problemas como el fenómeno de Runge y mejora la suavidad de la curva.

1. Tipos de Splines:

Existen diferentes tipos de splines, siendo los splines cúbicos los más comunes. Los splines cúbicos utilizan polinomios de tercer grado para cada segmento, y se prefieren debido a su buen equilibrio entre flexibilidad y simplicidad.

1. Aplicaciones Prácticas:

Los splines se aplican en diversas áreas, como gráficos por computadora, diseño asistido por computadora (CAD), animación, análisis de datos, y en general, en cualquier situación donde se requiera representar curvas suaves de manera eficiente.

1. Teoría de Nudos:

La teoría de nudos es un componente importante de los splines, donde la elección de los puntos de control (nudos) y la forma en que se conectan afectan la suavidad y flexibilidad de la curva resultante.

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 <stdio.h>  #include <iostream>  using namespace std;  void SplineInterpolation(int n, double\* x, double\* y, double xvalue, int degree)  {  double\* a = new double[n];  double\* b = new double[n];  double\* c = new double[n];  double\* d = new double[n];  for (int i = 0; i < n; i++)  a[i] = y[i];  // Calcula coeficientes para Splines de grado 1  if (degree >= 1)  {  for (int i = 0; i < n - 1; i++)  {  b[i] = (a[i + 1] - a[i]) / (x[i + 1] - x[i]);  c[i] = 0.0;  d[i] = 0.0;  }  // Impresion de las ecuaciones de las Splines de grado 1  cout << "\n\nEcuaciones de las Splines de Grado 1:\n";  for (int i = 0; i < n - 1; i++)  printf("Spline %d: f(x) = %g + %g(x - %g)\n", i + 1, a[i], b[i], x[i]);  }  // Calcula coeficientes para Splines de grado 2  if (degree >= 2)  {  for (int i = 0; i < n - 1; i++)  {  c[i] = (b[i + 1] - b[i]) / (x[i + 1] - x[i]);  d[i] = 0.0;  }  // Impresion de las ecuaciones de las Splines de grado 2  cout << "\nEcuaciones de las Splines de Grado 2:\n";  for (int i = 0; i < n - 1; i++)  printf("Spline %d: f(x) = %g + %g(x - %g) + %g(x - %g)^2\n", i + 1, a[i], b[i], x[i], c[i], x[i]);  }  // Calcula coeficientes para Splines de grado 3  if (degree == 3) {  for (int i = 0; i < n - 1; i++) {  d[i] = (c[i + 1] - c[i]) / (x[i + 1] - x[i]);  }  // Impresion de las ecuaciones de las Splines de grado 3  cout << "\nEcuaciones de las Splines de Grado 3:\n";  for (int i = 0; i < n - 1; i++)  printf("Spline %d: f(x) = %g + %g(x - %g) + %g(x - %g)^2 + %g(x - %g)^3\n", i + 1, a[i], b[i], x[i], c[i], x[i], d[i], x[i]);  }  // Realiza la interpolación  for (int i = 0; i < n - 1; i++)  {  if (xvalue >= x[i] && xvalue <= x[i + 1])  {  double xi = xvalue - x[i];  double spline\_result = a[i] + b[i] \* xi;  if (degree >= 2) {  spline\_result += c[i] \* xi \* xi;  }  if (degree == 3) {  spline\_result += d[i] \* xi \* xi \* xi;  }  printf("\nEl valor interpolado en x = %g es: %g\n", xvalue, spline\_result);  delete[] a;  delete[] b;  delete[] c;  delete[] d;  return;  }  }  cout << "\nEl valor de x está fuera del rango de interpolación.\n";  }  int main()  {  cout << "Programa hecho para calcular la interpolacion de Splines.\n\n";  Points:  int n;  cout << "Ingresa el numero de puntos:\t\t";  cin >> n;  if (n <= 0)  goto Points;  // Creamos arreglos de memoria dinamica  double\* x = new double[n];  double\* y = new double[n];  // Pedimos al usuario que ingrese los puntos  for (int i = 0; i < n; i++)  {  cout << "\n\nIngrese x\_" << i << " =\t\t"; cin >> x[i];  cout << "Ingrese f(x\_" << i << ") =\t"; cin >> y[i];  }  // Ingreso de valor de x  double xvalue;  cout << "\n\nIngresa el valor de x para la interpolacion:\t\t";  cin >> xvalue;  int degree;  cout << "\n\nIngresa el grado maximo de las splines (1, 2 o 3):\t\t";  cin >> degree;  SplineInterpolation(n, x, y, xvalue, degree);  // Liberamos memoria dinamica  delete[] x;  delete[] y;  return 0;  } |
| **Resultado** |

1. **Conclusiones**

En conclusión, los splines representan una poderosa y versátil herramienta en el ámbito de la matemática y la computación gráfica. Su enfoque de dividir una curva en segmentos y utilizar polinomios más simples para cada tramo ha demostrado ser eficaz en la interpolación y ajuste suave de curvas.

La evolución de los splines, desde sus orígenes en la década de 1940 hasta su aplicación generalizada en campos como gráficos por computadora, CAD, animación y análisis de datos, destaca su importancia y versatilidad.