MATH 352 - PROJECT

SIERRA BATTAN, JACK BROCKWAY, & RANDY CHEN

Program Guide:

- *NCIS.java* is our main file that reads user input, moves the points into our data structures, calls both our *tridAlgo* function (our tridiagonal system solving subroutine) and our *ncisAlgo* functions, and prints our findings.
- Note: these are the assumptions we have made for our program:
  - ∗ All inputs are of numeric form (i.e., ”cat dog” is an invalid point).
  - ∗ All inputs are entered in ascending order.
- **Usage Instructions:**
  - ∗ Launch a terminal
  - ∗ Navigate to the folder: $BBC352Project$
  - ∗ Compile our program: $javacNCIS.java$
  - ∗ Launch our program: $javaNCIS$
  - ∗ Input all points on new lines: x y
  - ∗ When done entering points, write: *done*
  - ∗ Our program will then output the resulting, simplified, natural cubic interpolation spline. ☺

Natural Cubic Interpolation Splines:

Given a table of data points, this algorithm aims to define an interpolating cubic spline function whose knots coincide with the ascending x values of the data points. A piecewise function must meet several conditions to qualify as a natural cubic interpolating spline. First, the function $S(x)$ consists of $n-1$ pieces (each a polynomial of degree ≤ 3), defined from $[t_i, t_{i+1}]$. In addition, the polynomial satisfies the interpolation conditions, $S(t_i) = y_i$. Furthermore, the continuity conditions are imposed at interior knots, such that $S^k(t_i) = S^k(t_{i+1})$ at each border point. Finally, since the function is natural, we make the choice that $S''(t_0) = S''(t_n) = 0$ for our two extra freedoms. After setting these conditions, we then run Gaussian Elimination on a tridiagonal system to solve for a myriad of variables to result in the polynomials: $S_i(x) = \frac{z_{i+1}}{6h_i}(x-t_i)^3 + \frac{z_i}{6h_i}(t_{i+1}-x)^3 + C_i(x-t_i) + D_i(t_{i+1}-x)$, which, for simplification purposes for our computer algorithm, we choose to rewrite in the form: $S_i(x) = D_i + C_i(x-t_i) + B_i(x-t_i)^2 + A_i(x-t_i)^3$. The resulting piecewise C^2 function, $S(x)$, is then called a natural cubic interpolation spline.

Test Cases:

- Test 1:
  Input: $(1,2),(2,3),(3,5)$
  Result: $S(x) =$
  $$S_0(x) = 2.0 + 0.75(x-1.0) + 0.0(x-1.0)^2 + 0.25(x-1.0)^3 \quad (0.0 \leq x \leq 1.0)$$
  $$S_1(x) = 3.0 + 1.5(x-2.0) + 0.75(x-2.0)^2 + -0.25(x-2.0)^3 \quad (1.0 \leq x \leq 2.0)$$
- Test 2: (Textbook Problem 9.2.32)
  Input: $(1,0),(2,1),(3,0),(4,1),(5,0))$
  Result: $S(x) =$
  $$S_0(x) = 0.0 + 1.714\ldots(x-1.0) + 0.0(x-1.0)^2 + -0.714\ldots(x-1.0)^3 \quad (0.0 \leq x \leq 1.0)$$
  $$S_1(x) = 1.0 + -0.428\ldots(x-2.0) + -2.142\ldots(x-2.0)^2 + 1.571\ldots(x-2.0)^3 \quad (1.0 \leq x \leq 2.0)$$
  $$S_2(x) = 0.0 + 1.110\ldots E-16(x-3.0) + 2.571\ldots(x-3.0)^2 + -1.571\ldots(x-3.0)^3 \quad (2.0 \leq x \leq 3.0)$$
  $$S_3(x) = 1.0 + 0.428\ldots(x-4.0) + -2.142\ldots(x-4.0)^2 + 0.714\ldots(x-4.0)^3 \quad (3.0 \leq x \leq 4.0)$$
- Test 3: (Textbook Problem 9.2.41)
  Input: $(0,1),(1,2),(2,3),(3,4),(4,5)$
  Result: $S(x) =$
  $$S_0(x) = 1.0 + 1.0(x-0.0) + 0.0(x-0.0)^2 + 0.0(x-0.0)^3 \quad (0.0 \leq x \leq 1.0)$$
  $$S_1(x) = 2.0 + 1.0(x-1.0) + 0.0(x-1.0)^2 + 0.0(x-1.0)^3 \quad (1.0 \leq x \leq 2.0)$$
  $$S_2(x) = 3.0 + 1.0(x-2.0) + 0.0(x-2.0)^2 + 0.0(x-2.0)^3 \quad (2.0 \leq x \leq 3.0)$$
  $$S_3(x) = 4.0 + 1.0(x-3.0) + 0.0(x-3.0)^2 + 0.0(x-3.0)^3 \quad (3.0 \leq x \leq 4.0)$$