Topic : Additional Interpolation Topics
Simulation : The Need for Spline Interpolation
Language : Mathcad 2001
Authors : Nathan Collier, Autar Kaw, Ginger Fisher
Date : 25 June 2002
Abstract : This simulation shows the need for spline interpolation as opposed to using polynomial interpolation. The function chosen was first used by Runge in 1901 when he wanted to show that higher order interpolation is a bad idea. He took a simple looking function f(x)=1/(1+25x2) and chose equidistantly spaced points to interpoalte the function.

INPUTS: Enter the following

Enter the number of equidistantly spaced points in [-1,1]

$n := 9$

SOLUTION

$f(x) := \dfrac{1}{1 + 25 \cdot x^2}$ $\qquad x := -1, -.99 .. 1$

When n is given, this function returns a matrix containing sequential x values

$$x_{points}(n) := \left| \begin{array}{l} \text{for } i \in 0 .. n-1 \\ \quad c_i \leftarrow \dfrac{2}{n-1} \cdot i - 1 \\ c \end{array} \right.$$

$Y(n) := f\left(x_{points}(n)\right)$

$X(n) := x_{points}(n)$

When x and y data and order is given, this function constructs the matrix whose inverse is needed to find coefficients of the polynomial which approximates the data.

$$M(n, X, Y) := \left| \begin{array}{l} \text{for } i \in 0 .. n-1 \\ \quad \text{for } j \in 0 .. n-1 \\ \quad\quad a_{i,j} \leftarrow \left(X_i\right)^j \\ a \end{array} \right.$$

This function generates the coefficients for the polynomial that approximates the x and y data.

$$a(n) := M(n, X(n), Y(n))^{-1} \cdot Y(n)$$

When given the polynomial order and specific x, this function uses the above calculated coefficients to calculate the approximated f(x)

```
f_polynomial(n, x) := | c ← 0
                      | for  i ∈ 0 .. n − 1
                      |    | d ← a(n)_i · x^i + c
                      |    | c ← d
                      | d
```

$$S := \text{cspline}(X(n), Y(n))$$

$$f_{spline}(x) := \text{interp}(S, X(n), Y(n), x)$$

Runge's function interpolated using polynomial interpolation and cubic spline interpolation