Differentiation of Discrete Functions

Ana Catalina Torres, Autar Kaw

University of South Florida

United States of America

kaw@eng.usf.edu

Introduction

This worksheet demonstrates the use of Maple to illustrate the differentiation of discrete functions using:

a) Forward Divided Difference Method. To find the derivative of a function given at discrete $n+1$ data points (x_1,y_1), (x_2,y_2),...,(x_n,y_n), the value of the $f'(x)$ for $x_i \leqslant x \leqslant x_{i+1}$, $i=1,...,n-1$, is given by

$$f'(x_i) \approx \frac{f(x_i + 1) - f(x_i)}{x_i + 1 - x_i}$$

b) By differentiation of a second order interpolated polynomial

c) By differentiation of a polynomial using all data points

Initialization

```
> restart;
   with (plots) :
  with(CurveFitting) :
```

Section 1: Input

The following simulation approximates the first derivative of a discrete function using Forward Divided Difference. The user inputs are

a) Data with x and y values. At least 3 data points are required. b) point at which the derivative is to be found, xv

The outputs include

a) approximate value of the derivative at the given point using Forward Divided Difference

b) approximate value of the derivative at the given point by differentiating a second order polynomial found using the three closest values to xv, and that bracket xv.

c) approximate value of the derivative at the given point using all data points and differentiating a n-1^{th} order polynomial that goes through the n data points

Data points, y vs. x

```
> x := [10, 15, 20, 21, 25];
  y := [100, 225, 400, 441, 625];
```

$$x := [10, 15, 20, 21, 25]$$
$$y := [100, 225, 400, 441, 625] \qquad \textbf{(3.1)}$$

Value of x at which $f\,'(x)$ is desired, xv

```
> xv := 11;
```

$$xv := 11 \qquad \textbf{(3.2)}$$

This is the end of the user section. All the information must be entered before proceeding to the next section. Re-execute the program.

Section 2: Calculation

The following loop estimates the solution of first derivate of a discrete function at a point xv using Forward Divided Difference method. The loop above checks if the value at which the solution is desired is between $x[1]$ and $x[n]$. If the value is between $x[1]$ and $x[n]$, then the slope from the closest points that bracket the value is found. The value of the slope is the value of the derivative at that point.

n = number of data points

```
> n := nops(x) :
  if (x[1] ≤ xv ≤ x[n]) then
   for i from 1 by 1 to n do
    if (x[i] ≤ xv < x[i+1]) then
    Av := (y[i+1] - y[i]) / (x[i+1] - x[i]);
    end if:
   end do:
  else
   print(Point where derivative was requested is outside the
     domain of x)
  end if:
```

The next method takes the three closest points to the given value to find a second order polynomial and differentiates it to find the value of $f'(x)$ at $x=xv$.

```
> if (x[1] ≤ xv ≤ x[n]) then
   if (x[1] ≤ xv < x[2]) then
    b := spline([x[1], x[2], x[3]], [y[1], y[2], y[3]], a) :
    SecDev := evalf(subs(a = xv, diff(b, a))) :
   end if:
   if (x[n-1] ≤ xv < x[n]) then
    b := spline([x[n-2], x[n-1], x[n]], [y[n-2], y[n-1], y[n]], a) :
    SecDev := evalf(subs(a = xv, diff(b, a))) :
   else for i from 2 by 1 to n-2 do
    if (x[i] ≤ xv < x[i+1]) then
     if (abs(x[i+2]-xv) ⩽ abs(xv - x[i-1])) then
      b := spline([x[i], x[i+1], x[i+2]], [y[i], y[i+1], y[i+2]],
     a) :
      SecDev := evalf(subs(a = xv, diff(b, a))) :
```

```
        else
         b := spline([x[i-1], x[i], x[i+1]], [y[i-1], y[i], y[i+1]],
        a) :
         SecDev := evalf(subs(a = xv, diff(b, a))) :
        end if:
      end if:
    end do:
    end if:
  end if:
```

Next, all data points are going to be used to find a *n-1* order polynomial. At the end, the polynomial is going to be differentiated and the value at which the derivative is wished to be found is going to be found.

```
> Poly := interp(x, y, a) :
  DerivPoly := evalf(subs(a = xv, diff(Poly, a))) :
```

Section 3: Spreadsheet

The next table shows the approximate values of the derivative at the given point using Forward Divided Difference, second and *n-1th* order polynomials.

```
> with(Spread) :
   tableoutput := CreateSpreadsheet("Differentiation of Discrete Functions") :
   SetCellFormula(tableoutput, 2, 1, "f'(x)") :
   SetCellFormula(tableoutput, 1, 2, "Straight Line") :
   SetCellFormula(tableoutput, 1, 3, "Second Order") :
   SetCellFormula(tableoutput, 1, 4, "n-1th Order") :
   SetCellFormula(tableoutput, 2, 2, evalf(Av)) :
   SetCellFormula(tableoutput, 2, 3, SecDev) :
   SetCellFormula(tableoutput, 2, 4, DerivPoly) :
   EvaluateSpreadsheet(tableoutput) :
```

Differentiation of Discrete Functions

	A	B	C	D
1		"Straight Line"	"Second Order"	"n-1th Order"
2	"f'(x)"	25.	22.80000000	22.

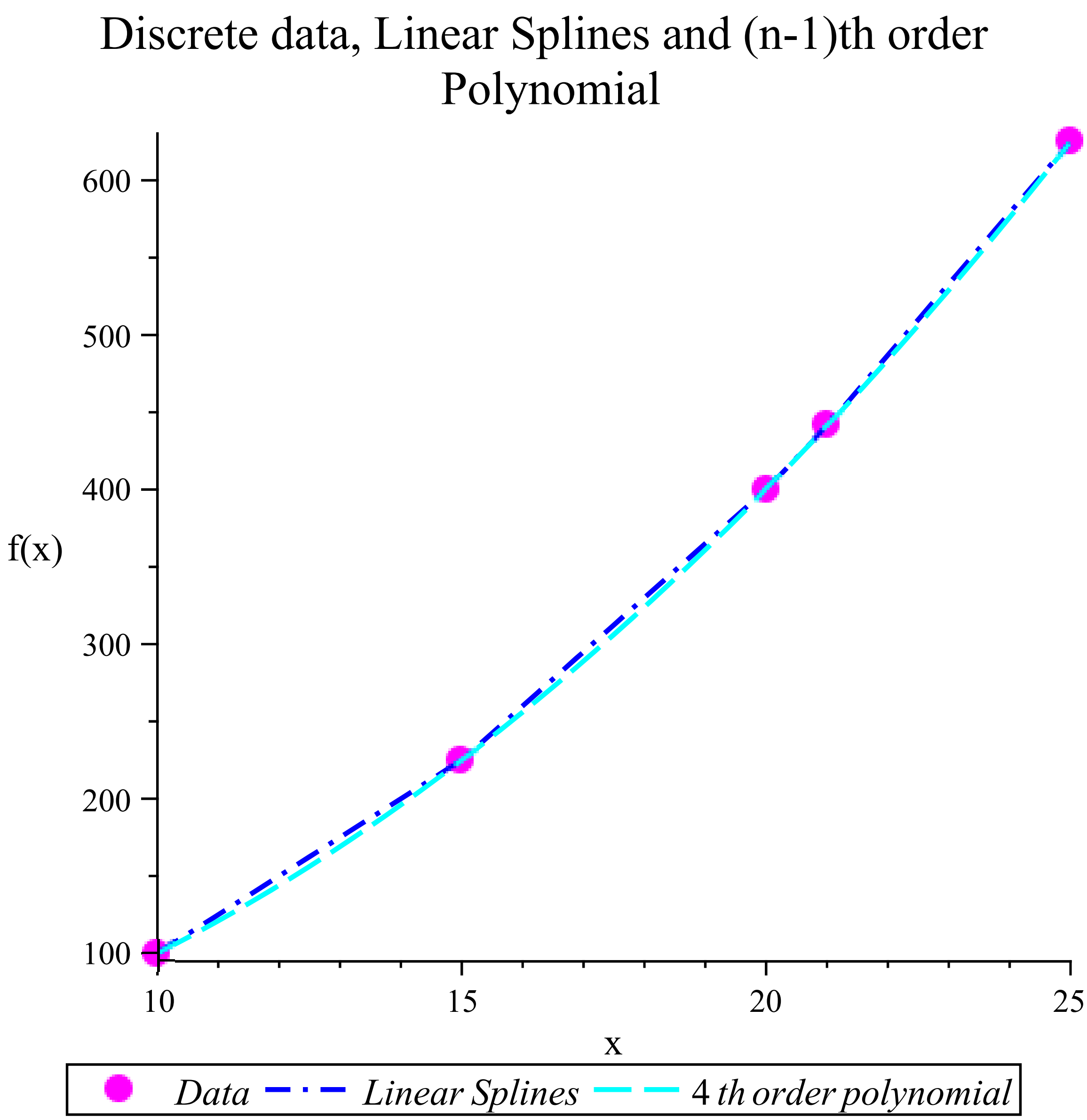
(5.1)

Section 5: Graphs

The following graph shows the discrete data, linear splines and the *n-1*th order polynomial.

```
> data := [seq([x[i], y[i]], i = 1 ..n)] :
  plot([data, spline(x, y, a, linear), interp(x, y, a)], a = x[1]
     ..x[n], style = [point, line, line], color = [magenta, blue,
```

```
cyan], thickness = 2, linestyle = [3, 4], symbol = solidcircle,
 symbolsize = 20, labels = ["x", "f(x)"], legend = [Data,
Linear Splines, (n − 1) th order polynomial], title
 = "Discrete data, Linear Splines and (n-1)th order Polynomial", titlefont
 = [TIMES, 15], labelfont = [TIMES, ROMAN, 12]);
```



>

References

Numerical Differentiation of Discrete Functions.
See http://numericalmethods.eng.usf.edu/mws/gen/02dif

Questions

1. The thermal expansion coefficient of steel is a function of temperature. Find the rate of change of the thermal expansion coefficient as a function of temperature at T=-200 F. Is this rate of change at T=-200 F more or less than that at T=50 F? Use Forward Divided Difference to answer this question.

2. The distance traveled by a rocket is given as a function of time

t, s	0	10	20	30	40
x, miles	0	16	28	39	53

Find the rocket velocity and acceleration at t=25s using numerical differentiation. Use all three methods illustrated in the worksheet.

Conclusions

The more data points taken to obtain the first derivative of a discrete function, more accurate the approximate value is. However, the more data points taken may result in oscillatory behavior generally observed with higher order interpolation. See http://numericalmethods.eng.usf.edu/mws/gen/05inp/mws_gen_inp_spe_higherorder.pdf