**Numerical Methods for Science and Engineering**

**Lecture Note 6**

**Numerical Differentiation**

**6.1 Introduction**

Numerical differentiation is the process of finding derivatives numerically for a function whose values are given in data form generated from an experiment. For evenly distributed data points and if we need the derivative at data points we may use the derivative formulas called finite differences. When the data points are not even or required derivatives are at points other than data points, we may use interpolating polynomials.

Numerical differentiation formulas can be derived by using the Taylor series expansion or by differentiating the interpolating polynomials. Here we shall consider both way of deriving the derivative formulas.

**6.2 Method of Numerical Differentiation**

Recall the definition of the derivative of a function



For finite ,



To find the derivative at , we choose another point  ahead of *x*i . This gives two point forward difference formula

 (1)

If  is chosen as a negative number, say  (*h*> 0), we have

 (2)

This is backward difference formula for first derivative.

Adding Eq.(1) and Eq.(2), we have

 (3)

which is a-point central difference formula for first derivative.

**6.3 Derivative Formula from Taylor Series**

For clear idea about the different formulas and their order of errors we may use the Taylor series expansion of .

From Taylor series expansion for *h* > 0, we have

 (1)

 (2)

From the expansion of , we have



which leads to the two-pont forward difference formula for  as



where the error series is



From the expansion of , we have 2-point backward difference formula



with error term 

In the two point formula the error series is of the form



where *a*’s does not depend on *h*.

By subtraction, we obtain



This leads to the 3-point central formula for approximating 



With 

Adding the Taylor series for  and , we get



When this is rearranged, we get 3-point central difference formula for 



where the error series is 

In the three point central difference formula the error series is of the form



**6.4 Formulas for Computing Derivatives**

**First Derivatives**

,  2-points forward difference

,  2-points backward difference

,  3-points central difference

,  3-points forward difference

,  3-points backward difference

,  5-points central difference

**Second Derivatives**

,  3-point central difference

,  3-point forward difference

,  3-point backward difference

, 

5-point central difference

**6.5 Richardson Extrapolation**

If the two approximations of order  for  are  and , then the Richardson’s extrapolated estimate  of *M* can be written as

 (1)

 (2)

where it is assumed that the constant multiplicative factor *A* is same for both cases.

Subtracting (1) from (2),



or 

Substituting in (1), we have

.

which can be written as



or



where .

This is known as the **Richardson extrapolation** formula.

Lower order formula and Richardson extrapolation can be used to deduce the higher order formula. For convenience we have used the notation to indicate clearly the approximation of  with step size *h* and .

Thus the 3-point central difference formula for first derivative will ne written as



**Example 6.1** The values of distance at various times are given below

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Time (*t*) | 4 | 6 | 8 | 10 | 12 |
| Distance(*s*) | 7.38 | 12.07 | 18.37 | 26.42 | 36.40 |

The speed and acceleration can be calculated by  and acceleration  .

1. Using three point central difference formula estimate the speeds at (i) , (ii) , and (iii) .
2. Using two point formulas and extrapolation estimate the speeds at (i) , and (ii) .
3. Use three points central difference formula and extrapolation to estimate speed at .
4. Use three points central or forward or backward formula to estimate the accelerations at (i) , (ii) , and (iii) .
5. Write down MATLAB code to estimate the speed and acceleration at time using three point central difference formulas.
6. Use MATAB functions “**sp=spline(x,y)**”, “**fnder(sp, dorder)**” and “**fnval(sp, xo)**” to estimate the speed and acceleration at time .

**Solution**

(a) Three point central derivative formula for first derivative is

1. Speed at
2. Speed at *.* Here .
3. Speed at *.* Here .

(b)

(i) For , we have to use forward difference formula

Speed at we need to use .

Extrapolated value is

(ii) For , we have to use backward difference formula

Speed at we need to use .

Extrapolated value is

(c) Three points central difference formula is

Speed at we need to use .

Extrapolated value is

(d)

(i) For , we have to use central difference formula (it gives better approximation).

Three point central derivative formula for second derivative is

Acceleration at is

(ii) For , three point forward difference formula for second derivative is

Acceleration at is

(iii) For , three point backward difference formula for second derivative is

Acceleration at is

(d)

>> clear

>> x=[6 8 10];

>> y=[12.07 18.37 26.42];

>> h=x(2)-x(1);

>> D1=(y(3)-y(1))/(2\*h);

>> D2=(y(3)-2\*y(2)+y(1))/h^2;

(e) >> clear

>> x=[4 6 8 10 12];

>> y=[7.38 12.07 18.37 26.42 36.40];

>> % syntax for derivative is “**fnder(f, dorder)**”

>> sp=spline(x,y); % generates spline function sp

>> D1sp=fnder(sp,1); % generate first derivative of spline function sp

>> ValD1=fnval(D1sp,[8.4, 11]) % gives values from D1sp

ValD1 =

3.7501 4.9860

>> D2sp=fnder(sp, 2); % gererates second derivative

>> ValD2=fnval(D2sp, [8.4, 11]) % gives values from D2sp

ValD2 =

0.4445 0.5063

**Exercise 6**

**Numerical Differentiation**

1. The distance *s* of a runner from a fixed point is measured (in metres) at intervals of half a second. The data obtained is

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Time t (s) | 0 | 0.5 | 1 | 1.5 | 2 |
| Distance s (m) | 0 | 3.65 | 6.8 | 9.9 | 12.15 |

1. Use two points difference formulas to approximate the runner’s speed at times t = 0s and t = 2s.
2. Use three points central difference formula to approximate the runner’s speed at times t = 0.5s and t = 1.25s.
3. Use three points central difference formula to approximate the runner’s acceleration at times t = 1 s.
4. Write down MATLAB code to estimate the speed and acceleration at time at time using three point central difference formulas.

3. The speed *v* (in m/s) of a rocket measured at half second intervals is

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Time *t* (s) | 0 | 0.5 | 1 | 1.5 | 2 |
| speed *v* (in m/s) | 0 | 11.860 | 26.335 | 41.075 | 59.05 |

1. Use central difference formula to approximate the acceleration of the rocket at times t = 1 s and t = 1.75s.
2. Use two-point backward difference formula and Richardson extrapolation to estimate the acceleration of the rocket at time t = 2 s.
3. Use three-point central difference formula and extrapolation to estimate the acceleration of the rocket at time t = 1 s.
4. Use MATAB to estimate the acceleration of the rocket at time using spline interpolation.

4. The voltage in an electric circuit obeys the differential equations

, where *R* is the resistance and *L* is the inductance. Use *L* =0.05,

*R*= 2 and the value of  in the table’

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *t* | 1.0 | 1.1 | 1.2 | 1.3 | 1.4 |
| *i*(*t*) | 8.2277 | 7.2428 | 5.9908 | 4.5260 | 2.9122 |

1. Find  using three point central difference formula and extrapolation to compute *E(1.2)*
2. Compare your result with the exact solution.
3. Use MATAB to estimate at each value of *t* using spline interpolation.

Write down the MATLAB commands to find the corresponding voltage .

5. The distance traveled by an object is given in the table below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *t (*s*)* | 8 | 9 | 10 | 11 | 12 |
| *s*(*t*) (m) | 17.453 | 21.460 | 25.752 | 30.302 | 35.084 |

The speed and acceleration can be calculated by and acceleration .

1. Using three point central difference formula estimate the speeds at (i) , and (ii)  .
2. Using two point formulas estimate the speeds at (i) , and (ii) .
3. Use three points central difference formula and extrapolation to estimate speed at .
4. Use three points central or forward or backward difference formula to estimate the accelerations at (i) , (ii) and (iii) .
5. Use MATAB to estimate the speed and acceleration at time t using spline interpolation.

6. The table below shows the values of at different values of *x*:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *x* | 0.8 | 1.0 | 1.2 | 1.4 | 1.6 |
| *f*(*x*) | 0.954 | 1.648 | 2.623 | 3.947 | 5.697 |

1. Using three point central difference formula estimate
2. Using two point forward difference formula and extrapolation estimate .
3. Use three points central difference formula and extrapolation to estimate .
4. Use three points backward formula to estimate .
5. Write down MATAB codes using “**sp=spline(x,y)**”, “**fnder(sp, dorder)**” and “**fnval(sp, xo)**” to estimate the values of at

.

7. A rod is rotating in a plane. The follow table gives the angle ( in radians) through which the rod has turned for various values of time .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Time t (s) | 0 | 0.2 | 0.4 | 0.6 | 0.8 |
| Angle | 0 | 0.12 | 0.49 | 1.12 | 2.02 |

1. Use two points difference formula to approximate the angular velocity of the rod at t = 0.6s.
2. Use three points central difference formula to approximate the angular velocity of the rod at t = 0.6s.
3. Use three points central difference formula to approximate the angular acceleration of the rod at t = 0.6s.
4. Write down MATLAB code to estimate the speed and acceleration at time using three point central difference formulas.

8. The table below gives the results of an observation, is the observed temperature in degree celsius of a vessel of cooling water, is the time in minutes from the beginning of observation.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Time t (min) | 1 | 3 | 5 | 7 | 9 |
| temperature | 85.3 | 74.5 | 67.0 | 60.5 | 54.3 |

1. Use two points difference formula to approximate the rate of cooling at and
2. Use three points central difference formula to approximate the rate of cooling at and
3. Write down MATLAB code to estimate the approximate rate of cooling at and three point central difference formulas.