4.4 DERIVATIVES USING NEWTON'S FORWARD DIFFERENCE INTERPOLATION FORMULA

(a) Elementary Approach (Using Interpolation Formula)

First-order Derivative

Newton's forward difference formula (3.2) is written as follows:

$$f_p = f_0 + p\Delta f_0 + \frac{1}{2}\left(p^2 - p\right)\Delta^2 f_0 + \frac{1}{6}\left(p^3 - 3p^2 + 2p\right)\Delta^3 f_0$$
$$+ \frac{1}{24}\left(p^4 - 6p^3 + 11p^2 - 6p\right)\Delta^4 f_0 + \cdots$$

Differentiating this formula with respect to p, we get,

$$\frac{df_p}{dp} = \Delta f_0 + \frac{1}{2}(2p-1)\Delta^2 f_0 + \frac{1}{6}\left(3p^2 - 6p + 2\right)\Delta^3 f_0$$
$$+ \frac{1}{24}\left(4p^3 - 18p^2 + 22p - 6\right)\Delta^4 f_0 + \ldots$$

Since, $f'_p = \frac{1}{h}\frac{df_p}{dp}$, we get the first derivative as follows:

$$f'_p = \frac{1}{h}\left\{\Delta f_0 + \frac{1}{2}(2p-1)\Delta^2 f_0 + \frac{1}{6}\left(3p^2 - 6p + 2\right)\Delta^3 f_0 + \frac{1}{12}\left(2p^3 - 9p^2 + 11p - 3\right)\Delta^4 f_0 + \cdots\right\} \quad \ldots (4.5)$$

Higher-order Derivatives

The method can be extended to find the higher-order derivatives. Differentiating (4.5) w.r.t.p., we get,

$$f''_p = \frac{1}{h}\frac{df'_p}{dp}$$
$$= \frac{1}{h^2}\left\{\Delta^2 f_0 + \frac{1}{6}(6p-6)\Delta^3 f_0 + \frac{1}{12}\left(6p^2 - 18p + 11\right)\Delta^4 f_0 + \cdots\right\}$$
$$= \frac{1}{h^2}\left\{\Delta^2 f_0 + (p-1)\Delta^3 f_0 + \frac{1}{12}\left(6p^2 - 18p + 11\right)\Delta^4 f_0 + \cdots\right\} \quad \ldots (4.6)$$

Similarly, $f_p''' = \frac{1}{h}\frac{d f_p''}{dp}$

$$= \frac{1}{h^3}\left\{\Delta^3 f_0 + \frac{1}{2}\left(2p^2 - 3\right)\Delta^4 f_0 + \cdots\right\} \qquad \ldots (4.7)$$

The same procedure can be repeated to calculate the derivatives of any order.

Special Cases

Formulas for derivatives when

(i) $p = 0$

$$f_0' = \frac{1}{h}\left\{\Delta f_0 - \frac{1}{2}\Delta^2 f_0 + \frac{1}{3}\Delta^3 f_0 - \frac{1}{4}\Delta^4 f_0 + \cdots\right\}$$

$$f_0'' = \frac{1}{h^2}\left\{\Delta^2 f_0 - \Delta^3 f_0 + \frac{11}{12}\Delta^4 f_0 - \cdots\right\}$$

$$f_0''' = \frac{1}{h^3}\left\{\Delta^3 f_0 - \frac{3}{2}\Delta^4 f_0 - \cdots\right\}$$

(ii) $p = \frac{1}{2}$

$$f_{\frac{1}{2}}' = \frac{1}{h}\left\{\Delta f_0 - \frac{1}{24}\Delta^3 f_0 + \frac{1}{24}\Delta^4 f_0 - \cdots\right\}$$

$$f_{\frac{1}{2}}'' = \frac{1}{h^2}\left\{\Delta^2 f_0 - \frac{1}{2}\Delta^3 f_0 + \frac{7}{24}\Delta^4 f_0 - \cdots\right\}$$

$$f_{\frac{1}{2}}''' = \frac{1}{h^3}\left\{\Delta^3 f_0 - \Delta^4 f_0 - \cdots\right\}$$

(b) Derivatives Using Difference Operators

First-order Derivative

We can also derive the above formulas using difference operators.

Since, $e^{hD} = E = 1 + \Delta$, taking logarithm of both sides, we get,

$$hD = \log(1 + \Delta).$$

Expanding the right hand side, we get,

$$hD = \Delta - \frac{\Delta^2}{2} + \frac{\Delta^3}{3} - \frac{\Delta^4}{4} + \dots$$

$$D = \frac{1}{h}\left\{\Delta - \frac{\Delta^2}{2} + \frac{\Delta^3}{3} - \frac{\Delta^4}{4} + \cdots\right\}$$

or, $$Df_0 = \frac{1}{h}\left\{\Delta - \frac{\Delta^2}{2} + \frac{\Delta^3}{3} - \frac{\Delta^4}{4} + \cdots\right\}f_0$$

Since, $Df_0 = f_0'$, we can write the above as follows:

$$f_0' = \frac{1}{h}\left\{\Delta f_0 - \frac{1}{2}\Delta^2 f_0 + \frac{1}{3}\Delta^3 f_0 - \frac{1}{4}\Delta^4 f_0 + \cdots\right\} \quad \dots (4.8)$$

Higher-order Derivatives

$$f_0'' = Df_0 \times Df_0$$

$$= \frac{1}{h^2}\left\{\Delta - \frac{\Delta^2}{2} + \frac{\Delta^3}{3} - \frac{\Delta^4}{4} + \cdots\right\}^2 f_0$$

$$= \frac{1}{h^2}\left\{\Delta^2 f_0 - \Delta^3 f_0 + \frac{11}{12}\Delta^4 f_0 + \cdots\right\} \quad \dots (4.9)$$

$$f_0''' = Df_0 \times Df_0 \times Df_0$$

$$= \frac{1}{h^3}\left\{\Delta - \frac{\Delta^2}{2} + \frac{\Delta^3}{3} - \frac{\Delta^4}{4} + \cdots\right\}^3 f_0$$

$$= \frac{1}{h^3}\left\{\Delta^3 f_0 - \frac{3}{2}\Delta^4 f_0 + \cdots\right\} \quad \dots (4.10)$$

The rest of the formulas in this chapter will be derived using the elementary approach (i.e., using interpolation formulas).

It must be recognized that numerical differentiation is subject to considerable error. It should also be noted that all these formulas involve division of a combination of differences (which are prone to loss of significance or cancellation errors, especially if h is small) by a positive power of h. Consequently, if we want to reduce the round-off errors, we should use a large value of h. In brief, large errors may occur in numerical differentiation, based on direct polynomial approximation, so that an error check is always advisable. There are alternative methods, based on polynomials, which use more sophisticated procedures such as least-squares or mini-max, and other alternatives involving other basis functions (for example, trigonometric functions). However, the best policy is probably to use numerical differentiation only when it cannot be avoided!

Example 1 (a) Use the following table of values,

x	0.0	0.5	1.0	1.5	2.0
f(x)	2.0286	2.4043	2.7637	3.1072	3.4350

to compute, $f'(.25)$, $f''(.25)$ and $f'''(.25)$.

(b) Write a computer program to implement the method for computing the first two derivatives.

Solution (a) $x_p = .25,\ \ h = 0.5,\ \ x_0 = 0.0$

$$p = \frac{(x_p - x_0)}{h} = \frac{(.25 - .00)}{.5} = 0.5$$

Difference Table

x	f(x)	Δ	Δ^2	Δ^3	Δ^4
$x_0 = 0.0$	2.0286				
		3757			
0.5	2.4043		−163		
		3594		4	
1.0	2.7637		−159		−2
		3435		2	
1.5	3.1072		−157		
		3278			
2.0	3.4350				

Substituting values of p and the required differences in (4.5), we get,

$$f'(.25) = f'_{.5} = \frac{1}{.5}\Big\{.3757 + \frac{1}{2}(2\times 0.5 - 1)(-.0163) + \frac{1}{6}(3\times .5^2 - 6\times .5 + 2)\times .0004$$

$$+\frac{1}{12}(2\times .5^3 - 9\times .5^2 + 11\times .5 - 3)\times -.002\Big\}$$

$$= \frac{1}{.5}\{.3757 - .000 - .0000 - .0000\} = 0.7514$$

Substituting values of p and the required differences in (4.6), we get,

$$f''(.25) = f''_{.5} = \frac{1}{.5^2}\Big\{-.0163 + (.5 - 1)\times .0004 + \frac{1}{12}(6\times .5\times .5 - 18\times .5 + 11)\times -.0002\Big\}$$

$$= \frac{1}{.25}\{.0163 - .0002 - .0001\}$$

$$= \frac{1}{.25} \times (-.0166) = -0.0664$$

Similarly, substituting values of p and the required differences in (4.7), we get,

$$f'''(.25) = f_{.5}''' = \frac{1}{.5^3}\left\{.0004 + \frac{1}{2}(2 \times .5 - 3) \times -.0002\right\}$$

$$= \frac{1}{.125}\{.0004 + (-.0002)\} = 0.0048$$

It is worthwhile to remember that the derivatives at non-tabular points can be obtained by extrapolation.

(b) Computer Program No. 5: Numerical Differences

```
# include<iostream.h>
# include<conio.h>

class NewForwDiff
  {
   public:
          NewForwDiff( );
          void input( );
          void result( );
          void get_1st_der( );
          void get_2nd_der( );
   private:
          int degree,values,actual_ degree;
          float delta[10],xp,p,x[10],fx[10],h;
  }

   NewForwDiff::NewForwDiff( )
   {
      clscr( );
      cout<<"\n\t\tFROM NEWTON'S FORWARD DIFFERENCE FORMULA\n\n";
      degree=values=xp=actual_degree=0;
      p= -2;
      for(int i=0; i<10; i++)
             delta[i]=x[i]=fx[i]=0.0;
   }
```

```
void NewForwDiff::input( )
{
        cout<<"How many values you want for x?\t";
        cin>>values;
        cout<<"Upto what power of Delta:\t";
        cin>>degree;
//      degree=degree>4 ||degree<1 ? 4 : degree;
        cout<<"\n Value of Xp:\t";
        cin>>xp;
        for(int i=0;i!=values;i++)
        {
            cout<<"\nEnter  X"<<i+1<<":\t";
            cin>>x[i];
            cout<<"Enter  F("<<i+1<<"):\t";
            cin>>fx[i];
        }
}

void NewForwDiff::result( )
{
        clscr( );
        cout<<"\nX\t";
        for(int i=0;i!=values;i++)
              cout<<"\t"<<x[i];
        cout<<endl;
        cout<<"\nF(x)\t";
        for(i=0;i!=values;i++)
             cout<<"\t"<<fx[i];
        cout≤endl;
        h=x[1]-x[0];
        for(int j=0;temp= −1;j<values && (p<0 || p>1); temp=j, j++)
              p=(xp-x[j])/h;

        cout<<"\nValue of P is :\t"<<p<<"\n";
        for(actual_degree=1, j=values; actual_degree<=degree&&j>1,actual_
            degree++, j--)
        {
             cout<<"\n\nDelta power "<<actual_degree<<":\t";
             for(int k=0;k<j-1;k++)
             {
                 fx[k]=fx[k+1]-fx[k];
                 cout<<fx[k]<<"\t";
             }
```

```
            delta[actual_degree-1]=fx[temp];
            cout<<delta[actual_degree-1];
        }
        get_1st_der( );
        get_2nd_der( );
}

void NewForwDiff::get_1st_der( );
{
      float parray[]={1,2*p-1,3*p*p-6p+2,4*p*p+18*p*p+22*p-6},
                  div[]={1,2,6,24},
                  ans=0;

      for(int i=0;i<actual_degree;i++)
            ans+=delta[i]*parray[i]/div[i];
      cout<<"\n\n\nf '("<<xp<<"):\t"<<ans/h;
}

   void NewForwDiff::get_2nd_der( );
   {
      float parray[]={1,p-1,6*p*p-18*p+11},
                  div[]={1,1,12},
                  ans=0;
      for(int i=1;i<actual_degree;i++)
            ans+=delta[i]*parray[i-1]/div[i-1];

      cout<<"\n\n\nf "("<<xp<<"):\t"<<ans/(h*h);
   }
void main (void)
{
   NewForwDiff obj;
   obj.input( );
   obj.result( );
   getch( );
}
```

Computer Output

FROM NEWTON'S FORWARD DIFFERENCE FORMULA

How many values you want for X? 5
Upto what power of Delta: 4

Value of Xp: .25

Enter X1: 0
Enter F(1): 2.0286

Enter X2: .5
Enter F(2): 2.4043

Enter X3: 1
Enter F(3): 2.7637

Enter X4: 1.5
Enter F(4): 3.1072

Enter X5: 2
Enter F(5): 3.435

X	0	0.5	1	1.5	2
F(X)	.2.0286	2.4043	2.7637	3.1072	3.435

Value of P is : 0.5

Delta Power 1:	0.3757	0.3594	0.3435	0.3278	0.375
Delta Power 2:	− 0.0163	− 0.0159	− 0.0157	0.0163	
Delta Power 3:	0.0004	0.0002	0.0004		
Delta Power 4:	− 0.0002	− 0.0002			

f′(0.25): 0.7512

f″(0.25): − 0.066232