Chap three

ShangXiaojin

2022 年 12 月 14 日

目录

1 Lagrange Polynomial & Neville's Iterated Interpolation & Divided Differences

1.1 Ex

1.1.1 Ex_1

$iff(0.1) = 0.62049958, f(0.2) = -0.28398668, f(0.3) = 0.00660095, f(0.4) = 0.24842440$,then , please obtain the approximations while$x = 0.25$

1.1.2 Ex_2

Use Neville's method to obtain the approximations for Lagrange interpolating polynomials of degrees one, two, and three to approximate each of the following:
$a.f(8.4), iff(8.1) = 16.94410, f(8.3) = 17.56492, f(8.6) = 18.50515, f(8.7) = 18.82091$
$b.f(-\frac{1}{3}), iff(-0.75) = -0.07181250, f(-0.5) = -0.02475000, f(-0.25) = 0.33493750, f(0) = 1.10100000$
$c.f(0.25), iff(0.1) = 0.62049958, f(0.2) = -0.28398668, f(0.3) = 0.00660095, f(0.4) = 0.24842440$
$d.f(0.9), iff(0.6) = -0.17694460, f(0.7) = 0.01375227, f(0.8) = 0.22363362, f(1.0) = 0.65809197$

1.2 code

To be frankly, I thougt I'd do this math problem all…and I did.

1.2.1 Preparations

In order to use iteration to complete this problem, I integrated all the information of this topic into one matrix.

```
clc , clear
format long
clc
clear
```

```
A= [8.1   8.3  8.6  8.7;
16.94410  17.56492  18.50515  18.82091;
-0.75  -0.5  -0.25  0;
-0.07181250  -0.02475000  0.33493750  1.1010000;
0.1  0.2  0.3  0.4;
0.62049958  -0.28398668  0.00660095  0.24842440;
0.6  0.7  0.8  1.0;
-0.17694460  0.01375227  0.22363362  0.65809197];
X = [8.4  -1/3  0.25  0.9];
Ans = zeros(4,3);
```

1.2.2 main function

Neville[1]

```
function f = Neville(X,F,a)
n = size(X,2);
Q = zeros(size(X,2));
for i = 1:n
    Q(i,1) = F(i);
end
for i = 2:n
    for j = 2:i
        Q(i,j) =( (a - X(i - j + 1))*Q(i,j-1) - (a - X(i))*Q(i-1,j-1))/(X(i)
    end
end
f = Q;
end
```

Lagrange

```
function g = Lagrange(X,F,a)
n = size(X,2);
```

[1]由于 Matlab 矩阵从 1 开始计数，对角标进行了适当的更改

```
L = zeros(1,n);
T = a*ones(1,n);
for i = 1:n
    temp = X(1,i)*ones(1,n);
    A = X;
    B = X;
    A(1,i) = a-1;
    B(1,i) = X(1,i) -1;
    L(1,i) = prod(T - A)/prod(temp - B);
end
g = dot(F,L);
end
```

Differences[2]

```
function t = Differences(X,F,a)
n = size(X,2);
x = a*ones(n);
t = F(1,1);
T = [F' zeros(n,n-1)];
for i = 2:n
    for j = 2:i
        T(i,j) = (T(i,j-1) - T(i-1,j-1))/(X(i) - X(i-j+1));
    end
end
for k = 2:n
    temp1 = x(1:k-1);
    temp2 = X(1:k-1);
    t = t + T(k,k)*prod(temp1 - temp2);
end
end
```

[2]由于 Matlab 矩阵从 1 开始计数，对角标进行了适当的更改

1.2.3 the body of code

Ex2

此处未按照作业要求对 1、(c) 进行求解二，而是按照该题原本要求作答，作答中由于已经将本题所有信息录入矩阵 A 之中，为方便起见进行两步运算：
第一步：利用迭代选取 x_i x_4 这 $4-i$ 个点运用 Neville 求 x 点函数近似值
第二步：根据 x 值，重新选取 x 附近的 x_i 点进行最终值的更正。具体代码如下：

```
temp1 = Neville(A(1,2:3),A(2,2:3),8.4);
temp2 = Neville(A(3,2:3),A(4,2:3),-1/3);
temp3 = Neville(A(5,2:3),A(6,2:3),0.25);

Ans(1,1) = temp1(2,2);
Ans(2,1) = temp2(2,2);
Ans(3,1) = temp3(2,2);
Ans
```

Ex1[3]

具体代码如下：

```
%f (0.25) if f (0.1) = 0.62049958, f (0.2) = -0.28398668, f (0.3) = 0.006600
tic,Lagrange(A(5,:),A(6,:),0.25),toc
tic,Neville(A(5,:),A(6,:),0.25),toc
tic,Differences(A(5,:),A(6,:),0.25),toc
```

[3]直接利用了矩阵 A 进行计算

1.3 Ans & Analyse

类型	耗时 (s)	结果
Lagrange Polynomial	0.020129	-0.210337221875000
Neville's Iterated Interpolation	0.015109	-0.210337221875000
Divided Differences	0.006151	-0.210337221875000

2 Hermite Polynomials

2.1 Ex

f, such that ,find P(x)

x	0.1	0.2	0.3	0.4
f(x)	-0.62049958	-0.28398668	0.00660095	0.24842440
$f'(x)$	3.58502082	3.14033271	2.66668043	2.16529366

2.2 code

2.2.1 main function

```
tic,Hermite([0.1 0.2 0.3 0.4],[-0.62049958 -0.28398668 0.00660095 0.24842440
],[3.58502082 3.14033271 2.66668043 2.16529366]),toc

function q = Hermite(X,F,F1)
n = size(X,2);
Q = zeros(2*n);
Z = zeros(1,2*n);
q = zeros(1,2*n);
for i = 1:n
    Z(1,2*i-1) = X(i);
    Z(1,2*i) = X(i);
    Q(2*i-1,1) = F(i);
    Q(2*i,1) = F(i);
    Q(2*i,2) = F1(i);
    if i > 1
        Q(2*i-1,2) = (Q(2*i-1,1) - Q(2*i-2,1))/(Z(1,2*i-1)-Z(1,2*i-2));
```

```
    end
for i = 3:2*n
    for j = 3:i
    Q(i,j) = (Q(i,j-1) - Q(i-1,j-1))/(Z(i)-Z(i-j+1));
    end
end
end
for i = 1:2*n
    q(i) = Q(i,i);
end
end
```

2.3 Ans

耗时 (s)	0.055735	
Ans	[-0.620499580000000	3.585020820000000
	-2.198918199999995	-0.490447000000094
	0.037205000000695	0.040474999992823
	-0.002527777740325	0.002962962817440]