DIS 006

Raehyun Kim*

1 Lagrange interpolating polynomials

lagInt.m

```
function [f, fh] = lagInt(xVal, fVal)
%sanity check length(xVal) == length(fVal)?
num = length(xVal);
syms x
% lag will include all lag interpolation poly
lag = cell(num,1);
for ielem = 1:num
    lag{ielem} = 1.0;
end
% contruct lag int poly
for ielem = 1:num
    for jelem = 1:num
        if ielem ~= jelem
            lag{ielem} = lag{ielem}*(x-xVal(jelem))/(xVal(ielem)-xVal(jelem));
        end
    end
end
% need to multiply the weights
for ielem = 1:num
    lag{ielem} = fVal(ielem)*lag{ielem};
end
f = 0.0;
for ielem = 1:num
    f = f+lag{ielem};
end
fh = matlabFunction(f);
end
```

*Department of Mathematics, University of California at Berkeley

2 Section 3.1. 6

```
xVal = [-1, -0.5, 0, 0.5];
fVal = [0.86199480, 0.95802009, 1.0986123, 1.2943767];
[f, fh] = lagInt(xVal, fVal);
disp(f)
disp(fh(0.25))

p = polyfit(xVal, fVal, length(xVal)-1);
disp(polyval(p,0.25))
```

3 Section 3.3. 8

```
clc; clear
xVal = [0.0, 0.1, 0.3, 0.6, 1.0];
fVal = [-6, -5.89483, -5.65014, -5.17788, -4.28172];
F = NDD1(xVal,fVal);
% print interpolation polynomial
syms x
f = 0.0;

for ielem=length(xVal)-1:-1:1
   f = F(ielem) + f .* (x-xVal(ielem));
end
disp(f)
```