```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Newton_interp.m
%--------------------------------------------------------------------------
% C Rocheleau, Colorado State University
% 9/23/23
%--------------------------------------------------------------------------
% This function performs Lagrange's method to create an interpolating
% polynomial from a list of given points and evaluates at points X
%--------------------------------------------------------------------------
% INPUTS
%   x_given: A vector of X positions of known points to use to find the
%       interpolating polynomial
%   y_given: A vector of Y positions of known points to use to find the
%       interpolating polynomial
%   X: Points at which to evaluate the interpolating polynomial
%--------------------------------------------------------------------------
% OUTPUTS
%   Y: Output of interpolating polynomial at given points
%   coeffs: {Optional} coefficients of interpolating polynomial
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Y, coeffs] = Newton_interp(x_given, y_given, X)
% Use Divided Differences to find coefficients for polynomial
coeffs = zeros(1,length(x_given));
for iOrder = 1:length(x_given)
    % Each order we go, we get one fewer divided difference
    temp = NaN(1,length(x_given) - iOrder + 1);
    if iOrder == 1
        temp = y_given;
    else
        for ii = 1:length(temp)
            temp(ii) = (divDiffs(ii+1) - divDiffs(ii))./ ...
                            (x_given(ii + iOrder - 1) - x_given(ii));
        end
    end
    divDiffs = temp;
    % All we need to store for coefficients is the divided difference with x_1
    coeffs(iOrder) = divDiffs(1);
end

% Use coefficients to calculate our y points using Hoerner's method
Y = Hoerner_poly(x_given, coeffs, X);
end


function Y = Hoerner_poly(x, coeffs, X)
    if length(x) ~= 1
        Y = coeffs(1) + (X - x(1)).*Hoerner_poly(x(2:end),coeffs(2:end),X);
    else
        Y = coeffs(1);
    end
end
```

```
Not enough input arguments.
Error in Newton_interp (line 24)
coeffs = zeros(1,length(x_given));
```

Published with MATLAB® R2022b