Numerical Methods: Computer Assignment #4

Hong Chenhui
drredthered@gmail.com

Zhejiang University — March 25, 2024

Info: Not specific value oriented.

1 Problem

- Consider the following function:
$$f(x) = \frac{x}{1+x^4}$$
Interpolating evenly using the lagrange polynomial as well as piecewise linear interpolation on x $\in [-5, 5]$. Compare the results.

1.1 Theoretical viewpoint

Question

A typical lagrangian polynomial intepolation method is defined as:

Algorithm 1: Lagrange Polynomial

Input: Data points $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$
Output: Lagrange polynomial $P(x)$

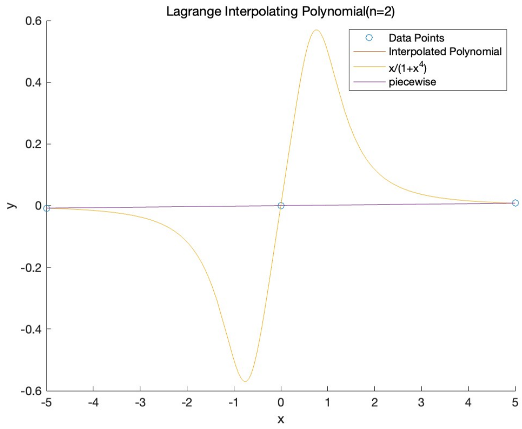
for $i = 0$ **to** n **do**
  $L_i(x) \leftarrow 1$;
  for $j = 0$ **to** n **do**
    if $i \neq j$ **then**
      $L_i(x) \leftarrow L_i(x) \cdot \frac{x-x_j}{x_i-x_j}$;
    end
  end
end
$P(x) \leftarrow 0$;
for $i = 0$ **to** n **do**
  $P(x) \leftarrow P(x) + y_i \cdot L_i(x)$;
end
return $P(x)$;

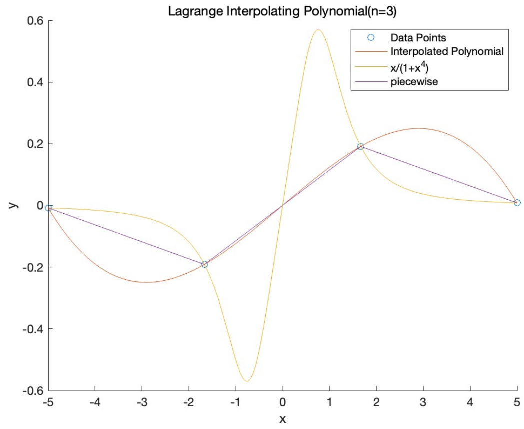
As defined, piecewise linear interpolation is just connecting those data points using straight line.

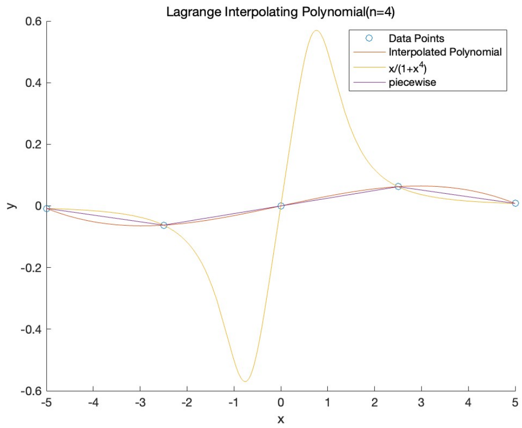
Algorithm 2: Piecewise Linear Interpolation

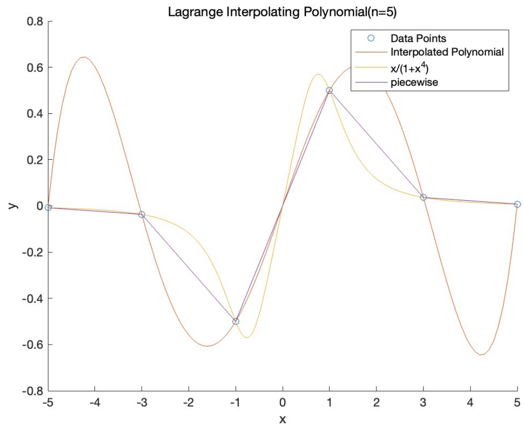
for $i = 0$ **to** $n - 1$ **do**
  $m \leftarrow \frac{y_{i+1}-y_i}{x_{i+1}-x_i}$;
  $b \leftarrow y_i - m \cdot x_i$;
  $y \leftarrow m \cdot x + b$;
end

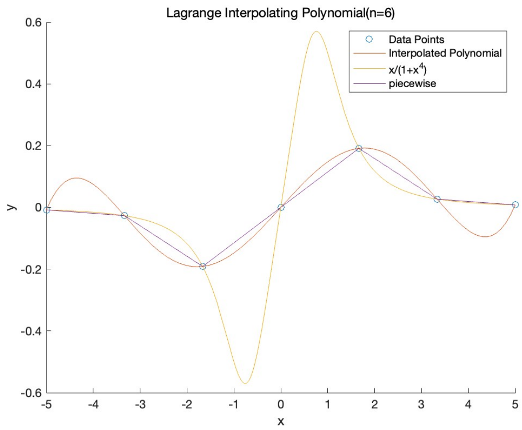
2 implementation

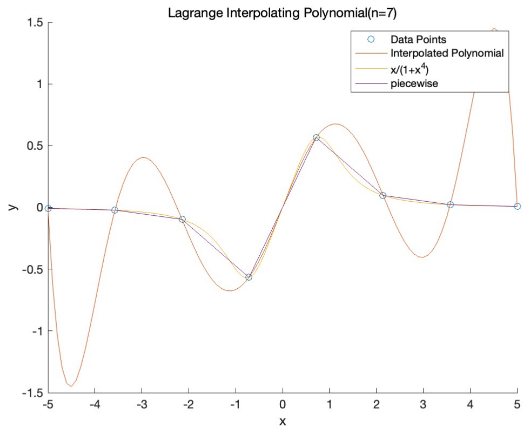


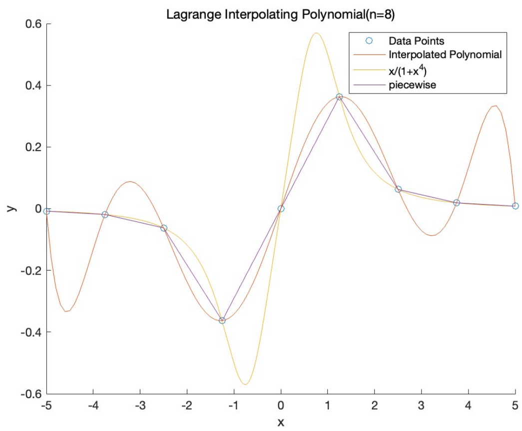


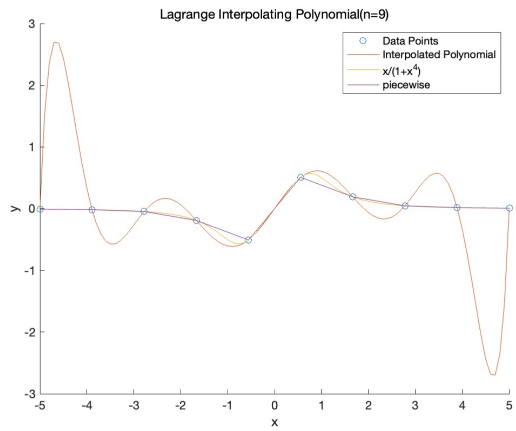


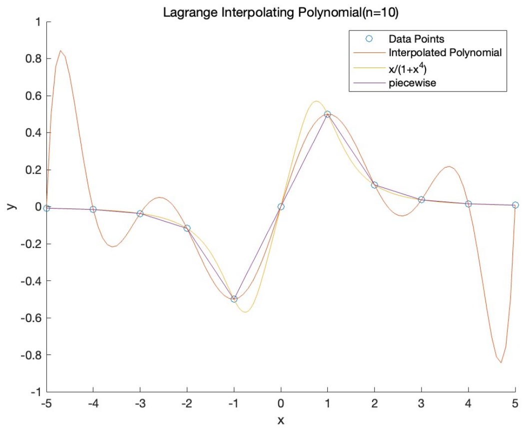












3 Analysis

Nothing noticable in Piecewise Linear Interpolation method. As expected, the discrepancy shortened with the increasing section number n.

Yet such speculation doesn't hold true in Lagrange Polynomial Method. At lower n, the polynomial value closer to the edge differs much less than the one closes to maxium of minimum value of $f(x)$. And at higher n, the exact opposite occurs. The former situation is dued to the lack of data points, and the latter one is severe Runge's phenomenon. the magnitude of the (n+1)-th derivative of Runge's function increases when n increases. The consequence is that the resulting upper bound tends to infinity when n tends to infinity.

Way to prevent this occasion varies. Including:

- Change of interpolation points
- S-Runge algorithm without resampling
- Use of piecewise polynomials
- Constrained minimization
- Least squares fitting
- Bernstein polynomial
- External fake constraints interpolation

As a plain computer assignment, we won't be discussing this matter.

4 Codes

//fuction

```
[lagrange_interp.m]
function lagrange_interp(x, y, x_val)
    n = length(x);
    L = ones(n,length(x_val));

    for i = 1:n
        for j = 1:n
            if i ~= j
                L(i,:) = L(i,:) .* (x_val - x(j)) / (x(i) - x(j));
            end
        end
    end

    interpolated_y = zeros(1,length(x_val));
    for i = 1:n
        interpolated_y = interpolated_y + y(i) * L(i,:);
    end

    disp('Interpolated values:');
    disp(interpolated_y);

    % Plot the interpolated polynomial
    plot(x, y, 'o', x_val, interpolated_y);
    xlabel('x');
    ylabel('y');
end


[piecewiselinearinterp.m]
function y_output = piecewiselinearinterp(x,y,x_input)
n = length(x);
nn = length(x_input);

for j=1:nn
for i=1:n-1
if (x_input(j)>x(i) && x_input(j)<=x(i+1))
y_output(j) = ((x_input(j)-x(i+1))/(x(i)-x(i+1)))*y(i)+(((x_input(j)-x(i))/(x(i+1)-x(i)))*y(i+1));
end
end
end

plot(x,y);

end
```

//scripts

```
[DrawInterpolar.m]
clear;
clc;

f = @(x) x ./ (1 + x .^ 4);

%init

n = 10;

for i = 1:(n+1)
x(i) = -5 + (10 * (i-1) / n);
end

y = f(x);

x_val = -5:0.1:5;

hold on
lagrange_interp(x, y, x_val);
fplot(f);
piecewiselinearinterp(x,y,x_val);
title('Lagrange Interpolating Polynomial(n=10)');
legend('Data Points', 'Interpolated Polynomial', 'x/(1+x^4)','piecewise');
```