Math 315 Lab 4

Table of Contents

The following lab examines the Runge phenomenon for 5 different ways of polynomial interpolation: Vandermonde / Power Series, Newton, Newton with Divided Differences, Lagrange, and Chebyshev interpolation. I will use these different methods of interpolation to approximate the exponential function and calculate the maximum absolute error of these approximations on the interval [-1, 1].

Vandermonde / Power Series Interpolation

The following code takes in nodes (x, f) and approximation points xx to outputs approximated values y by creating an interpolating polynomial through Vandermonde / Power Series interpolation.

```
disp(fileread('vandermonde.m'));

function [y] = vandermonde(x, f, xx)
    v = vander(x);
    p = v\f;
    y = polyval(p,xx);
end
```

Newton Textbook Interpolation

The following code takes in nodes (x, f) and approximation points xx to outputs approximated values y by creating an interpolating polynomial through the textbook version of Newton interpolation.

```
disp(fileread('vandermonde.m'));

function [y] = vandermonde(x, f, xx)
    v = vander(x);
    p = v\f;
    y = polyval(p,xx);
end
```

Newton Divided Differences Interpolation

The following code takes in nodes (x, f) and approximation points xx to outputs approximated values y by creating an interpolating polynomial through the divided differences version of Newton interpolation.

```
disp(fileread('vandermonde.m'));
```

```
function [y] = vandermonde(x, f, xx)
    v = vander(x);
    p = v\f;
    y = polyval(p,xx);
end
```

Lagrange Interpolation

The following code takes in nodes (x, f) and approximation points xx to outputs approximated values y by creating an interpolating polynomial through Lagrange interpolation.

```
disp(fileread('vandermonde.m'));
```

```
function [y] = vandermonde(x, f, xx)
    v = vander(x);
    p = v\f;
    y = polyval(p,xx);
end
```

Chebyshev Interpolation

The following code takes in nodes (x, f) and approximation points xx to outputs approximated values y by creating an interpolating polynomial through Chebyshev interpolation.

```
disp(fileread('vandermonde.m'));
```

```
function [y] = vandermonde(x, f, xx)
    v = vander(x);
    p = v\f;
    y = polyval(p,xx);
end
```

Equally Spaced Data Points on the Interval [-1, 1]

```
close all;
warning('off', 'MATLAB:nearlySingularMatrix');
vandermonde_err = zeros(10, 1);
newton_err = zeros(10, 1);
newton_divided_diff_err = zeros(10, 1);
lagrange_err = zeros(10, 1);
chebyshev_err = zeros(10, 1);

for n = 10:10:100
    x = linspace(-1, 1, n)';
    f = exp(x);
    xx = linspace(-1, 1, 1025.*n);
    t = linspace(-1, 1, 1000);

    % Vandermonde / Power Series Interpolation
    err = max(abs(vandermonde(x, f, xx) - exp(xx)));
```

```
    vandermonde_err(n ./ 10) = err;
    % disp(err);
    % f1 = figure(1);
    % plot(t, vandermonde(x, f, t));

    % Newton Textbook Interpolation
    err = max(abs(newton(x, f, xx) - exp(xx)));
    newton_err(n ./ 10) = err;
    % disp(err);
    % f2 = figure(2);
    % plot(t, newton(x, f, t));

    % Netwon with Divided Differences Interpolation
    err = max(abs(newton_divided_diff(x, f, xx) - exp(xx)));
    newton_divided_diff_err(n ./ 10) = err;
    % disp(err);
    % f3 = figure(3);
    % plot(t, newton_divided_diff(x, f, t));

    % Lagrange Interpolation
    err = max(abs(lagrange(x, f, xx) - exp(xx)));
    lagrange_err(n ./ 10) = err;
    % disp(err);
    % f4 = figure(4);
    % plot(t, lagrange(x, f, t));

    % Chebyshev Interpolation
    err = max(abs(chebfit(x, f, xx) - exp(xx)));
    chebyshev_err(n ./ 10) = err;
    % disp(err);
    % f5 = figure(5);
    % plot(t, chebfit(x, f, t));
end

vandermonde_err = categorical(compose('%.7e', round(vandermonde_err, 7,
'significant')));
newton_err = categorical(compose('%.7e', round(newton_err, 7,
'significant')));
newton_divided_diff_err = categorical(compose('%.7e',
round(newton_divided_diff_err, 7, 'significant')));
lagrange_err = categorical(compose('%.7e', round(lagrange_err, 7,
'significant')));
chebyshev_err = categorical(compose('%.7e', round(chebyshev_err, 7,
'significant')));

T = table(linspace(10,100,10)', vandermonde_err, newton_err,
newton_divided_diff_err, lagrange_err, ...
    chebyshev_err, 'VariableNames', {'n', 'Vandermonde', 'Newton', 'Newton
Divided Difference', 'Lagrange', 'Chebyshev'});
disp(T);
```

```
     n       Vandermonde        Newton        Newton Divided Difference
Lagrange         Chebyshev

    ___     _____________     _____________     _________________________
```

10	*3.8500810e-09*	*3.8500840e-09*	*3.8500830e-09*	*3.8500810e-09*	*3.8500820e-09*
20	*2.4069640e-13*	*3.5260680e-13*	*7.6383340e-14*	*1.6906480e-12*	*5.9990900e-13*
30	*1.8991340e-10*	*7.5025600e-11*	*3.6803670e-11*	*8.2540950e-10*	*5.0064400e-11*
40	*1.2017770e-07*	*1.8356500e-08*	*3.0007160e-08*	*4.6020290e-07*	*5.9304010e-08*
50	*7.1502350e-06*	*1.8936330e-05*	*1.2119690e-05*	*4.8220850e-04*	*2.7951690e-04*
60	*1.8725160e-02*	*6.1458640e-03*	*2.3528970e-02*	*4.3500440e-01*	*2.7934060e-02*
70	*1.0174630e+02*	*2.1747270e+04*	*1.4942710e+01*	*3.7496930e+02*	*2.9987960e+01*
80	*2.1759130e+05*	*7.6939990e+09*	*1.4798190e+04*	*2.3503340e+05*	*1.3600450e+01*
90	*1.9083950e+09*	*7.2711500e+15*	*9.5206610e+08*	*2.7500180e+08*	*8.8280160e+00*
100	*9.1045340e+13*	*6.0356540e+21*	*1.9562280e+13*	*2.4248560e+11*	*3.8543940e+01*

Chebyshev Points

```
close all;
warning('off', 'MATLAB:nearlySingularMatrix');
vandermonde_err = zeros(10, 1);
newton_err = zeros(10, 1);
newton_divided_diff_err = zeros(10, 1);
lagrange_err = zeros(10, 1);
chebyshev_err = zeros(10, 1);

for n = 10:10:100
    i = linspace(1, n, n);
    a = -1;
    b = 1;
    x = (((b + a) ./ 2) - ((b - a) ./ 2) .* cos((2 .* i + 1) .* pi ./ (2 .* n
+ 2)))';

    f = exp(x);
    xx = linspace(-1, 1, 1025.*n);
    t = linspace(-1, 1, 1000);

    % Vandermonde / Power Series Interpolation
    err = max(abs(vandermonde(x, f, xx) - exp(xx)));
    vandermonde_err(n ./ 10) = err;
    % disp(err);
    % f1 = figure(1);
    % plot(t, vandermonde(x, f, t));

    % Newton Textbook Interpolation
    err = max(abs(newton(x, f, xx) - exp(xx)));
    newton_err(n ./ 10) = err;
```

```
    % disp(err);
    % f2 = figure(2);
    % plot(t, newton(x, f, t));

    % Netwon with Divided Differences Interpolation
    err = max(abs(newton_divided_diff(x, f, xx) - exp(xx)));
    newton_divided_diff_err(n ./ 10) = err;
    % disp(err);
    % f3 = figure(3);
    % plot(t, newton_divided_diff(x, f, t));

    % Lagrange Interpolation
    err = max(abs(lagrange(x, f, xx) - exp(xx)));
    lagrange_err(n ./ 10) = err;
    % disp(err);
    % f4 = figure(4);
    % plot(t, lagrange(x, f, t));

    % Chebyshev Interpolation
    err = max(abs(chebfit(x, f, xx) - exp(xx)));
    chebyshev_err(n ./ 10) = err;
    % disp(err);
    % f5 = figure(5);
    % plot(t, chebfit(x, f, t));
end

vandermonde_err = categorical(compose('%.7e', round(vandermonde_err, 7,
'significant')));
newton_err = categorical(compose('%.7e', round(newton_err, 7,
'significant')));
newton_divided_diff_err = categorical(compose('%.7e',
round(newton_divided_diff_err, 7, 'significant')));
lagrange_err = categorical(compose('%.7e', round(lagrange_err, 7,
'significant')));
chebyshev_err = categorical(compose('%.7e', round(chebyshev_err, 7,
'significant')));

T = table(linspace(10,100,10)', vandermonde_err, newton_err,
newton_divided_diff_err, lagrange_err, ...
    chebyshev_err, 'VariableNames', {'n', 'Vandermonde', 'Newton', 'Newton
Divided Difference', 'Lagrange', 'Chebyshev'});
disp(T);
```

n	Vandermonde	Newton	Newton Divided Difference	Lagrange	Chebyshev
10	2.6972520e-08	2.6972520e-08	2.6972520e-08	2.6972510e-08	2.6972520e-08
20	2.2759570e-15	8.8817840e-16	8.8817840e-16	4.8294700e-14	2.2204460e-15
30	1.0658140e-14	7.2719610e-15	1.4988010e-15	5.4067860e-14	3.1086240e-15
40	1.9095840e-14	7.6605390e-15	9.9364960e-15		

```
5.5178080e-14    4.5963230e-14
     50    2.9865000e-14    1.6234970e-10          5.8359540e-11
2.0317080e-13    8.8928860e-14
     60    5.4160010e-12    1.3197300e-05          2.8741780e-06
3.7414520e-13    1.1102230e-14
     70    8.5729110e-10    6.5922410e+00          4.1672570e-01
2.2604140e-13    1.3233860e-13
     80    1.9435480e-06    2.8392880e+04          2.9230400e+04
2.3658850e-13    2.6173510e-13
     90    5.7297200e-03    1.7028360e+09          4.3492790e+08
1.0169640e-12    1.7746920e-13
    100    8.3667620e+01    7.1864990e+15          4.0966120e+13
7.6183500e-13    2.0278220e-13
```

Published with MATLAB® R2024b