Lab4Simons.m

Table of Contents

Lab by Madilyn Simons

Problem 1

Approximate f(0.8), f(1.2), and f(1.7) using Neville's method

```
% Given x and f(x) values
x = [0.5 0.7 1.0 1.3 1.5 1.6 2.0];
y = [1.772454 1.298055 1 0.897471 0.886227 0.893515 1];

% Values of x to approximate f(x)
val = [0.8 1.2 1.7];

% Approximate f(x) for each val and print the results
for i = 1:3
    ret = neville(val(i), x, y, 7);
    fprintf('f(%.1f) = %.6f\n', val(i), ret);
end
```

```
f(0.8) = 1.163081
f(1.2) = 0.918424
f(1.7) = 0.908099
```

Problem 2

Use Neville's method to approximate f(4.9) using N nodes of data generated from f(x) = 1 / (1+x^2);

```
% number of nodes for each approximation
N = [11 21 41 81 121 161];

% function used to generate data
f = @(x) 1 / (1 + x*x);

% Generate n number of nodes for each value of N and approximate
% f(4.9) using the nodes and Neville's method.  Then print the
 results.
for i = 1:6
    ret = neville_nodes(4.9, f, N(i));
    fprintf('N: %d\tApproximation: %.6f\n', N(i), ret);
end
```

```
N: 11 Approximation: 1.230317
```

```
N: 21 Approximation: -58.238141
N: 41 Approximation: -78688.997501
N: 81 Approximation: -40443044569.410362
N: 121 Approximation: 35235652991531112.000000
N: 161 Approximation: 44255285491083409063149568.000000
```

Problem 3

Use Neville's method to approximate f(4.9) using N Chebyshev nodes

```
% Number of nodes for each approximation
N = [11 21 41 81 121 161];

% function used to generate data
f = @(x) 1 / (1 + x*x);

% Generate n Chebyshev nodes for each value of N and approximate
 f(4.9)
% using the nodes and Neville's method.  Then print the results.
for i = 1:6
    ret = chebyshev(4.9, f, N(i));
    fprintf('N: %d\tApproximation: %.6f\n', N(i), ret);
end
```

```
N: 11 Approximation: 0.066370
N: 21 Approximation: 0.037059
N: 41 Approximation: 0.039944
N: 81 Approximation: 0.039984
N: 121 Approximation: 0.039984
N: 161 Approximation: 0.039984
```

Problem 4

Use Inverse Interpolation to approximate x - e^-x = 0.

```
% x value to approximate x - e^-x = 0 given p = f^-1(0)
val = 0;

% Given x and e^-x values
x = [0.3 0.4 0.5 0.6 0.7];
y = [0.740818 0.670320 0.606531 0.548812 0.496585];

% Use Neville's method to construct and interpolating polynomial and
% approximate p = f^-1(0)
p = neville(val, x, y, 5);

fprintf('p = %.6f\n', p);
```

```
p = 0.999871
```

Published with MATLAB® R2018b