
Table of Contents

.....	1
PROBLEM 1: VALUE SORTING	1
PROBLEM 2: MANUAL KAPREKAR PROCESS FOR THREE ITES	2
PROBLEM 3: MATRIX PLOT TEST	2
PROBLEM 4: NEWTON QUEST	3

```
% Pierce Zhang, CMOR220, FALL 2023, Competency string manipulation and
% matrix plots
% string_manip_matrix_plots.m
% Answers to competency on string manipulation and matrix plots
% Last modified: 30 September 2023
```

```
% Driver
function string_manip_matrix_plots
```

PROBLEM 1: VALUE SORTING

```
[vallasc, valldesc] = sort_digits(64782) %#ok<*NOPRT,*ASGLU>
[val2asc, val2desc] = sort_digits(16794)
[val3asc, val3desc] = sort_digits(25148)
```

```
vallasc =
```

```
24678
```

```
valldesc =
```

```
87642
```

```
val2asc =
```

```
14679
```

```
val2desc =
```

```
97641
```

```
val3asc =
```

```
12458
```

`val3desc =`

`85421`

PROBLEM 2: MANUAL KAPREKAR PROCESS FOR THREE ITES

```
kapr_396 = manual_kaprekar(396)
kapr_734 = manual_kaprekar(734)
kapr_947 = manual_kaprekar(947)
```

`kapr_396 =`

`495`

`kapr_734 =`

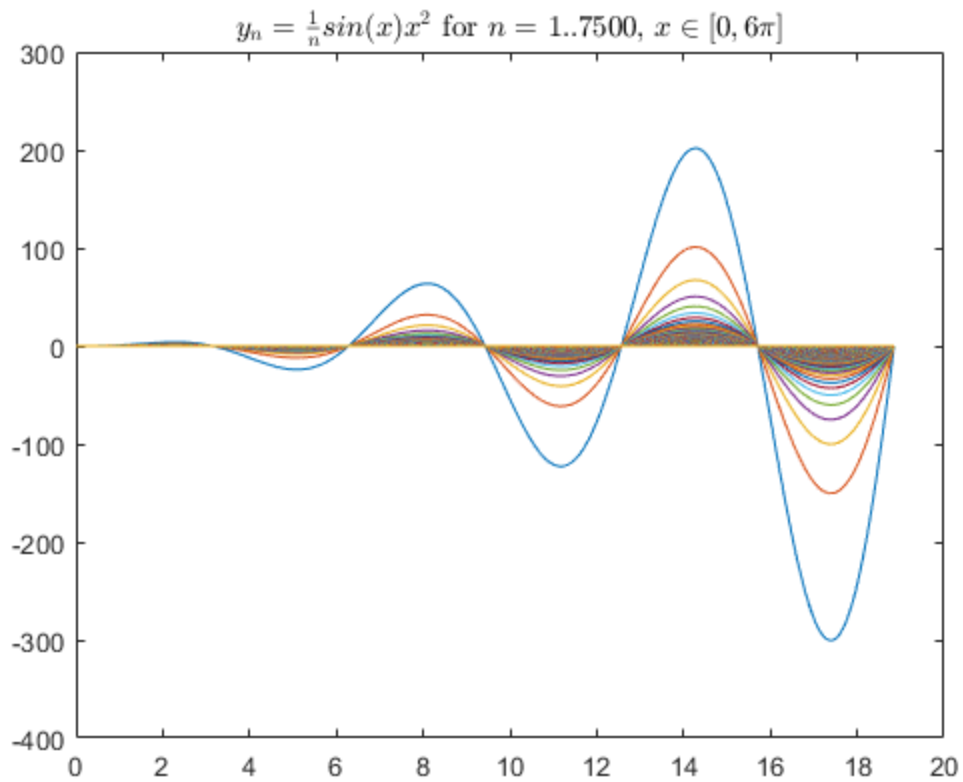
`495`

`kapr_947 =`

`495`

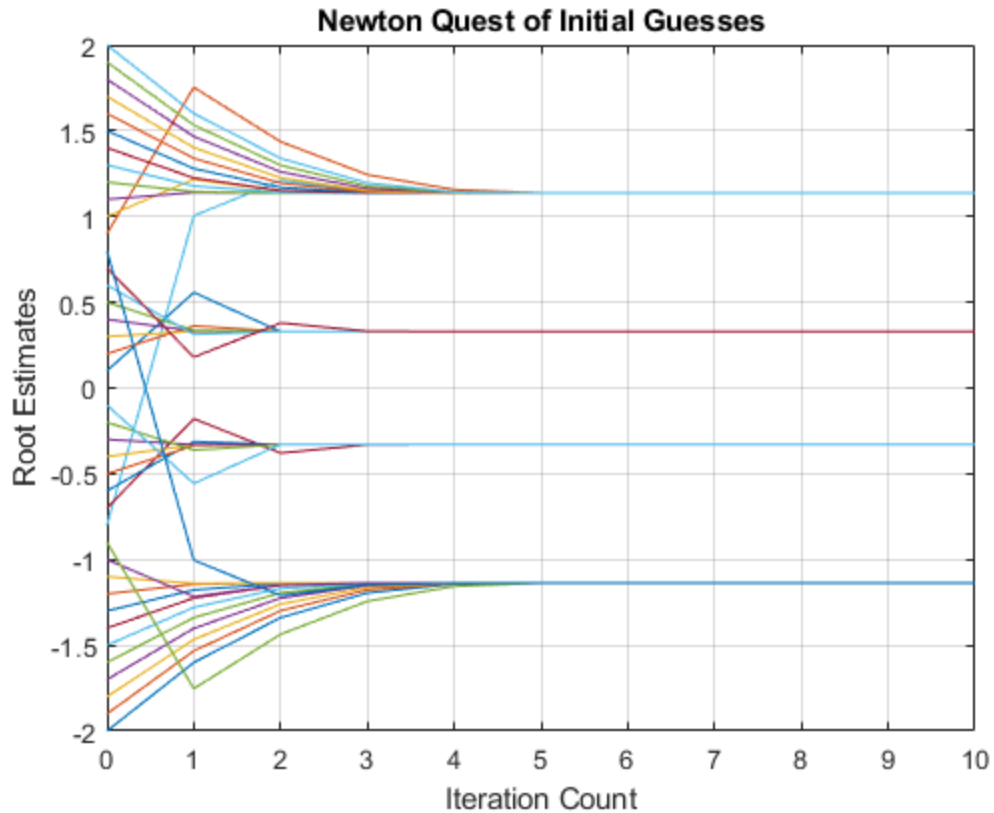
PROBLEM 3: MATRIX PLOT TEST

```
x_mt = linspace(0,6*pi,500);
A = matrix_test(x_mt);
figure();
plot(x_mt,A);
title("$y_n=\frac{1}{n}\sin(x)x^2$ for $n$ = 1..7500, $x$
\in[0,6\pi]$", 'Interpreter', 'latex');
```



PROBLEM 4: NEWTON QUEST

```
A_quest = newton_quest_test();  
figure();  
plot(0:10,A_quest);  
title("Newton Quest of Initial Guesses");  
xlabel("Iteration Count"); ylabel("Root Estimates");  
grid on
```



end

```
% Inputs: n, number to be sorted
% Outputs:
% - asc, number with digits of n in ascending order
% - desc, number with digits of n in descending order
function [asc, desc] = sort_digits(n)
    % Converts number to string, sorts string, converts string back to
    % double
    string_num = num2str(n);
    asc = str2double(sort(string_num, 'ascend'));
    desc = str2double(sort(string_num, 'descend'));
end
```

```
% Inputs: n, number on which to execute Kaprekar's quest three times
% Outputs: val, output of Kaprekar's quest after three iters
```

```
function [val] = manual_kaprekar(n)
    % Repeat Kaprekar process three times
    for i=1:3
        [asc, desc] = sort_digits(n);
        n = desc - asc;
    end
    val = n;
end
```

```
% Inputs: n, x as specified in function
```

```

% Outputs: yn, value of the given composition at n and x
function [yn] = matrix_test_func(n, x)
    yn = sin(x) .* x.^2 ./ n;
end

% Inputs: x, as specified in function
% Outputs: y, value of the given polynomial at x
function [y] = nq_func(x)
    y = -5*x.^4 + 7*x.^2 - 0.7;
end

% Inputs: x, as specified in the function
% Outputs: ydx, value of the derivative of the given polynomial OR this
% function at x
function [ydx] = nq_funcdx(x)
    ydx = -20*x.^3 + 14*x;
end

% Inputs: x, vector consisting of x-values on which to generate the table
% of values for the plot test
% Outputs: A, matrix consisting of 7500 rows of function outputs where each
% row corresponds to a different n in 1..7500 and 500 columns of function
% outputs where each column corresponds to a different x in [0,6pi]
function [A] = matrix_test(x)
    A = zeros(7500,500);
    for n=1:7500
        A(n,:) = matrix_test_func(n, x);
    end
end

% Inputs: none
% Outputs: A, matrix consisting of rows where each row represents a
% different starting value of Newton's method and columns where each column
% represents a different iteration for the test function and bounds
function [A] = newton_quest_test()
    maxiter = 10;
    x0 = -2:0.1:2;

    A = zeros(length(x0),11);
    A(:,1) = transpose(x0);
    for n=1:maxiter
        A(:,n+1) = A(:,n) - nq_func(A(:,n)) ./ nq_funcdx(A(:,n));
    end
end

```

Published with MATLAB® R2023a