
ENGR 451 - Chapter 2 Laboratory

Matlab tutorial

```
clear
x = sequence([1 2 3 4 5], 1);
y = sequence([5 3 1 -1 3 -2 2 3], -1);

% test plus
test_lab1('plus(x, y)')
test_lab1('plus(y, x)')
test_lab1('plus(1, x)')
test_lab1('plus(x, 1)')

y = sequence([5 3 1 0 3 -2 2 3], -4);
test_lab1('plus(x, y)')
test_lab1('plus(y, x)')

% test minustract
test_lab1('minus(x, y)')
test_lab1('minus(y, x)')
test_lab1('minus(1, x)')
test_lab1('minus(x, 1)')

% test timesiplication
test_lab1('times(x, y)')
test_lab1('times(3, x)')
test_lab1('times(x, 3)')

% test flip
test_lab1('flip(x)')

% test shift
test_lab1('shift(y, 2)')

%combinations
test_lab1('flip(minus(shift(plus(x, 2), 4), y))')
test_lab1('plus(flip(plus(x, y)), shift(y, -5))')
test_lab1('minus(plus(times(shift(flip(x), 4), shift(y, 3)), flip(y)),
    x)')

% test stem
set(clf, 'Position', [200 200 400 200])
stem(flip(2+(x-shift(y, -4).*y-3)))
title('y[n]');

% Program Listings
fprintf('\n\n')
disp('--- sequence.m -----')
type sequence

plus(x, y): sequence O.K.
plus(y, x): sequence O.K.
```

```

plus(1, x): sequence O.K.
plus(x, 1): sequence O.K.
plus(x, y): sequence O.K.
plus(y, x): sequence O.K.
minus(x, y): sequence O.K.
minus(y, x): sequence O.K.
minus(1, x): sequence O.K.
minus(x, 1): sequence O.K.
times(x, y): sequence O.K.
times(3, x): sequence O.K.
times(x, 3): sequence O.K.
flip(x): sequence O.K.
shift(y, 2): sequence O.K.
flip(minus(shift(plus(x, 2), 4), y)): sequence O.K.
plus(flip(plus(x, y)), shift(y, -5)): sequence O.K.
minus(plus(times(shift(flip(x), 4), shift(y, 3)), flip(y)), x):
sequence O.K.

```

```

--- sequence.m -----

```

```

classdef sequence
    properties
        data
        offset
    end

    methods(Static)
        % Pads the input sequences so that they are of the same
length.
        % Sequence with the lower offset will not have front padding.
This
        % returns the data portion of the sequences only.
function [a,b] = padData(x,y)
        % Find which sequence has the lower offset (furthest to
the
        % left).
        lo = sequence([],0);
        hi = sequence([],0);
        if(x.offset<y.offset)
            lo = x;
            hi = y;
        else
            lo = y;
            hi = x;
        end
        % Define ints for left and right padding of zeros.
        leftPad = hi.offset-lo.offset;
        rightPad = (length(lo.data)+lo.offset)-
(length(hi.data)+hi.offset);
        % Padding the left side of the sequence with the higher
offset
        % is easiest.
        hi.data = [zeros(1,leftPad),hi.data];
    end
end

```

```

        % Padd the right side of either the lower or higher offset
        % sequence depending on whether the difference in length
is
        % positive or negative.
        if(rightPad>0)
            hi.data = [hi.data, zeros(1,rightPad)];
        elseif(rightPad<0)
            lo.data = [lo.data, zeros(1,abs(rightPad))];
        end
        % Map lo and hi back to the order in which they came i.e.
a = x
        % and b = y.
        if(x.offset<y.offset)
            a=lo.data;
            b=hi.data;
        else
            a=hi.data;
            b=lo.data;
        end
    end
end

methods
function s = sequence(data, offset)
    % SEQUENCE    Sequence object
    %             S = SEQUENCE(DATA, OFFSET) creates sequence S
    %             using DATA and OFFSET
    %
    %             Your Name    1 Jan 2014
    s.data = data;
    s.offset = offset;
end

function display(s)
    var = inputname(1);
    if (isempty(var))
        disp('ans =');
    else
        disp([var '=']);
    end
    switch length(s.data)
        case 0
            disp('    data: []')
        case 1
            disp(['    data: ', num2str(s.data)])
        otherwise
            disp(['    data: [' num2str(s.data) ']]')
        end
    disp([' offset: ' num2str(s.offset)])
end

function y = flip(x)
    ofs = -(x.offset+length(x.data)-1);
    y = sequence(x.data(end:-1:1),ofs);

```

```

end

function y = shift(x, n0)
    y = sequence(x.data, x.offset+n0);
end

function z = plus(x, y)
    if(isa(x,'double'))
        z = sequence(x+y.data,y.offset);
    elseif(isa(y,'double'))
        z = sequence(x.data+y,x.offset);
    else
        [a, b] = sequence.padData(x,y);
        z = sequence(a+b,min(x.offset,y.offset));
    end
end

function z = minus(x, y)
    if(isa(x,'double'))
        z = sequence(x-y.data,y.offset);
    elseif(isa(y,'double'))
        z = sequence(x.data-y,x.offset);
    else
        [a, b] = sequence.padData(x,y);
        z = sequence(a-b,min(x.offset,y.offset));
    end
end

function z = times(x, y)
    if(isa(x,'double'))
        z = sequence(x.*y.data,y.offset);
    elseif(isa(y,'double'))
        z = sequence(x.data.*y,x.offset);
    else
        [a, b] = sequence.padData(x,y);
        z = sequence(a.*b,min(x.offset,y.offset));
    end
end

function stem(x)
    % STEM Display a Matlab sequence, x, using a stem plot.
    data_length = length(x.data);
    n_axis_indeces = linspace(1,data_length,data_length);
    n_axis_vals = n_axis_indeces
    +linspace(x.offset,x.offset,data_length)-1;

    figure
    stem(x.data)

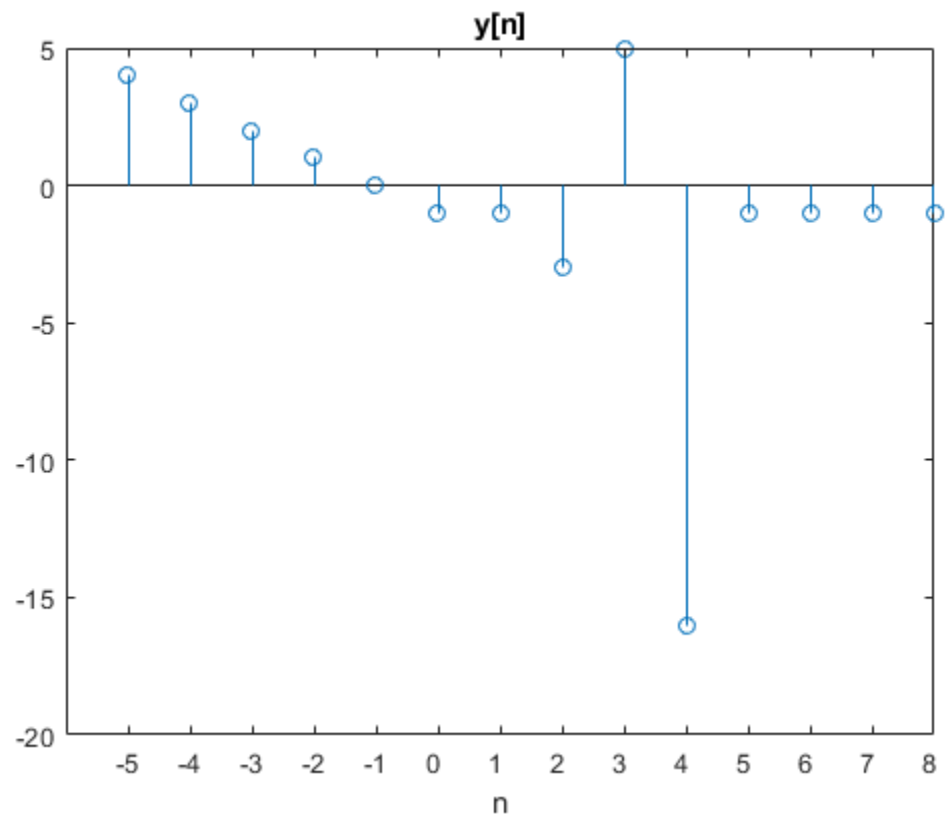
    xlabel('n'); title('x');

    set(gca,'XTick', n_axis_indeces );
    set(gca,'XTickLabel', n_axis_vals );

```

```
end
end
end

% When finished: publish Lab1 'pdf' or 'doc'
```



Published with MATLAB® R2019b