



การสร้างฟังก์ชัน ทดสอบการใช้งานฟังก์ชัน และวิเคราะห์ผลการใช้งานฟังก์ชัน

จัดทำโดย

นาย ปณัฑิต สีดาว 59010759 sec.1

เสนอ

อาจารย์จรัสศักดิ์ สิทธิกร

รายงานนี้เป็นส่วนหนึ่งของวิชา

วิชา Engineering Mathematics 3

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 1 ปีการศึกษา 2560

สารบัญ

เรื่อง	หน้า
คำนำ	3
ที่มาและความสำคัญ	4
ประโยชน์ที่ได้รับ	4
การสร้าง ฟังก์ชัน และ ทดสอบการใช้งานฟังก์ชัน และวิเคราะห์ผลการใช้งานฟังก์ชัน	5
1. ระหว่างฟังก์ชันใน Homework 1.5.4.1 และ Homework 1.6.6.1	
1. Source Code ของ Homework 1.5.4.1	5
2. Result ของ Homework 1.5.4.1	8
3. Source Code ของ Homework 16.6.1	9
4. Result ของ Homework 16.6.1	11
5. วิเคราะห์ผลการใช้งานระหว่าง Homework 1.5.4.1 กับ Homework 16.6.1	11
2. ระหว่างฟังก์ชันใน Homework 1.5.5.1 และ Homework 1.6.3.1	
1. Source Code ของ Homework 1.5.5.1	12
2. Result ของ Homework 1.5.5.1	16
3. Source Code ของ Homework 1.6.3.1	18
4. Result ของ Homework 1.6.3.1	20
5. วิเคราะห์ผลการใช้งานระหว่าง Homework 1.5.4.1 กับ Homework 16.6.1	20
3. ระหว่างฟังก์ชันใน Homework 4.3.2.5 และ ฟังก์ชันที่สร้างขึ้นเอง	
1. Source Code ของ Homework 4.3.2.5	21
2. Result ของ Homework 4.3.2.5	25
3. Source Code ของฟังก์ชันที่สร้างขึ้นเอง	27
4. Result ของฟังก์ชันที่สร้างขึ้นเอง	28
5. วิเคราะห์ผลการใช้งานระหว่าง Homework 1.5.4.1 กับฟังก์ชันที่สร้างขึ้นเอง	29

คำนำ

รายงานเล่มนี้ จัดทำขึ้นเพื่อเป็นส่วนหนึ่งของวิชา **01006003 Engineering Mathematics 3** เพื่อให้ได้ศึกษาหาความรู้เกี่ยวกับการการสร้างฟังก์ชัน ทดสอบการใช้งานฟังก์ชัน และวิเคราะห์ผลการใช้งานฟังก์ชันต่างๆ โดยใช้โปรแกรมในการคำนวณพีชคณิตเชิงเส้น (**Linear Algebra**) ได้แก่ **MATLAB** และโปรแกรม **Spark** เพื่อเป็นประโยชน์กับการเรียน

ผู้จัดทำหวังว่า รายงานเล่มนี้จะเป็นประโยชน์กับผู้อ่าน หรือนักเรียน นักศึกษา ที่กำลังหาข้อมูลเรื่องนี้ หากมีข้อเสนอและข้อผิดพลาดประการใด ผู้จัดทำขอน้อมรับไว้ด้วยประการทั้งปวง

บัณฑิต สีดาว

วันที่ 9 ธันวาคม 2560

ที่มาและความสำคัญ

รายงานนี้เกิดจากทางผู้สอนอยากให้นักศึกษาได้นำความรู้ด้านการเขียนโปรแกรมมาผสานกับความรู้ทางคณิตศาสตร์จึงเกิดรายงานเล่มนี้ขึ้นมา

ประโยชน์ที่ได้รับ

1. ได้ฝึกทักษะการเขียนโปรแกรมที่เกี่ยวข้องกับความรู้ทางคณิตศาสตร์
2. ใช้เวลาว่างให้เกิดประโยชน์

สร้าง ฟังก์ชัน และ ทดสอบการใช้งานฟังก์ชัน และวิเคราะห์ผลการใช้งานฟังก์ชัน

1. ระหว่างฟังก์ชันใน Homework 1.5.4.1 และ Homework 1.6.6.1

1. Source Code ของ Homework 1.5.4.1

laff_axpy.m

```
function [y_out] = laff_axpy(a, x, y)

[m_a, n_a] = size(a);
[m_x, n_x] = size(x);
[m_y, n_y] = size(y);

if (m_a ~= 1 & n_a ~= 1)
    y_out = 'FAILED';
    return
end

if ( m_x * n_x ~= m_y * n_y )
    y_out = 'FAILED';
    return
end

if (n_x == 1)

    if (n_y == 1)
        for i=1:m_x
            y(i, 1) = a*x(i, 1) + y(i, 1);
        end

    elseif (m_y == 1)
        for i=1:m_x
            y(1, i) = a*x(i, 1) + y(1, i);
        end
    end

elseif (m_x == 1)

    if (m_y == 1)
        for i=1:n_x
            y(1, i) = a*x(1, i) + y(1, i);
        end

    elseif (n_y == 1)
        for i=1:n_x
            y(i, 1) = a*x(1, i) + y(i, 1);
        end
    end

end

y_out = y;

end
```

test_laff_axpy.m

```
a = randi( [ -2, 2 ], 1 );
x = randi( [ -2, 2 ], 4, 1 );
y = randi( [ -2, 2 ], 4, 1 );

z = randi( [ -2, 2 ], 3, 1 );

disp( 'column - column axpy' )
if ( isequal( laff_axpy_59010759( a, x, y ), a * x + y ) )
    disp( 'TEST PASSED' )
else
    disp( 'TEST FAILED' )
end

disp( ' ' )

disp( 'column - row axpy' )
if ( isequal( laff_axpy_59010759( a, x, y' ), a * x' + y' ) )
    disp( 'TEST PASSED' )
else
    disp( 'TEST FAILED' )
end

disp( ' ' )

disp( 'row - column axpy' )
if ( isequal( laff_axpy_59010759( a, x', y ), a * x + y ) )
    disp( 'TEST PASSED' )
else
    disp( 'TEST FAILED' )
end

disp( ' ' )

disp( 'row -> row axpy' )
if ( isequal( laff_axpy_59010759( a, x', y' ), a * x' + y' ) )
    disp( 'TEST PASSED' )
else
    disp( 'TEST FAILED' )
end

disp( ' ' )

disp( 'column - column axpy (wrong size)' )
if ( isequal( laff_axpy_59010759( a, x, z ), 'FAILED' ) )
    disp( 'TEST PASSED' )
else
    disp( 'TEST FAILED' )
end

disp( ' ' )

disp( 'column - row axpy (wrong size)' )
if ( isequal( laff_axpy_59010759( a, x, z' ), 'FAILED' ) )
    disp( 'TEST PASSED' )
else
    disp( 'TEST FAILED' )
end
```

```

end

disp( ' ' )

disp( 'row -> column (wrong size)' )
if ( isequal( laff_axpy_59010759( a, x', z ), 'FAILED' ) )
    disp( 'TEST PASSED' )
else
    disp( 'TEST FAILED' )
end

disp( ' ' )

disp( 'row -> row axpy (wrong size)' )
if ( isequal( laff_axpy_59010759( a, x', z' ), 'FAILED' ) )
    disp( 'TEST PASSED' )
else
    disp( 'TEST FAILED' )
end

```

test_time_laff_axpy.m

```

limit = 1000;
size = 10;
while size <= limit
tic
    for time=1:limit

        a = randi( [ -2, 2 ], 1 );
        x = randi( [ -2, 2 ], size, 1 );
        y = randi( [ -2, 2 ], size, 1 );

        laff_axpy_59010759( a, x, y );

    end
    fprintf('Size : %5d, 1,000 times : ', size);
    toc
    size = 10*size;
end

```

2. Result ของ Homework 1.5.4.1

ผลการทดสอบของความถูกต้องของฟังก์ชัน

```
>> test_axpy_59010759
column - column axpy
TEST PASSED
```

```
column - row axpy
TEST PASSED
```

```
row - column axpy
TEST PASSED
```

```
row -> row axpy
TEST PASSED
```

```
column - column axpy (wrong size)
TEST PASSED
```

```
column - row axpy (wrong size)
TEST PASSED
```

```
row -> column (wrong size)
TEST PASSED
```

```
row -> row axpy (wrong size)
TEST PASSED
```

ผลการทดสอบประสิทธิภาพของฟังก์ชัน

```
>> test_time_laff_axpy_59010759
```

```
Size :    10, 1,000 times : Elapsed time is 0.199105
seconds.
```

```
Size :   100, 1,000 times : Elapsed time is 0.137768
seconds.
```

```
Size :  1000, 1,000 times : Elapsed time is 0.234531
seconds.
```


3. Source Code 16.6.1

Axpy_unb.m

```
% Copyright 2017 The University of Texas at Austin
%
% For licensing information see
%     http://www.cs.utexas.edu/users/flame/license.html
%
% Programmed by: Name of author
%               Email of author

function [ y_out ] = Axpy_unb( alpha, x, y )

    [ xT, ...
      xB ] = FLA_Part_2x1( x, ...
                          0, 'FLA_TOP' );

    [ yT, ...
      yB ] = FLA_Part_2x1( y, ...
                          0, 'FLA_TOP' );

    while ( size( xT, 1 ) < size( x, 1 ) )

        [ x0, ...
          chi1, ...
          x2 ] = FLA_Repart_2x1_to_3x1( xT, ...
                                       xB, ...
                                       1, 'FLA_BOTTOM' );

        [ y0, ...
          psi1, ...
          y2 ] = FLA_Repart_2x1_to_3x1( yT, ...
                                       yB, ...
                                       1, 'FLA_BOTTOM' );

        %-----%

        psi1 = alpha * chi1 + psi1;

        %-----%

        [ xT, ...
          xB ] = FLA_Cont_with_3x1_to_2x1( x0, ...
                                       chi1, ...
                                       x2, ...
                                       'FLA_TOP' );

        [ yT, ...
          yB ] = FLA_Cont_with_3x1_to_2x1( y0, ...
                                       psi1, ...
                                       y2, ...
                                       'FLA_TOP' );

    end

    y_out = [ yT
```

```
yB ];
```

```
return
```

test_Axpy_unb.m

```
a = randi( [ -2, 2 ], 1 );  
x = randi( [ -2, 2 ], 4, 1 );  
y = randi( [ -2, 2 ], 4, 1 );  
  
disp( 'axpy' )  
if ( isequal( Axpy_unb_59010759( a, x, y ), a * x + y ) )  
    disp( 'TEST PASSED' )  
else  
    disp( 'TEST FAILED' )  
end
```

test_time__Axpy_unb.m

```
limit = 1000;  
size = 10;  
while size <= limit  
tic  
    for time=1:limit  
  
        a = randi( [ -2, 2 ], 1 );  
        x = randi( [ -2, 2 ], size, 1 );  
        y = randi( [ -2, 2 ], size, 1 );  
  
        Axpy_unb_59010759( a, x, y );  
  
    end  
fprintf('Size : %5d, 1,000 times : ', size);  
toc  
size = 10*size;  
end
```

4. Result ของ Homework 16.6.1

ผลการทดสอบของความถูกต้องของฟังก์ชัน

```
>> test_Axpy_unb_59010759
axpy
TEST PASSED
```

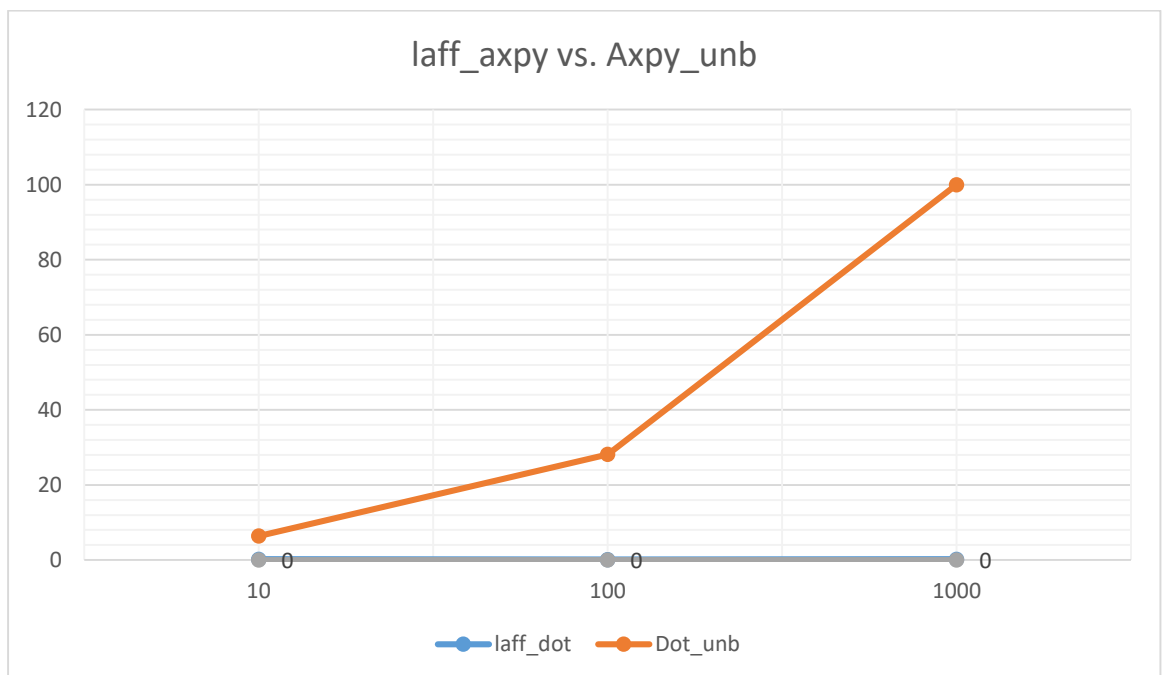
ผลการทดสอบประสิทธิภาพของฟังก์ชัน

```
>> test_time_Axpy_unb_59010759

Size :    10, 1,000 times : Elapsed time is 6.398407
seconds.

Size :   100, 1,000 times : Elapsed time is 28.132483
seconds.
```

5. วิเคราะห์ผลการใช้งานระหว่าง Homework 1.5.4.1 กับ Homework 16.6.1



ฟังก์ชัน laff_axpy ใช้เวลาน้อยกว่าฟังก์ชัน Axpy_unb เมื่อ row เพิ่มขึ้นเรื่อยๆ อันเนื่องจากการ Axpy_unb นั้นต้องใช้เวลาในการแบ่งเมทริกซ์ ซึ่งมีขนาดมากจะยิ่งแบ่งเยอะขึ้น

2. ระหว่างฟังก์ชันใน Homework 1.5.5.1 และ Homework 1.6.3.1

1. Source Code ของ Homework 1.5.5.1

laff_dot.m

```
function [a] = laff_dot(x, y)

[m_x, n_x] = size(x);
[m_y, n_y] = size(y);

if (m_x ~= 1 & n_x ~= 1)
    a = 'FAILED';
    return
end

if ( m_x * n_x ~= m_y * n_y )
    a = 'FAILED';
    return
end

a = 0;

if (n_x == 1)

    if (n_y == 1)
        for i=1:m_x
            a = a + x(i, 1) * y(i, 1);
        end

    else
        for i=1:m_x
            a = a + x(i, 1) * y(1, i);
        end
    end

else
    if (n_y == 1)
        for i=1:n_x
            a = a + x(1, i) * y(i, 1);
        end

    else
        for i=1:n_x
            a = a + x(1, i) * y(1, i);
        end
    end
end
end
```

test_laff_dot.m

```
x = randi( [ -2, 2 ], 4, 1 );
y = randi( [ -2, 2 ], 4, 1 );
```

```

a = randi( [ -2, 2 ], 1 );
z = randi( [ -2, 2 ], 3, 1 );

disp( 'dot product of column vector with column vector' )
if ( isequal( laff_dot_59010759( x, y ), x' * y ) )
    disp( 'TEST PASSED' )
else
    disp( 'TEST FAILED' )
end

disp( ' ' )

disp( 'dot product of column vector with row vector' )
if ( isequal( laff_dot_59010759( x, y' ), x' * y ) )
    disp( 'TEST PASSED' )
else
    disp( 'TEST FAILED' )
end

disp( ' ' )

disp( 'dot product of row vector with column vector' )
if ( isequal( laff_dot_59010759( x', y ), x' * y ) )
    disp( 'TEST PASSED' )
else
    disp( 'TEST FAILED' )
end

disp( ' ' )

disp( 'dot product of row vector with row vector' )
if ( isequal( laff_dot_59010759( x', y' ), x' * y ) )
    disp( 'TEST PASSED' )
else
    disp( 'TEST FAILED' )
end

disp( ' ' )

disp( 'dot product of column vector with column vector (wrong size)' )
if ( isequal( laff_dot_59010759( x, z ), 'FAILED' ) )
    disp( 'TEST PASSED' )
else
    disp( 'TEST FAILED' )
end

disp( ' ' )

disp( 'dot product of column vector with row vector (wrong size)' )
if ( isequal( laff_dot_59010759( x, z' ), 'FAILED' ) )
    disp( 'TEST PASSED' )
else
    disp( 'TEST FAILED' )
end

disp( ' ' )

disp( 'dot product of row vector with column vector (wrong size)' )

```

```

if ( isequal( laff_dot_59010759( x', z ), 'FAILED' ) )
    disp( 'TEST PASSED' )
else
    disp( 'TEST FAILED' )
end

disp( ' ' )

disp( 'dot product of row vector with row vector (wrong size)' )
if ( isequal( laff_dot_59010759( x', z' ), 'FAILED' ) )
    disp( 'TEST PASSED' )
else
    disp( 'TEST FAILED' )
end

disp( ' ' )

disp( 'dot product of column vector with alpha (wrong size)' )
if ( isequal( laff_dot_59010759( x, a ), 'FAILED' ) )
    disp( 'TEST PASSED' )
else
    disp( 'TEST FAILED' )
end

disp( ' ' )

disp( 'dot product of alpha with column vector (wrong size)' )
if ( isequal( laff_dot_59010759( a, x ), 'FAILED' ) )
    disp( 'TEST PASSED' )
else
    disp( 'TEST FAILED' )
end

disp( ' ' )

disp( 'dot product of row vector with alpha (wrong size)' )
if ( isequal( laff_dot_59010759( x', a ), 'FAILED' ) )
    disp( 'TEST PASSED' )
else
    disp( 'TEST FAILED' )
end

disp( ' ' )

disp( 'dot product of alpha with row vector (wrong size)' )
if ( isequal( laff_dot_59010759( a, x' ), 'FAILED' ) )
    disp( 'TEST PASSED' )
else
    disp( 'TEST FAILED' )
end

```

test_time_laff_dot.m

```

limit = 1000;
size = 10;
while size <= limit
tic

```

```
for time=1:limit

    x = randi( [ -2, 2 ], size, 1 );
    y = randi( [ -2, 2 ], size, 1 );

    laff_dot_59010759( x, y );

end
fprintf('Size : %5d, 1,000 times : ', size);
toc
size = 10*size;
end
```

2. Result ของ Homework 1.5.5.1

ผลการทดสอบของความถูกต้องของฟังก์ชัน

```
>> test_laff_dot_59010759
dot product of column vector with column vector
TEST PASSED

dot product of column vector with row vector
TEST PASSED

dot product of row vector with column vector
TEST PASSED

dot product of row vector with row vector
TEST PASSED

dot product of column vector with column vector (wrong
size)
TEST PASSED

dot product of column vector with row vector (wrong size)
TEST PASSED

dot product of row vector with column vector (wrong size)
TEST PASSED

dot product of row vector with row vector (wrong size)
TEST PASSED

dot product of column vector with alpha (wrong size)
TEST PASSED

dot product of alpha with column vector (wrong size)
TEST PASSED

dot product of row vector with alpha (wrong size)
TEST PASSED

dot product of alpha with row vector (wrong size)
TEST PASSED
```

ผลการทดสอบประสิทธิภาพของฟังก์ชัน

```
>> test_time_laff_dot_59010759

Size :      10, 1,000 times : Elapsed time is 0.161985
seconds.
```


Size : 100, 1,000 times : Elapsed time is 0.101619 seconds.

Size : 1000, 1,000 times : Elapsed time is 0.218669 seconds.


```
alpha_out = alpha;
```

```
return
```

test_Dot_unb.m

```
a = randi( [ -2, 2 ], 1 );  
x = randi( [ -2, 2 ], 4, 1 );  
y = randi( [ -2, 2 ], 4, 1 );  
  
disp( 'dot product' )  
if ( isequal( Dot_unb_59010759( a, x, y ), a + x' * y ) )  
    disp( 'TEST PASSED' )  
else  
    disp( 'TEST FAILED' )  
end
```

test_time_Dot_unb.m

```
limit = 1000;  
size = 10;  
while size <= limit  
    tic  
        for time=1:limit  
  
            x = randi( [ -2, 2 ], size, 1 );  
            y = randi( [ -2, 2 ], size, 1 );  
  
            Dot_unb_59010759( 0, x, y );  
  
        end  
        fprintf('Size : %5d, 1,000 times : ', size);  
        toc  
        size = 10*size;  
    end
```

4. Result ของ Homework 1.6.3.1

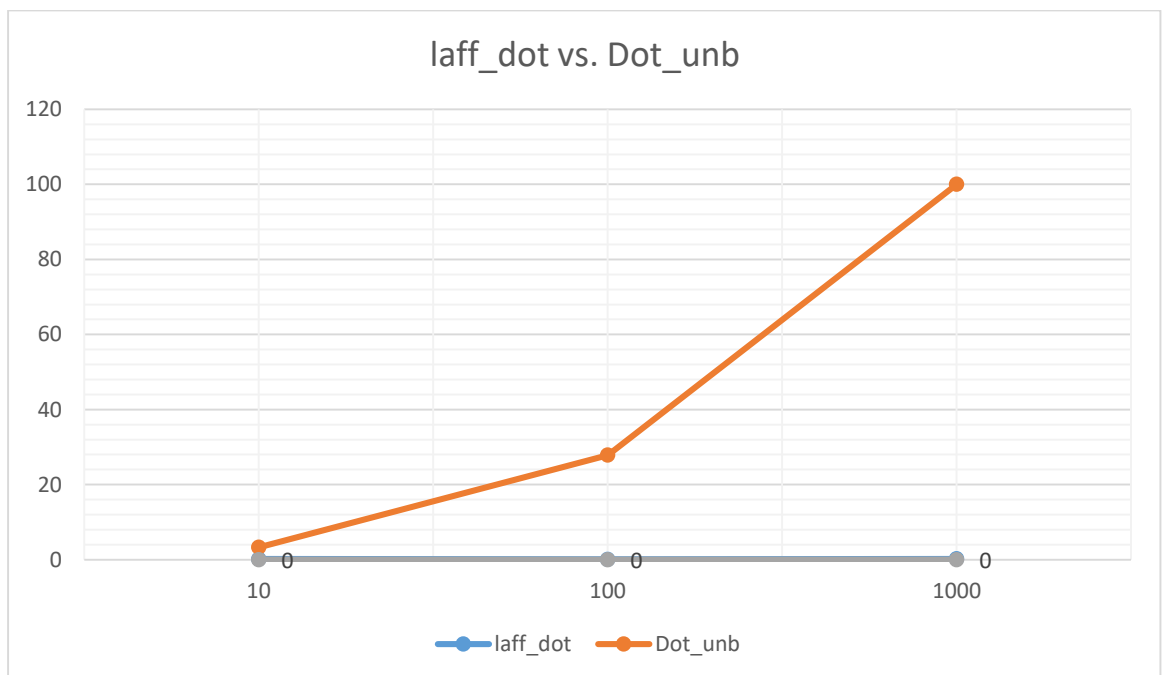
ผลการทดสอบของความถูกต้องของฟังก์ชัน

```
test_Dot_unb_59010759
dot product
TEST PASSED
```

ผลการทดสอบประสิทธิภาพของฟังก์ชัน

```
>> test_time_Dot_unb_59010759
Size :    10, 1,000 times : Elapsed time is 3.266834
seconds.
Size :   100, 1,000 times : Elapsed time is 27.839730
seconds.
```

5. วิเคราะห์ผลการใช้งานระหว่าง Homework 1.5.4.1 กับ Homework 16.6.1



ฟังก์ชัน laff_dot ใช้เวลาน้อยกว่าฟังก์ชัน Dot_unb เมื่อ row ของ vector เพิ่มขึ้นเรื่อยๆ อันเนื่องจากการ Dot_unb นั้นต้องใช้เวลาในการแบ่งเมทริกซ์ ยิ่งมีขนาดมากจะยิ่งแบ่งเยอะขึ้น

3. ระหว่างฟังก์ชันใน Homework 4.3.2.3 และฟังก์ชันที่สร้างเอง

1. Source Code ของ Homework 4.3.2.5

Trmvp_ln_unb_var1.m

```
function [ y_out ] = Trmvp_ln_unb_var1( L, x, y )

[ LTL, LTR, ...
  LBL, LBR ] = FLA_Part_2x2( L, ...
                              0, 0, 'FLA_TL' );

[ xT, ...
  xB ] = FLA_Part_2x1( x, ...
                       0, 'FLA_TOP' );

[ yT, ...
  yB ] = FLA_Part_2x1( y, ...
                       0, 'FLA_TOP' );

while ( size( LTL, 1 ) < size( L, 1 ) )

    [ L00,  L01,      L02,  ...
      l10t, lambda11, l12t, ...
      L20,  L21,      L22 ] = FLA_Repart_2x2_to_3x3( LTL, LTR, ...
                                                       LBL, LBR, ...
                                                       1, 1, 'FLA_BR'
    );

    [ x0, ...
      chi1, ...
      x2 ] = FLA_Repart_2x1_to_3x1( xT, ...
                                     xB, ...
                                     1, 'FLA_BOTTOM' );

    [ y0, ...
      psi1, ...
      y2 ] = FLA_Repart_2x1_to_3x1( yT, ...
                                     yB, ...
                                     1, 'FLA_BOTTOM' );

    %-----%

    psi1 = laff_dot_59010759( l10t, x0 ) + psi1;
    psi1 = laff_dot_59010759( lambda11, chi1 ) + psi1;

    %-----%

    [ LTL, LTR, ...
      LBL, LBR ] = FLA_Cont_with_3x3_to_2x2( L00,  L01,      L02,
    ...
                                             l10t, lambda11, l12t,
    ...
                                             L20,  L21,      L22, ...
                                             'FLA_TL' );

    [ xT, ...
      xB ] = FLA_Cont_with_3x1_to_2x1( x0, ...
                                       chi1, ...
                                       x2, ...
                                       'FLA_TOP' );
```



```

xB ] = FLA_Part_2x1( x, ...
                    0, 'FLA_TOP' );
[ yT, ...
yB ] = FLA_Part_2x1( y, ...
                    0, 'FLA_TOP' );

while ( size( LTL, 1 ) < size( L, 1 ) )

    [ L00,  l01,      L02,  ...
      l10t, lambda11, l12t, ...
      L20,  l21,      L22 ] = FLA_Repart_2x2_to_3x3( LTL, LTR, ...
                                                    LBL, LBR, ...
                                                    1, 1, 'FLA_BR'
    );

    [ x0, ...
      chi1, ...
      x2 ] = FLA_Repart_2x1_to_3x1( xT, ...
                                    xB, ...
                                    1, 'FLA_BOTTOM' );

    [ y0, ...
      psi1, ...
      y2 ] = FLA_Repart_2x1_to_3x1( yT, ...
                                    yB, ...
                                    1, 'FLA_BOTTOM' );

    %-----%

    psi1 = laff_axpy_59010759( chi1 , lambda11 , psi1 );
    y2 = laff_axpy_59010759( chi1 , l21 , y2 );

    %-----%

    [ LTL, LTR, ...
      LBL, LBR ] = FLA_Cont_with_3x3_to_2x2( L00,  l01,      L02,
...
                                           l10t, lambda11, l12t,
...
                                           L20,  l21,      L22, ...
                                           'FLA_TL' );

    [ xT, ...
      xB ] = FLA_Cont_with_3x1_to_2x1( x0, ...
                                       chi1, ...
                                       x2, ...
                                       'FLA_TOP' );

    [ yT, ...
      yB ] = FLA_Cont_with_3x1_to_2x1( y0, ...
                                       psi1, ...
                                       y2, ...
                                       'FLA_TOP' );

end

y_out = [ yT
          yB ];

return

```

test_Trmvp_ln_unb_var2.m

```
A = randi( [ -2, 2 ], 4, 4 )
x = randi( [ -2, 2 ], 4, 1 )
y = randi( [ -2, 2 ], 4, 1 )

if ( isequal( Trmvp_ln_unb_var2_59010759( A, x, y ), tril( A ) * x +
y ) )
    disp( 'Trmvp_ln_unb_var2 appears to be correct' )
else
    disp( 'Trmvp_ln_unb_var2 has a problem' )
end
```

test_time_Trmvp_ln_unb_var2.m

```
limit = 1000;
size = 10;
while size <= limit
tic
    for time=1:limit

        A = randi( [ -2, 2 ], size, size );
        x = randi( [ -2, 2 ], size, 1 );
        y = randi( [ -2, 2 ], size, 1 );

        Trmvp_ln_unb_var2_59010759( A, x, y );

    end
    fprintf('Size : %5d, 1,000 times : ', size);
    toc
    size = 10*size;
end
```


2. Result ของ Homework 4.3.2.5

ผลการทดสอบของความถูกต้องของฟังก์ชัน

```
>> test_Trmvp_ln_unb_var1_59010759
```

A =

-2	2	-2	-1
-2	-2	-1	0
0	0	-2	-2
1	0	1	1

x =

-1
1
1
1

y =

0
-2
-1
2

Trmvp_ln_unb_var1 appears to be correct

```
>> test_Trmvp_ln_unb_var2_59010759
```

A =

2	-2	-2	-2
0	-2	2	-1
0	-2	1	0
-1	0	-2	-2

x =

-2
1
-1
1

y =

0
0
-2
-2

Trmvp_ln_unb_var2 appears to be correct

ผลการทดสอบประสิทธิภาพของฟังก์ชัน

```
>> test_time_Trmvp_ln_unb_var1_59010759
Size :    10, 1,000 times : Elapsed time is 7.219494
seconds.
Size :   100, 1,000 times : Elapsed time is 96.388200
seconds.
```

```
>> test_time_Trmvp_ln_unb_var2_59010759
Size :    10, 1,000 times : Elapsed time is 9.325294
seconds.
Size :   100, 1,000 times : Elapsed time is 76.779085
seconds.
```

3. Source Code ของฟังก์ชันที่สร้างขึ้นเอง

test_Trmvp_Ln_unb_var2.m

```
function [ y_out ] = Trmvp_Ln( L, x, y )

    [m_L, n_L] = size(L);
    [m_x, n_x] = size(x);
    [m_y, n_y] = size(y);

    if (m_L ~= n_L) & ( m_x * n_x ~= m_y * n_y )
        y_out = 'FAILED';
        return
    end

    for i=1:m_L
        for j=1:i
            y(i) = L(i, j) * x(j) + y(i);
        end
    end

    y_out = y
end
```

test_Trmvp_Ln.m

```
A = randi( [ -2, 2 ], 4, 4 );
x = randi( [ -2, 2 ], 4, 1 );
y = randi( [ -2, 2 ], 4, 1 );

if ( isequal( Trmvp_Ln_59010759( A, x, y ), tril( A ) * x + y ) )
    disp( 'Trmvp_Ln appears to be correct' )
else
    disp( 'Trmvp_Ln has a problem' )
end
```

test_time_Trmvp_Ln.m

```
limit = 1000;
size = 10;
while size <= limit
    tic
        for time=1:limit

            A = randi( [ -2, 2 ], size, size );
            x = randi( [ -2, 2 ], size, 1 );
            y = randi( [ -2, 2 ], size, 1 );

            Trmvp_Ln_59010759( A, x, y );

        end
        fprintf('Size : %5d, 1,000 times : ', size);
        toc
        size = 10*size;
    end
```

4. Result ของฟังก์ชันที่สร้างขึ้นเอง

ผลการทดสอบของความถูกต้องของฟังก์ชัน

```
>> test_Trmvp_ln_59010759
```

A =

-1	2	-1	-2
0	2	-1	-2
-2	0	2	1
-2	-2	-2	1

x =

1
0
0
-1

y =

1
-2
1
-2

Trmvp_ln appears to be correct

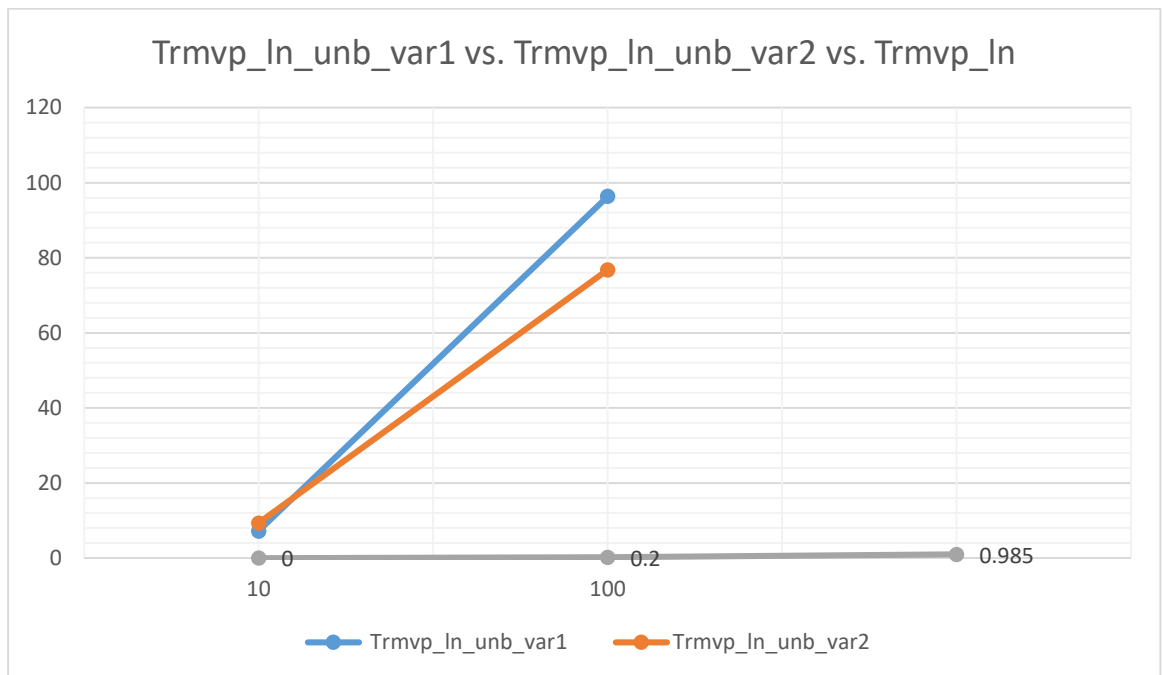
ผลการทดสอบประสิทธิภาพของฟังก์ชัน

```
>> test_time_Trmvp_ln_59010759
```

Size : 10, 1,000 times : Elapsed time is 0.208559 seconds.

Size : 100, 1,000 times : Elapsed time is 0.985248 seconds.

5. วิเคราะห์ผลการใช้งานระหว่าง Homework 1.5.4.1 กับ Homework 16.6.1



ในตอนแรกฟังก์ชัน Trmvp_ln_unb_var1 จะเร็วกว่า Trmvp_ln_unb_var2 เมื่อข้อมูลอยู่ในช่วงที่น้อยกว่า 20 แถวโดยประมาณ แล้วหลังจากนั้น Trmvp_ln_unb_var2 จะเร็วกว่าอย่างเห็นได้ชัดเนื่องจาก ฟังก์ชัน laff_axpy ใน Trmvp_ln_unb_var2 นั้นทำงานเร็วกว่าฟังก์ชัน laff_dot ใน Trmvp_ln_unb_var1

ส่วนฟังก์ชันที่สร้างขึ้นมานั้นเร็วกว่า Trmvp_ln_unb_var1 และ Trmvp_ln_unb_var2 เนื่องจากไม่ต้องเสียเวลาในการแบ่งเมทริกซ์ และการใช้ loop วนนั้นทำได้เร็วกว่า