Symmetric Matrix-Vector Multiplication Variant 1

Function Symv_unb_var1(A, x, y) (at the end of the file) implements the operation

$$Ax + y,$$

where A is symmetric and only stored in the lower triangular part of array A, corresponding to loop invariant.

$$\begin{pmatrix} y_T \\ y_B \end{pmatrix} = \begin{pmatrix} A_{TL} x_T + & \hat{y}_T \\ & \hat{y}_B \end{pmatrix}$$

The below tests this function. Notice that now the matrix is square (m x m)

```
m = 4;
```

Create random A, x, and y.

```
A = randi( [ -3, 3 ], [ m, m ] )
x = randi( [ -2, 2 ], [ m, 1 ] )
y = randi( [ -2, 2 ], [ m, 1 ] )
```

Notice that the matrix is NOT symmetric. To make it symmetric, we replace the strictly upper triangular part with the transpose of the strictly lower triangular part. In the command window you can type "help tril" to see how that function works.

```
Asym = tril( A ) + tril( A,-1 )'
```

Compute $Ax + y$ with Asym

```
Asym * x + y
```

Compare this with the result of the function at the end of this file, but using the original matrix A. Before you fix the function, it gives the wrong answer (unless, by accident, the randomly generated problem is special):

```
Symv_unb_var1( A, x, y )
```

```
if isequal( Symv_unb_var1( A, x, y ), Asym * x + y )
    disp( 'All is well' );
else
    disp( 'Hmmm, something seems to be wrong' );
end
```

The function SymMatVec1(A, x, y) follows below. For reference, we give the algorithm for Variant 1:

<table>
<tr><th>Step</th><th>Algorithm: $y := Ax + y$</th></tr>
<tr><td>1a</td><td>$\{ y = \hat{y} \}$</td></tr>
<tr><td>4</td><td>$A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array} \right), x \rightarrow \left(\begin{array}{c} x_T \\ \hline x_B \end{array} \right), y \rightarrow \left(\begin{array}{c} y_T \\ \hline y_B \end{array} \right)$
where A_{TL} is 0×0, x_T has 0 rows, y_T has 0 rows</td></tr>
<tr><td>2</td><td>$\left\{ \left(\begin{array}{c} y_T \\ \hline y_B \end{array} \right) = \left(\begin{array}{c} A_{TL} x_T + \hat{y}_T \\ \hline \hat{y}_B \end{array} \right) \right\}$</td></tr>
<tr><td>3</td><td>while $m(A_{TL}) < m(A)$ do</td></tr>
<tr><td>2,3</td><td>$\left\{ \left(\begin{array}{c} y_T \\ \hline y_B \end{array} \right) = \left(\begin{array}{c} A_{TL} x_T + \hat{y}_T \\ \hline \hat{y}_B \end{array} \right) \wedge m(A_{TL}) < m(A) \right\}$</td></tr>
<tr><td>5a</td><td>$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array} \right) \rightarrow \left(\begin{array}{c|cc} A_{00} & a_{01} & A_{02} \\ \hline a_{10}^T & \alpha_{11} & a_{12}^T \\ A_{20} & a_{21} & A_{22} \end{array} \right), \left(\begin{array}{c} x_T \\ \hline x_B \end{array} \right) \rightarrow \left(\begin{array}{c} x_0 \\ \hline \chi_1 \\ x_2 \end{array} \right), \left(\begin{array}{c} y_T \\ \hline y_B \end{array} \right) \rightarrow \left(\begin{array}{c} y_0 \\ \hline \psi_1 \\ y_2 \end{array} \right)$
where α_{11} is 1×1, χ_1 has 1 row, ψ_1 has 1 row</td></tr>
<tr><td>6</td><td>$\left\{ \left(\begin{array}{c} y_0 \\ \psi_1 \\ y_2 \end{array} \right) = \left(\begin{array}{ll} A_{00} x_0 + & \hat{y}_0 \\ & \hat{\psi}_1 \\ & \hat{y}_2 \end{array} \right) \right\}$</td></tr>
<tr><td>8</td><td>$y_0 := \chi_1 (a_{10}^T)^T + y_0$
$\psi_1 := a_{10}^T x_0 + \alpha_{11} \chi_1 + \psi_1$</td></tr>
<tr><td>7</td><td>$\left\{ \left(\begin{array}{c} y_0 \\ \psi_1 \\ y_2 \end{array} \right) = \left(\begin{array}{r} A_{00} x_0 + \chi_1 (a_{10}^T)^T + \hat{y}_0 \\ a_{10}^T x_0 + \alpha_{11} \chi_1 + \hat{\psi}_1 \\ \hat{y}_2 \end{array} \right) \quad (\text{Note: } (a_{10}^T)^T \chi_1 = \chi_1 (a_{10}^T)^T) \right\}$</td></tr>
<tr><td>5b</td><td>$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array} \right) \leftarrow \left(\begin{array}{cc|c} A_{00} & a_{01} & A_{02} \\ a_{10}^T & \alpha_{11} & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22} \end{array} \right), \left(\begin{array}{c} x_T \\ \hline x_B \end{array} \right) \leftarrow \left(\begin{array}{c} x_0 \\ \chi_1 \\ \hline x_2 \end{array} \right), \left(\begin{array}{c} y_T \\ \hline y_B \end{array} \right) \leftarrow \left(\begin{array}{c} y_0 \\ \psi_1 \\ \hline y_2 \end{array} \right)$</td></tr>
<tr><td>2</td><td>$\left\{ \left(\begin{array}{c} y_T \\ \hline y_B \end{array} \right) = \left(\begin{array}{c} A_{TL} x_T + \hat{y}_T \\ \hline \hat{y}_B \end{array} \right) \right\}$</td></tr>
<tr><td></td><td>endwhile</td></tr>
<tr><td>2,3</td><td>$\left\{ \left(\begin{array}{c} y_T \\ \hline y_B \end{array} \right) = \left(\begin{array}{c} A_{TL} x_T + \hat{y}_T \\ \hline \hat{y}_B \end{array} \right) \wedge \neg (m(A_{TL}) < m(A)) \right\}$</td></tr>
<tr><td>1b</td><td>$\{ y = Ax + \hat{y} \}$</td></tr>
</table>

Below implement SymvUnbVar1(A, x, y)

```
function [ y_out ] = Symv_unb_var1( A, x, y )
```

. $y = \widehat{y}$

```
[ ATL, ATR, ...
  ABL, ABR ] = FLA_Part_2x2( A, ...
                             0, 0, 'FLA_TL' );

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

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

. $\begin{pmatrix} y_T \\ y_B \end{pmatrix} = \begin{pmatrix} A_{TL}x_T + \widehat{y}_T \\ \widehat{y}_B \end{pmatrix}$

```
while ( size( ATL, 1 ) < size( A, 1 ) )
```

. $\begin{pmatrix} y_T \\ y_B \end{pmatrix} = \begin{pmatrix} A_{TL}x_T + \widehat{y}_T \\ \widehat{y}_B \end{pmatrix} \wedge m(A_{TL}) < m(A)$

```
  [ A00,  a01,     A02,  ...
    a10t, alpha11, a12t, ...
    A20,  a21,     A22 ] = FLA_Repart_2x2_to_3x3( ATL, ATR, ...
                                                  ABL, ABR, ...
                                                  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' );
```

. $\begin{pmatrix} y_0 \\ \psi_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} A_{00}x_0 + \widehat{y}_0 \\ \widehat{\psi}_1 \\ \widehat{y}_2 \end{pmatrix}$

```
  %------------------------------------------------------------%

  y0 = chi1 * a10t' + y0;
  psi1 = a10t * x0 + psi1;
  psi1 = alpha11 * chi1 + psi1;

  %------------------------------------------------------------%
```

$$\left(\begin{array}{c} y_0 \\ \psi_1 \\ y_2 \end{array}\right) = \left(\begin{array}{r} A_{00}x_0 + \chi_1 (a_{10}^T)^T + \widehat{y}_0 \\ a_{10}^T x_0 + \alpha_{11}\chi_1 + \widehat{\psi}_1 \\ \widehat{y}_2 \end{array}\right)$$

```
    [ ATL, ATR, ...
      ABL, ABR ] = FLA_Cont_with_3x3_to_2x2( A00,  a01,     A02,  ...
                                             a10t, alpha11, a12t, ...
                                             A20,  a21,     A22, ...
                                             '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' );
```

$$\left(\begin{array}{c} y_T \\ y_B \end{array}\right) = \left(\begin{array}{r} A_{TL}x_T + \widehat{y}_T \\ \widehat{y}_B \end{array}\right)$$

```
  end
```

$$\left(\begin{array}{c} y_T \\ y_B \end{array}\right) = \left(\begin{array}{r} A_{TL}x_T + \widehat{y}_T \\ \widehat{y}_B \end{array}\right) \wedge \neg(m(A_{TL}) < m(A))$$

$$y = Ax + \widehat{y}$$

```
  y_out = [ yT
            yB ];

end
```