Included Programs for Substitute Frame Analysis by STIFFNESS METHOD

appVersion(4) = "1.0.8348.30405" appVersion(−4) = "1.0.8348.30405"

Program 1 : Pairing non-diagonal indices of stiffness matrix

$$Pair\_STF(B\#) := \left| \begin{array}{l} nr\# := \text{rows}(B\#) - 1 \\ \text{for } k \in [1..(nr\#)] \\ \quad Z\#_k := \text{stack}(B\#_k, B\#_{k+1}) \\ Z\# \end{array} \right.$$

Define

$$X_0 := \text{augment}(0, 0)$$

Define

$$S_{BEAM} := \begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix}$$

Program 2 : To find the Stiffness Natrix : Calls Program 1
fdn#=1 if foundatio column base is FIXED ", fdn#=0.75 if foundation column base is PINNED

$$Find\_K(L\#, I\#_{UC}, I\#_{LC}, I\#_{beam}, H\#_{UC}, H\#_{LC}, fdn\#) := \left| \begin{array}{l} nr := \text{rows}(L\#) + 1 \\ \text{for } j \in [1..(nr)] \\ \quad M\#_j := 4 \cdot E \cdot \begin{cases} \frac{I\#_{UC}{}_j}{H\#_{UC}{}_j} + \frac{I\#_{LC}{}_j \cdot fdn\#}{H\#_{LC}{}_j} + \frac{I\#_{beam}{}_j}{L\#_j} & \text{if } j = 1 \\ \frac{I\#_{UC}{}_j}{H\#_{UC}{}_j} + \frac{I\#_{LC}{}_j \cdot fdn\#}{H\#_{LC}{}_j} + \frac{I\#_{beam}{}_j}{L\#_{j-1}} + \frac{I\#_{beam}{}_j}{L\#_j} & \text{if } 1 < j < nr \\ \frac{I\#_{UC}{}_j}{H\#_{UC}{}_j} + \frac{I\#_{LC}{}_j \cdot fdn\#}{H\#_{LC}{}_j} + \frac{I\#_{beam}{}_{j-1}}{L\#_{j-1}} & \text{otherwise} \end{cases} \\ (STF := \text{Diag}(M\#)) = \text{"Create diagonal matrix"} \\ C\# := [1..nr] \\ (D\# := Pair\_STF(C\#)) = \text{"Indices of non-diagonal elements"} \\ \text{Clear}(j) \\ \text{for } j \in [1..\text{rows}(L\#)] \\ \quad \left| \begin{array}{l} \left(M2\#_j := 2 \cdot E \cdot \frac{I\#_{beam}{}_j}{L\#_j}\right) = \text{"Non-diagonal elements"} \\ \left(r_j := D\#_{j\,1}\right) = \text{"Row index of non-diagonal elements"} \\ \left(c_j := D\#_{j\,2}\right) = \text{"Column index of non-diagonal elements"} \\ \left(STF_{r_j\,c_j} := M2\#_j\right) = \text{"Assign appropriate M2\# to STF"} \\ \left(STF_{c_j\,r_j} := M2\#_j\right) = \text{"Assign appropriate M2\# to STF"} \end{array} \right. \\ STF \end{array} \right.$$

PROGRAM 3 : To find joint moments & ST

$$Find\_ST\#(Z, aa(\blacksquare), kk) := \left| \begin{array}{l} j := [1..\text{rows}(Z)] \\ jm_j := aa(Z_j) \\ st_j := -kk^{-1} \cdot jm_j \\ st \end{array} \right.$$

For use with Main Program

Program 4 : Any function can be passed to this program

$$Arrange(Z\#, a(\blacksquare)) := \left| \begin{array}{l} j := [1..\text{rows}(Z\#)] \\ D_j := a(Z\#_j) \end{array} \right.$$

Program 5 : Find lower column moments.
Valid even if foundation columns have
equal or diff heights

$$Find\_MLC(st, d) := \left| \begin{array}{l} \text{for } j \in [1..\text{rows}(st)] \\ \quad \left| \begin{array}{l} a\# := st_j \\ A\#_j := \overrightarrow{a\# \cdot d} \end{array} \right. \\ A\# \end{array} \right.$$

Program 6 : Calls Program 4

$$Pair(B\#) := \left| \begin{array}{l} nrr := \overrightarrow{\text{rows}(B\#)}_1 - 1 \\ \text{for } k \in [1..(nrr)] \\ \quad \left| Z\#_k := \begin{bmatrix} YY_k \\ YY_{k+1} \end{bmatrix} \right. \\ FF(YY) := Z\# \\ Arrange(B\#, FF(YY)) \end{array} \right.$$

Program 7 : Main program to find support moments, upper & lower column moments
Calls Programs 3, 4, 5 & 6

$$NEW\_BMM(ff\#, K\#, h\#_{UC}, h\#_{LC}, I_b, I_C, I_L, fdn\#, L\#) := \left| \begin{array}{l} \text{for } j \in [1..\text{cols}(ff\#)] \\ \quad \left| fem\#_j := \frac{1}{12} \cdot \left(\overrightarrow{ff\#_{1\,j} \cdot L\#^2} \right) \right. \\ nrr := \overrightarrow{\text{rows}(fem\#)}_1 + 1 \\ \text{for } j \in [1..nrr] \\ \quad \left| M\#_j := \begin{cases} -Y_j & \text{if } j = 1 \\ Y_{j-1} - Y_j & \text{if } 1 < j < nrr \\ Y_{nrr-1} & \text{otherwise} \end{cases} \right. \\ F(Y) := M\# \\ st := Find\_ST\#(fem\#, F(Y), K\#) \\ stt := Pair(st) \\ \text{for } j \in [1..\text{cols}(ff\#)] \\ \quad \left| \begin{array}{l} \text{for } k \in [1..\text{rows}(L)] \\ \quad \left| M_{sup}{}_{j\,k} := \left(\frac{stt_{j\,k} \cdot E \cdot I_{b\,k}}{L_k} \right)^T \cdot S_{BEAM} + \begin{bmatrix} -fem\#_{j\,k} \\ fem\#_{j\,k} \end{bmatrix}^T \right. \end{array} \right. \\ Col\_bott\# := \frac{\overrightarrow{I_L}}{h\#_{LC}} \cdot 4 \cdot E \cdot fdn\# \\ M_{LCC} := Find\_MLC(st, Col\_bott\#) \\ Col\_upp\# := \frac{\overrightarrow{I_C}}{h\#_{UC}} \cdot 4 \cdot E \\ M_{UCC} := Find\_MLC(st, Col\_upp\#) \\ \text{augment}(M_{sup}, M_{UCC}, M_{LCC}) \end{array} \right.$$

Program 9 : Sum of Upper & Lpwer Col Moments

$$Sum\_Col\_M(M\#) := \left| \begin{array}{l} j := [1..\text{rows}(M\#)] \\ A\#_j := \overrightarrow{|M\#_{j\,1}| + |M\#_{j\,2}|} \end{array} \right.$$

Program 8A : MAX ABSOLUTE VALUE :
Nested Array of Diff Sizes

$$Max\_Abs(M\#) := \left| \begin{array}{l} \text{for } j \in [1..(\text{rows}(M\#))] \\ \quad \left| \begin{array}{l} tmp := M\#_j \\ Z\#_j := \text{Max}\left(\overrightarrow{|tmp|}\right) \end{array} \right. \\ Z\# \end{array} \right.$$

Program 8B : Min ABSOLUTE VALUE :
Nested Array of Diff Sizes

$$Min\_Abs(M\#) := \left| \begin{array}{l} \text{for } j \in [1..(\text{rows}(M\#))] \\ \quad \left| \begin{array}{l} tmp := M\#_j \\ Z\#_j := \text{Min}\left(\overrightarrow{|tmp|}\right) \end{array} \right. \\ Z\# \end{array} \right.$$

Program to find SF, Max Span Moment and Distance to Max Span Moment

PROGRAM 10 : For any Loading Case Find SF, Max Span BM, X_max

```
Param(M#, f#, L#) :=
    for j ∈ [1..rows(f#)]
        W_j := f#_j · L#_j
        V1_j := 0.5 · W_j − ( Σ M#_j / L#_j )
        V2_j := −(W_j − V1_j)
        M_max_span_j := −( (V1_j)^2 / (2 · f#_j) + M#_j_1 )
        X_max_j := V1_j / f#_j
    [ V1  V2  M_max_span  X_max ]
```

$W = f \cdot L$ — *Beam load*

$V1 = \left(\frac{W}{2} - \frac{(M_L + M_R)}{L} \right)$ — *SF LHS of beam +ve for plot*

$V2 = -(W - V1)$ — *SF RHS of beam -ve for plot*

$M_{max\_span} = \frac{V1^2}{2 \cdot f} + M_L$ — *Max span moment on beam*

$X_{max} = \frac{V1}{f}$ — *Dist to location of Max span moment from LHS of beam support*

PROGRAM 11 : Calls PROGRAM 10 : Find SF, Max Span BM & X_max for ALL Loading Cases

```
Find_All(M#, ff, L#) :=
    for j ∈ [1..rows(M#)]
        M_case_j := M#_j[1..rows(L#)]
        B_j := Param(M_case_j, ff_j, L#)
    B
```

PROGRAM 12 : Extract All SFF

```
Calc_SF(res) :=
    for j ∈ [1..rows(res)]
        A_j := res_j_1[1..2]
        if j = 1
            B := A_1
        else
            B := stack(B, A_j)
    B
```

MAX ABSOLUTE VALUE : Nested Array of Diff Sizes — **Disabled**

```
Abs_Array(M#) :=
    for j ∈ [1..(rows(M#))]
        tmp := M#_j
        Z#_j := |→tmp|
    Z#
```

Program 13 To round : By Hean Giraud

```
Round(U, n) := trunc(U) − mod(trunc(U), n)
```

PROGRAM 14 : Extract ALL Max Span BMM

```
Span_M(res) :=
    nc := cols(→res)_1
    for j ∈ [1..rows(res)]
        A_j := res_j_1(nc − 1)
    A
```

PROGRAM 15 : Extract All X-max

```
X_Max(res) :=
    nc := cols(→res)_1
    for j ∈ [1..rows(res)]
        A_j := res_j_1 nc
    A
```

Program 16 : Arrange Support moments

```
Arrange_M_Sup1(M#, n) :=
    nc := rows(M#)
    j := [1..nc]
    A1_j := col(→M#^T_n, j)
```

Program 17 : BMD for Any Case. Calls Program 16

$$BMD\_Any\left(m\#_{sup}, sff, L\#, ff, n_{case}, n_b\right) := \left| \begin{array}{l} sf1\# := \mathrm{col}\left(sff, 1\right)^{T} \\ m1\# := Arrange\_M\_Sup1\left(m\#_{sup}, 1\right) \\ \text{for } j \in \left[1..\mathrm{rows}\left(L\#\right)\right] \\ \quad \left| x\#_{j} := \left[0, 0.20\,\mathrm{m}..L\#_{j}\right] \right. \\ \mathrm{Clear}\left(j\right) \\ \text{for } j \in \left[1..\mathrm{rows}\left(x\#_{n_b}\right)\right] \\ \quad \left| A_{j} := \left| m1\#_{n_{case}\,n_b} \right| - sf1\#_{n_{case}\,n_b} \cdot x\#_{n_b\,j} + ff_{n_{case}\,n_b} \cdot \frac{\left(x\#_{n_b\,j}\right)^2}{2} \right. \\ \left[\frac{x\#_{n_b}}{\mathrm{m}} \quad \frac{A}{kNm} \right] \end{array} \right.$$

Program 18 : BMD for ALL Case. Calls Program 16

$$BMD\_All\_Cases\left(m\#_{sup}, sff, L\#, ff\right) := \left| \begin{array}{l} n := 0 \\ \text{for } j \in \left[1..\mathrm{cols}\left(ff\right)\right] \\ \quad \text{for } k \in \left[1..\mathrm{rows}\left(L\#\right)\right] \\ \quad\quad \left| n := n + 1 \right. \\ \quad\quad \left| A_{n} := BMD\_Any\left(m\#_{sup}, sff, L\#, ff, j, k\right) \right. \\ A^{T} \end{array} \right.$$

Cordinates of Max Span Moment for all Cases - For Plotting

Program 19 : Cordinates of Max Span Moment for all Cases - For Plotting

$$Get\_L\left(L\#\right) := \left| \begin{array}{l} j := \left[1..\mathrm{rows}\left(L\#\right)\right] \\ A\#_{j} := \begin{cases} 0 & \text{if } j = 1 \\ \sum\left(L\#_{\left[1..(j-1)\right]}\right) & \text{otherwise} \end{cases} \end{array} \right.$$

Program 20 : Cordinates of Max Span Moments for a single Case

$$Get\_Pts\#\left(x\_max\#, M\_span\#, L\#\right) := \left| \begin{array}{l} \text{for } j \in \left[1..\mathrm{rows}\left(x\_max\#\right)\right] \\ \quad \left| P_{j} := \left[\frac{x\_max\#_{j}}{\mathrm{m}} + \frac{Get\_L\left(L\#\right)}{\mathrm{m}}_{j} \quad \frac{M\_span\#_{j}}{kNm} \right] \right. \\ P \end{array} \right.$$

Program 21 Calls Program 20 for Cordinates of All cases

$$Pts\_All\_Cases\left(x\_max, M\_span, L\#\right) := \left| \begin{array}{l} \text{for } j \in \left[1..\mathrm{rows}\left(x\_max\right)\right] \\ \quad \left| A_{j} := Get\_Pts\#\left(x\_max_{j}, M\_span_{j}, L\#\right) \right. \\ A \end{array} \right.$$

Program 22 : To find dist to each support from LHS

$$F\left(L\#\right) := \left| \begin{array}{l} n\# := 0 \\ \text{for } j \in \left[1..\left(\mathrm{rows}\left(L\#\right) - 1\right)\right] \\ \quad \left| X_{\left[(j+n\#)..(j+n\#+1)\right]} := \sum L\#_{\left[1..(j)\right]} \right. \\ \quad\quad \left| n\# := n\# + 1 \right. \\ X \end{array} \right.$$

Program 23 : X cordinates to plot xupport moments

```
Sup_Xcord(L1#) :=
    nr := rows(L1#)
    for k ∈ [1..nr]
        C := stack(0, L1#)                          if k = 1
             stack(0, F(L1#), Σ L1#)                otherwise
    C
```

Program 24 : Get BMD Cases

```
Find_BMD_Cases(bmd, L#, ff) :=
    nr := rows(L#)
    for j ∈ [1..cols(ff)]
        n#_j := (j·nr − (nr − 1))
        A_j := eval(bmd[n#_j..(j·nr)]^T)
    A
```

Program 25 : Plot BMD for a given CASE

```
Plot_BMD(n#) :=
    for j ∈ [1..(rows(L))]
        A_j := col(BMD^T n#_j, 1) + Get_L(L)/m_j
        B_j := col(BMD^T n#_j, 2)
        C_j := augment(A_j, B_j)
        D_j := augment(col(C_j, 1)_1, col(C_j, 2)_1)
    D
```

Program 26 : Find absolute values of nested Msup and etc.

```
Find_Abs(M#, x_cord) :=
    a# :=⇉|M#|
    b# := sys2mat(mat2sys(a#))
    augment(x_cord, b#)
```

Program 27 : Arrange Support Moment values & X cordinates

```
Arrange_M_Sup_All(M#, x_cord) :=
    for j ∈ [1..cols(M#^T)]
        A_j := col(M#^T, j)
        B_j := Find_Abs(A_j, x_cord)
    B
```

Program 28 : Labels to plot

```
Plot_Labs(X#) :=
    for j ∈ [1..rows(X#)]
      if j = 1
        A#_j := var2str(X#_(j,2), 1)
        B# := eval(augment(X#_j, A#_j, 6, "red"))
      else
        A#_j := var2str(X#_(j,2), 1)
        B# := eval(stack(B#, augment(X#_j, A#_j, 6, "red")))
    B#
```

Program 29 : Arrange SF values to PLOT

```
Arrange_SF(x#, sf#) :=
    sf1 := col(sf#, 1)
    sf2 := col(sf#, 2)
    sff := augment(sf1, sf2)
    sff := sys2mat(mat2sys(sff))
    j := [1..rows(sff)]
    B#_j := augment(x# / m, sff / kN_j)
```

Program 30 :Labels to print SF values on SFD

```
Plot_Labs_SF(M#) :=
    nr := rows(M#)
    a# := col(M#, 2)
    a# := var2str(round(a#, 2), 1)
    aa#_[1..nr] := 6
    bb#_[1..nr] := "black"
    augment(M#, a#, aa#, bb#)
```

Program 31 : Labels to print SF zero x_cordinates on SFD

```
Find_Labs_ZeroSF(X#) :=
    nr := rows(X#)
    a# := var2str(round(X# / m, 2), 2)
    b#_[1..nr] := 6
    c#_[1..nr] := "red"
    d#_[1..nr] := 0
    augment((X# + Get_L(L)) / m, d#, a#, b#, c#)
```

Program 32 : Find max value of nested MMsup matrix

```
Find_Max_Abs(M#) :=
    a := sys2mat(mat2sys(M#))
    Max(Max(a))
```

Plotting Column Moments

Unlike BMD and SFD, plotting Column Moments is reatively difficult, The main reason is that both Column Moments and the Column locations have to be plotted on the X-azis. Hence, Column moment values are divided by total length of the horizontal beams. Then these values are converted to strings and plotted at the proper locations for both upper and lower columns.

Program 33 : X cordinates to plot Column moments

```
Col_X(L1#) :=
    nr := rows(L1#)
    for k ∈ [1..(nr + 1)]
        C_k := { 0                          if k = 1
               { Σ L1#_[1..(k - 1)]         otherwise
    C
```

Program 34: Plot Column Moments

```
Plot_Col_M(M1#, M2#) :=
    j := [1..rows(M1#)]
    B_j := stack(row(M1#, j), row(M2#, j))
```

Program 35: Stack relevant matrices

```
STACK(M#) :=
    nr := rows(M#)
    for j ∈ [1..nr]
        if j = 1
            B := row(M#_j, 2)
        else
            B := stack(B, row(M#_j, 2))
    B
```

Program 36: Io plot UPPER Column Moments

$$Plot\_UP\_Col\_M(M\#, L\#, n\#, H\#, N\#, SC\#) :=$$

$$X_{uc\_bot} := \left(\frac{N\# \cdot \frac{M\#}{kNm}\, n\#}{\sum\left(\frac{SC\# \cdot L\#}{m}\right)} + \frac{Col\_X(L\#)}{m} \right)$$

$$Y_{uc\_bot} := \text{matrix}(\text{rows}(L\#) + 1, 1)$$

$$M1_{uc\_bot} := \text{augment}\left(\frac{Col\_X(L\#)}{m}, N\# \cdot Y_{uc\_bot}\right)$$

$$M2_{uc\_bot} := \text{augment}(X_{uc\_bot}, N\# \cdot Y_{uc\_bot})$$

$$M_{uc\_bot} := Plot\_Col\_M(M1_{uc\_bot}, M2_{uc\_bot})$$

$$V1_{uc\_bot} := M1_{uc\_bot}$$

$$V2_{uc\_top} := \text{augment}\left(\frac{Col\_X(L\#)}{m}, \frac{N\# \cdot H\#}{m}\right)$$

$$V_{uc\_top} := Plot\_Col\_M(V1_{uc\_bot}, V2_{uc\_top})$$

$$X_{uc\_top} := \frac{\frac{M\#}{kNm}\, n\#}{\sum\left(\frac{SC\# \cdot L\#}{m}\right)} \cdot (-0.5) \cdot N\# + \frac{Col\_X(L\#)}{m}$$

$$Y_{uc\_top} := \frac{H\# \cdot N\#}{m}$$

$$M1_{uc\_top} := \text{augment}(X_{uc\_top}, Y_{uc\_top})$$

$$M2_{uc\_top} := \text{augment}\left(\frac{Col\_X(L\#)}{m}, Y_{uc\_top}\right)$$

$$M_{uc\_top} := Plot\_Col\_M(M1_{uc\_top}, M2_{uc\_top})$$

$$M_{uc\_connect} := Plot\_Col\_M(M1_{uc\_top}, M2_{uc\_bot})$$

$$\text{stack}(M_{uc\_bot}, V_{uc\_top}, M_{uc\_top}, M_{uc\_connect})$$

Program 37: Io plot LOWER Column Moments

$$Plot\_LOW\_Col\_M(M\#, L\#, n\#, H\#, N\#, fdn, SC\#) :=$$

$$X_{uc\_bot} := \left(\frac{N\# \cdot \frac{M\#}{kNm}\, n\#}{\sum\left(\frac{SC\# \cdot L\#}{m}\right)} + \frac{Col\_X(L\#)}{m} \right)$$

$$Y_{uc\_bot} := \text{matrix}(\text{rows}(L\#) + 1, 1)$$

$$M1_{uc\_bot} := \text{augment}\left(\frac{Col\_X(L\#)}{m}, N\# \cdot Y_{uc\_bot}\right)$$

$$M2_{uc\_bot} := \text{augment}(X_{uc\_bot}, N\# \cdot Y_{uc\_bot})$$

$$M_{uc\_bot} := Plot\_Col\_M(M1_{uc\_bot}, M2_{uc\_bot})$$

$$V1_{uc\_bot} := M1_{uc\_bot}$$

$$V2_{uc\_top} := \text{augment}\left(\frac{Col\_X(L\#)}{m}, \frac{N\# \cdot H\#}{m}\right)$$

$$V_{uc\_top} := Plot\_Col\_M(V1_{uc\_bot}, V2_{uc\_top})$$

$$X_{uc\_top} := \begin{cases} \frac{\frac{M\#}{kNm}\, n\#}{\sum\left(\frac{SC\# \cdot L\#}{m}\right)} \cdot (-0.5) \cdot N\# + \frac{Col\_X(L\#)}{m} & \text{if } fdn = 1 \\ \frac{Col\_X(L\#)}{m} & \text{otherwise} \end{cases}$$

$$Y_{uc\_top} := \frac{H\# \cdot N\#}{m}$$

$$M1_{uc\_top} := \text{augment}(X_{uc\_top}, Y_{uc\_top})$$

$$M2_{uc\_top} := \text{augment}\left(\frac{Col\_X(L\#)}{m}, Y_{uc\_top}\right)$$

$$M_{uc\_top} := Plot\_Col\_M(M1_{uc\_top}, M2_{uc\_top})$$

$$M_{uc\_connect} := Plot\_Col\_M(M1_{uc\_top}, M2_{uc\_bot})$$

$$\text{stack}(M_{uc\_bot}, V_{uc\_top}, M_{uc\_top}, M_{uc\_connect})$$

Program 38: Convert Top Col Moments to strings

$$MStr2_{top}(n\#, M\#) := \begin{bmatrix} a\# := \overrightarrow{\frac{\mathrm{col}(M\#, 1)_{n\#}}{kNm}} \\ b\# := \mathrm{var2str}\left(\overrightarrow{\mathrm{round}(a\#)}\right) \end{bmatrix}$$

Program 39: Convert Top Col Moments to strings

$$MStr2_{bot}(n\#, M\#) := \begin{bmatrix} a\# := \overrightarrow{\frac{\mathrm{col}(M\#, 2)_{n\#}}{kNm}} \\ b\# := \mathrm{var2str}\left(\overrightarrow{\mathrm{round}(a\#)}\right) \end{bmatrix}$$

Program 40:To plot labels in column BMD

$$Plot\_Pts(nn3, M1\#, M2\#) := \begin{bmatrix} A\# := Plot\_Labs\_SF(M1\#) \\ B\# := A\# \\ C\# := M2\# \\ \text{for } j \in [1..\mathrm{rows}(B\#)] \\ \quad B\#_{j,3} := C\#_j \\ B\# \end{bmatrix}$$

.