

Structural Analysis Software

- Features
- Algorithm / Technical Information
- Limitations
- Verification Examples

Prepared By:

Muhammad Usman Shamsi (CE-01)

Umair Ahmed Siddiqui (CE-03)

Muqheet Ahmed (CE-06)

Students of M. Engg. (Structure), Batch 2009-10

Acknowledgement

First of all, thanks to Almighty Allah who gave us the knowledge and skills to write this software. Without His help and wish nothing is possible. Secondly to our teachers who conveyed this knowledge to us and build our skills. We especially thank to Dr. Sarosh H. Lodi (Chairman, Civil Dept., NEDUET) and Dr. Asad-ur-Rehman (Co-Chairman, Civil Dept. NEDUET) for their efforts in conveyance of knowledge and concepts. We also thank to all our colleagues for their worthy suggestions, and contribution in testing of software. Finally thanks to the talented people at Microsoft™ who provided the platform for building the software.

Introduction:

Structural Analysis Software, as describes by its name, is a computer software, programmed on Microsoft™ Visual Basic 6.0™, to analyze structural systems. This software is developed during a semester of M.Engg. , while studying the subjects of Advanced Structural Analysis and Finite Element Method. The software is based on Stiffness Matrix Method of Structural Analysis, briefly described in the algorithm section. The name of software may seem to be promising much more than the software is actually capable to perform. But this is the very beginning and further additions shall be incorporated in this small program.

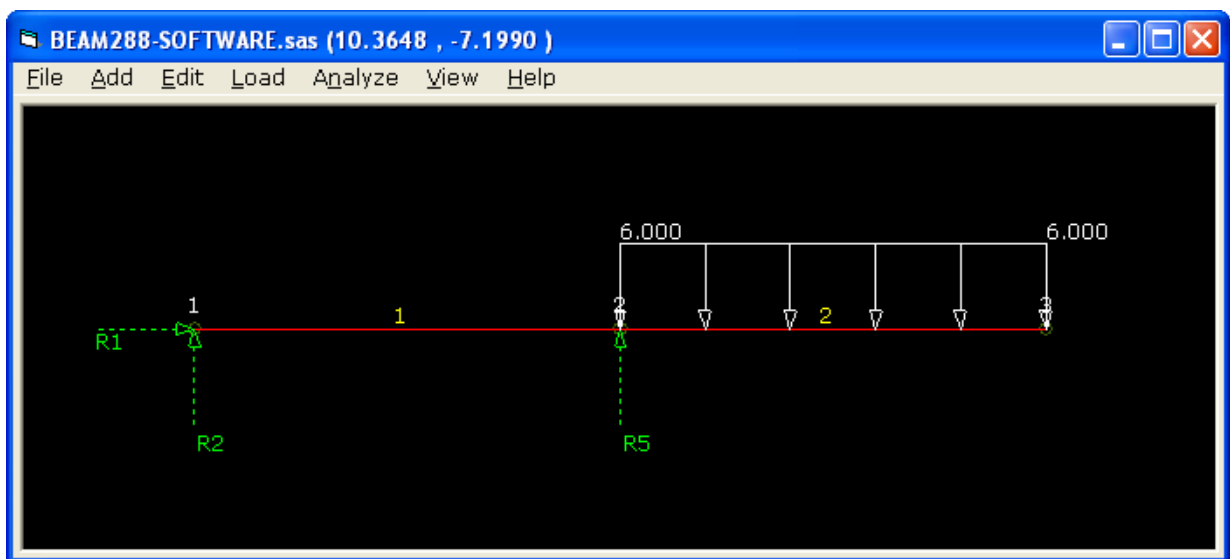
This report consists of four sections; *Features* section describes about the performance capabilities of the software and how it can be used to analyze frame structures. *Algorithm* briefly describes the underlined methodology. Technical details of every feature is not described rather emphasis is given to relatively important sections of algorithm. The software cannot go beyond certain boundaries that are covered in the *Limitations*. At the end some *Verification Examples* are presented to demonstrate the authenticity of software.

Section 1: Features

The software is written to handle 2-dimensional frame structures loaded with point loads and uniformly distributed loads. It can accept as many nodes and frame elements as you require. The software is capable of handling homogenous boundary conditions (displacement is set to be zero in certain degree of freedom). In following section we shall describe the features of the software.

Main Screen

The main screen of the software contains a menu bar and a plot screen. Menus provide the access to different features of the software. Plot screen displays the geometry of structure, applied load, deformed structure, shear force, bending moment and axial force diagrams. The Title bar of the window displays the current filename and coordinates at current mouse location.



File Menu

New: This command initializes a new file and close the one currently opened

Open: Open a previously saved file

Save / Save As: Save the current structure on disk with current / new file name.

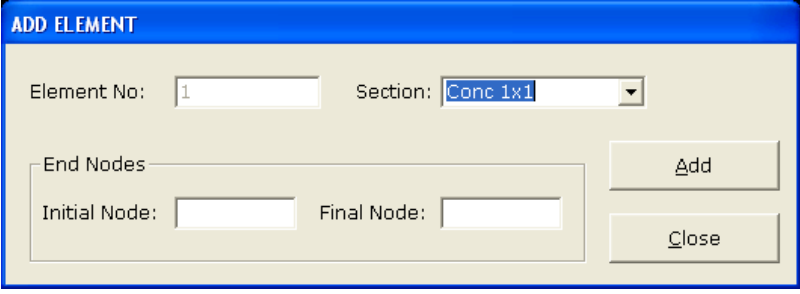
Exit: Quits the program

Add Menu

Add Node: Displays the add node dialog box which can be used to add nodes at desired coordinates.

The 'ADD NODE' dialog box is shown. It has a blue title bar. Inside, there is a 'Node No:' field with the value '1'. Below it is a 'Coordinates' section with three input fields: 'X:' with the value '0', 'Y:' with the value '0', and 'Z:' which is empty. To the right of these fields are two buttons: 'Add' and 'Close'.

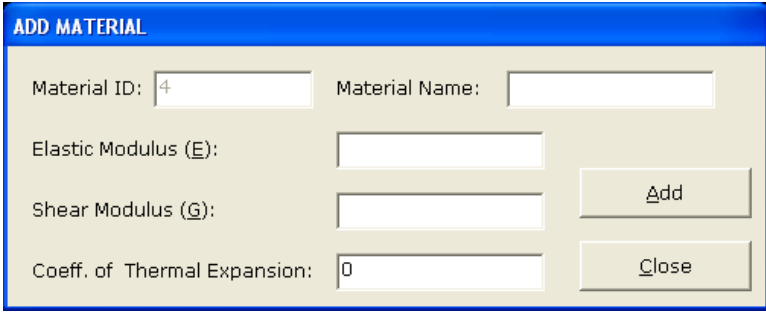
Add Element: Displays the add element dialogue box. The dialogue box can be used to add frame elements between two nodes. A cross section can also be selected for the new element.



The ADD ELEMENT dialog box has a blue title bar. It contains the following fields and controls:

- Element No:
- Section:
- End Nodes section containing:
 - Initial Node:
 - Final Node:
- Buttons: Add and Close

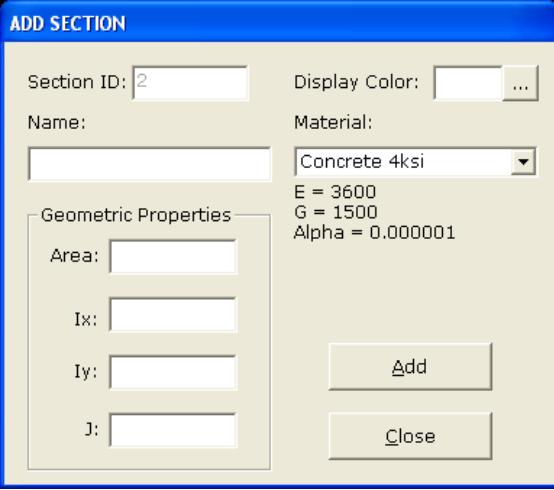
Add Material: shows the add material dialogue box. Here a new material can be defined with required parameters. At current, the values of G and Coefficient of thermal expansion has no function and can be entered as any value.



The ADD MATERIAL dialog box has a blue title bar. It contains the following fields and controls:

- Material ID:
- Material Name:
- Elastic Modulus (E):
- Shear Modulus (G):
- Coeff. of Thermal Expansion:
- Buttons: Add and Close

Add Section: Shows the add section dialogue box. It can be used to define a new section with required properties and display color. At current the the values of Iy and J has no function and can be entered as any value.



The ADD SECTION dialog box has a blue title bar. It contains the following fields and controls:

- Section ID:
- Display Color: ...
- Name:
- Material:
- Geometric Properties section containing:
 - Area:
 - Ix:
 - Iy:
 - J:
- Material Properties:
 - E = 3600
 - G = 1500
 - Alpha = 0.000001
- Buttons: Add and Close

Edit Menu

This menu is similar to Add Menu. The sub menus of Edit menu can be used to modify the nodes, elements, materials, and sections that are defined already.

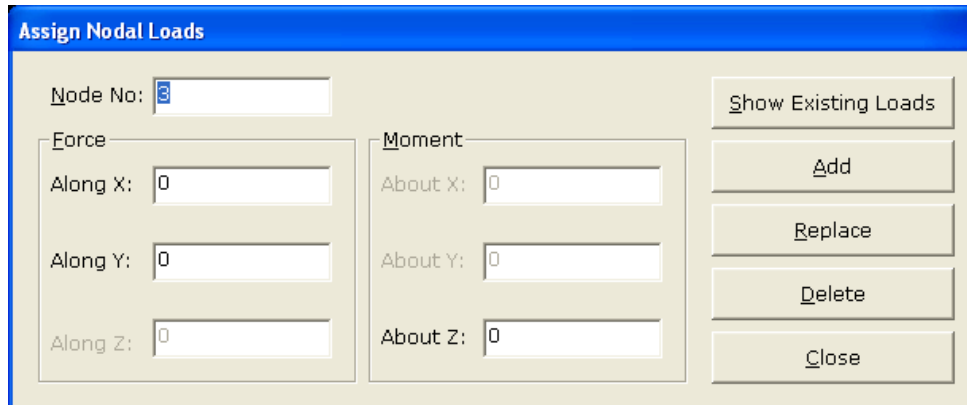
Load Menu

Assign Nodal Load: Provide the access to Assign Nodal Loads form. This form can be used to apply translational and rotational forces on nodes.

- Show Existing button displays the existing nodal load on the selected node number
- Add button adds the typed loads to the desired node
- Replace button replaces the current load of the desired node with the typed in values
- Delete button deletes any existing load from the desired node

Sign Convention:

- Transverse load is positive in the direction of positive global axes
- Moment is positive in the counter-clockwise direction



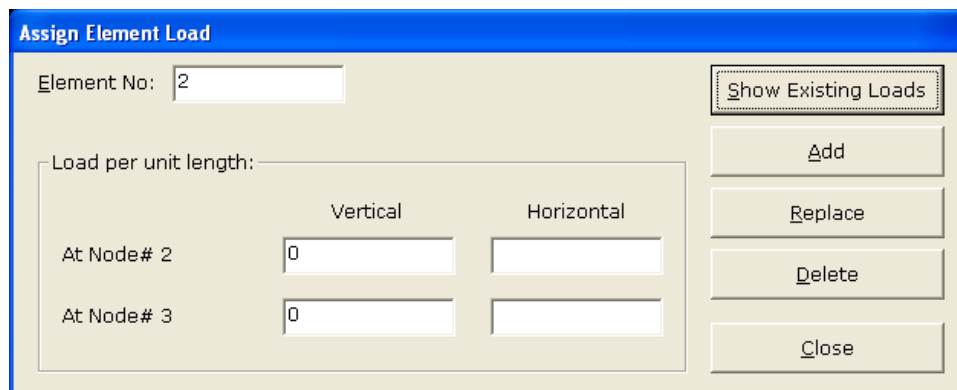
The 'Assign Nodal Loads' dialog box features a blue title bar. It includes a 'Node No:' field with the value '3'. To the right is a 'Show Existing Loads' button. Below the node number are two columns of input fields: 'Force' (Along X, Y, Z) and 'Moment' (About X, Y, Z), each with a value of '0'. On the right side, there are four stacked buttons: 'Add', 'Replace', 'Delete', and 'Close'.

Assign Element Load: It takes to the Add Element Load form. This form can be used to apply uniformly distributed loads in the direction vertical to element's local axis (Vertical) and along the local axis (Horizontal).

- Show Existing button displays the existing nodal load on the selected element number
- Add button adds the typed in loads to the desired element
- Replace button replaces the current load of the desired element with the typed in values
- Delete button deletes any existing load from the desired element

Sign Convention:

- Vertical load is positive in the direction of local vertical axis
- Horizontal load is positive in the direction of local horizontal axis



The 'Assign Element Load' dialog box has a blue title bar. It contains an 'Element No:' field with the value '2' and a 'Show Existing Loads' button. Below this is a section for 'Load per unit length:' with two columns: 'Vertical' and 'Horizontal'. There are two rows of input fields, one for 'At Node# 2' and one for 'At Node# 3', each with values of '0'. On the right, there are four stacked buttons: 'Add', 'Replace', 'Delete', and 'Close'.

Analyze Menu

Run Analysis: Analyze the structure and displays the deformed shape

Discard Analysis: Discard the current analysis results

Write Analysis to File: Writes a text file that contains detailed information about inputs, analysis, and outputs. It is recommended to use a file name other than the file used to save the model because sometimes it replaces the *.sas file with the output content.

View Menu

Deformed Shape: Shows the deformed shape on specified scale

Un-deformed Shape: Show the un-deformed shape

Show Loads: While checked, the loads on the structure are displayed on screen. Uncheck it deformed shape mode to clearly view the outputs

Bending Moment Diagram: Display the Bending Moment Diagram on specified scale.

Sign Convention: Bending moment is plotted on compression face i.e. positive bending moment cause sagging and vice versa.

Shear Force Diagram: Displays the Shear Force Diagram on specified scale.

Sign Convention: Shear force producing clockwise shear deformation is positive.

Axial Force Diagram: Displays the Axial Force Diagram on specified scale.

Sign Convention: Axial force causing compression is plotted as positive

Display Reactions: Displays a message box showing the values of reactions.

Section 2: Algorithm /Technical Information

In this section the basic methodology behind the software is described briefly:

The software uses Stiffness Matrix Method for the analysis of the structure. The basic governing equation of the method is

$$\{f\} = [K] \times \{d\}$$

Where,

$\{f\}$ = force vector in global coordinates,

$\{d\}$ = displacement vector in global coordinates

$[K]$ = Stiffness matrix of the structural system

For further details of the method, refer to any relevant book.

Following are the main methodology used in the software:

Handling of Nodes

The program uses the variable *NoOfNodes* to store the total number of nodes present in the structure. The coordinates of these nodes are stored by *XCoor()* and *YCoor()* arrays. When a node is to be inserted in the system, the program increases the *NoOfNodes* variable by 1, and store the location of new node in *XCoor()* and *YCoor()* arrays.

Every node of the structure is related to 3 General Degrees of Freedom. The following formulae are used to address corresponding degree of freedom at i^{th} node

Degree of Freedom	Address
Horizontal	$3 \times (i - 1) + 1$
Vertical	$3 \times (i - 1) + 2$
Rotational	$3 \times (i - 1) + 3$

Handling of Elements

The program uses the variable *NoOfElements* to store the total number of Elements present in the structure. The end nodes of these Elements are stored by *Endi()* and *Endj()* arrays. When a Element is to be inserted in the system, the program increases the *NoOfElements* variable by 1, and store the location of new Element in *Endi ()* and *Endj()* arrays.

Similar methodology is used in *AddMaterial* and *AddSection* procedures.

The Edit procedures for nodes, elements, material and sections are programmed for modification of existing data.

Handling of Restraints

The program uses *TXRest()*, *TYRest()*, and *RZRest()* arrays to store which degrees of freedom are locked. This information is used for the formation of restrains vector which is in turn used for reduction of stiffnes matrix and force vector. If a certain DOF is locked, the corresponding variables's value is set to 1.

Handling of Nodal Loads

The program stores the values of forces applied on nodes to XForce(), YForce() and ZMom(). This information is used in formation of force vector.

Handling of Element Load

The program stores the distributed force on elements in variables ElemLoadi() and ElemLoadj(). The corresponding end actions on each node of the element are computed during the analysis. These end actions are stored in an array EndActions() and are added to the ForceVector() array.

Solution Procedure

- The program forms elements' local stiffness matrices, transformation matrices and global stiffness matrices by calling corresponding procedures
- Then it calls for the assembling of elements' global stiffness matrix. This is done by considering the element end node numbers, computing the corresponding degrees of freedom by the above stated formulas and adding the value in global matrix in appropriate position. See code list 01.

CODE LIST 01 – ASSEMBLING GLOBAL STIFFNESS MATRIX

```

For i = 1 To NoOfElements
    Node1 = Endi(i): Node2 = Endj(i)

    Call GetElemGlobalStiff(i, temp())

    'FIRST QUADRANT
    '-----
    For m = (3 * Node1 - 3 + 1) To (3 * Node1 - 3 + 3)
        For n = (3 * Node1 - 3 + 1) To (3 * Node1 - 3 + 3)
            SysStiff(m, n) = SysStiff(m, n) + temp(m - (3 * Node1 - 3), n - (3 * Node1 - 3))
        Next n
    Next m

    'SECOND QUADRANT
    '-----
    For m = (3 * Node1 - 3 + 1) To (3 * Node1 - 3 + 3)
        For n = (3 * Node2 - 3 + 1) To (3 * Node2 - 3 + 3)
            SysStiff(m, n) = SysStiff(m, n) + temp(m - (3 * Node1 - 3), n - (3 * Node2 - 3) + 3)
        Next n
    Next m

    'THIRD QUADRANT
    '-----
    For m = (3 * Node2 - 3 + 1) To (3 * Node2 - 3 + 3)
        For n = (3 * Node1 - 3 + 1) To (3 * Node1 - 3 + 3)
            SysStiff(m, n) = SysStiff(m, n) + temp(m - (3 * Node2 - 3) + 3, n - (3 * Node1 - 3))
        Next n
    Next m

    'FOURTH QUADRANT
    '-----
    For m = (3 * Node2 - 3 + 1) To (3 * Node2 - 3 + 3)
        For n = (3 * Node2 - 3 + 1) To (3 * Node2 - 3 + 3)
            SysStiff(m, n) = SysStiff(m, n) + temp(m - (3 * Node2 - 3) + 3, n - (3 * Node2 - 3) + 3)
        Next n
    Next m

Next i

```

- Next, the program forms the force vector.
- Now the stiffness matrix and force vectors are reduced by deleting the rows and columns corresponding to those degrees of freedom which are locked and have a value equal to 1 in Restraints() vector.

- This reduced system of linear equations is passed to EqSolve() subroutine to get the global displacements of structure at unlocked degrees of freedom.
- In the next step the full displacement vector is formed comprising of both locked and unlocked degrees of freedom
- This full displacement vector is multiplied by the original stiffness matrix to get the final forces corresponding to each D.O.F.
- The End Actions corresponding to each distributed load on the structure is then subtracted from this force vector to get the actual forces on the structure.
- Next the internal forces on each element are computed by transforming the deflections on end nodes from global to local axes and multiplying by local stiffness matrix. See code list 02.

CODE LIST 02 – CALCULATION OF MEMBERS' INTERNAL FORCES

```

Sub SubCalcInternalForces()

Dim i As Long, j As Long
Dim tempGF() As Double, tempGDis() As Double, tempGSTiff() As Double
Dim TempLF() As Double, tempLDis() As Double, tempLStiff() As Double
Dim tempTran() As Double
Dim UDL1 As Double, UDL2 As Double, UDAL1 As Double, UDAL2 As Double
Dim Length As Double
Dim A1 As Double, A2 As Double
Dim V1 As Double, V2 As Double, M1 As Double, M2 As Double, Total As Double

For i = 1 To NoOfElements

    Length = GetElemLength(i)
    Call GetDisp(i, tempGDis())
    Call GetTrans(i, tempTran())
    Call MatMult(tempTran(), tempGDis(), tempLDis())
    Call GetElemLocalStiff(i, tempLStiff())
    Call MatMult(tempLStiff(), tempLDis(), TempLF())

    'ADJUST FORCES FOR ELEMENT END ACTIONS
    '-----
    UDL1 = ElemLoadi(i):    UDL2 = ElemLoadj(i)
    UDAL1 = ElemALoadi(i): UDAL2 = ElemALoadj(i)

    Total = (UDL1 + UDL2) / 2 * Length

    A1 = UDAL2 * Length / 6 + UDAL1 * Length / 3
    A2 = UDAL1 * Length / 6 + UDAL2 * Length / 3
    V1 = UDL2 * Length / 6 + UDL1 * Length / 3
    V2 = UDL1 * Length / 6 + UDL2 * Length / 3
    M1 = UDL1 * Length ^ 2 / 20 + UDL2 * Length ^ 2 / 30
    M2 = -(UDL2 * Length ^ 2 / 20 + UDL1 * Length ^ 2 / 30)

    TempLF(1, 1) = TempLF(1, 1) - A1
    TempLF(4, 1) = TempLF(4, 1) - A2
    TempLF(2, 1) = TempLF(2, 1) - V1
    TempLF(5, 1) = TempLF(5, 1) - V2
    TempLF(3, 1) = TempLF(3, 1) - M1
    TempLF(6, 1) = TempLF(6, 1) - M2

Next i

```

- The member forces are then interpolated at intermediate locations of member by realizing the fact that:
 - o The difference of shear force between two points is equal to the sum of loading between those points
 - o Difference of moment is equal to sum of shear force
 - o Difference of slope is equal to sum of Bending moment divided by EI.
 - o Difference of deflection is equal to sum of slope.

Section 3: *Limitations*

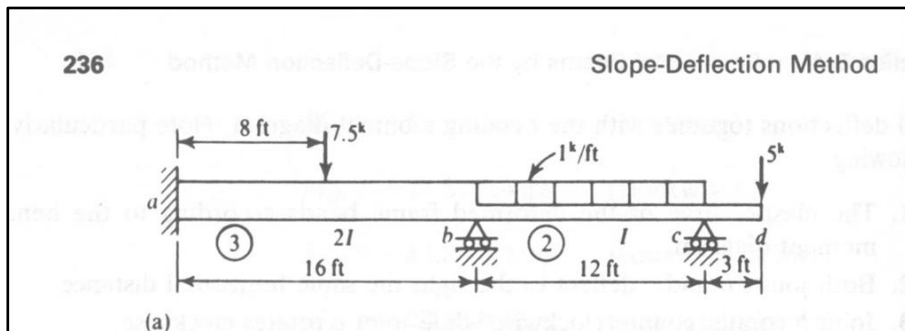
At present, the software has following limitations:

- 1- It can solve 2D frames only. It cannot handle 3D frames
- 2- Only Frame (Axial + Bending) elements are handled.
- 3- Support displacements cannot be handled
- 4- Internal hinges cannot be introduced
- 5- The intermediate output stations for any element are 9 with an interval of $L/10$. Exact value of the response can be read at these points only. For other locations it must be interpolated.

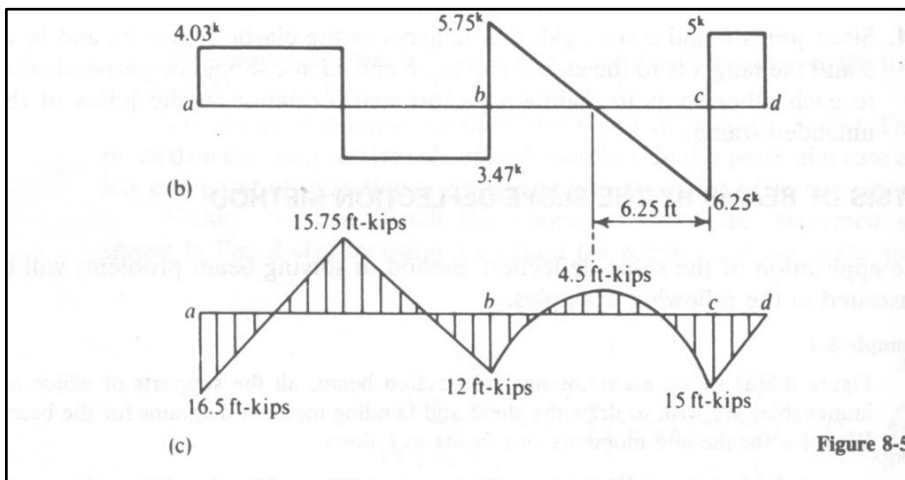
Section 4: Verification Examples

1- Shear Force and Bending Moment Diagram of Indeterminate Beam:

a) Slope Deflection Method

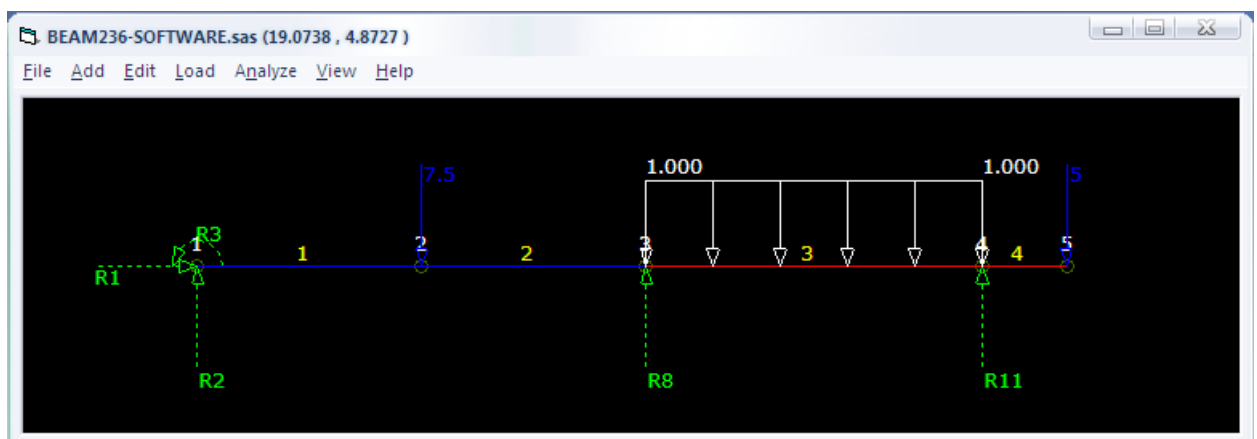


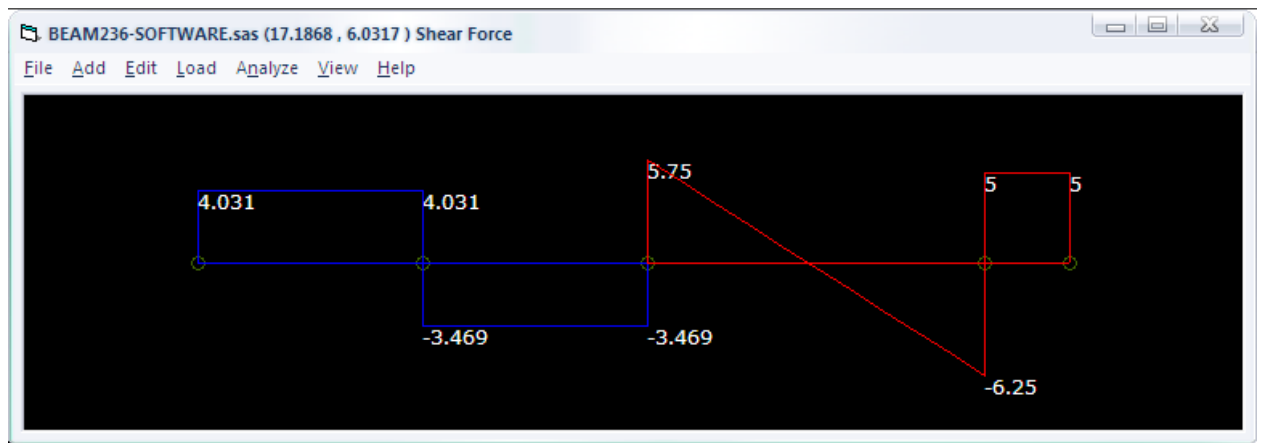
Problem # 8-1
ELEMENTARY
THEORY OF
STRUCTURES
(Fourth Edition)
Yuan-Yu Hsieh & S.T.
Mau



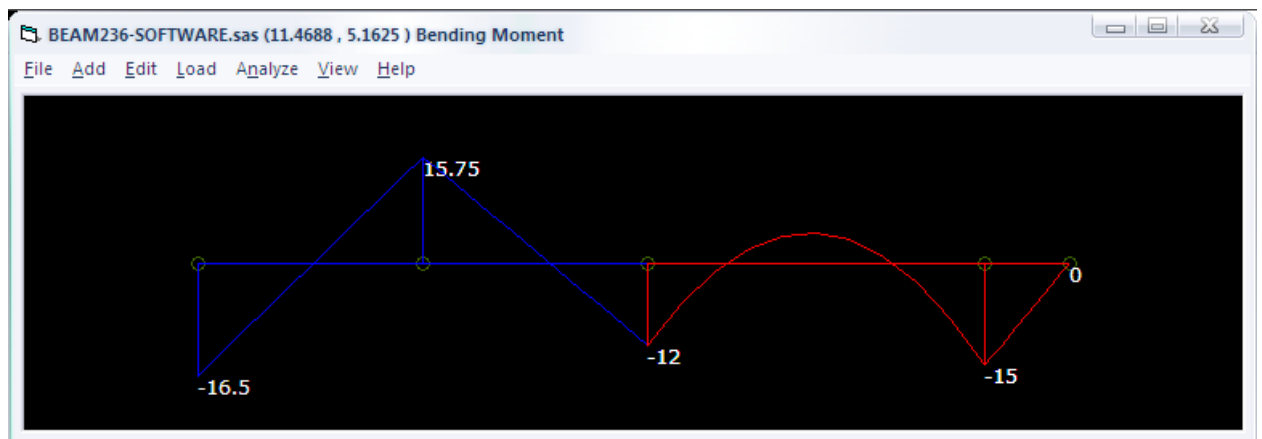
Shear Force and
Bending Moment
Diagram

b) Stiffness Method Using SAS (kip-ft)





(Shear Force Diagram)

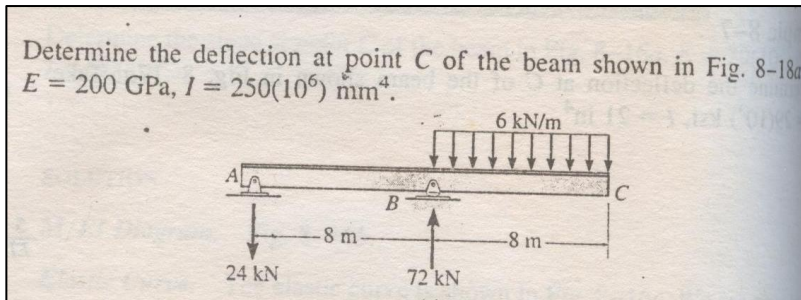


(Bending Moment Diagram)

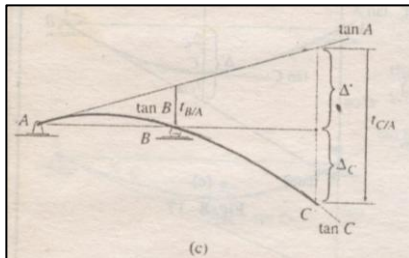
2- Deflection of Beam:

a) Moment

Area Method(kN-m)

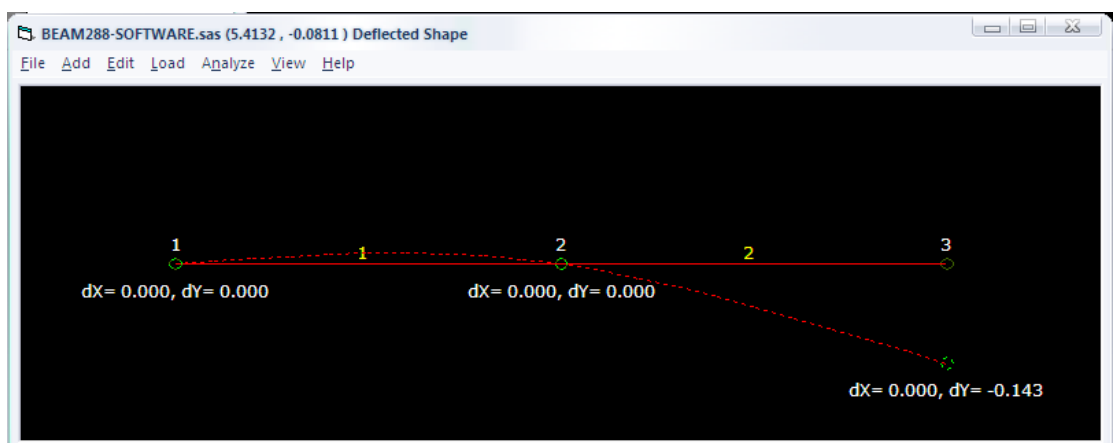
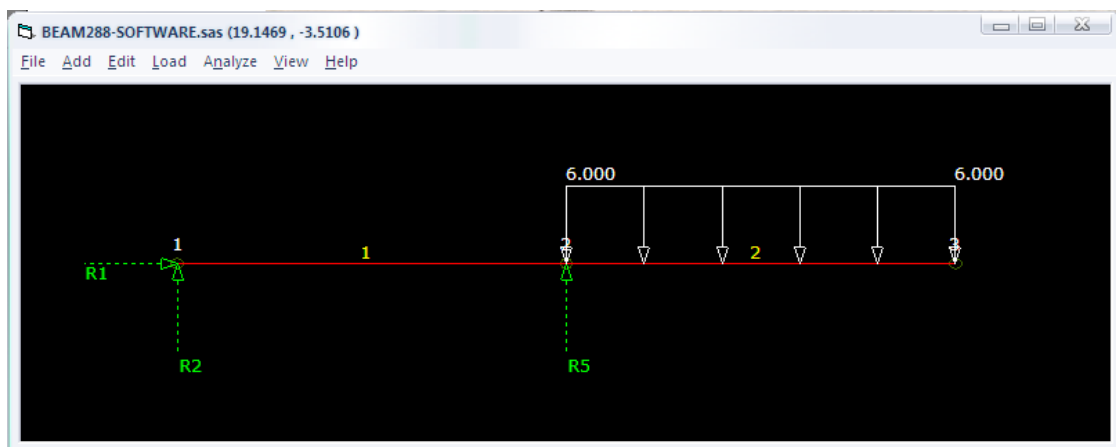


Problem # 8-8
 STRUCTURAL ANALYSIS
 (Fourth Edition)
 R.C. HIBBELER



$$\Delta_C = \frac{-7168 \text{ kN}\cdot\text{m}^3}{[200(10^6) \text{ kN/m}^2][250(10^6)(10^{-12}) \text{ m}^4]} = -0.143 \text{ m}$$

Deflection at free end 'C' by Moment Area Method



b) Stiffness Method using SAS(kN-m)

(Deflected shape)