22

FEM Programs for Plane Trusses and Frames

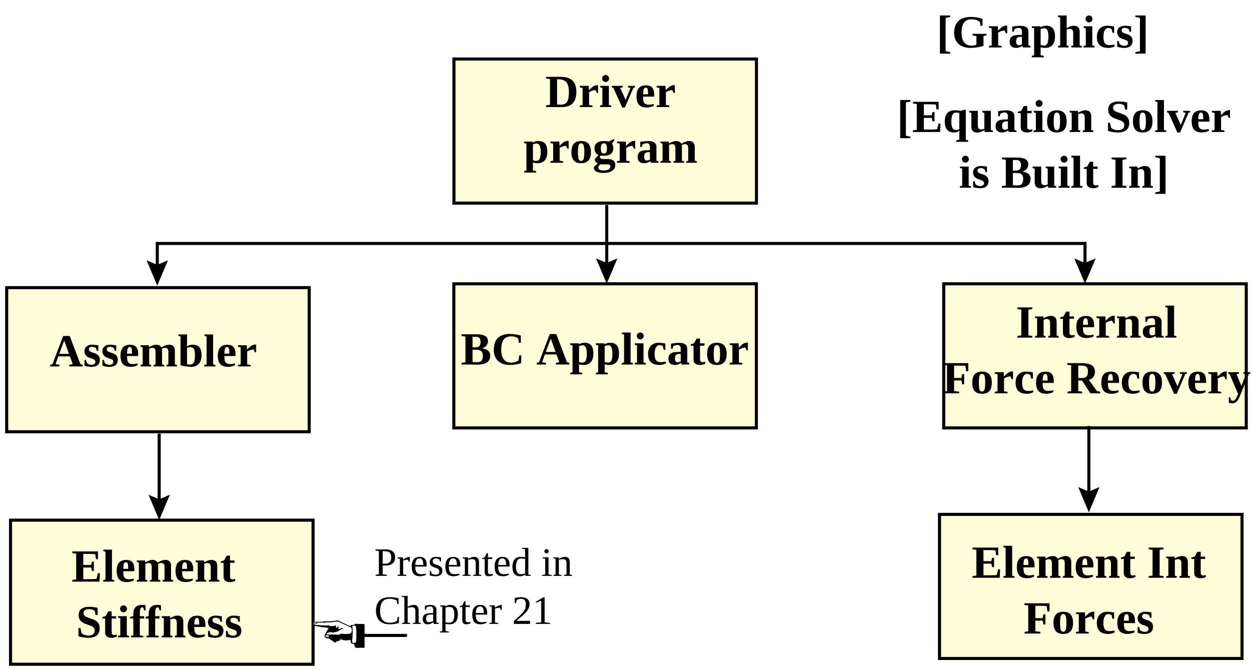
The Three Basic Stages of a FEM Program Based on the Direct Stiffness Method

Preprocessing **: defining the FEM model**

Processing **: setting up the stiffness equations and solving for displacements**

Postprocessing **: recovery of derived quantities and graphical presentation of results**

Plane Truss Demo Program



Assembler for Plane Truss Program

```
PlaneTrussMasterStiffness[nodcoor_,elenod_,
  elemat_,elefab_,eleopt_]:=Module[
  {numele=Length[elenod],numnod=Length[nodcoor],
  e,eNL,eftab,ni,nj,i,j,ncoor,mprop,fprop,opt,Ke,K},
  K=Table[0,{2*numnod},{2*numnod}];
  For [e=1, e<=numele, e++,
     eNL=elenod[[e]]; {ni,nj}=eNL;
     eftab={2*ni-1,2*ni,2*nj-1,2*nj};
     ncoor={nodcoor[[ni]],nodcoor[[nj]]};
     mprop=elemat[[e]]; fprop=elefab[[e]]; opt=eleopt;
     Ke=PlaneBar2Stiffness[ncoor,mprop,fprop,opt];
     neldof=Length[Ke];
     For [i=1, i<=neldof, i++, ii=eftab[[i]];
         For [j=i, j<=neldof, j++, jj=eftab[[j]];
             K[[jj,ii]]=K[[ii,jj]]+=Ke[[i,j]]
             ];
         ];
      ]; Return[K];
  ];
```

Assembler for Plane Truss Program (cont'd)

```
PlaneBar2Stiffness[ncoor_,mprop_,fprop_,opt_]:= Module[
 {x1,x2,y1,y2,x21,y21,Em,Gm,rho,alpha,A,numer,L,LL,LLL,Ke},
  {{x1,y1},{x2,y2}}=ncoor; {x21,y21}={x2-x1,y2-y1};
  {Em,Gm,rho,alpha}=mprop; {A}=fprop; {numer}=opt;
  If [numer,{x21,y21,Em,A}=N[{x21,y21,Em,A}]];
  LL=x21^2+y21^2; L=PowerExpand[Sqrt[LL]]; LLL=Simplify[LL*L];
  Ke=(Em*A/LLL)*{{ x21*x21, x21*y21,-x21*x21,-x21*y21},
                 { y21*x21, y21*y21,-y21*x21,-y21*y21},
                 {-x21*x21,-x21*y21, x21*x21, x21*y21},
                 {-y21*x21,-y21*y21, y21*x21, y21*y21}};
  Return[Ke]
];
nodcoor={{0,0},{10,0},{10,10}};
elenod= {{1,2},{2,3},{1,3}};
elemat= Table[{100,0,0,0},{3}];
elefab= {{1},{1/2},{2*Sqrt[2]}};
eleopt= {True};
K=PlaneTrussMasterStiffness[nodcoor,elenod,
                            elemat,elefab,eleopt];
Print["Master Stiffness of Example Truss:"];
Print[K//MatrixForm];
```

Displacement BC Applicator

```
ModifiedMasterStiffness[pdof_,K_] := Module[
  {i,j,k,n=Length[K],np=Length[pdof],Kmod}, Kmod=K;
  For [k=1,k<=np,k++, i=pdof[[k]];
      For [j=1,j<=n,j++, Kmod[[i,j]]=Kmod[[j,i]]=0];
      Kmod[[i,i]]=1
      ];
  Return[Kmod]
];
ModifiedNodeForces[pdof_,f_] := Module[
  {i,k,np=Length[pdof],fmod}, fmod=f;
     For [k=1,k<=np,k++, i=pdof[[k]]; fmod[[i]]=0];
     Return[fmod]
];
K=Array[Kij,{6,6}];
Print["Assembled Master Stiffness:"];Print[K//MatrixForm];
K=ModifiedMasterStiffness[{1,2,4},K];
Print["Master Stiffness Modified For Displacement B.C.:"];
Print[K//MatrixForm];
f=Array[fi,{6}];
Print["Node Force Vector:"]; Print[f];
f=ModifiedNodeForces[{1,2,4},f];
Print["Node Force Vector Modified For Displacement B.C.:"];
Print[f];
```

Restriction: prescribed displacements must be zero.
Else the code for **ModifiedNodeForces** is more complicated.

Plane Truss Internal Force Recovery

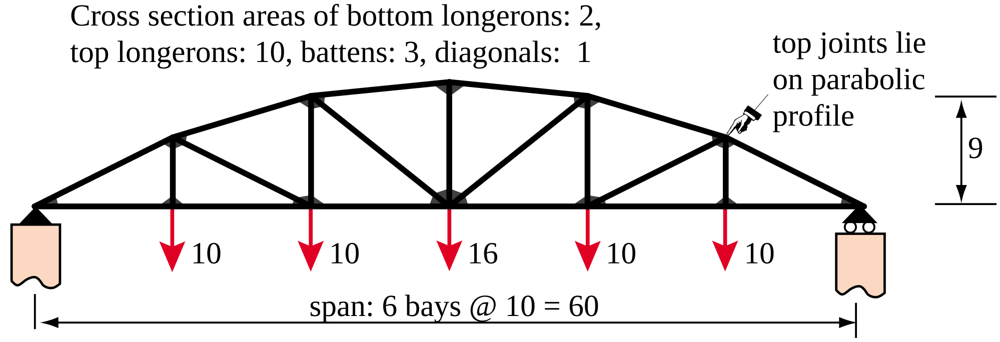
```
PlaneTrussIntForces[nodcoor_,elenod_,elemat_,elefab_,
  eleopt_,u_]:= Module[{numele=Length[elenod],
  numnod=Length[nodcoor],e,eNL,eftab,ni,nj,i,
  ncoor,mprop,fprop,opt,ue,p},
  p=Table[0,{numele}]; ue=Table[0,{4}];
  For [e=1, e<=numele, e++,
      eNL=elenod[[e]]; {ni,nj}=eNL;
      eftab={2*ni-1,2*ni,2*nj-1,2*nj};
      ncoor={nodcoor[[ni]],nodcoor[[nj]]};
      mprop=elemat[[e]]; fprop=elefab[[e]]; opt=eleopt;
      For [i=1,i<=4,i++, ii=eftab[[i]]; ue[[i]]=u[[ii]]];
      p[[e]]=PlaneBar2IntForce[ncoor,mprop,fprop,opt,ue]
      ];
  Return[p]
];
```

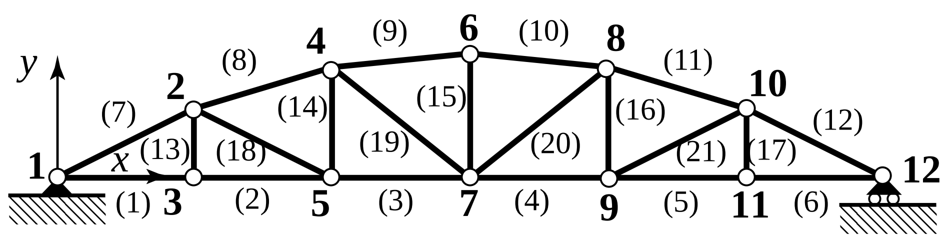
Plane Truss Internal Force Recovery (Cont'd)

```
PlaneBar2IntForce[ncoor_,mprop_,fprop_,opt_,ue_]:= Module[
  {x1,x2,y1,y2,x21,y21,Em,Gm,rho,alpha,A,numer,LL,pe},
  {{x1,y1},{x2,y2}}=ncoor; {x21,y21}={x2-x1,y2-y1};
  {Em,Gm,rho,alpha}=mprop; {A}=fprop; {numer}=opt;
  (*If [numer,{x21,y21,Em,A}=N[{x21,y21,Em,A}]];*)
  LL=x21^2+y21^2;
  pe=Em*A*(x21*(ue[[3]]-ue[[1]])+y21*(ue[[4]]-ue[[2]]))/LL;
  Return[pe]
];
nodcoor={{0,0},{10,0},{10,10}}; elenod= {{1,2},{2,3},{1,3}};
elemat= Table[{100,0,0,0},{3}]; elefab= {{1},{1/2},{2*Sqrt[2]}};
eleopt= {True}; u={0,0,0,0,0.4,-0.2};
p=PlaneTrussIntForces[nodcoor,elenod,elemat,elefab,eleopt,u];
Print["Int Forces of Example Truss:"];
Print[p];
```

Six-bay Bridge Truss Example

Elastic modulus E = 1000
Cross section areas of bottom longerons: 2,
top longerons: 10, battens: 3, diagonals: 1



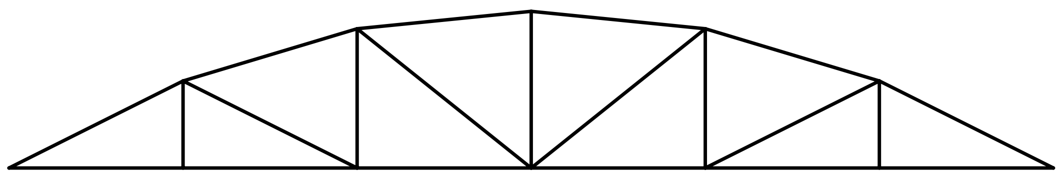


Driver Program: Node & Element Definition

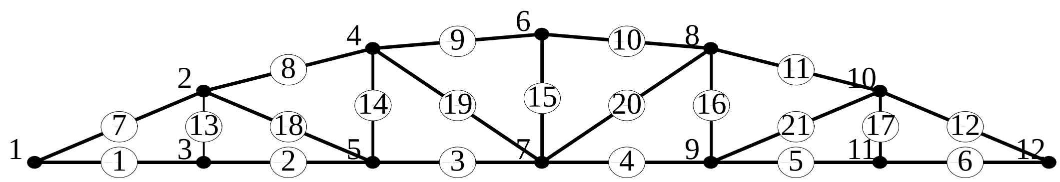
```
ClearAll[];
NodeCoordinates={{0,0},{10,5},{10,0},{20,8},{20,0},{30,9},
         {30,0},{40,8},{40,0},{50,5},{50,0},{60,0}};
ElemNodeLists= {{1,3},{3,5},{5,7},{7,9},{9,11},{11,12},
         {1,2},{2,4},{4,6},{6,8},{8,10},{10,12},
         {2,3},{4,5},{6,7},{8,9},{10,11},
         {2,5},{4,7},{7,8},{9,10}};
numnod=Length[NodeCoordinates];
numele=Length[ElemNodeLists]; numdof=2*numnod;
ElemMaterial= Table[{1000,0,0,0},{numele}];
Abot=2; Atop=10; Abat=3; Adia=1;
ElemFabrication=Join[Table[{Abot},{6}],Table[{Atop},{6}],
    Table[{Abat},{5}],Table[{Adia},{4}]];
ProcessOptions= {True}; aspect=0;
PlotLineElements[NodeCoordinates,ElemNodeLists,aspect,
   "test mesh"];
PlotLineElementsAndNodes[NodeCoordinates,ElemNodeLists,aspect,
   "test mesh with elem & node labels",{True,0.12},{True,0.05}];
```

Driver Program: Model Plot Output

test mesh



test mesh with elem & node labels



Driver Program: BC Definition

```
FreedomTag=FreedomValue=Table[{0,0},{numnod}];
FreedomValue[[3]]={0,-10}; FreedomValue[[5]]={0,-10};
FreedomValue[[7]]={0,-16};
FreedomValue[[9]]={0,-10}; FreedomValue[[11]]={0,-10};
Print["Applied node forces="]; Print[FreedomValue];
FreedomTag[[1]]= {1,1};     (* fixed node 1 *)
FreedomTag[[numnod]]={0,1}; (* hroller @ node 12 *)
```

Driver Program: Processing

```
f=Flatten[FreedomValue];
K=PlaneTrussMasterStiffness[NodeCoordinates,
  ElemNodeLists,ElemMaterial,ElemFabrication,ProcessOptions];
pdof={}; For[n=1,n<=numnod,n++, For[j=1,j<=2,j++,
            If[FreedomTag[[n,j]]>0, AppendTo[pdof,2*(n-1)+j]]]];
Kmod=ModifiedMasterStiffness[pdof,K];
fmod=ModifiedNodeForces [pdof,f];
u=LinearSolve[Kmod,fmod];  u=Chop[u];
Print["Computed Nodal Displacements:"]; Print[u];
```

Driver Program: Post-Processing

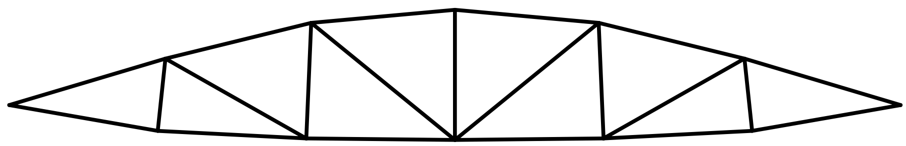
```
f=Simplify[K.u]; f=Chop[f];
Print["External Node Forces Including Reactions:"]; Print[f];
p=PlaneTrussIntForces[NodeCoordinates,ElemNodeLists,
    ElemMaterial,ElemFabrication,eleopt,u]; p=Chop[p];
sigma=Table[p[[i]]/ElemFabrication[[i,1]],{i,1,numele}];
Print["Internal Member Forces:"]; Print[p];
PlotTrussDeformedShape[NodeCoordinates,ElemNodeLists,u,
    1.0,aspect,"Deformed shape"];
PlotAxialStressLevel[NodeCoordinates,ElemNodeLists,sigma,
    1.0,aspect,"Axial stress level"];
```

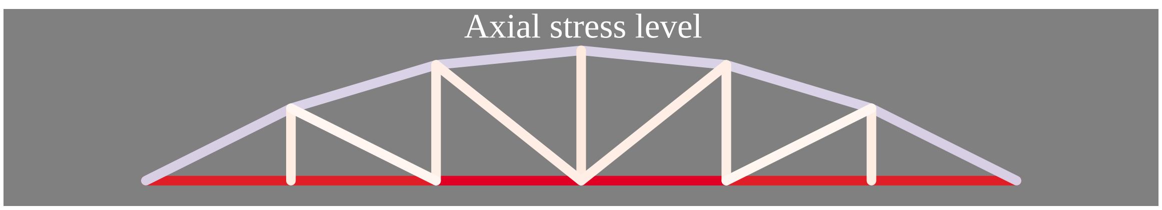
Driver Program: Printed Output

```
Applied node forces:
{{0, 0}, {0, 0}, {0, -10}, {0, 0}, {0, -10}, {0, 0},
 {0, -16}, {0, 0}, {0, -10}, {0, 0}, {0, -10}, {0, 0}}

Computed Nodal Displacements:
{0, 0, 0.809536, -1.7756, 0.28, -1.79226, 0.899001,
 -2.29193, 0.56, -2.3166, 0.8475, -2.38594,
 0.8475, -2.42194, 0.795999, -2.29193, 1.135, -2.3166,
 0.885464, -1.7756, 1.415, -1.79226, 1.695, 0}

External Node Forces Including Reactions:
{0, 28., 0, 0, 0, -10., 0, 0, 0, -10., 0, 0, 0,
 -16., 0, 0, 0, -10., 0, 0, 0, -10., 0, 28.}

Internal Member Forces:
{56., 56., 57.5, 57.5, 56., 56., -62.6099,
 -60.0318, -60.2993, -60.2993,
 -60.0318, -62.6099, 10., 9.25, 12., 9.25,
 10., 1.67705, 3.20156, 3.20156, 1.67705}
```

Driver Program: Result Plot Output

Deformed shape





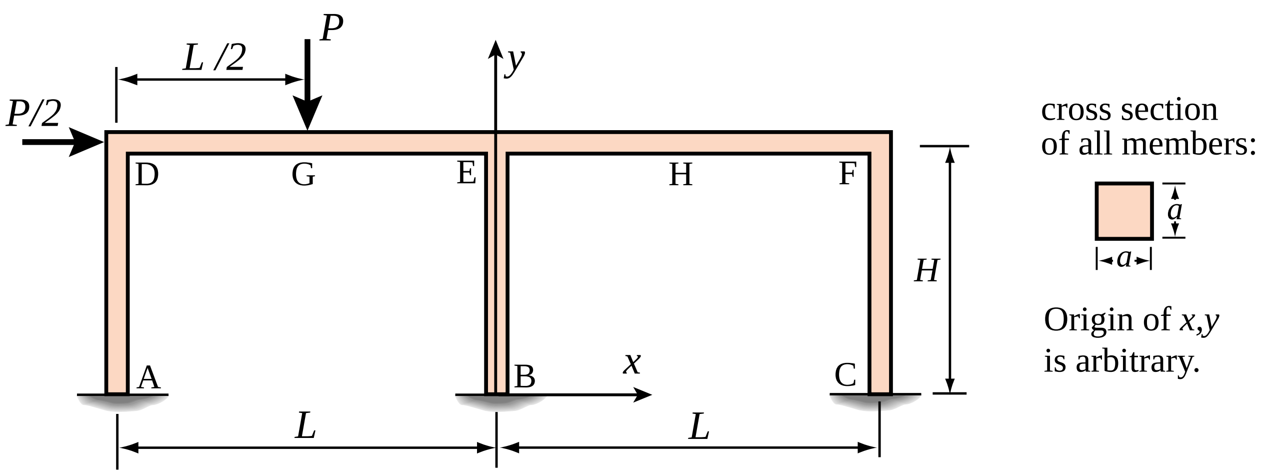
Homework Assignment - Ex 22.2

Complete a plane frame assembler module

Complete an internal force module for beam-column

Test completed code with statements provided in posted Notebook

Homework Assignment - Ex 22.3 Complete a Plane Frame Program to Analyze this Portal Frame Structure:



Plane Frame FEM Discretization for Homework Ex 22.3: