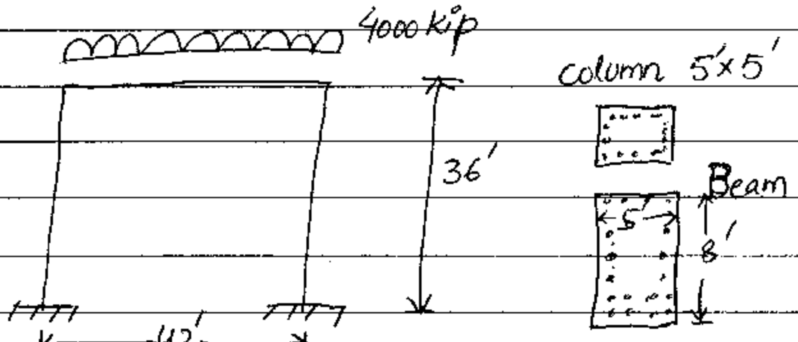
Opensees : Open System for Earthquake Engineering Simulation.

- The framework is a set of Inter-related classes such as domains (data structures), models, elements, solution algorithms, integraters, equation solvers and databases.
- The modeling approach is very flexible.
- Wide range of solution procedures and algorithms.
- Fully programable scripting language

- Three ways of execution of Opensees/Tcl commands.
  - Input directly at prompt
  - Execute Input file at OpenSees prompt (Source Inputfile.tcl)
  - " " " " MS-DOS prompt

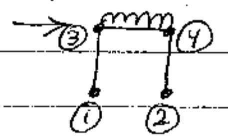
Example :-



- Columns will be modeled as "elastic". (Can be replaced by more refined element models)

- Three differend load cases.

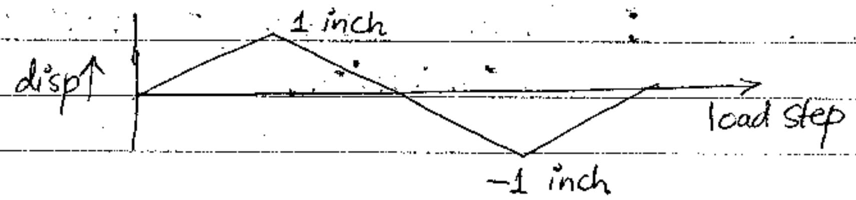
1) Displacement controlled lateral pushover (Static analysis)



push until a desired displacement.

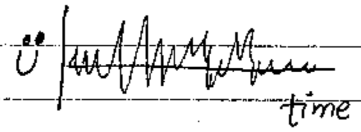
2) Displacement controlled reversed cyclic lateral loading (Static analysis).

A lateral load is applied at nodes 3 and 4 such that a predefined displacement history is achieved at node 3.



3) Dynamic groundmotion input transient analysis.

A uniform $\ddot{U}$ history is imposed at all nodes constrained in a horizontal x-direction (1, and 2)



Model Builder :-

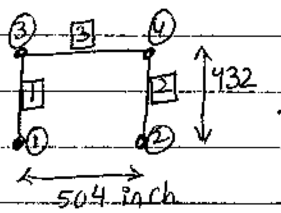
- defining model builder expands the Tcl command library to include OpenSees specific commands such as node and element definition.

ndm (no of dimensions)
ndf (no of dof per node)

model BasicBuilder -ndm 2 -ndf 3

2D

Nodes :-



node tag coordinates.

node 1 0 0
node 2 504 0
node 3 0 432
node 4 504 432

[for 3D
node - - - -
node tag (x, y, z)]

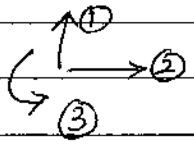
Boundary Conditions :-

dofs

fix 1 1 1 1
fix 2 1 1 1
fix 3 0 0 0
fix 4 0 0 0

0 = Un constrained
1 = Constrained.

In 3D,
eg fix 2 ~~0~~1 0 0 1 0
↓means
A node 2, axial elongation and torsion is restrained.



(dof numbering)

mass values corresponding to each nodal dof.

Mass :- mass 3 5.18 0 0
mass 4 5.18 0 0

can define
or together also,
eg node 3 0 432 -mass 3 5.18 0 0
OR fix 3 0 0 0 -mass 3 5.18 0 0

example mass = $\dfrac{4000/2 \text{ kip}}{32 \frac{ft}{sec^2} \times 12 \text{ in/ft}}$

Elements :- we will use here "Elastic beam Column Element"
for 2D, element elasticBeamColumn- - - - - - -
Tag Start node End node A E I Identifier for previously defined Coordinate transformation object.

But first,

no space

geomTransf Linear 1

used to construct a coordinate transformation object

- It transforms beam element stiffness and resisting force from the basic system to the global coordinate system.
- Three types.

a) Linear Transformation :

Performs a linear geometric transformation of beam stiffness and resisting force from basic $\rightarrow$ global

for 2D,

Identifier Tag

dxi dyi dxj dyj

geomTransf Linear ___ -jntOffset ___ ___ ___ ___

for 3D,

tag

$d_{x,y,z}$

geomTransf Linear ___ ___ ___ ___ -jntOffset ___ ___ ___ ___ ___

vecxzX vecxzY vecxzZ

dx_i, dy_i, dz_i = Joint offset values
Obsolute offsets specified w-r-t global for element-end node i.
Same for $dx_j \xrightarrow{etc}$ node j.

vecxzX, vecxzY, vecxzZ = X, Y, Z components of vector vecxz $\rightarrow$ The vector used to define the local x-z plane of local.

(opensees.berkeley.edu/opensees/manuals/usermanual/241.htm) local y axis = (vecxz) X (x-axis)
cross

These components are specified in the global coordinate system X, Y, Z and define a vector that is in a plane parallel to x-z plane of a local coordinate system.

b) P-Delta Transformation :-

- Performs a linear geometric transformation of K and resisting force from basic $\rightarrow$ global Considering 2nd Order PΔ effects.

for 2D, geomTransf PDelta — (Tag) -jntOffset ----

for 3D, geomTransf PDelta — — — — -jntOffset ---- (same)

3) Corotational Transformation:-

- performs an exact geometric transformation

geomTransf Corotational — -jntOffset ----

(Same as above 2)

For 3D element,

element elasticBeamColumn — — — — — — — — — —

For 2D → TAG node i node j A E I_z Tag Transform [Tag, start node, end node, A, E, G, J, I_y, I_z, Tag for Transform]

element elasticBeamColumn 1 1 3 3600 4227 108000 1

element elasticBeamColumn 2 2 4 3600 4227 108000 1

element elasticBeamColumn 3 3 4 5760 4227 4423680 1

	Columns	Beam
Area	(5x12)(5x12)	(5x12)(8x12)
I_z	$\frac{(5\times12)(5\times12)^3}{12}$	$\frac{(5\times12)(8\times12)^3}{12}$

Recorders :-

used to define the analysis output

a) recorder Node -file (file name) .out -time -node — — .... (Tags node numbers) -nodeRange — — (start node, end)

-region — (tag) -node all -dof — — ..... (Tags dof numbers) — (response type)

b) recorder Element -file .out -time -ele — — .... (Tags) -eleRange — —

-region — -ele all — — ..... (eg force)

In example,

Recorder Node -file Node3.out -time -dof 1 2 disp

Recorder Element -file Element1.out -time -ele 1 force

Before "Recorders" → loads should be defined.

in global coordinates.

→ recorder Element -file ele1global.out -time -ele 1 globalForce

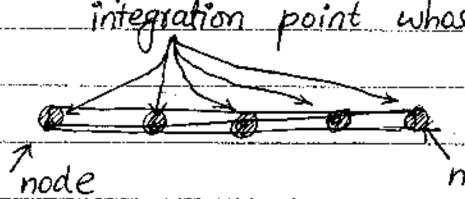
→ " " " ele1local.out " " " localForce

element resisting force in local coordinates

Sections:-

section ___

integration point whose data is to be output.



various responses eg,

a) recorder Element -file www.out -time -ele 1 Section 1 force

b) " " " " " " " 1 Section 1 deformation

c) " " " " " " " " Stiffness

d) " " " " section 1 fiber — — —

"stressStrain"

records σε response

local (x,y) coordinates of fiber to be monitored

previously defined material tag.

Output format :- typically dependent on element type.

general order,

Element	globalForce	2D, 3 dof	FX FY MZ
		3D, 6 dof	FX FY FZ MX MY MZ
	LocalForce	2D, 3 dof	F_x F_y M_z
		3D, 6 dof	F_x F_y F_z M_x M_y M_z
Section	force	Fx	Mx
	deformation	axial strain	Curvature
	stressStrain	Stress	strain

flexibility formulation

considers plasticity to be concentrated over specified hinge lengths.

$\boxed{\text{Beam-Column Elements}}$

Elastic Beam column element

NonLinear beam column element
- ~~based on non-iterative~~ or iterative force formulation
- Considers the spread of plasticity along the element.

displacement based beam Column element → distributed plasticity

Beam with hinges element

⇒ element nonlinear ~~disp~~BeamColumn _ _ _ _ _ _ -mass _ -iter _ _

element tag, start node, end node, no of integration points, section tag (previously defined), Transformation tag, mass density, max no of iterations, Tolerance

⇒ element dispBeamColumn ---- same as above ---- -mass —

⇒ element beamWithHinges _ _ _ _ _ _ _ _ _ _ _ _ -mass _ -iter _ _

element tag, inode, jnode, Section tag previously defined Corresponding to node "i", hinge Length at node "i", section tag "j", hinge length "j", E, A, Iz, Transform tag, density

for 3D, add here _ _ _ (Iy, G, J)

SECTIONS :-

for FD or $\sigma\epsilon$ relationships at beam-column and plate sample points.

Section Elastic _ E A Iz Iy G J

(_ → give any tag)

Section Uniaxial _ _ _

(tag, uniaxial material Tag, F-D quantity to be modeled by this section object.)

uses a previously defined uniaxial material to represent a single section F-D response.

P	axial F-D
MZ	Mϕ about Z
Vy	Shear F-D along y
My	Mϕ about y
Vz	Shear FD → z
T	Torsion F-D

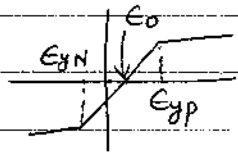
Uniaxial Materials:-

tag, E, damping tangent (optional, default "0")

→ uniaxialMaterial Elastic ___ ___ ___

→ " ElasticPP ___ ___ ___ ___ ___

ϵ_{yP} ϵ_{yN} ϵ_{o}



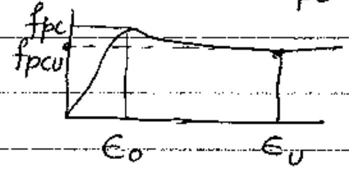
tag, E

→ " ~~Elastic~~ ENT ___ ___

elastic no tension.

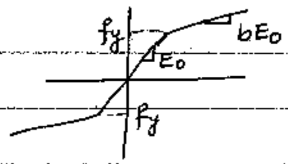
→ // Concrete01 ___ ___ ___ ___ ___

tag f_{pc} ϵ_0 ~~f_{pcu}~~ f_{pcu} ϵ_U



→ " Steel01 ___ ___ ___ ___

tag f_y E_o b



Elements :-

tag, i node, j node, A, previously defined material tag

→ element truss ___ ___ ___ ___ ___

or // // ___ ___ ___ ___ previously defined section tag.

→ elasticBeamColumn

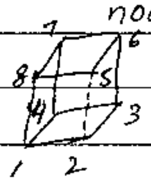
→ nonlinearBeamColumn ─┬→ ~~force based~~
- Distributed plasticity (NonlinearBeamColumn)
- Concentrated plasticity with elastic interior (beamWithHinges)

└→ Displacement Based (dispBeamColumn)

→ Flexure-Shear interaction displacement - based beam-column element. (by Leo massone) 2006.

→ element quad — — — — — — — — (tag, i, j, k, l, element thickness, behavior type: PlaneStrain or PlaneStress depend on nD material, previously defined nD material tag.)

→ // stdBrick — — — — — — — — — — (tag, node nos, previously defined nD material tag)



→ // beamColumnJoint — — — — -----

nD Materials :-

represents σ-ε relationships at the integration points of continuum and F-D elements.

nDMaterial ElasticIsotropic — — —
(tag, E, ν (poisson's ratio))

Loads and Analysis :-

Applying loads is a 3-step process.

1) define load pattern
2) // Analysis
3) Apply loads when you execute analysis.

1) Load definition :-

a) Plain pattern, for
   (i) Noda loads, gravity and lateral loads or (Load controlled nodal displacements.)
   (ii) Single point constraints (such as disp control at a node) typically used for constant δ ~~at a node~~
   (iii) Element loads, UDL

b) Uniform Excitation pattern $\ddot{U}_g$ to all fix nodes

c) MultipleSupport pattern $\ddot{U}_g$ to specified nodes

2) Analysis definition :-

for each analysis 7 items required.

a) Constraints

- enforce a relationship between dofs enforced in analysis

b) Numberer

- determines mapping b/w equation #s and dof how dof's are numbered.

c) System

- Constructs linear solver to store and solve the system of equations in the analysis.

d) Test

- Certain solution algorithm require a convergence test at the end of iteration step.

e) Algorithm

- determines the sequence of steps taken to solve the non-linear equation.

f) Integrator:-

- determines the meaning of the terms in the system of equation object used for,
  - (i) determine the predictive step for time $t+dt$
  - (ii) Specify the tangent matrix and residual vector at any iteration
  - (iii) determine the corrective step based on disp increment dU.

g) Analysis:

Three types available.

(i) Static Analysis → Solves $KU = R$ without M, C matrices

(ii) Transient Analysis → time dependent Analysis, time step = Constant

(iii) Variable Transient Analysis → time step = Variable

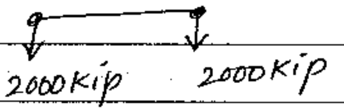
3) Analysis execution:-

analyze ___ ↓ no of steps

GRAVITY LOADS :- → independent of the type of lateral loading and hence they are considered part of the structural model.

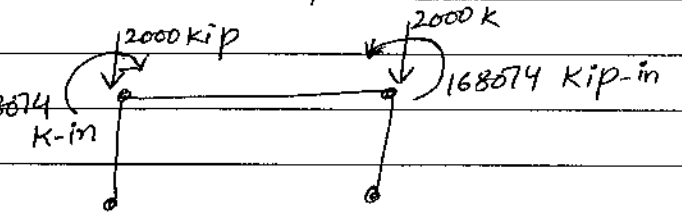
- nodal forces and moments.

$F = 4000/2 = 2000$ Kip



$M = \frac{wl^2}{12}$, $w = \frac{4000}{42 \times 12} = 7.94$ Kip/in

$M = 168674$ Kip in



- Load pattern definition :- · Plain pattern + Linear time Series

```
pattern plain ___ ___ {
              tag  time series type
load ___ ___ ___ ___
     node tag  Load values corresponding to 3 dofs eg in 2D
}
```

In example,

```
pattern Plain 1 Linear {
 load 3 0 -2000 -168074
 load 4 0 -2000 +168074
}
```

CREATE ANALYSIS :-

constraints Transformation
numberer RCM → Reverse Cuthill-McKee algorithm
System BandGeneral → banded system of equations

~~Test NormDispIncr 1.0e6 6~~

test NormDispIncr $\underset{\text{tolerance}}{\_\_}$ $\underset{\text{max number of interations}}{\_\_}$

The test is applied to,

$$K \Delta U = R$$

- Norm Unbalance checks $\sqrt{R^{\wedge}R} < tol$
- Norm disp Increment checks $\sqrt{\Delta U^T \Delta U} < tol$
- Energy Increment checks $\frac{1}{2}(\Delta U)^T(R) < tol$

Algorithm Newton

→ The system of nonlinear equations in static analysis is of the form

$$R(U, I) = \lambda P^* - F_R(U)$$

Type of integrator depends on → static or dynamic

for static analysis,

LoadControl $\lambda_n = \lambda_{n-1} + d\lambda$

DisplacementControl $U j_n = U j_{n-1} + dU j$

MinUnbalDispNorm $\frac{d}{d\lambda}(dU_n^T dU_n) = 0$

ArcLength $dU_n^T dU_n + \alpha^2 d\lambda_n = ds^2$

Integrator LoadControl $\underset{d\lambda_1}{0.1}$

analysis static

analyze 10

LoadConst -time 0.0

analyze $\underset{\text{no of increments}}{\_\_}$ $\underset{dt}{\_\_}$ $\underset{dt_{min}}{\_\_}$ $\underset{dt_{max}}{\_\_}$ $\underset{Jd}{\_\_}$

LATERAL LOADS - STATIC PUSHOVER :-

wipe (clears any previous openSees objects)

```
pattern plain 2 Linear {
 Load 3 100 0 0
 Load 4 100 0 0
}
```

- Previously in gravity loads LoadControl integrator was used. here DisplacementControl integrator will be used.

integrator DisplacementControl ___ ___ ___ (node tag, dof tag, dU_1)

Load is applied to node 3
in the dir of dof 1
with a disp increment of 0.1, so

```
integrator Displacement Control 3 1 0.1
```

for 1 inch Δ, 10 steps are needed

```
analyze 10
```

LATERAL LOADS - Cyclic Lateral Loads :-

```
pattern plain 3 Linear {
 Load 3 100 0 0
 Load 4 100 0 0
}
```

Load is applied to node 3 in the dir of dof 1
with $dU_1 = 0.1$, for reversal $dU_1 = -0.2 \rightarrow$ to amplitude -1 so,

```
integrator DisplacementControl 3 1 0.1
analyze 10
integrator Displacement Control 3 1 -0.2
analyze 10
integrator Displacement Control 3 1 0.1
analyze 10
```

This can be put in to a foreach loop.

```
foreach Dincr {0.1 -0.2 0.1} {
integrator DisplacementControl 3 1 $Dincr
analyze 10
}
```

LATERAL LOADS - Dynamic ground motion :-

- from PEER strong-motion database, dowhload BM68elc.th (ground motion)

- A number of tcl scripts are available to the user at OpenSees website.

ReadSMDFile.tcl is a script procedure which parses a ground motion record from the PEER strong motion database by finding dt in the record header then echoing data values to the output file. The file should be saved in same directory as the Opensees.exe.

- This file can be readin by the input file and used in analysis procedure where an acceleration-time series is defined and used in a UniformExcitation load pattern.

Series -dt ___ -filepath ___ -factor ___

$dt for file — name of file — 1

pattern UniformExcitation ___ ___ -accel ___

tag — direction no — time series type argument.

```
# Create load pattern
  source ReadSMDFile.tcl
  ReadSMDFile BM68elc.th BM68elc.acc dt
  set accelSeries "Series -dt $dt -filepath BM68elc.th
                                                  -factor 1";
  pattern UniformExcitation 2 1 -accel $accelSeries
```

- Stiffness and mass-proportional damping factors

rayleigh _ _ _ _

factors applied to

element/node mass matrix | element current stiffness matrix | element initial K | element committed K

Damping matrix $D = \alpha_M M + \beta_K K_{current} + \beta_{K_{in}} K_{initial} + \beta_{K_{comm}} K_{last\ comm}$

```
# set damping factors
  rayleigh 0 0 0 [expr 2*0.2/pow([
                               [eigen 1],0.5)]]
```

for math operations

Various analysis components,

```
# Create the analysis
  wipeAnalysis
  Constraints plain
  numberer RCM
  System UmfPack
  test NormDispIncr 1.0e-8 10
  algorithm Newton
  integrator Newmark 0.5 0.25
  Analysis Transient
  analyze [expr 10/0.02] 0.02
```

Time step does not have to be the same as the input ground motion. no of time steps $= \frac{\text{Total duration}}{\text{time step}} = \frac{10}{0.02}$

PBD Seminar by Graham H. Powel

Session 1

Present Vs. Future

a) Future methods are likely to based on probability.
eg a limit state might be statisfied if there only a 5% probability that its usage ratio > 1

b) Future methods may also use performance measures such as cost of repair ("dollars, downtime and deaths") (different from deformations or hinge rotations).

c) For the present, deterministic methods are enough of a challenge.

Session 1 ends.

Tips and Tricks for Computer aided Structural Analysis

by

Siakat Basak

In earlier generation analysis programs, user had to supply the programs the nodal coordinates, member incidences, material and sectional properties and loads. + how the structure is supported $\xrightarrow{\text{program calculates}}$ Memberforces, nodal reactions, joint displacements.

Now ⟶ draw your model on screen.

Linear Static Stress Analysis :-

Assumptions

a) Loads applied slowly (no inertial or damping forces)

b) Linearity assumptions F–D, if

(i) All materials ⟶ hook's Law

(ii) Induced disp are small enough to ignore the change in K caused by loading.

(iii) BC do not vary during loading.