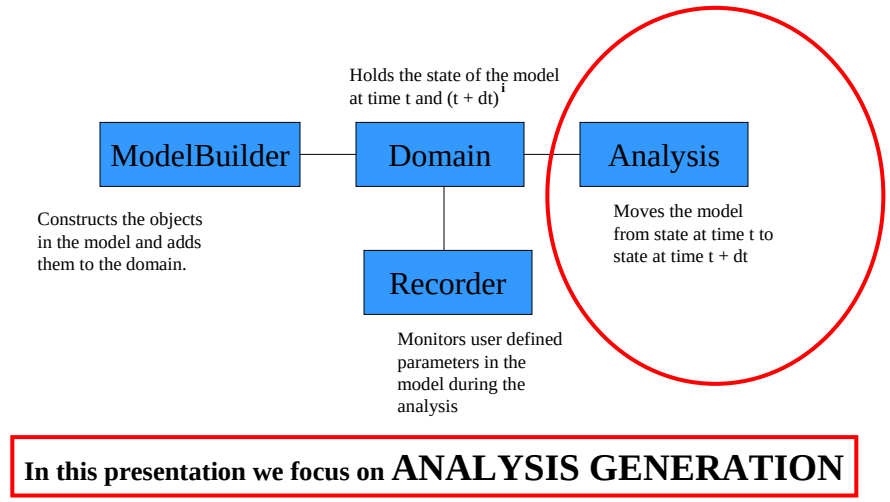
QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

OpenSees: Analysis

Frank McKenna

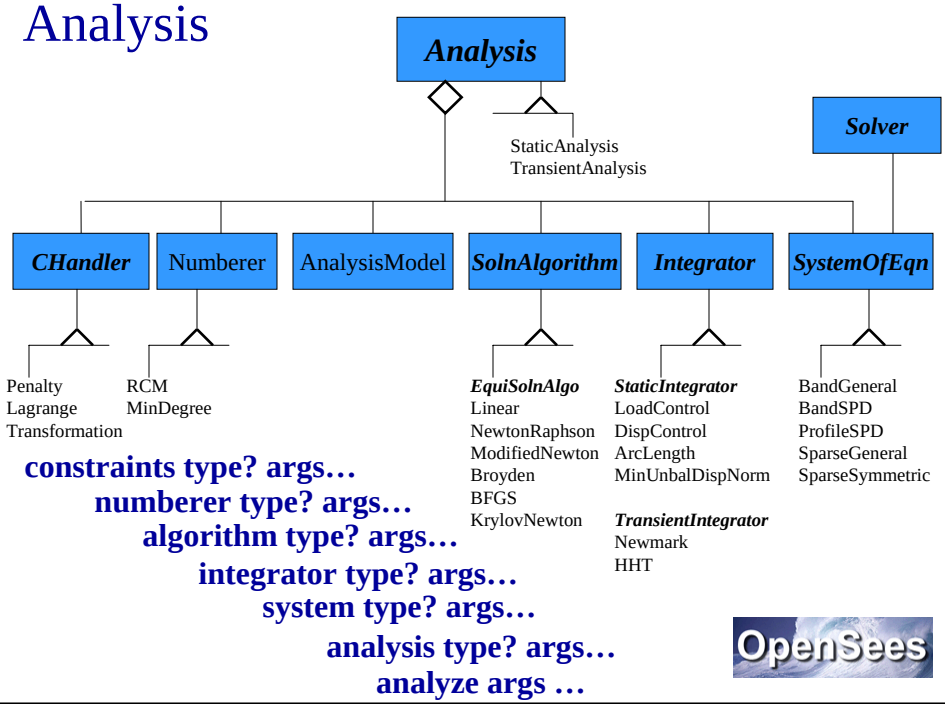
OpenSees and NEESgrid Simulation
Component Users Workshop
Sept 2-3, 2004

Main Abstractions in OpenSees



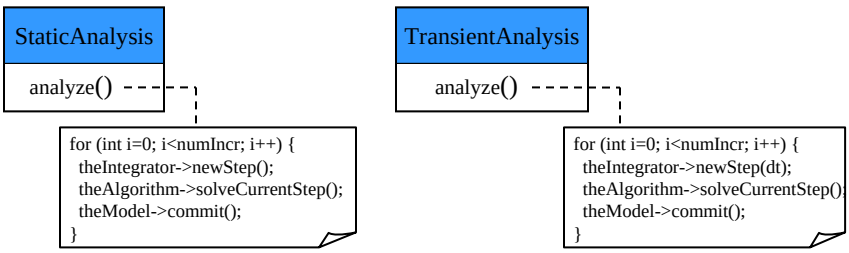
In this presentation we focus on ANALYSIS GENERATION

Analysis



analysis command:

- Static Analysis — *analysis static*
- Transient Analysis — *analysis transient*
  - both incremental solution strategies



- Eigenvalue
  - general eigenvalue problem
    $(K-\lambda M)\Phi=0$ — *eigen numModes? -general*
  - standard eigenvalue problem
    $(K-\lambda I)\Phi=0$ — *eigen numModes? -standard*

integrator command:

-determines the predictive step for time t+δt
-specifies the tangent matrix and residual vector at any iteration
-determines the corrective step based on ΔU

•Transient Integrator for Use in Transient Analysis

Nonlinear equation of the form:

$$\mathbf{R(U, \dot{U}, \ddot{U}) = P(t) - F_I(\ddot{U}) - F_R(U, \dot{U})}$$

▪Newmark Method

integrator Newmark α β

▪Hilbert-Hughes-Taylor Method

integrator Newmark α

•Static Integrators for Use in Static Analysis

Nonlinear equation of the form:

$$\mathbf{R(U, \lambda) = \lambda P^* - FR(U)}$$

▪Load Control

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

integrator LoadControl $\Delta\lambda$

*does not require a reference load, i.e. loads in load patterns with Linear series and all other loads constant.

▪Displacement Control

$Uj_n = Uj_{n-1} + \Delta Uj$

integrator DisplacementControl node dof $\Delta\lambda$

▪Arc Length

$\Delta U_n \wedge \Delta U_n + \alpha^2 \Delta\lambda_n = \Delta s^2$

integrator LoadControl α Δs

▪Minimum Unbalance Displacement Norm

$$\frac{d}{d\Delta\lambda}(\Delta U_n \wedge \Delta U_n) = 0$$

integrator LoadControl $\Delta\lambda$

algorithm command:

- to specify the steps taken to solve the nonlinear equation

•Linear Algorithm

```
theIntegrator->formUnbalance();
theIntegrator->formTangent();
theSOE->solve()
theIntegrator->update(theSOE->getX());
```

algorithm Linear

•Newton-Raphson Algorithm

```
theIntegrator->formUnbalance();
do {
  theIntegrator->formTangent();
  theSOE->solve()
  theIntegrator->update(theSOE->getX());
  theIntegrator->formUnbalance();
} while (theTest->test() == fail)
```

algorithm Newton

•Modified Newton Algorithm

algorithm ModifiedNewton <-initial>

•Accelerated Modified Newton Algorithm

algorithm KrylovNewton <-initial>

constraints command:

- to specify how the constraints are enforced

$\mathbf{U}_c = \mathbf{C}_{rc}\,\mathbf{U}_r$

$\mathbf{C}\,\mathbf{U} = \mathbf{0}$

$\mathbf{T}\,\mathbf{U}_r = [\mathbf{U}_r \;\; \mathbf{U}_c]^{\wedge}$

$[\mathbf{C}_r \;\; \mathbf{C}_c]^{\wedge}[\mathbf{U}_r \;\; \mathbf{U}_c] = 0$

•Transformation Handler

$K^* U_r = R^*$ $\quad K^*=T^{\wedge}KT$ $\quad R^*=T^{\wedge}R$

constraints Transformation

in OpenSees currently don't allow retained node in one constraint to be a constrained node in another constraint

•Lagrange Handler

$$\begin{bmatrix} \mathbf{K} & \mathbf{C}^{\wedge} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{R} \\ \mathbf{Q} \end{bmatrix}$$

constraints Lagrange

•Penalty Handler

$[\mathbf{K} + \mathbf{C}^{\wedge}\boldsymbol{\alpha}\mathbf{C}]\,\mathbf{U} = [\mathbf{R} + \mathbf{C}^{\wedge}\boldsymbol{\alpha}\mathbf{Q}]$

constraints Penalty $\alpha_{sp}?\alpha_{mp}?$

system command:

- to specify how matrix equation KU = R is stored and solved

•Profile Symmetric Positive Definite

system ProfileSPD

•Banded Symmetric Positive Definite

system BandSPD

•Sparse Symmetric Positive Definite

system SparseSPD

•Banded General

system BandGeneral

•Sparse Symmetric

system SparseGeneral

system Umfpack

numberer command:

- to specify how the degrees of freedom are numbered

•Plain Numberer

nodes are assigned dof arbitrarily

numberer Plain

•Plain Numberer

nodes are assigned dof using the Reverse Cuthill-McKee algorithm

numberer RCM

test command:

- to specify when convergence has been achieved

all look at system: **KU = R**

•Norm Unbalance

$\sqrt{\mathbf{R}\text{\^{}}\mathbf{R}} < \mathbf{tol}$ *test NormUnbalance tol? numIter? <flag?>*

•Norm Displacement Increment

$\sqrt{\mathbf{U}\text{\^{}}\mathbf{U}} < \mathbf{tol}$ *test NormDispIncr tol? numIter? <flag?>*

•Norm Energy Increment

$\frac{1}{2}$ **(U^R) < tol** *test NormEnergyIncr tol? numIter? <flag?>*

•Relative Tests

test RelativeNormUnbalance tol? numIter? <flag?>

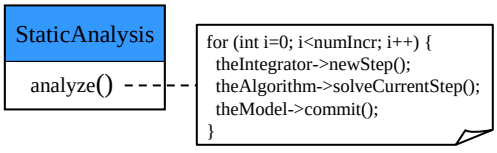
test RelativeNormDispIncr tol? numIter? <flag?>

test RelativeNormEnergyIncr tol? numIter? <flag?>

analyze command:

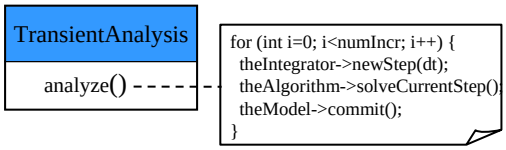
- to perform the static/transient analysis

•Static Analysis



analyze numIter?

•Transient Analysis



analyze numIter? Δt?

Example Analysis:

- Static Nonlinear Analysis with LoadControl

```
constraints transformation
numberer RCM
system BandGeneral
test NormDispIncr 1.0e-6 6 2
algorithm Newton
integrator LoadControl 0.1
analysis Static
analyze 10
```

- Transient Nonlinear Analysis with Newmark

```
constraints transformation
numberer RCM
system BandGeneral
test NormDispIncr 1.0e-6 6 2
algorithm Newton
integrator Newmark 0.5 0.25
analysis Transient
analyze 2000 0.01
```