

# Chapter 1: Development of Subroutines for ABAQUS

## 1.1. Introduction

Development of Subroutines for ABAQUS is one of the prominent features of this software, allowing the program to be customized for specific applications.

The range of issues covered by subroutines, along with a simple and understandable interface, has made ABAQUS an extremely useful tool for users globally for numerical analysis. However, are the default options available in ABAQUS capable of meeting the needs of users? As advancements are progressing quickly in modern science and the presentation of numerical models and structural equations describing the behavior of matter improves, can having software such as ABAQUS adopt solutions to meet the needs of researchers nowadays?

ABAQUS, along with all the major and minor selections available to the user, has the ability to provide convenient customizations to meet the demands of users with the aid of FORTRAN programming software (which is the foundation of large computing software such as ABAQUS, ANSYS, NASTRAN, etc., in research and industry). Controls and rules of finite elements and programming in FORTRAN has enabled the user to surpass any limitations in developing numerical solutions.

Both ABAQUS/Standard and ABAQUS/Explicit consist of subroutines that are listed in Table 1-1.

Table 1-1: Subroutines provided in ABAQUS solvers

ABAQUS/Standard	<p>CREEP,DFLOW,DFLUX,DISP,DLOAD,FILM,FLOW,FRIC,FRIC_COEF,GAP  CON,GAPELECTR,HARDINI,HETVAL,MPC,ORIENT,RSURFU,SDVINI,SI  GINI,UAMP,UANISOHYPER_INV,UANISOHYPER_STRAIN,UCORR,UDE  CURRENT,UDEMPOTENTIAL,UDMGINI,UEL,UELMAT,UDSECURRENT,  UEXPAN,UEXTERNALDB,UFIELD,UFLUID,UFLUIDLEAKOFF,UGENS,U  HARD,UHYPEL,UHYPER,UINTER,UMASFL,UMAT,UMATHT,UMESHMO  TION,UMOTION,UMULLINS,UPOREP,UPRESS,UPSD,URDFIL,USDFLD,U  TEMP,UTRACLOAD,UTRS,UVARM,UWAVE,VOIDRI,</p>
ABAQUS/Explicit	<p>VDISP,VDLOAD,VFABRIC,VFRIC,VFRIC_COEF,VFRICTION,VUAMP,VU  ANISOHYPER_INV,VUANISOHYPER_STRAIN,VUEL,VUFIELD,VUFLUI  DEXCH,VUFLUIDEXCHEFFAREA,VUHARD,VUINTER,VUINTERACTIO  N,VUMAT,VUMULLINS,VUSDFLD,VUTRS,VUVISCOSITY,VWAVE</p>

Figure 1-1 illustrates the basic data flow of data and actions from the beginning of analysis in ABAQUS/Standard to the end of the step. The figure also shows the way and where the steps are implemented during analysis.

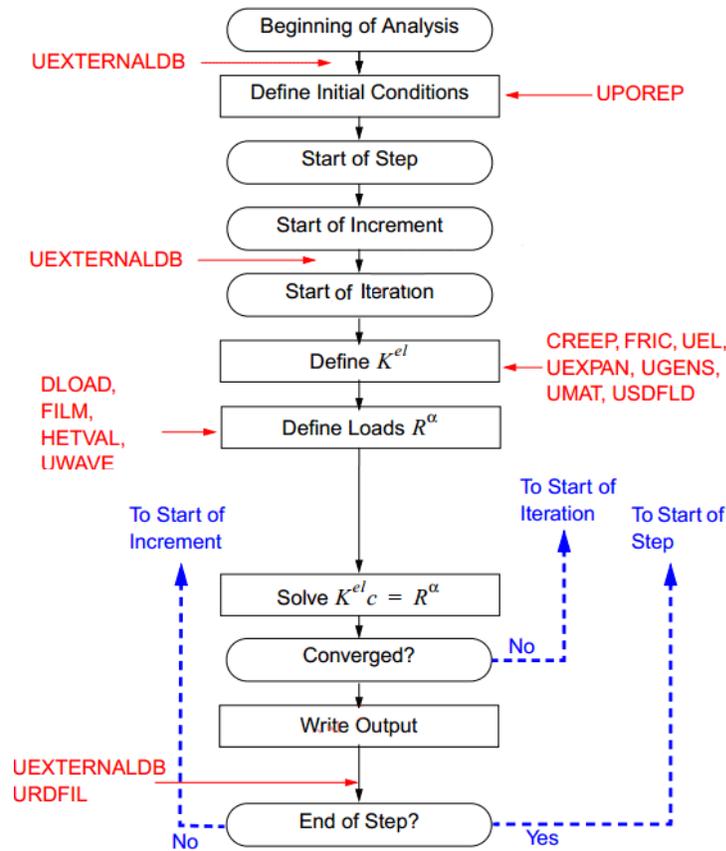


Figure 1-1: Dataflow and actions in ABAQUS/Standard

Each subroutine in ABAQUS consists of a set of variables and parameters that must be written in a predetermined and specific format.

UMAT<sup>1</sup> is the name of the ABAQUS sets used in material models developed by users (UMAT in implicit analysis mode and VUMAT equivalent to explicit analysis mode).

In UMAT, the user must introduce the behavior of matter to ABAQUS using the theory in the form of FORTRAN code (\*.for). With this program, the user informs the software how much the force or displacement of a substance is, how much stress and strain are and when to yield, and under what conditions failure or fatigue occurs. Whenever ABAQUS intends to load, calculate tension and strain, it requires the user to adopt a step-by-step approach. Figure 1-2 represents how a UMAT subroutine works.

<sup>1</sup> - User Material

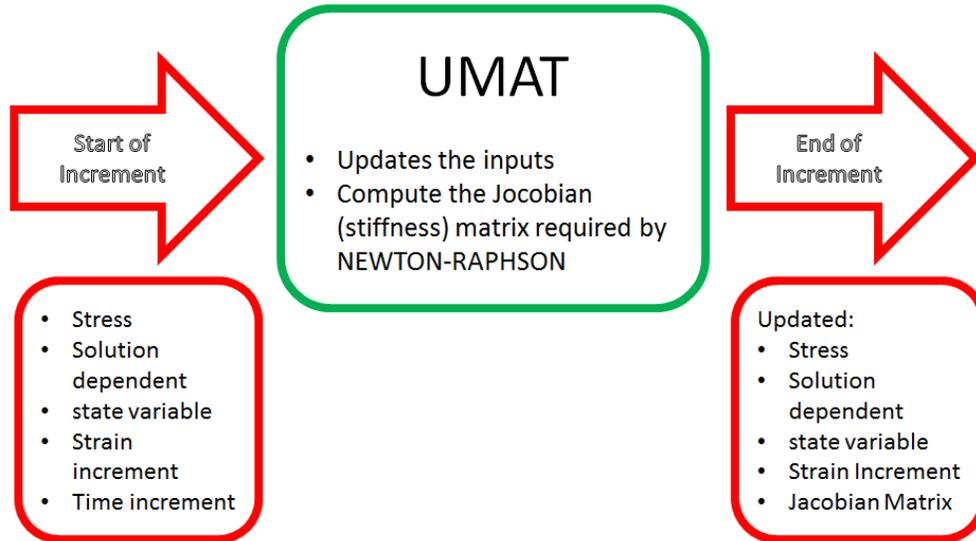


Figure 1-2: UMAT flowchart

## 1.2. Problem description

This sample demonstrates how to get started with using/writing a UMAT in ABAQUS. In this sample, a one-floor frame will be considered that is subjected to an earthquake (Figure 1-3).

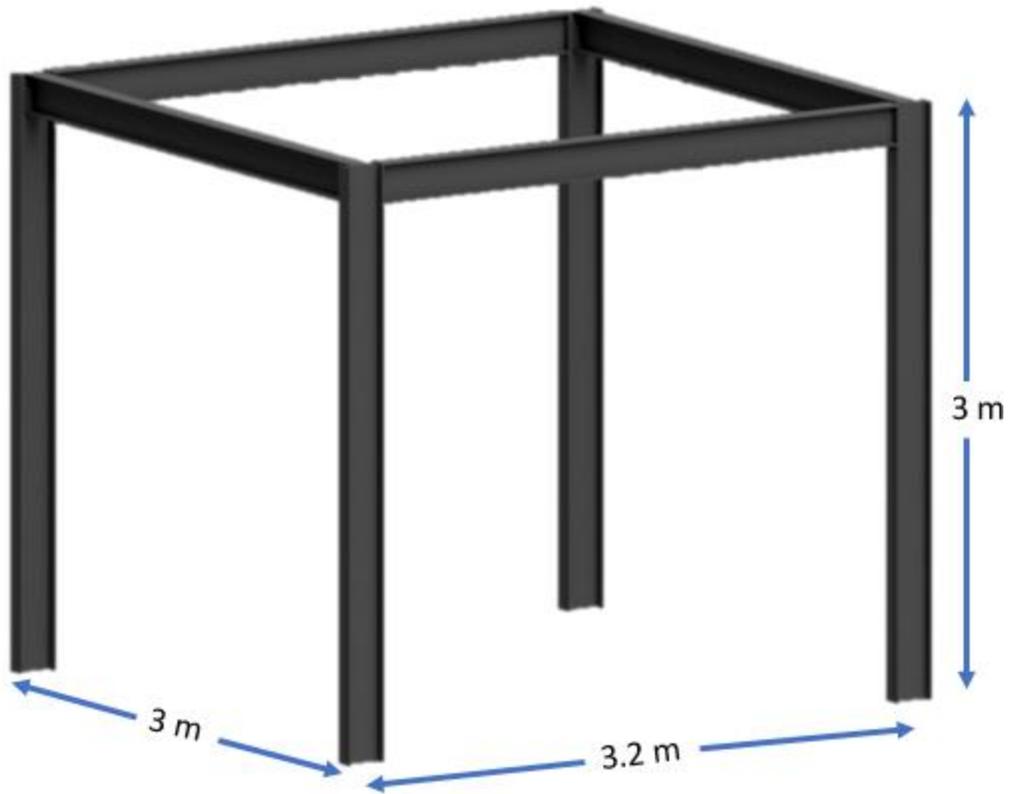


Figure 1-3: Overall dimensions (in meters) of a frame

The frame is made of IPE200 beams (Figure 1-4) in which all beams are welded together.

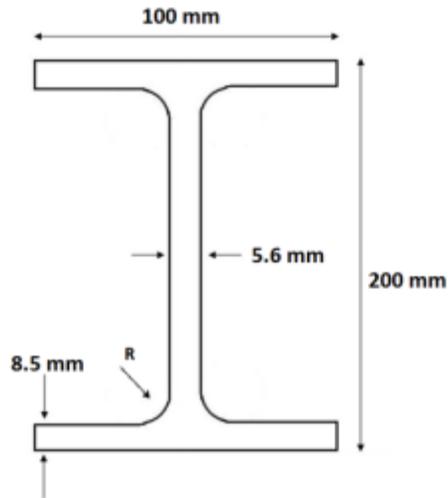


Figure 1-4: IPE200 dimensions

The frame vibrates by acceleration history that is registered as an “El Centro”<sup>2</sup> earthquake. This can be found in [www.peer.com](http://www.peer.com) (Figure 1-5).

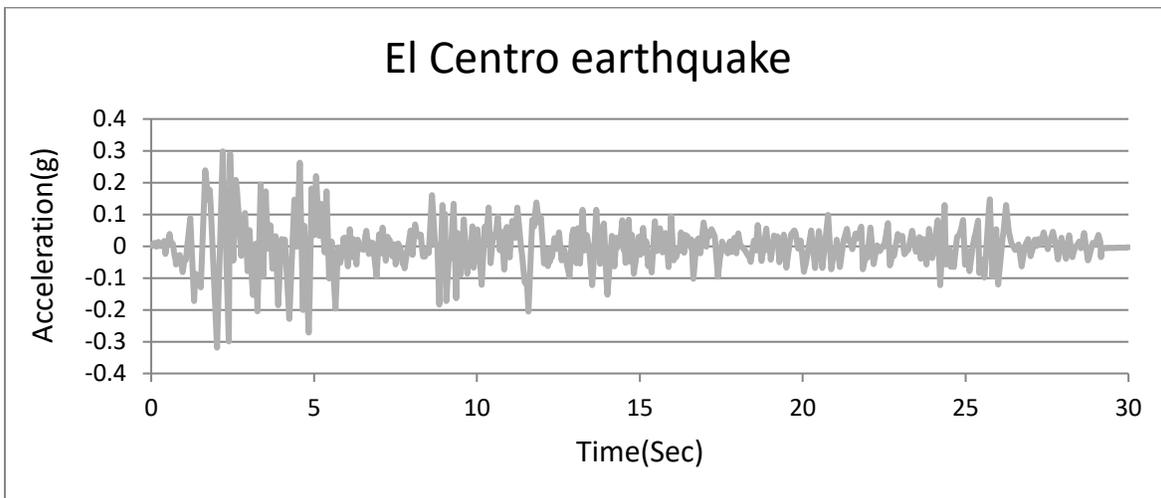


Figure 1-5: “El Centro” acceleration time history

Material for the beam is steel and is considered as linear isotropic elastic. The material will be defined in the UMAT subroutine.

---

<sup>2</sup> - USA - 1940

Isotropic linear elastic includes:

Isotropic: meaning all frame material properties are the same in all directions.

Linear elasticity: meaning when the structure is unloaded, it is returned to its initial state.

### 1.3. Objectives

- 1) To study writing subroutine for ABAQUS.
- 2) To practice an earthquake problem.

### 1.4. Modeling

**Writing UMAT:** Hook's law is the best-known equation in a structure that describes the relation between mechanical loads and displacement. In the Finite element method, many differential equations obtained from Hook's law should be solved, that deforms the structure incrementally and computes the resulting forces generated inside the material. Hook's law can be written in two forms as given in Equations 1 and 2:

$$X = C.F \tag{1}$$

$$F = K.X \tag{2}$$

Where C is compliance and K is the stiffness of the material. Finite element codes, such as ABAQUS use these equations to compute the stresses for a given deformation.

In UMAT, the user must determine the stiffness components and input them in such a way that the software can calculate the stiffness matrix. In the first step of this sample, the stiffness matrix has been generated directly. The purpose is to build a stiffness matrix, which defines the resistance of the structure in different directions. There are 3 directions in 3-dimensional spaces, and each direction generates a response in all 3 directions. Hence, it has a total of 9 (3×3) responses. So, the stiffness matrix [K] should include 9 resistance values. Table 1-2 describes the components of the stiffness matrix components.

Table 1-2: Stiffness matrix components

Stiffness component	Definition
---------------------	------------

$K_{xx}$	Stiffness in x direction due to loads in x direction
$K_{yy}$	Stiffness in y direction due to loads in y direction
$K_{zz}$	Stiffness in z direction due to loads in z direction
$K_{xy}$	Stiffness in x direction due to loads in y direction
$K_{yx}$	Stiffness in y direction due to loads in x direction
$K_{zy}$	Stiffness in z direction due to loads in y direction
$K_{yz}$	Stiffness in y direction due to loads in z direction
$K_{xz}$	Stiffness in x direction due to loads in z direction
$K_{zx}$	Stiffness in z direction due to loads in x direction

The next question is, “How can the values for the components of [K] matrix be determined?”.

The main step in writing UMAT is extracting components of the stiffness matrix according to the material model, which contains plasticity, viscoelasticity, hyperelasticity, damage, degradation, and so forth.

For isotropic linear elastic material, two independent material parameters are needed to define the state of the material. One constant is needed to measure the resistance in the direction of the load and the other constant measures how much the lattice is dragged sideways for a vertical (perpendicular) load. The former is called Young’s modulus (E), and the latter is Poisson’s ratio ( $\nu$ ).

For extracting the components in this sample, Hooke’s law should be written in 3 dimensional according to these two constants, as shown in Equation3.

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{zx} \\ \sigma_{xy} \end{bmatrix} = \frac{E}{(1+\nu)+(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-2\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-2\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 1-2\nu \end{bmatrix} \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \epsilon_{yz} \\ \epsilon_{zx} \\ \epsilon_{xy} \end{bmatrix} \quad (3)$$

In the next step, , the equation must be entered in the UMAT subroutine. The subroutine; such as all other subroutines, have a common interface, as shown in Figure 1-6.

```

SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
 1 RPL,DDSDDT,DRPLDE,DRPLDT,
 2 STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,
 3 NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,
 4 CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,KSTEP,KINC)
C
  INCLUDE 'ABA_PARAM.INC'
C
  CHARACTER*80 CMNAME
  DIMENSION STRESS(NTENS),STATEV(NSTATV),
 1 DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS),
 2 STRAN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1),
 3 PROPS(NPROPS),COORDS(3),DROT(3,3),DFGRD0(3,3),DFGRD1(3,3)

  user coding to define DDSDDE, STRESS, STATEV, SSE, SPD, SCD
  and, if necessary, RPL, DDSDDT, DRPLDE, DRPLDT, PNEWDT

  RETURN
  END

```

Figure 1-6: UMAT subroutine interface

There are 3 essential arguments for the subroutine, including DDSDDE, STRESS, STRAIN that is shown in Table 1-3.

Table 1-3: UMAT essential arguments

Argument in ABAQUS subroutines	Definition
DDSDDE	Stiffness matrix <sup>3</sup> of the material
STRESS	Stress tensor matrix
STRAIN	Strain tensor matrix

In every increment, the STRAIN tensor is provided as input to the UMAT once. Then, UMAT using DDSDDE and STRAIN matrices updates the STRESS matrix and returns it as an output for the current increment before moving to the next increment. The final UMAT for this sample has been written below. In the subroutine, PROPS (1) and PROPS (2) are mechanical constants that

<sup>3</sup> - In ABAQUS documentation, stiffness matrix called Jacobian matrix

should be defined in the Create Material dialog box in ABAQUS that will be explained in the next section.

```
SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,  
 1 RPL,DDSDDT,DRPLDE,DRPLDT,  
 2 STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,  
 3 NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,  
 4 CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,JSTEP,KINC)  
  
C  
  INCLUDE 'ABA_PARAM.INC'  
  
C  
  CHARACTER*80 CMNAME  
  
  DIMENSION STRESS(NTENS),STATEV(NSTATV),  
 1 DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS),  
 2 STRAN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1),  
 3 PROPS(NPROPS),COORDS(3),DROT(3,3),DFGRD0(3,3),DFGRD1(3,3),  
 4 JSTEP(4)  
  
C  ELASTIC USER SUBROUTINE  
  
  PARAMETER (ONE=1.0D0, TWO=2.0D0)  
  
    E=PROPS(1)  
    NU=PROPS(2)  
  
    A=E/(ONE+NU)/(ONE-TWO*NU)  
  
    B=(ONE-NU)  
  
    C=(ONE-TWO*NU)  
  
    DO I=1,NTENS
```

```

DO J=1,NTENS
DDSDDE(I,J)=0.0D0
ENDDO
ENDDO
DDSDDE(1,1)=(A*B)
DDSDDE(2,2)=(A*B)
DDSDDE(3,3)=(A*B)
DDSDDE(4,4)=(A*C)
DDSDDE(5,5)=(A*C)
DDSDDE(6,6)=(A*C)
DDSDDE(1,2)=(A*NU)
DDSDDE(1,3)=(A*NU)
DDSDDE(2,3)=(A*NU)
DDSDDE(2,1)=(A*NU)
DDSDDE(3,1)=(A*ANU)
DDSDDE(3,2)=(A*NU)
DO I=1,NTENS
DO J=1,NTENS
STRESS(I)=STRESS(I)+DDSDDE(I,J)*DSTRAN(J)
ENDDO
ENDDO
RETURN
END

```

For using this subroutine, it should be saved with (\*.), for the extension.

As mentioned earlier, ABAQUS subroutines are Fortran commands. Therefore, it is necessary to create them in Fortran programming software, where they can be ‘debugged’ by the software. This will then allow the program to be installed and placed on the user’s computer, linking it to ABAQUS, as described in the ABAQUS Documentation. Advanced users might not need to write[code] in the Fortran, as they could write the subroutine in Notepad.exe and save it with an (\*.\*) extension. In any case, Fortran and ABAQUS should be linked.

Save the subroutine as “UMAT.for”.

**ABAQUS modeling:** In the next step, the FEA model should be prepared in ABAQUS/CAE. In this example, a 3D model of a one-floor frame containing one beam is considered. The frame and section dimensions are shown in Figures 3 and 4. To simplify modeling, fillets in the section (r) have been ignored.

From the start menu, open ABAQUS/CAE and close the Start Session dialog box (Figure 1-7).

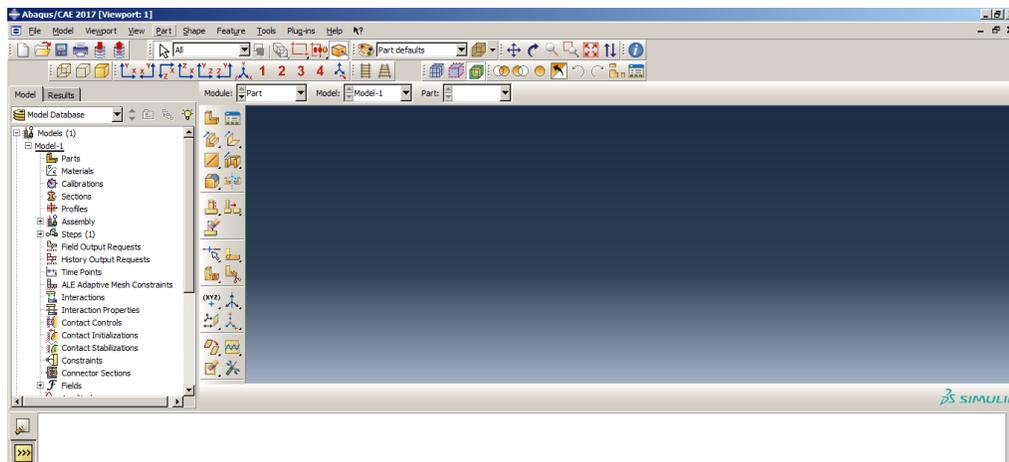
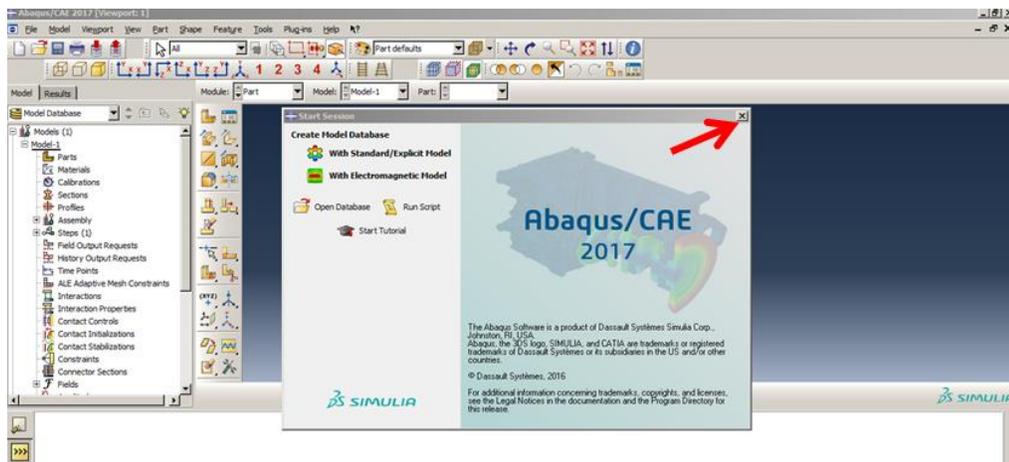


Figure 1-7: ABAQUS/CAE

### 1.4.1. Parts Module

The first step is to define an IPE shape, and then it should be properly constrained. The dimensions should first be defined.

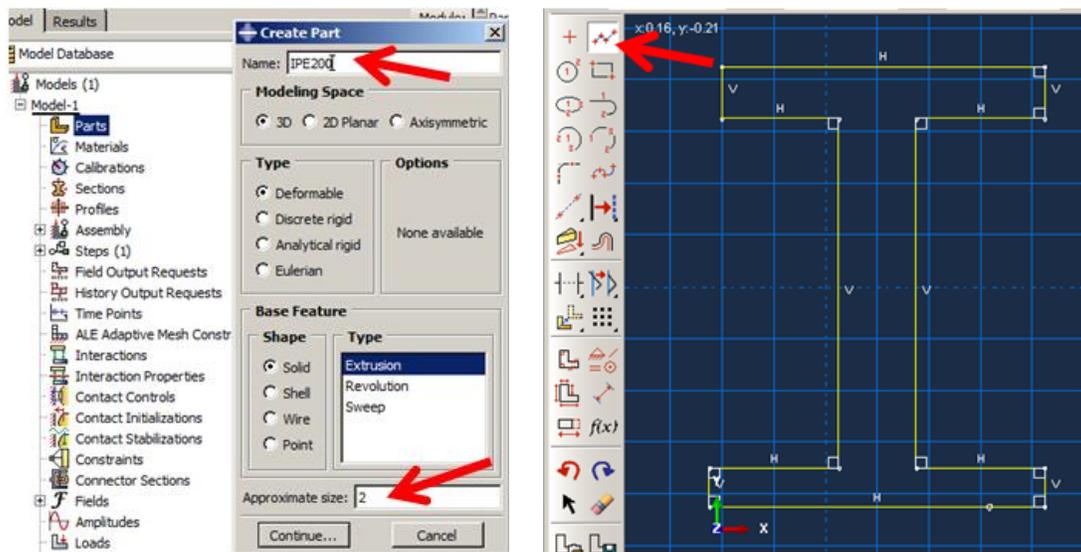
Double click on Parts in the Model Tree to open the Create Part dialog box and name the part. Set the approximate size to 2 and click Continue to enter sketcher.

Select Create Lines: Connected and draw an IPE section.

Select Add Constraint and make corresponding lines of equal length.

Select Add Dimension and determine the dimensions according to Figure 4.

Click the middle mouse button twice to open the Edit Base Extrusion dialog box. Consider 3 for the Depth and click Ok to apply and close the dialog box (Figure 1-8).



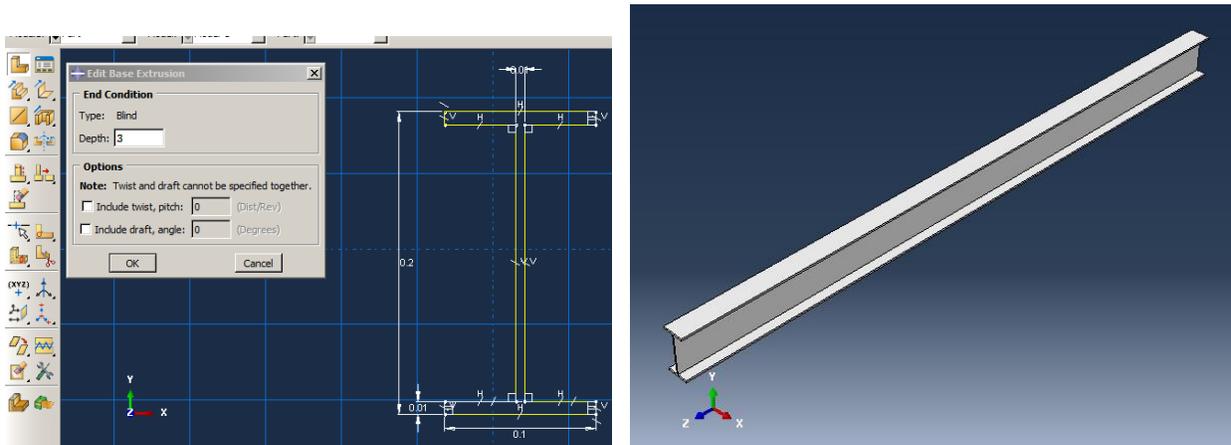
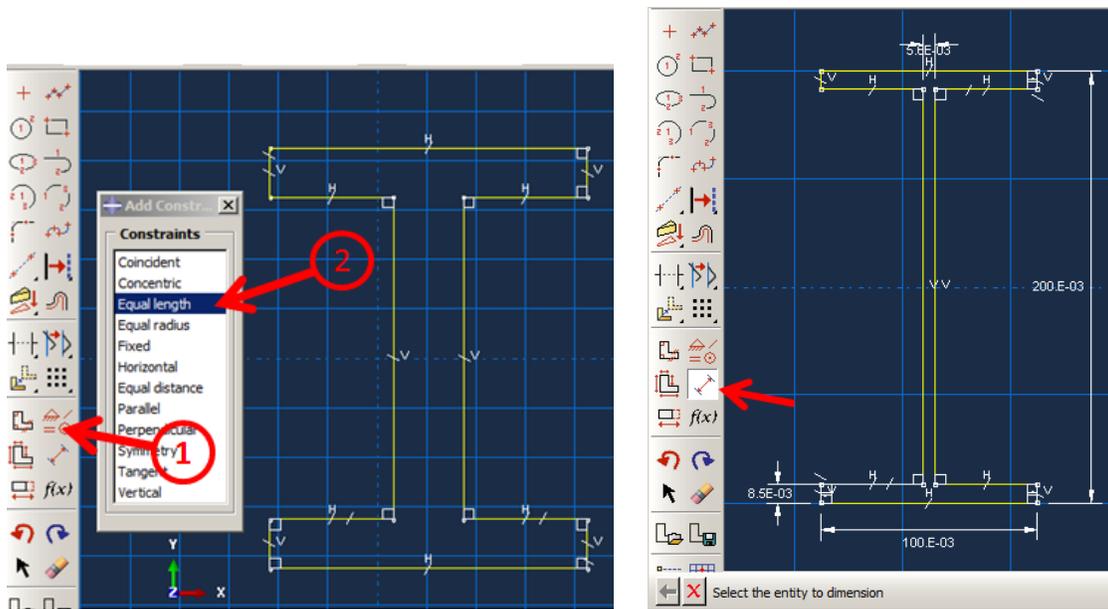


Figure 1-8: Creating the IPE 200 beam

## 1.4.2. Property Module

### 1.4.2.1. Material Properties

All structures are made of steel with the material properties listed in Table 1-4.

Table 1-4: Steel material properties as linear elastic and isotropic

Steel Properties	
Young's modulus	210 GPa

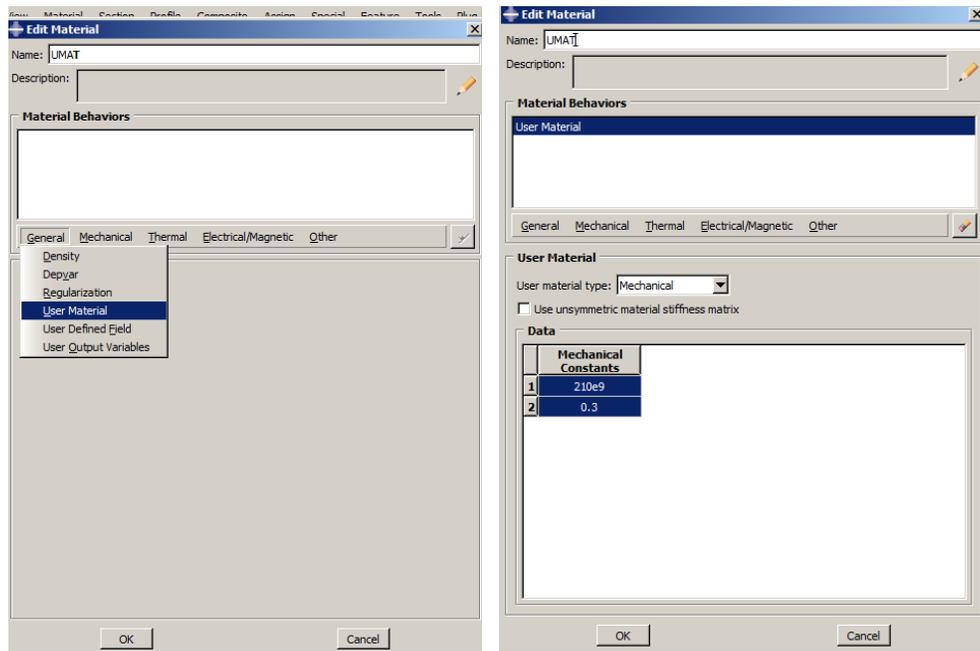
Poisson's Ratio	0.3
Density	7850 kg/m <sup>3</sup>

Double click on Material in the Model Tree to open the Create Material dialog box and name the material.

Select General → User Material to define the UMAT subroutine Mechanical Constants; Young's modulus and Poisson's ratios defined in Table 1-3. After entering Young's modulus, next, create the second row to input Poisson's ratio. They are ROPS(1) and PROPS(2) which are implemented in the UMAT subroutine.

Select General → Density to and enter 7850 as Mass Density, as shown in Figure 9.

Click Ok to save the material and close the dialog box (Figure 1-9).



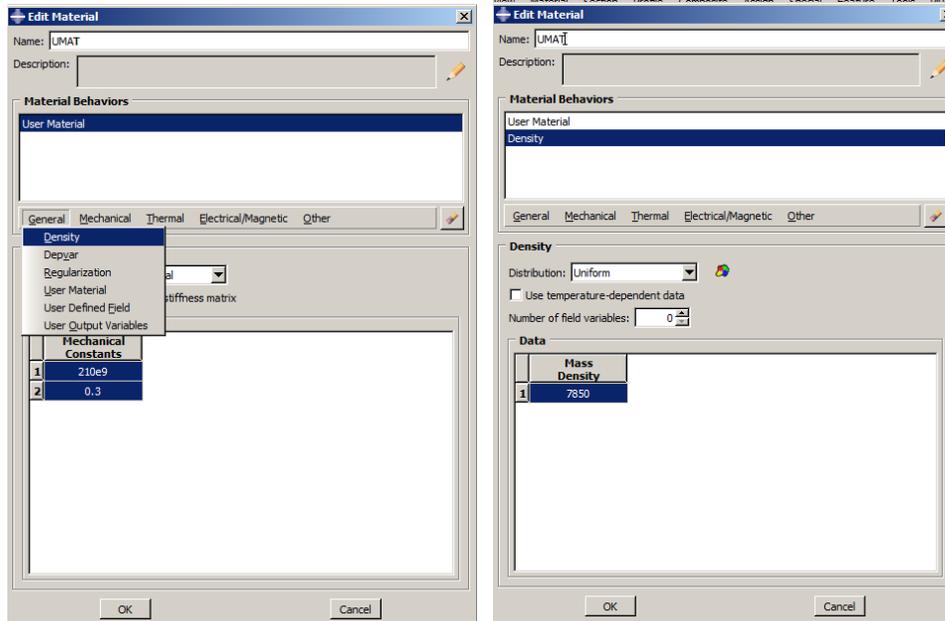


Figure 1-9: Defining the material

### 1.4.2.2. Section Properties

As Steel was assumed as homogenous, this means that the material properties are similar in all locations.

Double click on Sections in the Model Tree and then open the Create Section dialog box. Name the section and keep the defaults by selecting Continue and clicking Ok to define a solid homogenous section (Figure 1-10).

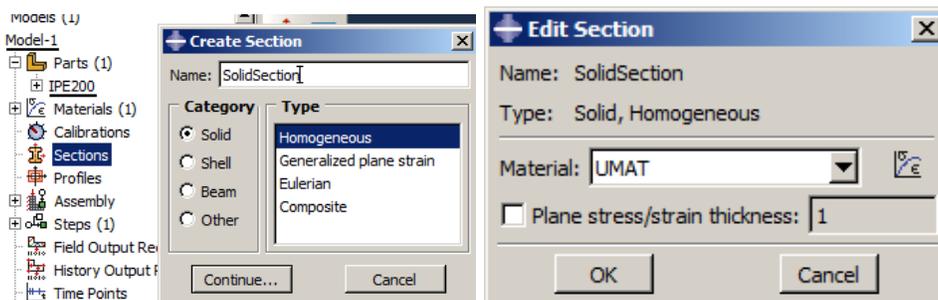


Figure 1-10: Creating a solid homogenous section

### 1.4.2.3. Section Assignment

In the next step, the section that was defined early should be assigned to the beam.

Double click on Section Assignments underneath IPE200 in the Model Tree. Then, select beam and click the middle mouse button to open the Edit Section Assignment dialog box. To assign the section, click Ok, and close the dialog box (Figure 1-11).

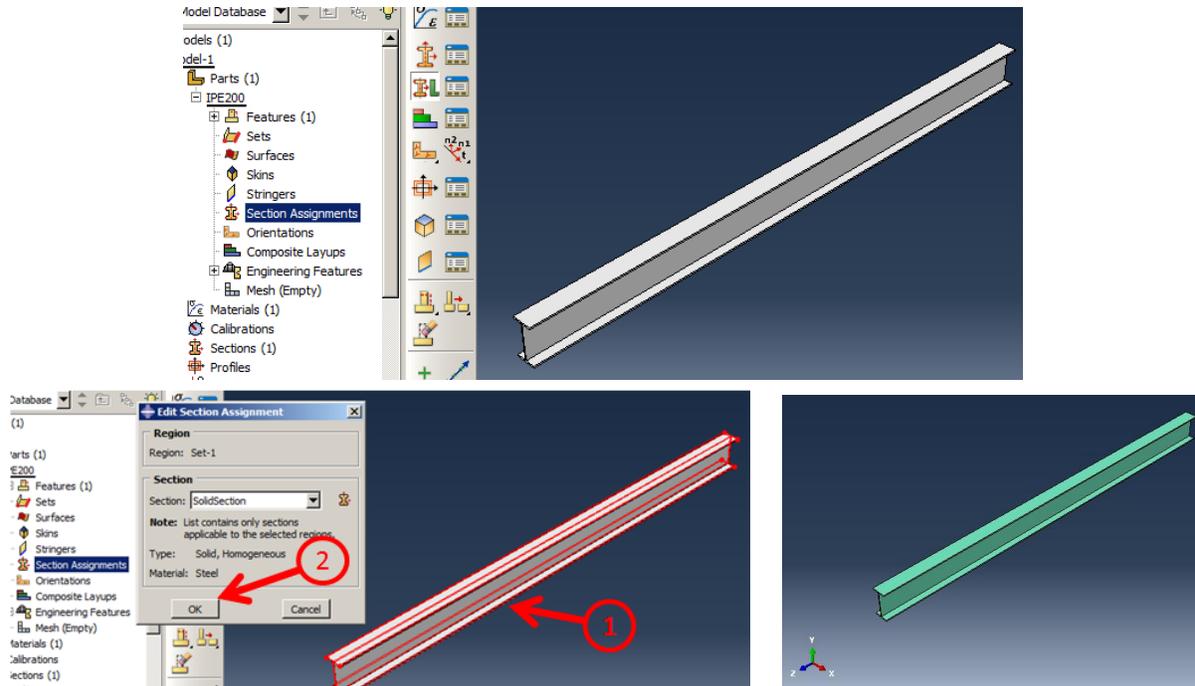


Figure 1-11: Assigning the section

### 1.4.3. Assembly Module

Next, the assembly should be generated by defining the dependent part instances from IPE200, which includes 8 instances.

Double click on Instances underneath of Assembly in the Model Tree to open the Create Instance dialog box. Click Ok to all and add IPE200 instance to assembly.

Select instance → Rotate, then choose beam instance as the instance to rotate and click the middle mouse button. Then choose points, as shown in Figure 11 that defines the axis of rotation. Click the middle mouse button to accept 90 degrees as the angle of rotation. Then click the middle mouse button to accept the new position (Figure 1-12).

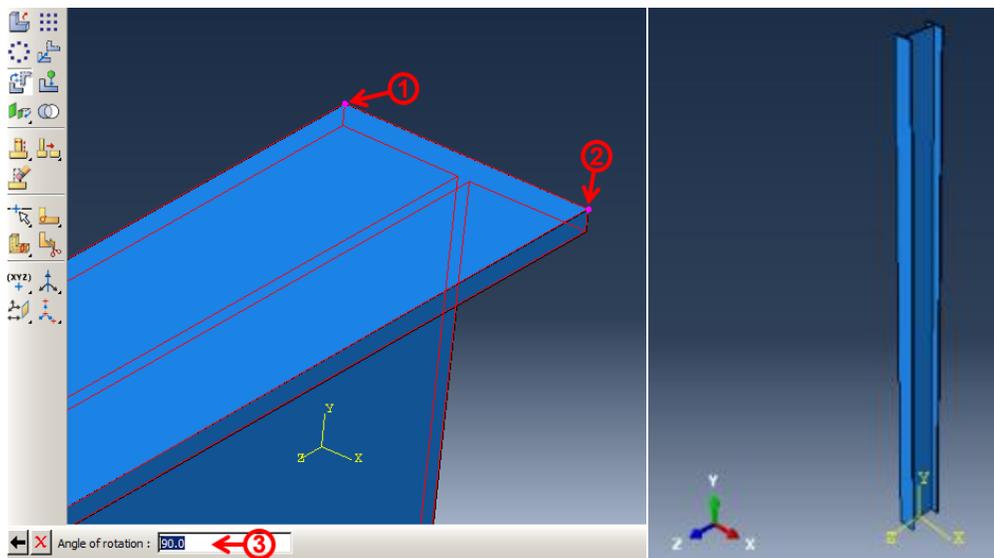
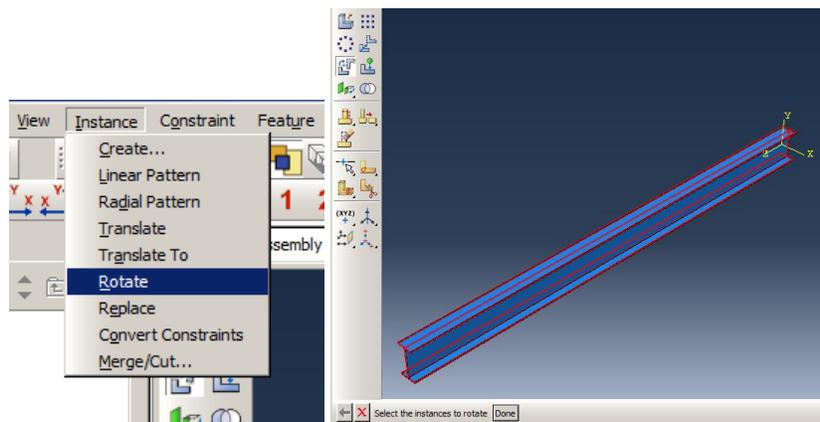
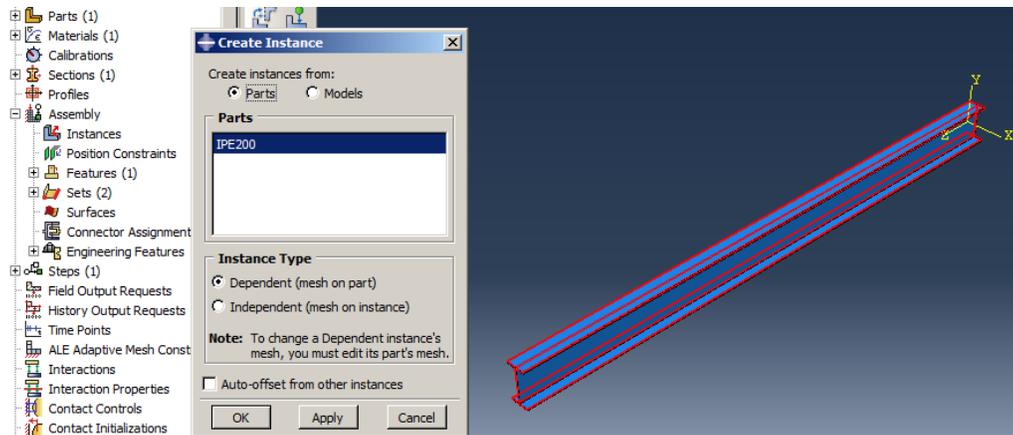


Figure 1-12: Rotating the beam

Select Instance → Linear Pattern to make an array of the frame columns. Click on the instance and click the middle mouse button to open the Linear Pattern dialog box. Click on the Edit icon in the

dialog box and select axis z and click the middle mouse button. Then enter 3.2 and 3.0056 as offset along the z and x directions respectively and click Ok to apply and close the dialog box (Figure 1-13).

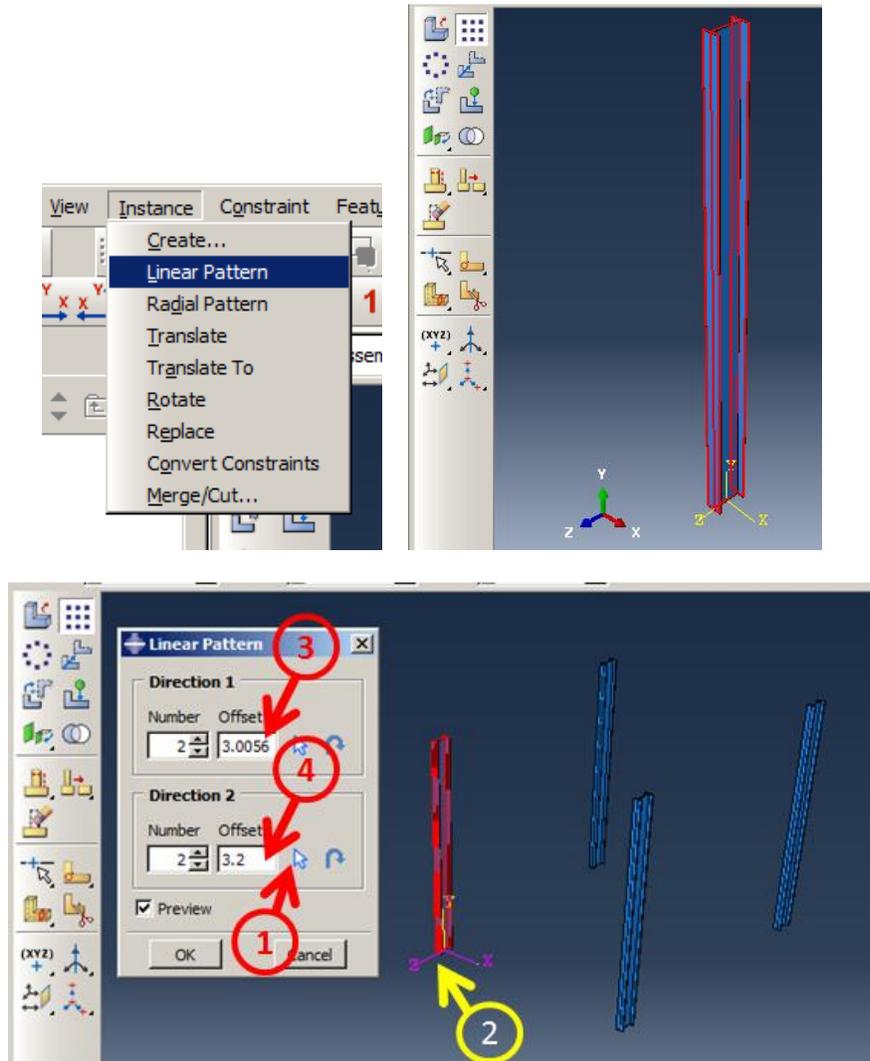
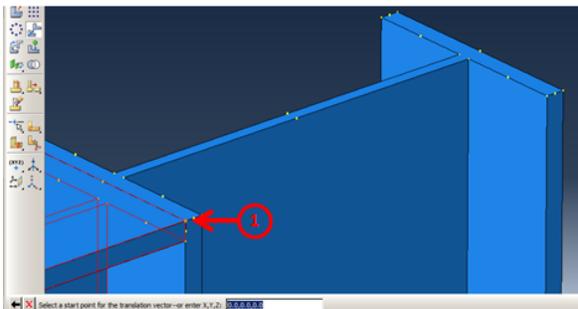
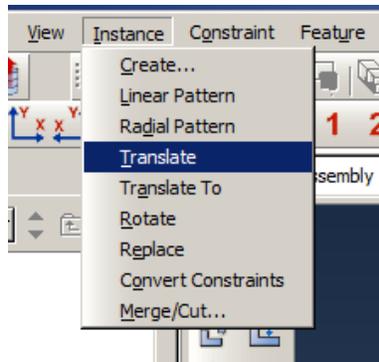
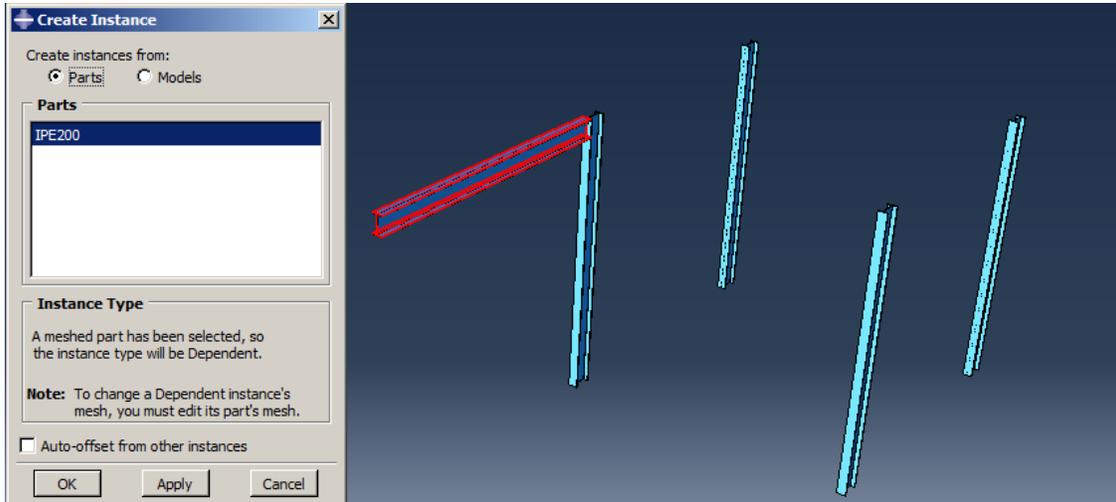
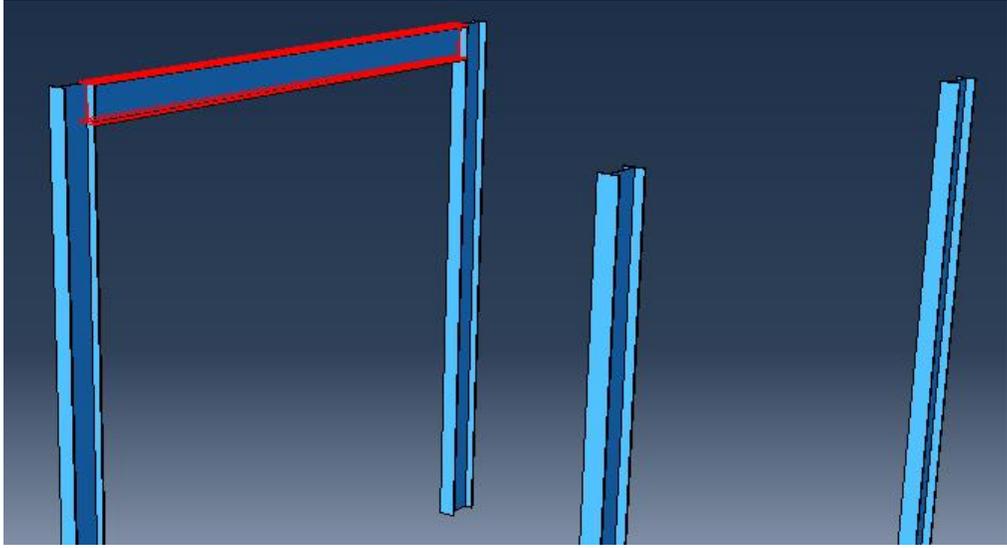


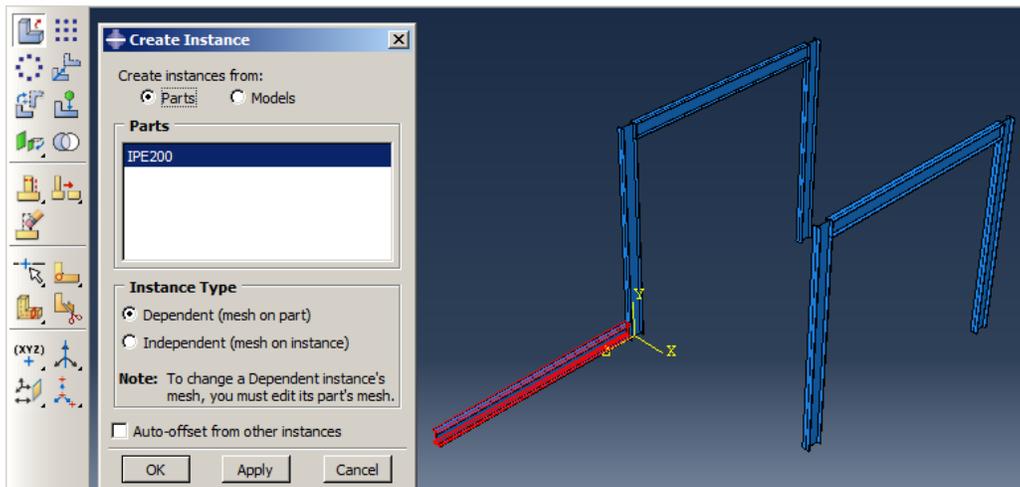
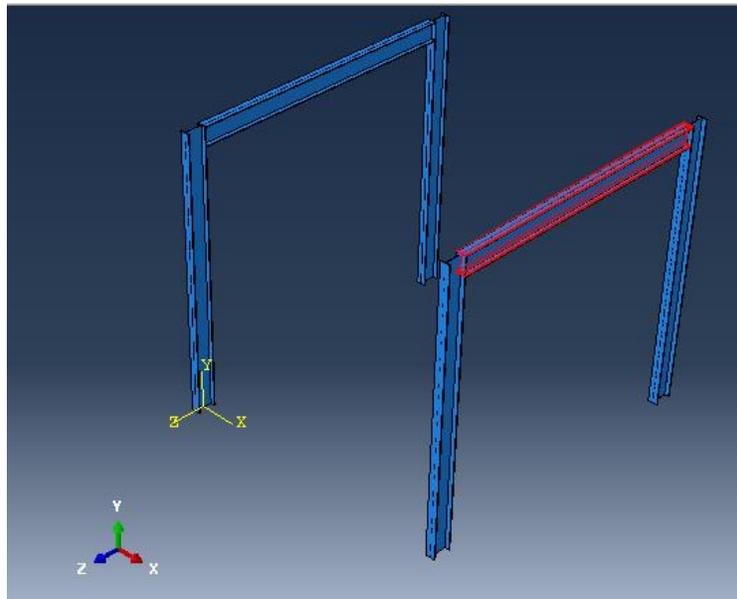
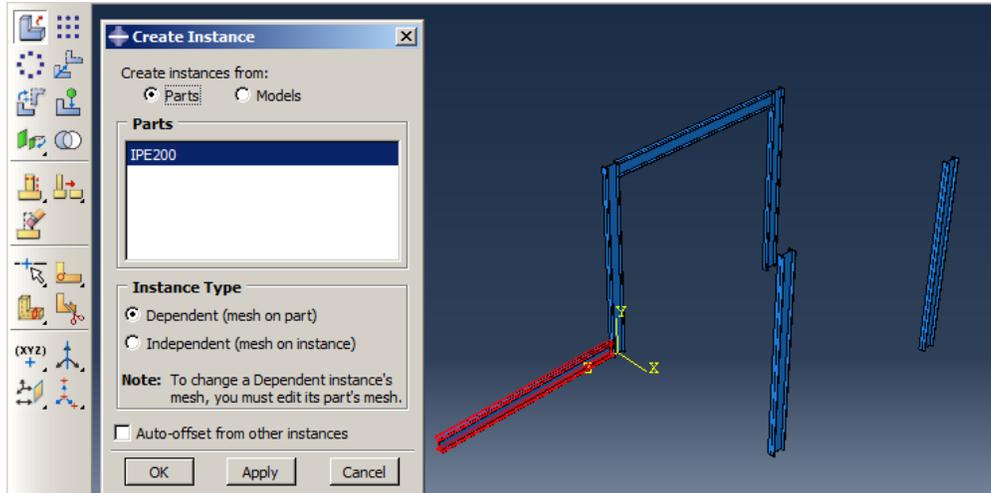
Figure 1-13: Defining a pattern of the instance as columns

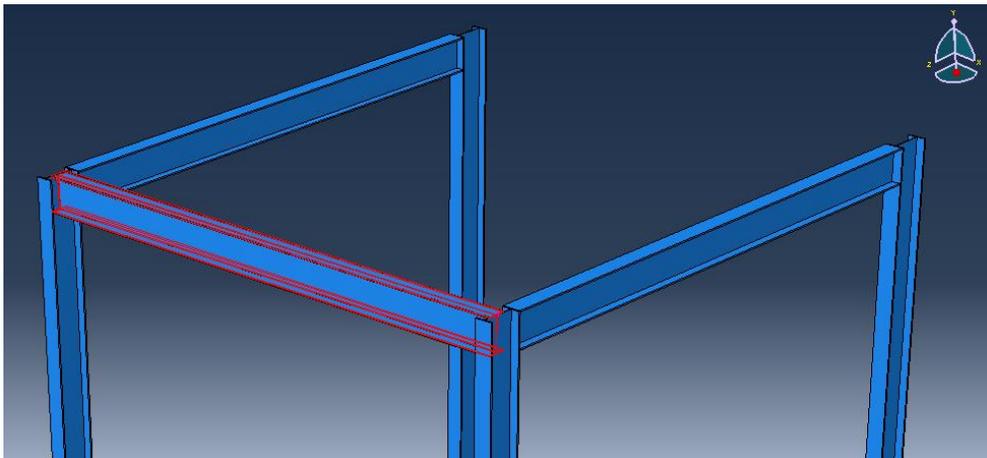
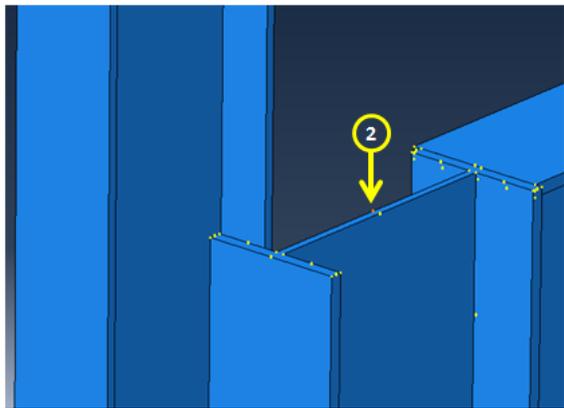
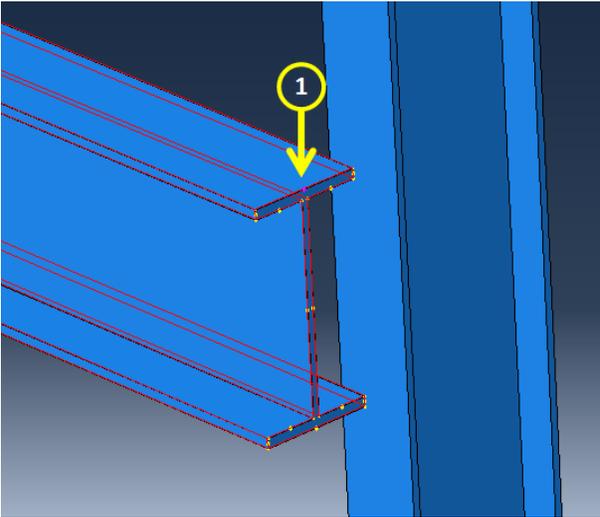
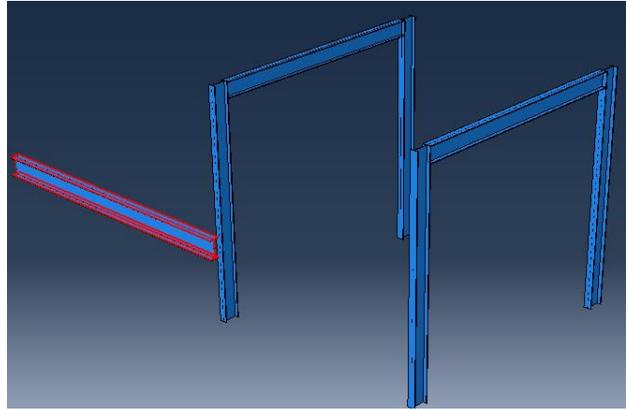
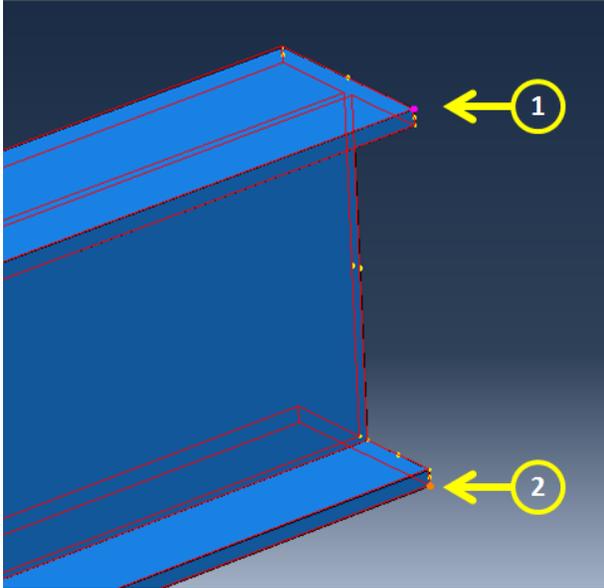
Double click on Instances and add a new instance into the assembly. Then select instance →Translate and choose the earlier instance and click the middle mouse button. Then select point, as shown in Figure 13 to reposition it properly. Do the same for another instance to the corresponding position.

Add a new instance and rotate it by 90 degrees, so that it is in the y-direction. Then translate it to the right position. Perform the same step for another beam (Figure 1-14).









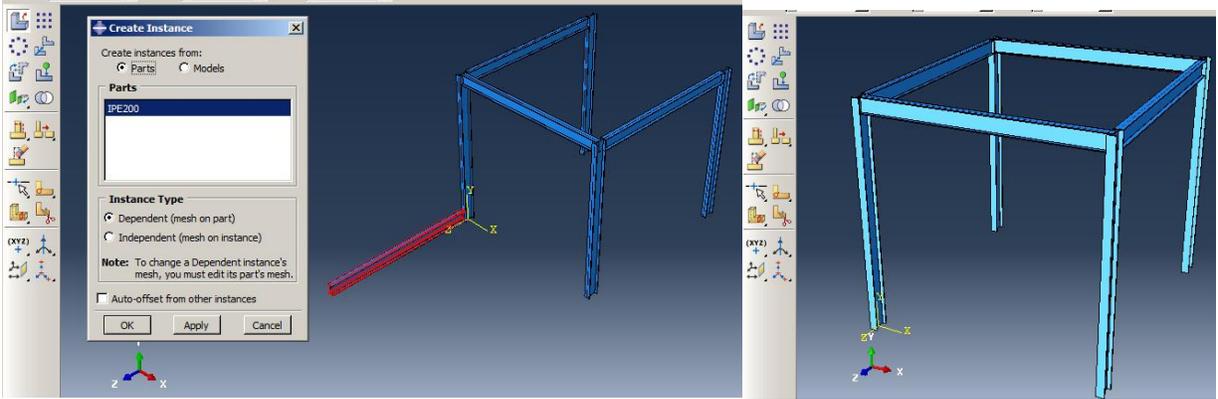


Figure 1-14: Adding and positioning new instances as beams

#### 1.4.4. Step Module

This case is an example of dynamic analysis given the many variations and the key role of acceleration as input data. ABAQUS software includes two methods for solving dynamic problems:

#### Implicit dynamics

In this method, as used in the ABAQUS/Standard solver software, the Newton-Raphson process, and simultaneous solving equations in each development are used to analyze the problem. The response in each increment is calculated based on the matrix methods and relies on the number of degrees of freedom. The required condition for solving the problem is to satisfy equilibrium equations in each increment. In fact, in this process, due to the nonlinear behavior of the structure, the load will be applied in an incremental process, so that in each increment, linear behavior of the structure is considered. The equilibrium equation for each increment and each degree of freedom is defined as given in Equation 4

$$F - f = m\ddot{x} \quad (4)$$

Where  $F$  is the external force of a node,  $f$  is an internal force,  $m$  is the mass of the node and  $\ddot{x}$  is the acceleration of the node. For example, for a model with 120 degrees of freedom (included 20 nodes, and 6 degrees of freedom per node), the equation of equilibrium will be formed. It is necessary to use matrix methods to solve these equations simultaneously and to calculate the variables.

Depending on the complexity and smoothness of nonlinear behaviors, the number of increments may either be high or low. In complex nonlinear problems, many increments need to be formed, and if the problem has a large matrix for analysis, this will result in long and expensive

computations. The speed of analysis of these calculations highly depends on the hardware resources. Although the exact solution of the equations involves high accuracy. The implicit method can also be used for static problems in which the right side of Equation 4 is considered to be zero. Linear issues are solved in one increment using this method. Given the approach of the implicit solution method for solving equations simultaneously, this method has been introduced as a powerful method for analyzing a wide range of problems.

### **Explicit Dynamics**

In this method, which is used in the ABAQUS/Explicit solver software, the equations are solved sequentially based on the central difference method, in which the values of the variables in time  $t + \Delta t$  are obtained based on the values at time  $t$ . In the explicit method, the central difference is conditionally stable, and this condition depends on stability time. Equations in this method form a diagonal matrix, due to the cost-effective equations to solve simultaneously, which are extremely fast. Also, in this method, the size of the elements will affect ability time. This is because the longitudinal decoupled matrix solving problems using the method does not depend on hardware resources. Nonlinearities as large displacements can also be implemented using this method, while no complexity in analysis.

As the example includes earthquake effects, the dynamic procedure should be defined. For this reason, a Dynamic procedure should be defined. However, the Implicit and Explicit method could be considered in the analysis. According to educational goals, the implicit method should be selected in the sample. Therefore, the sample should be defined according to “EL Centro” acceleration time history.

Double click on Steps in the Model Tree to open the Create Step dialog box and select Dynamic,Implicit and click Continue to open the Edit Step dialog box.

In the Basic tabbed page, change the time period to 30, in the acceleration time history.

In the Incrementation tabbed page, consider 0.01 for the initial increment size; 1e-5 for minimum increment size, specify 0.1 for the maximum increment size, and 10000 for the maximum number of increments. All other options in the dialog box should remain unchanged. Then click Ok to create the step and close the dialog box (Figure 1-15)

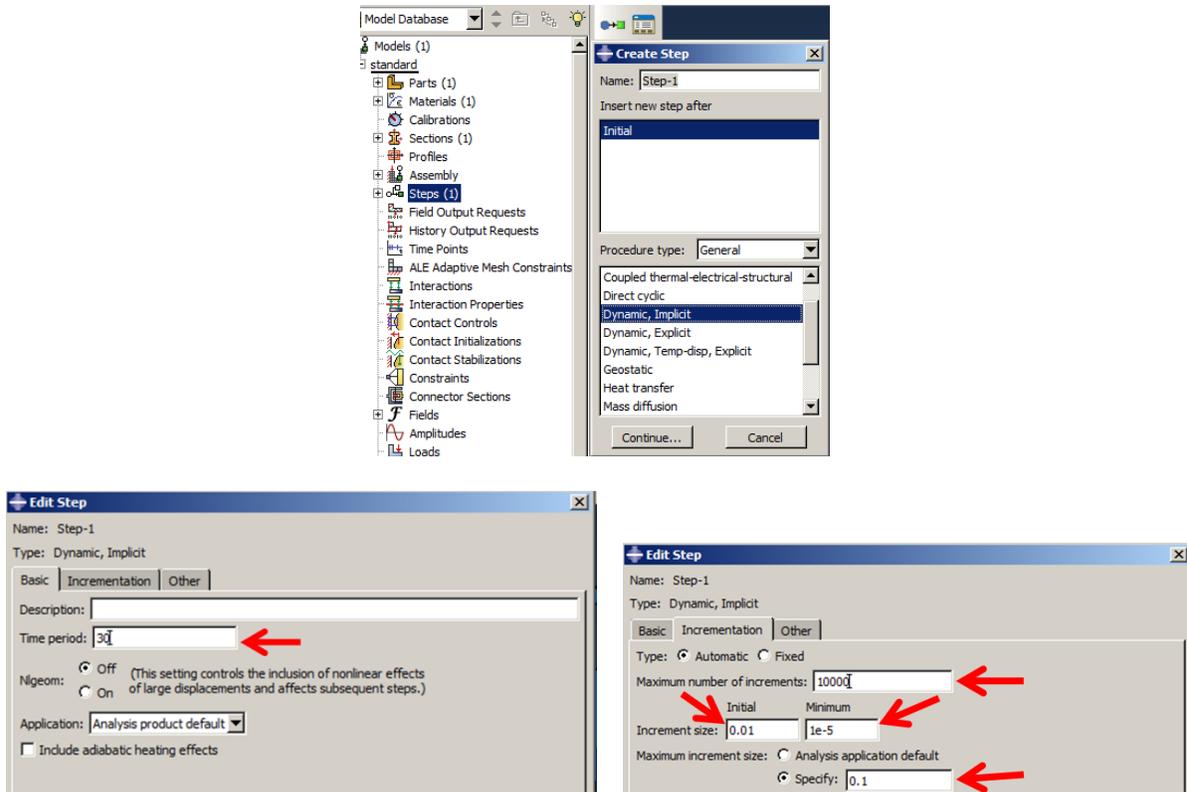


Figure 1-15: Defining the analysis step

### 1.4.5. Required Output

In the next step, the output variables should be defined in Field Output Requests. In the sample, preselected defaults variables are sufficient. However, they should be written properly to extract a proper diagram.

Double click on F-Output-1 underneath of Field Output Request in the Model Tree to open the Edit field Output Request dialog box. Change the output frequency to 300 intervals, and the output frequency should be changed, as shown in Figure 1-16. This will impact the results that will be achieved in 300-time points (every 0.1 seconds, one frame will be generated that contains the results). Click Ok to apply these changes, while all other options remained unchanged (Figure 1-16).

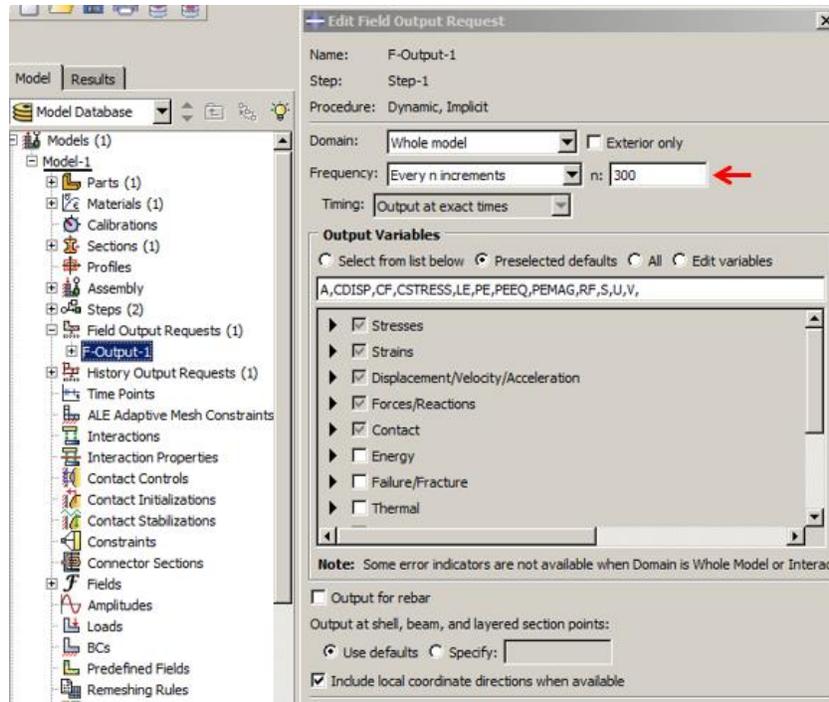


Figure 1-16: Changing the output frequency

#### 1.4.6. Constraint

In this example, all elements are assumed to be welded together. The best way to define clamping in ABAQUS is via the Tie constraint that is presented in the Interaction module. The Tie constraint contains pair surfaces (whether Node-surface or Element surface) as the Master and Slave. The master surface is a surface that can control and affect another one (slave) surface. In other words, the master surface is a more stiffened surface (that usually includes more Young's modulus) than a slave surface. Although, in some models (such as in this example), the master and slave material are the same, and there is no difference between the Master surface and the Slave surface. For this example, all beam cross-sections are assumed as Master, and all column surfaces (that interact with beams) are considered as Slave.

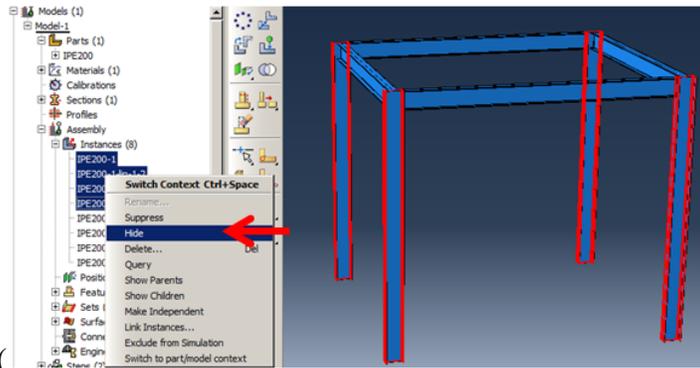
Double click on Constraint in the Model Tree to open the Create Constraint dialog box and select Tie as the type of constraint.

Hide all columns by right-clicking on them in the Model Tree and select "Hide". Then, click the middle mouse button and select Surface as the Master type. Hold the shift key on the keyboard and select the beams cross sections and finally click the middle mouse button and the Master surfaces will be colored red.

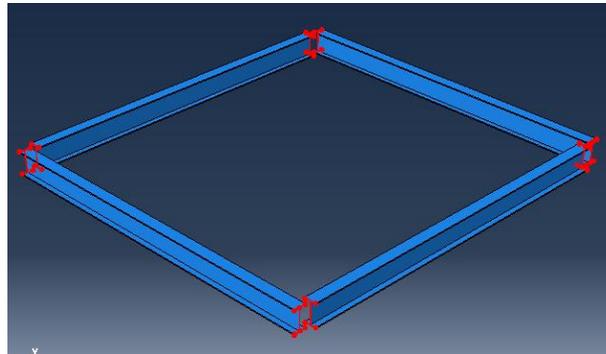
Show the columns by right-clicking on them and hide the beams by right clicking and select "Hide". Then, click the middle mouse button and after choosing Surface as the Slave type, select

corresponding wings and flanges of columns while holding down the shift key. The Slave surfaces will be colored purple.

Click the middle mouse button to open the Edit Constraint dialog box. As a default, ABAQUS uses 5 percent of the element length to constrain the parts that seem suitable for the example. The nodes that are generated in the slave surfaces should be placed in the initial position at the beginning of the analysis. In this example, because of clamping columns and beams, rotational DOF in addition to translational DOF should be a constraint. Therefore, any changes in the dialog box are not required.



Click Ok to define the constraint (



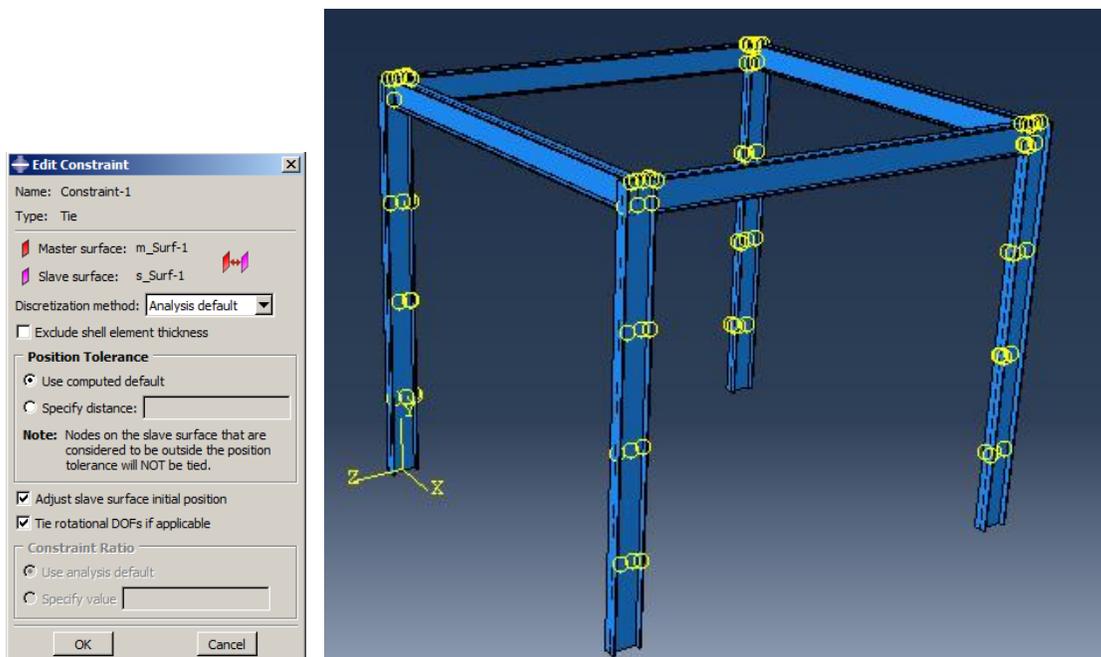
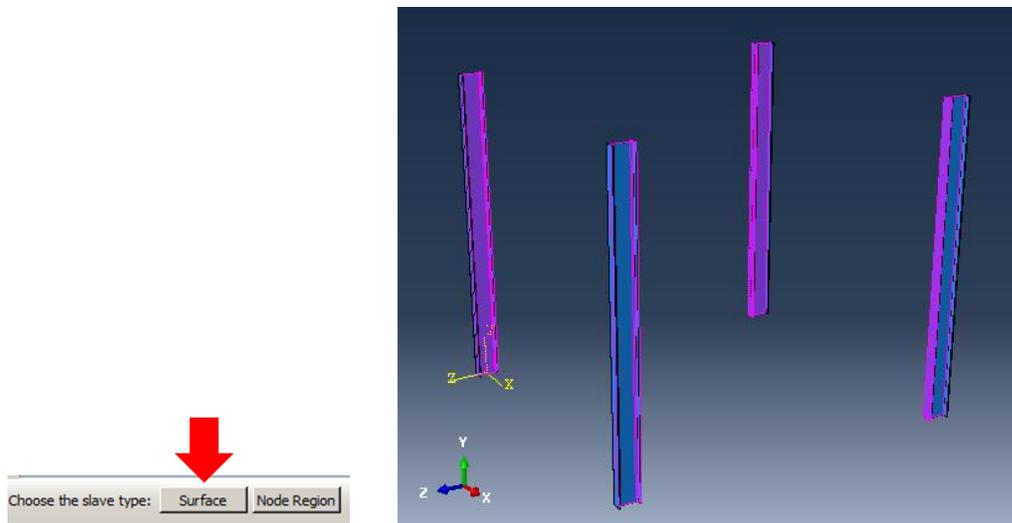
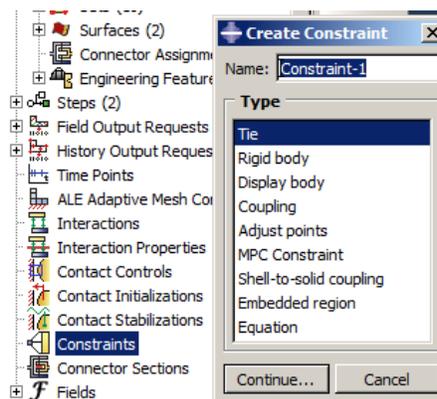
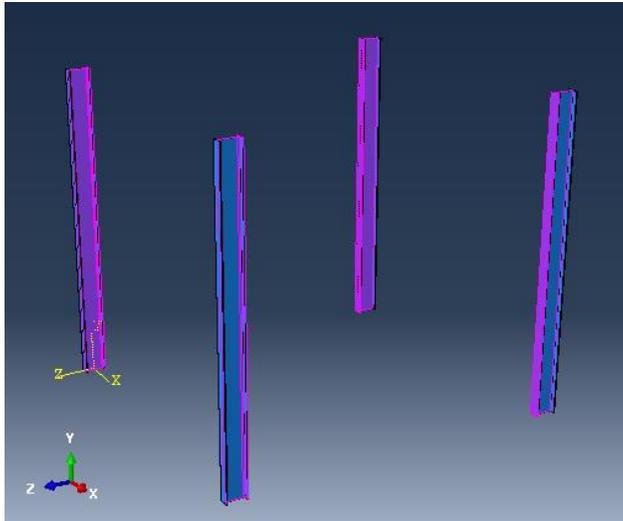
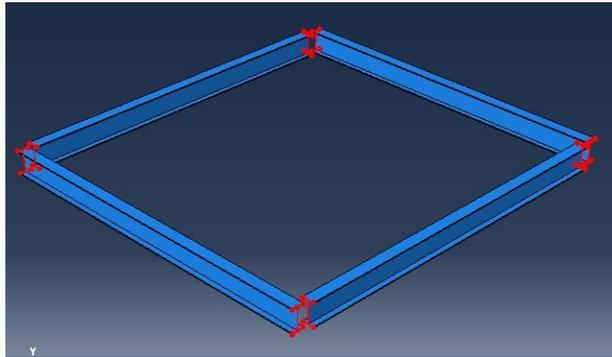
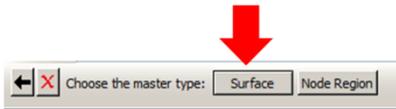
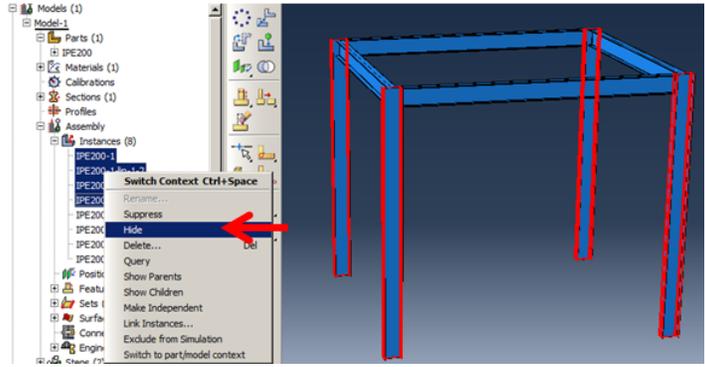


Figure 1-17).





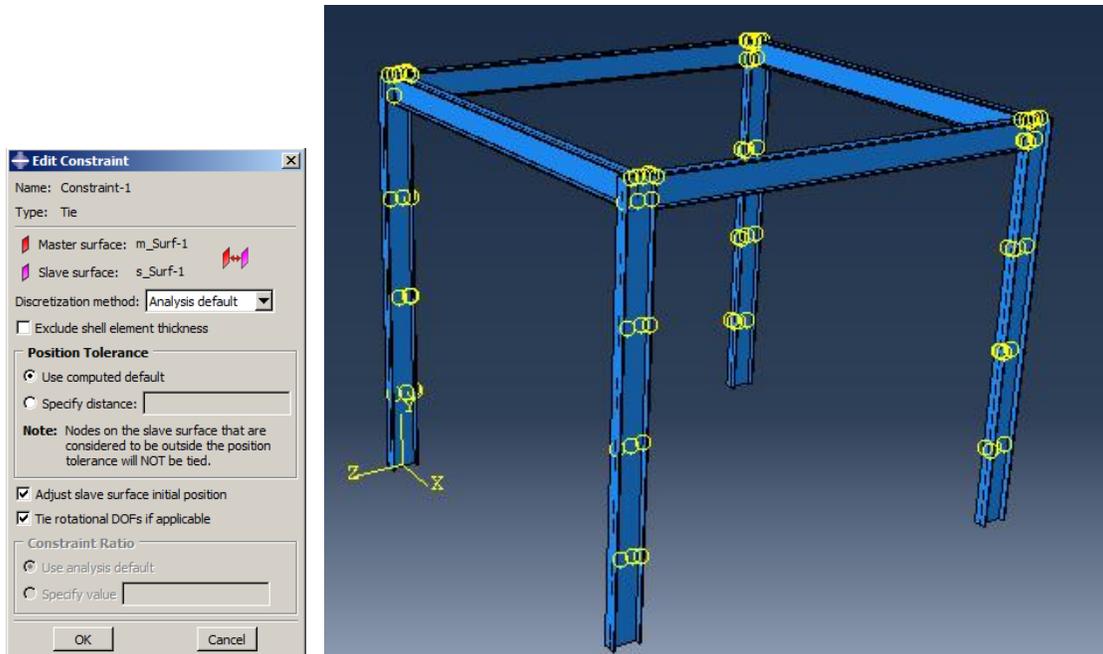


Figure 1-17: Defining the tie constraint

### 1.4.7. Load/Boundary Condition Module

At this point, boundary conditions should be determined. In the sample, two boundary conditions should be defined. The first one is about limiting displacement of the basis of the frame (all four columns) that move along the x-direction, and another one is the acceleration time history that is considered along the x-direction. The earthquake acceleration time history should be defined as amplitude that the acceleration boundary condition refers to.

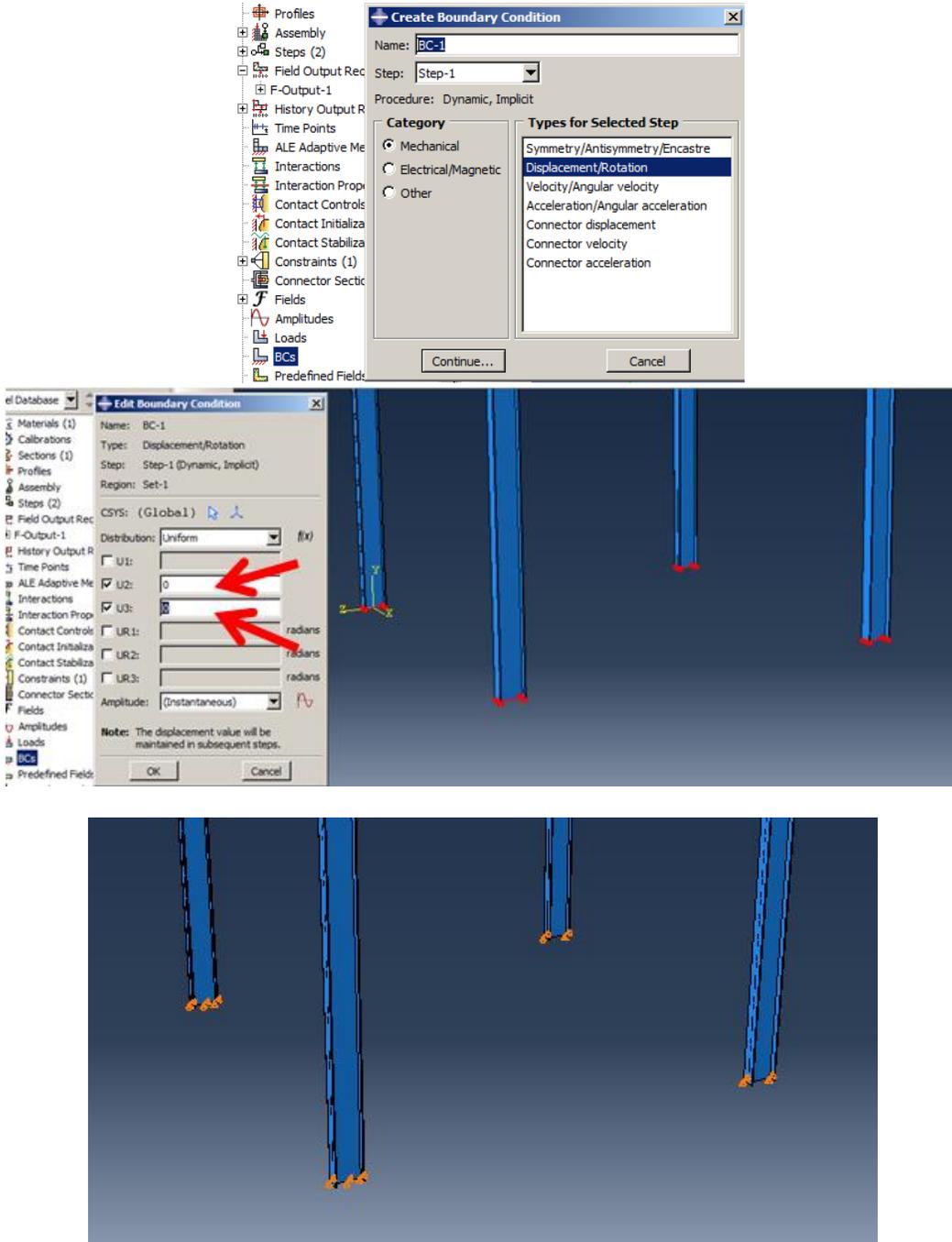
Double click on BCs and open the Create Boundary Condition dialog box. Then, select Displacement/Rotation as the type and click Continue.

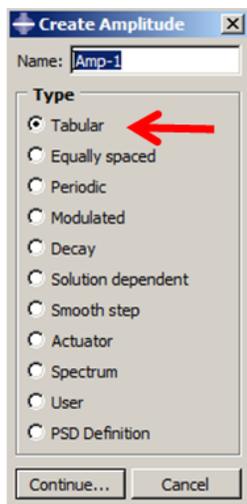
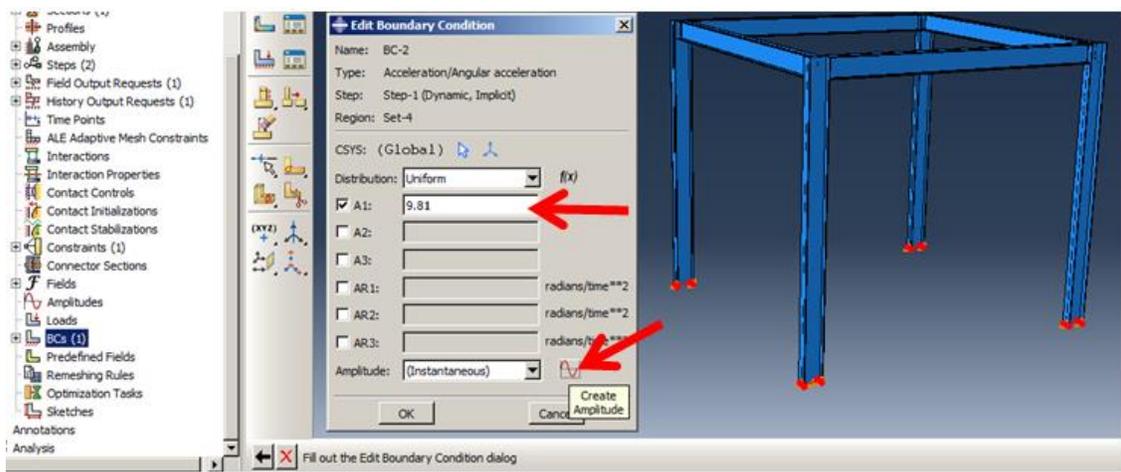
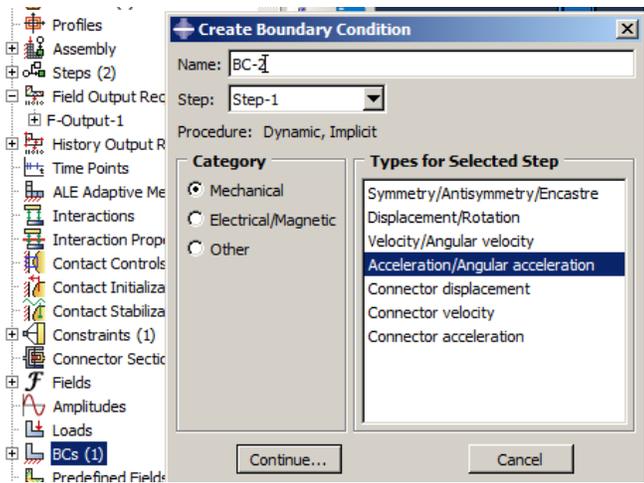
Choose all four column bottom faces and click the middle mouse button to open the Edit Boundary Condition dialog box. Check U2 and U3 as the y and z directions, where the frame will not move along them, respectively, and click Ok to apply and close the dialog box.

Again, double click on BCs and select Acceleration/Angular acceleration as the boundary condition type and select Continue. Select all four column bottom faces and click the middle mouse button to open the Edit Boundary Condition dialog box. Check A1 and consider 9.81 as a magnitude.

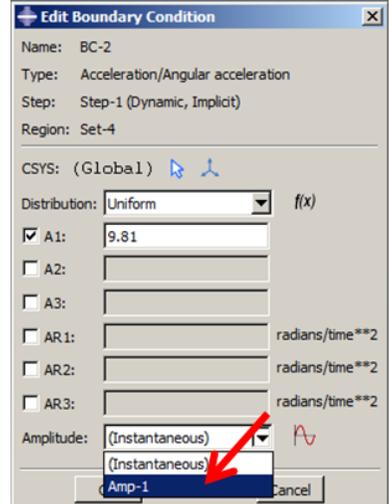
Click the Create Amplitude icon to open the Create Amplitude dialog box. Click Continue and input time/acceleration data provided in the attached \*.xlsx file. This could be copied. Click Ok to create the amplitude and close the dialog box.

Select Amp-1 in Edit Boundary Condition and click Ok to apply the boundary condition and close the dialog box (Figure 1-18).





	Time/Frequency	Amplitude
1	0	0.061803
2	0.02	0.0357084
3	0.04	0.0097119
4	0.06	0.0419868
5	0.08	0.0743598
6	0.1	0.1066347
7	0.12	0.0669042
8	0.14	0.0271737
9	0.16	-0.0125568
10	0.18	0.0361008
11	0.2	0.0847584
12	0.22	0.133416



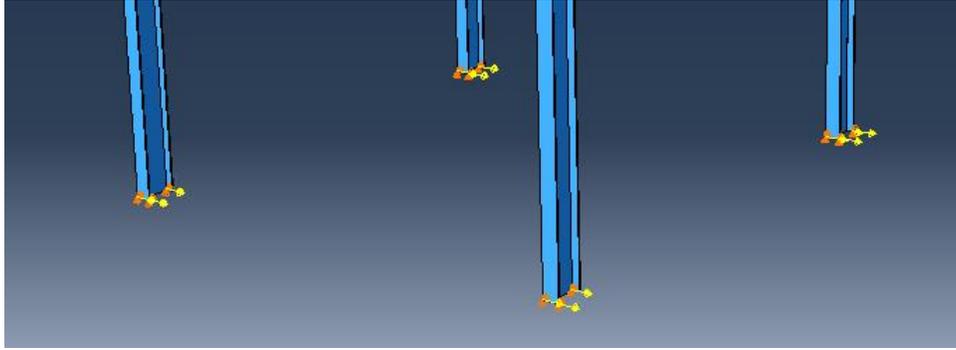


Figure 1-18: Defining the boundary conditions

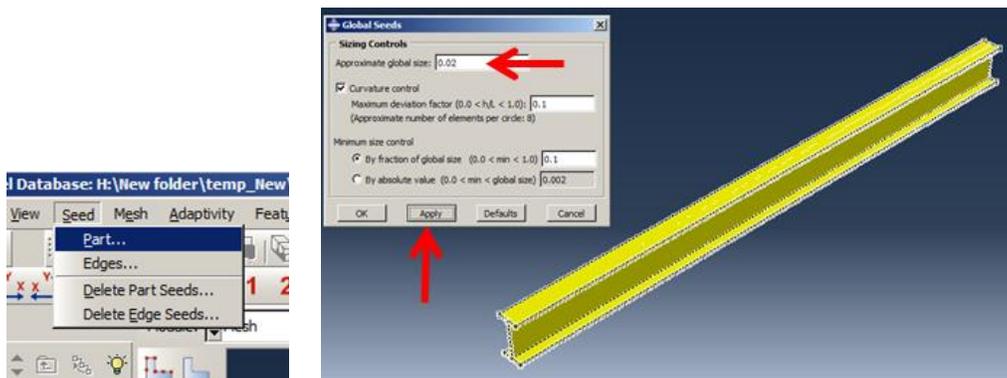
### 1.4.8. Mesh Module

The next step is about the meshing, which includes specifying the element size and generating the mesh.

Double click on Mesh (Empty) underneath of IPE200 in the Model Tree to enter the Mesh module.

Select Seed → Part to open the Global Seeds dialog box. Enter 0.02 as the Approximate Global Size and click Apply to check the approximate position for the nodes and then click Ok to close the dialog box.

Select Mesh → Part and click Yes in the prompt area for mesh generation (Figure 1-19).



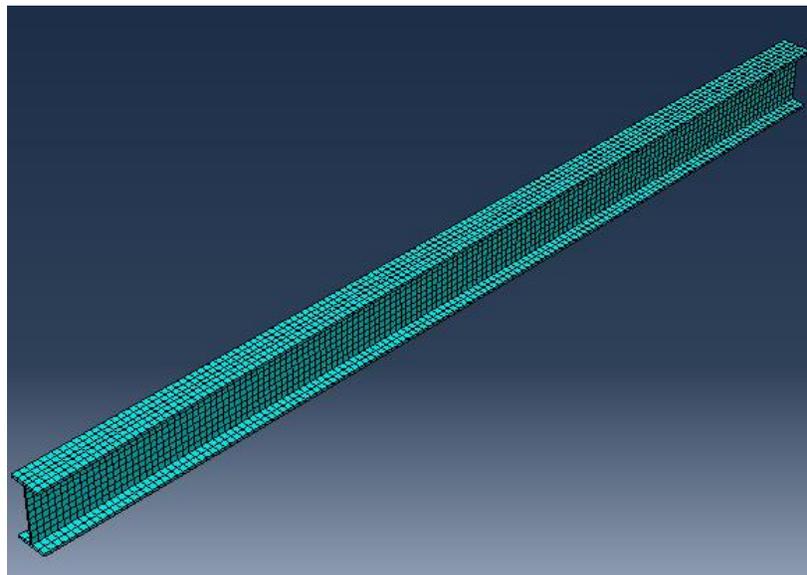
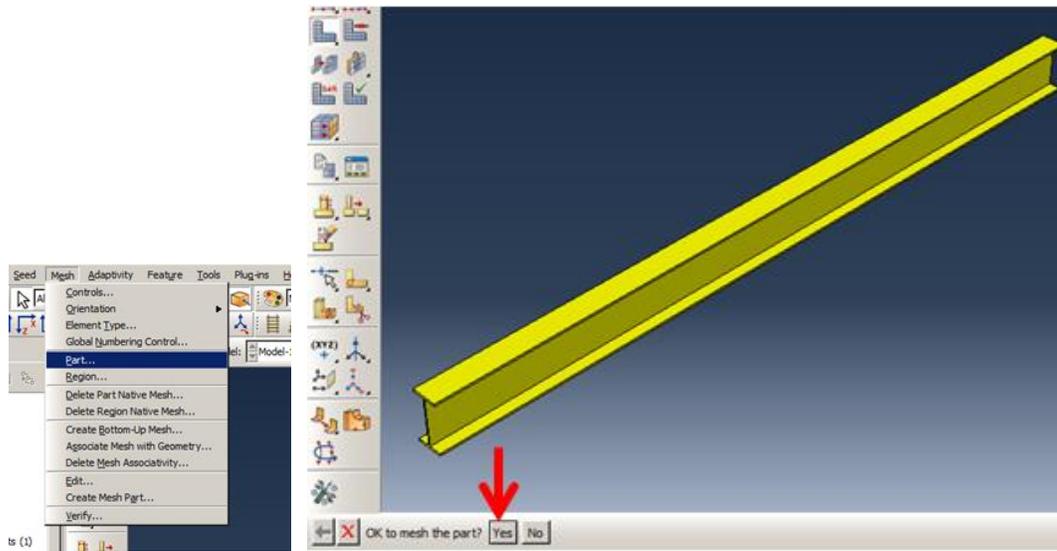


Figure 1-19: Mesh generation

### 1.5. Analysis: Job Module

Then, the job should be defined. The job contains the UMAT subroutine that will be implemented in the procedure and computing material behavior.

Double click on Jobs in the Model Tree to open the Create Job dialog box and name the job. Then, click Continue to open the Edit Job dialog box.

In the tabbed page, browse the subroutine file that was saved as UMAT.for and click Ok to define the job and close the dialog box.

Right click on the job underneath Jobs in the Model Tree and select Submit to begin the analysis (Figure 1-20).

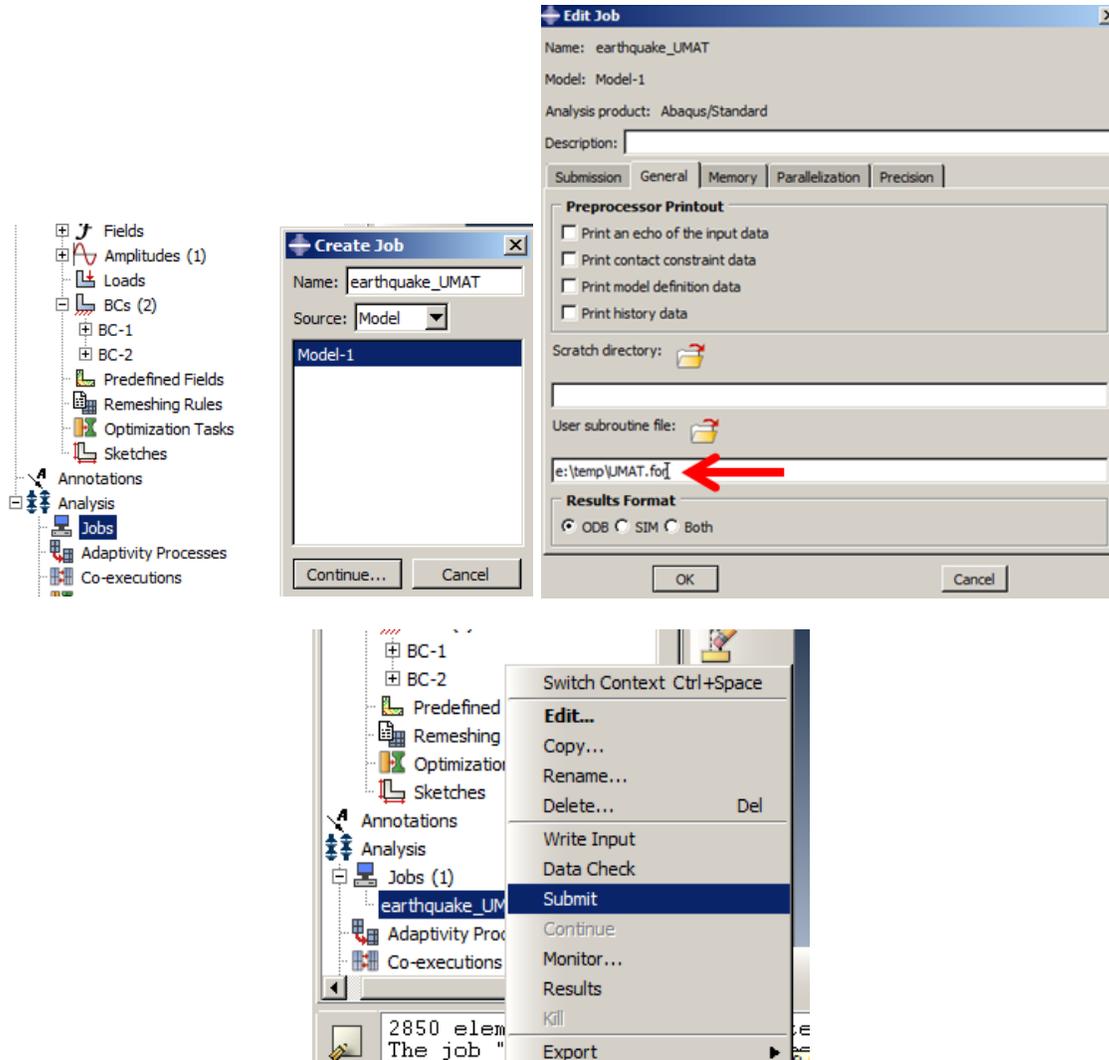


Figure 1-20: Defining the job

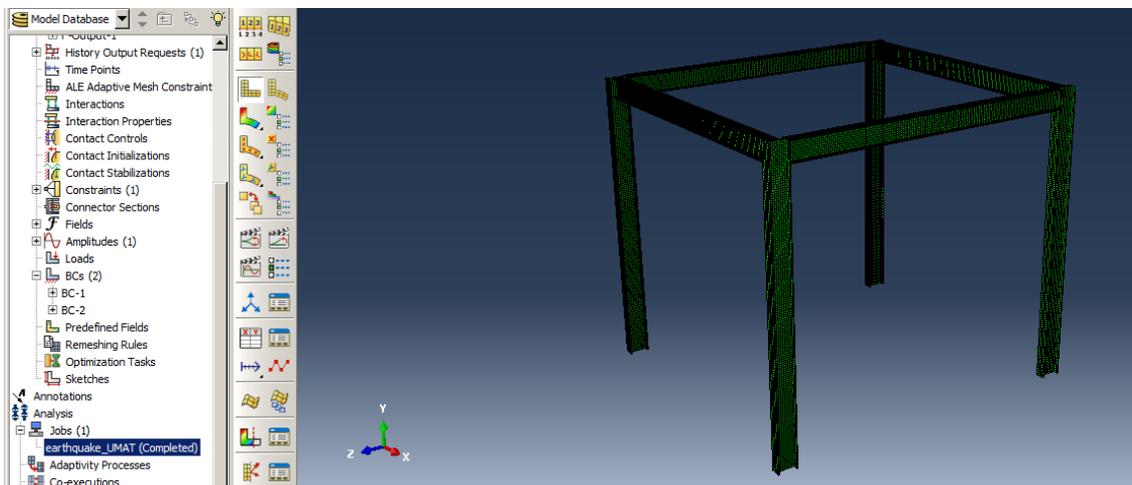
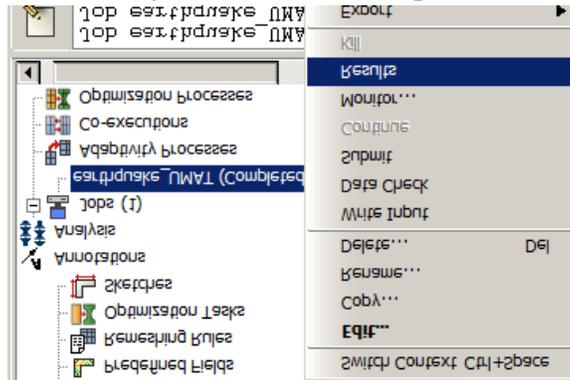
## 1.6. Analysis Results

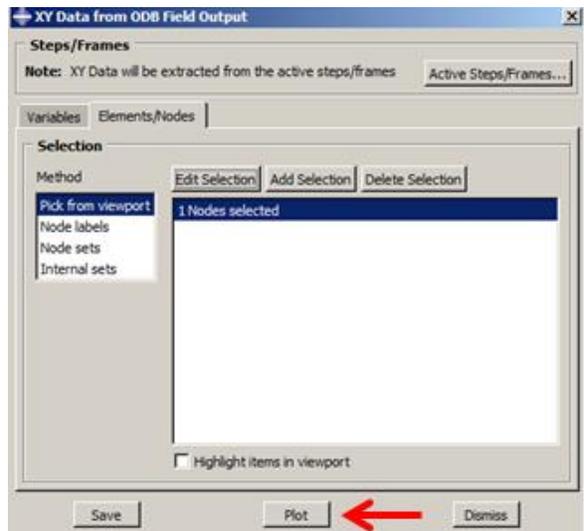
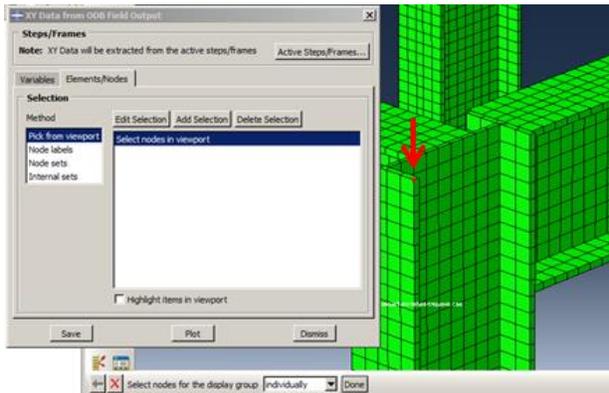
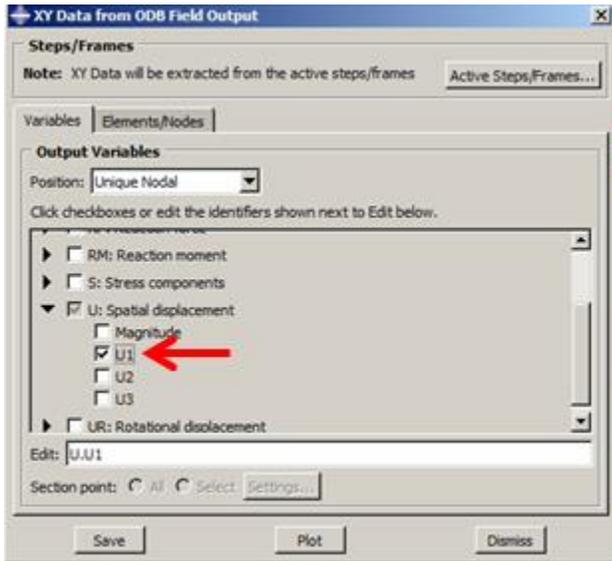
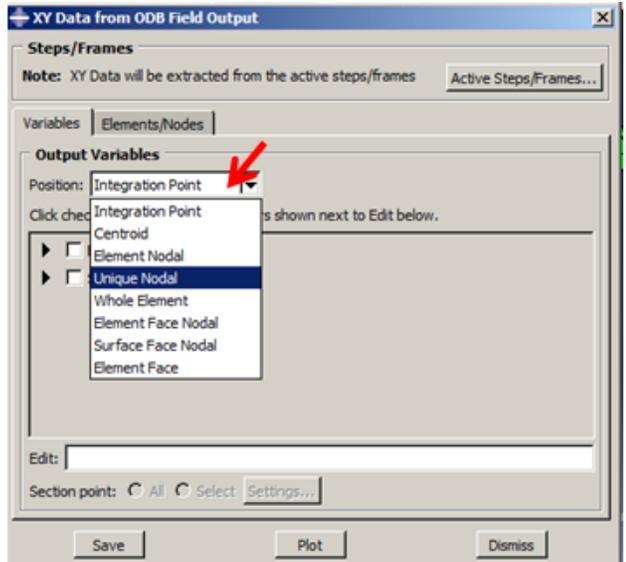
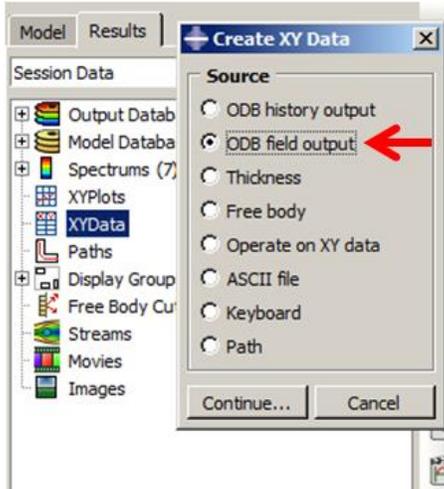
After running the analysis, the results can be extracted. The results are saved in \*.Odb files and can be checked in the Visualization module. In the sample, displacement is top of the frame as a time history diagram, and Von Mises stress in the frame will be investigated.

Right click on the completed job and select Results to enter the Visualization module.

Double click on XYData to open the Create XY Data box. Select ODB Field Output and click Continue to open XYData from the ODB Field Output dialog box. In the Variables tabbed page, select Unique Nodal as the Position of the Output Variables and choose U1.

Then, in the Elements/Nodes tabbed page, click Edit Selection and select a top vertex and click the middle mouse button to note “1 Node selected” in the box, then click Plot to plot displacement time history. The maximum displacement is about 20cm (





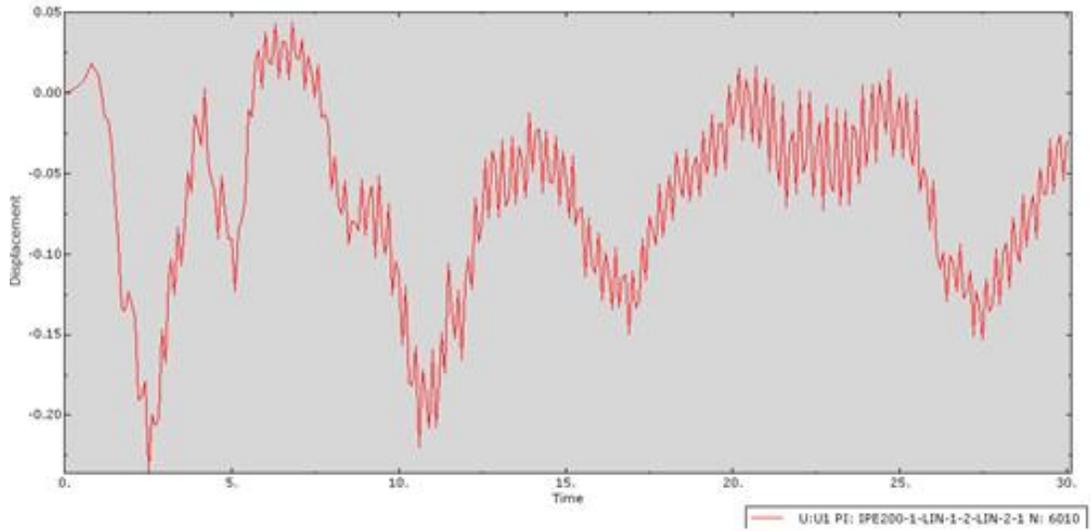
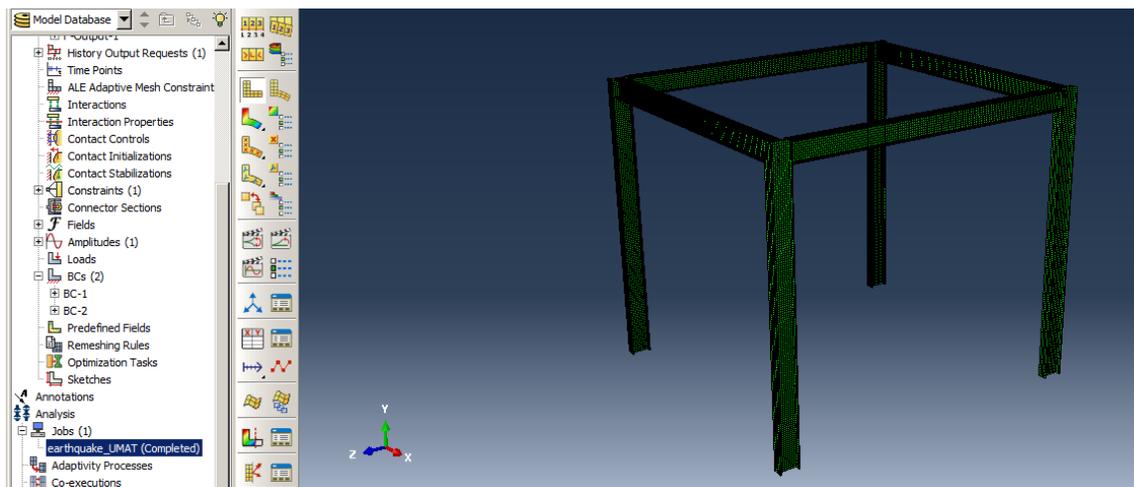
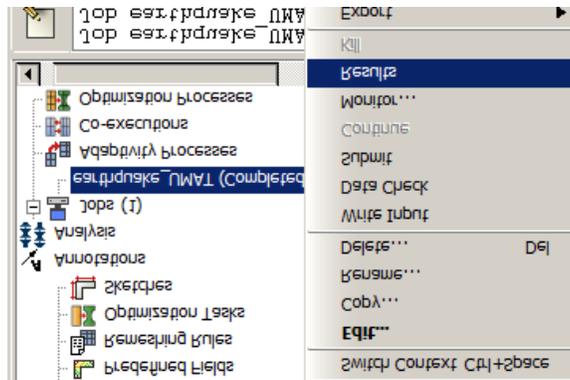
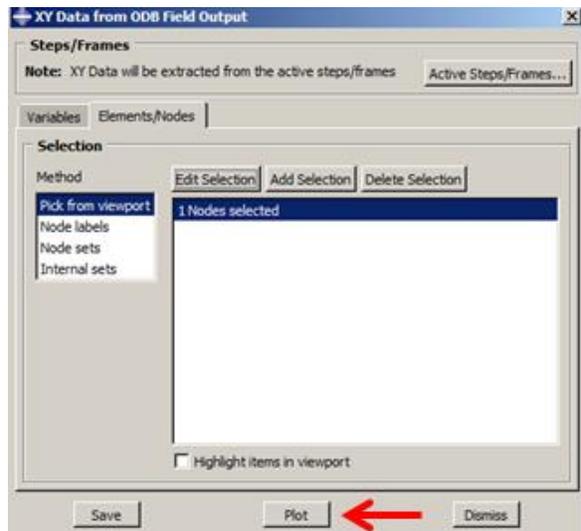
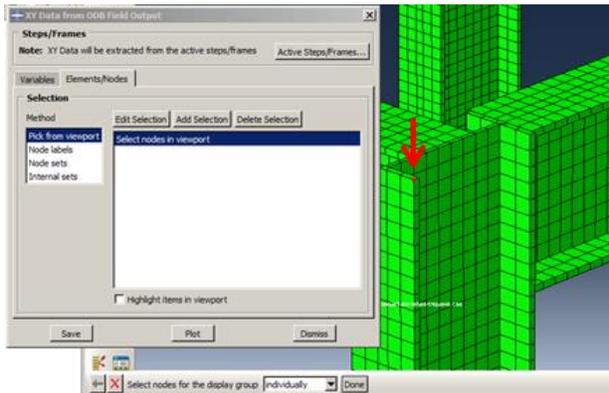
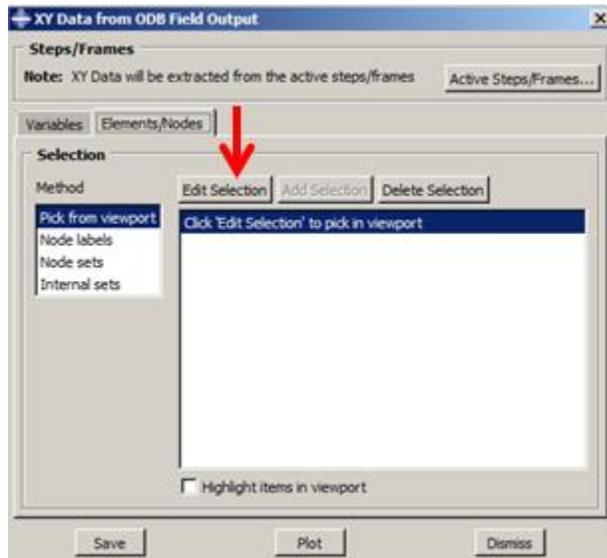
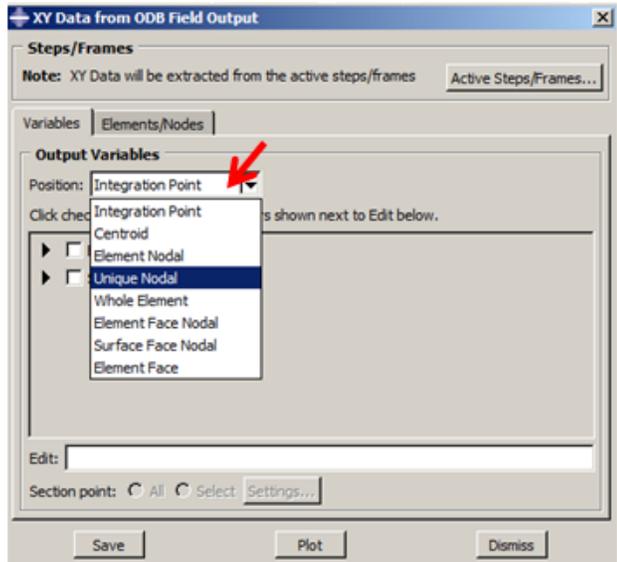
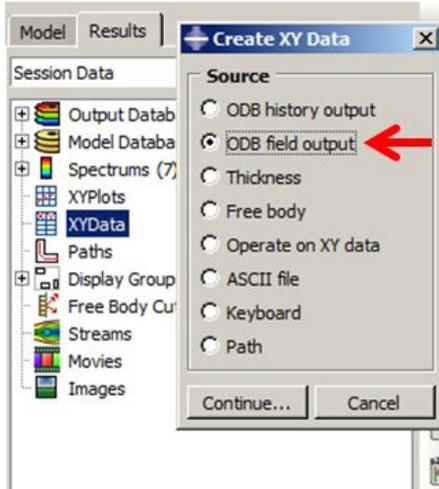


Figure 1-21).





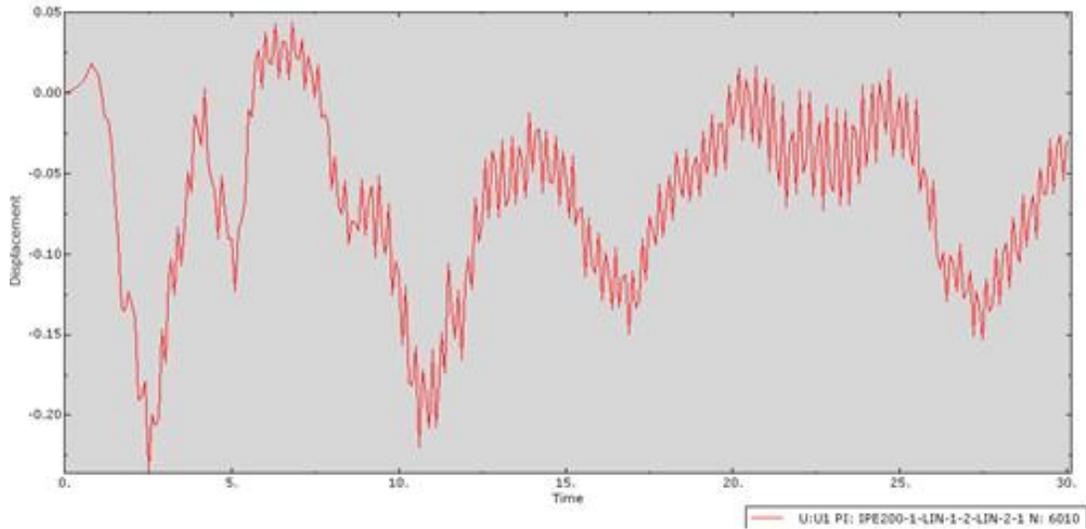
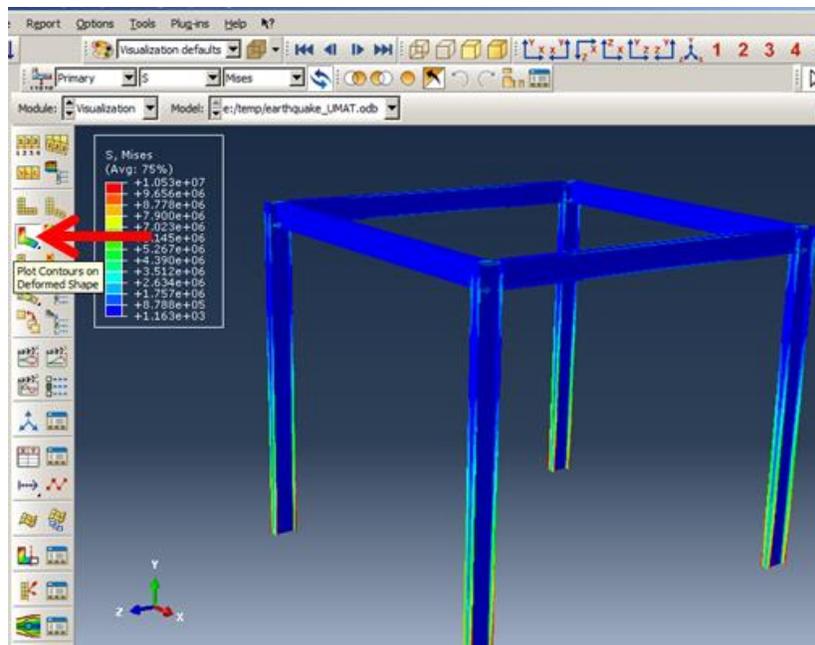


Figure 1-21: Plot displacement time history

Click Plot Contours on the Deformed Shape icon to extract the Von Mises stress contour. The maximum Von Mises stress at the end of the earthquake is about 10.5MPa around the bottom of the columns.

Click Animate Time History to check the maximum Von Mises stress during the earthquake. The maximum is about 118.5 MPa (Figure 1-22).



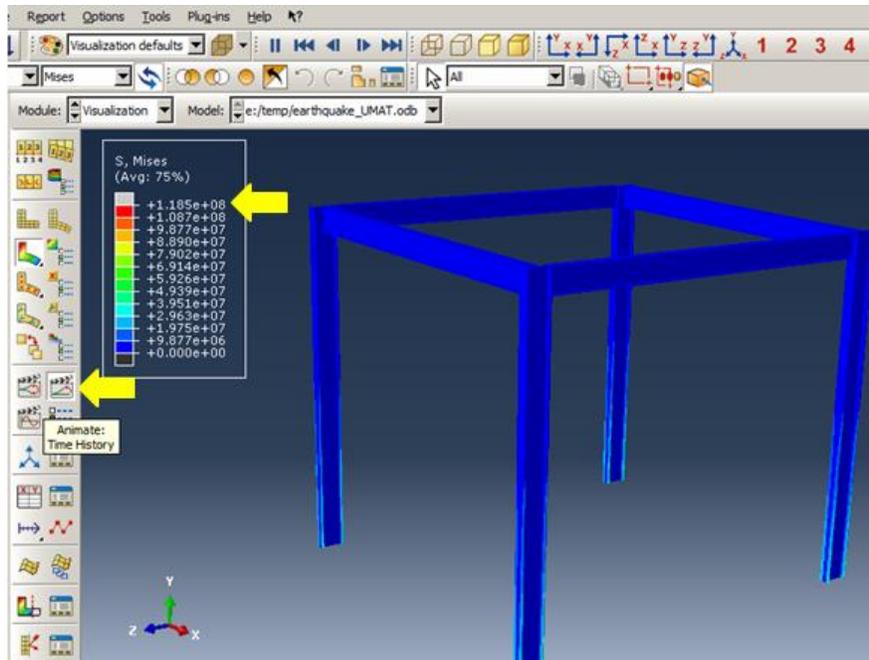


Figure 1-22: Extracting Von Mises stress contour at the end and during an earthquake