Bangladesh University of Engineering and Technology

DEPARTMENT OF CIVIL ENGINEERING

Term Project

Evaluation of the temperature of fire-exposed columns using a simplified method and finite element approach

Course Teacher:
Dr. Ishtiaque Ahmed
and
Dr. Tanvir Manzur

Submitted by: Md. Mehedi Hassan Bhuiyan 0421042311

November 15, 2022



Abstract

This term project compares two simplified models for figuring out the temperature of reinforcements with a tested finite element method. A total of 112 models are created in this study using two different cross-sectional columns; each column has two different effective cover (50 mm and 75 mm) specimens that are subjected to different durations (30, 60, 90, 120, 150, 180, and 240 minutes) of standard fire tests in a finite element environment using a verified method. According to the findings of this study, simplified methods and FE outputs had an average 20% error in temperature evaluation for 60- to 240-min fire-exposed models. Furthermore, when those columns are exposed to fire on two, three, or all of their faces, the temperature in the reinforcement calculated using the simplified method has a margin of error of less than 15% for the 90-240 minute range duration. But when column samples are exposed to a 30-minute fire, both simplified methods show a wider range of percentage errors. There was no difference in temperature between the simplified and FEA ways of reinforcing square or rectangular columns. This is because the location of the reinforcement was the same in both types of columns because the same effective cover was used. Both simplified methods can be used for reinforcement temperature evaluation purposes for structures exposed to fire for more than 60 minutes, but for better output, proposed verified FEA approaches may be adapted.

Contents

Al	strac	t	i
	1	Introduction	1
	2	Thermal Analysis	3
		2.1 Thermal Property	3
		2.2 FE Modeling Approach	3
	3	FE Model Validation	4
	4	Parametric Study	4
	5	Conclusion	9
References			10
A	Aba	qus Scripting with python	11

List of Figures

1	Separation of concrete member into areas for use in determining internal	
	temperatures	2
2	Thermal Property Conductivity as per EC-2. [1]	3
3	Fire-exposed finite element model	4
4	Dimensions and reinforcement details of RC columns	5
5	Comparison of predicted (FEA) and measured column temperatures during	
	fire exposure. [2]	5
6	Selected specimens for study (a) one exposed side (b) two exposed sides	
	(c) three exposed sides (d) four exposed sides to ASTM E119 [3] standard	
	fire	6
7	The temperature of Rebar exposed to fire on 1 side for different durations .	7
8	The temperature of Rebar exposed to fire on 2 side for different durations .	7
9	The temperature of Rebar exposed to fire on 3 side for different durations .	8
10	The temperature of Rebar exposed to fire on 3 side for different durations.	8

1 Introduction

A fire hazard is an unwanted event in this world. Any structure can be affected by this event, but specially reinforced concrete structures may be less affected than other materials because the material properties of the outer core concrete degrade slower than the inner steel reinforcement [4]. A few research groups have come up with a simplified way to figure out the residual capacity of RC beams [5, 6] and RC columns [7] after a fire event, and both of these simplified methods need to know the temperature of the reinforcements. It is hard to get a more precise look at the temperature of the inner core reinforcement. Most of the time, engineers and their concerns depend on firefighter reports to find out the temperature of this structure. In reality, fire fighters need some time to reach the fire event place, and most of the time it is not possible to delineate the real event scenario in a report. Assuming that firefighters report actual fire event durations and peak temperatures on their reports, it is time-consuming and costly to determine the actual fire temperature of inner reinforcement by advance machinery or finite element approaches. To overcome this problem, there are many simplified methods available to determine the inner core reinforcement temperature that are easy, less time-consuming, and less costly. Among them, we discuss two simplified methods that are proposed by Wickstrom [8] and Kodur [9].

Wickstrom [8] introduces two methods for determining the temperature of structures: one for one-dimensional heat transfer and another for two-dimensional heat transfer. The author proposes equations (1)-(3) to evaluate temperature for 1D heat transfer purposes, and for 2D heat transfer purposes, equations (1), (3) and (4) are used. In equations (1)-(3) η_w is the ratio of the temperature at the surface of the concrete to the temperature of the fire, η_z is the heat transfer factor induced through a single surface that is exposed to the fire, and T_f is the temperature of the fire. The heat conduction that takes place in two different directions (z and y) needs to be accounted for in order to obtain temperatures at corner locations of a structural member. This is done through the use of η_z and η_y , with η_y being calculated in the same manner as η_z in the equation. (2). The temperature Tc that is reached after being exposed to fire in both the x and y directions is as follows. Where η_z and η_y represent the heat transfer factors that are caused by exposure on either side of the fire. Due to the fact that it does not take into account the various rates at which temperatures can increase during a fire [10], Wickstrom's empirical equation can only provide a rough estimate of the cross-sectional temperatures. Additionally, the influence of aggregate type (siliceous or carbonate), newer concrete types (high strength concrete), and the variation of thermal properties with temperature are not accounted for in this equation. As a result, it is possible that this equation is not applicable to various types of concrete.

$$T_c = \eta_z \, \eta_w \, T_f \tag{1}$$

$$\eta_z = 0.18ln\left(\frac{t_h}{z^2}\right) - 0.81\tag{2}$$

$$\eta_w = 1 - 0.0616t_h^{-0.88} \tag{3}$$

$$T_c = \left[\eta_w \left(\eta_z + \eta_y - 2\eta_z \eta_y \right) + \eta_z \eta_y \right] T_f \tag{4}$$

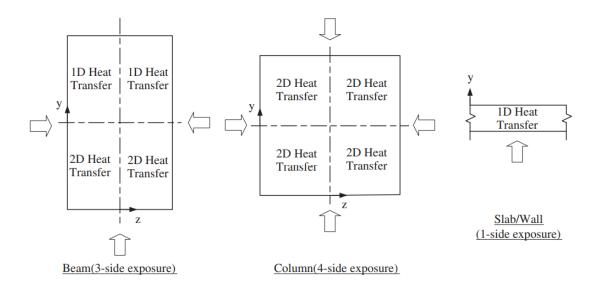


Figure 1: Separation of concrete member into areas for use in determining internal temperatures.

By taking aggregate type into account, Kodur [9] overcomes the gap in Wickstrom's proposed method. Kodur approaches consider aggregate types like carbonate aggregate (CA) and silicious aggregate (SA), as well as types of concrete such as normal-strength concrete (NSC) and high-strength concrete (HSC). Following Wickstrom, Kodur also provides two methods for considering 1D and 2D heat transfer. Kodur proposes equations (5) and (6) to evaluate temperature for 1D heat transfer purposes, and for 2D heat transfer purposes, equations (6) and (7) are used.

$$T_c = c_1 \eta_z (at^n) \tag{5}$$

$$\eta_z = 0.155 ln\left(\frac{t}{z^{1.5}}\right) - 0.348\sqrt{z} - 0.371\tag{6}$$

$$T_c = c_2 \left[-1.481 \left(\eta_z \eta_y \right) + 0.985 \left(\eta_z + \eta_y \right) + 0.017 \right] (at^n)$$
 (7)

For ISO 834 [11] fire, a=935 and n=0.168, and for ASTM E119 [3] fire, a=910 and n=0.148 and c_1 are 1.0, 1.01, 1.12 and 1.12 for NSC-CA, HSC-CA, NSC-SA and HSC-SA, respectively; c_2 are 1.0, 1.06, 1.12 and 1.20 for NSC-CA, HSC-CA, NSC-SA and HSC-SA, respectively. The 1D and 2D heat transfer and coordinate systems are depicted in the y and z directions, respectively, in Figure 1.

The two simplified methods mentioned above are widely used in the temperature calculation of the Reinforce structure element. As mentioned before, rebar temperature plays an important role in estimating residual capacity. So, the goal of this term's project is to compare how accurately these two methods measure the temperature of the rebar to the verified FE computational method.

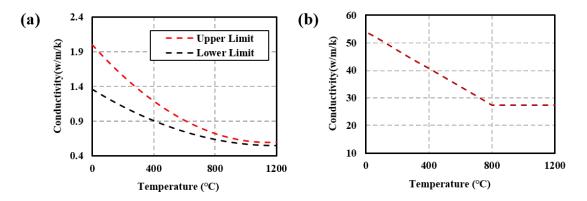


Figure 2: Thermal Property Conductivity as per EC-2. [1]

2 Thermal Analysis

A computational FE approach is required to evaluate the temperature of the rebar in a fire-exposed column. To do computational thermal analysis and create an environment similar to a fire event, the model needs to include properties of concrete and rebar that change with temperature. Material properties and FE modeling approaches are described in detail in the sections that follow.

2.1 Thermal Property

Eurocode 2 [1] properties are widely used for FEA verification purposes. Kodur [7, 6, 2], Bhuiyan [5], and other research groups also use Eurocode 2 properties for verification in FE analysis. For thermal analysis of concrete and rebar density, specific heat and conductivity are required, which are followed as per Eurocode 2. As per the code specified, normal-strength concrete density is 2300 kg/m^3 , and steel reinforcement density is 7850 kg/m^3 as used in the FE model. Also, temperature-dependent specific heat and conductivity are implemented in this model as well. The thermal conductivity of both concrete and steel is presented in Figure 2. The code specifies that any reasonable concrete thermal conductivity may be implemented in between the upper and lower bound ranges in the FE model. A research group verified their experimental result with the FE model and found a better result when lower bound thermal conductivity was implemented as a material property of the concrete element. Also, for a fire-exposed specimen with an exposed surface, the convective heat transfer coefficient is taken to be 25 W/m^2k [1] and W/m^2k for an unexposed surface [12]. The emissivity for radiative heat transfer at the exposed surfaces of the concrete member is taken as 0.8, Stefan-Boltzmann constant $5.67 \times 10^{-8} \ W/m^2 k^4$ and absolute zero temperature was taken -273.16 K. As mentioned, properties were also incorporated into this study for better verification purposes.

2.2 FE Modeling Approach

To make sure the concrete and rebar are connected, a few segments are made and tied together. This helps the heat move from the concrete to the rebar. For heat transfer modeling

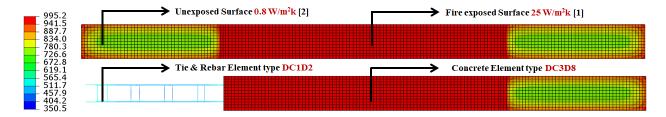


Figure 3: Fire-exposed finite element model

purposes, the FE package Abaqus was used in this study, with a concrete element using DC3D8 and a rebar element using DC1D2. Finally, the concrete and rebar are applied to a 25-mm mesh. To evaluate the concrete and rebar temperatures, a few sets of nodes were created in this model, which work as a thermocouple to read the specific location temperature over time. Figure 3 depicts a post-analysis contour diagram, which also shows element type and exposed and unexposed surface conditions.

3 FE Model Validation

An experimental study [2] was recreated in the FE environment, and all dependency was incorporated as per the discussion in Section 2. There is also a furnace temperature implemented in this model, and a predefined field impliment that the specimen is at room temperature. All four faces of the column in the central 1.7-meter height were exposed to fire, as shown in Figure 4. The column was tested for a 90-minute heating phase that followed that of ASTM E119 [3]. The experimental and FEA results are plotted and shown in Figure 5. From this plot, it is clearly seen that the experimental and FEA rebar temperatures are almost similar, but the concrete center temperature shows some deficit in the last 15 minutes of fire duration. After taking into account all of the factors in the FE model, the output results are also in line with the experimental results. This means that the approach has been verified and is ready for more parametric study.

4 Parametric Study

After the experimental results were validated in a finite element environment, similar methods are now being used for the parametric study. A number of studies are conducted in an FE environment to determine which simplified method provides better precision rebar temperature. Two different cross-sectional columns were chosen for this study, and each specimen was created for two effective cover sizes, which are 50 and 75 mm. These four categories are now being tested under standard fire [3] exposure for 30, 60, 90, 120, 150, 180, and 240 minute durations. Also, each specimen is exposed to fire on one face, two faces, three faces, and all faces of columns. Now, the number of studies required (2 column size x 2 effective covers x 7 fire exposure durations x 4 exposure faces = 112 different models) in the finite element model is 112. The rectangular column specimens cross-sectional profiles and fire exposure faces are described in Figure 6, similar follows for square columns.

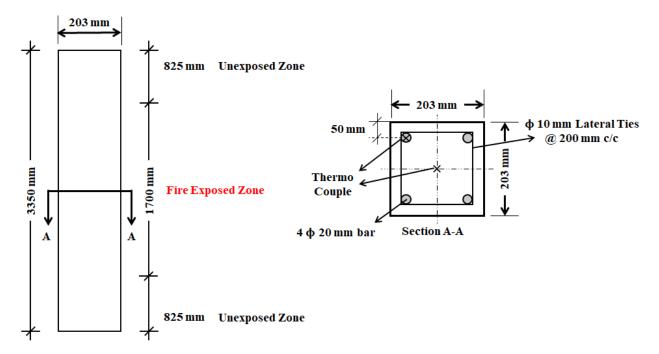


Figure 4: Dimensions and reinforcement details of RC columns.

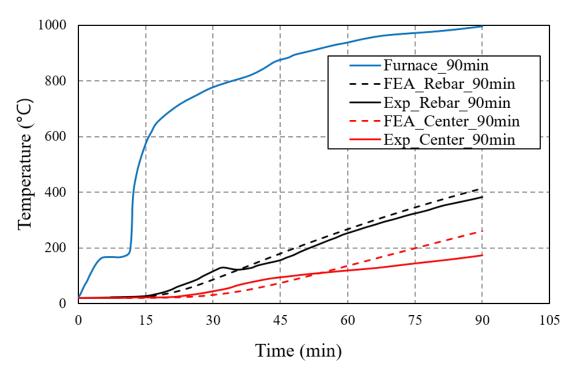


Figure 5: Comparison of predicted (FEA) and measured column temperatures during fire exposure. [2]

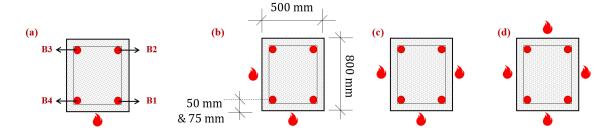


Figure 6: Selected specimens for study (a) one exposed side (b) two exposed sides (c) three exposed sides (d) four exposed sides to ASTM E119 [3] standard fire.

Preparing 112 models manually in a finite element environment is a time-consuming process. To reduce model preparation time, the Abaqus scripting [13] approach is adopted. This scripting approach helps create all of those models, and using this approach, anyone can prepare any number of models within a short period of time. The detailed scripting approaches are mentioned in Appendix A.

Because the effective cover-hold rebar position was the same for rectangular and square-shaped columns, extensive computational studies revealed that the rebar temperature remained constant. For this reason, the results of square column 56 studies are not shown in this term paper to avoid repetition. So, the remaining 56 studies are presented below.

Before talking about the results of parametric studies, it's important to explain how specimen ID is written. Such as "F240-C50" for a specimen ID, where "F" stands for "fire exposure," "240" stands for "240 minutes" of exposure, and "C50" stands for "50 mm effective cover." At first, discuss the finite element analysis results of one-sided fire exposure in rectangular columns. Compare the FE output among Wickstrom and Kodur simplified methods, as displayed in Figure 7. According to the studies, the simplified method performs poorly for 30-minute fire durations and performs better for 60-240 minute fire durations. There was no significant error found in the rebar temperature estimation by the simplified method with respect to varying the effective cover size of columns.

Figures 8, 9, and 10 show that there is a similar temperature found in "B4" rebar for fire exposure in two-, three-, and four-faced columns because "B4" rebar experiences 2D heat transfer in both cases. Wickstrom's proposed methods show higher precision in the estimation of rebar temperature in comparison to the Kodur method during 60–240 minute fire durations. But for a 30-minute fire duration, similar abrupt behavior was found in both methods. There is a significant effect noticed in Kodur's proposed method for 75 mm effective cover that shows more precision in estimating rebar temperature than a 50 mm effective cover column specimen during 150–240 minute fire durations.

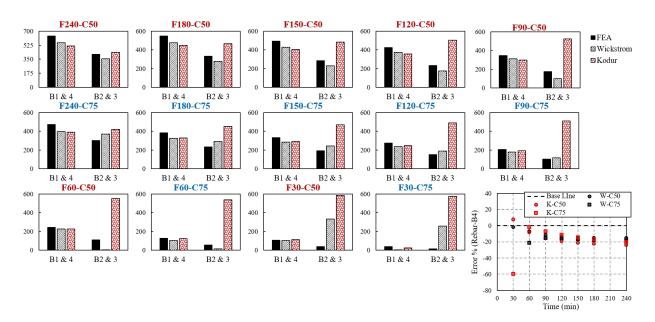


Figure 7: The temperature of Rebar exposed to fire on 1 side for different durations

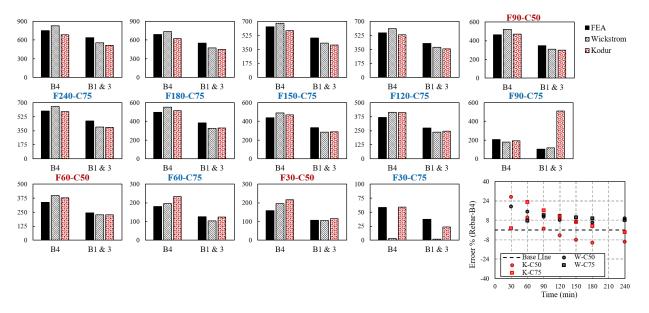


Figure 8: The temperature of Rebar exposed to fire on 2 side for different durations

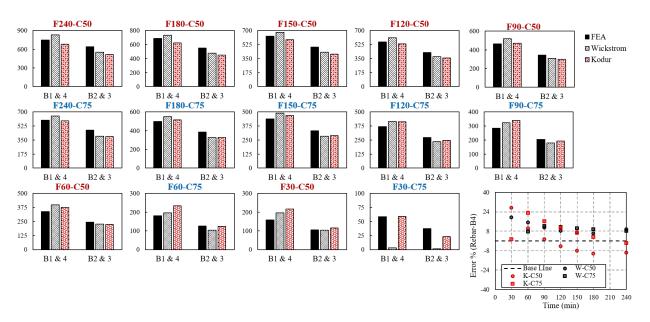


Figure 9: The temperature of Rebar exposed to fire on 3 side for different durations

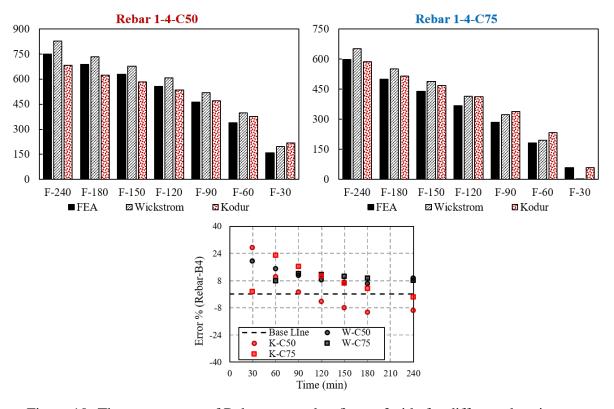


Figure 10: The temperature of Rebar exposed to fire on 3 side for different durations

5 Conclusion

Following an extensive 112 FE computational model study and comparison with two simplified temperature estimation techniques proposed by Wickstrom and Kodur, a few conclusions were reached. The following are the ideas generated or conclusions drawn from this study.

- 1. For Rebar B4, 30 min fire (C75) exposure shows an unusual error found in both simplified methods in all conditions (1, 2, 3, 4 side fire exposure).
- 2. The C50 Rebar B4 for one-sided fire exposure specimen shows less than a 15% error in temperature evaluation in the Wickstrom method and less than a 24% error in the Kodur proposed method. And the C75 Rebar B4 specimen for one sided fire exposure shows less than a 20% error in both methods.
- 3. Both methods show less than 15% error in temperature evaluation for C50 and C75 specimens' fire exposure on 2, 3, and 4 sides with 90-240 min fire exposure.
- 4. The percentage error in temperature estimation decreases as fire duration decreases (240-60 min range) in 1 side fire exposure in the Kodur approach. But no increasing or decreasing pattern was found for 2, 3, and 4 side fire exposure in both methods.
- 5. For a longer duration of fire exposure (60-240 min range), both simplified methods show an average 20temperature evaluation. Both simplified methods can be used in the temperature evaluation process for the above mentioned duration, but FEA can be used for greater precision.

References

- [1] Européen, C. Eurocode 2: Design of concrete structures—part 1-1: General rules and rules for buildings. *London: British Standard Institution* (2004).
- [2] Kodur, V., Hibner, D. & Agrawal, A. Residual response of reinforced concrete columns exposed to design fires. *Procedia engineering* **210**, 574–581 (2017).
- [3] ASTM Astm e119: Standard test methods for fire tests of building construction and materials (ASTM West Conshohocken, Pa, 2001).
- [4] Kodur, V. Properties of concrete at elevated temperatures. *International Scholarly Research Notices* **2014** (2014) .
- [5] Bhuiyan, M. H. & Ahmed, S. Arthur, S., Saitoh, M. & Pal, S. K. (eds) *Post-fire residual capacity of reinforced concrete beam.* (eds Arthur, S., Saitoh, M. & Pal, S. K.) *Advances in Civil Engineering*, 117–126 (Springer Singapore, Singapore, 2022).
- [6] Kodur, V. & Agrawal, A. An approach for evaluating residual capacity of reinforced concrete beams exposed to fire. *Engineering Structures* **110**, 293–306 (2016).
- [7] Kodur, V., Raut, N., Mao, X. & Khaliq, W. Simplified approach for evaluating residual strength of fire-exposed reinforced concrete columns. *Materials and Structures* **46** (12), 2059–2075 (2013).
- [8] Wickström, U. A very simple method for estimating temperaturein fire exposed concrete structures (1986).
- [9] Kodur, V., Yu, B. & Dwaikat, M. A simplified approach for predicting temperature in reinforced concrete members exposed to standard fire. *Fire Safety Journal* **56**, 39–51 (2013).
- [10] Buchanan, A. H. & Abu, A. K. *Structural design for fire safety* (John Wiley & Sons, 2017).
- [11] for Standardization), I. I. O. Iso 834: Fire-resistance tests-elements of building construction (1975).
- [12] KODUR, V., BHATT, P. & MATSAGAR, V. Numerical study on steel fiber reinforced concrete columns subjected to fire. *Structures in Fire* (2016).
- [13] Manual, A. S. U. Abaqus 6.11. http://130.149 89 (2080), v6 (2012).

Appendix A

Abaqus Scripting with python

```
This is my first Appendix.
 1 from part import *
 2 from material import *
 3 from section import *
 4 from assembly import *
 5 from step import *
 6 from interaction import *
7 from load import *
 8 from mesh import *
9 from optimization import *
10 from job import *
11 from sketch import *
12 from visualization import *
13 from connectorBehavior import *
14
  session.journalOptions.setValues(replayGeometry=COORDINATE,
15
      recoverGeometry=COORDINATE)
16
17 #Create Part
18 W=500
19 D=500
20 H=4000
21 C=75
22 C1 = 0.5 *W-C
23 C2=0.5*D-C
24 \quad t = 240 \quad \#input \quad as \quad min
25 d=25 #Mesh Size
26 b=H/(2*d)
27
28 mdb. models ['Model-1']. set Values (absolute Zero = -273.16,
      stefanBoltzmann = 5.667e - 11
```

```
29 mdb. models ['Model-1']. Constrained Sketch (name='_profile_',
      sheetSize = 200.0)
30 mdb.models['Model-1'].sketches['_profile__'].rectangle(
      point1 = (0.5*W, 0.5*D), point2 = (-0.5*W, -0.5*D)
   mdb.models['Model-1'].Part(dimensionality=THREE_D, name='
      Beam', type=DEFORMABLE_BODY)
   mdb. models ['Model-1']. parts ['Beam']. Base Solid Extrude (depth=H
      , sketch=mdb.models['Model-1'].sketches['_-profile_-'])
33
34
   mdb. models ['Model-1']. Constrained Sketch (name='__profile__',
      sheetSize = 200.0)
   mdb. models ['Model-1']. sketches ['_profile__']. Line (point1
      =(0.0, 0.0), point2=(H, 0.0)
36 mdb.models['Model-1'].sketches['_profile__'].
      HorizontalConstraint(addUndoState=False, entity=mdb.
      models ['Model-1']. sketches ['_profile__']. geometry [2])
37 mdb. models ['Model-1']. Part (dimensionality=THREE_D, name='Bar
      ', tvpe=DEFORMABLE_BODY)
   mdb. models ['Model-1']. parts ['Bar']. BaseWire (sketch=mdb.
      models['Model-1'].sketches['_-profile_-'])
39
40 #Cell
41
42 mdb. models ['Model-1']. parts ['Beam'].
      DatumPlaneByPrincipalPlane (offset = 0.0, principalPlane =
      YZPLANE)
43 mdb. models ['Model-1']. parts ['Beam'].
      DatumPlaneByPrincipalPlane (offset = 0.0, principalPlane =
      XZPLANE)
   mdb. models ['Model-1']. parts ['Beam'].
      DatumPlaneByPrincipalPlane(offset=C1, principalPlane=
      YZPLANE)
45 mdb. models ['Model-1']. parts ['Beam'].
      DatumPlaneByPrincipalPlane(offset=-C1, principalPlane=
      YZPLANE)
46 mdb. models ['Model-1']. parts ['Beam'].
      DatumPlaneByPrincipalPlane(offset=C2, principalPlane=
      XZPLANE)
47 mdb. models ['Model-1']. parts ['Beam'].
      DatumPlaneByPrincipalPlane(offset=-C2, principalPlane=
      XZPLANE)
48
49 #Cut
50
```

```
51 mdb. models ['Model-1']. parts ['Beam'].
      PartitionCellByDatumPlane(cells=mdb.models['Model-1'].
      parts ['Beam']. cells . findAt (((-0.5*W, -0.5*D, H), )),
      datumPlane=mdb.models['Model-1'].parts['Beam'].datums[2])
52 mdb. models ['Model-1']. parts ['Beam'].
      PartitionCellByDatumPlane(cells=mdb.models['Model-1'].
      parts ['Beam']. cells . findAt (((0.5*W, 0.5*D, H), ),((-0.5*W
      , -0.5*D, H),),), datumPlane=mdb.models['Model-1'].parts[
      'Beam']. datums [3])
53 mdb. models ['Model-1']. parts ['Beam'].
      PartitionCellByDatumPlane(cells=mdb.models['Model-1'].
      parts ['Beam']. cells . findAt (((0.5*W, 0.5*D, H), ), ((0.5*W, 0.5*D, H), ))
       -0.5*D, H),), datumPlane=mdb.models['Model-1'].parts['
      Beam']. datums [4])
54 mdb. models ['Model-1']. parts ['Beam'].
      PartitionCellByDatumPlane(cells=mdb.models['Model-1'].
      parts ['Beam']. cells.findAt(((-0.5*W, 0.5*D, H), ),((-0.5*
      W, -0.5*D, H),), datumPlane=mdb.models['Model-1'].parts
      ['Beam']. datums [5])
55 mdb. models ['Model-1']. parts ['Beam'].
      PartitionCellByDatumPlane(cells=mdb.models['Model-1'].
      parts ['Beam']. cells . findAt (((0.5*W, 0.5*D, H), ), ((-0.5*W), (0.5*D, H))
      (0.5*D, H), ((-0.5*C1, 0.5*D, H), ((0.5*C1, 0.5*D, H))
      ,),), datumPlane=mdb.models['Model-1'].parts['Beam'].
      datums [6])
56 mdb. models ['Model-1']. parts ['Beam'].
      PartitionCellByDatumPlane(cells=mdb.models['Model-1'].
      parts ['Beam']. cells . findAt (((0.5*W, -0.5*D, H), ),((-0.5*
      W, -0.5*D, H),, ((-0.5*C1, -0.5*D, H)), ((0.5*C1, -0.5*D, H))
       H),),), datumPlane=mdb.models['Model-1'].parts['Beam'].
      datums [7])
57
58 #Material
59
60 mdb. models ['Model-1']. Material (name='Concrete')
61 mdb. models ['Model-1']. materials ['Concrete']. Density (table
      =((2.3e-09,),)
   mdb. models ['Model-1']. materials ['Concrete']. Conductivity (
      table = ((1.33, 20.0), (
        1.23, 100.0), (1.11, 200.0), (1.0, 300.0), (0.91, 400.0)
63
           , (0.82, 500.0), (
64
       0.75, 600.0), (0.69, 700.0), (0.64, 800.0), (0.6, 900.0)
           (0.57, 1000.0), (
       0.55, 1100.0), (0.55, 1200.0)), temperatureDependency=ON
65
          )
```

```
66 mdb. models ['Model-1']. materials ['Concrete']. Specific Heat (
      table = ((900000000.0,
67
       (20.0), (900000000.0, (10000000000.0, (200.0),
          (1050000000.0, 300.0),
68
       (1100000000.0, 400.0), (1100000000.0, 500.0),
          (1100000000.0, 600.0), (
       1100000000.0, 700.0), (1100000000.0, 800.0),
69
          (1100000000.0, 900.0), (
70
       1100000000.0, 1000.0), (1100000000.0, 1100.0),
          (11000000000.0, 1200.0)),
71
       temperatureDependency=ON)
72
73 mdb.models['Model-1'].Material(name='Steel')
74 mdb. models ['Model-1']. materials ['Steel']. Density (table
      =((7.85e-09, ), ))
   mdb. models ['Model-1']. materials ['Steel']. Conductivity (table
      =((53.3, 20.0), (
       50.7, 100.0), (47.3, 200.0), (44.0, 300.0), (40.7,
76
          400.0), (37.4, 500.0), (
       34.0, 600.0), (30.7, 700.0), (27.3, 800.0), (27.3, 900.0)
77
          900.0), (27.3, 1000.0),
       (27.3, 1100.0), (27.3, 1200.0)), temperature Dependency =
78
          ON)
   mdb.models['Model-1'].materials['Steel'].SpecificHeat(table
79
      =((439801760.0,
       (20.0), (487620000.0, (100.0), (529760000.0, (200.0),
80
          (564740000.0, 300.0), (
       605880000.0, 400.0), (666500000.0, 500.0), (760217391.3,
81
           600.0), (
82
       1008157895.0, 700.0), (5000000000.0, 735.0),
          (803260869.6, 800.0), (
83
       650000000.0, 900.0), (650000000.0, 1000.0),
          (6500000000.0, 1100.0), (
84
       650000000.0, 1200.0), temperatureDependency=ON)
85
86 mdb. models ['Model-1']. Homogeneous Solid Section (material='
      Concrete', name='C', thickness=None)
   mdb. models ['Model-1']. Truss Section (area = 200.0, material='
87
      Steel', name='S')
  mdb. models ['Model-1']. parts ['Beam']. Section Assignment (offset
      =0.0, offsetField='', offsetType=MIDDLE_SURFACE, region=
      Region (cells = mdb. models ['Model-1']. parts ['Beam']. cells.
      findAt(((0.5*C1, 0.5*C2, H), ), ((-0.5*C1, 0.5*C2, H), ),
       ((0.5*C1, -0.5*C2, H), ), ((-0.5*C1, -0.5*C2, H), ),
      ((0.5*C1, 0.5*D, H), ), ((0.5*C1, -0.5*D, H), ), ((-0.5*C1, -0.5*D, H), )
```

```
C1, 0.5*D, H), ), ((-0.5*C1, -0.5*D, H), ), ((1.05*C1, -0.5*D, H),
       0.5*C2, H), ), ((1.05*C1, -0.5*C2, H), ), ((-1.05*C1,
       0.5*C2, H), ), ((-1.05*C1, -0.5*C2, H), ), ((0.45*W,
       0.45*D, H), ((0.45*W, -0.45*D, H), ), ((-0.45*W, 0.45*)
       D, H), ), ((-0.45*W, -0.45*D, H), ), sectionName='C',
        thickness Assignment=FROM_SECTION)
89 mdb. models ['Model-1']. parts ['Bar']. Section Assignment (offset
       =0.0, offsetField='', offsetType=MIDDLE_SURFACE, region=
       Region (edges=mdb. models ['Model-1']. parts ['Bar']. edges.
       findAt(((0.5*H, 0.0, 0.0), ), )), sectionName='S',
       thickness Assignment=FROM_SECTION)
90
91 #Assembly
92
93 mdb.models['Model-1'].rootAssembly.DatumCsysByDefault(
       CARTESIAN)
94
   mdb. models ['Model-1']. rootAssembly. Instance (dependent=ON,
       name='Bar-1', part=mdb.models['Model-1'].parts['Bar'])
95 mdb. models ['Model-1']. rootAssembly. LinearInstancePattern (
       direction 1 = (1.0, 0.0, 0.0), direction 2 = (0.0, 1.0, 0.0),
       instanceList=('Bar-1', ), number1=1, number2=2, spacing1=
       H, spacing 2=2*C2)
96 mdb. models ['Model-1']. rootAssembly. rotate (angle = -90.0,
       axisDirection = (0.0, 2*C2, 0.0), axisPoint = (0.0, 0.0, 0.0)
       , instanceList=('Bar-1', 'Bar-1-lin-1-2'))
97 mdb.models['Model-1'].rootAssembly.LinearInstancePattern(
       direction 1 = (1.0, 0.0, 0.0), direction 2 = (0.0, 1.0, 0.0),
       instanceList = ('Bar - 1', 'Bar - 1 - 1in - 1 - 2'), number 1 = 2,
       number 2=1, spacing 1=2*C1, spacing 2=2*C2)
   mdb. models ['Model-1']. rootAssembly. Instance (dependent=ON,
       name='Beam-1', part=mdb.models['Model-1'].parts['Beam'])
   mdb.models['Model-1'].rootAssembly.translate(instanceList=('
       Bar-1', 'Bar-1-lin-1-2', 'Bar-1-lin-2-1', 'Bar-1-lin-1-2-
       \lim_{n \to \infty} -2-1'), vector=(-C1, -C2, 0.0))
100
101
102 #Step
103
104 mdb. models ['Model-1']. HeatTransferStep (deltmx = 25.0,
       initialInc=10.0, maxInc=200.0, maxNumInc=10000000, minInc
       =0.1, name='Fire', previous='Initial', timePeriod=t*60)
105 mdb. models ['Model-1']. fieldOutputRequests ['F-Output-1'].
       setValues (variables = ('NT', ))
106
107 #Amplitude
```

```
108
109 mdb. models ['Model-1']. Tabular Amplitude (data = ((0.0, 20.0)),
       (60.0, 332.5304157),
        (120.0, 426.044443), (180.0, 487.12866), (240.0, 
110
           532.516808), (300.0,
111
        568.4577615), (360.0, 598.0496271), (420.0, 623.0742538)
            (480.0,
112
        644.6569038), (540.0, 663.5558175), (600.0, 680.3069656)
            , (660.0,
113
        695.303428), (720.0, 708.8420627), (780.0, 721.1524826),
            (840.0,
114
        732.4158538), (900.0, 742.7775408), (960.0, 752.3558747)
            , (1020.0.
        761.248392), (1080.0, 769.5363737), (1140.0,
115
           777.2882137), (1200.0,
116
        784.5619596), (1260.0, 791.4072598), (1320.0, 791.4072598)
           797.8668765), (1380.0,
        803.9778734), (1440.0, 809.7725613), (1500.0,
117
           815.2792562), (1560.0,
        820.5228938), (1620.0, 825.5255316), (1680.0, 825.5255316)
118
           830.3067617), (1740.0,
119
        834.8840538), (1800.0, 839.2730404)), name='Fire-30',
           smooth=SOLVER_DEFAULT
120
         , timeSpan=STEP)
121
122
    mdb. models ['Model-1']. Tabular Amplitude (data = ((0.0, 20.0)),
       (60.0, 332.5304157),
        (120.0, 426.044443), (180.0, 487.12866), (240.0,
123
           532.516808), (300.0,
        568.4577615), (360.0, 598.0496271), (420.0, 623.0742538)
124
            , (480.0,
        644.6569038), (540.0, 663.5558175), (600.0, 680.3069656)
125
            , (660.0,
126
        695.303428), (720.0, 708.8420627), (780.0, 721.1524826),
            (840.0,
127
        732.4158538), (900.0, 742.7775408), (960.0, 752.3558747)
            (1020.0)
128
        761.248392), (1080.0, 769.5363737), (1140.0, 769.5363737)
           777.2882137), (1200.0,
        784.5619596), (1260.0, 791.4072598), (1320.0, 791.4072598)
129
           797.8668765), (1380.0,
        803.9778734), (1440.0, 809.7725613), (1500.0,
130
           815.2792562), (1560.0,
131
        820.5228938), (1620.0, 825.5255316), (1680.0, 825.5255316)
           830.3067617), (1740.0,
```

```
132
        834.8840538), (1800.0, 839.2730404), (1860.0, 839.2730404)
            843.4877575), (1920.0,
133
        847.5408468), (1980.0, 851.4437275), (2040.0,
            855.2067436), (2100.0,
        858.8392889), (2160.0, 862.349916), (2220.0, 865.746429)
134
            (2280.0,
        869.0359653), (2340.0, 872.2250658), (2400.0, 872.2250658)
135
            875.3197372), (2460.0,
        878.3255053), (2520.0, 881.2474635), (2580.0,
136
            884.0903142), (2640.0,
137
        886.8584059), (2700.0, 889.5557668), (2760.0, 889.5557668)
            892.1861333), (2820.0,
        894.7529764), (2880.0, 897.2595254), (2940.0.
138
            899.708788), (3000.0,
        902.1035698), (3060.0, 904.4464908), (3120.0, 904.4464908)
139
            906.7400006), (3180.0,
        908.9863921), (3240.0, 911.187814), (3300.0,
140
            913.3462819), (3360.0,
141
        915.4636885), (3420.0, 917.5418131), (3480.0, 917.5418131)
            919.5823298), (3540.0,
        921.5868151), (3600.0, 923.5567552)), name='Fire-60',
142
            smooth=SOLVER_DEFAULT
143
         , timeSpan=STEP)
144
    mdb. models ['Model-1']. Tabular Amplitude (data = ((0.0, 20.0)),
145
       (60.0, 332.5304157),
146
        (120.0, 426.044443), (180.0, 487.12866), (240.0,
            532.516808), (300.0,
147
        568.4577615), (360.0, 598.0496271), (420.0, 623.0742538)
            , (480.0,
148
        644.6569038), (540.0, 663.5558175), (600.0, 680.3069656)
            , (660.0,
149
        695.303428), (720.0, 708.8420627), (780.0, 721.1524826),
             (840.0,
150
        732.4158538), (900.0, 742.7775408), (960.0, 752.3558747)
            (1020.0,
151
        761.248392), (1080.0, 769.5363737), (1140.0, 769.5363737)
            777.2882137), (1200.0,
152
        784.5619596), (1260.0, 791.4072598), (1320.0,
            797.8668765), (1380.0,
153
        803.9778734), (1440.0, 809.7725613), (1500.0, 809.7725613)
            815.2792562), (1560.0,
154
        820.5228938), (1620.0, 825.5255316), (1680.0, 825.5255316)
            830.3067617), (1740.0,
```

```
155
         834.8840538), (1800.0, 839.2730404), (1860.0, 839.2730404)
            843.4877575), (1920.0,
156
         847.5408468), (1980.0, 851.4437275), (2040.0,
            855.2067436), (2100.0,
         858.8392889), (2160.0, 862.349916), (2220.0, 865.746429)
157
            (2280.0,
         869.0359653), (2340.0, 872.2250658), (2400.0,
158
            875.3197372), (2460.0,
159
         878.3255053), (2520.0, 881.2474635), (2580.0,
            884.0903142), (2640.0,
160
         886.8584059), (2700.0, 889.5557668), (2760.0, 889.5557668)
            892.1861333), (2820.0,
         894.7529764), (2880.0, 897.2595254), (2940.0,
161
            899.708788), (3000.0,
162
        902.1035698), (3060.0, 904.4464908), (3120.0, 904.4464908)
            906.7400006), (3180.0,
         908.9863921), (3240.0, 911.187814), (3300.0,
163
            913.3462819), (3360.0,
164
         915.4636885), (3420.0, 917.5418131), (3480.0, 917.5418131)
            919.5823298), (3540.0,
        921.5868151), (3600.0, 923.5567552), (3660.0, 923.5567552)
165
            925.4935524), (3720.0,
166
         927.3985305), (3780.0, 929.272941), (3840.0, 929.272941)
            931.1179673), (3900.0,
         932.9347297), (3960.0, 934.7242894), (4020.0,
167
            936.4876525), (4080.0,
168
        938.2257733), (4140.0, 939.9395579), (4200.0, 939.9395579)
            941.629867), (4260.0,
169
        943.2975188), (4320.0, 944.9432918), (4380.0,
            946.5679266), (4440.0,
170
        948.1721291), (4500.0, 949.7565716), (4560.0, 949.7565716)
            951.3218955), (4620.0,
         952.8687125), (4680.0, 954.3976068), (4740.0, 954.3976068)
171
            955.9091364), (4800.0,
172
         957.4038344), (4860.0, 958.8822108), (4920.0, 958.8822108)
            960.3447534), (4980.0,
173
        961.7919291), (5040.0, 963.224185), (5100.0,
            964.6419498), (5160.0,
174
        966.0456339), (5220.0, 967.4356315), (5280.0,
            968.8123203), (5340.0,
175
        970.1760633), (5400.0, 971.5272087)), name='Fire-90',
            smooth=SOLVER_DEFAULT
176
         , timeSpan=STEP)
```

177

- 178 mdb. models ['Model-1']. Tabular Amplitude (data = ((0.0, 20.0), (60.0, 332.5304157),
- 179 (120.0, 426.044443), (180.0, 487.12866), (240.0, 532.516808), (300.0,
- 180 568.4577615), (360.0, 598.0496271), (420.0, 623.0742538), (480.0,
- 181 644.6569038), (540.0, 663.5558175), (600.0, 680.3069656), (660.0,
- 182 695.303428), (720.0, 708.8420627), (780.0, 721.1524826), (840.0,
- 183 732.4158538), (900.0, 742.7775408), (960.0, 752.3558747), (1020.0,
- 761.248392), (1080.0, 769.5363737), (1140.0, 777.2882137), (1200.0,
- 784.5619596), (1260.0, 791.4072598), (1320.0, 797.8668765), (1380.0,
- 186 803.9778734), (1440.0, 809.7725613), (1500.0, 815.2792562), (1560.0,
- 187 820.5228938), (1620.0, 825.5255316), (1680.0, 830.3067617), (1740.0,
- 188 834.8840538), (1800.0, 839.2730404), (1860.0, 843.4877575), (1920.0,
- 189 847.5408468), (1980.0, 851.4437275), (2040.0, 855.2067436), (2100.0,
- 190 858.8392889), (2160.0, 862.349916), (2220.0, 865.746429), (2280.0,
- 191 869.0359653), (2340.0, 872.2250658), (2400.0, 875.3197372), (2460.0,
- 192 878.3255053), (2520.0, 881.2474635), (2580.0, 884.0903142), (2640.0,
- 193 886.8584059), (2700.0, 889.5557668), (2760.0, 892.1861333), (2820.0,
- 194 894.7529764), (2880.0, 897.2595254), (2940.0, 899.708788), (3000.0,
- 902.1035698), (3060.0, 904.4464908), (3120.0, 906.7400006), (3180.0,
- 908.9863921), (3240.0, 911.187814), (3300.0, 913.3462819), (3360.0,
- 915.4636885), (3420.0, 917.5418131), (3480.0, 919.5823298), (3540.0,
- 198 921.5868151), (3600.0, 923.5567552), (3660.0, 925.4935524), (3720.0,
- 199 927.3985305), (3780.0, 929.272941), (3840.0, 931.1179673), (3900.0,

```
200
         932.9347297), (3960.0, 934.7242894), (4020.0, 934.7242894)
            936.4876525), (4080.0,
201
         938.2257733), (4140.0, 939.9395579), (4200.0, 939.9395579)
            941.629867), (4260.0,
         943.2975188), (4320.0, 944.9432918), (4380.0,
202
            946.5679266), (4440.0,
203
         948.1721291), (4500.0, 949.7565716), (4560.0, 949.7565716)
            951.3218955), (4620.0,
204
         952.8687125), (4680.0, 954.3976068), (4740.0,
            955.9091364), (4800.0,
205
         957.4038344), (4860.0, 958.8822108), (4920.0, 958.8822108)
            960.3447534), (4980.0,
         961.7919291), (5040.0, 963.224185), (5100.0, 963.224185)
206
            964.6419498), (5160.0,
         966.0456339), (5220.0, 967.4356315), (5280.0,
207
            968.8123203), (5340.0,
         970.1760633), (5400.0, 971.5272087), (5460.0, 971.5272087)
208
            972.8660913), (5520.0,
209
         974.1930328), (5580.0, 975.5083426), (5640.0, 975.5083426)
            976.8123183), (5700.0,
         978.1052462), (5760.0, 979.387402), (5820.0, 979.387402)
210
            980.6590513), (5880.0,
211
         981.9204499), (5940.0, 983.1718443), (6000.0,
            984.4134723), (6060.0,
212
         985.6455631), (6120.0, 986.8683381), (6180.0,
            988.0820108), (6240.0,
213
         989.2867873), (6300.0, 990.4828668), (6360.0, 989.2867873)
            991.6704417), (6420.0,
214
         992.849698), (6480.0, 994.0208155), (6540.0,
            995.1839681), (6600.0,
215
         996.339324), (6660.0, 997.4870459), (6720.0, 997.4870459)
            998.6272916), (6780.0,
         999.7602136), (6840.0, 1000.88596), (6900.0,
216
            1002.004673), (6960.0,
217
         1003.116492), (7020.0, 1004.221553), (7080.0, 1004.221553)
            1005.319984), (7140.0,
218
         1006.411913), (7200.0, 1007.497462)), name='Fire-120',
            smooth=
219
         SOLVER_DEFAULT, timeSpan=STEP)
220
    mdb. models ['Model-1']. Tabular Amplitude (data = ((0.0, 20.0)),
221
        (60.0, 332.5304157)
222
         (120.0, 426.044443), (180.0, 487.12866), (240.0,
            532.516808), (300.0,
```

```
223 568.4577615), (360.0, 598.0496271), (420.0, 623.0742538), (480.0,
```

- 224 644.6569038), (540.0, 663.5558175), (600.0, 680.3069656), (660.0,
- 225 695.303428), (720.0, 708.8420627), (780.0, 721.1524826), (840.0,
- 226 732.4158538), (900.0, 742.7775408), (960.0, 752.3558747), (1020.0,
- 227 761.248392), (1080.0, 769.5363737), (1140.0, 777.2882137), (1200.0,
- 784.5619596), (1260.0, 791.4072598), (1320.0, 797.8668765), (1380.0,
- 229 803.9778734), (1440.0, 809.7725613), (1500.0, 815.2792562), (1560.0,
- 230 820.5228938), (1620.0, 825.5255316), (1680.0, 830.3067617), (1740.0,
- 231 834.8840538), (1800.0, 839.2730404), (1860.0, 843.4877575), (1920.0,
- 232 847.5408468), (1980.0, 851.4437275), (2040.0, 855.2067436), (2100.0,
- 233 858.8392889), (2160.0, 862.349916), (2220.0, 865.746429), (2280.0,
- 234 869.0359653), (2340.0, 872.2250658), (2400.0, 875.3197372), (2460.0,
- 235 878.3255053), (2520.0, 881.2474635), (2580.0, 884.0903142), (2640.0,
- 236 886.8584059), (2700.0, 889.5557668), (2760.0, 892.1861333), (2820.0,
- 237 894.7529764), (2880.0, 897.2595254), (2940.0, 899.708788), (3000.0,
- 238 902.1035698), (3060.0, 904.4464908), (3120.0, 906.7400006), (3180.0,
- 239 908.9863921), (3240.0, 911.187814), (3300.0, 913.3462819), (3360.0,
- 240 915.4636885), (3420.0, 917.5418131), (3480.0, 919.5823298), (3540.0,
- 241 921.5868151), (3600.0, 923.5567552), (3660.0, 925.4935524), (3720.0,
- 242 927.3985305), (3780.0, 929.272941), (3840.0, 931.1179673), (3900.0,
- 243 932.9347297), (3960.0, 934.7242894), (4020.0, 936.4876525), (4080.0,
- 244 938.2257733), (4140.0, 939.9395579), (4200.0, 941.629867), (4260.0,

- 245 943.2975188), (4320.0, 944.9432918), (4380.0, 946.5679266), (4440.0,
- 246 948.1721291), (4500.0, 949.7565716), (4560.0, 951.3218955), (4620.0,
- 247 952.8687125), (4680.0, 954.3976068), (4740.0, 955.9091364), (4800.0,
- 248 957.4038344), (4860.0, 958.8822108), (4920.0, 960.3447534), (4980.0,
- 249 961.7919291), (5040.0, 963.224185), (5100.0, 964.6419498), (5160.0,
- 250 966.0456339), (5220.0, 967.4356315), (5280.0, 968.8123203), (5340.0,
- 251 970.1760633), (5400.0, 971.5272087), (5460.0, 972.8660913), (5520.0,
- 252 974.1930328), (5580.0, 975.5083426), (5640.0, 976.8123183), (5700.0,
- 253 978.1052462), (5760.0, 979.387402), (5820.0, 980.6590513), (5880.0,
- 254 981.9204499), (5940.0, 983.1718443), (6000.0, 984.4134723), (6060.0,
- 255 985.6455631), (6120.0, 986.8683381), (6180.0, 988.0820108), (6240.0,
- 256 989.2867873), (6300.0, 990.4828668), (6360.0, 991.6704417), (6420.0,
- 257 992.849698), (6480.0, 994.0208155), (6540.0, 995.1839681), (6600.0,
- 258 996.339324), (6660.0, 997.4870459), (6720.0, 998.6272916), (6780.0,
- 259 999.7602136), (6840.0, 1000.88596), (6900.0, 1002.004673), (6960.0,
- 260 1003.116492), (7020.0, 1004.221553), (7080.0, 1005.319984), (7140.0,
- 261 1006.411913), (7200.0, 1007.497462), (7260.0, 1008.576751), (7320.0,
- 262 1009.649895), (7380.0, 1010.717007), (7440.0, 1011.778196), (7500.0,
- 263 1012.833568), (7560.0, 1013.883225), (7620.0, 1014.927268), (7680.0,
- 264 1015.965794), (7740.0, 1016.998898), (7800.0, 1018.026672), (7860.0,
- 265 1019.049206), (7920.0, 1020.066586), (7980.0, 1021.078898), (8040.0,
- 266 1022.086223), (8100.0, 1023.088643), (8160.0, 1024.086236), (8220.0,

```
267
         1025.079079), (8280.0, 1026.067244), (8340.0, 1026.067244)
            1027.050806), (8400.0,
268
         1028.029835), (8460.0, 1029.004399), (8520.0, 1029.004399)
            1029.974566), (8580.0,
         1030.940401), (8640.0, 1031.901968), (8700.0, 1031.901968)
269
            1032.859331), (8760.0,
         1033.812549), (8820.0, 1034.761682), (8880.0, 1034.761682)
270
            1035.706788), (8940.0,
         1036.647924), (9000.0, 1037.585145)), name='Fire-150'.
271
            smooth=
272
        SOLVER_DEFAULT, timeSpan=STEP)
273
274
    mdb. models ['Model-1']. Tabular Amplitude (data = ((0.0, 20.0)),
       (60.0, 332.5304157),
         (120.0, 426.044443), (180.0, 487.12866), (240.0, 
275
            532.516808), (300.0,
276
         568.4577615), (360.0, 598.0496271), (420.0, 623.0742538)
            , (480.0,
277
         644.6569038), (540.0, 663.5558175), (600.0, 680.3069656)
            , (660.0,
278
         695.303428), (720.0, 708.8420627), (780.0, 721.1524826),
             (840.0,
279
         732.4158538), (900.0, 742.7775408), (960.0, 752.3558747)
            , (1020.0,
280
         761.248392), (1080.0, 769.5363737), (1140.0, 769.5363737)
            777.2882137), (1200.0,
281
         784.5619596), (1260.0, 791.4072598), (1320.0,
            797.8668765), (1380.0,
282
         803.9778734), (1440.0, 809.7725613), (1500.0,
            815.2792562), (1560.0,
283
         820.5228938), (1620.0, 825.5255316), (1680.0, 825.5255316)
            830.3067617), (1740.0,
         834.8840538), (1800.0, 839.2730404), (1860.0, 839.2730404)
284
            843.4877575), (1920.0,
285
         847.5408468), (1980.0, 851.4437275), (2040.0,
            855.2067436), (2100.0,
286
         858.8392889), (2160.0, 862.349916), (2220.0, 865.746429)
            , (2280.0,
287
         869.0359653), (2340.0, 872.2250658), (2400.0, 872.2250658)
            875.3197372), (2460.0,
288
         878.3255053), (2520.0, 881.2474635), (2580.0,
            884.0903142), (2640.0,
289
         886.8584059), (2700.0, 889.5557668), (2760.0, 889.5557668)
            892.1861333), (2820.0,
```

- 290 894.7529764), (2880.0, 897.2595254), (2940.0, 899.708788), (3000.0,
- 291 902.1035698), (3060.0, 904.4464908), (3120.0, 906.7400006), (3180.0,
- 292 908.9863921), (3240.0, 911.187814), (3300.0, 913.3462819), (3360.0,
- 293 915.4636885), (3420.0, 917.5418131), (3480.0, 919.5823298), (3540.0,
- 294 921.5868151), (3600.0, 923.5567552), (3660.0, 925.4935524), (3720.0,
- 295 927.3985305), (3780.0, 929.272941), (3840.0, 931.1179673), (3900.0,
- 296 932.9347297), (3960.0, 934.7242894), (4020.0, 936.4876525), (4080.0,
- 297 938.2257733), (4140.0, 939.9395579), (4200.0, 941.629867), (4260.0,
- 298 943.2975188), (4320.0, 944.9432918), (4380.0, 946.5679266), (4440.0,
- 299 948.1721291), (4500.0, 949.7565716), (4560.0, 951.3218955), (4620.0,
- 300 952.8687125), (4680.0, 954.3976068), (4740.0, 955.9091364), (4800.0,
- 301 957.4038344), (4860.0, 958.8822108), (4920.0, 960.3447534), (4980.0,
- 302 961.7919291), (5040.0, 963.224185), (5100.0, 964.6419498), (5160.0,
- 303 966.0456339), (5220.0, 967.4356315), (5280.0, 968.8123203), (5340.0,
- 304 970.1760633), (5400.0, 971.5272087), (5460.0, 972.8660913), (5520.0,
- 305 974.1930328), (5580.0, 975.5083426), (5640.0, 976.8123183), (5700.0,
- 306 978.1052462), (5760.0, 979.387402), (5820.0, 980.6590513), (5880.0,
- 307 981.9204499), (5940.0, 983.1718443), (6000.0, 984.4134723), (6060.0,
- 308 985.6455631), (6120.0, 986.8683381), (6180.0, 988.0820108), (6240.0,
- 309 989.2867873), (6300.0, 990.4828668), (6360.0, 991.6704417), (6420.0,
- 310 992.849698), (6480.0, 994.0208155), (6540.0, 995.1839681), (6600.0,
- 311 996.339324), (6660.0, 997.4870459), (6720.0, 998.6272916), (6780.0,

```
312 999.7602136), (6840.0, 1000.88596), (6900.0, 1002.004673), (6960.0,
```

- 313 1003.116492), (7020.0, 1004.221553), (7080.0, 1005.319984), (7140.0,
- 314 1006.411913), (7200.0, 1007.497462), (7260.0, 1008.576751), (7320.0,
- 315 1009.649895), (7380.0, 1010.717007), (7440.0, 1011.778196), (7500.0,
- 316 1012.833568), (7560.0, 1013.883225), (7620.0, 1014.927268), (7680.0,
- 317 1015.965794), (7740.0, 1016.998898), (7800.0, 1018.026672), (7860.0,
- 318 1019.049206), (7920.0, 1020.066586), (7980.0, 1021.078898), (8040.0,
- 319 1022.086223), (8100.0, 1023.088643), (8160.0, 1024.086236), (8220.0,
- 320 1025.079079), (8280.0, 1026.067244), (8340.0, 1027.050806), (8400.0,
- 321 1028.029835), (8460.0, 1029.004399), (8520.0, 1029.974566), (8580.0,
- 322 1030.940401), (8640.0, 1031.901968), (8700.0, 1032.859331), (8760.0,
- 323 1033.812549), (8820.0, 1034.761682), (8880.0, 1035.706788), (8940.0,
- 324 1036.647924), (9000.0, 1037.585145), (9060.0, 1038.518505), (9120.0,
- 325 1039.448058), (9180.0, 1040.373854), (9240.0, 1041.295944), (9300.0,
- 326 1042.214379), (9360.0, 1043.129205), (9420.0, 1044.04047), (9480.0,
- 327 1044.948221), (9540.0, 1045.852502), (9600.0, 1046.753358), (9660.0,
- 328 1047.650832), (9720.0, 1048.544966), (9780.0, 1049.435802), (9840.0,
- 329 1050.32338), (9900.0, 1051.20774), (9960.0, 1052.088921), (10020.0,
- 330 1052.966961), (10080.0, 1053.841898), (10140.0, 1054.713767), (10200.0,
- 331 1055.582606), (10260.0, 1056.448448), (10320.0, 1057.311329), (10380.0,
- 332 1058.171282), (10440.0, 1059.028341), (10500.0, 1059.882537), (10560.0,
- 333 1060.733903), (10620.0, 1061.58247), (10680.0, 1062.428268), (10740.0,

```
334
        1063.271329), (10800.0, 1064.11168)), name='Fire -180',
           smooth=
335
        SOLVER_DEFAULT, timeSpan=STEP)
336
337
    mdb. models ['Model-1']. Tabular Amplitude (data = ((0.0, 20.0)),
       (60.0, 332.5304157),
        (120.0, 426.044443), (180.0, 487.12866), (240.0,
338
           532.516808), (300.0,
        568.4577615), (360.0, 598.0496271), (420.0, 623.0742538)
339
            (480.0,
340
        644.6569038), (540.0, 663.5558175), (600.0, 680.3069656)
            , (660.0,
341
        695.303428), (720.0, 708.8420627), (780.0, 721.1524826),
             (840.0,
        732.4158538), (900.0, 742.7775408), (960.0, 752.3558747)
342
            (1020.0,
        761.248392), (1080.0, 769.5363737), (1140.0,
343
           777.2882137), (1200.0,
344
        784.5619596), (1260.0, 791.4072598), (1320.0,
           797.8668765), (1380.0,
        803.9778734), (1440.0, 809.7725613), (1500.0,
345
           815.2792562), (1560.0,
346
        820.5228938), (1620.0, 825.5255316), (1680.0,
           830.3067617), (1740.0,
347
        834.8840538), (1800.0, 839.2730404), (1860.0, 839.2730404)
           843.4877575), (1920.0,
348
        847.5408468), (1980.0, 851.4437275), (2040.0,
           855.2067436), (2100.0,
349
        858.8392889), (2160.0, 862.349916), (2220.0, 865.746429)
            , (2280.0,
350
        869.0359653), (2340.0, 872.2250658), (2400.0, 872.2250658)
           875.3197372), (2460.0,
351
        878.3255053), (2520.0, 881.2474635), (2580.0,
           884.0903142), (2640.0,
352
        886.8584059), (2700.0, 889.5557668), (2760.0, 889.5557668)
           892.1861333), (2820.0,
353
        894.7529764), (2880.0, 897.2595254), (2940.0,
           899.708788), (3000.0,
354
        902.1035698), (3060.0, 904.4464908), (3120.0, 904.4464908)
           906.7400006), (3180.0,
355
        908.9863921), (3240.0, 911.187814), (3300.0,
           913.3462819), (3360.0,
356
        915.4636885), (3420.0, 917.5418131), (3480.0, 917.5418131)
```

919.5823298), (3540.0,

- 357 921.5868151), (3600.0, 923.5567552), (3660.0, 925.4935524), (3720.0,
- 358 927.3985305), (3780.0, 929.272941), (3840.0, 931.1179673), (3900.0,
- 359 932.9347297), (3960.0, 934.7242894), (4020.0, 936.4876525), (4080.0,
- 360 938.2257733), (4140.0, 939.9395579), (4200.0, 941.629867), (4260.0,
- 361 943.2975188), (4320.0, 944.9432918), (4380.0, 946.5679266), (4440.0,
- 362 948.1721291), (4500.0, 949.7565716), (4560.0, 951.3218955), (4620.0,
- 363 952.8687125), (4680.0, 954.3976068), (4740.0, 955.9091364), (4800.0,
- 364 957.4038344), (4860.0, 958.8822108), (4920.0, 960.3447534), (4980.0,
- 365 961.7919291), (5040.0, 963.224185), (5100.0, 964.6419498), (5160.0,
- 366 966.0456339), (5220.0, 967.4356315), (5280.0, 968.8123203), (5340.0,
- 367 970.1760633), (5400.0, 971.5272087), (5460.0, 972.8660913), (5520.0,
- 368 974.1930328), (5580.0, 975.5083426), (5640.0, 976.8123183), (5700.0,
- 369 978.1052462), (5760.0, 979.387402), (5820.0, 980.6590513), (5880.0,
- 370 981.9204499), (5940.0, 983.1718443), (6000.0, 984.4134723), (6060.0,
- 371 985.6455631), (6120.0, 986.8683381), (6180.0, 988.0820108), (6240.0,
- 372 989.2867873), (6300.0, 990.4828668), (6360.0, 991.6704417), (6420.0,
- 373 992.849698), (6480.0, 994.0208155), (6540.0, 995.1839681), (6600.0,
- 374 996.339324), (6660.0, 997.4870459), (6720.0, 998.6272916), (6780.0,
- 375 999.7602136), (6840.0, 1000.88596), (6900.0, 1002.004673), (6960.0,
- 376 1003.116492), (7020.0, 1004.221553), (7080.0, 1005.319984), (7140.0,
- 377 1006.411913), (7200.0, 1007.497462), (7260.0, 1008.576751), (7320.0,
- 378 1009.649895), (7380.0, 1010.717007), (7440.0, 1011.778196), (7500.0,

```
379 1012.833568), (7560.0, 1013.883225), (7620.0, 1014.927268), (7680.0,
```

- 380 1015.965794), (7740.0, 1016.998898), (7800.0, 1018.026672), (7860.0,
- 381 1019.049206), (7920.0, 1020.066586), (7980.0, 1021.078898), (8040.0,
- 382 1022.086223), (8100.0, 1023.088643), (8160.0, 1024.086236), (8220.0,
- 383 1025.079079), (8280.0, 1026.067244), (8340.0, 1027.050806), (8400.0,
- 384 1028.029835), (8460.0, 1029.004399), (8520.0, 1029.974566), (8580.0,
- 385 1030.940401), (8640.0, 1031.901968), (8700.0, 1032.859331), (8760.0,
- 386 1033.812549), (8820.0, 1034.761682), (8880.0, 1035.706788), (8940.0,
- 387 1036.647924), (9000.0, 1037.585145), (9060.0, 1038.518505), (9120.0,
- 388 1039.448058), (9180.0, 1040.373854), (9240.0, 1041.295944), (9300.0,
- 389 1042.214379), (9360.0, 1043.129205), (9420.0, 1044.04047), (9480.0,
- 390 1044.948221), (9540.0, 1045.852502), (9600.0, 1046.753358), (9660.0,
- 391 1047.650832), (9720.0, 1048.544966), (9780.0, 1049.435802), (9840.0,
- 392 1050.32338), (9900.0, 1051.20774), (9960.0, 1052.088921), (10020.0,
- 393 1052.966961), (10080.0, 1053.841898), (10140.0, 1054.713767), (10200.0,
- 394 1055.582606), (10260.0, 1056.448448), (10320.0, 1057.311329), (10380.0,
- 395 1058.171282), (10440.0, 1059.028341), (10500.0, 1059.882537), (10560.0,
- 396 1060.733903), (10620.0, 1061.58247), (10680.0, 1062.428268), (10740.0,
- 397 1063.271329), (10800.0, 1064.11168), (10860.0, 1064.949352), (10920.0,
- 398 1065.784372), (10980.0, 1066.616769), (11040.0, 1067.44657), (11100.0,
- 399 1068.273802), (11160.0, 1069.098491), (11220.0, 1069.920664), (11280.0,
- 400 1070.740345), (11340.0, 1071.55756), (11400.0, 1072.372334), (11460.0,

```
401
         1073.184691), (11520.0, 1073.994654), (11580.0, 1073.994654)
            1074.802247), (11640.0,
402
         1075.607493), (11700.0, 1076.410414), (11760.0, 1076.410414)
            1077.211033), (11820.0,
403
         1078.009372), (11880.0, 1078.805452), (11940.0, 1078.805452)
            1079.599294), (12000.0,
404
         1080.390919), (12060.0, 1081.180347), (12120.0, 1081.180347)
            1081.967599), (12180.0,
         1082.752694), (12240.0, 1083.535652), (12300.0,
405
            1084.316493), (12360.0,
         1085.095234), (12420.0, 1085.871895), (12480.0,
406
            1086.646493), (12540.0,
         1087.419047), (12600.0, 1088.189575), (12660.0, 1088.189575)
407
            1088.958094), (12720.0,
         1089.724622), (12780.0, 1090.489174), (12840.0, 1090.489174)
408
            1091.251768), (12900.0,
         1092.012421), (12960.0, 1092.771148), (13020.0, 1092.771148)
409
            1093.527966), (13080.0,
         1094.28289), (13140.0, 1095.035936), (13200.0, 1095.035936)
410
            1095.787118), (13260.0,
411
         1096.536453), (13320.0, 1097.283955), (13380.0, 1097.283955)
            1098.029638), (13440.0,
412
         1098.773518), (13500.0, 1099.515608), (13560.0, 1099.515608)
            1100.255922), (13620.0,
         1100.994474), (13680.0, 1101.731278), (13740.0, 1101.731278)
413
            1102.466347), (13800.0,
         1103.199695), (13860.0, 1103.931334), (13920.0, 1103.931334)
414
            1104.661278), (13980.0,
415
         1105.38954), (14040.0, 1106.116131), (14100.0,
            1106.841064), (14160.0,
         1107.564353), (14220.0, 1108.286007), (14280.0, 1108.286007)
416
            1109.006041), (14340.0,
         1109.724465), (14400.0, 1110.441291)), name='Fire -240',
417
            smooth=
418
         SOLVER_DEFAULT, timeSpan=STEP)
419
420 #Flim_Condition
421
422 mdb.models['Model-1'].FilmConditionProp(dependencies=0, name
       ='F', property = ((0.025, ), ), temperature Dependency = OFF)
    mdb. models ['Model-1']. FilmConditionProp (dependencies = 0, name
       ='NF', property = ((0.008, ), ), temperature Dependency = OFF)
424
425 #Surface_Create
426
```

```
427 mdb. models ['Model-1']. rootAssembly. Surface (name='F-1',
       side1Faces=mdb.models['Model-1'].rootAssembly.instances['
       Beam-1']. faces. findAt (((0.5*C1, -0.5*D, 0.5*H), ),
       ((-0.5*C1, -0.5*D, 0.5*H), ), ((0.5*C+C1, -0.5*D, 0.5*H),
        ), ((-0.5*C-C1, -0.5*D, 0.5*H), )
428 mdb. models ['Model-1']. rootAssembly. Surface (name='F-2',
       side1Faces=mdb.models['Model-1'].rootAssembly.instances['
      Beam-1']. faces. findAt(((-0.5*W, 0.5*C2, 0.5*H), ),
       ((-0.5*W, -0.5*C2, 0.5*H), ), ((-0.5*W, 0.5*C+C2, 0.5*H),
        ), ((-0.5*W, -0.5*C-C2, 0.5*H), ), ((0.5*C1, -0.5*D,
       0.5*H), ), ((-0.5*C1, -0.5*D, 0.5*H), ), ((0.5*C+C1,
       -0.5*D, 0.5*H), ), ((-0.5*C-C1, -0.5*D, 0.5*H), ), ))
429 mdb. models ['Model-1']. rootAssembly. Surface (name='F-3',
       side1Faces=mdb.models['Model-1'].rootAssembly.instances['
      Beam-1']. faces. findAt (((0.5*W, 0.5*C2, 0.5*H), ), ((0.5*W, 0.5*C2, 0.5*H))
       (0.5*C2, 0.5*H), ((0.5*W, 0.5*C+C2, 0.5*H), )
       ((0.5*W, -0.5*C-C2, 0.5*H), ), ((0.5*C1, -0.5*D, 0.5*H),
       ), ((-0.5*C1, -0.5*D, 0.5*H), ), ((0.5*C+C1, -0.5*D, 0.5*)
      H), ), ((-0.5*C-C1, -0.5*D, 0.5*H), ), ((-0.5*W, 0.5*C2, 
       0.5*H), ), ((-0.5*W, -0.5*C^2, 0.5*H), ), ((-0.5*W, 0.5*C^2, 0.5*H)
      C2, 0.5*H), ((-0.5*W, -0.5*C-C2, 0.5*H), ),
430 mdb. models ['Model-1']. rootAssembly. Surface (name='F-4',
       side1Faces=mdb.models['Model-1'].rootAssembly.instances['
      Beam-1']. faces. findAt (((0.5*W, 0.5*C2, 0.5*H), ), ((0.5*W
       , -0.5*C2, 0.5*H), ), ((0.5*W, 0.5*C+C2, 0.5*H), ),
       ((0.5*W, -0.5*C-C2, 0.5*H), ), ((0.5*C1, 0.5*D, 0.5*H), )
       , ((-0.5*C1, 0.5*D, 0.5*H), ), ((0.5*C+C1, 0.5*D, 0.5*H),
        ), ((-0.5*C-C1, 0.5*D, 0.5*H), ), ((-0.5*W, 0.5*C2, 0.5*)
      H), ), ((-0.5*W, -0.5*C2, 0.5*H), ), ((-0.5*W, 0.5*C+C2,
       0.5*H), ), ((-0.5*W, -0.5*C-C2, 0.5*H), ), ((0.5*C1,
       -0.5*D, 0.5*H), ), ((-0.5*C1, -0.5*D, 0.5*H), ), ((0.5*C+
       C1, -0.5*D, 0.5*H), ((-0.5*C-C1, -0.5*D, 0.5*H), )
431 mdb. models ['Model-1']. rootAssembly. Surface (name='NF-1',
       side1Faces=mdb.models['Model-1'].rootAssembly.instances['
      Beam-1']. faces. findAt(((0.5*C1, 0.5*C2, H), ), ((-0.5*C1,
        0.5*C2, H), ), ((0.5*C1, -0.5*C2, H), ), ((-0.5*C1,
       -0.5*C2, H), ), ((0.5*C+C1, 0.5*C2, H), ), ((-0.5*C-C1,
       0.5*C2, H), ), ((0.5*C+C1, -0.5*C2, H), ), ((-0.5*C-C1,
       -0.5*C2, H), ), ((0.5*C1, 0.5*C+C2, H), ), ((-0.5*C1,
       0.5*C+C2, H), ), ((0.5*C1, -0.5*C-C2, H), ), ((-0.5*C1,
       -0.5*C-C2, H), ), ((0.5*C+C1, 0.5*C+C2, H), ), ((-0.5*C-
      C1, 0.5*C+C2, H), ), ((0.5*C+C1, -0.5*C-C2, H), ),
       ((-0.5*C-C1, -0.5*C-C2, H), ), ((0.5*C1, 0.5*C2, 0), ),
       ((-0.5*C1, 0.5*C2, 0), ), ((0.5*C1, -0.5*C2, 0), ),
       ((-0.5*C1, -0.5*C2, 0), ), ((0.5*C+C1, 0.5*C2, 0), ),
```

```
((-0.5*C-C1, 0.5*C2, 0), ), ((0.5*C+C1, -0.5*C2, 0), ),
       ((-0.5*C-C1, -0.5*C2, 0), ), ((0.5*C1, 0.5*C+C2, 0), ),
       ((-0.5*C1, 0.5*C+C2, 0), ), ((0.5*C1, -0.5*C-C2, 0), ),
       ((-0.5*C1, -0.5*C-C2, 0), ), ((0.5*C+C1, 0.5*C+C2, 0), ),
       ((-0.5*C-C1, 0.5*C+C2, 0), ((0.5*C+C1, -0.5*C-C2, 0),
        ), ((-0.5*C-C1, -0.5*C-C2, 0), ), ((0.5*W, 0.5*C2, 0.5*H)
       ), ), ((0.5*W, -0.5*C2, 0.5*H), ), ((0.5*W, 0.5*C+C2,
       0.5*H), ), ((0.5*W, -0.5*C-C2, 0.5*H), ), ((0.5*C1, 0.5*D
       (0.5*H), ((-0.5*C1, 0.5*D, 0.5*H), ((0.5*C+C1, 0.5*D)
       0.5*D, 0.5*H), ), ((-0.5*C-C1, 0.5*D, 0.5*H), ), ((-0.5*W)
       (0.5*C2, 0.5*H), ((-0.5*W, -0.5*C2, 0.5*H), )
       ((-0.5*W, 0.5*C+C2, 0.5*H), ), ((-0.5*W, -0.5*C-C2, 0.5*H)
       ), ), ))
432 mdb. models ['Model-1']. rootAssembly. Surface (name='NF-2',
       side1Faces=mdb.models['Model-1'].rootAssembly.instances['
      Beam-1']. faces. findAt(((0.5*C1, 0.5*C2, H), ), ((-0.5*C1,
        0.5*C2, H), ), ((0.5*C1, -0.5*C2, H), ), ((-0.5*C1,
       -0.5*C2, H), ), ((0.5*C+C1, 0.5*C2, H), ), ((-0.5*C-C1,
       0.5*C2, H), ), ((0.5*C+C1, -0.5*C2, H), ), ((-0.5*C-C1,
       -0.5*C2, H), ), ((0.5*C1, 0.5*C+C2, H), ), ((-0.5*C1,
       0.5*C+C2, H), ), ((0.5*C1, -0.5*C-C2, H), ), ((-0.5*C1,
       -0.5*C-C2, H), ), ((0.5*C+C1, 0.5*C+C2, H), ), ((-0.5*C-
      C1, 0.5*C+C2, H), ), ((0.5*C+C1, -0.5*C-C2, H), ),
       ((-0.5*C-C1, -0.5*C-C2, H), ), ((0.5*C1, 0.5*C2, 0), ),
       ((-0.5*C1, 0.5*C2, 0), ), ((0.5*C1, -0.5*C2, 0), ),
       ((-0.5*C1, -0.5*C2, 0), ), ((0.5*C+C1, 0.5*C2, 0), ),
       ((-0.5*C-C1, 0.5*C2, 0), ), ((0.5*C+C1, -0.5*C2, 0), ),
       ((-0.5*C-C1, -0.5*C2, 0), ), ((0.5*C1, 0.5*C+C2, 0), ),
       ((-0.5*C1, 0.5*C+C2, 0), ), ((0.5*C1, -0.5*C-C2, 0), ),
       ((-0.5*C1, -0.5*C-C2, 0), ), ((0.5*C+C1, 0.5*C+C2, 0), ),
        ((-0.5*C-C1, 0.5*C+C2, 0), ), ((0.5*C+C1, -0.5*C-C2, 0),
        ), ((-0.5*C-C1, -0.5*C-C2, 0), ), ((0.5*W, 0.5*C2, 0.5*H)
       ), ), ((0.5*W, -0.5*C2, 0.5*H), ), ((0.5*W, 0.5*C+C2,
       0.5*H), ), ((0.5*W, -0.5*C-C2, 0.5*H), ), ((0.5*C1, 0.5*D
       (0.5*H), ((-0.5*C1, 0.5*D, 0.5*H), ((0.5*C+C1, 0.5*D)
       0.5*D, 0.5*H), ((-0.5*C-C1, 0.5*D, 0.5*H), ),
433 mdb. models ['Model-1']. rootAssembly. Surface (name='NF-3',
       side1Faces=mdb.models['Model-1'].rootAssembly.instances['
      Beam-1']. faces. findAt(((0.5*C1, 0.5*C2, H), )), ((-0.5*C1,
       0.5*C2, H), ), ((0.5*C1, -0.5*C2, H), ), ((-0.5*C1,
       -0.5*C2, H), ), ((0.5*C+C1, 0.5*C2, H), ), ((-0.5*C-C1,
       0.5*C2, H), ), ((0.5*C+C1, -0.5*C2, H), ), ((-0.5*C-C1,
       -0.5*C2, H), ), ((0.5*C1, 0.5*C+C2, H), ), ((-0.5*C1,
       0.5*C+C2, H), ), ((0.5*C1, -0.5*C-C2, H), ), ((-0.5*C1,
       -0.5*C-C2, H), ), ((0.5*C+C1, 0.5*C+C2, H), ), ((-0.5*C-C2, H), )
```

```
C1, 0.5*C+C2, H), ), ((0.5*C+C1, -0.5*C-C2, H), ),
       ((-0.5*C-C1, -0.5*C-C2, H), ), ((0.5*C1, 0.5*C2, 0), ),
       ((-0.5*C1, 0.5*C2, 0), ), ((0.5*C1, -0.5*C2, 0), ),
       ((-0.5*C1, -0.5*C2, 0), ), ((0.5*C+C1, 0.5*C2, 0), ),
       ((-0.5*C-C1, 0.5*C2, 0), ), ((0.5*C+C1, -0.5*C2, 0), ),
       ((-0.5*C-C1, -0.5*C2, 0), ), ((0.5*C1, 0.5*C+C2, 0), ),
       ((-0.5*C1, 0.5*C+C2, 0), ), ((0.5*C1, -0.5*C-C2, 0), ),
       ((-0.5*C1, -0.5*C-C2, 0), ), ((0.5*C+C1, 0.5*C+C2, 0), ),
        ((-0.5*C-C1, 0.5*C+C2, 0), ), ((0.5*C+C1, -0.5*C-C2, 0),
       ), ((-0.5*C-C1, -0.5*C-C2, 0), ), ((0.5*C1, 0.5*D, 0.5*H)
       ), ), ((-0.5*C1, 0.5*D, 0.5*H), ), ((0.5*C+C1, 0.5*D,
       0.5*H), ), ((-0.5*C-C1, 0.5*D, 0.5*H), ), ))
434 mdb. models ['Model-1']. rootAssembly. Surface (name='NF-4',
       side1Faces=mdb.models['Model-1'].rootAssembly.instances['
      Beam-1']. faces. findAt(((0.5*C1, 0.5*C2, H), ), ((-0.5*C1, 0.5*C2, H))
        0.5*C2, H), ), ((0.5*C1, -0.5*C2, H), ), ((-0.5*C1,
       -0.5*C2, H), ), ((0.5*C+C1, 0.5*C2, H), ), ((-0.5*C-C1,
       0.5*C2, H), ), ((0.5*C+C1, -0.5*C2, H), ), ((-0.5*C-C1, -0.5*C)
       -0.5*C2, H), ), ((0.5*C1, 0.5*C+C2, H), ), ((-0.5*C1,
       0.5*C+C2, H), ), ((0.5*C1, -0.5*C-C2, H), ), ((-0.5*C1,
       -0.5*C-C2, H), ), ((0.5*C+C1, 0.5*C+C2, H), ), ((-0.5*C-
      C1, 0.5*C+C2, H), ), ((0.5*C+C1, -0.5*C-C2, H), ),
       ((-0.5*C-C1, -0.5*C-C2, H), ), ((0.5*C1, 0.5*C2, 0), ),
       ((-0.5*C1, 0.5*C2, 0), ), ((0.5*C1, -0.5*C2, 0), ),
       ((-0.5*C1, -0.5*C2, 0), ), ((0.5*C+C1, 0.5*C2, 0), ),
       ((-0.5*C-C1, 0.5*C2, 0), ), ((0.5*C+C1, -0.5*C2, 0), ),
       ((-0.5*C-C1, -0.5*C2, 0), ), ((0.5*C1, 0.5*C+C2, 0), ),
       ((-0.5*C1, 0.5*C+C2, 0), ), ((0.5*C1, -0.5*C-C2, 0), ),
       ((-0.5*C1, -0.5*C-C2, 0), ), ((0.5*C+C1, 0.5*C+C2, 0), ),
        ((-0.5*C-C1, 0.5*C+C2, 0), ), ((0.5*C+C1, -0.5*C-C2, 0),
        ), ((-0.5*C-C1, -0.5*C-C2, 0), ), )
435
436 #Interaction_Tie
   #Change_If_Need
438
   mdb. models ['Model-1']. FilmCondition (createStepName='Fire',
       definition=
        PROPERTY_REF, interactionProperty='F', name='F',
439
           sinkAmplitude='Fire-240',
440
        sinkDistributionType=UNIFORM, sinkFieldName='',
           sinkTemperature = 1.0,
441
        surface=mdb.models['Model-1'].rootAssembly.surfaces['F-4
442 mdb. models ['Model-1']. FilmCondition (createStepName='Fire',
       definition =
```

```
PROPERTY_REF, interactionProperty='NF', name='NF',
443
           sinkAmplitude='Fire -240'
        , sinkDistributionType=UNIFORM, sinkFieldName='',
444
           sinkTemperature = 1.0,
        surface=mdb.models['Model-1'].rootAssembly.surfaces['NF
445
           -4'])
    mdb. models ['Model-1']. RadiationToAmbient (ambientTemperature
446
447
        ambientTemperatureAmp='Fire-240', createStepName='Fire',
            distribution Type =
        UNIFORM, emissivity = 0.8, field = '', name = 'FR',
448
           radiation Type = AMBIENT,
        surface=mdb.models['Model-1'].rootAssembly.surfaces['F-4
449
           '])
450
451
   mdb. models ['Model-1']. rootAssembly. Set (edges=
        mdb.models['Model-1'].rootAssembly.instances['Beam-1'].
452
           edges.findAt(((
        C1, -C2, 0.75*H), )), name='B1')
453
    mdb. models ['Model-1']. rootAssembly. Set (edges=
454
455
        mdb.models['Model-1'].rootAssembly.instances['Beam-1'].
           edges.findAt(((
456
        C1, C2, 0.75*H), )), name='B3')
   mdb. models ['Model-1']. rootAssembly. Set (edges=
457
        mdb.models['Model-1'].rootAssembly.instances['Beam-1'].
458
           edges.findAt(((
        -C1, C2, 0.75*H), )), name='B5')
459
    mdb. models ['Model-1']. rootAssembly. Set (edges=
460
        mdb.models['Model-1'].rootAssembly.instances['Beam-1'].
461
           edges.findAt(((
        -C1, -C2, 0.75*H), )), name='B7')
462
   mdb.models['Model-1'].rootAssembly.Set(edges=
463
        mdb. models ['Model-1']. rootAssembly.instances ['Bar-1-lin
464
           -2-1']. edges . findAt (
465
        ((C1, -C2, 0.25*H), )), name='B2')
   mdb.models['Model-1'].rootAssembly.Set(edges=
466
        mdb. models ['Model-1']. rootAssembly.instances ['Bar-1-lin
467
           -1-2-lin -2-1']. edges.findAt(
        ((C1, C2, 0.25*H), )), name='B4')
468
   mdb. models ['Model-1']. rootAssembly. Set (edges=
469
        mdb. models ['Model-1']. rootAssembly.instances ['Bar-1-lin
470
           -1-2']. edges . findAt (
471
        ((-C1, C2, 0.25*H), )), name='B6')
472 mdb. models ['Model-1']. rootAssembly. Set (edges=
```

```
473
        mdb.models['Model-1'].rootAssembly.instances['Bar-1'].
           edges.findAt(((
474
        -C1, -C2, 0.25*H), )), name='B8')
475
476
   mdb. models ['Model-1']. Tie (adjust=ON, master=
477
        mdb.models['Model-1'].rootAssembly.sets['B1'], name='T1'
478
        positionToleranceMethod=COMPUTED, slave=
479
        mdb.models['Model-1'].rootAssembly.sets['B2'], thickness
           =ON, tieRotations=
480
        ON)
481
482
   mdb. models ['Model-1']. Tie (adjust=ON, master=
        mdb.models['Model-1'].rootAssembly.sets['B3'], name='T2'
483
484
        positionToleranceMethod=COMPUTED, slave=
        mdb.models['Model-1'].rootAssembly.sets['B4'], thickness
485
           =ON, tieRotations=
486
        ON)
487
488
   mdb. models ['Model-1']. Tie (adjust=ON, master=
        mdb.models['Model-1'].rootAssembly.sets['B5'], name='T3'
489
490
        positionToleranceMethod=COMPUTED, slave=
491
        mdb.models['Model-1'].rootAssembly.sets['B6'], thickness
           =ON, tieRotations=
492
        ON)
493
494
   mdb. models ['Model-1']. Tie (adjust=ON, master=
        mdb.models['Model-1'].rootAssembly.sets['B7'], name='T4'
495
        positionToleranceMethod=COMPUTED, slave=
496
        mdb.models['Model-1'].rootAssembly.sets['B8'], thickness
497
           =ON, tieRotations=
498
        ON)
499
500
501 #Mesh
502
503 mdb. models ['Model-1']. parts ['Bar']. seedPart (deviationFactor
       =0.1, minSizeFactor=0.1, size=d)
   mdb. models ['Model-1']. parts ['Bar']. generateMesh()
504
505 mdb. models ['Model-1']. parts ['Bar']. setElementType (elemTypes
       =(ElemType(elemCode=DC1D2, elemLibrary=STANDARD), ),
       regions = (mdb. models ['Model-1']. parts ['Bar']. edges. find At
```

```
(((0.5*H, 0.0, 0.0), )), )
506
507
   mdb. models ['Model-1']. parts ['Beam']. seedPart (deviationFactor
       =0.1, minSizeFactor=0.1, size=d)
508 mdb. models ['Model-1']. parts ['Beam']. generateMesh()
509 mdb. models ['Model-1']. parts ['Beam']. setElementType (elemTypes
       =(ElemType(elemCode=DC3D8, elemLibrary=STANDARD),
       ElemType(elemCode=DC3D6, elemLibrary=STANDARD), ElemType(
       elemCode=DC3D4, elemLibrary=STANDARD)), regions=(mdb.
       models ['Model-1']. parts ['Beam']. cells . findAt (((0.5 * C1,
       0.5*C2, H), ), ((-0.5*C1, 0.5*C2, H), ), ((0.5*C1, -0.5*
       C2, H), ((-0.5*C1, -0.5*C2, H), ), ((0.5*C1, 0.5*D, H))
       (0.5*C1, -0.5*D, H), ((-0.5*C1, 0.5*D, H), ),
       ((-0.5*C1, -0.5*D, H), ), ((1.05*C1, 0.5*C2, H), ),
       ((1.05*C1, -0.5*C2, H), ), ((-1.05*C1, 0.5*C2, H), ),
       ((-1.05*C1, -0.5*C2, H), ), ((0.45*W, 0.45*D, H), ),
       ((0.45*W, -0.45*D, H), ), ((-0.45*W, 0.45*D, H), ),
       ((-0.45*W, -0.45*D, H), ), )
510 mdb.models['Model-1'].rootAssembly.regenerate()
511
512 #Interaction_Node
513
514 mdb. models ['Model-1']. rootAssembly. DatumPointByCoordinate (
       coords = (0.0, 0.0, 0.5*H)
515 mdb. models ['Model-1']. rootAssembly. DatumPointByCoordinate (
       coords = (-C1, C2, 0.5*H)
   mdb. models ['Model-1']. rootAssembly. DatumPointByCoordinate (
516
       coords = (C1, C2, 0.5*H)
   mdb. models ['Model-1']. rootAssembly. DatumPointByCoordinate (
517
       coords = (-C1, -C2, 0.5*H)
   mdb. models ['Model-1']. rootAssembly. DatumPointByCoordinate (
518
       coords = (C1, -C2, 0.5*H)
519
520 #nodes_[element:node]
521
522 #mdb. models ['Model-1']. rootAssembly. Set (name='TC5-C', nodes=
       mdb.models['Model-1'].rootAssembly.instances['Beam-1'].
       nodes[1843:1844])
523 mdb. models ['Model-1']. rootAssembly. Set (name='TC1-B1', nodes=
       mdb.models['Model-1'].rootAssembly.instances['Bar-1-lin
       -2-1']. nodes [b:b+1])
524 mdb. models ['Model-1']. rootAssembly. Set (name='TC1-B2', nodes=
       mdb.models['Model-1'].rootAssembly.instances['Bar-1-lin
       -1-2-1 in -2-1']. nodes [b:b+1])
```

```
525 mdb. models ['Model-1']. rootAssembly. Set (name='TC1-B3', nodes=
      mdb.models['Model-1'].rootAssembly.instances['Bar-1-lin
       -1-2']. nodes [b:b+1])
526 mdb. models ['Model-1']. rootAssembly. Set (name='TC1-B4', nodes=
      mdb.models['Model-1'].rootAssembly.instances['Bar-1'].
       nodes[b:b+1]
527
528 #Multiple_Model
529
530 mdb. Model (name='Model-2', objectToCopy=mdb. models ['Model-1'
   mdb.models['Model-2'].interactions['F'].setValues(definition
531
      =PROPERTY_REF,
        interactionProperty='F', sinkAmplitude='Fire-240',
532
           sinkTemperature = 1.0,
533
        surface=mdb.models['Model-2'].rootAssembly.surfaces['F-3
    mdb.models['Model-2'].interactions['FR'].setValues(
534
       ambientTemperature = 1.0,
        ambientTemperatureAmp='Fire-240', distributionType=
535
           UNIFORM, emissivity = 0.8,
        field='', radiationType=AMBIENT, surface=
536
        mdb.models['Model-2'].rootAssembly.surfaces['F-3'])
537
   mdb.models['Model-2'].interactions['NF'].setValues(
538
       definition=PROPERTY_REF,
        interactionProperty='NF', sinkAmplitude='Fire-240',
539
           sinkTemperature = 1.0,
        surface=mdb.models['Model-2'].rootAssembly.surfaces['NF
540
           -3'])
541
542 mdb. Model (name='Model-3', objectToCopy=mdb. models ['Model-2'
       1)
543 mdb. models ['Model-3']. interactions ['F']. setValues (definition
      =PROPERTY_REF,
        interactionProperty='F', sinkAmplitude='Fire-240',
544
           sinkTemperature = 1.0,
        surface=mdb.models['Model-3'].rootAssembly.surfaces['F-2
545
           '1)
   mdb. models ['Model-3']. interactions ['FR']. setValues (
546
       ambientTemperature = 1.0,
        ambientTemperatureAmp='Fire-240', distributionType=
547
           UNIFORM, emissivity = 0.8,
548
        field='', radiationType=AMBIENT, surface=
        mdb.models['Model-3'].rootAssembly.surfaces['F-2'])
549
```

```
550 mdb. models ['Model-3']. interactions ['NF']. setValues (
       definition=PROPERTY_REF,
        interactionProperty='NF', sinkAmplitude='Fire-240',
551
           sinkTemperature = 1.0,
        surface=mdb.models['Model-3'].rootAssembly.surfaces['NF
552
           -2']
553
554 mdb. Model (name='Model-4', objectToCopy=mdb. models ['Model-3'
       1)
555
   mdb. models ['Model-4']. interactions ['F']. setValues (definition
       =PROPERTY_REF,
        interactionProperty='F', sinkAmplitude='Fire-240',
556
           sinkTemperature = 1.0,
        surface=mdb.models['Model-3'].rootAssembly.surfaces['F-1
557
           '1)
558
   mdb. models ['Model-4']. interactions ['FR']. setValues (
       ambientTemperature = 1.0,
        ambientTemperatureAmp='Fire-240', distributionType=
559
           UNIFORM, emissivity = 0.8,
        field='', radiationType=AMBIENT, surface=
560
561
        mdb.models['Model-3'].rootAssembly.surfaces['F-1'])
   mdb. models ['Model-4']. interactions ['NF']. setValues (
562
       definition=PROPERTY_REF,
        interactionProperty='NF', sinkAmplitude='Fire-240',
563
           sinkTemperature = 1.0,
564
        surface=mdb.models['Model-3'].rootAssembly.surfaces['NF
           -1'1)
565
566 #Job
567
   mdb.Job(atTime=None, contactPrint=OFF, description='',
       echoPrint=OFF,
        explicitPrecision=SINGLE, getMemoryFromAnalysis=True,
568
           history Print = OFF,
        memory=90, memoryUnits=PERCENTAGE, model='Model-1',
569
           modelPrint=OFF,
570
        multiprocessing Mode = DEFAULT, name = 'F-240-4s',
           nodalOutputPrecision=SINGLE,
        numCpus=8, numDomains=8, numGPUs=0, queue=None,
571
           resultsFormat=ODB, scratch=
        '', type=ANALYSIS, userSubroutine='', waitHours=0,
572
           waitMinutes = 0)
573
574 mdb. Job (atTime=None, contactPrint=OFF, description='',
       echoPrint=OFF,
```

```
575
        explicitPrecision=SINGLE, getMemoryFromAnalysis=True,
           history Print=OFF,
576
        memory=90, memoryUnits=PERCENTAGE, model='Model-2',
           modelPrint=OFF,
        multiprocessing Mode = DEFAULT, name = 'F-240-3s',
577
           nodalOutputPrecision=SINGLE,
        numCpus=8, numDomains=8, numGPUs=0, queue=None,
578
           resultsFormat=ODB, scratch=
579
        '', type=ANALYSIS, userSubroutine='', waitHours=0,
           waitMinutes = 0
580
581
   mdb.Job(atTime=None, contactPrint=OFF, description='',
       echoPrint=OFF,
        explicitPrecision=SINGLE, getMemoryFromAnalysis=True,
582
           history Print = OFF,
583
        memory=90, memoryUnits=PERCENTAGE, model='Model-3',
           modelPrint=OFF,
        multiprocessing Mode = DEFAULT, name = 'F-240-2s',
584
           nodalOutputPrecision=SINGLE,
        numCpus=8, numDomains=8, numGPUs=0, queue=None,
585
           resultsFormat=ODB, scratch=
        '', type=ANALYSIS, userSubroutine='', waitHours=0,
586
           waitMinutes = 0)
587
588
   mdb.Job(atTime=None, contactPrint=OFF, description='',
       echoPrint=OFF.
        explicitPrecision=SINGLE, getMemoryFromAnalysis=True,
589
           history Print=OFF,
590
        memory=90, memoryUnits=PERCENTAGE, model='Model-4',
           modelPrint=OFF,
591
        multiprocessing Mode = DEFAULT, name = 'F-240-1s',
           nodalOutputPrecision=SINGLE,
592
        numCpus=8, numDomains=8, numGPUs=0, queue=None,
           resultsFormat=ODB, scratch=
593
        '', type=ANALYSIS, userSubroutine='', waitHours=0,
           waitMinutes = 0
594
595 #mdb.jobs['F-240'].submit(consistencyChecking=OFF)
596
597 #Done
```