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

List of Figures

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.

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

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

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

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

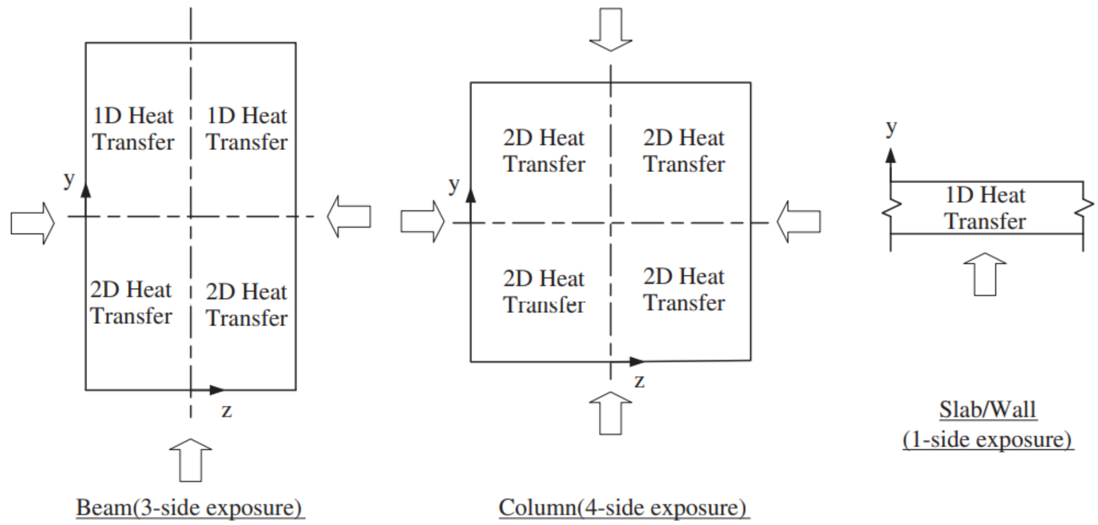

Figure 1: Separation of concrete member into areas for use in determining internal temperatures.

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.

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

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

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

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.

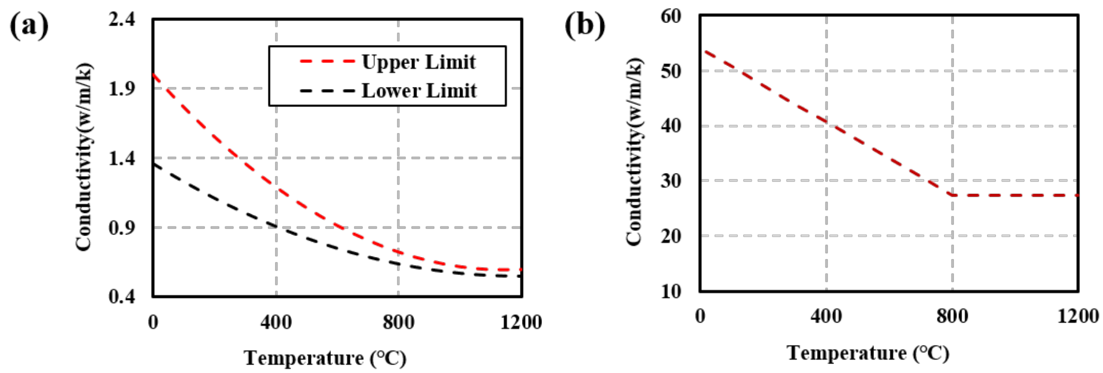

Figure 2: Thermal Property Conductivity as per EC-2. [1]

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 5.67×10^{-8} W/m^2k^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

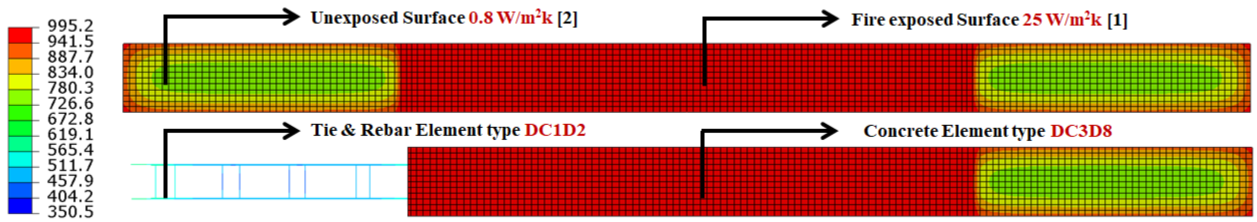

Figure 3: Fire-exposed finite element model

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.

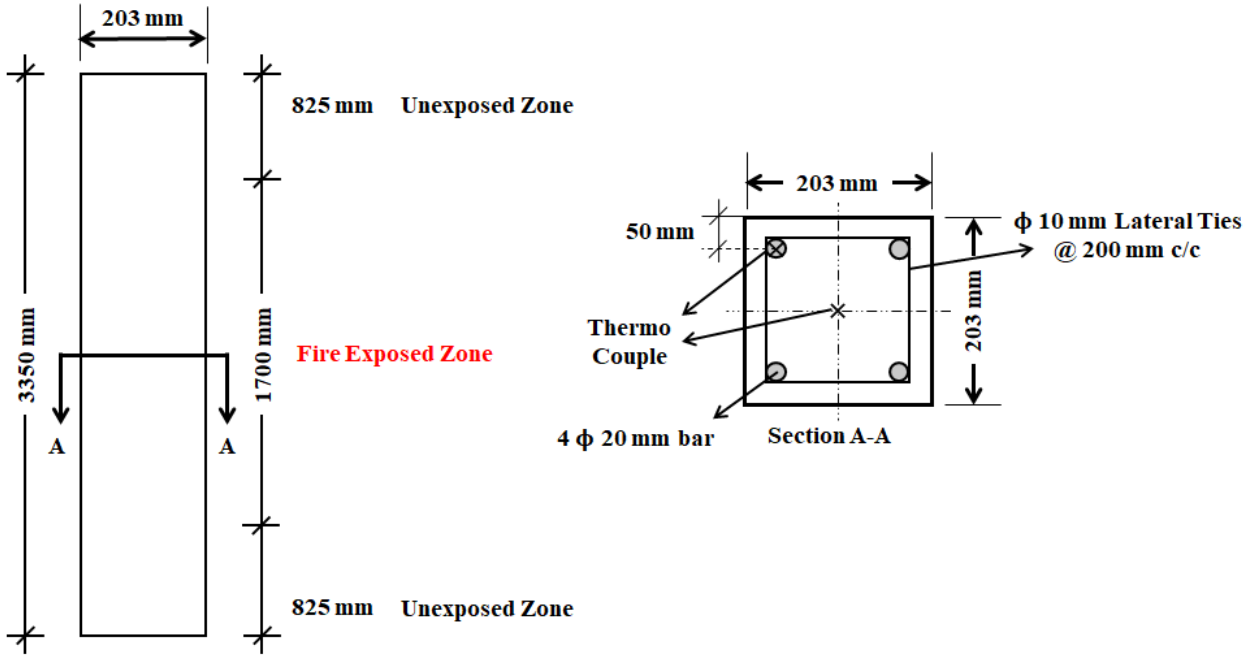

Figure 4: Dimensions and reinforcement details of RC columns.

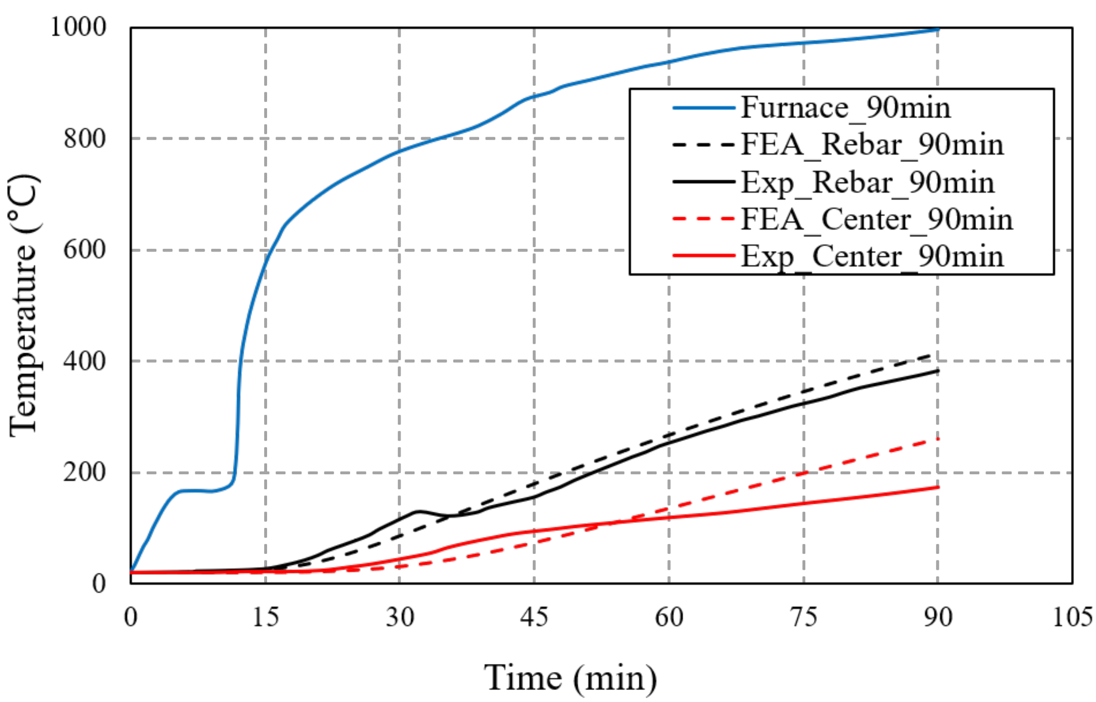

Figure 5: Comparison of predicted (FEA) and measured column temperatures during fire exposure. [2]

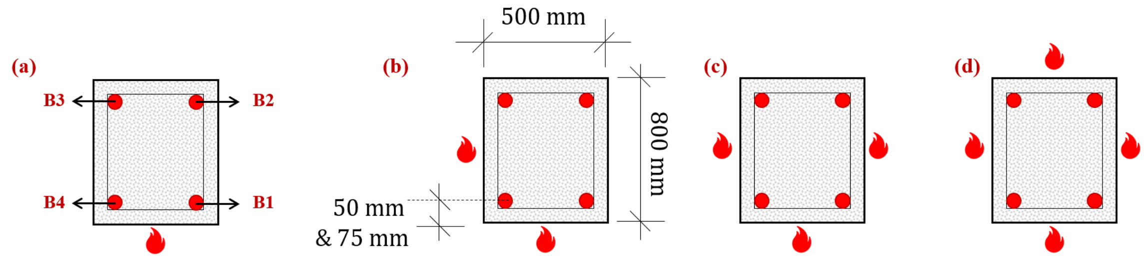

Figure 6: Selected specimens for study (a) one exposed side (b) two exposed sides (c) three exposed sides (d) four exposed sides to ASTM E119 [3] standard fire.

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.

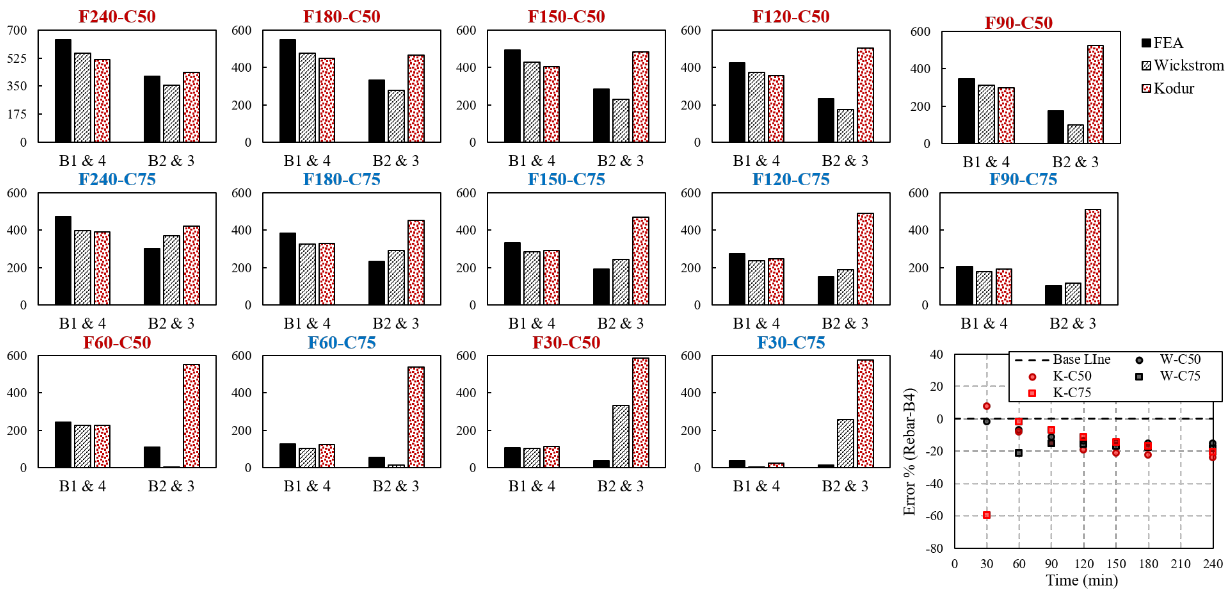

Figure 7: The temperature of Rebar exposed to fire on 1 side for different durations

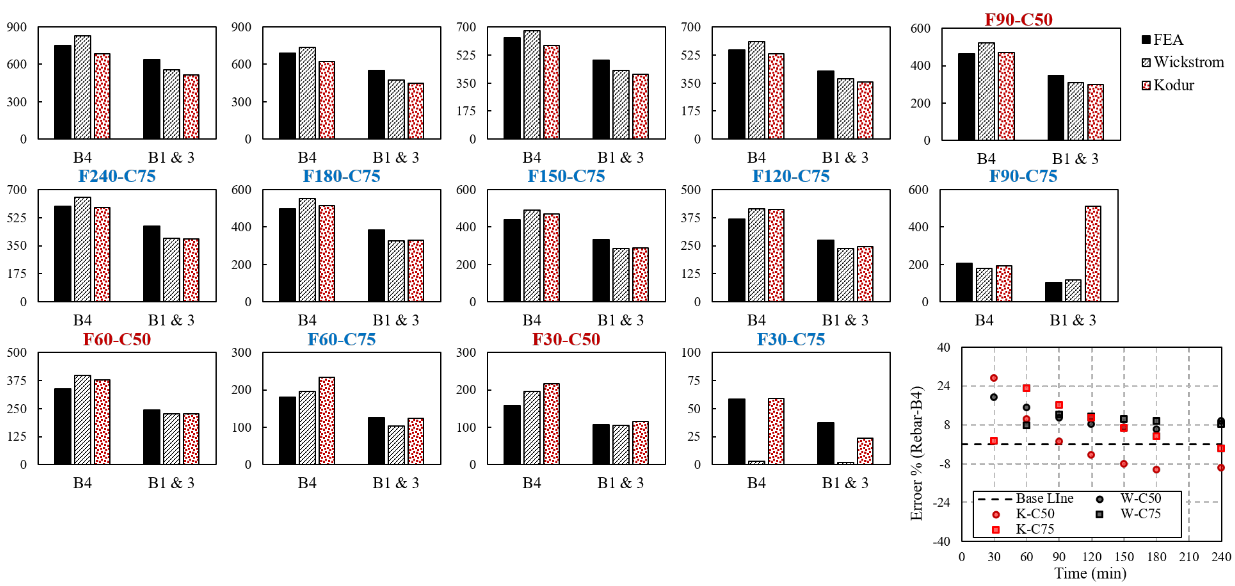

Figure 8: The temperature of Rebar exposed to fire on 2 side for different durations

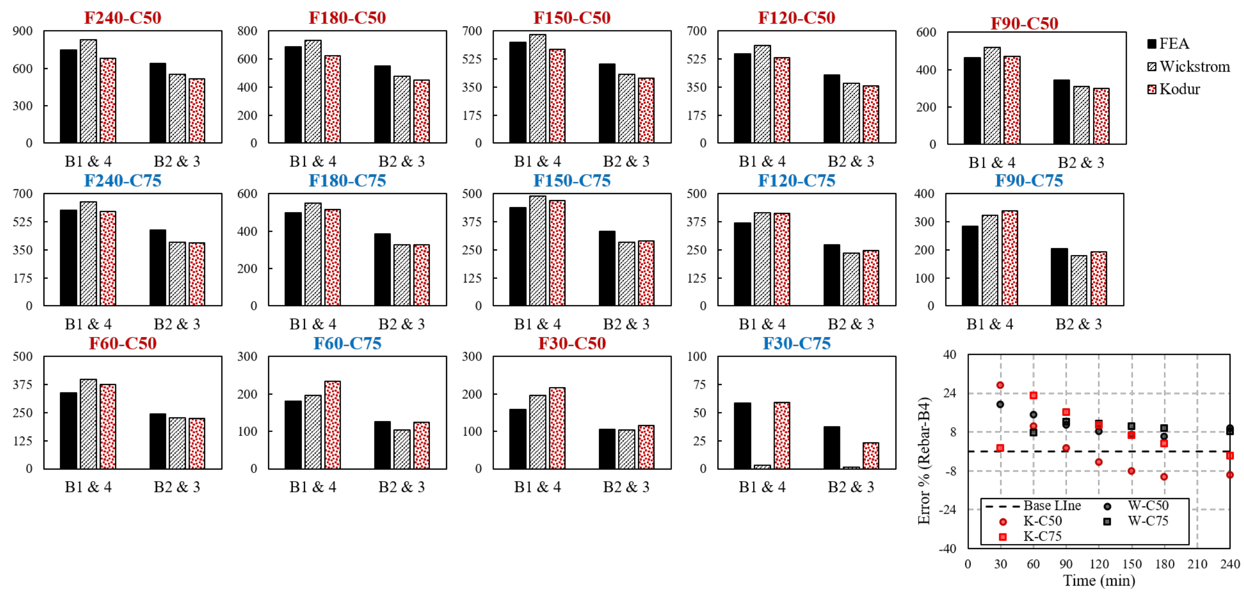

Figure 9: The temperature of Rebar exposed to fire on 3 side for different durations

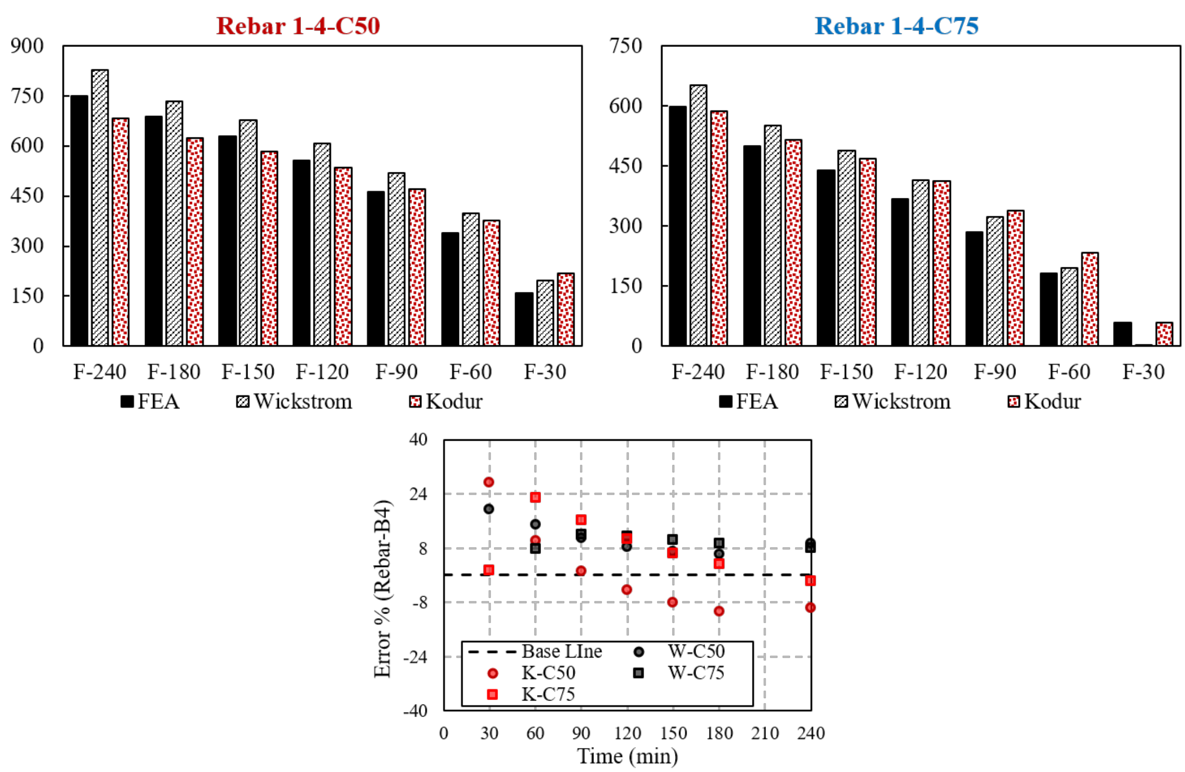

Figure 10: The temperature of Rebar exposed to fire on 3 side for different durations

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 .

```python
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from optimization import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *

session.journalOptions.setValues(replayGeometry=COORDINATE, recoverGeometry=COORDINATE)

#Create Part
W=500
D=500
H=4000
C=75
C1=0.5*W-C
C2=0.5*D-C
t=240 #input as min
d=25  #Mesh Size
b=H/(2*d)

mdb.models['Model-1'].setValues(absoluteZero=-273.16, stefanBoltzmann=5.667e-11)
```

```
mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=200.0)
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'].BaseSolidExtrude(depth=H, sketch=mdb.models['Model-1'].sketches['__profile__'])

mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=200.0)
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(0.0, 0.0), point2=(H, 0.0))
mdb.models['Model-1'].sketches['__profile__'].HorizontalConstraint(addUndoState=False, entity=mdb.models['Model-1'].sketches['__profile__'].geometry[2])
mdb.models['Model-1'].Part(dimensionality=THREE_D, name='Bar', type=DEFORMABLE_BODY)
mdb.models['Model-1'].parts['Bar'].BaseWire(sketch=mdb.models['Model-1'].sketches['__profile__'])

#Cell

mdb.models['Model-1'].parts['Beam'].DatumPlaneByPrincipalPlane(offset=0.0, principalPlane=YZPLANE)
mdb.models['Model-1'].parts['Beam'].DatumPlaneByPrincipalPlane(offset=0.0, principalPlane=XZPLANE)
mdb.models['Model-1'].parts['Beam'].DatumPlaneByPrincipalPlane(offset=C1, principalPlane=YZPLANE)
mdb.models['Model-1'].parts['Beam'].DatumPlaneByPrincipalPlane(offset=-C1, principalPlane=YZPLANE)
mdb.models['Model-1'].parts['Beam'].DatumPlaneByPrincipalPlane(offset=C2, principalPlane=XZPLANE)
mdb.models['Model-1'].parts['Beam'].DatumPlaneByPrincipalPlane(offset=-C2, principalPlane=XZPLANE)

#Cut

```

```
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])
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])
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[4])
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])
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),),), datumPlane=mdb.models['Model-1'].parts['Beam'].datums[6])
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),),), datumPlane=mdb.models['Model-1'].parts['Beam'].datums[7])

#Material

mdb.models['Model-1'].Material(name='Concrete')
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), (0.82, 500.0), (
    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)
```

```
mdb.models['Model-1'].materials['Concrete'].SpecificHeat(table=((900000000.0,
    20.0), (900000000.0, 100.0), (1000000000.0, 200.0), (1050000000.0, 300.0),
    (1100000000.0, 400.0), (1100000000.0, 500.0), (1100000000.0, 600.0), (
    1100000000.0, 700.0), (1100000000.0, 800.0), (1100000000.0, 900.0), (
    1100000000.0, 1000.0), (1100000000.0, 1100.0), (1100000000.0, 1200.0)),
    temperatureDependency=ON)

mdb.models['Model-1'].Material(name='Steel')
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, 400.0), (37.4, 500.0), (
    34.0, 600.0), (30.7, 700.0), (27.3, 800.0), (27.3, 900.0), (27.3, 1000.0),
    (27.3, 1100.0), (27.3, 1200.0)), temperatureDependency=ON)
mdb.models['Model-1'].materials['Steel'].SpecificHeat(table=((439801760.0,
    20.0), (487620000.0, 100.0), (529760000.0, 200.0), (564740000.0, 300.0), (
    605880000.0, 400.0), (666500000.0, 500.0), (760217391.3, 600.0), (
    1008157895.0, 700.0), (5000000000.0, 735.0), (803260869.6, 800.0), (
    650000000.0, 900.0), (650000000.0, 1000.0), (650000000.0, 1100.0), (
    650000000.0, 1200.0)), temperatureDependency=ON)

mdb.models['Model-1'].HomogeneousSolidSection(material='Concrete', name='C', thickness=None)
mdb.models['Model-1'].TrussSection(area=200.0, material='Steel', name='S')
mdb.models['Model-1'].parts['Beam'].SectionAssignment(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), ), ((-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), ), )), sectionName='C',
     thicknessAssignment=FROM_SECTION)
mdb.models['Model-1'].parts['Bar'].SectionAssignment(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',
    thicknessAssignment=FROM_SECTION)

#Assembly

mdb.models['Model-1'].rootAssembly.DatumCsysByDefault(
    CARTESIAN)
mdb.models['Model-1'].rootAssembly.Instance(dependent=ON,
    name='Bar-1', part=mdb.models['Model-1'].parts['Bar'])
mdb.models['Model-1'].rootAssembly.LinearInstancePattern(
    direction1=(1.0, 0.0, 0.0), direction2=(0.0, 1.0, 0.0),
    instanceList=('Bar-1', ), number1=1, number2=2, spacing1=
    H, spacing2=2*C2)
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'))
mdb.models['Model-1'].rootAssembly.LinearInstancePattern(
    direction1=(1.0, 0.0, 0.0), direction2=(0.0, 1.0, 0.0),
    instanceList=('Bar-1', 'Bar-1-lin-1-2'), number1=2,
    number2=1, spacing1=2*C1, spacing2=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-
    lin-2-1'), vector=(-C1, -C2, 0.0))


#Step

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)
mdb.models['Model-1'].fieldOutputRequests['F-Output-1'].
    setValues(variables=('NT', ))

#Amplitude
```

```
mdb.models['Model-1'].TabularAmplitude(data=((0.0, 20.0), (60.0, 332.5304157),
    (120.0, 426.044443), (180.0, 487.12866), (240.0, 532.516808), (300.0,
    568.4577615), (360.0, 598.0496271), (420.0, 623.0742538), (480.0,
    644.6569038), (540.0, 663.5558175), (600.0, 680.3069656), (660.0,
    695.303428), (720.0, 708.8420627), (780.0, 721.1524826), (840.0,
    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,
    803.9778734), (1440.0, 809.7725613), (1500.0, 815.2792562), (1560.0,
    820.5228938), (1620.0, 825.5255316), (1680.0, 830.3067617), (1740.0,
    834.8840538), (1800.0, 839.2730404)), name='Fire-30', smooth=SOLVER_DEFAULT
    , timeSpan=STEP)

mdb.models['Model-1'].TabularAmplitude(data=((0.0, 20.0), (60.0, 332.5304157),
    (120.0, 426.044443), (180.0, 487.12866), (240.0, 532.516808), (300.0,
    568.4577615), (360.0, 598.0496271), (420.0, 623.0742538), (480.0,
    644.6569038), (540.0, 663.5558175), (600.0, 680.3069656), (660.0,
    695.303428), (720.0, 708.8420627), (780.0, 721.1524826), (840.0,
    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,
    803.9778734), (1440.0, 809.7725613), (1500.0, 815.2792562), (1560.0,
    820.5228938), (1620.0, 825.5255316), (1680.0, 830.3067617), (1740.0,
```

```
    834.8840538), (1800.0, 839.2730404), (1860.0, 843.4877575), (1920.0,
    847.5408468), (1980.0, 851.4437275), (2040.0, 855.2067436), (2100.0,
    858.8392889), (2160.0, 862.349916), (2220.0, 865.746429), (2280.0,
    869.0359653), (2340.0, 872.2250658), (2400.0, 875.3197372), (2460.0,
    878.3255053), (2520.0, 881.2474635), (2580.0, 884.0903142), (2640.0,
    886.8584059), (2700.0, 889.5557668), (2760.0, 892.1861333), (2820.0,
    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,
    921.5868151), (3600.0, 923.5567552)), name='Fire-60', smooth=SOLVER_DEFAULT
    , timeSpan=STEP)

mdb.models['Model-1'].TabularAmplitude(data=((0.0, 20.0), (60.0, 332.5304157),
    (120.0, 426.044443), (180.0, 487.12866), (240.0, 532.516808), (300.0,
    568.4577615), (360.0, 598.0496271), (420.0, 623.0742538), (480.0,
    644.6569038), (540.0, 663.5558175), (600.0, 680.3069656), (660.0,
    695.303428), (720.0, 708.8420627), (780.0, 721.1524826), (840.0,
    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,
    803.9778734), (1440.0, 809.7725613), (1500.0, 815.2792562), (1560.0,
    820.5228938), (1620.0, 825.5255316), (1680.0, 830.3067617), (1740.0,
```

```
834.8840538), (1800.0, 839.2730404), (1860.0, 843.4877575), (1920.0,
847.5408468), (1980.0, 851.4437275), (2040.0, 855.2067436), (2100.0,
858.8392889), (2160.0, 862.349916), (2220.0, 865.746429), (2280.0,
869.0359653), (2340.0, 872.2250658), (2400.0, 875.3197372), (2460.0,
878.3255053), (2520.0, 881.2474635), (2580.0, 884.0903142), (2640.0,
886.8584059), (2700.0, 889.5557668), (2760.0, 892.1861333), (2820.0,
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,
921.5868151), (3600.0, 923.5567552), (3660.0, 925.4935524), (3720.0,
927.3985305), (3780.0, 929.272941), (3840.0, 931.1179673), (3900.0,
932.9347297), (3960.0, 934.7242894), (4020.0, 936.4876525), (4080.0,
938.2257733), (4140.0, 939.9395579), (4200.0, 941.629867), (4260.0,
943.2975188), (4320.0, 944.9432918), (4380.0, 946.5679266), (4440.0,
948.1721291), (4500.0, 949.7565716), (4560.0, 951.3218955), (4620.0,
952.8687125), (4680.0, 954.3976068), (4740.0, 955.9091364), (4800.0,
957.4038344), (4860.0, 958.8822108), (4920.0, 960.3447534), (4980.0,
961.7919291), (5040.0, 963.224185), (5100.0, 964.6419498), (5160.0,
966.0456339), (5220.0, 967.4356315), (5280.0, 968.8123203), (5340.0,
970.1760633), (5400.0, 971.5272087)), name='Fire-90', smooth=SOLVER_DEFAULT
, timeSpan=STEP)
```

```
mdb.models['Model-1'].TabularAmplitude(data=((0.0, 20.0), (60.0, 332.5304157),
    (120.0, 426.044443), (180.0, 487.12866), (240.0, 532.516808), (300.0,
    568.4577615), (360.0, 598.0496271), (420.0, 623.0742538), (480.0,
    644.6569038), (540.0, 663.5558175), (600.0, 680.3069656), (660.0,
    695.303428), (720.0, 708.8420627), (780.0, 721.1524826), (840.0,
    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,
    803.9778734), (1440.0, 809.7725613), (1500.0, 815.2792562), (1560.0,
    820.5228938), (1620.0, 825.5255316), (1680.0, 830.3067617), (1740.0,
    834.8840538), (1800.0, 839.2730404), (1860.0, 843.4877575), (1920.0,
    847.5408468), (1980.0, 851.4437275), (2040.0, 855.2067436), (2100.0,
    858.8392889), (2160.0, 862.349916), (2220.0, 865.746429), (2280.0,
    869.0359653), (2340.0, 872.2250658), (2400.0, 875.3197372), (2460.0,
    878.3255053), (2520.0, 881.2474635), (2580.0, 884.0903142), (2640.0,
    886.8584059), (2700.0, 889.5557668), (2760.0, 892.1861333), (2820.0,
    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,
    921.5868151), (3600.0, 923.5567552), (3660.0, 925.4935524), (3720.0,
    927.3985305), (3780.0, 929.272941), (3840.0, 931.1179673), (3900.0,
```

```
    932.9347297), (3960.0, 934.7242894), (4020.0,
        936.4876525), (4080.0,
    938.2257733), (4140.0, 939.9395579), (4200.0,
        941.629867), (4260.0,
    943.2975188), (4320.0, 944.9432918), (4380.0,
        946.5679266), (4440.0,
    948.1721291), (4500.0, 949.7565716), (4560.0,
        951.3218955), (4620.0,
    952.8687125), (4680.0, 954.3976068), (4740.0,
        955.9091364), (4800.0,
    957.4038344), (4860.0, 958.8822108), (4920.0,
        960.3447534), (4980.0,
    961.7919291), (5040.0, 963.224185), (5100.0,
        964.6419498), (5160.0,
    966.0456339), (5220.0, 967.4356315), (5280.0,
        968.8123203), (5340.0,
    970.1760633), (5400.0, 971.5272087), (5460.0,
        972.8660913), (5520.0,
    974.1930328), (5580.0, 975.5083426), (5640.0,
        976.8123183), (5700.0,
    978.1052462), (5760.0, 979.387402), (5820.0,
        980.6590513), (5880.0,
    981.9204499), (5940.0, 983.1718443), (6000.0,
        984.4134723), (6060.0,
    985.6455631), (6120.0, 986.8683381), (6180.0,
        988.0820108), (6240.0,
    989.2867873), (6300.0, 990.4828668), (6360.0,
        991.6704417), (6420.0,
    992.849698), (6480.0, 994.0208155), (6540.0,
        995.1839681), (6600.0,
    996.339324), (6660.0, 997.4870459), (6720.0,
        998.6272916), (6780.0,
    999.7602136), (6840.0, 1000.88596), (6900.0,
        1002.004673), (6960.0,
    1003.116492), (7020.0, 1004.221553), (7080.0,
        1005.319984), (7140.0,
    1006.411913), (7200.0, 1007.497462)), name='Fire-120',
        smooth=
    SOLVER_DEFAULT, timeSpan=STEP)

mdb.models['Model-1'].TabularAmplitude(data=((0.0, 20.0),
    (60.0, 332.5304157),
    (120.0, 426.044443), (180.0, 487.12866), (240.0,
        532.516808), (300.0,
```

```
568.4577615), (360.0, 598.0496271), (420.0, 623.0742538), (480.0,
644.6569038), (540.0, 663.5558175), (600.0, 680.3069656), (660.0,
695.303428), (720.0, 708.8420627), (780.0, 721.1524826), (840.0,
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,
803.9778734), (1440.0, 809.7725613), (1500.0, 815.2792562), (1560.0,
820.5228938), (1620.0, 825.5255316), (1680.0, 830.3067617), (1740.0,
834.8840538), (1800.0, 839.2730404), (1860.0, 843.4877575), (1920.0,
847.5408468), (1980.0, 851.4437275), (2040.0, 855.2067436), (2100.0,
858.8392889), (2160.0, 862.349916), (2220.0, 865.746429), (2280.0,
869.0359653), (2340.0, 872.2250658), (2400.0, 875.3197372), (2460.0,
878.3255053), (2520.0, 881.2474635), (2580.0, 884.0903142), (2640.0,
886.8584059), (2700.0, 889.5557668), (2760.0, 892.1861333), (2820.0,
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,
921.5868151), (3600.0, 923.5567552), (3660.0, 925.4935524), (3720.0,
927.3985305), (3780.0, 929.272941), (3840.0, 931.1179673), (3900.0,
932.9347297), (3960.0, 934.7242894), (4020.0, 936.4876525), (4080.0,
938.2257733), (4140.0, 939.9395579), (4200.0, 941.629867), (4260.0,
```

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

```
    1025.079079), (8280.0, 1026.067244), (8340.0,
        1027.050806), (8400.0,
    1028.029835), (8460.0, 1029.004399), (8520.0,
        1029.974566), (8580.0,
    1030.940401), (8640.0, 1031.901968), (8700.0,
        1032.859331), (8760.0,
    1033.812549), (8820.0, 1034.761682), (8880.0,
        1035.706788), (8940.0,
    1036.647924), (9000.0, 1037.585145)), name='Fire-150',
        smooth=
    SOLVER_DEFAULT, timeSpan=STEP)

mdb.models['Model-1'].TabularAmplitude(data=((0.0, 20.0),
    (60.0, 332.5304157),
    (120.0, 426.044443), (180.0, 487.12866), (240.0,
        532.516808), (300.0,
    568.4577615), (360.0, 598.0496271), (420.0, 623.0742538)
        , (480.0,
    644.6569038), (540.0, 663.5558175), (600.0, 680.3069656)
        , (660.0,
    695.303428), (720.0, 708.8420627), (780.0, 721.1524826),
        (840.0,
    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,
    803.9778734), (1440.0, 809.7725613), (1500.0,
        815.2792562), (1560.0,
    820.5228938), (1620.0, 825.5255316), (1680.0,
        830.3067617), (1740.0,
    834.8840538), (1800.0, 839.2730404), (1860.0,
        843.4877575), (1920.0,
    847.5408468), (1980.0, 851.4437275), (2040.0,
        855.2067436), (2100.0,
    858.8392889), (2160.0, 862.349916), (2220.0, 865.746429)
        , (2280.0,
    869.0359653), (2340.0, 872.2250658), (2400.0,
        875.3197372), (2460.0,
    878.3255053), (2520.0, 881.2474635), (2580.0,
        884.0903142), (2640.0,
    886.8584059), (2700.0, 889.5557668), (2760.0,
        892.1861333), (2820.0,
```

```
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,
921.5868151), (3600.0, 923.5567552), (3660.0, 925.4935524), (3720.0,
927.3985305), (3780.0, 929.272941), (3840.0, 931.1179673), (3900.0,
932.9347297), (3960.0, 934.7242894), (4020.0, 936.4876525), (4080.0,
938.2257733), (4140.0, 939.9395579), (4200.0, 941.629867), (4260.0,
943.2975188), (4320.0, 944.9432918), (4380.0, 946.5679266), (4440.0,
948.1721291), (4500.0, 949.7565716), (4560.0, 951.3218955), (4620.0,
952.8687125), (4680.0, 954.3976068), (4740.0, 955.9091364), (4800.0,
957.4038344), (4860.0, 958.8822108), (4920.0, 960.3447534), (4980.0,
961.7919291), (5040.0, 963.224185), (5100.0, 964.6419498), (5160.0,
966.0456339), (5220.0, 967.4356315), (5280.0, 968.8123203), (5340.0,
970.1760633), (5400.0, 971.5272087), (5460.0, 972.8660913), (5520.0,
974.1930328), (5580.0, 975.5083426), (5640.0, 976.8123183), (5700.0,
978.1052462), (5760.0, 979.387402), (5820.0, 980.6590513), (5880.0,
981.9204499), (5940.0, 983.1718443), (6000.0, 984.4134723), (6060.0,
985.6455631), (6120.0, 986.8683381), (6180.0, 988.0820108), (6240.0,
989.2867873), (6300.0, 990.4828668), (6360.0, 991.6704417), (6420.0,
992.849698), (6480.0, 994.0208155), (6540.0, 995.1839681), (6600.0,
996.339324), (6660.0, 997.4870459), (6720.0, 998.6272916), (6780.0,
```

```
999.7602136), (6840.0, 1000.88596), (6900.0, 1002.004673), (6960.0,
1003.116492), (7020.0, 1004.221553), (7080.0, 1005.319984), (7140.0,
1006.411913), (7200.0, 1007.497462), (7260.0, 1008.576751), (7320.0,
1009.649895), (7380.0, 1010.717007), (7440.0, 1011.778196), (7500.0,
1012.833568), (7560.0, 1013.883225), (7620.0, 1014.927268), (7680.0,
1015.965794), (7740.0, 1016.998898), (7800.0, 1018.026672), (7860.0,
1019.049206), (7920.0, 1020.066586), (7980.0, 1021.078898), (8040.0,
1022.086223), (8100.0, 1023.088643), (8160.0, 1024.086236), (8220.0,
1025.079079), (8280.0, 1026.067244), (8340.0, 1027.050806), (8400.0,
1028.029835), (8460.0, 1029.004399), (8520.0, 1029.974566), (8580.0,
1030.940401), (8640.0, 1031.901968), (8700.0, 1032.859331), (8760.0,
1033.812549), (8820.0, 1034.761682), (8880.0, 1035.706788), (8940.0,
1036.647924), (9000.0, 1037.585145), (9060.0, 1038.518505), (9120.0,
1039.448058), (9180.0, 1040.373854), (9240.0, 1041.295944), (9300.0,
1042.214379), (9360.0, 1043.129205), (9420.0, 1044.04047), (9480.0,
1044.948221), (9540.0, 1045.852502), (9600.0, 1046.753358), (9660.0,
1047.650832), (9720.0, 1048.544966), (9780.0, 1049.435802), (9840.0,
1050.32338), (9900.0, 1051.20774), (9960.0, 1052.088921), (10020.0,
1052.966961), (10080.0, 1053.841898), (10140.0, 1054.713767), (10200.0,
1055.582606), (10260.0, 1056.448448), (10320.0, 1057.311329), (10380.0,
1058.171282), (10440.0, 1059.028341), (10500.0, 1059.882537), (10560.0,
1060.733903), (10620.0, 1061.58247), (10680.0, 1062.428268), (10740.0,
```

```
    1063.271329), (10800.0, 1064.11168)), name='Fire-180', smooth=
    SOLVER_DEFAULT, timeSpan=STEP)

mdb.models['Model-1'].TabularAmplitude(data=((0.0, 20.0), (60.0, 332.5304157),
    (120.0, 426.044443), (180.0, 487.12866), (240.0, 532.516808), (300.0,
    568.4577615), (360.0, 598.0496271), (420.0, 623.0742538), (480.0,
    644.6569038), (540.0, 663.5558175), (600.0, 680.3069656), (660.0,
    695.303428), (720.0, 708.8420627), (780.0, 721.1524826), (840.0,
    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,
    803.9778734), (1440.0, 809.7725613), (1500.0, 815.2792562), (1560.0,
    820.5228938), (1620.0, 825.5255316), (1680.0, 830.3067617), (1740.0,
    834.8840538), (1800.0, 839.2730404), (1860.0, 843.4877575), (1920.0,
    847.5408468), (1980.0, 851.4437275), (2040.0, 855.2067436), (2100.0,
    858.8392889), (2160.0, 862.349916), (2220.0, 865.746429), (2280.0,
    869.0359653), (2340.0, 872.2250658), (2400.0, 875.3197372), (2460.0,
    878.3255053), (2520.0, 881.2474635), (2580.0, 884.0903142), (2640.0,
    886.8584059), (2700.0, 889.5557668), (2760.0, 892.1861333), (2820.0,
    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,
```

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

```
1012.833568), (7560.0, 1013.883225), (7620.0, 1014.927268), (7680.0,
1015.965794), (7740.0, 1016.998898), (7800.0, 1018.026672), (7860.0,
1019.049206), (7920.0, 1020.066586), (7980.0, 1021.078898), (8040.0,
1022.086223), (8100.0, 1023.088643), (8160.0, 1024.086236), (8220.0,
1025.079079), (8280.0, 1026.067244), (8340.0, 1027.050806), (8400.0,
1028.029835), (8460.0, 1029.004399), (8520.0, 1029.974566), (8580.0,
1030.940401), (8640.0, 1031.901968), (8700.0, 1032.859331), (8760.0,
1033.812549), (8820.0, 1034.761682), (8880.0, 1035.706788), (8940.0,
1036.647924), (9000.0, 1037.585145), (9060.0, 1038.518505), (9120.0,
1039.448058), (9180.0, 1040.373854), (9240.0, 1041.295944), (9300.0,
1042.214379), (9360.0, 1043.129205), (9420.0, 1044.04047), (9480.0,
1044.948221), (9540.0, 1045.852502), (9600.0, 1046.753358), (9660.0,
1047.650832), (9720.0, 1048.544966), (9780.0, 1049.435802), (9840.0,
1050.32338), (9900.0, 1051.20774), (9960.0, 1052.088921), (10020.0,
1052.966961), (10080.0, 1053.841898), (10140.0, 1054.713767), (10200.0,
1055.582606), (10260.0, 1056.448448), (10320.0, 1057.311329), (10380.0,
1058.171282), (10440.0, 1059.028341), (10500.0, 1059.882537), (10560.0,
1060.733903), (10620.0, 1061.58247), (10680.0, 1062.428268), (10740.0,
1063.271329), (10800.0, 1064.11168), (10860.0, 1064.949352), (10920.0,
1065.784372), (10980.0, 1066.616769), (11040.0, 1067.44657), (11100.0,
1068.273802), (11160.0, 1069.098491), (11220.0, 1069.920664), (11280.0,
1070.740345), (11340.0, 1071.55756), (11400.0, 1072.372334), (11460.0,
```

```
    1073.184691), (11520.0, 1073.994654), (11580.0, 1074.802247), (11640.0,
    1075.607493), (11700.0, 1076.410414), (11760.0, 1077.211033), (11820.0,
    1078.009372), (11880.0, 1078.805452), (11940.0, 1079.599294), (12000.0,
    1080.390919), (12060.0, 1081.180347), (12120.0, 1081.967599), (12180.0,
    1082.752694), (12240.0, 1083.535652), (12300.0, 1084.316493), (12360.0,
    1085.095234), (12420.0, 1085.871895), (12480.0, 1086.646493), (12540.0,
    1087.419047), (12600.0, 1088.189575), (12660.0, 1088.958094), (12720.0,
    1089.724622), (12780.0, 1090.489174), (12840.0, 1091.251768), (12900.0,
    1092.012421), (12960.0, 1092.771148), (13020.0, 1093.527966), (13080.0,
    1094.28289), (13140.0, 1095.035936), (13200.0, 1095.787118), (13260.0,
    1096.536453), (13320.0, 1097.283955), (13380.0, 1098.029638), (13440.0,
    1098.773518), (13500.0, 1099.515608), (13560.0, 1100.255922), (13620.0,
    1100.994474), (13680.0, 1101.731278), (13740.0, 1102.466347), (13800.0,
    1103.199695), (13860.0, 1103.931334), (13920.0, 1104.661278), (13980.0,
    1105.38954), (14040.0, 1106.116131), (14100.0, 1106.841064), (14160.0,
    1107.564353), (14220.0, 1108.286007), (14280.0, 1109.006041), (14340.0,
    1109.724465), (14400.0, 1110.441291)), name='Fire-240', smooth=
    SOLVER_DEFAULT, timeSpan=STEP)

#Flim_Condition

mdb.models['Model-1'].FilmConditionProp(dependencies=0, name='F', property=((0.025, ), ), temperatureDependency=OFF)
mdb.models['Model-1'].FilmConditionProp(dependencies=0, name='NF', property=((0.008, ), ), temperatureDependency=OFF)

#Surface_Create

```

```
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), ), ))
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), ), ))
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*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), ), ))
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), ), ))
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*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
), ), ))
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*H), ), ((-0.5*C-C1, 0.5*D, 0.5*H), ), ))
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-
```

```
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), ), ))
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*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), ), ))

#Interaction_Tie
#Change_If_Need
mdb.models['Model-1'].FilmCondition(createStepName='Fire',
    definition=
     PROPERTY_REF, interactionProperty='F', name='F',
        sinkAmplitude='Fire-240',
     sinkDistributionType=UNIFORM, sinkFieldName='',
        sinkTemperature=1.0,
     surface=mdb.models['Model-1'].rootAssembly.surfaces['F-4
        '])
mdb.models['Model-1'].FilmCondition(createStepName='Fire',
    definition=
```

```
    PROPERTY_REF, interactionProperty='NF', name='NF', sinkAmplitude='Fire-240'
    , sinkDistributionType=UNIFORM, sinkFieldName='', sinkTemperature=1.0,
    surface=mdb.models['Model-1'].rootAssembly.surfaces['NF-4'])
mdb.models['Model-1'].RadiationToAmbient(ambientTemperature=1.0,
    ambientTemperatureAmp='Fire-240', createStepName='Fire', distributionType=
    UNIFORM, emissivity=0.8, field='', name='FR', radiationType=AMBIENT,
    surface=mdb.models['Model-1'].rootAssembly.surfaces['F-4'])

mdb.models['Model-1'].rootAssembly.Set(edges=
    mdb.models['Model-1'].rootAssembly.instances['Beam-1'].edges.findAt(((
    C1, -C2, 0.75*H), )), name='B1')
mdb.models['Model-1'].rootAssembly.Set(edges=
    mdb.models['Model-1'].rootAssembly.instances['Beam-1'].edges.findAt(((
    C1, C2, 0.75*H), )), name='B3')
mdb.models['Model-1'].rootAssembly.Set(edges=
    mdb.models['Model-1'].rootAssembly.instances['Beam-1'].edges.findAt(((
    -C1, C2, 0.75*H), )), name='B5')
mdb.models['Model-1'].rootAssembly.Set(edges=
    mdb.models['Model-1'].rootAssembly.instances['Beam-1'].edges.findAt(((
    -C1, -C2, 0.75*H), )), name='B7')
mdb.models['Model-1'].rootAssembly.Set(edges=
    mdb.models['Model-1'].rootAssembly.instances['Bar-1-lin-2-1'].edges.findAt(
    ((C1, -C2, 0.25*H), )), name='B2')
mdb.models['Model-1'].rootAssembly.Set(edges=
    mdb.models['Model-1'].rootAssembly.instances['Bar-1-lin-1-2-lin-2-1'].edges.findAt(
    ((C1, C2, 0.25*H), )), name='B4')
mdb.models['Model-1'].rootAssembly.Set(edges=
    mdb.models['Model-1'].rootAssembly.instances['Bar-1-lin-1-2'].edges.findAt(
    ((-C1, C2, 0.25*H), )), name='B6')
mdb.models['Model-1'].rootAssembly.Set(edges=
```

```
    mdb.models['Model-1'].rootAssembly.instances['Bar-1'].edges.findAt(((
    -C1, -C2, 0.25*H), )), name='B8')

mdb.models['Model-1'].Tie(adjust=ON, master=
    mdb.models['Model-1'].rootAssembly.sets['B1'], name='T1',
    positionToleranceMethod=COMPUTED, slave=
    mdb.models['Model-1'].rootAssembly.sets['B2'], thickness=ON, tieRotations=
    ON)

mdb.models['Model-1'].Tie(adjust=ON, master=
    mdb.models['Model-1'].rootAssembly.sets['B3'], name='T2',
    positionToleranceMethod=COMPUTED, slave=
    mdb.models['Model-1'].rootAssembly.sets['B4'], thickness=ON, tieRotations=
    ON)

mdb.models['Model-1'].Tie(adjust=ON, master=
    mdb.models['Model-1'].rootAssembly.sets['B5'], name='T3',
    positionToleranceMethod=COMPUTED, slave=
    mdb.models['Model-1'].rootAssembly.sets['B6'], thickness=ON, tieRotations=
    ON)

mdb.models['Model-1'].Tie(adjust=ON, master=
    mdb.models['Model-1'].rootAssembly.sets['B7'], name='T4',
    positionToleranceMethod=COMPUTED, slave=
    mdb.models['Model-1'].rootAssembly.sets['B8'], thickness=ON, tieRotations=
    ON)


#Mesh

mdb.models['Model-1'].parts['Bar'].seedPart(deviationFactor=0.1, minSizeFactor=0.1, size=d)
mdb.models['Model-1'].parts['Bar'].generateMesh()
mdb.models['Model-1'].parts['Bar'].setElementType(elemTypes=(ElemType(elemCode=DC1D2, elemLibrary=STANDARD), ), regions=(mdb.models['Model-1'].parts['Bar'].edges.findAt
```

```
(((0.5*H, 0.0, 0.0), )), ))

mdb.models['Model-1'].parts['Beam'].seedPart(deviationFactor=0.1, minSizeFactor=0.1, size=d)
mdb.models['Model-1'].parts['Beam'].generateMesh()
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), ), ), ))
mdb.models['Model-1'].rootAssembly.regenerate()

#Interaction_Node

mdb.models['Model-1'].rootAssembly.DatumPointByCoordinate(coords=(0.0, 0.0, 0.5*H))
mdb.models['Model-1'].rootAssembly.DatumPointByCoordinate(coords=(-C1, C2, 0.5*H))
mdb.models['Model-1'].rootAssembly.DatumPointByCoordinate(coords=(C1, C2, 0.5*H))
mdb.models['Model-1'].rootAssembly.DatumPointByCoordinate(coords=(-C1, -C2, 0.5*H))
mdb.models['Model-1'].rootAssembly.DatumPointByCoordinate(coords=(C1, -C2, 0.5*H))

#nodes_[element:node]

#mdb.models['Model-1'].rootAssembly.Set(name='TC5-C', nodes=mdb.models['Model-1'].rootAssembly.instances['Beam-1'].nodes[1843:1844])
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])
mdb.models['Model-1'].rootAssembly.Set(name='TC1-B2', nodes=mdb.models['Model-1'].rootAssembly.instances['Bar-1-lin-1-2-lin-2-1'].nodes[b:b+1])
```

```
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])
mdb.models['Model-1'].rootAssembly.Set(name='TC1-B4', nodes=mdb.models['Model-1'].rootAssembly.instances['Bar-1'].nodes[b:b+1])

#Multiple_Model

mdb.Model(name='Model-2', objectToCopy=mdb.models['Model-1'])
mdb.models['Model-2'].interactions['F'].setValues(definition=PROPERTY_REF,
    interactionProperty='F', sinkAmplitude='Fire-240', sinkTemperature=1.0,
    surface=mdb.models['Model-2'].rootAssembly.surfaces['F-3'])
mdb.models['Model-2'].interactions['FR'].setValues(ambientTemperature=1.0,
    ambientTemperatureAmp='Fire-240', distributionType=UNIFORM, emissivity=0.8,
    field='', radiationType=AMBIENT, surface=
    mdb.models['Model-2'].rootAssembly.surfaces['F-3'])
mdb.models['Model-2'].interactions['NF'].setValues(definition=PROPERTY_REF,
    interactionProperty='NF', sinkAmplitude='Fire-240', sinkTemperature=1.0,
    surface=mdb.models['Model-2'].rootAssembly.surfaces['NF-3'])

mdb.Model(name='Model-3', objectToCopy=mdb.models['Model-2'])
mdb.models['Model-3'].interactions['F'].setValues(definition=PROPERTY_REF,
    interactionProperty='F', sinkAmplitude='Fire-240', sinkTemperature=1.0,
    surface=mdb.models['Model-3'].rootAssembly.surfaces['F-2'])
mdb.models['Model-3'].interactions['FR'].setValues(ambientTemperature=1.0,
    ambientTemperatureAmp='Fire-240', distributionType=UNIFORM, emissivity=0.8,
    field='', radiationType=AMBIENT, surface=
    mdb.models['Model-3'].rootAssembly.surfaces['F-2'])
```

```
mdb.models['Model-3'].interactions['NF'].setValues(definition=PROPERTY_REF,
    interactionProperty='NF', sinkAmplitude='Fire-240', sinkTemperature=1.0,
    surface=mdb.models['Model-3'].rootAssembly.surfaces['NF-2'])

mdb.Model(name='Model-4', objectToCopy=mdb.models['Model-3'])
mdb.models['Model-4'].interactions['F'].setValues(definition=PROPERTY_REF,
    interactionProperty='F', sinkAmplitude='Fire-240', sinkTemperature=1.0,
    surface=mdb.models['Model-3'].rootAssembly.surfaces['F-1'])
mdb.models['Model-4'].interactions['FR'].setValues(ambientTemperature=1.0,
    ambientTemperatureAmp='Fire-240', distributionType=UNIFORM, emissivity=0.8,
    field='', radiationType=AMBIENT, surface=
    mdb.models['Model-3'].rootAssembly.surfaces['F-1'])
mdb.models['Model-4'].interactions['NF'].setValues(definition=PROPERTY_REF,
    interactionProperty='NF', sinkAmplitude='Fire-240', sinkTemperature=1.0,
    surface=mdb.models['Model-3'].rootAssembly.surfaces['NF-1'])

#Job
mdb.Job(atTime=None, contactPrint=OFF, description='', echoPrint=OFF,
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model='Model-1', modelPrint=OFF,
    multiprocessingMode=DEFAULT, name='F-240-4s', nodalOutputPrecision=SINGLE,
    numCpus=8, numDomains=8, numGPUs=0, queue=None, resultsFormat=ODB, scratch=
    '', type=ANALYSIS, userSubroutine='', waitHours=0, waitMinutes=0)

mdb.Job(atTime=None, contactPrint=OFF, description='', echoPrint=OFF,
```

```
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model='Model-2', modelPrint=OFF,
    multiprocessingMode=DEFAULT, name='F-240-3s', nodalOutputPrecision=SINGLE,
    numCpus=8, numDomains=8, numGPUs=0, queue=None, resultsFormat=ODB, scratch=
    '', type=ANALYSIS, userSubroutine='', waitHours=0, waitMinutes=0)

mdb.Job(atTime=None, contactPrint=OFF, description='', echoPrint=OFF,
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model='Model-3', modelPrint=OFF,
    multiprocessingMode=DEFAULT, name='F-240-2s', nodalOutputPrecision=SINGLE,
    numCpus=8, numDomains=8, numGPUs=0, queue=None, resultsFormat=ODB, scratch=
    '', type=ANALYSIS, userSubroutine='', waitHours=0, waitMinutes=0)

mdb.Job(atTime=None, contactPrint=OFF, description='', echoPrint=OFF,
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model='Model-4', modelPrint=OFF,
    multiprocessingMode=DEFAULT, name='F-240-1s', nodalOutputPrecision=SINGLE,
    numCpus=8, numDomains=8, numGPUs=0, queue=None, resultsFormat=ODB, scratch=
    '', type=ANALYSIS, userSubroutine='', waitHours=0, waitMinutes=0)

#mdb.jobs['F-240'].submit(consistencyChecking=OFF)

#Done
```