CEE 282 – Nonlinear Structural Analysis
Prof. Gregory Deierlein – Winter 2020

Structural Response of Timber under Fire Loading

Authors: Andrés Richa, Davyd Tamrazov

March 29, 2020

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

Structural fire engineering in the United States has not historically been a prominent source of research nor was it something that is extensively designed for in a typical building with the exception of buildings of high priority. However, this changed after the 9/11 attacks and the resulting collapse of the World Trade Center towers in 2001. After a comprehensive investigation, it was determined that the ultimate source of the collapse was the heat-induced sagging of floor trusses which caused buckling of exterior columns and initiated the progressive collapse of the entire structure [1]. This has instigated more research in structural response to thermal (fire) loads and established structural fire engineering as a distinct discipline.

On the other hand, timber construction is an even more recent area of structural engineering which has been getting increased interest in recent years both due to aesthetics and carbon-sequestration capacity of timber products. Generally, timber structures are divided into 'heavy timber' and 'light wood frame' categories. Principal structural members are always heavy timber structures made from cross-laminated timber (CLT), glue-laminated timber (glulam), Laminated Veneer Lumber (LVL) or traditional solid timber [2].

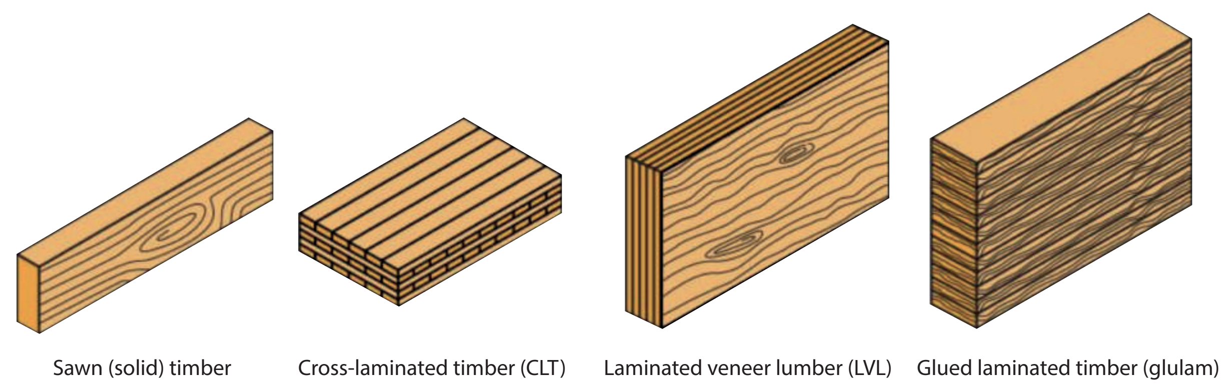

Figure 1: *Heavy timber products [9]*

When it comes to fire performance, timber is a flammable material, and so, unlike steel and concrete, it has the capacity to fuel a fire. However, timber has a low thermal conductivity and under fire tends to burn at a slow rate. At the same time, its charring rate is well documented and conforms with the developed models, thereby making it very predictable [2]. As a result, only a thin layer of timber is affected by the fire, while the formed char insulates the rest of the section and minimizes timber capacity to increase fuel load.

The current standard practice for structural fire engineering for non essential facilities is to treat fire as an afterthought. In the United States, building is usually assigned a generic fire resistance rating and a certain amount of fire proofing is applied to structural elements. Specifically for heavy timber design, a sacrificial layer is added to the section to account for a charred layer of wood in case of the fire. In contrast, Eurocodes provide a more rigorous procedure to analyzing timber under fire loading and provide these calculations methods in the 'informative' appendix [3].

1.1.1 Scope of the Study

The purpose of this project to investigate the stability of timber columns under fire loading using a rigorous fiber-based approach taking advantage of both the geometric and material non-linear effects. Eurocode 5 methods for calculating charring rate and temperature distribution across the heat affected layer will be used to determine section profile at each fire exposure time-step. Python interpreter of OpenSees, OpenSeesPy, will be utilized to set up the model and perform a 2nd-order inelastic analysis of the column with the fiber-discretized section based on the temperature distribution at each step. Results will be post-processed to determine the reduction of the critical load with fire exposure time given a specific model geometry. Given the required fire resistance rating, the developed routine will be able to produce critical load reduction factor that can aid the design process of determining fire proofing strategies.

2 Material Definition

2.1 Mechanical Properties

Timber is an anisotropic material meaning that its properties vary in different directions. As shown in Figure 2, due to the concentric growth rings of a tree, timber mechanical properties are normally characterized in three directions: longitudinal, tangential, and radial. The longitudinal and tangential direction usually exhibit similar properties and are normally grouped into parallel to the grain, while the radial direction is considered perpendicular to the grain. Parallel and perpendicular to the grain properties tend to vary significantly depending on the type of loading that the timber element experiences. As such, tension and shear are normally weaker parallel to the grain due to splitting.

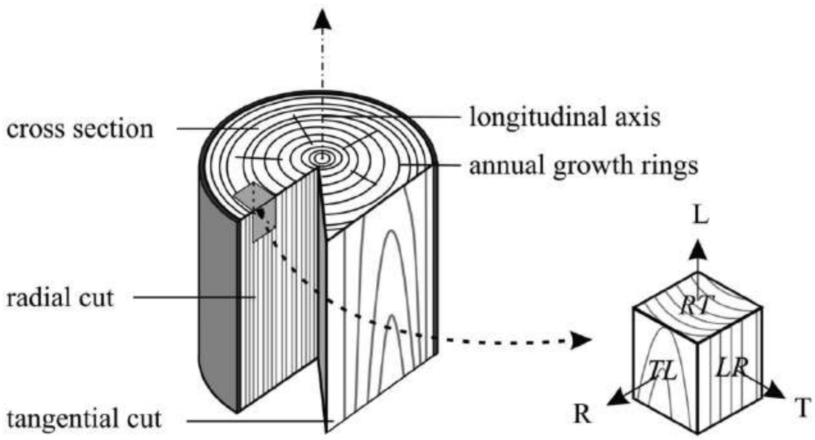

Figure 2: *Wood grain directions [5]*

The variation in mechanical properties of wood can also be attributed to specific wood characteristics such as density and moisture content. Species dependent structural behavior contribute to further variability in mechanical properties, with certain species undergoing strain softening or strain hardening depending on the direction of loading and whether a specimen is being loaded in compression or tension.

2.2 Constitutive Model

For the purpose of limiting the scope of this study, timber was modeled as uniaxial material. As shown in Figure 3, a bilinear elastic-plastic constitutive model was used to approximate stress-strain behavior of the material [11]. A slightly modified version of the bilinear stress-strain curve was used to approximate strain softening in compression, as suggested by Bazan relation [8]. Ultimate strains in both tension and compression (i.e. rupture points) can be represented by a vertical drop in stress to 0 at those strain values. Some adjustments have to me made to define this model in OpenSees, which will be discussed in more detail in the Methodology section.

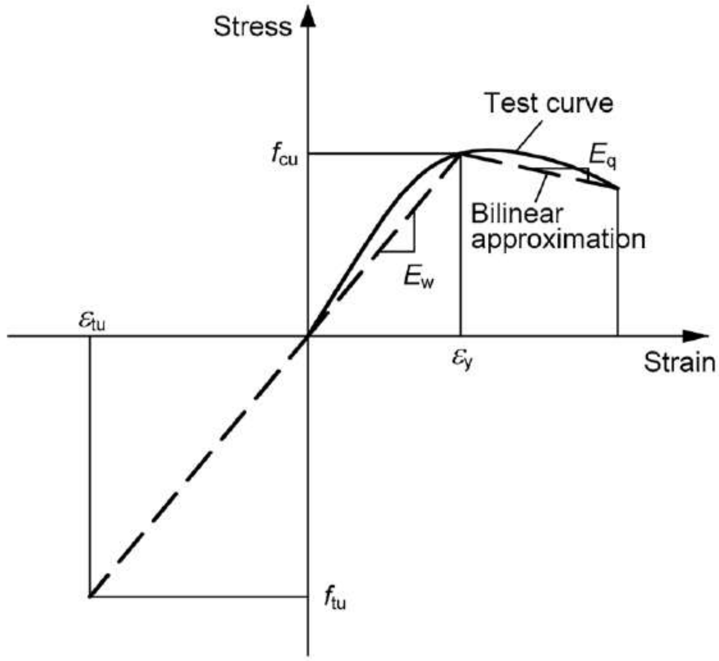

Figure 3: *Bilinear strain softening model [11]*

As mentioned previously, actual strength and elastic modulus values vary considerably between different directions and wood species. Ultimately, the purpose of this study was *not* to assess the behavior of a specific wood type under fire loading, but rather identify trends that any type of wood would follow when its mechanical properties degrade with the increased temperature. Hence, in order to represent realistic timber properties, test results from the study performed by Song were chosen [8]. Key parallel to grain parameters obtained from these test results are summarized in Table 1 below.

Parameter	*Value*
Ultimate compressive stress (f_{cu})	45 MPa
Yield compression strain (ϵ_y)	$4e{-}3$ mm/mm
Modulus of elasticity (E_w)	$11,250$ MPa
Compressive stress at rupture ($f_{cr} = 0.75 \cdot f_{cu}$)	33.75 MPa
Compressive strain at rupture (ϵ_{cu})	$13e{-}3$ mm/mm
Modulus of elasticity, softening (E_q)	1250 MPa
Ultimate tensile stress (f_{tu})	80 MPa
Tensile strain at rupture (ϵ_{tu})	$7.1e{-}3$ mm/mm

Table 1: *Key values used to define bilinear stress-strain curve*

2.3 Response to Fire

2.3.1 Char Layer

When timber is exposed to fire and the surface temperature reaches 300°C, its outer layer will ignite and begin to combust — a process called pyrolysis [2]. During pyrolysis, the burned layer of wood turns into the char, which forms an insulating boundary between the fire and the normal wood, as illustrated in Figure 4. Char continues to accumulate at a decreasing rate as the char layer grows thicker with longer fire exposure times.

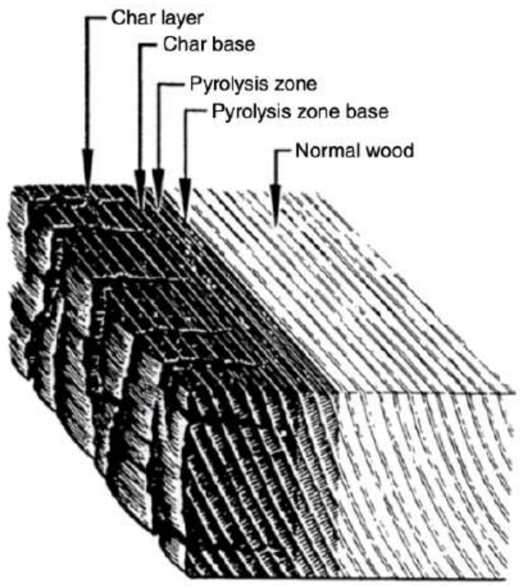

Figure 4: *Structure of the char layer*

Several models were developed through empirical fire testing to estimate the thickness of the char layer at a certain exposure time. In the United States, American Wood Council recommends calculating the thickness of the char layer based on the non-linear model for the charring rate proposed by White, outlined in the equation below [10].

$$c = 2.58\beta_n t^{0.813} \tag{1}$$

where:

- β_n = nominal char rate (0.635 mm/min)
- t = exposure time (min)

The temperature of the char layer is close to that of the fire (300°C). Hence, the char layer, depth of which is calculated from Equation 1, is assumed to have no strength and does not contribute to the capacity of the section. Knowing the depth of the char layer at a certain exposure time allows effectively reducing the cross-sectional area and, as a result, the capacity of a structural element.

2.3.2 Temperature Below Char Layer

Below the char layer, temperature gradient decays following a quadratic function. This temperature profile across the heat affected layer can be defined by the relation proposed by Janssens & White and currently used in Eurocode 5 [6]. From experimental data in the previously mentioned paper, it was determined that for hardwoods, fire tends to penetrate wood up to 35mm, where temperature profile decays to the ambient value (assumed to be 20°C as per Eurocode 5). From the non-linear regression analysis, the best fit equation was found to be:

$$T = T_i + (T_p - T_i) \cdot \left(1 - \frac{x}{a}\right)^2 \tag{2}$$

- x = distance below char layer (mm)
- a = fire penetration depth (35 mm)
- T_i = initial temperature of wood (20 °C)
- T_p = temperature at which charring is initiated (300 °C)

2.3.3 Thermal Expansion

Normally, in reinforced concrete and steel structures, thermal expansion has a significant detrimental effect, generating large internal forces and causing cracking of the reinforced concrete matrix. However, as displayed in Table 2, the coefficient of thermal expansion of timber is around 3 times smaller than that of other structural materials. At the same time, due to low heat conductivity, elevated temperature profile penetrates wood only up to 35 mm, thereby making heat affected area of timber section very small. Therefore, for the purpose of analysis of the structural behavior of timber under fire loading, expansion effects are often negligible and can be ignored.

Material	α (°C^{-1})
Steel (A-36)	$11.7e{-}6$
Concrete	$10.0e{-}6$
Wood, Eastern Spruce	$3.0e{-}6$

Table 2: *Thermal coefficients of linear expansion [7]*

2.3.4 Thermal Degradation

Similarly to other structural materials, the strength of timber is temperature-dependent. As discussed in the previous section, fire loading will create a thin layer on the surface of the timber section where mechanical properties will be affected by the heat. At elevated temperatures, the strength and the elastic modulus of timber reduce, thereby making section weaker and more plastic. Eurocode 5 provides temperature-dependent reduction factors for softwood, which allow determining the reduced strength and modulus of elasticity of timber at temperatures ranging from 20°C up to

300°C — the charring temperature of wood. These reduction factors are defined for parallel to the grain properties and vary depending on the type of loading (see Figures 5 and 6).

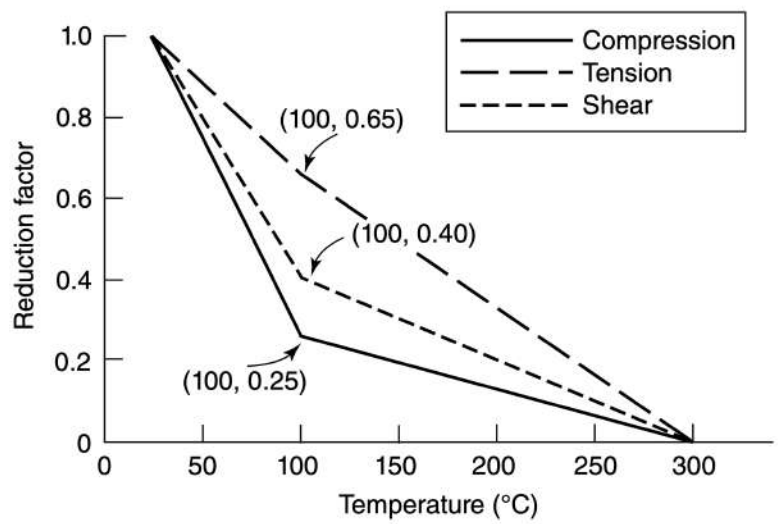

Figure 5: *Compression, tension and shear strength temperature dependent reduction factors [3]*

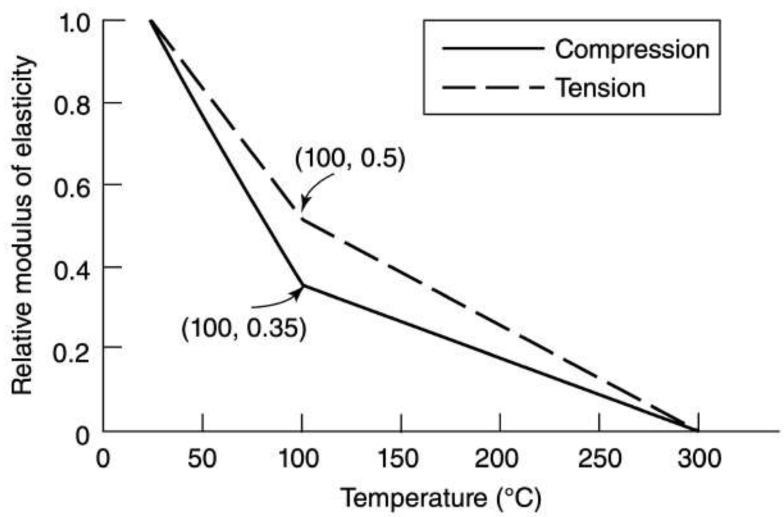

Figure 6: *Compression and tension elastic modulus temperature dependent reduction factors [3]*

Eurocode 5 also provides a simplified procedure for designing heavy timber members under fire loading. This involves analyzing structural member with an effective cross-section reduced by the layer of zero strength, which includes char depth and sacrificial layer of heat-affected wood [4]. Eurocode 5 routine would provide a more conservative estimate of the structural capacity and, hence, one of the objectives of this study is to develop a more rigorous procedure for the analysis of timber members under fire loading.

3 Methodology

A general framework of the Python implementation of the fire loading analysis of a timber column is shown on the flowchart below. Initially, basic model parameters are defined, including the geometry of the arrangement and fire loading temperatures. Section dimensions are then specified along with the mesh size, which affects the accuracy of the fiber analysis of the section — a denser mesh will more closely replicate actual temperature profile across the heat affected layer.

After model parameters are defined, RunAnalysis function is called, in which the main OpenSeesPy analysis routine is performed; this will be further discussed in Section 3.1. As a part of that routine, GetFibers function is called to define patches of fibers based on the temperature profile and the mesh size. The fiber discretization algorithm is described in Section 3.2. This procedure is then repeated for each time-step and analysis outputs are used for post-processing and plotting of the graphs presented later.

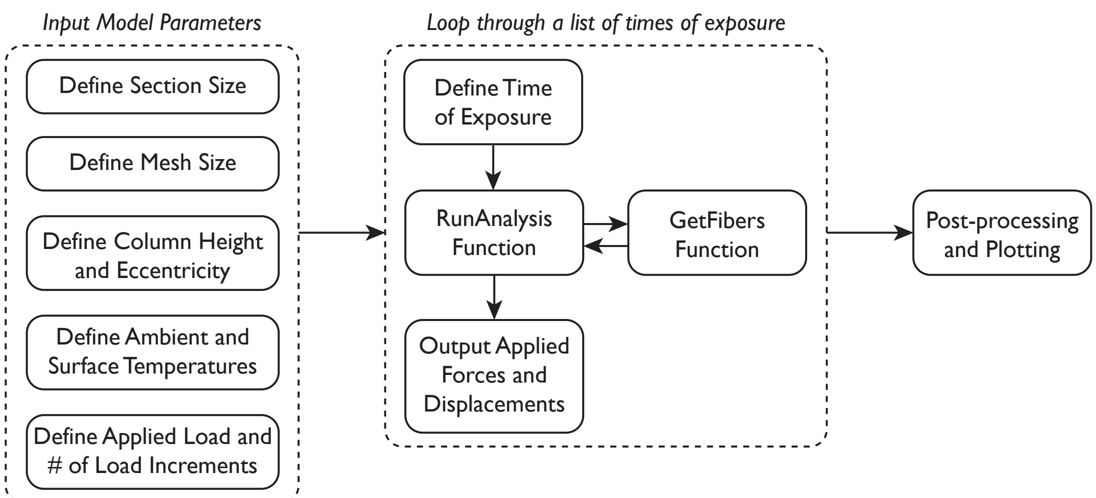

Figure 7: *General routine flowchart*

3.1 OpenSeesPy Routine

Python interpreter of OpenSees (OpenSeesPy) follows a similar routine to the structural analysis process in any other software (see Figure 8). First, all previous results are wiped out and the model is initialized from scratch as a 'basic' 2D structure with three degrees of freedom at each node.

Afterward, fiber section is initialized and discretized into rectangular patches using the algorithm in GetFiber function. Since each patch will have a different temperature and, as a result, a material property associated with it, a separate material has to be defined for each fiber. This allows accounting for material degradation due to temperature profile applied across the section. Using Eurocode 5 curves, reduced strength and elastic modulus values are found for each temperature, which then allows to back-calculate strain values on the stress-strain curve.

Timber material properties are defined using the Hysteretic Material option. Despite no cyclic loading applied to the structure, this built-in function allows defining a uniaxial bilinear material by specifying up to three stress-strain points in both tension and compression directions. Hence, it was found to be the most suitable material option to replicate the constitutive model described in Section 2.2. To model failure of the material at a certain strain, third points at zero stress and a slight increase in strain (1e-8) were added to both tension and compression. This was done to imitate a vertical drop in stress, which, however, cannot be modeled in OpenSees due to a requirement of having a single unique stress value at a given strain.

The geometry of the model and the loading are then set up using function input parameters as described in the previous section. In order to track horizontal displacement of the middle node, recorders are defined for that degree of freedom in the structure. The applied load at the structure is found by summing up the vertical reaction forces at boundary nodes.

To perform a 2nd-order inelastic analysis of the structure, the following parameters are specified:

- Analysis type: Static
- Solution algorithm type: Load Control with increment size predefined from input parameters;
- Numberer: Reverse Cuthill-McKee (RCM), allows to reduce bandwidth of the sparse matrix;
- Force recovery approach: Transformation;
- Convergence test: Norm Unbalance test with the tolerance of $1e{-6}$.

Finally, after the analysis is performed, RunAnalysis function outputs a list of recorded displacements of the middle node and applied load at each load increment, which are then used for post-processing.

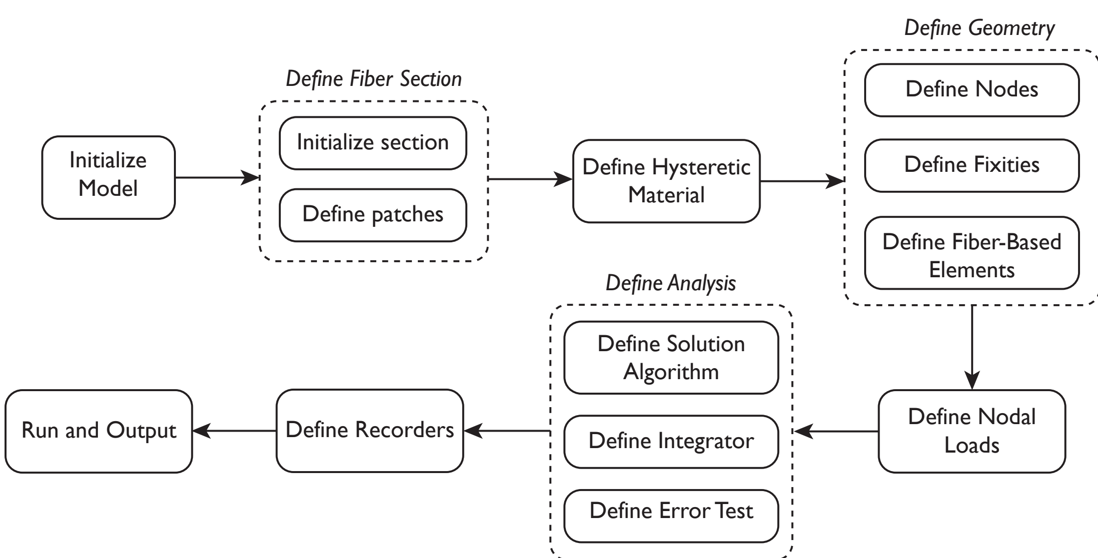

Figure 8: *OpenSeesPy routine flowchart*

3.2 Fiber Discretization

In order to perform fiber analysis of the structure, OpenSeesPy requires to subdivide the initialized section into a number of patches. Rectangular patches are defined by two corner coordinates within the section domain and contain a specific number of fibers in the y and z directions with an assigned predefined material property. To simulate temperature profile across the section, separate patches are defined for each temperature layer of the meshed section, thereby capturing difference in the material properties in the heat affected layers. Hence, the accuracy of the representation of the temperature profile is directly correlated with the mesh size specified.

A Python algorithm was written to obtain a set of patches with all the pertinent parameters mentioned previously that can be directly inputted into the OpenSeesPy model. This routine based on the predefined model parameters, namely, section size, time of exposure, mesh size and ambient temperature. Fiber discretization routine is performed in the following order:

1. Section domain is defined at the ambient temperature based on the section size inputted;
2. Temperature profile at the provided exposure time is calculated using Equation 2 to determine the decay of temperature with depth and Equation 1 to compute the char layer depth;
3. Section domain is meshed with a specified size and the temperature of each mesh element is calculated by taking the average temperature of four points that define that element;
4. Using calculated temperatures at the center of each mesh element, these are grouped into patches with the same temperature;
5. The temperature of each patch is assigned with a tag and used to output a dictionary of unique temperatures and the corresponding tags. These are later used to calculate material properties under those temperatures and define materials when OpenSeesPy analysis routine is performed;
6. Number of fibers in y and z directions and the coordinates of the corners of the patch are determined and put in the format required by OpenSeesPy for fiber section definition.

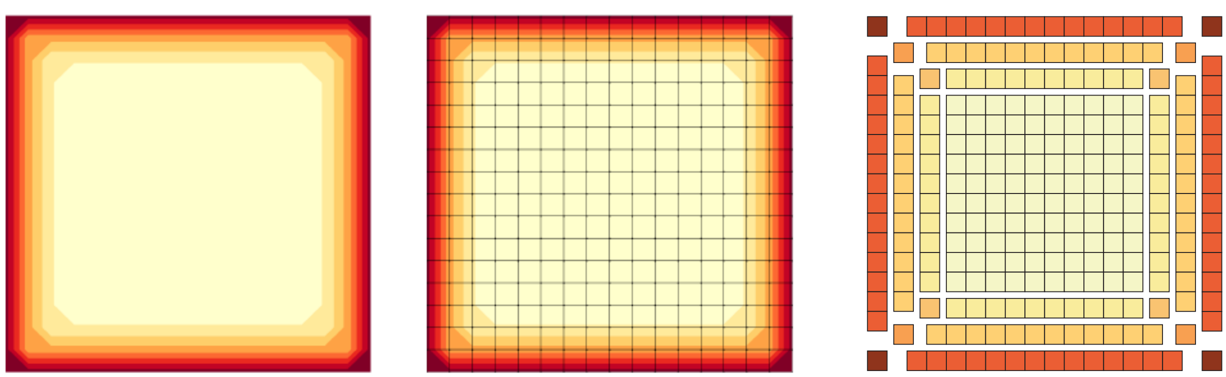

Figure 9: *Fiber discretization routine: Temperature profile across the section domain (left); Meshed section domain (middle); Section domain discretized into patches based on temperatures (right)*

4 Results

This section explores the quality of the results obtained using the developed method for analyzing timber structures under fire loading. As discussed previously, the model is generated in Python using OpenSeesPy interpreter for each time step, which corresponds to a specific section size and temperature profile across the section. This analysis is repeated for a specified number of steps to generate results used later for post-processing.

The problem considered is the reduction of the critical load of a timber column as it is exposed to the constant and uniform fire loading from all sides (i.e. 4-sided exposure). As shown in the figure below, the pin-pin arrangement is analyzed with the initial eccentricity of L/1000. For the purpose of this study, the following properties of the column are selected to replicate realistic sizes that could be used for a single-story column member:

- Section size: 400x400 mm
- Column length: 4000 mm
- Material is assumed to be uni-axial with properties as summarized in Section 2.3

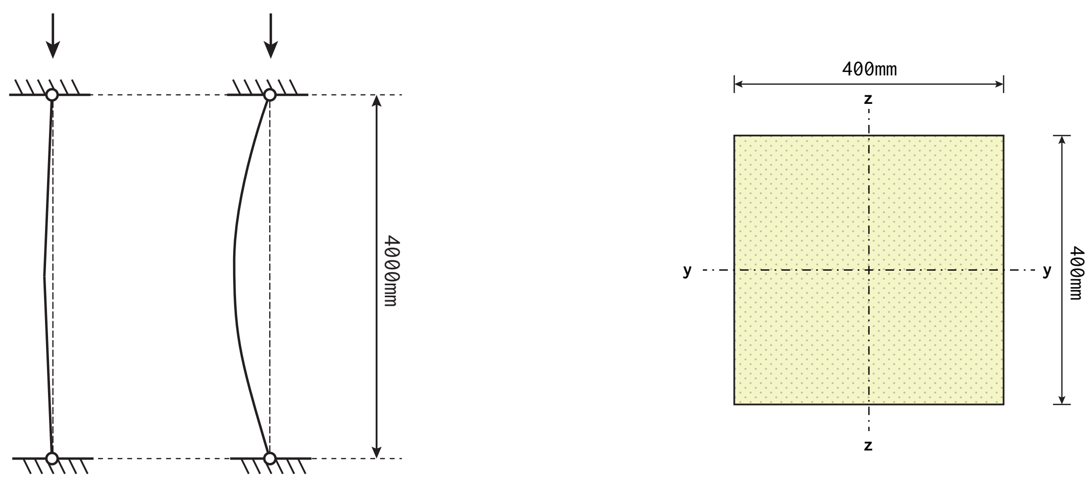

Figure 10: *Column buckling problem definition*

4.1 Force-Displacement Relationship

To obtain applied force versus displacement plot, the arrangement was analyzed in OpenSeesPy with material properties and fibers defined to reflect temperature change and char formation at each time step. For simplification purposes and analysis limitations in OpenSeesPy, the model was redefined before each analysis, meaning that displacement and internal forces were not accumulated with time. A sufficiently large load was applied (15,000 kN in this case) to reach a limit point in the analysis and cause the failure of the column.

The timber section was discretized into a 5mm square mesh to obtain a more accurate fiber representation of temperature variability across the section profile (see Figure 11).

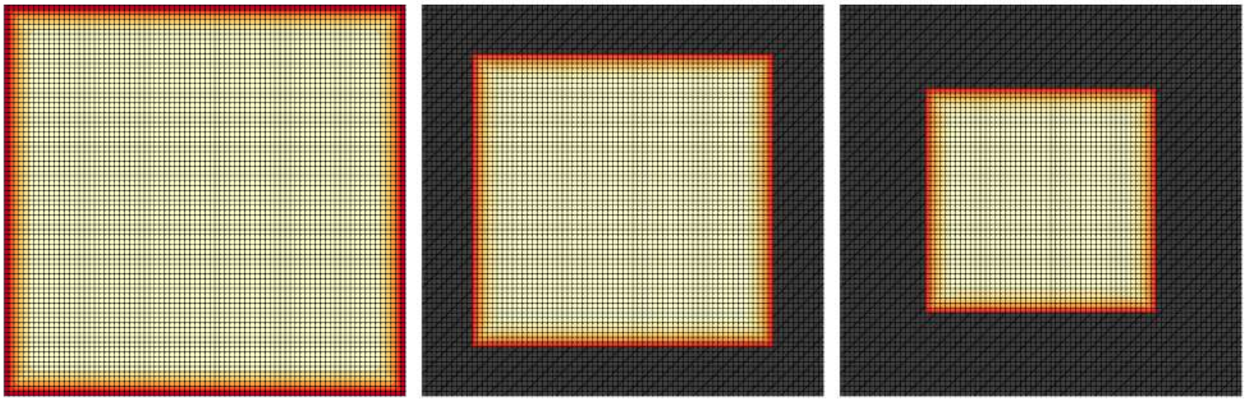

Figure 11: *Section profile with exposure time: 0, 60 and 120 minutes (left to right)*

As the burning of the timber column progresses with time and the char layer grows, a significant reduction in the load at failure can be observed (see Figure 12). Additionally, temperature profile affects larger percentages of the section at longer fire exposure times (i.e. temperature penetration is constant, but effective section size is reducing). As a result, an increase in plasticity is also evident, which conforms with the material properties and reduction factors assumed in our analysis.

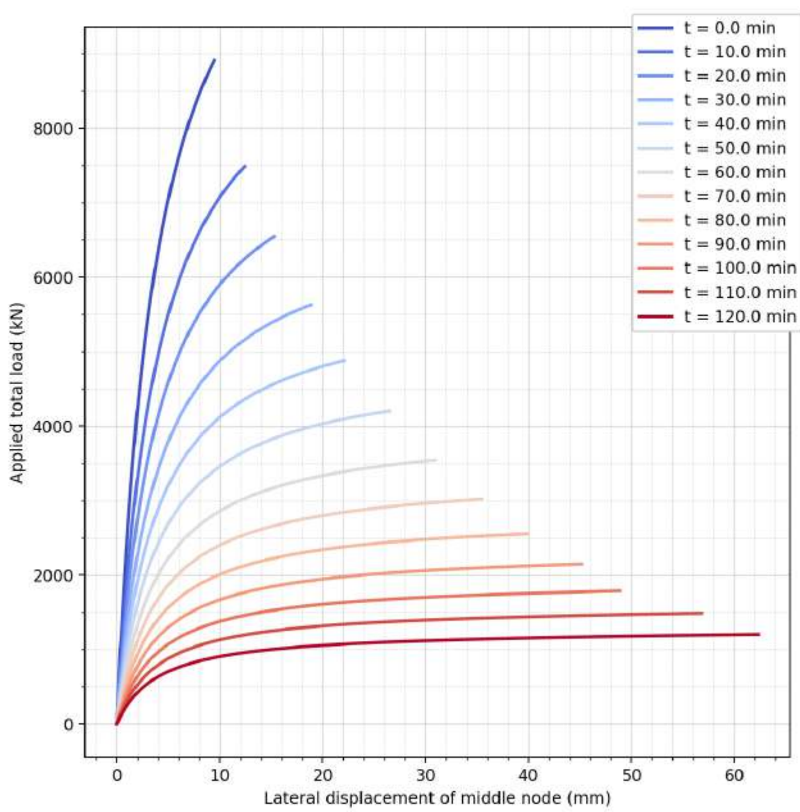

Figure 12: *Applied total load versus displacement plot for a pin-pin column arrangement*

Some inconsistencies are present after 90 mins of fire exposure when the strength of the section becomes very sensitive to the mesh size and load step size used. The trend of decreasing strength and increasing plasticity should continue in a consistent manner if a finer mesh is used as the section gets smaller. Further, to get even more accurate results, the section can be further discretized at the 35mm heat affected layer.

4.2 Critical Load Reduction Factor

Depending on the chosen section size and column length, the critical load will be affected by the combination of two processes:

- Degradation of material properties with temperature increase;
- Reduction of effective cross-sectional area due to char formation.

For the purpose of this analysis, the critical load reduction factor is defined as the ratio of the critical load at time step t, normalized by the original critical load of the column (i.e. the entire section at ambient temperature). This factor allows quantifying the effect of the applied fire load on the strength of the column at a specific exposure time.

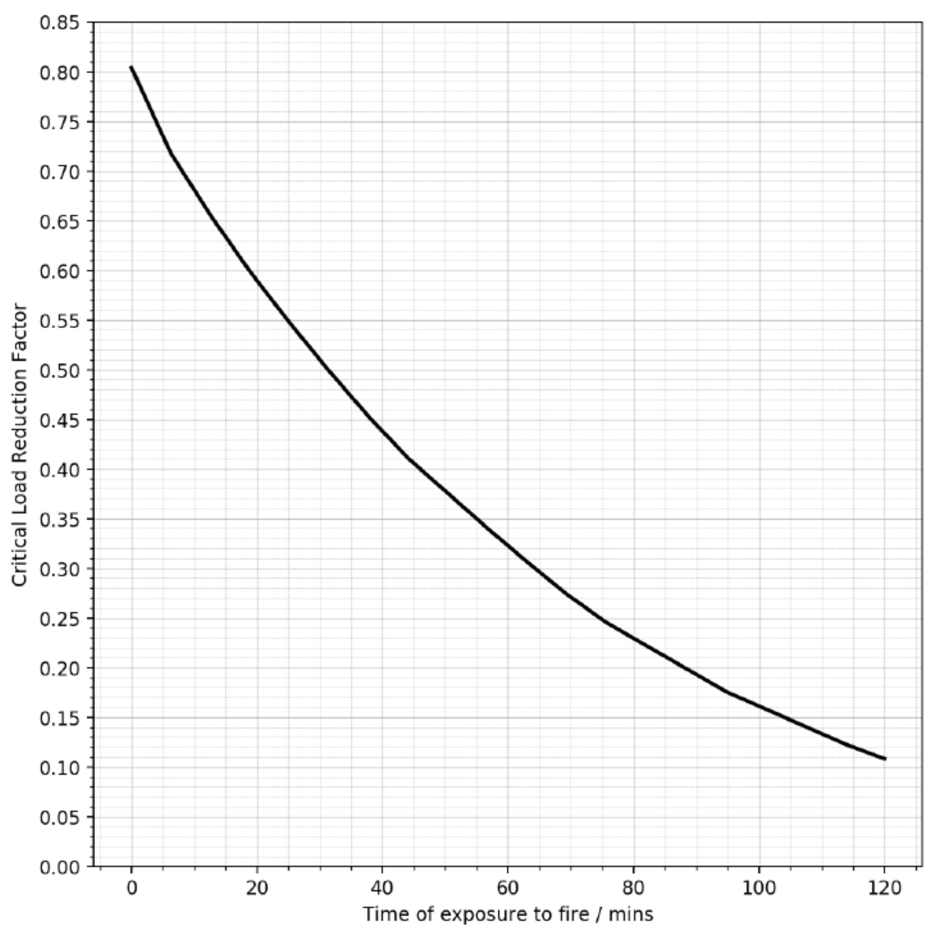

Figure 13: *Critical load reduction factor versus exposure time plot*

From the plot in Figure 13, as the exposure time to fire increases, the drop in section strength appears to follow some form of exponential decay. By the time of 60 mins, critical load drops approximately than 70%, whilst at 2 hours of fire exposure, the section is only at 10% of its original capacity. This plot emphasizes how significant the effect of fire can be on the timber section and allows to predict the strength of the column at any specified time of exposure. Furthermore, it is important to note that at time 0, the critical load reduction factor does not start at 1, but rather section capacity is already reduced. This arises due to the fact that time is assumed to start from when the first layer of char is formed and the initial heating-up phase of the section is not included.

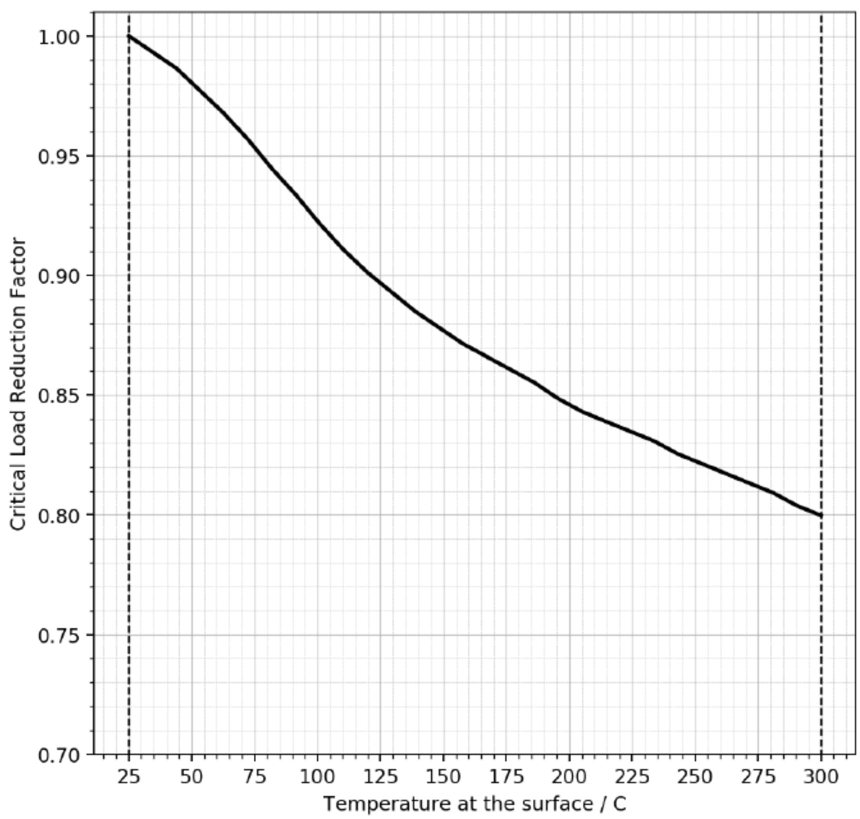

Figure 14: *Critical load reduction factor versus surface temperature plot*

In this particular case, change in the surface temperature of the section has a considerable effect on the overall strength of the section. As can be seen from Figure 14, critical load drops by close to 20% even before the first layer of char is formed (i.e. surface temperature reaches 300°C), underlining the significance of the instantaneous effect of fire loading.

However, in order to be able to quantify the timescale of this effect, timber heat conduction model has to be utilized. For the presented study, a constant temperature profile across the section was modeled using empirical equations, meaning that the timescale of the heating up effect on the section strength could not be obtained.

4.3 Sensitivity analyses

4.3.1 Section Size

The same fire loading analysis was run with 4 different sections to assess the difference in impact fire loading has on the strength of each of the columns. A larger time step was chosen to accelerate the analysis and obtain curves detailed enough to be able to identify the trend.

From Figure 15, it can be observed that the stockier the column section, the lower is the effect of fire loading on the critical strength of the member. Also, as the size of the section increases, exponential decay of the critical load reduction factor becomes less evident and can be interpreted as linear for large enough sections.

As mentioned previously, temperature profile model chosen defines heat affected layer depth (35mm) as constant and independent of the section size. Hence, smaller sections will be more impacted by the fire loading, because temperature increase and the formation of char layer affects a larger percentage of the entire section. Similarly, this explains a more linear trend for larger members and the lower initial drop in strength when the member goes from ambient to charring temperature since decrease in strength of the member is less significant.

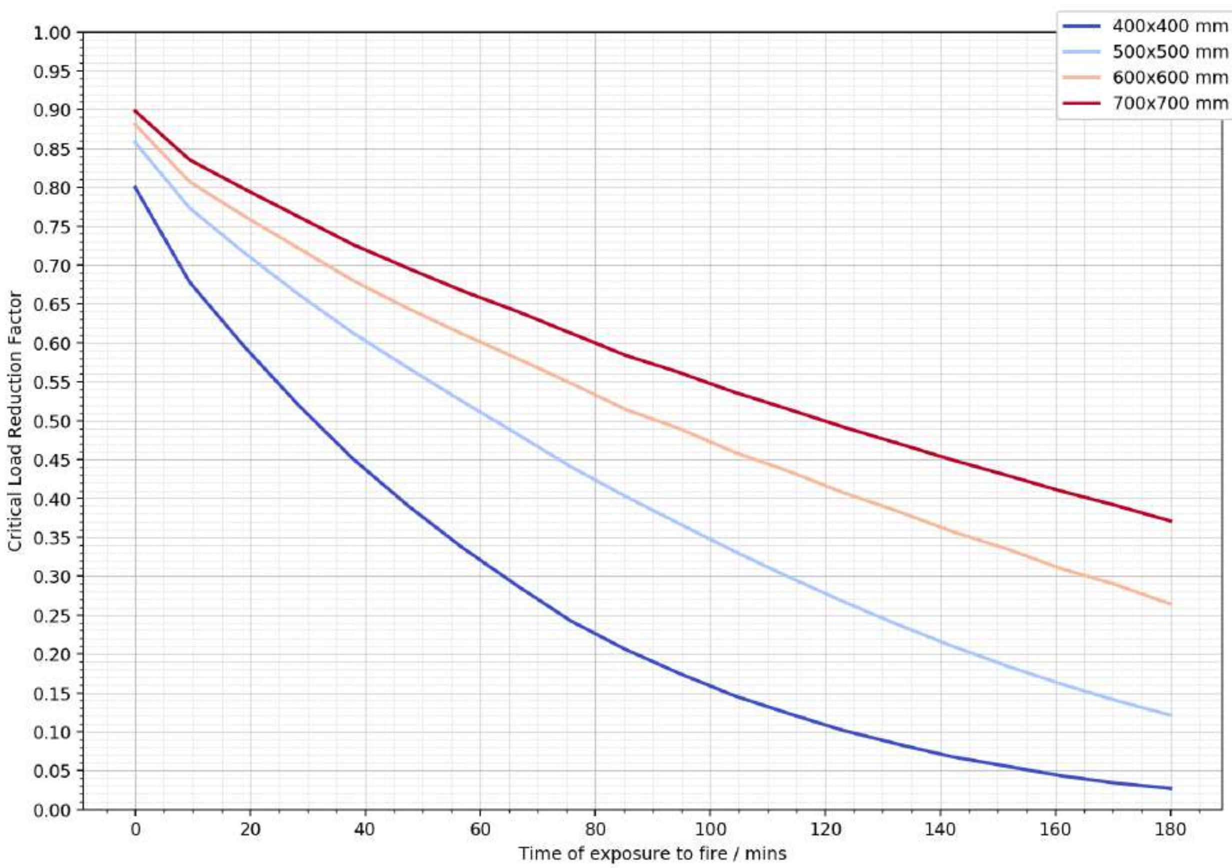

Figure 15: *Critical load reduction factor sensitivity to section size*

4.3.2 Boundary Conditions

Three typical boundary conditions used in the column buckling problem were assessed in the following analysis:

1. Pin-ended column (used for all previous analyses)
2. Column with one fixed end and one pinned end
3. Fully-fixed column

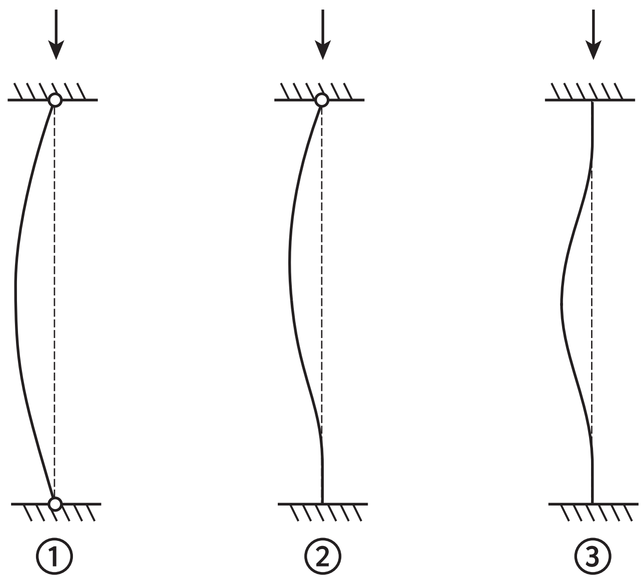

Figure 16: *Three boundary conditions considered*

The same 400x400mm section was analyzed using the developed routine in OpenSeesPy and the critical load reduction factor curves are shown on Figure 17. As the column is less prone to buckling with the increased stiffness of the arrangement, it can be observed that critical load is also marginally less affected by the applied fire load. Interesting to note that columns with boundary conditions 2 and 3 start almost at the same point and slowly diverge with time before all three curves start to converge again. This convergence at the end happens due to the fact that the section has reduced significantly and boundary conditions of the slender members have a smaller effect on the critical buckling load.

From these results, it can be concluded that critical load reduction factor is correlated with the stiffness of the arrangement, similarly so with the results obtained by increasing section size. In this case, fixed boundary conditions redistribute stresses from the middle section of the column to its ends, thereby increasing overall capacity of the column. As a result, fire loading has a smaller effect on those arrangements, what can be seen from the obtained critical load reduction factor curves.

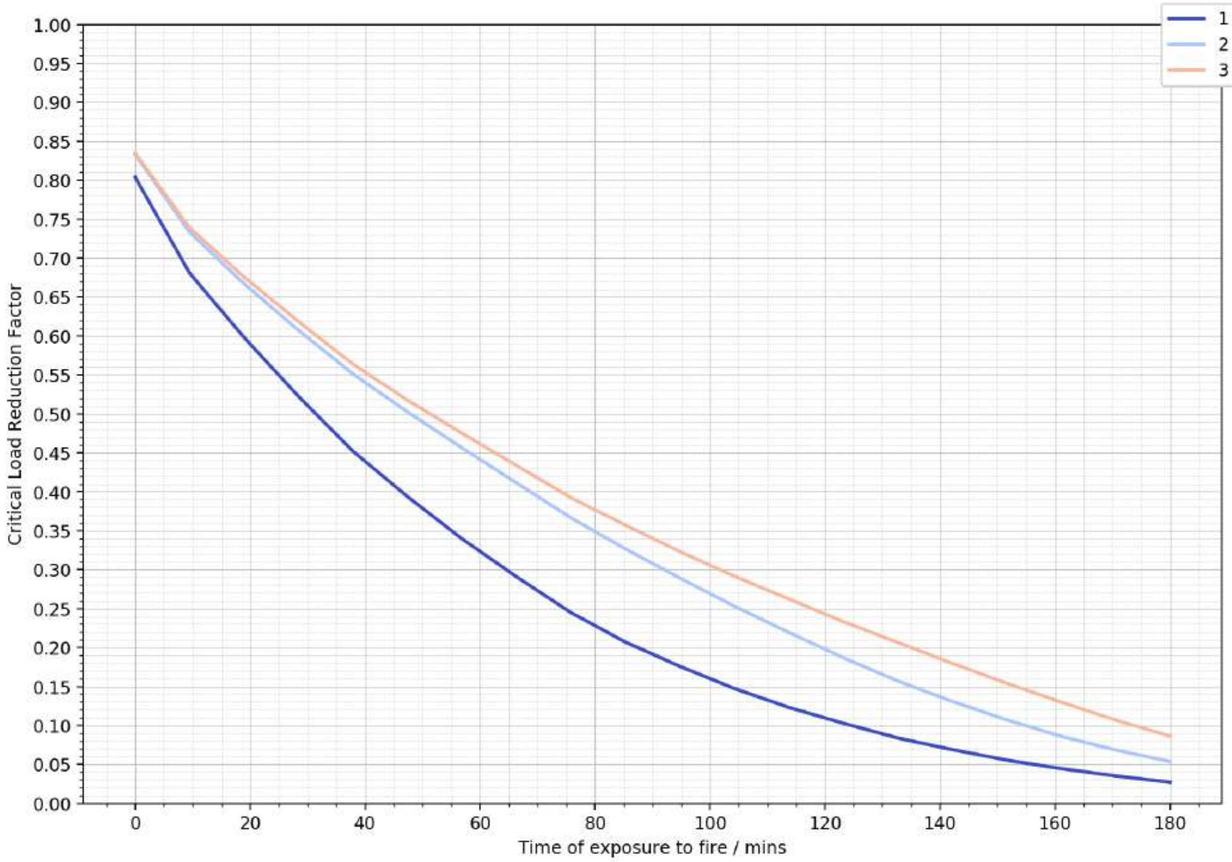

Figure 17: *Critical load reduction factor sensitivity to boundary conditions*

4.3.3 Fire Exposure

The one-sided exposure analysis was performed in order to assess the response of the internal column as compared to that of an external column, where fire is applied from all sides. Temperature profiles that developed in the section from one-sided exposure under different fire exposure times are illustrated on the figure below.

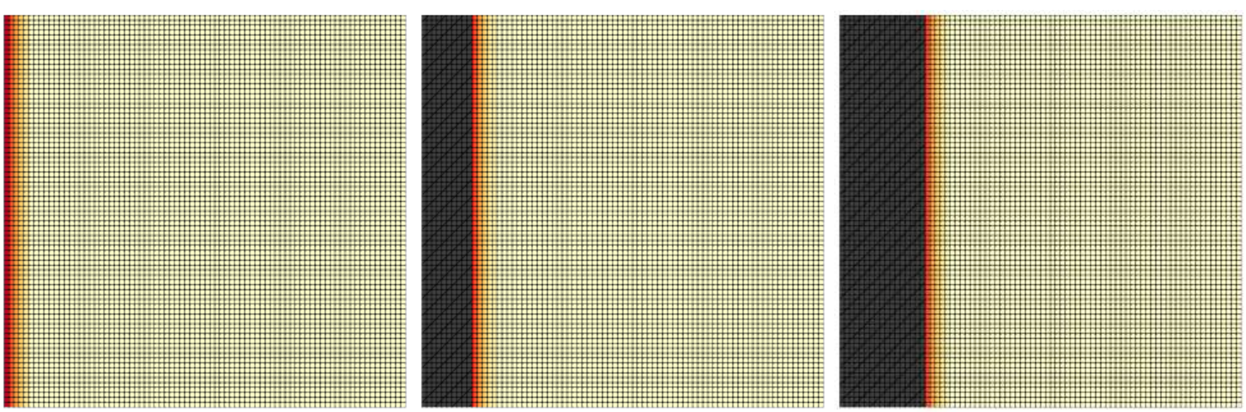

Figure 18: *One-sided fire exposure section profile: 0, 60 and 120 minutes (left to right)*

Results have shown that in the pin-pin case considered, exposure does not play a significant role in the strength reduction of the section. It can be observed from Figure 15 that by being exposed only from one side, timber section is only insignificantly stronger than at the 4-sided exposure. This difference, however, becomes negligible with the increasing exposure times, when two curves start to merge. This result can be explained by the fact that buckling strength of the column largely depends on the material properties of the extreme fibers, which in both cases are affected by the temperature.

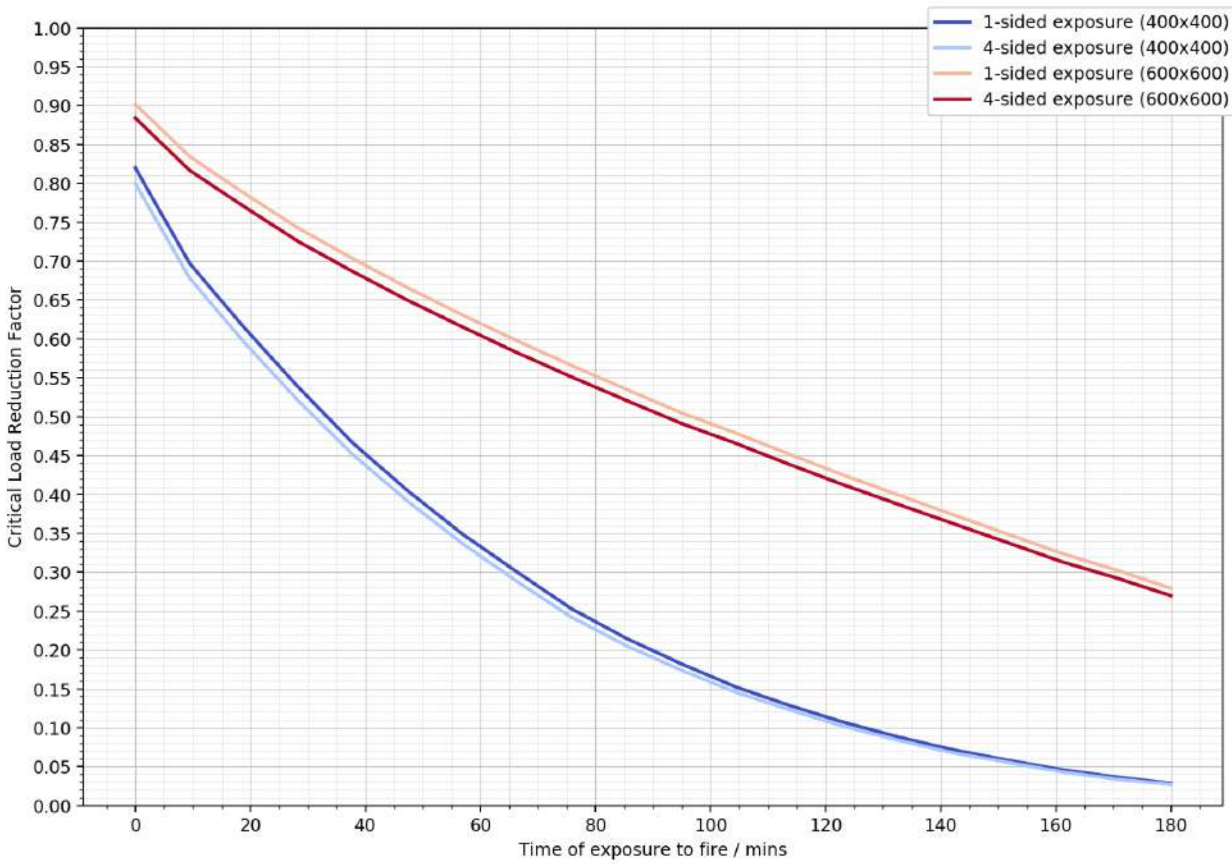

Figure 19: *Critical load reduction factor sensitivity to fire exposure*

As discussed previously, when larger section is analyzed, temperature profile plays a less significant role in the strength reduction of the member. Instead, decrease in the overall cross-sectional area due to charring plays a more significant role. Despite causing similar temperatures at extreme fibers, one-sided exposure has a smaller charring layer and, as a result, one-sided and four-sided exposure curves are further apart for larger section (600x600mm in this case).

Ultimately, the difference between one-sided and four-sided exposures is rather small, suggesting that even with the fire being applied to one side of the interior column, reduction in strength will be comparable to that of the external column under fire.

5 Conclusion

Timber structures under fire loading exhibit a different response to that of other structural materials and, therefore, require a different analysis approach. Eurocode 5 provides a simplified way of analyzing timber structures under fire loading by considering effective area of the charred section at a certain time of exposure to fire. The purpose of this study to develop a more rigorous analysis approach by including heat-affected layer under char and performing inelastic non-linear fiber analysis of the section.

Using Python interpreter of OpenSees, a routine for the 2nd-order inelastic analysis was set up and algorithms were developed to calculate temperature profile across the heat-affected layer as well as to discretize section domain into a series of patches. This allowed to define fiber section for the analysis at each fire exposure time-steps and obtain critical load reduction curves.

The results have shown that the critical load is affected by the combination of two concurrent processes, namely, degradation of material properties with temperature increase and the reduction of cross-sectional area due to char formation. Overall, reduction of critical load with exposure time was found to follow some form of exponential decay. Via sensitivity studies, it was also demonstrated that the magnitude and rate of the decay depends on design parameters, such as section size, boundary conditions and fire exposure. Section size was found to have the largest effect on the critical load reduction factor curve — large members are less affected by the fire loading. This is because depth temperature profile across the section is always constant regardless of its dimensions and, as a result, smaller sections will have larger percentage affected by the heat penetration and char formation.

To conclude, the proposed routine allows to develop critical load reduction curves when section size and applied vertical load is knows. This, in turn, can give an estimate of the failure time of the column under the uniform fire loading. This more rigorous type of analysis can be important when assigning fire rating and determine the required fire proofing for key structural elements.

6 Recommendations for future work

The performed study can be further expanded by considering the following:

- Different element types can be analyzed, such as beams or frames, where shear reduction will have to be accounted for at the heat affected layer.
- To get a more accurate representation of timber behavior, material model will have to be expanded to include anisotropic properties of wood (i.e. distinct parallel and perpendicular to the grain material properties).
- Moisture content has a large effect on timber properties. Hence, interface between fire loading and moisture content of wood can be investigated to determine how moisture content reduces with heat and what is the effect of that on the structural behavior.
- Elements can be analyzed under various fire loading scenarios, including ones that are not evenly distributed across the structure. This will entail defining different fiber section at locations of distinct fire loads.

References

[1] Banovic, S. W., et al. (2007) *The Role of Metallurgy in the NIST Investigation of the World Trade Center Towers Collapse.* Jom, vol. 59(11), pp. 22–30. doi:10.1007/s11837-007-0136-y.

[2] Buchanan, A. H., & Abu, A. (2017) *Structural Design for Fire Safety.* Chichester, West Sussex, United Kingdom: John Wiley & Sons Inc. doi: 10.1002/9781118700402.

[3] CEN (2004) *Eurocode 5: Design of timber structures - Part 1-2: General – Structural fire design. EN 1995-1-2.* European Committee for Standardization, Brussels. doi:10.3403/bsen1995-1.

[4] Fahrni, R., et al. (2019) *Fire Tests on Glued-Laminated Timber Beams with Specific Local Material Properties.* Fire Safety Journal, vol. 107, pp. 161–169. Elsevier, doi: 10.1016/j.firesaf.2017.11.003.

[5] Franke, B., & Pierre, Q. (2011) *Numerical Modeling of the Failure Behavior of Dowel Connections in Wood.* Journal of Engineering Mechanics, vol. 137(3), pp. 186–195. doi: 10.1061/(asce)em.1943-7889.0000217.

[6] Janssens, M. L., & White, R. H. (1994) *Short Communication: Temperature Profiles in Wood Members Exposed to Fire.* Fire and Materials, vol. 18(4), pp. 263–265., doi: 10.1002/fam.810180410.

[7] Logan, D. L. (2017). A First Course in the Finite Element Method (6th ed.). Stamford, CT: Cengage Learning.

[8] Song, J. K., Kim, S. Y., & Oh, S. W. (2007). *The Compressive Stress-strain Relationship of Timber.* Presented at the International Conference on Sustainable Building Asia, Seoul, South Korea. Retrieved from `https://www.irb.fraunhofer.de/CIBlibrary/search-quick-result-list.jsp?A&idSuche=CIB+DC8227`

[9] TRADA (2016) *Timber 2016 Industry Yearbook.* Retrieved March 28, 2020, from `https://issuu.com/trada/docs/trada_2016_low-res/121`

[10] White, R. H. (2004) *Fire resistance of exposed wood members.* Presented at Wood & fire safety 5th international scientific conference, Svolen, Slovakia. Faculty of Wood Sciences and Technology, Technical University of Zvolen, 2004: Pages 337-344.

[11] Zhang, J., et al. (2015) *Research on residual bending capacities of used wood members based on the correlation between non-destructive testing results and the mechanical properties of wood.* J. Zhejiang University Science A, vol 16, pp. 541–550. Retrieved from `https://link.springer.com/article/10.1631/jzus.A1400276`

Appendix A: Fiber Discretization Algorithm

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Rectangle


# Function to compute temperature profile across the section for a given
# exposure time (t) and discretize the section into fibers
#
# ----------------------------------------------------------------------------
# INPUT:
#   t           = time of exposure
#   b           = section width (mm)
#   h           = section height (mm)
#   Ti          = ambient temperature (Celsius)
#   Tp          = surface temperature (Celsius)
#   a           = temperature penetration depth (mm)
#   mSize       = mesh size (mm)
#   plot        = boolean to display section temperature profile plot
#   disp_fiber  = boolean to display mesh on the plot
#   side        = number of sides from which fire is applied (either 1 or 4)
# ----------------------------------------------------------------------------
#
# ----------------------------------------------------------------------------
# OUTPUT:
#   patchList   = list of patches in OpenSeesPy format:
#                 (mattag, ny, nz, coord1, coord2)
#   mattag      = dictionary of unique temperatures in discretized section and
#                 corresponding material tags
# ----------------------------------------------------------------------------
#

def GetFibers(t,b,h,Ti,Tp,a,mSize,plot,disp_fiber,side = 4):

    ## Calculate temperature profile across the section
    # Charring depth at time (t) in mm
    bn = 0.635 #mm/min
    c = 2.58*bn*t**(0.813)

    # Generate mesh within section domain
    x, y = np.meshgrid(np.linspace(0,b,int(b/mSize)+1), np.linspace(0,h,int(h/mSize)+1))

    # Calculate temperature profile in x and y directions (depending on side
    # parameter) and find overall maximum temperature profile section-wide
    if side == 4:
        Tx = np.maximum(Temp(x,a,c,Ti,Tp),Temp(b-x,a,c,Ti,Tp))
        Ty = np.maximum(Temp(y,a,c,Ti,Tp),Temp(h-y,a,c,Ti,Tp))
        z = np.maximum(Tx,Ty)
    elif side == 1:
        z = Temp(x,a,c,Ti,Tp)

    # Map temperature (z) values to meshgrid format
    mask = np.zeros_like(z,dtype=bool)
    z = np.ma.array(z,mask=mask)

    # Generate  contour plot of the temperature profile across the section
    if plot:
        fig, ax = plt.subplots(dpi=175)
        factor = round(max(b,h)/10,0)
        fig.set_size_inches(b/factor,h/factor)
```

```

        ax.set(xlim = [0,b], ylim = [0,h])

        if disp_fiber:
            plt.plot(x,y,marker='.', color='k', linestyle='-', markersize = 3, linewidth = 0.75)
            plt.plot(y,x,color='k', linestyle='-', linewidth = 0.75)
            plt.plot(y,h-x,color='k', linestyle='-', linewidth = 0.75)
        else: plt.contour(x,y,z,colors='gray', linewidths=0.75,linestyles='--')

        plt.contourf(x,y,z,cmap='YlOrRd',vmin = Ti, vmax = 300, zorder=2)

        ax.set_axis_off()
        ax.add_patch(Rectangle((0,0),b,h,fill=True, facecolor = 'black', alpha = 0.75, hatch='/'))


    ## Calculate average temperature of each mesh element (fiber) and group
    ## into patches based on those temperatures

    # Mesh size
    mDim = z.shape

    # Initialize an empty list of fiber temperatures
    fbT = []

    # Loop through the each point in the meshgrid and calculate the average
    # temperature at four adjacent points that define that element
    for idR in range(mDim[0]-1):
        tmp = []
        for idC in range(mDim[1]-1):
            Tmean = np.mean(z[idR: idR + 2, idC: idC + 2])
            tmp.append(round(Tmean,12))
        fbT.append(tmp)

    # Replace NaN values (char layer) with the burning temperature
    fbT = np.where(np.isnan(fbT), Tp, fbT)

    # Define dictionary of unique fiber temperatures and the corresponding
    # material tags
    mattag = np.unique(fbT)
    mattag = {k:(v+1) for v, k in enumerate(mattag)}

    # Calculate number of temperature layers given  mesh size
    Nlayers = int(0.5*min(b,h)/mSize)

    # Initialize an empty list of patch parameters
    patchList = []  # (mattag, ny, nz, coord1, coord2)

    # Loop through each layer and append patch parameters to the list
    for i in range(Nlayers):

        # Check if we are in the core, which can be defined as one rectangular
        # patch. This is because beyond this point nomore layers are needed,
        # as temperature remains unchanged with depth
        if fbT[i][i] == Ti:
            c1,c2 = [(mSize*(i),mSize*(i)), (b-mSize*(i),h-mSize*(i))]
            ny,nz = [(b-2*mSize*(i))/mSize,(h-2*mSize*(i))/mSize]
            patchList.append([mattag[Ti],ny,nz,c1,c2])
            break

        # Otherwise define four corners and two patches in each horintal and
        # vertical directions
        else:
            # Define 4 corners
```

```
            ny,nz = [1,1]

            coord1,coord2 = [(mSize*(i),mSize*(i)), (mSize*(i+1),mSize*(i+1))]
            patchList.append([mattag[fbT[i][i]],ny,nz,coord1,coord2])

            coord1,coord2 = [(b-mSize*(i+1),mSize*(i)),(b-mSize*(i),mSize*(i+1))]
            patchList.append([mattag[fbT[i][i]],ny,nz,coord1,coord2])

            coord1,coord2 = [(mSize*(i),h-mSize*(i+1)), (mSize*(i+1),h-mSize*(i))]
            patchList.append([mattag[fbT[i][i]],ny,nz,coord1,coord2])

            coord1,coord2 = [(b-mSize*(i+1),h-mSize*(i+1)),(b-mSize*(i),h-mSize*(i))]
            patchList.append([mattag[fbT[i][i]],ny,nz,coord1,coord2])


            # Define 2 horizontal fibers
            ny,nz = [(b-2*mSize*(i+1))/mSize,1]

            c1,c2 = [(mSize*(i+1),mSize*(i)), (b-mSize*(i+1),mSize*(i+1))]
            patchList.append([mattag[fbT[i][i+1]],ny,nz,c1,c2])

            c1,c2 = [(mSize*(i+1),h-mSize*(i+1)), (b-mSize*(i+1),h-mSize*(i))]
            patchList.append([mattag[fbT[i][i+1]],ny,nz,c1,c2])

            # Define 2 vertical fibers
            ny,nz = [1,(h-2*mSize*(i+1))/mSize]

            c1,c2 = [(mSize*(i),mSize*(i+1)), (mSize*(i+1),h-mSize*(i+1))]
            patchList.append([mattag[fbT[i+1][i]],ny,nz,c1,c2])

            c1,c2 = [(b-mSize*(i+1),mSize*(i+1)), (b-mSize*(i),h-mSize*(i+1))]
            patchList.append([mattag[fbT[i+1][i]],ny,nz,c1,c2])

    return patchList, mattag
```

```
# Function to calculate temperature at depth (x) from the surface
#
# ---------------------------------------------------------------------------
# INPUT:
#   x           = depth from the surface
#   a           = temperature penetration depth (mm)
#   c           = char depth (mm)
#   Ti          = ambient temperature (Celsius)
#   Tp          = surface temperature (Celsius)
# ---------------------------------------------------------------------------
#
# ---------------------------------------------------------------------------
# OUTPUT:
#   T           = temperature at depth (x) from the surface (NaN if char)
# ---------------------------------------------------------------------------
#

def Temp(x,a,c,Ti,Tp):
    xi = x-c
    xi[xi<0] = np.NaN
    T = Ti+(Tp-Ti)*((1-xi/a).clip(0))**2
    return T
```

Appendix B: OpenSeesPy Model Analysis

```python
import openseespy.opensees as ops
import numpy as np
import FiberSection

# Function to perform 2nd-order inelastic analysis of the column in OpenSeesPy
#
# ----------------------------------------------------------------------------
# INPUT:
#   t           = time of exposure
#   b           = section width (mm)
#   h           = section height (mm)
#   L           = column height (mm)
#   Ti          = ambient temperature (Celsius)
#   Tp          = surface temperature (Celsius)
#   a           = temperature penetration depth (mm)
#   mSize       = mesh size (mm)
#   P           = applied vertical load (N)
#   numIncr     = number of load increments
#   plot        = boolean to display section temperature profile plot
#   disp_fiber  = boolean to display mesh on the plot
#   side        = number of sides from which fire is applied (either 1 or 4)
# ----------------------------------------------------------------------------
#
# ----------------------------------------------------------------------------
# OUTPUT:
#   appliedLoad = list of the total applied loads at each load step
#   nodeDisp    = list of the horizontal middle node displacements at each
#                 load step
# ----------------------------------------------------------------------------
#

def RunAnalysis(t,b,h,L,Ti,Tp,a,mSize,e,P,numIncr,plot=False,disp_fiber=False,side=4):

    ## Initialize the 2D model with three DOFs at each node
    ops.wipe()
    ops.model('basic','-ndm',2,'-ndf',3)

    ## Calculate temperature profile, define unique temperature tags and group section patches
    [FiberList, MatList] = FiberSection.GetFibers(t, b, h, Ti, Tp, a, mSize, plot, disp_fiber,  side)

    ## Define material properties
    fcu = 45
    fcr = fcu*0.75
    ftu = -80
    Ew = fcu/4e-3
    Eq = (fcu-fcr)/(4e-3-13e-3)

    # Define unique materials from temperature tags
    for T, mattag in MatList.items():

        # Update material properties given the temperature
        [s3t,s2t,s1t,s1c,s2c,s3c], [e3t,e2t,e1t,e1c,e2c,e3c] = StressStrain(T,Ti,fcu,fcr,ftu,Ew,Eq)

        # Define hysteretic material

 ops.uniaxialMaterial('Hysteretic',mattag,s1c,e1c,s2c,e2c,s3c,e3c,s1t,e1t,s2t,e2t,s3t,e3t,1,1,1,1)

    ## Define fiber section
    sectag = 1
```

```python
    ops.section('Fiber',sectag)

    for fiber in FiberList:
        mattag, ny, nz, coord1, coord2 = fiber
        ops.patch('rect',mattag,ny,nz,*coord1,*coord2)

    ## Create nodes
    ops.node(1,0,0)
    ops.node(2,e,L/2)
    ops.node(3,0.0,L)

    ## Define fixities (pin-pin)
    ops.fix(1,1,1,0)
    ops.fix(3,1,0,0)

    ## Create fiber-based elements
    ops.geomTransf('PDelta',1)
    numIntgrPts = 10
    ops.beamIntegration('Lobatto',1,sectag,numIntgrPts)

    ops.element('forceBeamColumn',1,1,2,1,1)
    ops.element('forceBeamColumn',2,2,3,1,1)

    ## Apply loads
    ops.timeSeries('Linear', 1)
    ops.pattern('Plain',1,1)
    ops.load(3,0,-P,0)

    # Load step increment (N)
    dP = 1/numIncr

    ## Define recorders
    ops.recorder('Node','-file','disp_node2.out','-node',2,'-dof',1,'disp')
    ops.recorder('Node','-file','reaction_node1.out','-node',1,'-dof',2,'reaction')
    ops.recorder('Node','-file','reaction_node3.out','-node',3,'-dof',2,'reaction')

    ## Set up analysis
    ops.constraints('Transformation')
    ops.numberer('RCM')
    ops.system('BandGeneral')
    ops.test('NormUnbalance',1e-6,100)
    ops.algorithm('Newton')
    ops.integrator('LoadControl',dP)
    ops.analysis('Static')

    ## Run analysis
    ops.record()
    ok = ops.analyze(numIncr)

    ## Report analysis status
    if ok == 0: print("Analysis done.")
    else:       print("Convergence issue.")

    ops.wipe()

    ## Compile output
    nodeDisp = abs(np.loadtxt('disp_node2.out'))
    reaction_node1 = np.loadtxt('reaction_node1.out')
    reaction_node3 = np.loadtxt('reaction_node3.out')
    appliedLoad = reaction_node1+reaction_node3

    return appliedLoad, nodeDisp
```

```python
# Function to calculate temperature dependent material properties
#
# ---------------------------------------------------------------------------
# INPUT:
#   T           = temperature of the material (Celsius)
#   Ti          = ambient temperature (Celsius)
#   fcu         = ultimate compressive stress (MPa)
#   fcr         = compressive stress at rupture (MPa)
#   ftu         = ultimate tensile stress (MPa)
#   Ew          = modulus of elasticity (MPa)
#   Eq          = modulus of elasticity, softening (MPa)
# ---------------------------------------------------------------------------
#
# ---------------------------------------------------------------------------
# OUTPUT:
#   stress      = list of stress points
#   strain      = list of strain points
# ---------------------------------------------------------------------------
#

def StressStrain(T,Ti,fcu,fcr,ftu,Ew,Eq):

    # Strength temperature reduction factors (Eurocode 5)
    kSC = min(1,1-(T-Ti)*0.35/(100-Ti)) if T<100 else max(0.0001,0.65-(T-100)*0.65/200)
    kST = min(1,1-(T-Ti)*0.60/(100-Ti)) if T<100 else max(0.0001,0.40-(T-100)*0.40/200)

    # Modulus of elasticity temperature reduction factors (Eurocode 5)
    kEC = min(1,1-(T-Ti)*0.65/(100-Ti)) if T<100 else max(0.0001,0.35-(T-100)*0.35/200)
    kET = min(1,1-(T-Ti)*0.50/(100-Ti)) if T<100 else max(0.0001,0.50-(T-100)*0.50/200)

    # Update modulus of elasticity values
    Ew_t = Ew*kET
    Ew_c = Ew*kEC
    Eq *= kEC

    # Update strength values
    fcu *= kSC
    fcr *= kSC
    ftu *= kST

    # Stress points (t = tension, c = compression)
    stress = [0,0,ftu,fcu,fcr,0]

    # Strain points (t = tension, c = compression)
    e1c = fcu/Ew_t
    e2c = e1c + (fcr-fcu)/Eq
    e3c = e2c + 1e-8

    e1t = ftu/Ew_c
    e2t = e1t - 1e-8
    e3t = e2t - 1e-8

    strain = [e3t,e2t,e1t,e1c,e2c,e3c]

    return stress,strain
```

Appendix C: Timber Fire Analysis Examples

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import FiberSection as fb
import OpenSeesAnalysis as ops
import matplotlib.pyplot as plt
import numpy as np

# ----------------------------------------------------------------------------
# NOTE:
#
# The following imports developed OpenSeesAnalysis and FiberSection files and
# displays the following outputs:
#   1. Section temperature profile under one-sided fire exposure
#   2. Section temperature profile under four-sided fire exposure
#   3. Applied total load vs lateral displacement plot (pin-pin column)
#   4. Critical load vs time of exposure to fire plot (pin-pin column)
#
#   * Please allow several minutes for the code to run.
#   ** Convergence issue displayed means that limit point has been reached for
#      that particular analysis and the warning message can be ignored.
# ----------------------------------------------------------------------------
#

#%%# Input Parameters

# Mesh size
mSize = 5

# Section dimensions (mm)
b = 400
h = 400

# Temperature distribution at distance x from the char front
Ti = 20     # Celcius
Tp = 300    # Celcius
a = 35      # mm

# Column height and eccentricity
L = 4000 #mm
e = L/1000

# Applied Load
P = 15000e3
numIncr = 1000
```

```
#%% Example 1: Plot fiber section at time t
plot = True
disp_fiber = True

# Time of exposure (minutes)
t = 120

# One-sided exposure
[FiberList, MatList] = fb.GetFibers(t, b, h, Ti, Tp, a, mSize, plot, disp_fiber, 1)

# Four-sided exposure
[FiberList, MatList] = fb.GetFibers(t, b, h, Ti, Tp, a, mSize, plot, disp_fiber, 4)
```

```
#%% Example 2: Structural response of a pin-pin column under four-side fire exposure
n = 25
tList = np.linspace(0,120,n)
plt.rcParams["axes.prop_cycle"] = plt.cycler("color", plt.cm.coolwarm(np.linspace(0,1,n)))

Pmax = []

# Calculate critical load at ambient temperature
force,_ = ops.RunAnalysis(0,b,h,L,Ti,Ti,a,mSize,e,P,numIncr)
P0 = max(force)

# Set up plot
fig, ax = plt.subplots(dpi=175)
fig.set_size_inches(7,7)
ax.set( xlabel = 'Lateral displacement of middle node (mm)', ylabel = 'Applied total load (kN)')
ax.grid('on')
ax.minorticks_on()
ax.grid(b = True, which = 'major', linestyle='-', linewidth=0.5, alpha=0.75)
ax.grid(b = True, which = 'minor', color = 'gray', linestyle='--', linewidth=0.25, alpha=0.5)

# Loop through time steps
for t in tList:
    [force, disp] = ops.RunAnalysis(t,b,h,L,Ti,Tp,a,mSize,e,P,numIncr)
    Pmax.append(max(force)/1000)
    ax.plot(disp,force/1000,linewidth = 2, label = 't = ' + str(t) + ' min')

fig.legend(framealpha=1)
fig.tight_layout()
```

```
#%% Plot Critical Load Reduction Factor
fig, ax = plt.subplots(dpi=175)
fig.set_size_inches(7,7)
ax.set(xlabel = 'Time of exposure to fire / mins', ylabel = 'Critical Load Reduction Factor',
        ylim = [0,0.85], yticks = np.arange(0,1.05,.05), xticks = np.arange(0,200,20))
ax.grid('on')
ax.minorticks_on()
ax.grid(b = True, which = 'major', linestyle='-', linewidth=0.5, alpha=0.75)
ax.grid(b = True, which = 'minor', color = 'gray', linestyle='--', linewidth=0.25, alpha=0.5)
ax.plot(tList,[i/P0*1000 for i in Pmax], color = 'black', linewidth = 2)
fig.tight_layout()
```