THE UNIVERSITY
of EDINBURGH

An extended travelling fire method framework with an OpenSees-based integrated tool SIFBuilder

XU DAI

Doctor of Philosophy

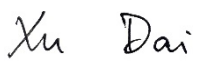
The University of Edinburgh

2017

To Chengchun, Yumin, and Sha

Declaration

This thesis and the research described and reported within has been completed solely by Xu Dai under the supervision of Dr. Stephen Welch and Professor Asif Usmani. Where other sources are quoted, full references are given. This work has not been submitted for any other degree or professional qualification.



Xu Dai

November 2017

Abstract

Many studies of the fire induced thermal and structural behaviour in large compartments, carried out over the past two decades, show a great deal of non-uniformity, unlike the homogeneous compartment temperature assumption in the current fire safety engineering practice. Furthermore, some large compartment fires may burn locally and they tend to move across entire floor plates over a period of time as the fuel is consumed. This kind of fire scenario is beginning to be idealized as 'travelling fires' in the context of performance-based structural and fire safety engineering. However, the previous research of travelling fires still relies on highly simplified travelling fire models (i.e. Clifton's model and Rein's model); and no equivalent numerical tools can perform such simulations, which involves analysis of realistic fire, heat transfer and thermo-mechanical response in one single software package with an automatic coupled manner. Both of these hinder the advance of the research on performance-based structural fire engineering. The author develops an extended travelling fire method (ETFM) framework and an integrated comprehensive tool with high computational expediency in this research, to address the above-mentioned issues.

The experiments conducted for characterizing travelling fires over the past two decades are reviewed, in conjunction with the current available travelling fire models. It is found that no performed travelling fire experiment records both the structural response and the mass loss rate of the fuel (to estimate the fire heat release rate) in a single test, which further implies closer collaboration between the structural and the fire engineers' teams are needed, especially for the travelling fire research topic. In addition, an overview of the development of OpenSees software framework for modelling structures in fire is presented, addressing its theoretical background, fundamental assumptions, and inherent limitations. After a decade of development, OpenSees has modules including fire, heat transfer, and thermo-mechanical analysis. Meanwhile, it is one of the few structural fire modelling software which is open source and free to the entire community, allowing interested researchers to use and contribute with no expense.

An OpenSees-based integrated tool called SIFBuilder is developed by the author and co-workers, which can perform fire modelling, heat transfer analysis, and thermo-

mechanical analysis in one single software with an automatic coupled manner. This manner would facilitate structural engineers to apply fire loading on their design structures like other mechanical loading types (e.g. seismic loading, gravity loading, etc.), without transferring the fire and heat transfer modelling results to each structural element manually and further assemble them to the entire structure. This feature would largely free the structural engineers' efforts to focus on the structural response for performance-based design under different fire scenarios, without investigating the modelling details of fire and heat transfer analysis. Moreover, the efficiency due to this automatic coupled manner would become more superior, for modelling larger structures under more realistic fire scenarios (e.g. travelling fires). This advantage has been confirmed by the studies carried out in this research, including 29 travelling fire scenarios containing total number of 696 heat transfer analysis for the structural members, which were undertaken at very modest computational costs. In addition, a set of benchmark problems for verification and validation of OpenSees/SIFBuilder are investigated, which demonstrates good agreement against analytical solutions, ABAQUS, SAFIR, and the experimental data. These benchmark problems can also be used for interested researchers to verify their own numerical or analytical models for other purposes, and can be also used as an induction guide of OpenSees/SIFBuilder.

Significantly, an extended travelling fire method (ETFM) framework is put forward in this research, which can predict the fire severity considering a travelling fire concept with an upper bound. This framework considers the energy and mass conservation, rather than simply forcing other independent models to 'travel' in the compartment (i.e. modified parametric fire curves in Clifton's model, 800°C-1200°C temperature block and the Alpert's ceiling jet in Rein's model). It is developed based on combining Hasemi's localized fire model for the fire plume, and a simple smoke layer calculation by utilising the FIRM zone model for the areas of the compartment away from the fire. Different from mainly investigating the thermal impact due to various ratios of the fire size to the compartment size (e.g. 5%, 10%, 25%, 75%, etc.), as in Rein's model, this research investigates the travelling fire thermal impact through explicit representation of the various fire spread rates and fuel load densities, which are the key input parameters in the ETFM framework. To represent the far field thermal exposures, two zone models

(i.e. ASET zone model & FIRM zone model) and the ETFM framework are implemented in SIFBuilder, in order to provide the community a 'vehicle' to try, test, and further improve this ETFM framework, and also the SIFBuilder itself.

It is found that for 'slow' travelling fires (i.e. low fire spread rates), the near-field fire plume brings more dominant thermal impact compared with the impact from far-field smoke. In contrast, for 'fast' travelling fires (i.e. high fire spread rates), the far-field smoke brings more dominant thermal impact. Furthermore, the through depth thermal gradients due to different travelling fire scenarios were explored, especially with regards to the 'thermal gradient reversal' due to the near-field fire plume approaching and leaving the design structural member. This 'thermal gradient reversal' would fundamentally reverse the thermally-induced bending moment from hogging to sagging. The modelling results suggest that the peak thermal gradient due to near-field approaching is more sensitive to the fuel load density than fire spread rate, where larger peak values are captured with lower fuel load densities. Moreover, the reverse peak thermal gradient due to near-field leaving is also sensitive to the fuel load density rather than the fire spread rate, but this reverse peak value is inversely proportional to the fuel load densities. Finally, the key assumptions of the ETFM framework are rationalised and its limitations are emphasized. Design instructions with relevant information which can be readily used by the structural fire engineers for the ETFM framework are also included. Hence more optimised and robust structural design under such fire threat can be generated and guaranteed, where we believe these efforts will advance the performance-based structural and fire safety engineering.

Publications

Journal papers

X. Dai, S. Welch, A. Usmani, A critical review of "travelling fire" scenarios for performance-based structural engineering. Fire Safety Journal 91C (2017) pp. 568-578.

Conference papers

Xu Dai, Stephen Welch, Asif Usmani, 'Structural implications due to an extended travelling fire methodology (ETFM) framework using SIFBuilder', the 10th International Conference on Structures in Fire (SiF 2018), 6-8 June 2018, FireSERT, Ulster University, Belfast, UK, pp455-462.

Xu Dai, Yaqiang Jiang, Liming Jiang, Stephen Welch, Asif Usmani, 'Implementation of Fire Models in OpenSees', 1st European Conference on OpenSees, 19-20 June 2017, Porto, Portugal, pp47-50.

Xu Dai, Liming Jiang, Jamie Maclean, Stephen Welch, Asif Usmani, 'A Conceptual Framework for a Design Travelling Fire for Large Compartments with Fire Resistant Islands', 14th International Interflam Conference, 4-6 July 2016, London, UK, vol (2), pp1039-1050.

Xu Dai, Liming Jiang, Jamie Maclean, Stephen Welch, Asif Usmani, 'Implementation of a New Design Travelling Fire Model for Global Structural Analysis', 9th International Conference on Structures in Fire (SiF 2016), 8-10 June 2016, Princeton University, New Jersey, USA, pp959-966.

Jiayu Hu, **Xu Dai**, Asif Usmani, Ricky Carvel, 'Design fires for performance-based engineering of bridges', the 10th International Conference on Structures in Fire (SiF 2018), 6-8 June 2018, FireSERT, Ulster University, Belfast, UK, pp3-10.

Liming Jiang, **Xu Dai**, Asif Usmani and Praveen Kamath, 'OpenSees-based Integrated Tool for Modelling Structures in Fire', 1st International Conference on Structural Safety under Fire & Blast, 2-4 September 2015, Glasgow, Scotland, UK, pp461-468.

Payam Khazaeinejad, **Xu Dai** and Asif Usmani, 'Analysis of Heated Beams: Modelling Benchmarks', 1st International Conference on Structural Safety under Fire & Blast, 2-4 September 2015, Glasgow, Scotland, UK, pp469-473.

Marion Charlier, Antonio Gamba, **Xu Dai**, Stephen Welch, Olivier Vassart, Jean-Marc Franssen, 'CFD analyses used to evaluate the influence of compartment geometry on the possibility of development of a travelling fire', the 10th International Conference on Structures in Fire (SiF 2018), 6-8 June 2018, FireSERT, Ulster University, Belfast, UK, pp341-348.

Acknowledgements

I would like to express my sincere gratitude to my supervisors Dr. Stephen Welch and Prof. Asif Usmani for their strong support, expert guidance, and consistent encouragement during the past four years. Prof. Asif Usmani opened the gate of numerical modelling of structures in fire for me, and further inspired me on doing research through coding. Dr. Stephen Welch guided me on fire dynamics and shared me his great insights on fire modelling, which sailed my research to the right end. Their wide knowledge and endless hard working would always be an example for me.

I would also like to thank Prof. Luke Bisby for his support as the fire group leader. I want to thank Dr. Ricky Carvel for sharing his suggestions and advices.

Thanks to Liming Jiang for his great help during the early stage of my research on understanding and programming OpenSees. Thanks to Payam Khazaeinejad for the long discussions and collaborations about these benchmarking problems. Thanks to Praveen Kamath for his help in many occasions.

Thanks to Emran Baharudin for the enjoyable discussions on research and life, and also his openness on collaborations. Special thanks to Jamie Maclean, who is always willing to listen, share, and collaborate in many occasions, especially for his industrial insights on the development of travelling fires in my research. Thanks to Benjamin Ralph for his help on the zone modelling approach utilised in this research. Thanks to Jiayu Hu for her help on using ABAQUS for benchmarking OpenSees.

Thanks to Eric Muller, Mo Houssami, Jan Christian Thomas, Zaid Al-Azzawi, Mohamed Kiari, Daryan Othman, Emma McIntyre, Liam Ingram, and Ieuan Rickard, who were my first office colleagues and friends in John Muir 1.4. Final thanks all the colleagues in the BRE Centre for Fire Safety Engineering, who built up such a good research environment during the past four years in my life.

Contents

List of Figures

List of Tables

Chapter 1.

Introduction

1.1 BACKGROUND

In structural fire design, a key principle that every engineer must follow is to ensure that the fire resistance ability of the structure is larger than the impact of the fire severity. In order to satisfy this principle quantitatively, rather than just qualitatively, there are typically three design domains that structural fire engineers can follow: the time domain, the temperature domain, and the strength domain. In fact these three design domains are interchangeable if the same structural failure criterion is adopted [1]. However, the reliabilities of these respective methods would diminish when different fire exposure models are used as the fire severity input. For example, the standard time-temperature curves (e.g. ISO-834 standard fire [2]) are normally adopted for the structural fire design in the time domain. It is assumed that all the structural members in the compartment share the same temperature histories. This is a reasonable assumption when the compartment size is relatively small. But in other cases, such as vehicles burning in an open car park, localised fire models (e.g. Hasemi localised fire model [3]) may be considered to be more appropriate as the fire severity input for the structural fire design. Then the design in time domain becomes inappropriate compared with the design in the temperature domain, or in the strength domain for this case. Hence it is apparent that this dissimilation basically arises from the increasing level of complexity of the structural layout and corresponding appropriate fire scenarios.

This situation would become worse when the design compartment is so large that no existing fire exposure model can be readily be used by the structural engineers. It implies that even if the design satisfies the criteria in the strength domain, that will not guarantee its reliability, due to the unknown fire severity paired with the large design compartment. The structural failure events observed with high fire inhomogeneity have been reported several times such as: the WTC buildings in New York in 2001 [4], the Windsor Tower in Madrid in 2005 [5], the Faculty of TU Delft Architecture building in Netherlands in 2008 [6], and more recently the Plasco building in Tehran in 2017 [7]. These facts underline the urgent need for a better description of fire scenarios for structural design, recognising the radically different spatial layouts preferred in contemporary architecture.

One possible solution is performing Computational Fluid Dynamics (CFD) simulations, which can be used as the fire severity input for structural design. However, using CFD is not feasible on the day-to-day routine design basis for structural engineers, due to the massive computational expenses required. Moreover, the high fidelity and uncertainties that CFD models would generate, may become to unnecessary or even misleading thermal input information for the structural engineers, as it requires professional fire science knowledge to interpret and judge these results.

Another potential solution is exploiting this type of fire scenarios by developing a simple design framework, to address the problem in a practical manner, enabling the structural engineers to utilise the fire design concept without resorting to large computational calculations. An appropriate and efficient level of detail in the model is required to handle these fire scenarios realistically. The work in this PhD thesis is developed on this basis. Moreover, this kind of fire scenario is beginning to be idealized more widely as so called 'travelling fires' in the context of performance-based structural and fire safety engineering. Nevertheless, the research of travelling fires has hitherto relied on two oversimplified travelling fire models (i.e. Clifton's model [8] and Rein's model[9]), which are developed by simply forcing other existing models to 'travel' across a floor plan, without considering the essential energy and mass balances.

Accompanying with the advancement of design fires, however, there are no matchable numerical tools can perform integrated thermal and structural simulations with high computational efficiency, which involves analysis of realistic fire, heat transfer and thermo-mechanical response in one single software package with an automatic coupled manner. The Open System for Earthquake Engineering Simulation (OpenSees) is a C++ object-oriented, open source software framework developed at the University of California, Berkeley by McKenna in the late 1990s [10]. It was originally for providing an advanced finite-element simulation tool to perform structural and geotechnical analysis under seismic loadings. It has now become a common platform, for researchers within the Pacific Earthquake Engineering Research Center (PEER, which is a multi-institutional research and education centre in US), for the development, sharing, and dissemination of new ideas to further earthquake engineering research around the world. Three key references including the OpenSees user manual [11], the

OpenSees main website [12], and McKenna's PhD thesis [10], have reached citation numbers to 764, 898, and 330 respectively, according to the latest records from Google Scholar in September 2017. However, in the research community of structural fire engineering, the available software are normally lacking in code transparency, have limited modelling capabilities, but with high purchase expenses. In general, there are two types of computer programs for simulating structural behaviours in fire: commercially-oriented and research-oriented. Commercial software (e.g. ABAQUS, ANSYS and DIANA) normally offer advantages such as extensive verifications and validations, user-friendly interfaces for pre-processing and post-processing, and professional support for maintenance. Nevertheless, there are inherent limitations. For instance, development of commercial codes is often dictated by the requirements of the most profitable applications and rarely addresses the needs of academic researchers, thus developments typically lag well behind research. An alternative is to use proprietary software developed by researchers. Well known examples of such codes in the structural fire engineering community are: SAFIR [13] from University of Liège, Vulcan [14] from University of Sheffield, ADAPTIC [15] from Imperial College London, each for analysis of structures subjected to fire (and earthquakes in the case of ADAPTIC). These codes typically suffer from tightly bound architecture as a result of using procedural programming. Furthermore, because they are often developed by a small dedicated team of researchers at the original host institution, they are not designed or suited for a devolved community of developers, and the codes are not open source. They also typically have uncertain resourcing and great dependency on key individuals for support, maintenance and development. Hence, they cannot generally be considered sustainable in the long term.

The limitations discussed above can in principle be overcome by the development of open source software using objected-oriented programming (OOP), in which OpenSees is an ideal option. Due to the above reasons and inspired by the success of OpenSees in its own research community, Usmani *et al.* [16] at the University of Edinburgh initiated the adaptation of OpenSees for analysing structural response subjected to fire in 2008. After nearly ten years of development, OpenSees now has modules which can perform fire modelling [17, 18], heat transfer analysis [17, 18], and thermo-mechanical analysis

for large structures under realistic fire scenarios [18–21]. The research publication outcome contains 5 PhD theses [17–21], and 11 journal papers [16, 22–30]. In addition, OpenSees as an open source software tool for research [31], can promote a sense of community and facilitate greater collaboration between research groups with similar interests irrespective of geographical location. Moreover, OOP using C++ permits OpenSees to fit in to the research environment better. This is because it permits researchers to quickly view and gain an understanding of the workings of the codes, based on their own research interests on the specific module, instead of going through all the procedures and functions as required using procedural programming (e.g. FORTRAN). However, there are no previous integrated thermal and structural modelling software in the current community, including OpenSees. It means the users have to transfer the thermal modelling results to each individual structural member manually, then assemble them into the structural model to analyse the global structural behaviour. This would largely distract the structural design flow and efficiency, and even reduce the robustness of corresponding structural design due to the complexities through coupling advanced design fires (e.g. travelling fires) to a complex structural model.

1.2 OBJECTIVES AND ORIGINALITY

This research aims to address some of the issues as discussed above, with a particular focus on developing an appropriate representation of fire in a generalised framework for structural design, along with an integrated numerical tool to perform efficient structural fire analysis. The objectives of this research include:

- Develop an extended travelling fire method framework to determine the fire severity in large compartments for performance-based structural design.
- Develop and benchmark a thermal and structural coupled tool for modelling structures under travelling fires.
- Investigate the thermal and structural response of a steel composite structure under the developed travelling fire method framework with the coupled tool.

1.3 LAYOUT OF THESIS

Chapter 2 presents a literature review of the large-scale experiments conducted for characterizing travelling fires, in conjunction with the current analytical travelling fire models. The limitations of these experiments are emphasized, and the features of different travelling fire models are compared.

Chapter 3 presents an overview of the developed OpenSees modules including fire modelling, heat transfer analysis, and thermo-mechanical analysis. It summarizes OpenSees modelling capabilities, addressing its theoretical background, fundamental assumptions, and inherent limitations. An up-to-date reference list for the corresponding OpenSees thermal and structural modules is provided, for the users and developers to refer to the current updated literature.

Chapter 4 presents an extended travelling fire method (ETFM) framework, which can predict the fire severity in the light of a travelling fire concept with an upper bound, through taking the energy and mass conservations into account. It is developed by 'mobilising' Hasemi's localized fire model for the fire plume, and combining with a simple smoke layer calculation by utilising the FIRM zone model for the areas of the compartment away from the fire. The key assumptions of the ETFM framework are rationalised and its limitations are emphasized. In addition, design instructions with relevant information which can be readily used by the structural fire engineers are also included.

Chapter 5 presents an integrated tool (i.e. SIFBuilder), including its key components, model generation, available material and element libraries, fire loading types, heat transfer scheme, and data transmission. The difference between OpenSees and SIFBuilder is emphasized, and relevant online documentation information is also provided. A series of benchmark problems are investigated for further verifying and validating OpenSees and SIFBuilder, against analytical solutions, ABAQUS, SAFIR, and experimental data.

Chapter 6 presents the implementations of smoke zone models and the ETFM framework in SIFBuilder. A case study using the zone models in SIFBuilder is

investigated, for validating its results against the original ASET-QB and FIRM-QB software [32].

Chapter 7 presents the investigations of the impact of the ETFM parameters (i.e. fire spread rate, fuel load density) in the temperature design domain, through quantifying the cross-sectional time-temperature evolution of the instrumented steel beam with a case study. A total of 29 travelling fire scenarios are investigated, and 696 heat transfer analysis performed. Furthermore, the through depth thermal gradients due to different travelling fire scenarios are extensively explored, especially with regards to the 'thermal gradient reversal' due to the near-field fire plume approaching and leaving the design structural member. The thermal response implications on the subsequent structural responses (e.g. the temperature rise induced thermal expansion axial force, the change of through depth induced thermal bowing bending moment) are also discussed in this chapter. Further, a demonstration of a full 3D steel-framed structure under the fire impact due to the ETFM framework using SIFBuilder is presented. The impact of the travelling fires on this structure is investigated, through qualitatively demonstrating the structural response of several steel beams in the large compartment.

Chapter 8 presents a summary of the key findings of this research, and recommendations of the future work.

1.4 REFERENCES

[1] A. H. Buchanan and A. K. Abu, *Structural Design for Fire Safety*, 2nd Ed. John Wiley and Sons, Ltd., 2017.

[2] ISO 834-1:1999(E), "Fire-resistance tests - Elements of building construction - Part 1: General requirements." International Organization for Standardization, 1999.

[3] Y. Hasemi, Y. Yokobayashi, T. Wakamatsu, and A. V. Ptchelintsev, "Modeling of heating mechanism and thermal response of structural components exposed to localized fires: a new application of diffusion flame modeling to fire safety engineering," in *Thirteenth meeting of the UJNR planel on fire research and safety*, 1996, pp. 237–247.

[4] A. S. Usmani, Y. C. Chung, and J. L. Torero, "How did the WTC towers collapse: A new theory," *Fire Safety Journal*, vol. 38, no. 6, pp. 501–533, 2003.

[5] I. Fletcher, A. Borg, N. Hitchen, and S. Welch, "Performance of concrete in fire: A review of the state of the art, with a case study of the Windsor Tower fire," in *4th International Workshop in Structures in Fire*, pp. 779–790.

[6] D. M. Zannoni, J. G. H. Bos, D. K. E. Engel, and P. dr. U. Rosenthal, "Brand bij Bouwkunde," COT Instituut voor Veiligheids - en Crisismanagement, 2008.

[7] "Plasco Building Fire in Tehran," *Wikipedia*, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Plasco_Building. [Accessed: 08-Oct-2017].

[8] C. G. Clifton, "Fire Models for Large Firecells. HERA Report R4-83," HERA publications, New Zealand, 1996.

[9] E. Rackauskaite, C. Hamel, A. Law, and G. Rein, "Improved formulation of travelling fires and application to concrete and steel structures," *Structures*, vol. 3, pp. 250–260, 2015.

[10] F. T. McKenna, "Object-Oriented Finite Element Programming: Frameworks for Analysis, Algorithms and Parallel Computing," PhD Thesis, University of California, 1997.

[11] S. Mazzoni, F. Mckenna, M. H. Scott, and G. L. Fenves, "Open System for Earthquake Engineering Simulation (OpenSees): User Command-Language Manual," *Pacific Earthquake Engineering Research Center, University of California, Berkeley*, 2006. [Online]. Available: http://opensees.berkeley.edu/OpenSees/manuals/usermanual/index.html. [Accessed: 02-Sep-2017].

[12] F. McKenna, G. L. Fenves, and M. H. Scott, "Open System for Earthquake Engineering Simulation (OpenSees)," *Pacific Earthquake Engineering Research Center*, 2000. [Online]. Available: http://opensees.berkeley.edu. [Accessed: 02-Sep-2017].

[13] J. Franssen and T. Gernay, "Modeling structures in fire with SAFIR: theoretical background and capabilities," *Journal of Structural Fire Engineering*, vol. 8, no. 3, pp. 300–323, 2017.

[14] "Vulcan Software Package," *Vulcan Solutions Ltd.*, 2005. [Online]. Available: www.vulcan-solutions.com. [Accessed: 04-Sep-2017].

[15] B. Izzuddin, "Nonlinear dynamic analysis of framed structures," PhD Thesis, Department of Civil Engineering, Imperial College London, 1991.

[16] A. Usmani, J. Zhang, J. Jiang, Y. Jiang, and I. May, "Using OpenSees for structures in fire," *Journal of Structural Fire Engineering*, vol. 3, no. 1, pp. 57–70, 2012.

[17] Y. Jiang, "Development and Application of A Thermal Analysis Framework in OpenSees for Structures in Fire," PhD Thesis, School of Engineering, University of Edinburgh, 2012.

[18] L. Jiang, "Development of An Integrated Computational Tool for Modelling Structural Frames in Fire Considering Local Effects," PhD Thesis, School of Engineering, University of Edinburgh, 2016.

[19] J. Zhang, "Developing OpenSees Software Framework for Modelling Structures in Fire," PhD Thesis, School of Engineering, University of Edinburgh, 2014.

[20] J. Jiang, "Nonlinear Thermomechanical Analysis of Structures Using OpenSees," PhD Thesis, School of Engineering, University of Edinburgh, 2012.

[21] P. Kotsovinos, "Analysis of the Structural Response of Tall Buildings under Multifloor and Travelling Fires," PhD Thesis, School of Engineering, University of Edinburgh, 2013.

[22] P. Kotsovinos, Y. Jiang, and A. Usmani, "Effect of vertically travelling fires on the collapse of tall buildings," *International Journal of High Rise Buildings*, vol. 2, no. 1, pp. 49–62, 2013.

[23] J. Jiang and A. Usmani, "Modeling of steel frame structures in fire using OpenSees," *Computers & Structures*, vol. 118, pp. 90–99, 2013.

[24] P. Kotsovinos and A. Usmani, "The World Trade Center 9/11 disaster and progressive collapse of tall buildings," *Fire Technology*, vol. 49, pp. 741–765, 2013.

[25] J. Jiang, A. Usmani, and G. Li, "Modelling of steel-concrete composite structures in fire using OpenSees," *Advances in Structural Engineering*, vol. 17, no. 2, pp. 249–264, 2014.

[26] J. Jiang, G. Li, and A. Usmani, "Progressive collapse mechanisms of steel frames exposed to fire," *Advances in Structural Engineering*, vol. 17, no. 3, pp. 381–398, 2014.

[27] J. Jiang, G. Li, and A. Usmani, "Analysis of composite steel-concrete beams exposed to fire using OpenSees," *Journal of Structural Fire Engineering*, vol. 6, no. 1, pp. 1–20, 2015.

[28] J. Jiang, G. Q. Li, and A. Usmani, "Effect of bracing systems on fire-induced progressive collapse of steel structures using OpenSees," *Fire Technology*, vol. 51, no. 5, pp. 1249–1273, 2015.

[29] N. E. Khorasani, M. E. M. Garlock, and S. E. Quiel, "Modeling steel structures in OpenSees: Enhancements for fire and multi-hazard probabilistic analyses," *Computers and Structures*, vol. 157, pp. 218–231, 2015.

[30] J. Jiang, L. Jiang, P. Kotsovinos, J. Zhang, A. Usmani, F. McKenna, and G.-Q. Li, "OpenSees software architecture for the analysis of structures in fire," *Journal of Computing in Civil Engineering*, vol. 29, no. 1, 2015.

[31] "OpenSees for Fire - Source Code Download," 2016. [Online]. Available: http://openseesforfire.github.io/download.html. [Accessed: 05-Sep-2017].

[32] M. L. Janssens, *An Introduction to Mathematical Fire Modeling*, 2nd ed. CRC Press, 2000.

Chapter 2.

A Critical Review of "Travelling Fire" Scenarios for Performance-Based Structural Engineering

2.1 INTRODUCTION

The "travelling fire" methodology originating at the University of Edinburgh in 2007, due to Rein *et al.* [1], postulates that fires may burn locally and move across the entire floor plate over a period of time in large compartments. It was proposed on the basis of observed fire dynamics from real fires and a few experimental programmes that have occurred over the past two decades, such as [2–6].

In real life, travelling fires have been observed in several structural failures especially since 2000: the World Trade Center Towers [7] in New York City in 2001, the Windsor Tower [8] in Madrid in 2005, and the Faculty of TU Delft Architecture building [9] in Netherlands in 2008. Looking closely at an example of an open-plan modern building, i.e. the Informatics Forum that opened at the University of Edinburgh in 2009, a statistical survey indicated that traditional fire safety design methods were applicable to only 8% of the total volume of the building (other areas being out-of-range by Eurocode limitations, e.g. opening factor (>0.2), compartment height (>4m), size of the compartment (>500m^2) [10]). These facts underline the need for a better description of fire scenarios that recognise the radically different spatial layouts preferred in contemporary architecture. There is currently greatly increased interest in methodologies for representation of more realistic fire scenarios for the purposes of fire safety engineering design.

In 2012, a review paper was published by Stern-Gottfried & Rein [11]. It summarized several fire tests conducted in the large compartments (e.g. [3–5]) as experimental evidence which clearly showed the temperature heterogeneity in such compartments. There have been three further large scale travelling fire tests performed from 2011 to 2015. In 2011, to investigate how the travelling fires impact the steel structural components especially for beam-to-column connections, a full-scale travelling fire test was conducted at the upper floor of a two-storey steel composite building in Veselí, in the Czech Republic [12]. In 2013 a series of experiments were conducted at the Building Research Establishment (BRE) in UK as part of the EPSRC funded research project 'Real Fires for Safety Design of Tall Buildings' [13]. The project intended to obtain a better understanding how a fire progresses in a large compartment and affects the temperature distribution spatially and temporally. In 2015, another experiment called the Tisova Fire

Test [14, 15] was conducted in the Czech Republic inside a 4-storey concrete frame building, in order to test the travelling fire methodology put forward by Stern-Gottfried & Rein [16].

Moreover, two main theoretical representations of travelling fire models can be found in the current literature, hereinafter referred to as: Clifton's model [17]; and Rein's model [11, 16]. Clifton developed a fire model, which divides the whole large compartment into several design areas, which are then subjected to time-temperature curves individually and sequentially. In Rein's model, Alpert's correlation is adopted to calculate far field smoke temperature, and a uniform temperature (800°C-1200°C) is assumed for the near field.

This chapter is divided into two sections: firstly, several large-scale fire experiments are reviewed, especially the ones labelled as travelling fire tests; secondly, a literature review of the current analytical travelling fire models is summarized.

2.2 EXPERIMENTS CONDUCTED FOR CHARACTERISING TRAVELLING FIRES

This section reviews the experiments that fires in which a 'travelling' nature in large compartments, with a particular emphasis on the ones labelled as travelling fire tests conducted for the past five years.

2.2.1 Fire Tests of a 'Travelling' Nature Before 2010

Although true dynamics of travelling fires has received "zero attention" in large scale structural fire tests [18] (as summarized by Bisby *et al.* in 2013), there are still some experiments where a 'travelling' nature of the fire is recorded in the literature.

In 1993, to validate the 'Time Equivalent' formula given in Eurocode 1 for buildings with large compartments, a series of nine tests were carried out at BRE Cardington laboratory [3]. The dimensions of the test compartment were 22.8m long × 5.6m wide × 2.75m high (128m^2 floor area) with uniform wood cribs as the fuel load, and the

ventilation was at one end of the long compartment. The fuel was ignited at the opposite end to the ventilation (apart from Test 9, which was ignited simultaneously for comparison), and it was observed that the fire spread quickly to the ventilation side, consumed all the fuel near the vent region, and then the fire travelled back to the ignition region and burned out. Both the gas temperatures and steel temperatures of the protected and unprotected steel members were recorded for the entire duration. Cooke [19] took additional measurements including thermal radiation, gas analysis, air flow, and crib weight loss in the experiment.

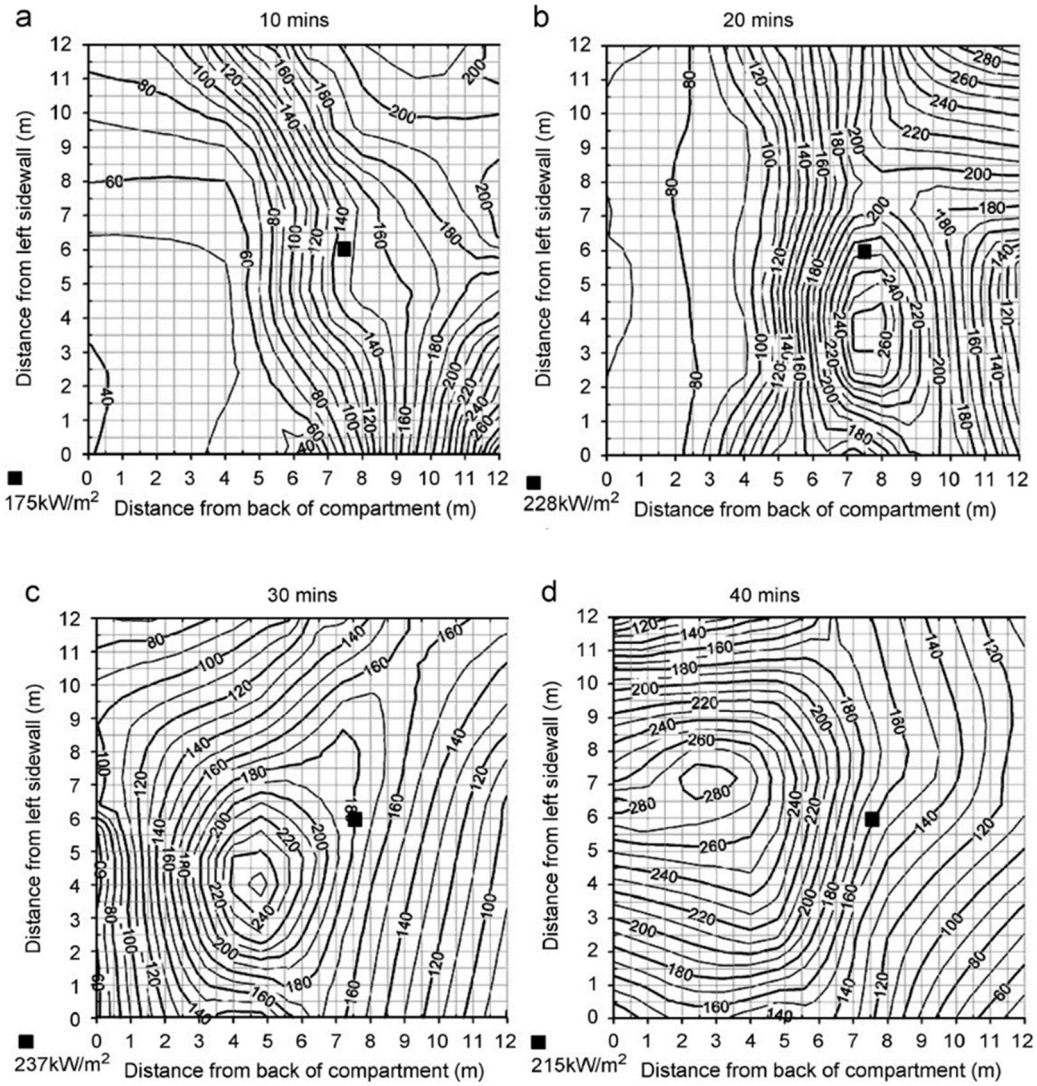


Figure 2.1. Heat flux map under the compartment ceiling, reprinted from Welch *et al.* [6] with permission from Elsevier.

In 1995-1996, an experimental testing programme took place on an eight-storey steel-framed structure, at BRE Cardington Large Building Test Facility (LBTF). This research programme contains four tests, in which the fourth one - Demonstration Furniture Test - was to investigate the impact of a more realistic fire scenario to the whole structure [20]. The test compartment was 18m wide and up to 10m deep (135m^2 floor area), to represent an open plan office with modern day furnishings, computers and filing systems, which are equivalent to the fuel load density of 45.6kg of wood/m^2. Both the ignition method and the ventilation conditions were designed to assist the fire growth, which generated non-uniform (migrating) fire scenarios during the test [21]. The gas temperatures, beam and column temperatures, and the connection temperatures were all measured. Moreover, the structural response was also recorded, including the strain along the columns, the deflections of the beams and floor slabs. All these test data can be found at the One Stop Shop web site, which is maintained by the University of Manchester [22].

In 1999-2000, a series of eight large compartment fire tests were undertaken at BRE Cardington LBTF, to validate the zone models as part of the Natural Fire Safety Concept (NFSC) framework. These eight tests were full-scale post-flashover fires conducted in a large compartment with approximate dimensions 12m × 12m × 3m high (144m^2 floor area), with different opening situations, fire load compositions (wood cribs only, or 80% wood cribs + 20% plastic), and the compartment boundary linings [23]. Thermocouples were distributed throughout the compartment for recording gas temperatures, and the steel temperatures were measured for both the structural components with and without protections. Mass loss was also recorded through the tests by using load cells. The spatial and temporal change of the heat flux fields under the ceiling were produced by Welch *et al.* [6] in Figure 2.1. The maximum recorded temperature was over 1330°C.

In 2005, a series of eight experiments were conducted by Thomas *et al.* [24] for investigating the fire behaviour in a deep enclosure with various openings in one end. The steel enclosure for the tests has dimensions of 8.0m long × 2.0m wide × 0.6m high (16m^2 in area), with sixteen steel fuel trays containing 97% ethanol (see Figure 2.2). Only the front end of the enclosure was ventilated with different opening sizes for the eight tests. Both gas temperatures and steel temperatures were recorded during the test (maximum thermocouple temperature was around 850°C) (see Figure 2.3), and a

calorimeter hood was used for collecting the outgoing combustion products to estimate the heat release rate. A load cell was placed beneath each tray to record the mass loss of the fuel throughout the test.

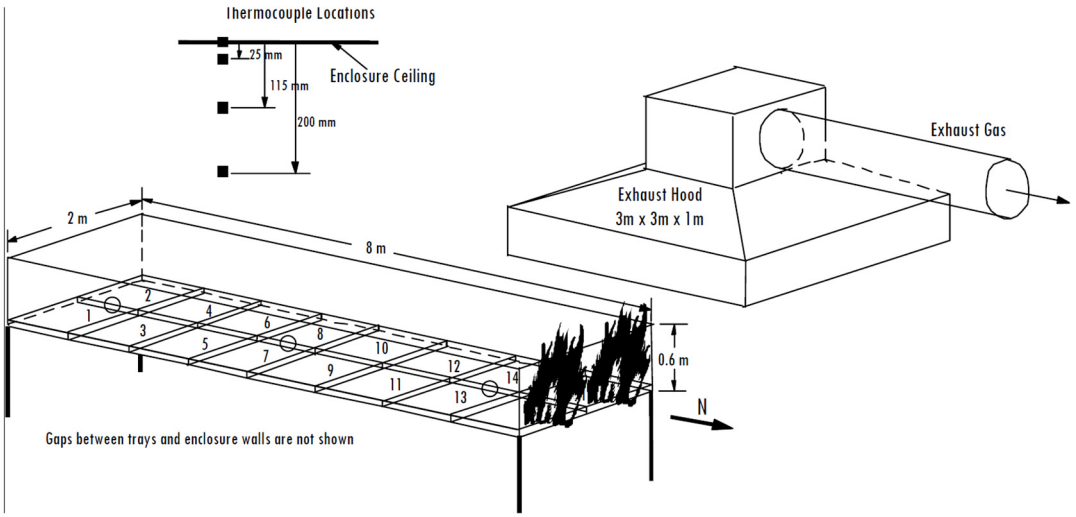


Figure 2.2. Schematic diagram of experimental test configuration, Thomas *et al.* [24].

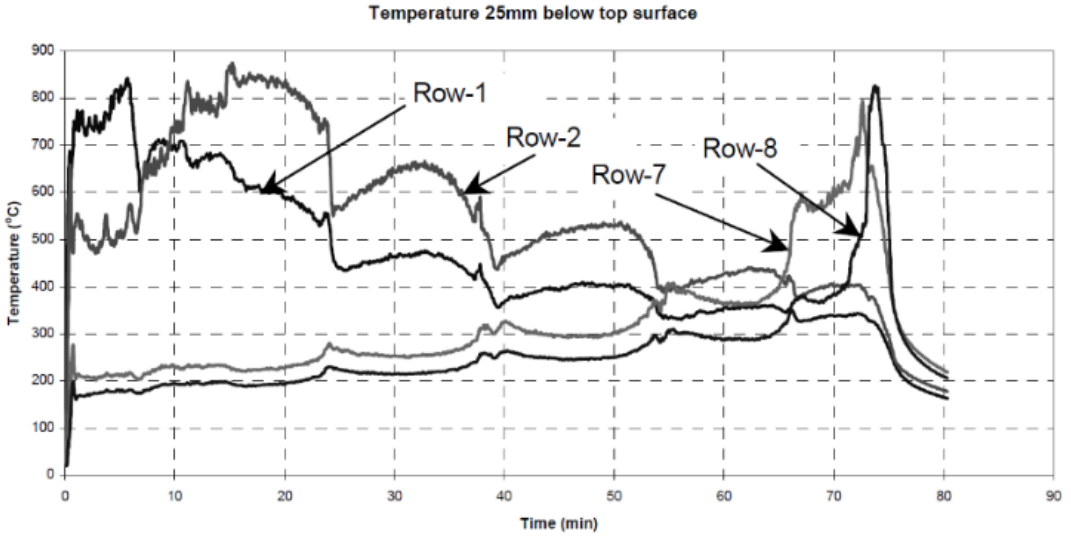


Figure 2.3. Gas time-temperature curves above the front and back rows, Thomas *et al.* [24].

In three of the four sets of tests mentioned above: Kirby [3] & Cooke [19] in 1993, NFSC - BRE Cardington [6, 23] in 1999-2000, and Thomas *et al.* [24] in 2005, all showed similar ventilation controlled fire dynamics in large compartments. In all three cases, the fire was

ignited away from the ventilation area, rapidly spreading towards the area of abundant oxygen near the vent, exhausting the fuel near the vent, and then slowly burning back away from the vent area, consuming the majority of the available fuel.

Although more fire tests with a 'travelling' nature can be found in the literature, such as the St. Lawrence Burns project reviewed by Gales [25], the emphasis of this chapter is about the state of the art of the travelling fires, hence only typical 'spreading' fire tests are reviewed as above. The following subsections present three large scale experiments labelled as *travelling fire tests*.

2.2.2 Veselí Travelling Fire Test (Czech Republic, 2011)

Figure 2.4. Experimental building during Veselí travelling fire test
(photo provided by Horová K., CVUT in Prague).

This test was part of a European-funded project called COMPFIRE [26], investigating the behaviour and robustness of the practical beam-to-column connections under travelling fire scenarios. The experimental building was a 10.4m × 13.4m in plane × 9m in height (139m^2 floor area) two-storey steel composite structure, with a 2m × 5m unglazed opening for each floor to provide enough ventilation for a smooth development of the fire (see Figure 2.4).

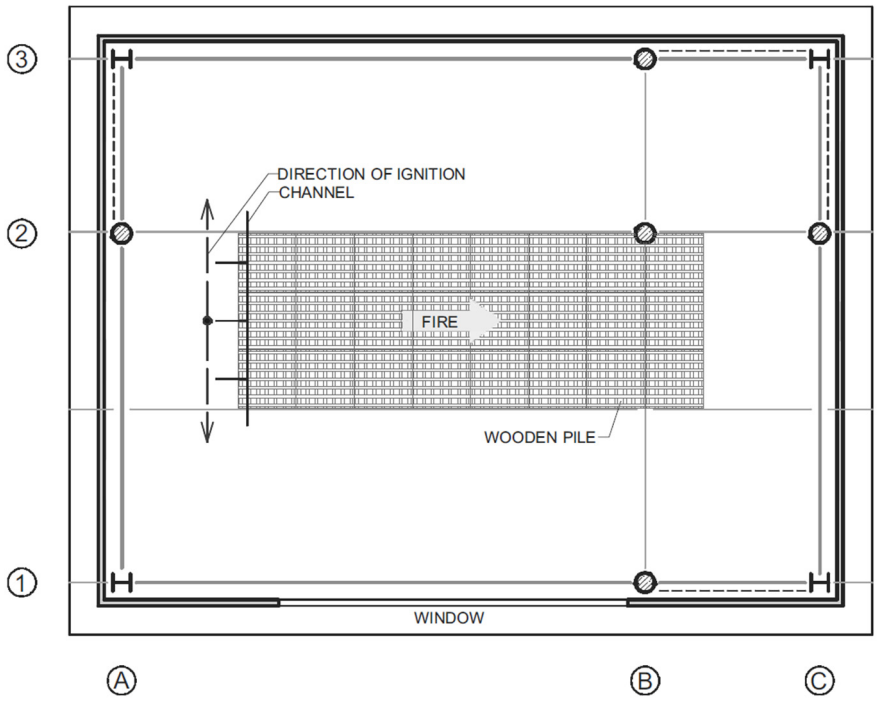


Figure 2.5. Fuel load scheme, in hatched, on the upper floor of the Veselí travelling fire experimental building, Wald *et al.* [27].

The height of each floor was 4m. The fuel load was wood cribs dried to moisture content of 12%, with density 173.5 MJ/m^2, distributed on the second floor, with a 3m × 8m rectangular shape as shown in Figure 2.5, with the desired fire path parallel to the ventilation opening rather than perpendicular to it. In addition, no mechanical load was applied during the entire travelling fire test [26].

Figure 2.6 shows the fire development with a time step of every 5 min during the 40 min test duration. The fire was first ignited with a linear source on the left-hand side, then the flame spread gradually to the right, accompanied by a smoke layer generation beneath the ceiling for the beginning 15 minutes. Then more fuel was on fire with a maximum gas temperature recorded of 979°C at 26 min. From 30-40 min, the process of the fire burn out can be clearly seen. Furthermore, neither flashover nor structural failure was observed during the test [26, 28].

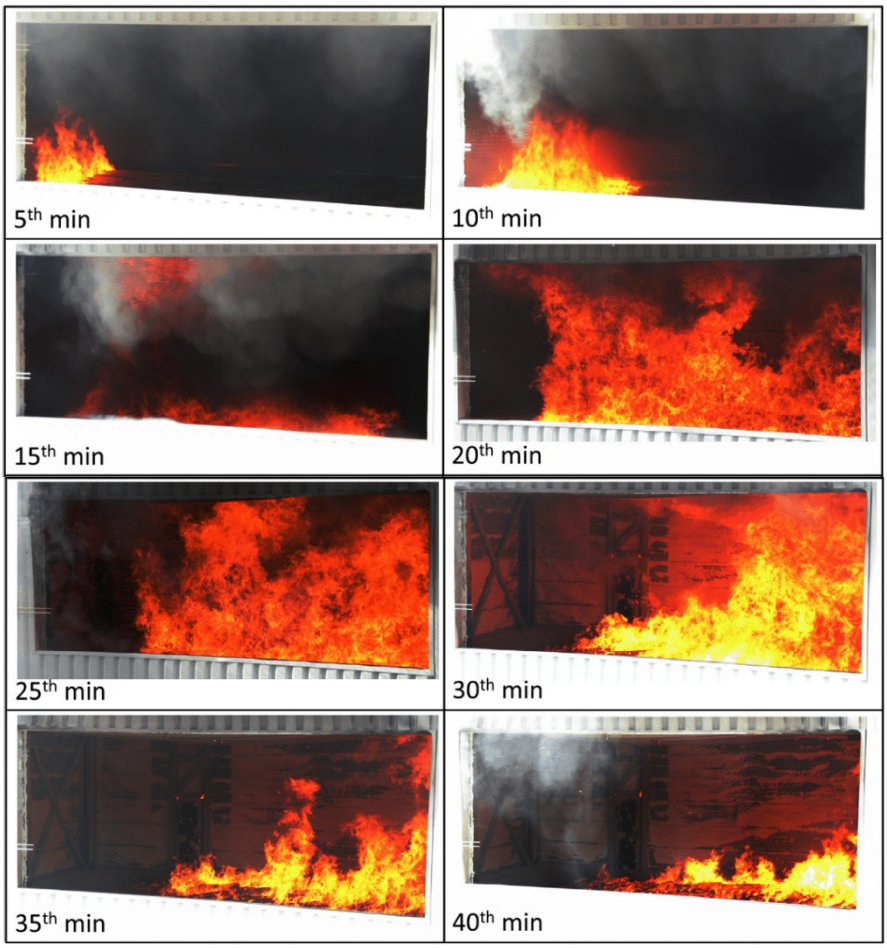


Figure 2.6. Development of the fire during the Veselí travelling fire test (photos provided by Horová K., CVUT in Prague).

Importantly the temperatures of the gas atmosphere, steel beam at mid-span, connections, composite slabs, and columns were all measured. Unlike many similar tests, the structural response was also extensively recorded, including the vertical and horizontal displacement of the slab, the deflection of the beam mid-span, and the strain gauge on the columns for estimating the forces of the connections [27].

2.2.3 BRE Travelling Fire Test (UK, 2013)

In 2013 a series of experiments in support of the project 'Real Fires for Safe Design of Tall Buildings' [13, 29] was conducted by the University of Edinburgh at the BRE in UK. The aim of these experiments is for obtaining a better understanding of how a fire

progresses in a large compartment and affects the temperature distribution spatially and temporally.

The experimental compartment was 5m × 18m in plane × 2m in height (90m^2 in area), with 15 potential openings (1.5m high × 1m wide) along the front of the compartment. These openings were adjusted in the course of the tests to allow different ventilation progressions; one series of tests adopted sequentially ignited gas burners with different fire spread rate and ventilation combinations, and the other, wood cribs, these being ignited at one end of the compartment to allow the fire to propagate parallel to the openings. Load cells were used to measure the crib mass loss.

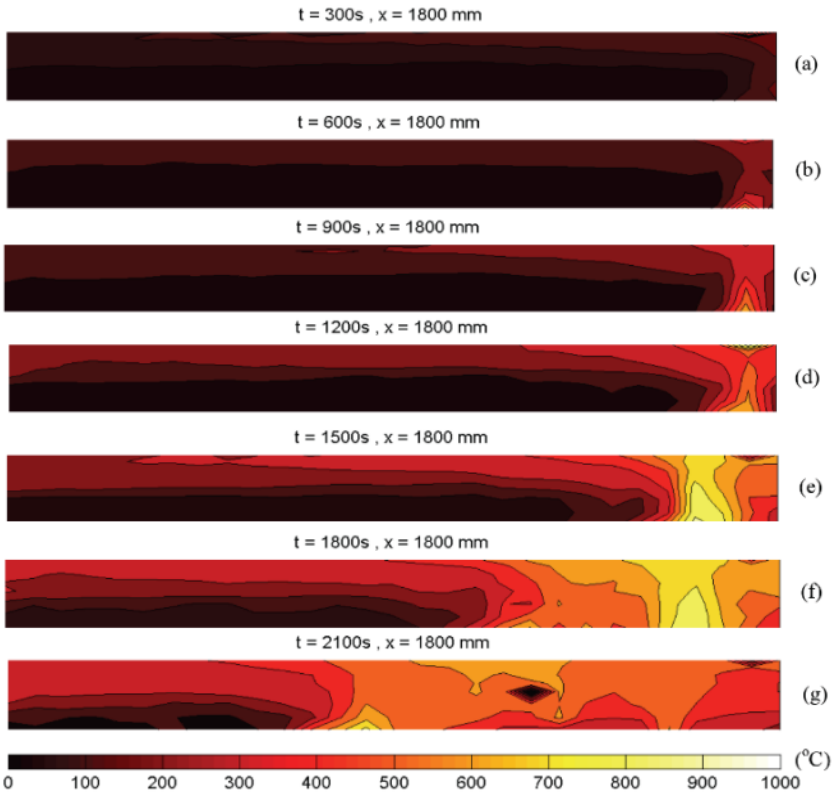


Figure 2.7. Temperature distributions along the plane of the openings, Torero *et al.* [13].

An example of the fire development in the experimental compartment can be seen in Figure 2.7, which presents the temperature distributions of the plane parallel to the compartment openings in the wood crib fire test [13]. During this test, the ventilations were fully open to allow the maximum of smoke to evacuate. The fire was ignited at the

right-hand corner of the long compartment, and it spread very slowly compared to the propagation of the smoke under the ceiling (Figure 2.7(a) to (d)). At about 1500 seconds Figure 2.7(e)), the temperature of the smoke exceeded 500°C, and a localized flashover was observed in the right half of the compartment (Figure 2.7(f)). Then the flame continued to spread to the left-hand side of the compartment, however no further flashover was observed due to the evacuation of the smoke and strong air entrainment from left side to the right side (Figure 2.7(g)).

2.2.4 Tisová Travelling Fire Test (Czech Republic, 2015)

This is the latest travelling fire test reported in the literature [14, 15, 30]. It was conducted by a team from SP, the University of Edinburgh, Imperial College London, Luleå Technical University, Technical University Ostrava, Majaczech, CSTB and CERIB, to investigate travelling fires and their impact on concrete and composite structures. The Tisová fire test structure was a four-storey concrete frame centred around a lift shaft, with the travelling fire test compartment located on the ground floor with a total area of 230m^2 in plane × 4.4m in height (see Figure 2.8 & Figure 2.9). The large test compartment was well ventilated to fit in with the idealization made by Rein in his travelling fire model, which assumes that the travelling fire is entirely fuel controlled, i.e. ventilation is not limiting [16]. The fuel load was wood cribs uniformly distributed on the whole floor with a density of approximately 680 MJ/m^2.

Figure 2.9 shows the fire ignition (FI) point, fire path, and the instrumented column C1 (30cm × 30cm) in the fire compartment. Once the fire was ignited, it spread very slowly and the measured temperature near the ceiling was below 100°C. This was apparently not as challenging a fire to the structure as intended. Therefore, the team decided to reduce the ventilation and add 10 litres of hydrocarbon accelerant to the wood cribs at time 2.5 h. This produced a more severe fire, however when the fire proceeded to the north of the compartment, the spread rate slowed down again. It was concluded that the poor severity of the fire was mainly because of the high moisture content of the wood cribs, i.e. 18-22%, rather than the targeted 11% [30].

Figure 2.8. View inside of the Tisová travelling fire experimental compartment (photo provided by Rush D., Tisova Fire Test-2015, report forthcoming).

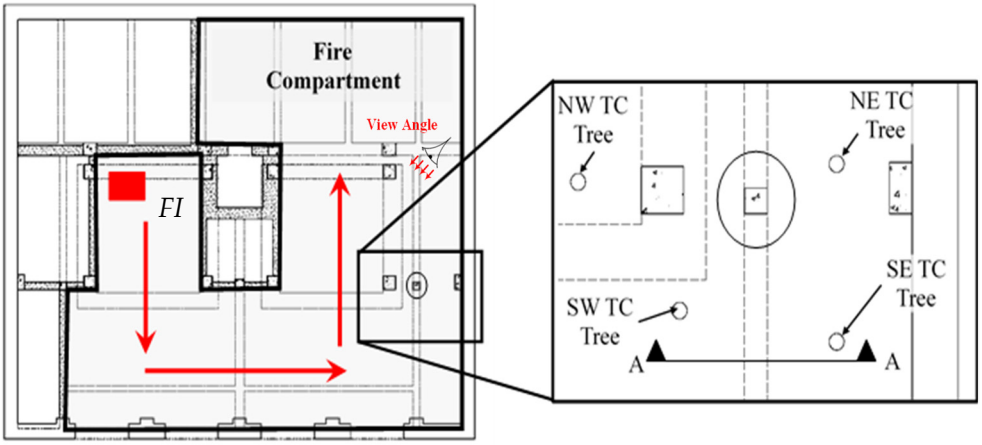


Figure 2.9. Fire path and instrumented column of the Tisová travelling fire test, reprinted from Rush *et al.* [30], with permission from DEStech Publications, Inc. (The view angle of Figure 2.8 is shown in Figure 2.9).

In this travelling fire test, gas temperature, the concrete column temperature, and the slab deflections were measured. Of particular note, it was found that the smoke preheated the top of the column C1, which was located far away from the fire ignition point. When

the fire had travelled to the vicinity of the column, it was found that the lower part of the compartment experienced higher temperatures than near the ceiling. It was also shown that the equivalent time method under the ISO-834 fire curve is not appropriate for predicting the temperature of the columns under the travelling fire scenarios, which implies that a new design method for columns in large compartment under fire may be needed in the future [30]. Analysis of the thermal and structural response of these tests is still ongoing at the University of Edinburgh and SP.

2.2.5 Summary of the Experiments

In general, these experimental reviews are focused on tests which have used 'spreading' fires, with a particular emphasis on the latest three large scale travelling fire experiments. The reviews aim at obtaining a better understanding of the travelling fire research frontier, and providing recommendations for future experimental research needs on this topic. Table 2.1 summarises the tests reviewed in the previous subsections. It is categorized with respect to the scale of the experiment, the fuel load type, the measurement of the thermal response, structural response (strain, deflections, etc.), and the mass loss of the fuel.

Table 2.1. Summaries of the experiments reviewed for travelling fires.

Experiments	**Categories**				
	Dimensions	**Fuel load**	**Thermal response**	**Structural response**	**Mass loss measurement**
Kirby [3] & Cooke [19], 1993	22.8m × 5.6m × 2.75m	Wood cribs	Gas and steel temperatures	None	Yes
LBTF – Denmonstration Furniture [20–22], 1995-1996	135m^2	Furniture	Gas and steel temperatures	Strain and deflections	No
NFSC - BRE Cardington [6, 23], 1999-2000	12m × 12m × 3m	Wood cribs only, or 80% wood cribs + 20% plastic	Gas and steel temperatures	None	Yes
Thomas *et al.* [24], 2005	8m × 2m × 0.6m	Commercial grade methylated spirits (97% ethanol)	Gas and steel temperatures	None	Yes
St. Lawrence Burns project [25], 1958	11.2m × 12.8m, and 13m × 9m	Wood waste	Gas temperatures	None	No
Veselí Travelling Fire Test [12, 26–28], 2011	10.4m × 13.4m × 4m	Wood cribs	Gas, steel, and concrete temperatures	Strain and deflections	No
BRE Travelling Fire Test [13, 29], 2013	5m × 18m × 2m	Gas burners, or wood cribs	Gas-phase temperature, incident radiant heat flux, obscuration, etc	N/A	Yes
Tisová Travelling Fire Test [14, 30], 2015	230m^2 × 4.4m	Wood cribs + hydrocarbon accelerant	Gas and concrete temperatures	Deflections	No

It is obvious in Table 2.1 that most experimental floor areas were larger than 100m^2, and wood cribs were commonly used as the fuel load. Gas and structural member temperatures were typically recorded for most cases. Of special interest is a finding that the test in which structural response was recorded did not record the mass loss rate of the fuel, and vice versa. It suggests that the researchers who conducted the measurement of the structural response (probably structural engineers), had less interest in the fire dynamics, as the mass loss of the fuel is a key factor to estimate heat release rate (HRR). Conversely, the researchers who conducted the measurement of the mass loss (probably fire engineers), took less interest in the fire impact induced structural response. This finding confirms a viewpoint expressed by Buchanan in 2008 [31], that "fire engineers and structural engineers need to talk to each other much more than they do now, and each group needs to learn as much as possible of the other discipline." This argument becomes more essential for the advancement of the topic of travelling fire research, as all the current analytical travelling fires models were developed for structural fire design, and based on simple fire dynamic assumptions and experimental observations. The details of these analytical models are reviewed below.

2.3 ANALYTICAL MODELS FOR TRAVELLING FIRES

Whilst there is still a large amount of uncertainty regarding the fire dynamics within large compartments, there have been calls to bring the travelling fire concept into structural design. Commencing about twenty years ago different theoretical modelling approaches have been developed: Clifton's model [17], and Rein's model [11, 16].

2.3.1 Clifton's Travelling Fire Model

The first methodology of representing travelling fires in large compartments was put forward by Clifton in 1996 [17]. This model divides the large compartment into several design areas (named as firecells), which are then subjected to modified parametric-fire curves individually and sequentially. In the model, ventilation to firecells, pre-heating of firecells, smoke logged, and cooling after burnout are all considered. A schematic of the model is shown in Figure 2.10.

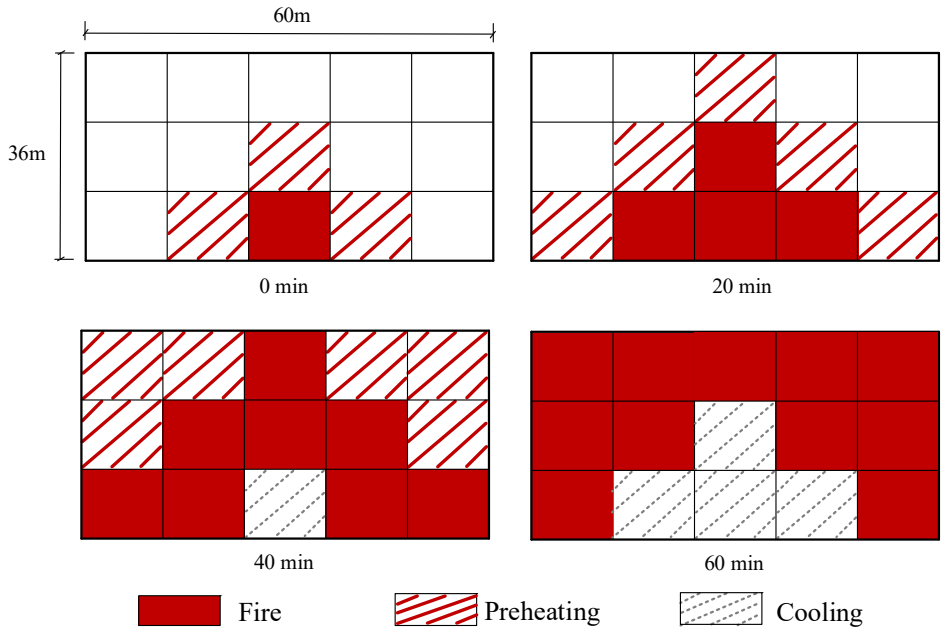


Figure 2.10. Conceptual illustration of Clifton's model, adapted from [32].

Although this pioneering model introduced aspects which are not considered in the conventional uniform burning assumption, it was not widely used in the fire safety engineering. Wang *et al.* [32] suggested reasons for the poor uptake, including lack of

documentation for the procedures to implement it, and insufficient experimental validation.

2.3.2 Rein's Travelling Fire Model

In 2007, Rein *et al.* [1] put forward an alternative travelling fire methodology, based on a series of computational fluid dynamics (CFD) analyses and engineering simplifications. It proposed a near field (fire plume near the structure) and a far field (smoke) in the model, to replace the simultaneous burning assumption used in the conventional design approach (see Figure 2.11). Figure 2.12 shows a family of far field travelling fire curves that were generated by this method with different fire sizes, using a standard fire curve and a parametric fire curve (420 MJ/m^2 fuel load density, 25% ventilation) for comparison. The family of fires is generated by covering the full range of all possible fire sizes. It is assumed in the model that each time the fire would burn a certain surface area, A_b (m^2), which is a percentage of the total floor area, A_t (m^2), ranging from 1% to 100% [33].

This model was further developed by Stern-Gottfried & Rein, and eventually put forward as a design methodology in 2012 [16]. Alpert's ceiling jet correlation [34] was adopted to calculate the far field smoke temperature (see Equation 2.1), and a uniform 1200°C was assumed for the near field:

$$T - T_\infty = \frac{5.38}{H} \cdot \left(\frac{\dot{Q}}{r}\right)^{2/3} \tag{2.1}$$

where T (°C) is the peak ceiling jet temperature, T_∞ (°C) is the ambient temperature, $\dot{Q}$ (kW) is the heat release rate of the fire plume, H (m) is the height of the compartment ceiling, r (m) is the distance from the centre of the fire plume.

The fire size within the model is governed by the available ventilation, which is usually difficult to estimate [32], hence the user is required to parametrically assess the range of structural responses to various fire sizes. Another important feature is the local near field burning time for each fire size, which is decided by the fuel load density and the heat release rate per unit area, and for a typical office building, was suggested to be 19 min by Stern-Gottfried & Rein [16]. Moreover, the fire path of the near field is not specified in the

model, as there are too many uncertainties, such as the ignition point, ventilation conditions, and fuel load distributions, which combine to preclude determination of the actual fire path in a real building [32].

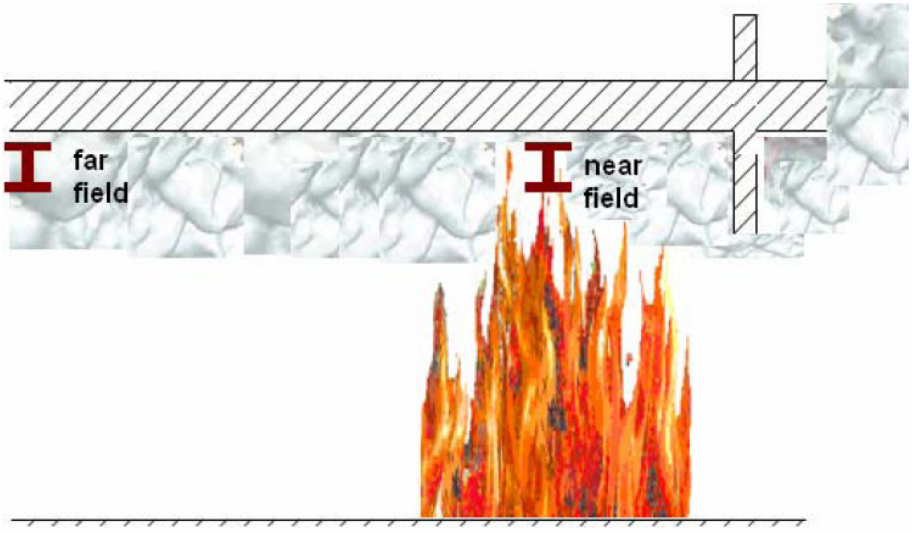


Figure 2.11. Rein's near field and far field temperature schematic [1].

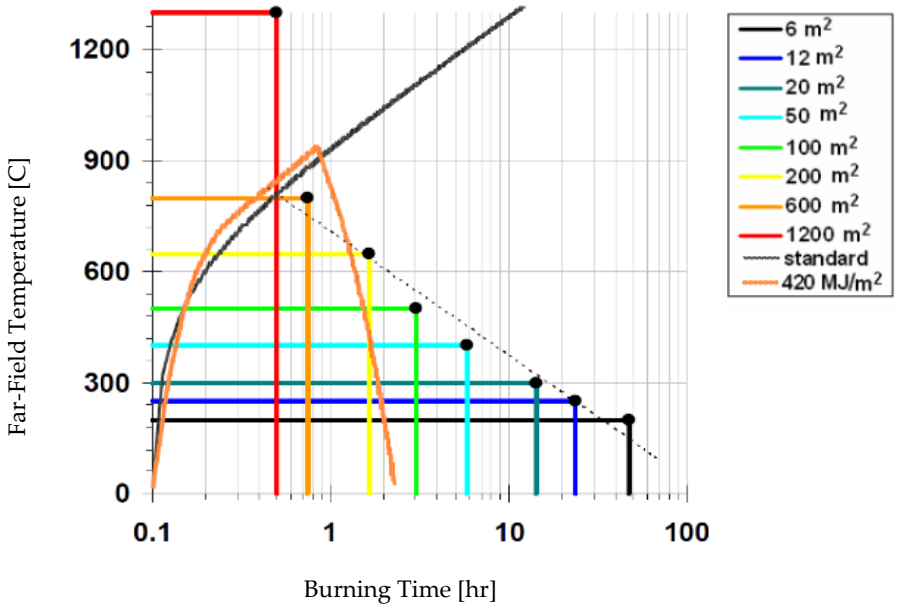


Figure 2.12. Time-temperature curves for the far-field using Rein's model [1].

More recently, Rackauskaite *et al.* [35] further improved Rein's travelling fire model by taking into account more localised fire dynamics, specifically, reducing the range of possible fire sizes which should be implemented by realistically considering fire spread rates. Furthermore, the concept of flapping angle was introduced (see Figure 2.13), to account for the near field temperature range from 800°C to 1200°C, rather than the conservative 1200°C used in the previous version. This may lower the ceiling temperatures for some fire sizes but remains a crude approximation.

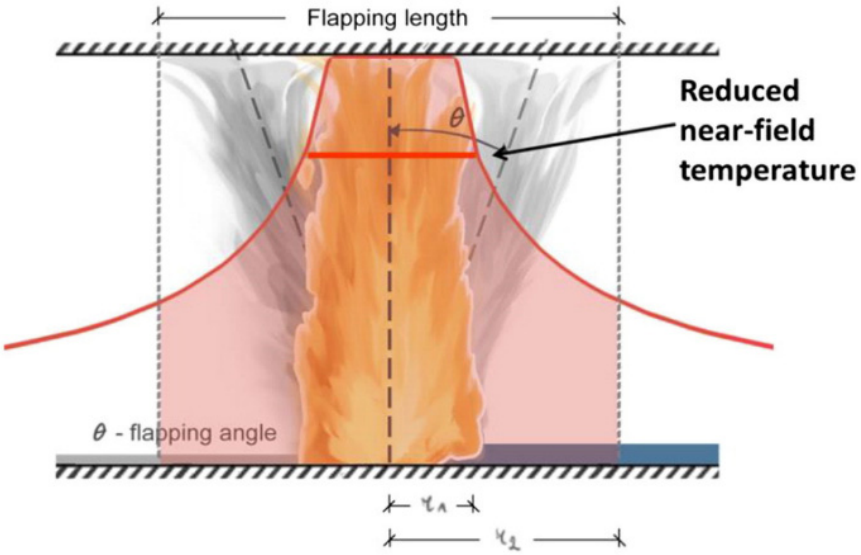


Figure 2.13. Flapping angle and reduced near field temperature in the improved Rein's travelling fire model, Rackauskaite *et al.* [35].

2.3.3 Summary of the Analytical Models

Table 2.2 summaries the above analytical models by categorizing with different model features, such as the heat release rate consideration, fire size determination, and fire path type, etc. Clifton's model, as the earliest version of travelling fire analytical model, is actually a way of applying modified parametric fire curves for series of firecells with a time lag. The fire science knowledge it involved is mainly from the utilization of these fire curves, where the fuel load density, compartment boundary conditions, and ventilations are considered. Rein's travelling fire model contains more fire dynamics, such as the considerations of heat release rate (HRR), mass conservation, and flapping angles, etc.

However, the development of these analytical models is based on simple theoretical assumptions and experimental observations, necessarily neglecting some aspects of the fire dynamics. Firstly, fire temperatures are constant for the near field in both models, i.e., the uniform 800°C-1200°C assumption in Rein's model, while all elements in one firecell share the same fire exposure history in Clifton's model. Secondly, smoke accumulation under the ceiling is ignored in both models. Thirdly and more importantly, both models have not attempted to explicitly account for the energy conservation and the mass conservation in the design compartment. Hence, the high crudeness of the current travelling models would generate, may become to oversimplified or even misleading thermal input information for the performance-based design in a large compartment.

Table 2.2. Summaries of the travelling fire analytical models.

Categories	**Models**	
	Clifton's Travelling Fire Model, [17, 32]	**Rein's Travelling Fire Model,** [1, 16, 35]
Near field temperature	Time-temperature curve for firecells	800°C -1200°C, flapping angle
Far field temperature (i.e. smoke)		Alpert's ceiling jet model
Smoke accumulation	No	No
Fire path	Firecell to neighboured ones	Not defined
Spread rate	From observation [35]	From observation [35]
Non-uniform fuel	No	No
Fire size	Decided by fuel load density	Decided by fuel & fire spread
HRR consideration	No	Yes
Mass conservation	No	Yes
Energy conservation	No	No
Compartment boundary	Yes	No
Ventilation/fuel controlled	Ventilation controlled	Fuel controlled

2.4 CONCLUSIONS

This chapter has examined experiments conducted for characterizing travelling fires, in conjunction with a review of the current analytical travelling fire models. It is apparent that travelling fire research is still at an early phase of development, and the main limitation to progress is the lack of detailed measurements of required parameters in realistic large-scale tests. The role of experiments in the analytical development is providing sufficiently general data and characterising likely worst credible conditions, to facilitate researchers in developing and validating their models, although in reality large uncertainties will remain. However, it is worth noting that travelling fire models generally don't consider the conditions in which a travelling fire may develop. They are implemented in the analysis by forcing the development of a fire moving across the floor area. Therefore, more experiments are needed to characterise these conditions in more generalised scenarios.

More importantly, an extended travelling fire methodology is needed, which is developed based on first principles considering both the mass and energy balance in the design compartment. Correspondingly, design of appropriate tests can be effectively informed by the modelling studies, and requires close collaboration between structural and fire engineers' teams. Their value will be in providing better insights into fire behaviour in realistic travelling fire scenarios, which will ultimately provide a robust methodology for performance-based structural fire engineering.

2.5 REFERENCES

[1] G. Rein, X. Zhang, P. Williams, B. Hume, A. Heise, A. Jowsey, B. Lane, and J. L. Torero, "Multi-storey fire analysis for high-rise buildings," in *Proceedings of the 11th International Interflam Conference, London, UK*, 2007, pp. 605–616.

[2] J. Stern-Gottfried, G. Rein, L. A. Bisby, and J. L. Torero, "Experimental review of the homogeneous temperature assumption in post-flashover compartment fires," *Fire Safety Journal*, vol. 45, no. 4, pp. 249–261, 2010.

[3] B. R. Kirby, D. E. Wainman, L. N. Tomlinson, T. R. Kay, and B. N. Peacock, "Natural Fires in Large Scale Compartments," British Steel Technical, Fire Research Station Collaborative Project Report, Rotherham, UK, 1994.

[4] I. R. Thomas and I. D. Bennetts, "Fires in enclosures with single ventilation openings - Comparison of long and wide enclosures," in *Fire Safety Science Proceedings of the Sixth International Symposium*, 2000, pp. 941–952.

[5] G. Rein, C. Abecassis-Empis, and R. Carvel, "The Dalmarnock Fire Tests: Experiments and Modelling," the School of Engineering and Electronics, University of Edinburgh, Edinburgh, UK, 2007.

[6] S. Welch, A. Jowsey, S. Deeny, R. Morgan, and J. L. Torero, "BRE large compartment fire tests-Characterising post-flashover fires for model validation," *Fire Safety Journal*, vol. 42, no. 8, pp. 548–567, 2007.

[7] R. G. Gann, A. Hamins, K. McGrattan, H. E. Nelson, T. J. Ohlemiller, K. R. Prasad, and W. M. Pitts, "Reconstruction of the fires and thermal environment in World Trade Center buildings 1, 2, and 7," *Fire Technology*, vol. 49, pp. 679–707, 2013.

[8] I. Fletcher, A. Borg, N. Hitchen, and S. Welch, "Performance of concrete in fire: A review of the state of the art, with a case study of the Windsor Tower fire," in *4th International Workshop in Structures in Fire*, pp. 779–790.

[9] D. M. Zannoni, J. G. H. Bos, D. K. E. Engel, and P. dr. U. Rosenthal, "Brand bij Bouwkunde," COT Instituut voor Veiligheids - en Crisismanagement, 2008.

[10] A. M. Jonsdottir and G. Rein, "Out of range," *Fire Risk Management*, pp. 14–17, 2009.

[11] J. Stern-Gottfried and G. Rein, "Travelling fires for structural design–Part I: Literature review," *Fire Safety Journal*, vol. 54, pp. 74–85, 2012.

[12] K. Horová, T. Jána, and F. Wald, "Temperature heterogeneity during travelling fire on experimental building," *Advances in Engineering Software*, vol. 62–63, pp. 119–130, 2013.

[13] J. L. Torero, A. H. Majdalani, C. Abecassis Empis, and A. Cowlard, "Revisiting the compartment fire," in *11th International Symposium on Fire Safety Science*, 2014, pp. 28–45.

[14] J. Degler, A. Eliasson, J. Anderson, D. Lange, and D. Rush, "A-priori modelling of the Tisova Fire Test as input to the experimental work," in *The First International Conference on Structural Safety under Fire & Blast,Glasgow, Scotland, UK*, 2015, pp. 429–438.

[15] D. Rush and D. Lange, "Towards a fragility assessment of a concrete column exposed to a real fire – Tisova Fire Test," *Engineering Structures*, vol. 150, pp. 537–549, 2017.

[16] J. Stern-Gottfried and G. Rein, "Travelling fires for structural design-Part II: Design

methodology," *Fire Safety Journal*, vol. 54, pp. 96–112, 2012.

[17] C. G. Clifton, "Fire Models for Large Firecells. HERA Report R4-83," HERA publications, New Zealand, 1996.

[18] L. Bisby, J. Gales, and C. Maluk, "A contemporary review of large-scale non-standard structural fire testing," *Fire Science Reviews*, vol. 2, no. 1, pp. 1–27, 2013.

[19] G. Cooke, "Tests to Determine the Behaviour of Fully Developed Natural Fires in a Large Compartment," IHS BRE Press, 1998.

[20] British Steel, "The Behaviour of a Multi-Storey Steel Framed Building Subjected to Fire Attack," *British Steel plc*. p. 332, 1998.

[21] P. J. Moss and G. C. Clifton, "Modelling of the cardington LBTF steel frame building fire tests," *Fire and Materials*, vol. 28, no. 2–4, pp. 177–198, 2004.

[22] University of Manchester, "One Stop Shop in Structural Fire Engineering." [Online]. Available: www.structuralfiresafety.org. [Accessed: 14-Jul-2017].

[23] T. Lennon and D. Moore, "The natural fire safety concept - full-scale tests at Cardington," *Fire Safety Journal*, vol. 38, no. 7, pp. 623–643, 2003.

[24] I. Thomas, K. Moinuddin, and I. Bennetts, "Fire development in a deep enclosure," in *Fire Safety Science - Proceedings of the 8th International Symposium*, 2005, vol. 8, pp. 1277–1288.

[25] J. Gales, "Travelling fires and the St. Lawrence Burns project," *Fire Technology*, vol. 50, no. 6, pp. 1535–1543, 2014.

[26] L. Simões da Silva et al., "Design of Composite Joints for Improved Fire Robustness," RFCS Compfire project-final report (RFSR-CT-2009-00021), European Commission, Luxembourg, 2014.

[27] F. Wald, T. Jána, and K. Horová, "Design of Joints to Composite Columns for Improved Fire Robustness: To Demonstration Fire Tests," Česká technika - nakladatelství ČVUT, 2011.

[28] K. Horová, "Modelling of Fire Spread in Structural Fire Engineering," PhD Thesis, CVUT Prague, 2015.

[29] J. P. Hidalgo, A. Cowlard, C. Abecassis-Empis, C. Maluk, A. H. Majdalani, S. Kahrmann, R. Hilditch, M. Krajcovic, and J. L. Torero, "An experimental study of full-scale open floor plan enclosure fires," *Fire Safety Journal*, vol. 89, pp. 22–40, 2017.

[30] D. Rush, D. Lange, J. Maclean, and E. Rackauskaite, "Modelling the thermal and structural performance of a concrete column exposed to a travelling fire – Tisova Fire Test," in *the Proceedings of the 9th International Conference on Structures in Fire*, 2016, pp. 110–118..

[31] A. Buchanan, "The challenges of predicting structural performance in fires," in *Fire Safety Science – Proceedings of the Ninth International Symposium*, 2008, pp. 79–90.

[32] Y. Wang, I. Burgess, F. Wald, and M. Gillie, *Preformance-Based Fire Engineering of Structures*. CRC Press, 2012.

[33] J. Stern-Gottfried, G. Rein, B. Lane, and J. Torero, "An innovative approach to design fires for structural analysis of non-conventional buildings: A case study," in *the Proceedings of International Conference Applications of Structural Fire Engineering*, 2009,

pp. 34–40.

[34] R. L. Alpert, "Calculation of response time of ceiling-mounted fire detectors," *Fire Technology*, vol. 8, no. 3, pp. 181–195, 1972.

[35] E. Rackauskaite, C. Hamel, A. Law, and G. Rein, "Improved formulation of travelling fires and application to concrete and steel structures," *Structures*, vol. 3, pp. 250–260, 2015.

Chapter 3.

An overview of the development of OpenSees software framework for modelling structures in fire

3.1 INTRODUCTION

The Open System for Earthquake Engineering Simulation (OpenSees) is a C++ object-oriented[1], open source software framework developed at the University of California, Berkeley by McKenna in the late 1990s [1]. It was originally for providing an advanced finite-element simulation tool to perform structural and geotechnical analysis under seismic loadings. It has now become a common platform, for researchers within the Pacific Earthquake Engineering Research Center (PEER, which is a multi-institutional research and education centre in US), for the development, sharing, and dissemination of new ideas to further earthquake engineering research around the world. Three key references including the OpenSees user manual [2], OpenSees main website [3], and McKenna's PhD thesis [1], have reached citation numbers to 764, 898, and 330 respectively, according to the latest records from Google Scholar in September 2017.

However, in the research community of structural fire engineering, the available software are normally lacking in code transparency, having limited modelling capabilities, but with high purchase expenses. In general, there are two types of computer programs for simulating structural behaviours in fire: commercially-oriented and research-oriented. Commercial software packages (e.g. ABAQUS, ANSYS and DIANA) normally offer advantages such as extensive verifications & validations, user-friendly interfaces for pre-processing & post-processing, and professional support for maintenance. Nevertheless, there are inherent limitations. For instance, development of commercial codes is often dictated by the requirements of the most profitable applications and rarely addresses the needs of academic researchers, thus developments typically lag well behind research. An alternative is to use proprietary software developed by researchers. Well known examples of such codes in the structural fire engineering community are: SAFIR [4] from University of Liège, Vulcan [5] from University of Sheffield, ADAPTIC [6] from Imperial College London, each for analysis of structures

[1] Object-oriented programming (OOP), is a programming language paradigm designed to focus on manipulating 'objects' which are bundled with data and algorithms. OOP is ideal for developing large-scale software framework (popular languages such as C++, Java, and Ruby). It allows developers much easier to reuse the developed codes, and to secure the data without corruption.

subjected to fire (and earthquakes in the case of ADAPTIC). These codes typically suffer from tightly bound architecture as a result of using procedural programming [2]. Furthermore, because they are often developed by a small dedicated team of researchers at the original host institution, they are not designed or suited for a devolved community of developers, and the codes are not open source. They also typically have uncertain resourcing and great dependency on key individuals for support, maintenance and development. Hence, they cannot generally be considered sustainable in the long term.

The limitations discussed above can in principle be overcome by the development of open source software using objected-oriented programming (OOP), in which OpenSees is an ideal option. Due to the above reasons and inspired by the success of OpenSees in its own research community, Usmani *et al.* [7] at the University of Edinburgh initiated to adapt OpenSees for analysing structural response subjected to fire in 2008. After nearly ten years of development, OpenSees now has modules which can perform fire modelling [8, 9], heat transfer analysis [8, 9], and thermo-mechanical analysis for large structures under realistic fire scenarios [9–12]. The research publication outcome contains 5 PhD theses [8–12], and 11 journal papers [7, 13–22]. According to Usmani's latest Google Scholar web page in September 2017, 'OpenSees in Fire' (OiF) related papers have reached 138 citations in total [23].

In addition, OiF as an open source software tool for research (source codes can be downloaded through [24]), can promote a sense of community and facilitate greater collaboration between research groups with similar interests irrespective of geographical location. For instance, the implementation of steel material at high temperature was initially developed by the research group at the University of Edinburgh. Due to OiF's free access nature, this steel material model was used by the research group from Princeton University for post-earthquake fire probabilistic analysis. It was found that the capabilities of original steel material model for capturing cooling phase was not accurate enough [20], hence the model was fixed by the Princeton group and sent back to the

[2] By contrast, procedural programming is another programming language paradigm designed to focus on procedures which gather input, process input data with series of functions, and deliver output. It offers ease and transparency for less complicated applications (popular languages such as C, Fortran, and Pascal).

research group at the University of Edinburgh. Afterwards the codes were updated in the public shared repository as mentioned above in [24]. Moreover, OOP using C++ permits OiF to fit in the research environment better. This is because it permits researchers to quickly view and gain the understandings of the workings of the codes, based on their own research interests on the specific module, instead of going through all the procedures and functions using procedural programming (e.g. Fortran). It would further allow the researchers more likely to become developers, who would lead the way in defining priorities and contribute to development of the software tool (i.e. OiF) on their own research benefit, rather than predominantly being 'users' as the current situation in the structural fire engineering research community.

Nevertheless, developing and maintaining a software framework is not regarded as an original scientific contribution to the research community. It is found that the research outcome of thousands of lines of codes for OiF, are 'scattered' in different conference papers, journal papers, internal reports, and PhD theses, without being systematically summarised and published in the sense of software development, including complete OiF's capabilities with fire modelling, heat transfer analysis, and thermo-mechanical analysis. Moreover, the need for such tools in the structural fire engineering research community is still growing [25–27]. Hence absence of a summary for the software's transparency to further identify its capabilities and limitations, would probably risk losing the potential users [28] and future developers, in which is the exact circumstances that OiF has.

This chapter aims to provide an overview of the developed OiF's modules. It summarizes the developed OiF's modelling capabilities for the past ten years of codes development, with addressing its theoretical background, fundamental assumptions, and inherent limitations. It will provide a resource, for OiF's users to cite this software with full current updated information, and for OiF's developers to redirect to the corresponding literature.

3.2 MODELLING CAPABILITIES

Adapted from original OpenSees, OiF in nature is a software framework rather than a single software, which means a developer can build an application for the users with a selected collection of modules/classes under different operating systems (i.e. Windows, or Mac OS), with corresponding integrated development environment (IDE).

As a user for OiF, Tcl [29] is used to build up the model with a text editor (e.g. Notepad++). Tcl is a high-level[3], string-based, scripting language. It is embedded into the OiF application to operate as an interpreter to process the commands from the users. In OiF, Tcl command library allows to be redefined or further extended in the C++ source codes. Another way of building models requires the user has certain level of C++ knowledge, to provide a 'main' function using the 'quickMain' module with an IDE [30]. This way is more feasible for developers to test or benchmark their new codes, which is not recommended at a user level. There is no tool for pre-processing & post-processing with a graphical interface for OiF at this stage, although there have been quite a few options available for original OpenSees, such as Build-X [31], GiD + OpenSees [32], and STKO [33]. Hence the user has to use the Tcl commands to script specific requests, for the interested thermal/structural response (e.g. time-temperature histories over the structural member cross section, nodal displacement, cross-sectional stress and strain, etc.).

To perform a complete set of 'structures in fire' analysis in OpenSees, three modules are developed: the fire module, heat transfer module, and structural module. The relationship among these three modules is illustrated in Figure 3.1. The fire module is to provide thermal boundary conditions for structural elements to carry out heat transfer analysis. Depending on the complexity of the requested fire scenarios, fire module in OpenSees can either utilise simple fire model libraries (e.g. standard fire, parametric fire, etc. [34]), or more realistic fire scenarios (i.e. localised fire). The interface between the fire module and heat transfer module is called at each instant of time during the heat transfer analysis, to pass the fire information as transient thermal boundary conditions (i.e. temperature, or heat flux). After the heat transfer analysis, the time-temperature histories

[3] A high-level language is a programming language which contains more abstractions and closer to human languages [48] (e.g. C, C++, Fortran, Pascal, Tcl, etc). In contrast, A low-level language contains little abstractions and closer to machine languages [49].

of the heat transfer nodes are recorded. Due to the mesh difference between a heat transfer analysis and a subsequent thermo-mechanical analysis in OpenSees, these time-temperature histories from the heat transfer nodes are interpolated at a number of required data points over the cross section for the structural element, according to different cross section types and element types [8]. For instance, the number of required data points with time-temperature histories over the cross section for thermo-mechanical analysis can be as many as 15, if a 3D I-beam model is to be built up.

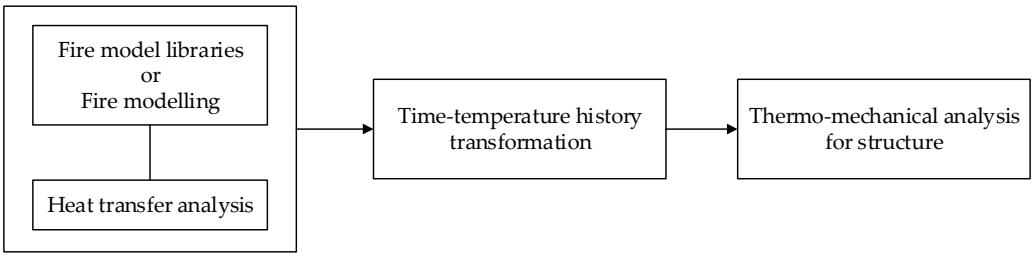


Figure 3.1. Modelling scheme of the fire module, heat transfer module, and structural module in OpenSees.

In general, OiF follows a traditional 'structures in fire' one-way coupling strategy between the heat transfer analysis and thermo-mechanical analysis. It means that there is no feedback from the thermo-mechanical analysis to the heat transfer analysis [35]. This coupling strategy is valid in most 'structures in fire' modelling situations, especially for estimating generic large structural behaviours under realistic fires, as local modelling details are not required. However, it is worth to note that the modelling scenarios including concrete spalling, fire induced large structural deformations which change the compartment dimensions significantly, and separation of insulation material of the protected structural members under fire, are typical examples that OiF is not able to handle so far.

3.2.1 Fire Module – Determining Fire Imposed Boundary Conditions

Fire module in OpenSees is to determine the thermal boundary conditions for the subsequent heat transfer analysis of the corresponding structural members. It contains simple fire model libraries and fire modelling capabilities, depending on the complexity

and accuracy requirement of the structural fire modelling task. The selection of the fire models in OiF should follow the concept of "consistent level of crudeness" [28]. For example, using a highly simplified fire model (e.g. standard fire) as the thermal input to predict the structural response of a real building (e.g. full 3D steel composite structure) is pointless, and vice versa.

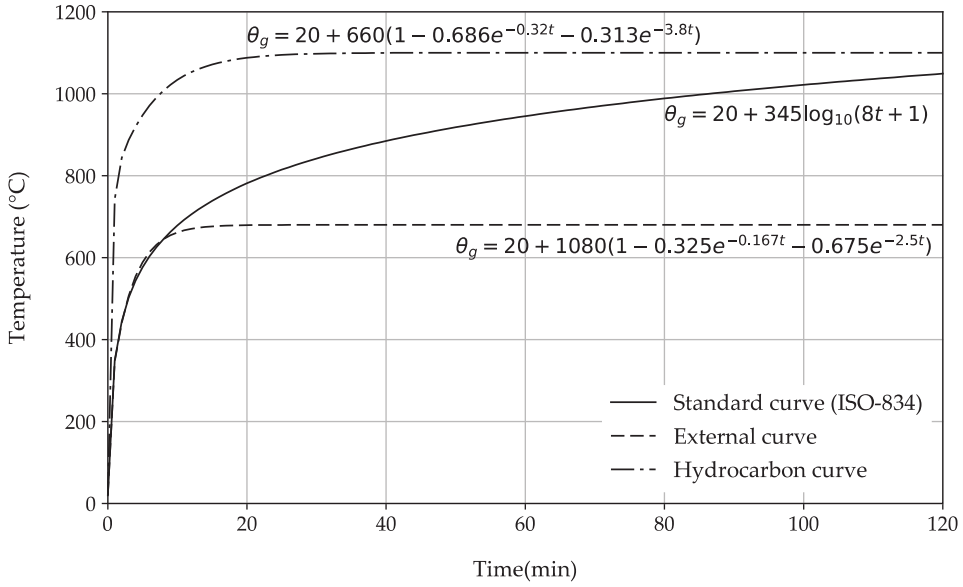


Figure 3.2. Three different nominal time-temperature curves in Eurocode.

Conventional structural fire design codes are based on isolated single structural members with simply-supported boundary conditions under standard fire test exposures, which refers to a heating curve such as ISO-834 standard fire [36], or ASTM-E119 fire [37]. The standard curve along with the external curve, and the hydrocarbon curve are categorized as the nominal curves in the Eurocode [34]. This type of curves is basically a time-temperature relationship, which stands for the case of a fully developed fire in a compartment. Figure 3.2 illustrates these nominal time-temperature curves along with their mathematical expressions, where t (min) is the time, θ_g (°C) is the gas temperature. All the above-mentioned time-temperature curves are added in the fire module of OpenSees [8], with Tcl input keywords as *Standard*, *ASTM*, *External*, and *Hydrocarbon* respectively [9].

Different from the nominal time-temperature curves, the parametric fire curves incorporate more realistic input parameters, involving fire load density and

characteristics of the compartment (e.g. thermal boundaries, ventilation conditions, geometric quantities, etc.) [34]. This type of fire curve has a heating phase and a cooling phase to represent post-flashover fires in the compartment. The application of the parametric fire model in Eurocode has limitations as follows: the compartment height (<4m), the size of the compartment (<500m^2), the opening factor (0.02m$^{1/2}$< O <0.20m$^{1/2}$), and the thermal inertia of the linings (100J/m^{2}s$^{1/2}$K< O <2200J/m^{2}s$^{1/2}$K). This model is also included in OpenSees [8] according to the equations provided in the Eurocode [34], with Tcl input keyword as *Parametric* [9].

Although these fire models are relatively simple, they are still widely used for both research and design purposes in the structural fire engineering. Further, a user defined fire curve is also included in OpenSees for providing more flexibility [8], with Tcl input keyword as *Userdefined* [9].

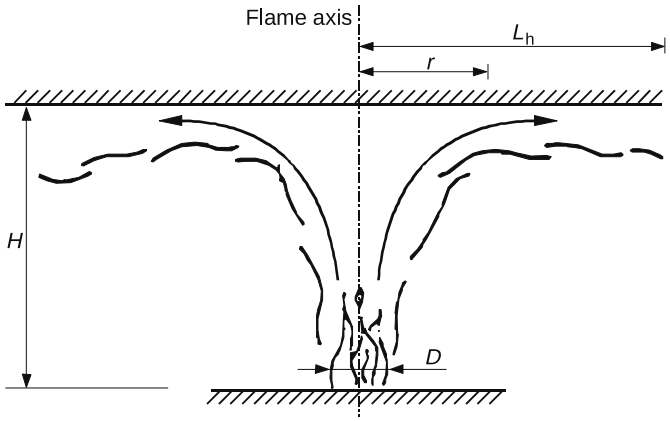


Figure 3.3. Localised fire model in Eurocode [34].

All the above-mentioned fire models are idealised uniform fires, assuming that the entire compartment has the same time-temperature histories at a specific time. However, in the case of isolated fuels burning in a large space (e.g. vehicles burning in a car park, items burning in the airport terminals, etc.), localised fire models are regarded to be appropriate for simulating such burning scenarios (see Figure 3.3). The Hasemi localised fire model [38], which is adopted in the Eurocode illustrated in Figure 3.3, provides the heat flux with input parameters: heat release rate Q (W), diameter of the fire D (m), compartment ceiling height H (m), horizontal flame length L_h (m), and radial distance from the fire origin centre to the calculation point r (m) [34]. This model is limited to be

used when the fire flame is impinging the ceiling, the fire diameter (≤10m), and heat release rate (≤50MW). Hasemi localised fire model is implemented in OpenSees [8] according to the equations provided in the Eurocode [34], with Tcl input keyword as *LocalizedFireEC1* [9]. In addition, Alpert ceiling jet model [39], SFPE handbook-based localised fire model [40], and a user defined idealised local fire model, are added in OpenSees with Tcl input keywords as *AlpertCeilingJetModel*, *LocalizedFireSFPE*, *Idealised_Local_Fire* respectively [8, 9]. These localised fire models, in their mathematical nature, are all correlational equations between the incidental heat fluxes on the structural surfaces and the radial distance from the fire source.

Table 3.1 summaries all the implemented fire models in OpenSees, by categorizing with respect to the fire model type, the fire model name, and the Tcl input keyword. Key references of the model and its implementation in OpenSees are also provided, for the ease of being consulted by the users and developers.

Table 3.1. Summaries of the fire models in OpenSees.

Fire model type	**Categories**		
	Fire model name	**Tcl input keyword**	**Key references**
Nominal time-temperature curves	ISO-834 standard curve	*Standard*	[8, 34, 36]
	ASTM-E119 curve	*ASTM*	[8, 34, 37]
	External curve	*External*	[8, 34]
	Hydrocarbon curve	*Hydrocarbon*	[8, 34]
Parametric time-temperature curves	Parametric curves	*Parametric*	[8, 34]
User-defined time-temperature curves	User-defined curve	*Userdefined*	[8]
Localised fires	Hasemi model	*LocalizedFireEC1*	[8, 34, 38]
	Alpert ceiling jet model	*AlpertCeilingJetModel*	[8, 34, 39]
	SFPE handbook-based model	*LocalizedFireSFPE*	[9, 34, 40]
	User-defined idealised model	*Idealised_Local_Fire*	[9]

3.2.2 Heat Transfer – Establishing Temperature Histories in the Structure

Heat transfer module in OpenSees is developed based on the nonlinear finite element method (NFEM). It is capable of handling 1D, 2D, and 3D heat conduction problems, with the fire-induced heat fluxes as its thermal boundary conditions [8]. The interface between the fire module and heat transfer module is called at each instant of time during the heat transfer analysis, to pass the fire information as transient thermal boundary conditions (i.e. temperature, or heat flux). The complexity of the heat fluxes is dependent on the chosen fire model type. For instance, parametric time-temperature curves generate homogeneous compartment temperatures, which would be converted to spatially uniform convective and radiative heat fluxes. However, the travelling fires would generate both spatially and temporally non-uniform heat fluxes.

A generalised expression of the fire-induced heat fluxes $\bar{q}$ in OpenSees [8][41] is given as:

$$\bar{q} = q_c + q_r + q_{pr} \tag{3.1}$$

where q_c is the convective heat flux, q_r is the radiative heat flux, and q_{pr} is the prescribed heat flux. The global governing equation of the heat transfer module in OpenSees [8][41], using the NFEM and general trapezoidal time-integration rule[4], can be given in a discrete format as:

$$C_{n+1}\dot{T}_{n+1} + K_{n+1}T_{n+1} = Q_{n+1} \tag{3.2}$$

$$\dot{T}_{n+1} = \frac{1}{\alpha\,\Delta t}\left[T_{n+1} - T_n - (1-\alpha)\,\Delta t\,\dot{T}_n\right] \tag{3.3}$$

where n, $n+1$ are two neighbouring time steps, T is the temperature vector, $\dot{T}$ is the vector of temperature to time derivative, C is the heat capacity matrix, K is the conductivity matrix, and Q is the load vector including generalised heat fluxes $\bar{q}$ which is mentioned in Equation 3.1. Further, Δt is the time size from time step n to time step $n+1$, and $0 \leq \alpha \leq 1$. Since the material conductivity K, heat capacity C, and generalised heat

[4] Trapezoidal rule is an implicit numerical method to solve ODEs for approximating integrals [50].

fluxes $\bar{q}$ are all temperature-dependant, a predictor-corrector algorithm with Newton-Raphson iteration method is employed in the heat transfer module to solve Equation 3.2. It is worth noting that apart from the Newton-Raphson solution algorithm, modified Newton-Raphson algorithm and linear algorithm are also available as alternatives in OpenSees [8].

To handle 1D, 2D, and 3D heat conduction problems, 2-noded line element [9], 4-noded & 8-noded quadrilateral elements [8], and 8-noded brick element [8] are introduced respectively, into the heat transfer module of OpenSees. However, the users are not required to specify the heat transfer element type in the Tcl input file. Instead, a 'heat transfer entity' has to be defined, which is used to construct the structural member cross section with necessary mesh seeds information. Then the heat transfer elements would be determined and generated by OpenSees automatically, based on the modelling type (i.e. 1D, 2D, or 3D) defined by the user beforehand [9]. Table 3.2 is a summary of the heat conduction dimensional type, heat transfer element type, and the available heat transfer entity choices with their Tcl input keywords.

Table 3.2. Summaries of the heat transfer elements in OpenSees.

Heat conduction dimensional type	**Heat transfer element**	**Heat transfer entity Tcl input keyword**	**Key references**
1D	2-noded line element	*Line1D*	[9]
2D	4-noded quadrilateral element	*Block2D*	[8, 9]
		Isection2D	
		ProtectedIsection2D	
		Composite2D	
3D	8-noded brick element	*Brick3D*	[8, 9]
		Isection3D	
		Composite3D	

Moreover, a variety of heat transfer material options are available in OpenSees, which are to be assigned to the corresponding heat transfer elements during the analysis. The temperature dependant properties of the heat transfer materials normally include thermal conductivity and specific heat. Conventional construction material types (i.e. steel and concrete) are covered in the heat transfer material libraries [8], following the recommended empirical models from Eurocode [42, 43] and ASCE manual [44]. It is important to note that, the phase change of the materials would induce prompt variations of the specific heat in a short period of time, which further causes numerical difficulties during the heat transfer analysis. This numerical problem is overcome by introducing the enthalpy method [45], which is employed to refine the heat capacity as a smooth function of the temperature [8]. In addition, a 'simple material' is provided in the heat transfer material library, which requires the users to provide its density, thermal conductivity, and specific heat. Different from other heat transfer materials in OpenSees, the parameters of 'simple material' would be kept constant during the heat transfer analysis [8]. More recently, a spray-applied fire-resistive material (SFRM) is implemented in OpenSees [9]. It is a material coating type to protect the steel structural members from fire. An extensive discussion about the features of this type of fire protection can be found in [9].

Table 3.3. Summaries of the heat transfer materials in OpenSees.

Heat transfer material type	**Categories**		
	Heat transfer material	**Heat transfer material Tcl input keyword**	**Key references**
Steel	Eurocode 3-based carbon steel	*CarbonSteelEC3*	[8]
	ASCE manual-based carbon steel	*SteelASCE*	
Concrete	Normal-weight concrete	*NWConcreteEC2*	
	Light-weight concrete	*LWConcreteEC4*	
Fire protection	Spray-applied fire-resistive coating	*SFRMCoating*	[9]
User-defined	A 'simple material' with temperature-independent properties	*GenericMaterial*	[8]

Table 3.3 summarizes the heat transfer materials implemented in OpenSees. It is categorized with respect to the application field of the heat transfer material, corresponding Tcl input keyword, and relevant key references.

Furthermore, to carry out the heat transfer analysis in OpenSees, its fundamental implementations can be found in [8], and the complete Tcl input keyword libraries can be found in [9] with an example Tcl script in its appendix.

3.2.3 Thermo-mechanical Analysis – Predicting the Structural Response

The capability of performing thermo-mechanical analysis in OpenSees, is developed through incorporating a large number of temperature-dependant material and element libraries [21]. So far OpenSees can handle both 2D and 3D thermo-mechanical problems, with the heat transfer generated time-temperature histories as its thermal loading inputs. The input format is a separate individual time-temperature history file for each structural member in manually manner. Due to the mesh difference between a heat transfer analysis and a subsequent thermo-mechanical analysis in OpenSees, an interpolation scheme is used to obtain the desired time-temperature histories at the 'fibres' of the cross sections of the FEM-based structural elements (e.g. displacement-based beam-column element, shell element, etc.) [8, 9]. Fibres in the NFEM refer to a model that subdivides the element into longitudinal fibres, which are assigned with uniaxial materials to be integrated to determine the section flexibilities and element states [46].

Though complex, the users are not required to specify the time-temperature histories at the exact locations of these fibres. Instead, it is required to provide a certain number of data points containing time-temperature histories over the cross section with user-specified locations. The number of required data points and their locations depend on the thermal loading type, the structural member cross section type, and its corresponding structural element type. For example, for a steel beam with rectangular cross section under uniform heating regime (e.g. temperature linearly rises from 20°C to 1000°C), 2 data points which represent the temperature at the top and bottom of the beam member section have to be defined; for an I-beam under user-defined time-temperature histories, 9 data points are preferably required over the depth of the cross section for a 2D model,

or 15 data points are preferably required over the two flanges and the web for a 3D model (see Figure 3.4).

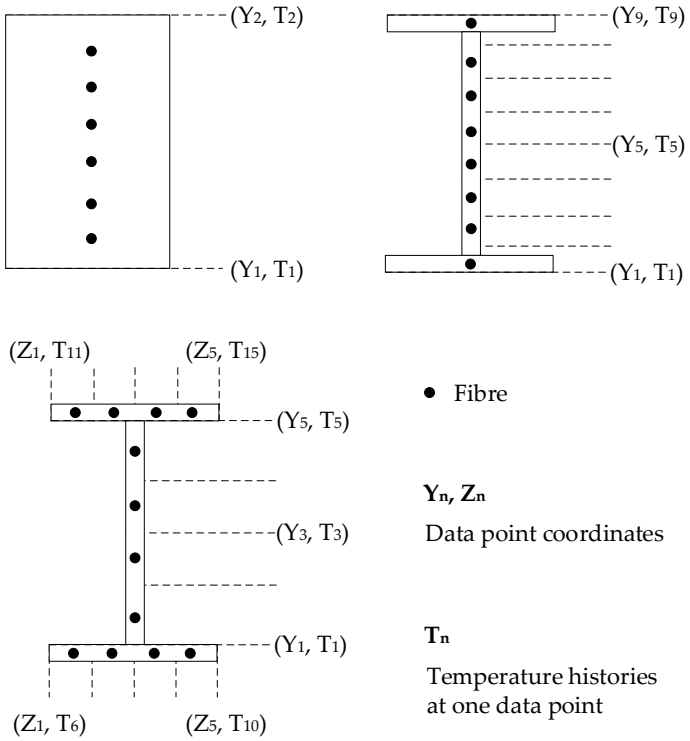


Figure 3.4. Data points locations and fibres in OpenSees for the thermo-mechanical analysis.

Through interpolation to obtain the time-temperature histories at the fibres, the section states and the element states can be determined with considering the thermal elongations and material degradations [10]. It is worth noting that material degradations and thermal expansions are involved with the strain calculations for both steel and concrete materials. However, strains due to creep effect and concrete spalling at high temperatures are not considered in the current version of OpenSees [9].

In OpenSees, the thermal strain $\varepsilon_{thermal}$ is calculated based on the temperature dependant material model assigned with the fibre, and the mechanical strain $\varepsilon_{mechanical}$ is subsequently obtained by deducting $\varepsilon_{thermal}$ from the total strain ε_{total} [14]:

$$\varepsilon_{mechanical} = \varepsilon_{total} - \varepsilon_{thermal} \tag{3.4}$$

The thermo-mechanical analysis of OpenSees follows an incremental-iterative manner with three phases: predictor, corrector, and convergence check. This process is well documented in [11, 14], and schematically displayed in Figure 3.5.

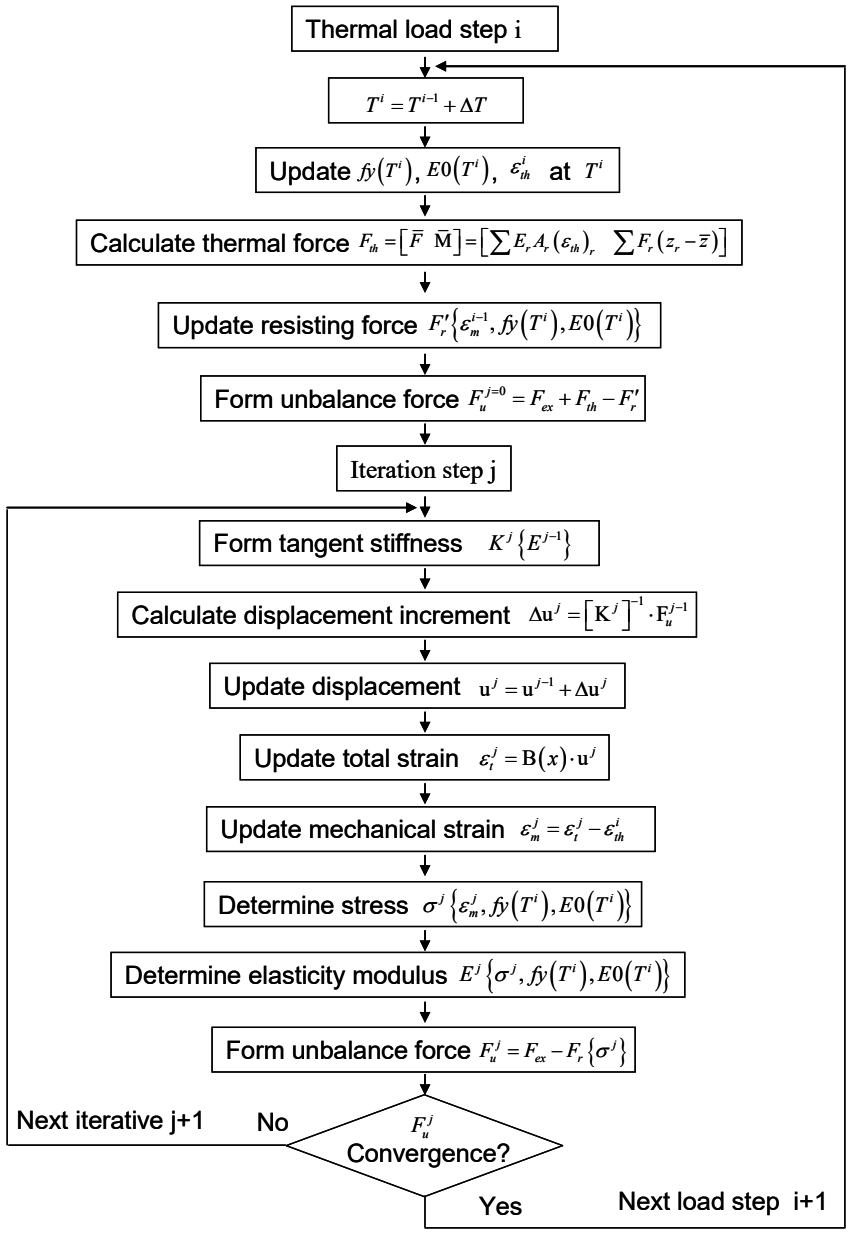


Figure 3.5. Flowchart of the thermo-mechanical analysis in OpenSees [14].

Inherited from the original OpenSees, there are a number of available procedure algorithms for solving the involved nonlinear equations (as illustrated in Figure 3.5), such as the Newton-Raphson algorithm, modified Newton-Raphson algorithm, Krylov-Newton algorithm, and the Broyden–Fletcher–Goldfarb–Shanno (BFGS), etc. [3]. There

are also a number of available choices for the convergence test, such as the norm unbalance test, norm displacement increment test, energy increment test, etc. [3]. Just like other FEM-based software, it is users' responsibility to choose the appropriate solution algorithms and convergence test methods for the analysis. When the structural analysis fails to converge, it is also users' responsibility to judge if it is owing to structural failure, or owing to numerical failure. Pure numerical problems can probably be overcome or improved through changing to the appropriate solution algorithms, or switching the analysis types (e.g. changing from the implicit static analysis to the implicit dynamic analysis).

To model the structural response under fire situations, both geometrical nonlinearity and material nonlinearity are included in OpenSees by incorporating a large number of temperature-dependant material and element libraries [21].

Table 3.4. Structural material types in OpenSees.

ElasticThermal	Steel01Thermal
SteelECThermal	Steel02Thermal
ConcreteECThermal	Concrete02Thermal
DruckerPrager3DThermal	DruckerPragerConcreteThermal

Table 3.5. Structural element types in OpenSees.

dispBeamColumn2dThermal	dispBeamColumn3dThermal
forceBeamColumn2dThermal	ShellMITC4Thermal

Table 3.4 and Table 3.5 are summaries of the structural material types and elements in the current version of OpenSees. Relevant details have been summarized by Jiang [9].

3.3 CONCLUSIONS

In a prescriptive structural fire design code, the fire exposure is usually constrained by the code with limited room for discussion (e.g. nominal time-temperature curves as mentioned above). In a performance-based structural fire design code, the practitioners have greater flexibility and the fire is usually related to advanced fire loading types (e.g. localised fire models, zone models [47], and travelling fires [22], etc.). Although the heat transfer module and thermo-mechanical analysis module in OpenSees are well developed, in order to determine those analyses under more realistic fire boundary conditions, zone models and travelling fires need to be added in OpenSees with transient fire analysis capabilities.

Meanwhile, the development of the OpenSees thermal version is an ongoing project and was initially limited to only a few elements, material models and fire scenarios. Furthermore, a single software to carry out the full set of analyses which includes relatively realistic fire load modelling (e.g. localised and travelling fires); heat transfer to structural components (by radiation, convection, conduction); and the entire structural response, is still unavailable. In order to move towards a more comprehensive solution for a unified analysis, the development of an OpenSees-based integrated research tool is needed, with the aim of producing a framework to perform automated structural fire analyses for large structures under realistic fires. It is intended to be a comprehensive computational tool, which could enable structural engineers to obtain the structural response automatically with the application of the fire load on the structure in the same manner as any other form of load, and so provide a performance-based structural fire engineering tool.

3.4 REFERENCES

[1] F. T. McKenna, "Object-Oriented Finite Element Programming: Frameworks for Analysis, Algorithms and Parallel Computing," PhD Thesis, the University of California, 1997.

[2] S. Mazzoni, F. Mckenna, M. H. Scott, and G. L. Fenves, "Open System for Earthquake Engineering Simulation (OpenSees): User Command-Language Manual," *Pacific Earthquake Engineering Research Center, University of California, Berkeley*, 2006. [Online]. Available: http://opensees.berkeley.edu/OpenSees/manuals/usermanual/index.html. [Accessed: 02-Sep-2017].

[3] F. McKenna, G. L. Fenves, and M. H. Scott, "Open System for Earthquake Engineering Simulation (OpenSees)," *Pacific Earthquake Engineering Research Center*, 2000. [Online]. Available: http://opensees.berkeley.edu. [Accessed: 02-Sep-2017].

[4] J. Franssen and T. Gernay, "Modeling structures in fire with SAFIR: theoretical background and capabilities," *Journal of Structural Fire Engineering*, vol. 8, no. 3, pp. 300–323, 2017.

[5] "Vulcan Software Package," *Vulcan Solutions Ltd.*, 2005. [Online]. Available: www.vulcan-solutions.com. [Accessed: 04-Sep-2017].

[6] B. Izzuddin, "Nonlinear dynamic analysis of framed structures," PhD Thesis, Department of Civil Engineering, Imperial College London, 1991.

[7] A. Usmani, J. Zhang, J. Jiang, Y. Jiang, and I. May, "Using OpenSees for structures in fire," *Journal of Structural Fire Engineering*, vol. 3, no. 1, pp. 55–70, 2012.

[8] Y. Jiang, "Development and Application of a Thermal Analysis Framework in OpenSees for Structures in fire," https://www.era.lib.ed.ac.uk/handle/1842/7941, PhD Thesis, School of Engineering, the University of Edinburgh, 2012.

[9] L. Jiang, "Development of An Integrated Computational Tool for Modelling Structural Frames in Fire Considering Local Effects," https://www.era.lib.ed.ac.uk/handle/1842/19563, PhD Thesis, School of Engineering, University of Edinburgh, 2016.

[10] J. Zhang, "Developing OpenSees Software Framework for Modelling Structures in Fire," PhD Thesis, School of Engineering, University of Edinburgh, 2014.

[11] J. Jiang, "Nonlinear Thermomechanical Analysis of Structures Using OpenSees," https://www.era.lib.ed.ac.uk/handle/1842/7749, PhD Thesis, School of Engineering, University of Edinburgh, 2012.

[12] P. Kotsovinos, "Analysis of the Structural Response of Tall Buildings under Multifloor and Travelling Fires," https://www.era.lib.ed.ac.uk/handle/1842/8007, PhD Thesis, School of Engineering, University of Edinburgh, 2013.

[13] P. Kotsovinos, Y. Jiang, and A. Usmani, "Effect of vertically travelling fires on the collapse of tall buildings," *International Journal of High Rise Buildings*, vol. 2, no. 1, pp. 49–62, 2013.

[14] J. Jiang and A. Usmani, "Modeling of steel frame structures in fire using OpenSees," *Computers & Structures*, vol. 118, pp. 90–99, 2013.

[15] P. Kotsovinos and A. Usmani, "The World Trade Center 9/11 disaster and progressive

collapse of tall buildings," *Fire Technology*, vol. 49, no. 3, pp. 741–765, 2013.

[16] J. Jiang, A. Usmani, and G. Li, "Modelling of steel-concrete composite structures in fire using OpenSees," *Advances in Structural Engineering*, vol. 17, no. 2, pp. 249–264, 2014.

[17] J. Jiang, G. Li, and A. Usmani, "Progressive collapse mechanisms of steel frames exposed to fire," *Advances in Structural Engineering*, vol. 17, no. 3, pp. 381–398, 2014.

[18] J. Jiang, G. Li, and A. Usmani, "Analysis of composite steel-concrete beams exposed to fire using OpenSees," *Journal of Structural Fire Engineering*, vol. 6, no. 1, pp. 1–20, 2015.

[19] J. Jiang, G. Q. Li, and A. Usmani, "Effect of Bracing Systems on Fire-Induced Progressive Collapse of Steel Structures Using OpenSees," *Fire Technology*, vol. 51, no. 5, pp. 1249–1273, 2014.

[20] N. E. Khorasani, M. E. M. Garlock, and S. E. Quiel, "Modeling steel structures in OpenSees enhancements for fire and multi-hazard probabilistic analyses," *Computers and Structures*, vol. 157, pp. 218–231, 2015.

[21] J. Jiang, L. Jiang, P. Kotsovinos, J. Zhang, A. Usmani, F. McKenna, and G.-Q. Li, "OpenSees software architecture for the analysis of structures in fire," *Journal of Computing in Civil Engineering*, vol. 29, no. 1, 2015.

[22] X. Dai, S. Welch, and A. Usmani, "A critical review of 'travelling fire' scenarios for performance-based structural engineering," *Fire Safety Journal*, vol. 91, no. C, pp. 568–578, 2017.

[23] "Asif Usmani's Google Scholar Web Page." [Online]. Available: https://scholar.google.co.uk/citations?user=bAlPTWoAAAAJ&hl=en. [Accessed: 04-Sep-2017].

[24] "OpenSees for Fire - Source Code Download," 2016. [Online]. Available: http://openseesforfire.github.io/download.html. [Accessed: 05-Sep-2017].

[25] J. Franssen, "Structures in fire yesterday, today and tomorrow," in *Fire Safety Science – Proceedings of the Eighth International Symposium*, 2005, pp. 21–35.

[26] V. K. R. Kodur, M. Garlock, and N. Iwankiw, "Structures in fire: state-of-the-art, research and training needs," *Fire Technology*, vol. 48, no. 4, pp. 825–839, 2012.

[27] A. Law, "The role of modelling in structural fire engineering design," *Fire Safety Journal*, vol. 80, pp. 89–94, 2016.

[28] A. Buchanan, "The challenges of predicting structural performance in fires," in *Fire Safety Science – Proceedings of the Ninth International Symposium*, 2008, pp. 79–90.

[29] B. B. Welch, K. Jones, and J. Hobbs, *Practical Programming in Tcl and TK*, 4th Ed. Prentice Hall, 2003.

[30] P. Kamath and L. Jiang, "A Beginner's Guide to Development in OpenSees," 2015. [Online]. Available: https://www.wiki.ed.ac.uk/display/opensees/Download. [Accessed: 13-Sep-2017].

[31] N. Psyrras, D. Panagiotopoulos, N. Gantsidis, N. Lesgidis, and A. Sextos, "Build-X: Seismic Analysis and Assessment of 3D Buildings with OpenSees," 2017. [Online]. Available: https://www.buildx4opensees.eu/. [Accessed: 15-Sep-2017].

[32] T. Kartalis-Kaounis, V. K. Protopapadakis, T. Papadopoulos, and V. K. Papanikolaou, "The ultimate connection between GiD & OpenSees," 2017. [Online]. Available:

http://gidopensees.rclab.civil.auth.gr/. [Accessed: 15-Sep-2017].

[33] M. Petracca, F. Candeloro, and G. Camata, "Scientific ToolKit for OpenSees (STKO)," 2017. [Online]. Available: https://asdeasoft.net/stko/. [Accessed: 15-Sep-2017].

[34] European Standard EN 1991-1-2, "Eurocode1: Actions on Structures - Part 1-2: General Actions - Actions on Structures Exposed to Fire." CEN, Brussels, 2002.

[35] S. Welch, S. Miles, S. Kumar, T. Lemaire, and A. Chan, "FIRESTRUC - Integrating advanced three-dimensional modelling methodologies for predicting thermo-mechanical behaviour of steel and composite structures subjected to natural fires," in *Fire Safety Science – Proceedings of the Ninth International Symposium*, 2008, pp. 1315–1326.

[36] ISO 834-1:1999(E), "Fire-resistance tests - Elements of building construction - Part 1: General requirements." International Organization for Standardization, 1999.

[37] "Standard Test Methods for Fire Tests of Building Construction and Materials, ASTM E119." American Society for Testing and Materials, West Conshohocken, PA, 2016.

[38] Y. Hasemi, Y. Yokobayashi, T. Wakamatsu, and A. V. Ptchelintsev, "Modeling of heating mechanism and thermal response of structural components exposed to localized fires: a new application of diffusion flame modeling to fire safety engineering," in *Thirteenth meeting of the UJNR planel on fire research and safety*, 1996, pp. 237–247.

[39] R. L. Alpert, "Calculation of response time of ceiling-mounted fire detectors," *Fire Technology*, vol. 8, no. 3, pp. 181–195, 1972.

[40] B. Y. Lattimer, "Heat fluxes from fires to surfaces," in *SFPE Handbook of Fire Protection Engineering*, Third Edit., National Fire Protection Association, 2002.

[41] Y. Jiang, A. Usmani, and S. Welch, "Development of heat transfer modelling capability in OpenSees for structures in fire," in *Proceedings of the International Conference on Applications of Structural Fire Engineering*, 2011.

[42] European Standard EN 1992-1-2, "Eurocode 2: Design of concrete structures - Part 1-2: General rules - Structural fire design." CEN, Brussels, 2004.

[43] European Standard EN 1993-1-2, "Eurocode 3: Design of steel structures - Part 1-2: General rules - Structural fire design." CEN, Brussels, 2005.

[44] "Structural fire protection." Vol.78, American Society of Civil Engineers, 1992.

[45] H.-C. Huang and A. S. Usmani, *Finite Element Analysis for Heat Transfer: Theory and Software*, 1st Ed. Springer-Verlag London, 1994.

[46] F. F. Taucer, E. Spacone, and F. C. Filippou, "A Fiber Beam-Column Element for Seismic Response Analysis of Reinforced Concrete Structures, Report No. UCB/EERC-91/17," Earthquake Engineering Research Center, College of Engineering, University of California, Berkeley, 1991.

[47] J. G. Quintiere, "Compartment fire modeling," in *The SFPE Handbook of Fire Protection Engineering*, 3rd Ed., National Fire Protection Association, 2002, pp. 162–170.

[48] "High-level programming language - Wikipedia," 2017. [Online]. Available: https://en.wikipedia.org/wiki/High-level_programming_language. [Accessed: 13-Sep-2017].

[49] "Low-level programming language - Wikipedia," 2017. [Online]. Available:

https://en.wikipedia.org/wiki/Low-level_programming_language. [Accessed: 13-Sep-2017].

[50] A. Iserles, *A First Course in the Numerical Analysis of Differential Equations*, 2nd Ed. Cambridge University Press, 2008.

Chapter 4.

An Extended Travelling Fire Method Framework for Performance-Based Structural Design

4.1 INTRODUCTION

In structural fire design, a key principle that every engineer must follow is to ensure that the fire resistance ability of the structure is larger than the fire severity. In order to satisfy this principle quantitatively rather than qualitatively, there are typically three design domains that structural fire engineers can follow: the time domain, the temperature domain, and the strength domain. The design in the time domain generally refers to the failure time of a structural element under the standard fire, which should be greater than the design fire duration (e.g. one hour, two hours, etc.). The design in the temperature domain normally relates to the maximum temperature in the structural solid under the expected fire, which should be less than the temperature which induces the structural element to fail. And last, the design in the strength domain requires the load capacity of the structural element under the fire to be larger than the applied load, to prevent structural failure [1].

By their definitions, it is obvious that these three design domains are interchangeable if the same structural failure criterion is adopted. However, the reliabilities of these respective methods would diminish when different fire exposure models are used as their fire severity. For example, the standard time-temperature curves (e.g. ISO-834 standard fire [2], or ASTM-E119 fire [3]) are adopted for the structural fire design in the time domain. It assumes that all the structural members in the compartment share the same time-temperature histories at any specific time. This is a reasonable assumption when the compartment size is small. But in the case of vehicles burning in an open car park, which means the burning fuels are relatively small compared with the whole compartment, localised fire models (e.g. Hasemi localised fire model [4]) are considered to be appropriate as the fire severity for the structural fire design. Then the design in time domain becomes inappropriate compared with the design in the temperature domain, or in the strength domain for this case. Hence it is apparent that this dissimilation basically arises from the increasing level of complexity of the structural layout and corresponding appropriate fire scenarios.

This situation would become worse when the design compartment is so large that no existing fire exposure model can be readily used by the structural engineers. It implies

that even if the design satisfies the strength domain, that will not guarantee its reliability, due to the unknown fire severity paired with the large design compartment. A classic example is the 911 terrorism attack on the World Trade Centre (WTC) buildings in New York City in 2001, where fire is regarded as one of the main reasons that caused the buildings' collapse [5]. The National Institute of Standards and Technology (NIST) reconstructed the fire impact inside of the building compartments using the Fire Dynamics Simulator (FDS) code, to further investigate the collapse of the WTC buildings. One of the key findings is that averaging the gas temperature would lead to large errors for analysing the following thermal and structural response, since fire was highly inhomogeneous and travelled in the large compartment, as shown in Figure 4.1 [6].

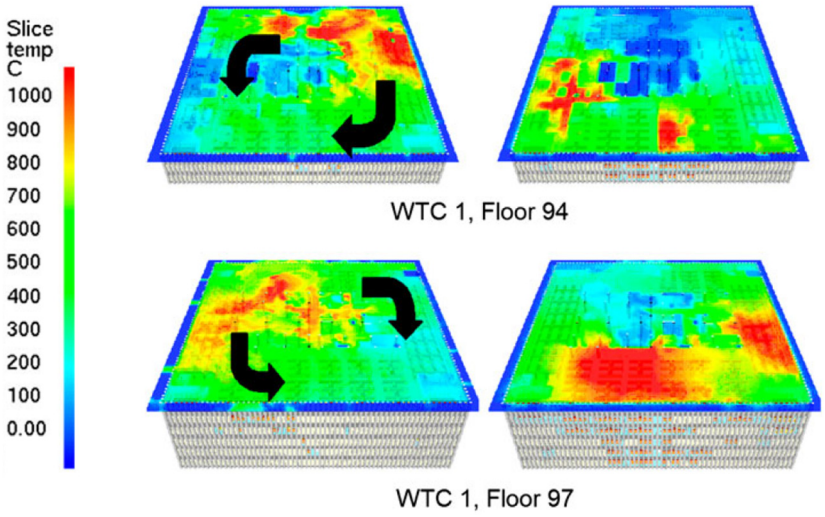


Figure 4.1. FDS simulated fire movement on floors 94 and 97 of WTC 1, adapted from [6].

Apart from the WTC buildings, similar structural failure events observed with high fire inhomogeneity, have been reported several times such as: the Windsor Tower in Madrid in 2005 [7], the Faculty of TU Delft Architecture building in Netherlands in 2008 [8], and more recently the Plasco building in Tehran in 2017 [9]. Furthermore, experimental evidence has also shown high temperature heterogeneity in such compartments and its corresponding threat to the structures. These experiments were reviewed by Stern-Gottfried & Rein in 2012 [10], and the author in 2017 [11]. We may consider the example of the Veselí travelling fire test, which aimed to investigate how the

travelling fires impact the structural components especially for beam-to-column connections, and was conducted at the upper floor of a two-storey steel composite building in the Czech Republic in 2011 [12]. It suggested that the travelling fire should be taken into account as the worst-case scenario, since the cyclic heating and cooling due to the fire movement would cause cyclic deflections of the structural members (see Figure 4.2). This type of fire scenario is not considered in traditional structural fire design assuming the whole compartment holds homogeneous temperature [13].

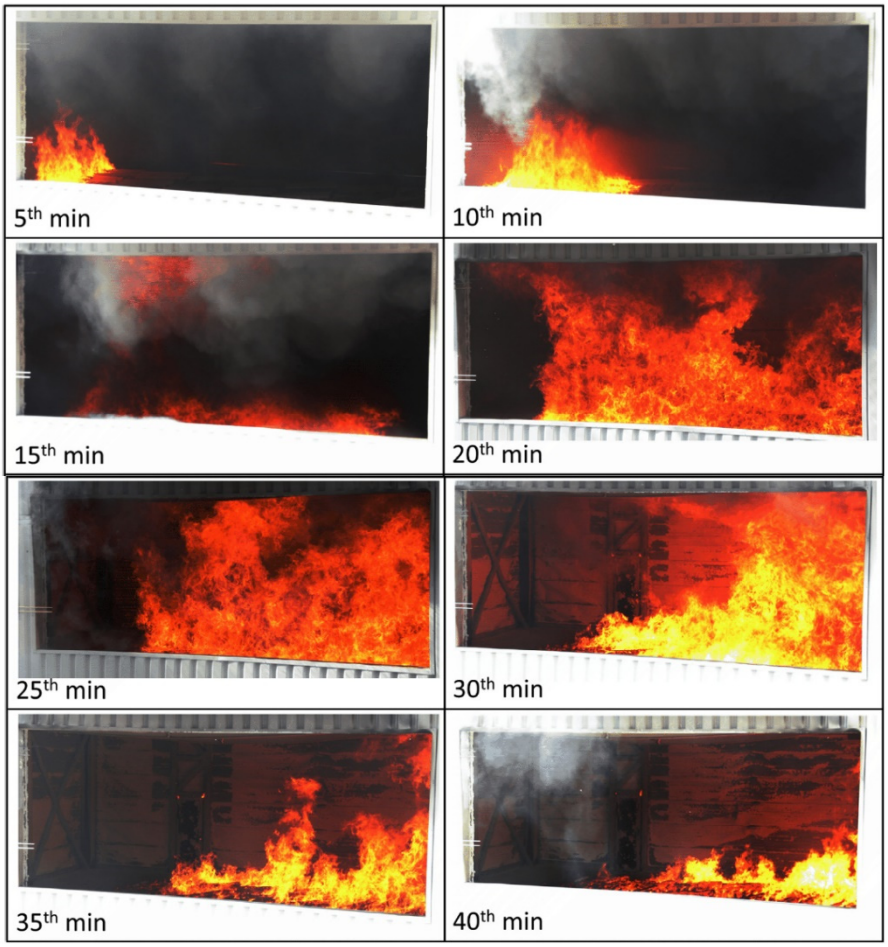


Figure 4.2. Movement of the fire during the Veselí travelling fire test (photos provided by Horová K., CVUT in Prague).

These facts underline the urgent need for a better description of fire scenarios for structural design, recognising the radically different spatial layouts preferred in contemporary architecture. One possible solution is performing Computational Fluid Dynamics (CFD) simulations, which can be used as the fire severity input for structural

design. However, using CFD is not feasible on the day-to-day routine design basis for structural engineers, due to the massive computational expenses required. Moreover, the high fidelity and uncertainties that CFD models would generate, may become to unnecessary or even misleading thermal input information for the structural engineers, as it requires professional fire science knowledge to interpret and judge these results. An alternative solution, as proposed here, is to represent this type of fire scenarios by developing a simple design framework, to address the problem in a practical manner, enabling the structural engineers to utilise the concept without resorting to large computational calculations. An appropriate and efficient level of detail in the model is required to handle these fire scenarios realistically. The work in this chapter is developed on this basis.

The first attempt to characterize the travelling fires in large compartments for structural design, was a design methodology put forward by Clifton in 1996 [14]. This model divides the large compartment into several design areas (named as firecells), which are then subjected to modified parametric fire curves individually and sequentially. In the model, ventilation to firecells, pre-heating of firecells, smoke logging, and cooling after burnout are all considered. In 2007, Rein *et al.* [15–17] put forward an alternative travelling fire methodology, informed by a series of Computational Fluid Dynamics (CFD) analyses with engineering simplifications. It proposed a near field (fire plume near the structure) and a far field (smoke) in the model, to replace the simultaneous burning assumption used in the conventional design approach. In the model, a uniform temperature (800°C-1200°C) is assumed for the near field, and Alpert's ceiling jet model [18] is adopted to calculate far field smoke temperature.

However, in nature, both Clifton's and Rein's travelling fire design methodologies are basically forcing other existing models to 'travel' as a temperature zone (i.e. modified parametric fire curves, 800°C-1200°C temperature block, and the Alpert's ceiling jet model), without making any attempt to consider the essential energy balance in the design compartment. Furthermore, the accumulation of a hot smoke layer is ignored in both models. In Clifton's model, all elements in one 'firecell' (one design area) share the same fire exposure history. In Rein's model, Alpert's ceiling jet was originally developed for weak plume-driven flow field, but its original engineering purpose is for the

sprinklers rather than for structural elements. The uniform 800°C-1200°C assumption is a relatively arbitrary representation of near field thermal exposures. Moreover, both models lack design instructions with relevant information, which can be readily used by the structural fire engineers.

An extended travelling fire methodology (ETFM) framework is proposed in this chapter. It is developed by 'mobilising' Hasemi's localized fire model [4] for the fire plume near the structure, and combined with a simple smoke layer calculation by utilising the FIRM zone model [19] for the areas of the compartment away from the fire. This combined fire model enables the analysis to capture both spatial and temporal changes of the thermal field. It means the heat fluxes generated by the ETFM framework will enable both a heating phase and a cooling phase for each structural member in the large compartment. This chapter presents the whole set of ETFM framework in a systematic manner, rationalising the key assumptions and emphasizing the importance of including the FIRM zone model in the framework. In addition, design instructions with relevant information which can be readily used by the structural fire engineers are also included.

4.2 ETFM FRAMEWORK

The ETFM framework is based on a 'mobile' version of Hasemi's localized fire model [4], which quantifies the local effect of a fire on adjacent structural members, and is combined with the FIRM zone model [19] for the areas of the compartment away from the fire. This ETFM framework accounts for the accumulation of a hot smoke layer under the ceiling, variable fuel load distributions, ventilations, and a predetermined plume propagation trajectory along which it travels (see Figure 4.3 & Figure 4.4).

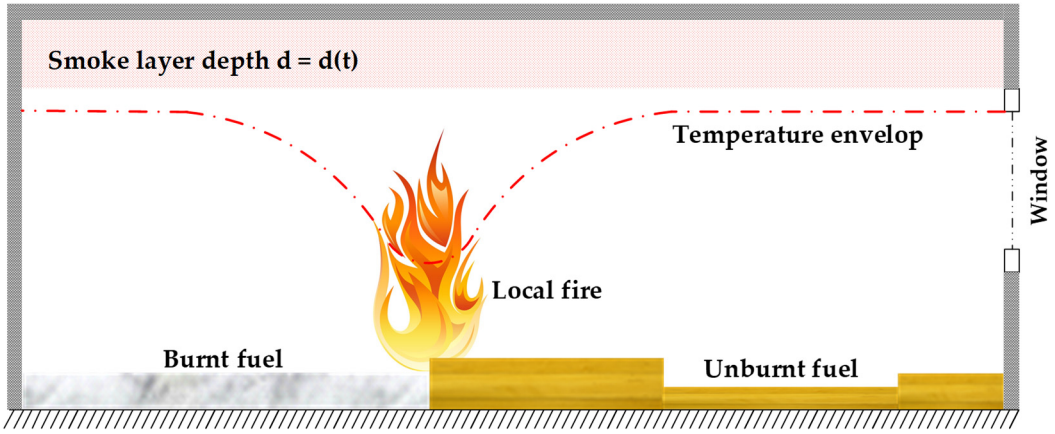


Figure 4.3. ETFM in sectional elevation view.

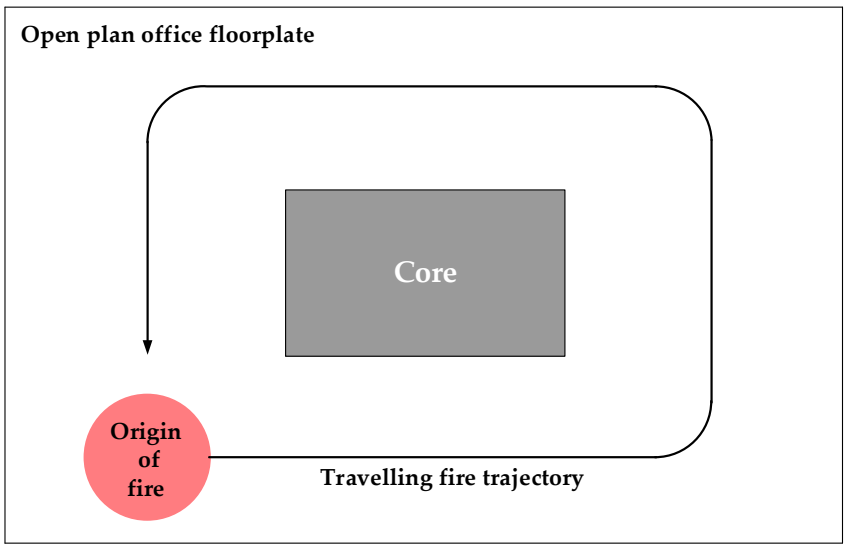


Figure 4.4. ETFM in sectional plan view.

This combined fire model enables the analysis to capture both spatial and temporal changes of the thermal field, thus addressing more fire dynamics than Clifton's model

and Rein's model. Fire temperatures are variable for the near field, contrasting the uniform 800°C-1200°C assumption in Rein's model, while all elements in one firecell share the same fire exposure history in Clifton's model. It is important to note that Hasemi's localised fire model is developed for the situation when fire is impinging the ceiling, and the model has been adopted in the Eurocode for structural design [20].

More importantly, utilising the FIRM zone model into the ETFM framework, means the energy conservation and the mass conservation are both obeyed for the design compartment, rather than simply forcing other existing models to 'travel' (i.e. modified parametric fire curves in Clifton's model, 800°C-1200°C temperature block and the Alpert's ceiling jet in Rein's model). Simply 'moving' other existing simple models in a large compartment, would generate a thermal impact on the structural elements which is not bounded, containing very limited engineering meaning for the structural engineers to use in practice. In addition, the FIRM zone model also enables the ETFM to consider smoke accumulation under the ceiling, which is ignored in both previous models.

4.2.1 Near Field: Hasemi's Localized Fire Model

For quantifying the local effect of the travelling fire on adjacent structural members, Hasemi's localized fire model [4] is utilized in the ETFM framework. This correlation model was originally based on a series of laboratory scale fire tests [21–24] in Japan, with maximum heat release rates up to 900 kW. Then additional validation tests of this model were conducted in Europe with heat release rates from 2 MW to 60 MW, for both large compartments and car parks [25]. Franssen [26] put forward three correlation equations which provide the external heat flux received at the level of the ceiling. These correlations were eventually adopted in the Eurocode 1 [20] for the localized fire model: when the fire plume is impinging the ceiling, the external heat flux, $\dot{h}$ (W/m^2), is given as,

$$\begin{aligned} \dot{h} &= 100000 && \text{if} && y \le 0.30 \\ \dot{h} &= 136300 - 121000y && \text{if} && 0.30 < y \le 1.0 \\ \dot{h} &= 15000y^{-3.7} && \text{if} && y \ge 1.0 \end{aligned} \tag{4.1}$$

The parameter y is obtained by calculating the following equation:

$$y = \frac{r + H + z'}{L_h + H + z'} \tag{4.2}$$

where r (m) is the horizontal distance between the vertical axis of the fire and the point along the ceiling in which the heat flux is calculated, H (m) is the distance between the fire source and the ceiling, L_h (m) is the horizontal flame length (see Figure 4.5):

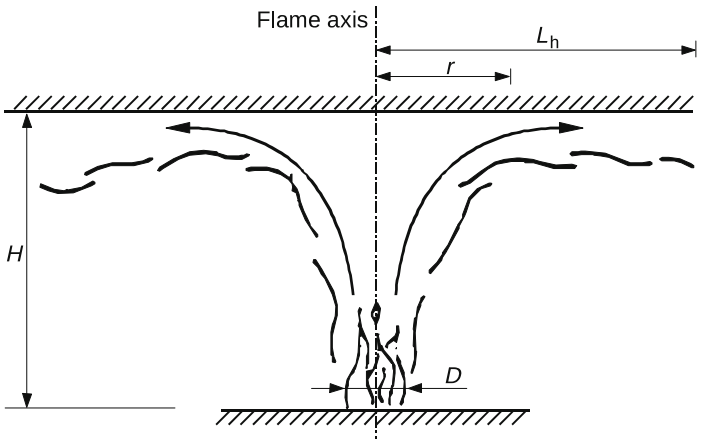


Figure 4.5. Hasemi's localized fire model in Eurocode [20].

L_h (m) is given by the following relation:

$$L_h = (2.9H(Q_H^*)^{0.33}) - H \tag{4.3}$$

with Q_H^* a non-dimensional heat release rate given by:

$$Q_H^* = \dot{Q}/(1.11 \times 10^6 \times H^{2.5}) \tag{4.4}$$

z' (m) is the vertical distance between the virtual fire origin and the fire source, which is given by:

$$\begin{aligned} z' &= 2.4D\left(Q_D^{*\,2/5} - Q_D^{*\,2/3}\right) && \text{if} && Q_D^* < 1.0 \\ z' &= 2.4D\left(1.0 - Q_D^{*\,2/5}\right) && \text{if} && Q_D^* \geq 1.0 \end{aligned} \tag{4.5}$$

where $Q_D^* = \dot{Q}/(1.11 \times 10^6 \times D^{2.5})$, D (m) is the diameter of the fire, $\dot{Q}$ (W) is the heat release rate of the localised fire.

Hence, in order to employ Hasemi's localized fire model into the ETFM framework, three key parameters should be decided transiently: fire origin, fire diameter, D (m), and heat release rate, $\dot{Q}$ (W), because they are each constantly changing when fire 'travels' in the compartment. Details of how these parameters are approximated according to the features of the travelling fire is illustrated in the following several sections.

4.2.2 Far Field: FIRM Zone Model

In most practical buildings, smoke will probably accumulate under the ceiling if its movement is interrupted, due to the walls or smoke protection soffits which are built around the ceiling edges. Therefore, the smoke accumulation is brought into the ETFM framework through utilising a zone model in an elementary way. The depth of the smoke layer is time-dependent and uniformly distributed over the whole ceiling (see Figure 4.6). This feature would reproduce pre-heating and post-heating effects for the structural design, which is the hallmark of this ETFM framework.

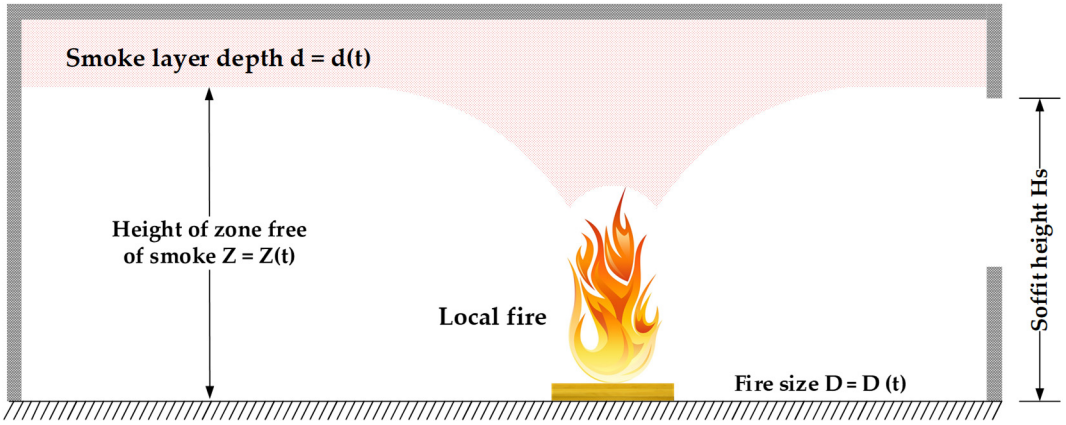


Figure 4.6. Schematic of the smoke for far field in the ETFM framework.

There are several zone models available in the literature, with two popular ones being: OZone [27, 28], and CFAST [29]. However, the ETFM framework employs the FIRM zone model [19] for its smoke layer calculation. The main reason is that FIRM is relatively simple and easy to implement, which matches the ETFM's nature – a framework. At the same time FIRM possesses all the basic components that a zone model should have: fire source, smoke plume, air entrainment, hot upper layer, cold lower layer, smoke flow

through vents, and heat losses through thermal boundaries (i.e. walls, ceilings). Meanwhile, FIRM is the first fire model which is fully documented, validated, verified, and evaluated following the ASTM guidelines back in 2000 [19, 30].

Two time-varying outputs are represented through the FIRM zone model, one is the transient upper smoke layer temperature, T_u (K), and the other one is the evolution of the smoke layer interface height, Z_i (m). The determination of these two variables are via solving a set of ordinary differential equations (ODEs) based on the mass and energy conservations, such as the mass conservation of the lower ambient air [19]:

$$\frac{dZ_i}{dt} = \frac{\dot{m}_a - \dot{m}_l - \dot{m}_e}{\rho_a A} \tag{4.6}$$

where t (s) is the time, $\dot{m}_a$ (kg/s) is the vent flow rate of the ambient air entering the compartment, $\dot{m}_l$ (kg/s) is the lower layer vent flow rate leaving the compartment, $\dot{m}_e$ (kg/s) is the air entrainment mass flow rate, $\rho_a = 1.2$ (kg/m^3) is the density of the ambient air, and A (m^2) is the total compartment area.

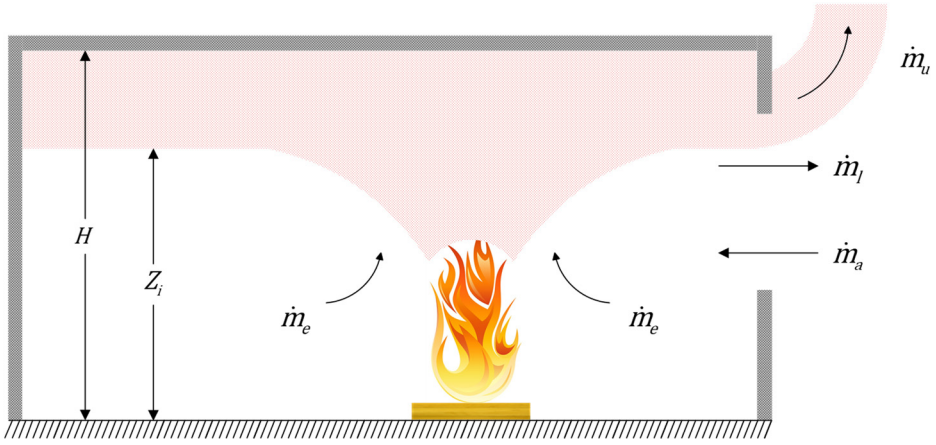


Figure 4.7. Schematic of the mass conservation of the FIRM zone model in ETFM.

Figure 4.7 schematically illustrates the mass balance of the design compartment. In addition to this figure, $\dot{m}_u$ (kg/s) is the smoke vent flow rate leaving the compartment, and H (m) is the clear height of the compartment. Another key ODE is about the upper smoke layer energy conservation [19], which is given as:

$$\frac{dT_u}{dt} = \frac{T_u\left[(1-L_c)\dot{Q} - \dot{m}_e c_p (T_u - T_a)\right]}{c_p \rho_a T_a A (H - Z_i)} \tag{4.7}$$

where $\dot{Q}$ (kW) is the heat release rate of the fire, $T_a = 294.26$ (K) is the ambient air temperature, $c_p = 1.004$ (kJ/kg·K) is the constant specific heat, and L_c is the total heat loss fraction ratio (0.6 ~ 0.9 as recommended in [19]). In addition, L_r is the radiative loss fraction of the fire plume (0.35 as recommended in [19]). The relationship between L_c and L_r is explained in Equation 4.8, where the total heat loss fraction L_c consists of $L_{ceiling}$ (the fraction of heat losses in the form of ceiling convection), L_r, and L_{other} (heat losses fraction due to the roughness of ceilings or aspect ratio of the compartment, suggested from 0 for very smooth ceilings or high aspect ratio compartment, to 0.3 for very rough ceilings or low aspect ratio compartment [19]). In the ETFM framework, only L_c and L_r need to be specified, hence $L_{ceiling} + L_{other}$ can be obtained as a lumped value.

$$L_c = L_r + L_{ceiling} + L_{other} \tag{4.8}$$

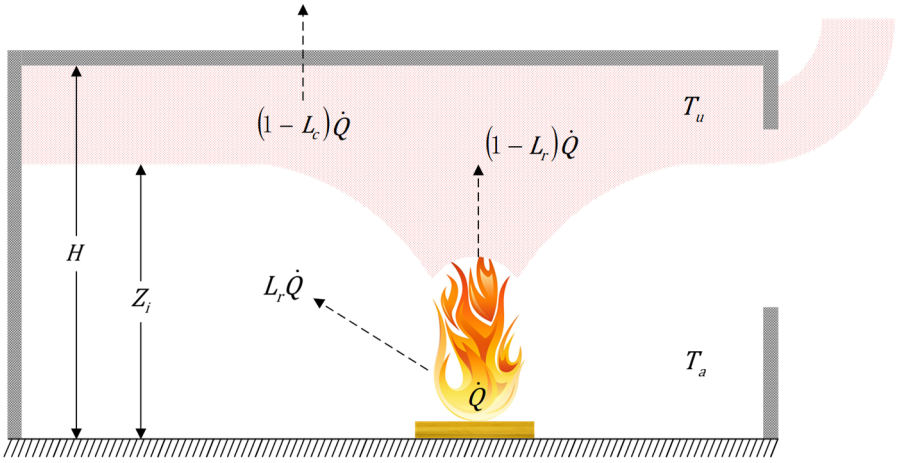


Figure 4.8. Schematic of the energy conservation of the FIRM zone model in ETFM.

Figure 4.8 schematically illustrates the energy balance of the design compartment. Generally, a part of the total heat release rate of fire $L_r\dot{Q}$ radiates away from the combustion region, and the rest $(1-L_r)\dot{Q}$ is convected up through the plume into the formation of the upper hot smoke layer. A fraction of this energy $(L_{ceiling} + L_{other})\dot{Q}$ is lost from the smoke layer to the compartment boundaries through convection and radiation. Then the remaining energy at the upper layer $(1-L_c)\dot{Q}$ would directly

contribute to the gas temperature of the smoke. It is important to note that another main source of energy loss is through the openings. It is accounted by the format of mass loss from the hot smoke layer venting, and explicitly calculated by the mass balance of the upper layer in calculation.

It is worth noting that L_c and L_r are both empirical values but very fundamental to the resultant smoke layer temperature calculations. In addition, there may be no significant differences in terms of smoke temperature rises by using different air entrainment models [31]. Two air entrainment models can be selected in the ETFM framework. One is the Thomas model [32], which is widely used in the UK for venting calculations [33]:

$$\dot{m}_e = 0.188 W_{fi} (Z_i)^{3/2} \tag{4.9}$$

where W_{fi} (m) is the perimeter of the fire. The other one is Zukoski's model [34], which is given as:

$$\dot{m}_e = K(1 - L_r)^{1/3} \dot{Q}^{1/3} (\Delta Z_i)^{5/3} \tag{4.10}$$

where K = 0.076, ΔZ_i is the distance between the fuel top surface and the smoke layer interface. In the ETFM framework, the thickness of the fuel is ignored, hence $\Delta Z_i = Z_i$.

Finally, some limitations should be noted, in particular that the FIRM zone model is applicable with the ventilations with vertical openings through the walls, rather than horizontal openings through the ceiling. Furthermore, the venting is associated with natural ventilations. Forced ventilations are not considered.

4.2.3 Combination of the Near Field and the Far Field

Since Hasemi's equation is applicable to localized fires in an unconfined space and smoke accumulation is not considered in his model, this may lead to the far field predicted gas temperature based on Hasemi's localized fire calculation in a confined space being lower than the actual temperature. Therefore, it is proposed here to combine Hasemi's model with a hot smoke layer calculation (i.e. the FIRM zone model) in the ETFM framework. In other words, the radiant and convective heat fluxes to structural

surfaces can be calculated, based on the summation of heat flux from Hasemi's localized fire model and the heat flux from the FIRM zone model (see Figure 4.9).

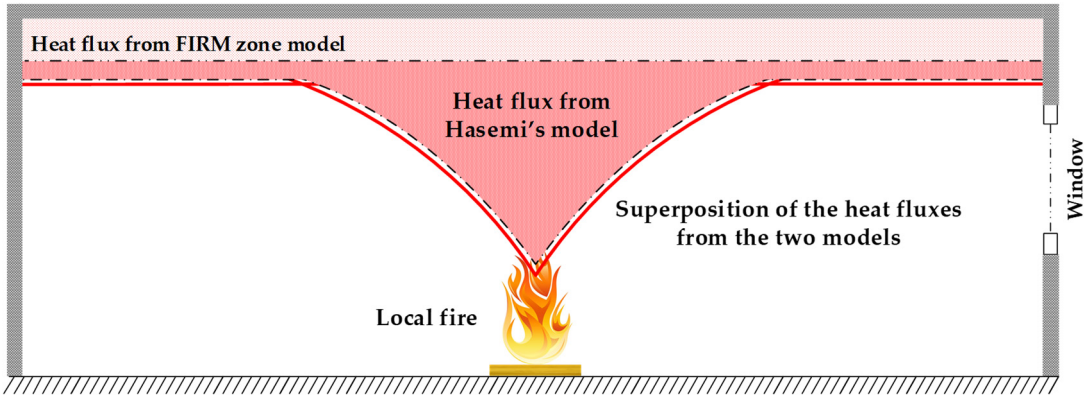


Figure 4.9. Heat fluxes 'combination' of the two models.

The proposed way of combining two models maybe over-conservative for adding heat fluxes from two models in the zone where both have a significant value. However, this is considered to represent a relatively small amount of uncertainty in the general complexity of the overall travelling fire framework for the structural design. A "consistent level of crudeness" should be maintained all through our predictions for the structural fire performance problems [35]. It is an important concept that obtaining one part of the analysis with very accurate data (e.g. fire model analysis), but accompanying with an out of balanced level of accuracy for other part of the analysis (e.g. thermo-mechanical analysis) which is least well defined, is inefficient.

Furthermore, another way of combing two models can also be found in a European research project called 'Development of design rules for steel structures subjected to natural fires in large compartments' [36]. This project aimed to update the Eurocode for the design of steel structures with large compartments under the natural fire safety concept. It was proposed that taking the highest temperature (rather than the superposing heat fluxes in ETFM framework) along the length of the beam, among which was predicted by the Hasemi's localized fire model, or the two-zone model.

4.2.4 Heat Release Rate $\dot{Q}$

The two most important parameters in the ETFM framework are: the travelling fire speed which determines how long the moving Hasemi's localized fire will affect the structural element involved in the localised burning; and its total heat release rate (HRR), $\dot{Q}$, which determines how efficiently the thermal energy will be released due to the fire plume. The total HRR, $\dot{Q}$, discussed in this section is to be used for Equation 4.4 and Equation 4.5 to implement Hasemi's localized fire model, and Equation 4.7 and Equation 4.10 to implement the FIRM zone model, both transiently. Although in Eurocode 1 the expression for calculating HRR, $\dot{Q}$, during the fire growth phase (t-squared fire evolution) is specified [20], the development phase of the travelling localized fire is not considered significant from the structural design point of view. Due to the above reason, and for retaining the simplicity of the ETFM framework, the total heat release rate, $\dot{Q}$(W), is given by the following expression according to Eurocode 1:

$$\dot{Q} = 1000 \times RHR_f \times A_{fi} \tag{4.11}$$

where A_{fi} (m^2) is the burning area of the fuel, RHR_f (kW/m^2) is the maximum heat release rate per unit area. The determination of RHR_f for different occupancies can be referred to Eurocode 1, which is shown in the following table:

Table 4.1. Maximum RHR_f depending on occupancies, adapted from Eurocode 1 [20].

Maximum heat release rate per unit area RHR_f			
Occupancy	**RHR_f (kW/m^2)**	**Occupancy**	**RHR_f (kW/m^2)**
Dwelling	250	Classroom of a school	250
Hospital (room)	250	Shopping centre	250
Hotel (room)	250	Theatre (cinema)	500
Library	500	Transport (public space)	250
Office	250		

Since RHR_f is a value corresponding to the stationary state of the fire, it implies that the fire in the ETFM framework is actually a 'partial post-flashover' fire which covers a certain burning area of the fuel (i.e. A_{fi}), and travels on the floor plate as time evolves. Furthermore, the entrainment-controlled burning is considered in FIRM, which means the upper bound values are assumed for the air mass flow rate $\dot{m}_e$ and corresponding HRR, $\dot{Q}$ [19]. Assuming Zukoski's plume model is employed, $\dot{m}_e$ is changed to:

$$(\dot{m}_e)_{max} = 55K^{3/2}(1 - L_r)^{1/2}(\Delta Z_i)^{5/2} \tag{4.12}$$

and $\dot{Q}$ is changed to:

$$(\dot{Q})_{max} = 3030(\dot{m}_e)_{max} \tag{4.13}$$

As this ETFM framework is basically a localized fire travelling along a predefined trajectory, i.e. one-dimensional, the burning area of fuel A_{fi} is determined by three variables: the travelling fire front edge location derived from the assumed constant fire spread rate, v (mm/s), the travelling fire back edge location derived from the burn-out time, t_b (s), and the compartment width derived from the floor plan dimensions. Figure 4.10 illustrates how the burning area of fuel, A_{fi} (m^2), is obtained schematically. For simplicity and clarity, the lumped fuel is not included in the drawing.

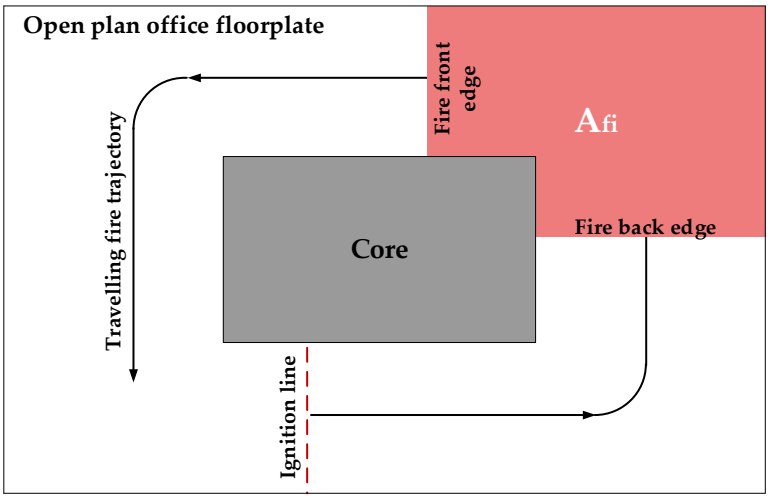


Figure 4.10. The determination of burning area of fuel A_{fi}.

4.2.5 Speed of the Travelling Fire

Table 4.2. Fire spread rate v from experiments and real fire observations summarised by Rackauskaite *et al.* [17].

Data sources	**Fire spread rate v (mm/s)**
Wood cribs in the open	0.1 - 2
Lateral or downward spread on thick solids	1
Tests on natural fires in large scale compartments	1.5 - 19.3
Reconstruction of WTC fires	2.5 – 16.7
St. Lawrence Burns tests	7.5 - 13
First Interstate Bank fire	14.5

As one of the two most important parameters in ETFM framework, the speed of the travelling fire is decomposed into two variables: the constant fire spread rate, v (mm/s), which determines the front edge location of the travelling fire, and the burn-out time, t_b (s), which determines the back edge location of the travelling fire. The resultant of these two variables decides the travelling fire centroid coordinates along the trajectory at each instant of time. The spread rate v is assumed to be a constant, whose corresponding data can be extracted from some experiments and real fire observations, such as the summary done by Rackauskaite *et al.* [17] adapted into Table 4.2.

4.2.6 Burn-out Time t_b

Table 4.3. Characteristic fuel load densities $q_{f,k}$ depending on occupancies, adapted from Eurocode 1 [20].

Characteristic fuel load densities $q_{f,k}$ (average)			
Occupancy	**$q_{f,k}$ (MJ/m²)**	**Occupancy**	**$q_{f,k}$ (MJ/m²)**
Dwelling	780	Classroom of a school	285
Hospital (room)	230	Shopping centre	600
Hotel (room)	310	Theatre (cinema)	300
Library	1500	Transport (public space)	100
Office	420		

In this ETFM framework it is assumed that all fuel would be consumed over the entire fire duration. Therefore, in order to determine the travelling fire back edge location, a burn-out time, t_b (s), is introduced in the ETFM framework. t_b is a similar variable assumed in Rein's travelling fire model [17] for quantifying the time needed for burning out a certain area of fuel completely. It is obtained by the following equation:

$$t_b = 1000 \times q_{f,k}/RHR_f \tag{4.14}$$

where $q_{f,k}$ (MJ/m²) is the characteristic fuel load density. The reference values of $q_{f,k}$ for different occupancies can be referred to Eurocode1 [20], which is adapted into Table 4.3. Figure 4.11 schematically illustrates how the burning area of fuel A_{fi} is determined with the burn-out time t_b concept.

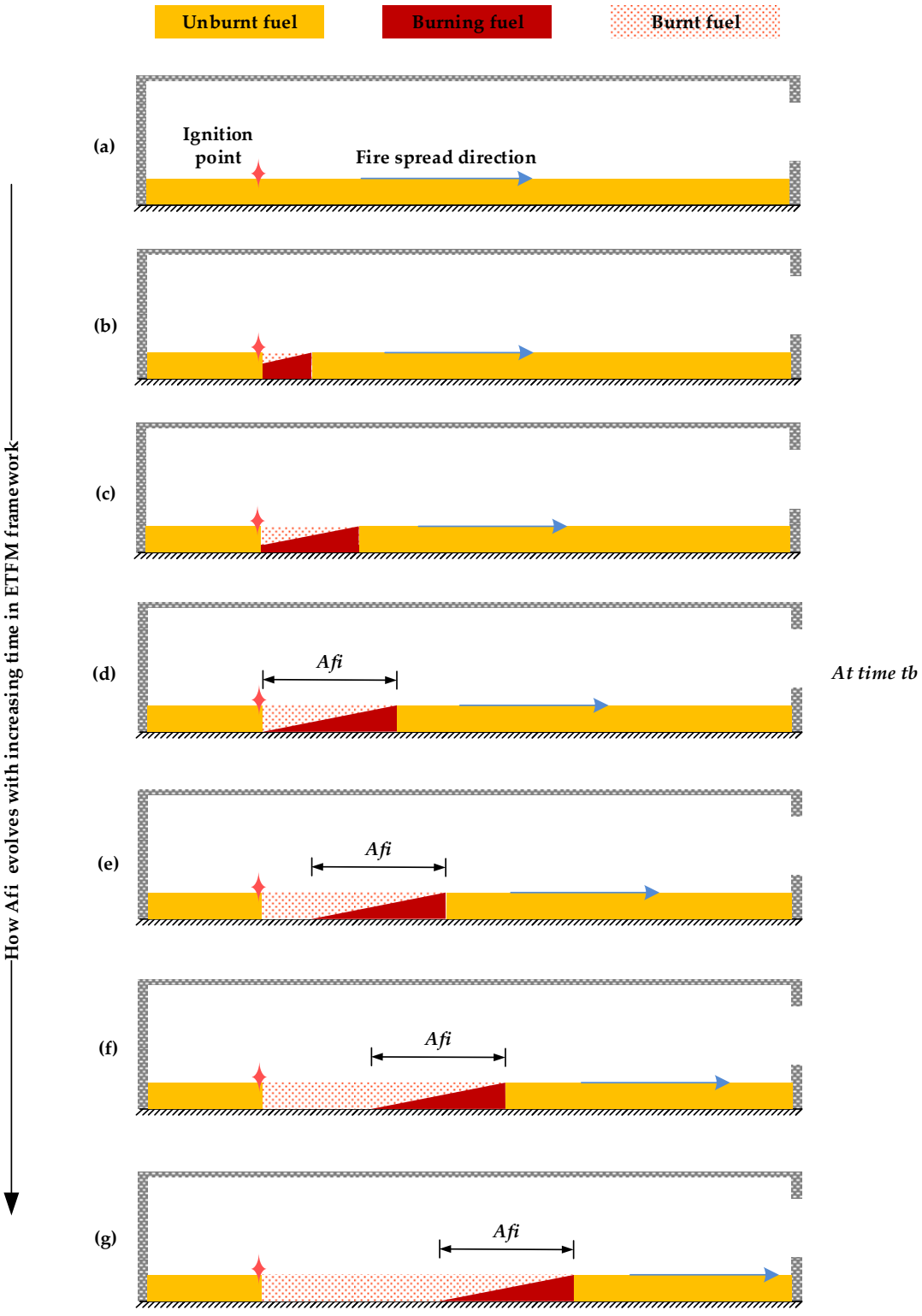


Figure 4.11. Determination of burning area A_{fi} with burn-out time t_b concept – in elevation view.

4.2.7 Approximation of Fire Origin and Fire Diameter D

Once the burning area of fuel A_{fi} is determined, the fire origin of the travelling Hasemi's localized fire model can be obtained, which is defined as the centre of the distance between the travelling fire front edge and back edge along the trajectory. Furthermore, the fire diameter D (m) of the travelling Hasemi's localized fire, can be approximated as the diameter of a circular source of the same burning area of fuel A_{fi} (m^2), which is given by:

$$D = \sqrt{4A_{fi}/\pi} \tag{4.15}$$

4.2.8 Fire Trajectory

As the final objective of this ETFM framework is for its application to structural design, the travelling fire trajectory is assumed to be under the mid-span of the main beams (see Figure 4.4), which would normally represent the worst case for the structural response.

4.2.9 Regulatory Minimum Fuel Depth (RMFD)

A concept of regulatory minimum fuel depth (RMFD) is introduced into the ETFM framework, corresponding to a reference travelling fire spread rate, v (see Table 4.2), and a certain level of fuel load density, $q_{f,k}$ (see Table 4.3). This RMFD is a layer of fuel uniformly distributed over the entire floor plate, and contributes to the total heat flux calculation. The unburnt fuel in Figure 4.11 is a RMFD.

4.2.10 Lumped Fuel

In order to include more flexibility into the ETFM framework, an agreed quantity of additional lumped fuel is placed next to the most critical and/or most vulnerable parts of the structure, identified in consultation with the structural engineer according to performance-based design principles. It is worth noting that lumped fuel is optional in the use of ETFM framework, not like RMFD concept which is compulsory.

Figure 4.12 conceptually illustrates how the travelling fire evolves when the lumped fuel is considered in the ETFM framework. Before the fire travels to the lumped fuel, it follows the same mechanism as it travels on a layer of RMFD (Figure 4.12(a) to Figure 4.12(b)). Further, it is assumed that both the travelling fire spread rate v and the fuel burning rate would not change even if the fire spreads on a local lumped fuel (Figure 4.12 (c) to Figure 4.12(e)). As the fire spread rate is fundamentally a surface property, so it can be regarded that it ought not to depend on what lies below. This is clearly oversimplified but there may be some truth in it in some circumstances.

Figure 4.12(f) and Figure 4.12(g) present the assumption, that the only lumped fuel affected parameter is the burn-out time t_b, which would increase due to the increasing fuel load density $q_{f,k}$ from the lumped fuel (see Equation 4.14). Following that, the burning area of the fuel A_{fi} also enlarges. In addition, when the fire continues travelling, A_{fi} may in principle get 'separated' due to the different burn-out time t_b from a layer of RMFD and a local lumped fuel (Figure 4.12(h)). If the travelling fire is in such circumstances, a hypothesis is made that each burning local fuel would have its own HRR which contributes to the Hasemi's localised fire model and the FIRM zone model. This hypothesis follows a similar way of calculating total external heat fluxes, $\dot{h}_{total}$ (W/m^2), from several separate localised fires recommended in Eurocode 1 – Annex C [20]:

$$\dot{h}_{total} = \dot{h}_1 + \dot{h}_2 + \dot{h}_3 \ldots \leq 100000 \tag{4.16}$$

where $\dot{h}_1, \dot{h}_2, \dot{h}_3$ .... (W/m^2) are the individual external heat fluxes from different local fires obtained from Equation 4.1. The value of 100000 W/m^2 is the upper limit of $\dot{h}_{total}$.

4.2.11 Other Key Assumptions

The ETFM framework considers both the fuel-controlled and ventilation-controlled conditions, with the assumption that sufficient air is available at the beginning and subsequently the glazing adjacent to the fire plume breaks. This is likely to happen in many fires considering window glazing failure at 150°C-200°C [37]. Then the analysis may step into the entrainment-controlled burning, depending on the transient relationship between the air entrainment mass flow rate, $\dot{m}_e$, and HRR, $\dot{Q}$. All fuel is

assumed to be consumed over the design fire duration with: non-uniform burn-out time, t_b, of the travelling fire along the trajectory; changing fuel load density; and variable heat release rates.

A flashover scenario arises naturally in this model, and the fire transitioning from a localized travelling fire to a whole compartment fire when a defined threshold is met, e.g. the temperature of the hot smoke layer reaches 500 °C. It is noted that for determining when flashover occurs in the compartment, there are three commonly-used indicators [33]: 1) the temperature of the smoke in the whole compartment reaches to 500 °C; 2) heat fluxes produced from the fire are as high as 20 kW/m^2 at the floor level; 3) for ventilation-controlled fire, flames flow out of the window. Although the above three are very crude indicators of the flashover state, clause 1) is adopted for the simplicity of implementation of the current ETFM framework.

4.2.12 Limitations of the ETFM Framework

This ETFM framework is developed for providing a more realistic tool for structural design of fire resistance. There are several inevitable limitations in the model. Firstly, it is essentially a 1D trajectory-based travelling fire model, which is currently only applicable to floor plans with a core, or rectangular floorplan shape. Secondly, a potential limitation of the ETFM framework is the applicability of Hasemi's localized fire model, which is only strictly valid for fire diameters is less than 10m, and the rate of heat release less than 50 MW [20], though these are very large fire sizes for typical compartments. Finally, the FIRM zone model is applicable for the ventilations with vertical openings through the walls, rather than horizontal openings through the ceiling, though the latter are much less common, and apply to scenarios with other complexities, e.g. basement fires. Furthermore, the venting is associated with natural ventilations, and forced ventilations are not considered [19].

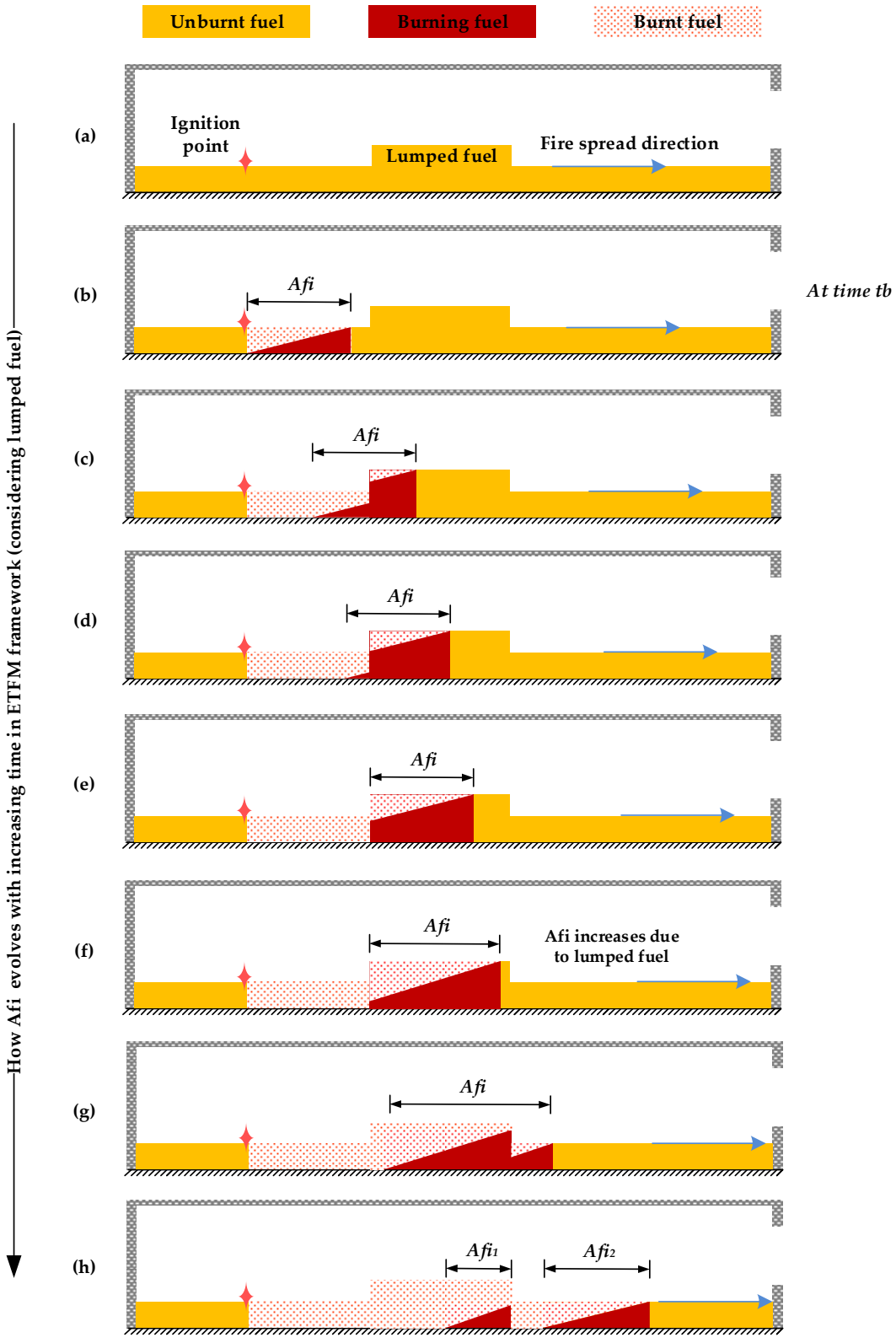


Figure 4.12. Determination of burning area A_{fi} with lumped fuel and RMFD concept – in elevation view

4.3 CONCLUSIONS

This chapter presents an extended travelling fire method (ETFM) framework, which can predict the fire severity in the light of a travelling fire concept with an upper bound, through taking the energy and mass conservations into account. It is developed by 'mobilising' Hasemi's localized fire model for the fire plume near the structure, and combined with a simple smoke layer calculation by utilising the FIRM zone model for the areas of the compartment away from the fire. The key assumptions of the ETFM framework are rationalised and its limitations are emphasized in this chapter. In addition, design instructions with relevant information which can be readily used by the structural fire engineers are also included.

Furthermore, in order to fully (i.e. quantitatively and qualitatively) exploit the thermal and structural response due to the proposed ETFM framework, a numerical tool is needed to implement the framework. Those efforts are presented in the subsequent two chapters.

4.4 REFERENCES

[1] A. H. Buchanan and A. K. Abu, *Structural Design for Fire Safety*, 2nd Ed. John Wiley and Sons, Ltd., 2017.

[2] ISO 834-1:1999(E), "Fire-resistance tests - Elements of building construction - Part 1: General requirements." International Organization for Standardization, 1999.

[3] "Standard Test Methods for Fire Tests of Building Construction and Materials, ASTM E119." American Society for Testing and Materials, West Conshohocken, PA, 2016.

[4] Y. Hasemi, Y. Yokobayashi, T. Wakamatsu, and A. V. Ptchelintsev, "Modeling of heating mechanism and thermal response of structural components exposed to localized fires: a new application of diffusion flame modeling to fire safety engineering," in *Thirteenth meeting of the UJNR planel on fire research and safety*, 1996, pp. 237–247.

[5] A. S. Usmani, Y. C. Chung, and J. L. Torero, "How did the WTC towers collapse: A new theory," *Fire Safety Journal*, vol. 38, no. 6, pp. 501–533, 2003.

[6] R. G. Gann, A. Hamins, K. McGrattan, H. E. Nelson, T. J. Ohlemiller, K. R. Prasad, and W. M. Pitts, "Reconstruction of the fires and thermal environment in World Trade Center buildings 1, 2, and 7," *Fire Technology*, vol. 49, pp. 679–707, 2013.

[7] I. Fletcher, A. Borg, N. Hitchen, and S. Welch, "Performance of concrete in fire: A review of the state of the art, with a case study of the Windsor Tower fire," in *4th International Workshop in Structures in Fire*, pp. 779–790.

[8] D. M. Zannoni, J. G. H. Bos, D. K. E. Engel, and P. dr. U. Rosenthal, "Brand bij Bouwkunde," COT Instituut voor Veiligheids - en Crisismanagement, 2008.

[9] "Plasco Building Fire in Tehran," *Wikipedia*, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Plasco_Building. [Accessed: 08-Oct-2017].

[10] J. Stern-Gottfried and G. Rein, "Travelling fires for structural design–Part I: Literature review," *Fire Safety Journal*, vol. 54, pp. 74–85, 2012.

[11] X. Dai, S. Welch, and A. Usmani, "A critical review of 'travelling fire' scenarios for performance-based structural engineering," *Fire Safety Journal*, vol. 91, no. C, pp. 568–578, 2017.

[12] L. Simões da Silva et al., "Design of Composite Joints for Improved Fire Robustness," RFCS Compfire project-final report (RFSR-CT-2009-00021), European Commission, Luxembourg, 2014.

[13] K. Horová, T. Jána, and F. Wald, "Temperature heterogeneity during travelling fire on experimental building," *Advances in Engineering Software*, vol. 62–63, pp. 119–130, 2013.

[14] C. G. Clifton, "Fire Models for Large Firecells. HERA Report R4-83," HERA publications, New Zealand, 1996.

[15] G. Rein, X. Zhang, P. Williams, B. Hume, A. Heise, A. Jowsey, B. Lane, and J. L. Torero, "Multi-storey fire analysis for high-rise buildings," in *Proceedings of the 11th International Interflam Conference, London, UK*, 2007, pp. 605–616.

[16] J. Stern-Gottfried and G. Rein, "Travelling fires for structural design-Part II: Design methodology," *Fire Safety Journal*, vol. 54, pp. 96–112, 2012.

[17] E. Rackauskaite, C. Hamel, A. Law, and G. Rein, "Improved formulation of travelling fires and application to concrete and steel structures," *Structures*, vol. 3, pp. 250–260, 2015.

[18] R. L. Alpert, "Calculation of response time of ceiling-mounted fire detectors," *Fire Technology*, vol. 8, no. 3, pp. 181–195, 1972.

[19] M. L. Janssens, *An Introduction to Mathematical Fire Modeling*, 2nd ed. CRC Press, 2000.

[20] European Standard EN 1991-1-2, "Eurocode1: Actions on Structures - Part 1-2: General Actions - Actions on Structures Exposed to Fire." CEN, Brussels, 2002.

[21] Y. Hasemi and T. Tokunaga, "Flame geometry effects on the buoyant plumes from turbulent diffusion flames," *Fire Science and Technology*, vol. 4, pp. 15–26, 1984.

[22] Y. Hasemi, Y. Yokobayashi, T. Wakamatsu, and A. Ptchelintsev, "Fire safety of building components exposed to a localized fire - scope and experiments on ceiling beam system exposed to a localized fire," in *ASIAFLAM's 95, Hong Kong*, 1995.

[23] A. Pchelintsev, Y. Hasemi, T. Wakamatsu, and Y. Yokobayashi, "Experimental and numerical study on the behaviour of a steel beam under ceiling exposed to a localized fire," in *Fire Safety Science - Proceedings of the 5th International Symposium*, 1997, pp. 1153–1164.

[24] T. Wakamatsu, Y. Hasemi, and A. V. Ptchelintsev, "Heating mechanism of building components exposed to a localized fire - FDM thermal analysis of a steel beam under ceiling," in *Proceedings of the 1997 16th International Conference on Offshore Mechanics and Arctic Engineering*, 1997, pp. 51–58.

[25] J. Kruppa, D. Joyeux, and B. Zhao, "Scientific background to the harmonization of structural Eurocodes," *HERON*, vol. 50, no. 4, pp. 219–235, 2005.

[26] Commission of the European Community (CEC) (1997b), "Development of Design Rules for Steel Structures Subjected to Natural Fires in Closed Car Parks," 1997.

[27] J.-F. Cadorin and J.-M. Franssen, "A tool to design steel elements submitted to compartment fires—OZone V2. Part 1: pre- and post-flashover compartment fire model," *Fire Safety Journal*, vol. 38, no. 5, pp. 395–427, 2003.

[28] J.-F. Cadorin, D. Pintea, J.-C. Dotreppe, and J.-M. Franssen, "A tool to design steel elements submitted to compartment fires—OZone V2. Part 2: Methodology and application," *Fire Safety Journal*, vol. 38, no. 5, pp. 429–451, 2003.

[29] R. D. Peacock, K. B. Mcgrattan, G. P. Forney, and P. A. Reneke, "CFAST – Consolidated Fire and Smoke Transport (Version 7) Volume 1 : Technical Reference Guide," National Institute of Standards and Technology, 2017.

[30] "ASTM E1355-04, Standard Guide for Evaluating the Predictive Capabilities of Deterministic Fire Models." American Society for Testing and Materials, West Conshohocken, Pennsylvania, 2004.

[31] S. Kumar, R. Chitty, and S. Welch, "Development and validation of integrated fire modelling methodology for smoke ventilation design and hazard assessment in large buildings," in *Proceedings of the 4th International Seminar on Fire & Explosion Hazards*, pp. 629–641,2003.

[32] P. H. Thomas, P. L. Hinkley, and C. R. Theobald, *Investigations into the Flow of Hot Gases in Roof Venting*. BRE Archive, 1963.

[33] D. Drysdale, *An Introduction to Fire Dynamics*, Third Edition. A John Wiley & Sons, Ltd., Publication, 2011.

[34] E. E. Zukoski, T. Kubota, and B. Cetegen, "Entrainment in fire plumes," *Fire Safety Journal*, vol. 3, no. 3, pp. 107–121, 1981.

[35] A. Buchanan, "The challenges of predicting structural performance in fires," in *Fire Safety Science – Proceedings of the Ninth International Symposium*, 2008, pp. 79–90.

[36] Commission of the European Community (CEC) (1997a), "Development of Design Rules for Steel Structures Subjected to Natural Fires in Large Compartments," 1997.

[37] V. Babrauskas, "Glass breakage in fires," *Fire Place, Washington Chapter IAAI Newsletter*, pp. 15–17, 1998.

Chapter 5.

An OpenSees-based integrated tool for modelling structures in fire: SIFBuilder

5.1 INTRODUCTION

In modern day design of structures, durability under natural and man-made hazards has been given more prominence to ensure their longevity. There has been a trend in the structural design industry to adopt performance-based design due to numerous advantages over the conventional prescriptive approach. The performance-based design methodology is now being extended to enhance the fire performance of structures. To achieve this, structural engineers are required to have better understanding of the global behaviour of structures under more realistic fire scenarios, such as parametric fires, localised fires, or even travelling fires for buildings with large compartments. One way to make this happen is by performing automated sequential thermo-mechanical analyses using a single numerical tool. The need for such a tool was recommended in the proceedings of a workshop held at the National Institute of Standards and Technology NIST (NIST) in US in 2002, suggesting to establish a framework (or more likely a patchwork) of models to couple the fire exposure, heat transfer to the structure and structural response in order to support performance-based design [1].

In general, there are two types of computer programs for simulating structural behaviours in fire: research-oriented and for commercial use. The former such as SAFIR [2], VULCAN [3], and ADAPTIC [4] address specific modelling problems, because of a limited number of users and a typically much smaller team of developers. Therefore, these frameworks have natural limitations in their capabilities. The development of such software is academic-research driven and highly prone to the risk of losing valuable development work when team members leave the research group. Commercial software packages such as ABAQUS, ANSYS and LS-DYNA are used by researchers and industry across the world. However limited access to source codes; lack of transparency of the computational framework; and the high cost of purchase and maintenance are major limitations.

In 1997, an open source software framework, Open System for Earthquake Engineering Systems (OpenSees) was developed at the University of California, Berkeley by McKenna [5]. It was initially designed to simulate non-linear response of structural frames under seismic excitations. OpenSees has an object-oriented software architecture

and is written mainly in C++. Object-oriented capabilities enable structural engineers to focus on modelling objects that also have their own attributes and functions rather than just object data. Major attributes such as elements, materials, analysis procedures, and solution algorithms are designed as individual objects and they can be added into the framework freely by anyone anywhere [6]. An active group of OpenSees experts moderate the framework using a version control system, Subversion. This attracts researchers from across the globe to contribute their piece of code to the original framework to expand it and also to help make it more robust and bug-free.

In 2009, OpenSees was adopted at the University of Edinburgh to further develop it to perform structural fire analysis. Significant contributions in terms of heat transfer and fire modules have been made to the framework in developing the 'Thermal' version of OpenSees [7]. Temperature dependent formulations have been incorporated for basic element types (i.e. beam column elements and shell elements) to account for the thermal effects [8]. The material library of the original framework has also been updated by adding new temperature dependant material models for steel and concrete based on the Eurocodes [9].

The development of the OpenSees thermal version is an ongoing project and was initially limited to only a few elements, material models and fire scenarios. Furthermore, a single software to carry out the full set of analyses which includes relatively realistic fire load modelling (e.g. localised and travelling fires); heat transfer to structural components (by radiation, convection, conduction); and the entire structural response, was previously unavailable. In order to move towards a more comprehensive solution for a unified analysis, development of an OpenSees-based research tool named SIFBuilder[1] was started by the author and co-workers in 2014 [10], with the aim of producing a framework to perform automated structural fire analyses for large structures under realistic fires. It is

[1] It is worth to note that SIFBuilder is co-developed with another two colleagues, who are Liming Jiang and Praveen Kamath. Until September 2015, Liming Jiang was the primary developer, the author was the secondary developer, and Praveen Kamath was mainly in charge of software documentations. After 2015 the author was mainly in charge of fire modelling capabilities of SIFBuilder in the University of Edinburgh (see Chapter 5 and Chapter 6), Liming Jiang was mainly in charge of thermo-mechanical modelling capabilities of SIFBuilder in the Hong Kong Polytechnic University.

intended to be a comprehensive computational tool, which could enable structural engineers to obtain the structural response automatically with the application of the fire load on the structure in the same manner as any other form of load, and so provide a performance-based structural fire engineering tool.

This chapter first introduces the key components of SIFBuilder, including model generation, available material and element libraries, fire loading types, heat transfer scheme, and data transmission (which is the key feature of SIFBuilder). The difference between OpenSees and SIFBuilder is emphasized, and relevant online documentation information is also provided. Then a series of benchmark problems are presented chapter, for further verifying and validating OpenSees and SIFBuilder, against analytical solutions, ABAQUS, SAFIR, and experimental data.

5.2 SIFBUILDER KEY COMPONENTS

Unlike other commercial-based software packages, there is no graphical user interface (GUI) in SIFBuilder. Instead, it has a script-based user input capability. Tcl is used to build up the model with a text editor (e.g. Notepad++). Tcl is a high-level, string-based, scripting language [11]. It is embedded into the SIFBuilder application to operate as an interpreter to process the commands from the users. However, building up a SIFBuilder model follows the same workflow as other commonly used finite element method (FEM) software. It first requires the user to input basic structural information for generating the structural model. Procedural scripts are written to specify the geometry, materials, loads, heat transfer parameters, fire type, analysis procedures, solution algorithm and output requirements using corresponding Tcl commands. A typical user input Tcl script for model generation includes: model type definition for identifying the dimension of analysis (i.e. 3D whole structure, 2D plane frame, or single structural element with boundary restraints), geometry of the structure (i.e. bay length in each direction in a Cartesian coordinate system), material type and cross section type for the structure members, and boundary conditions for the structural model.

Following the structural model generation, the user defines the structural loading and thereafter the fire loading information. SIFBuilder is programmed to hold the thermal loading information throughout the structural analysis. Subsequently, the heat transfer analysis module launches for each structural member, and the nodal temperature histories are automatically mapped to the fibres of the structural mesh. Following the heat transfer analysis, a structural analysis is performed on the structure, accounting for the degraded material properties and thermal expansions. Hence, the output generated will be the result of a thermo-mechanical analysis in response to corresponding fire scenarios. Figure 5.1 shows the flowchart of different operation components in SIFBuilder.

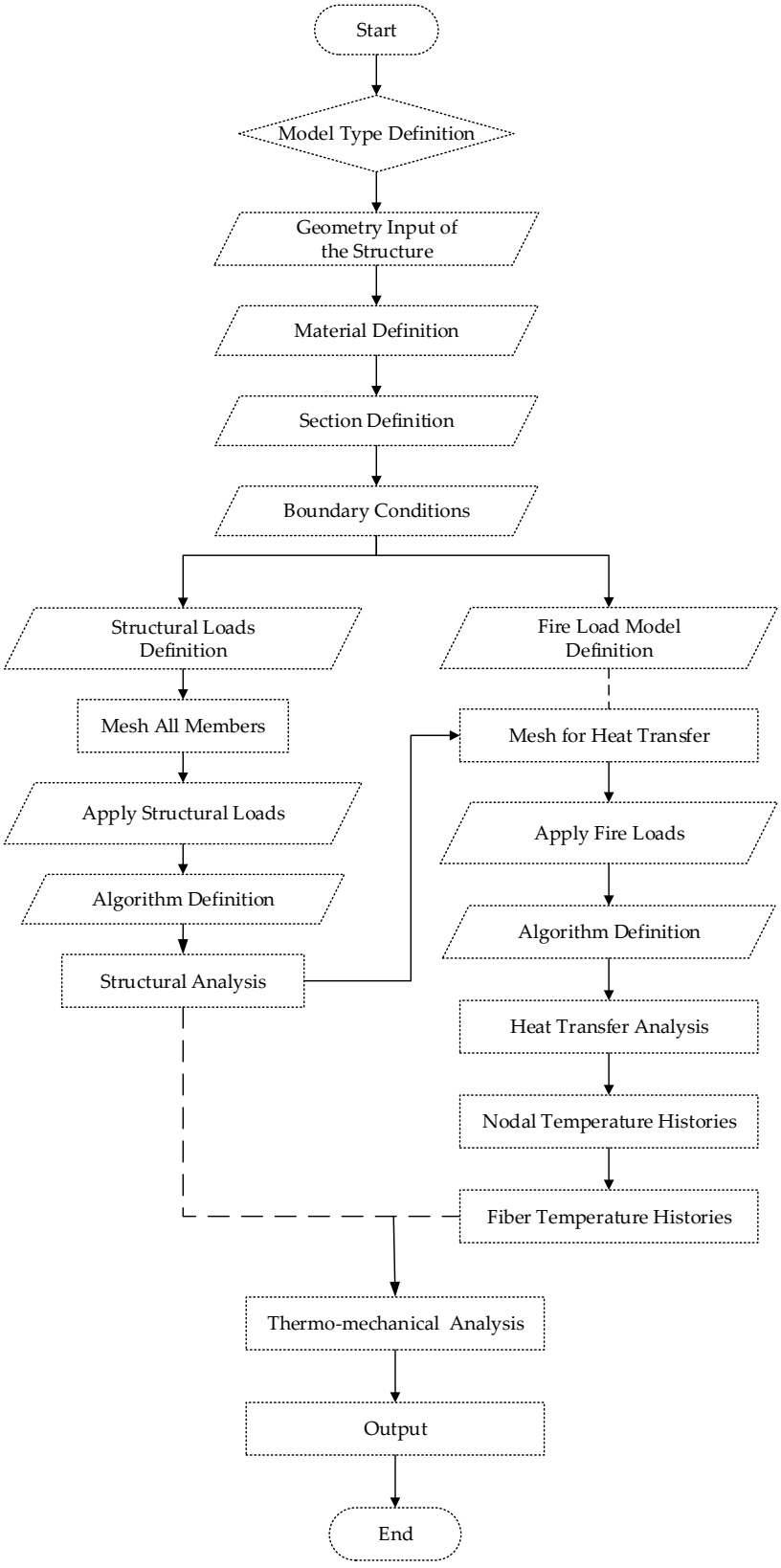


Figure 5.1. Flowchart of SIFBuilder key components.

5.2.1 Model Generation

As mentioned in the previous section, the SIFBuilder software framework lacks a pre-processor, and the procedures for model generation are written in Tcl. A series of higher level Tcl commands are created for generating relatively complex structural models easily and quickly, by just providing the geometry information (i.e. bay length in each direction) of the structure. Currently, SIFBuilder is capable of generating a set of structure types, such as the regular 3D framed-structure including the floor slabs (see Figure 5.2).

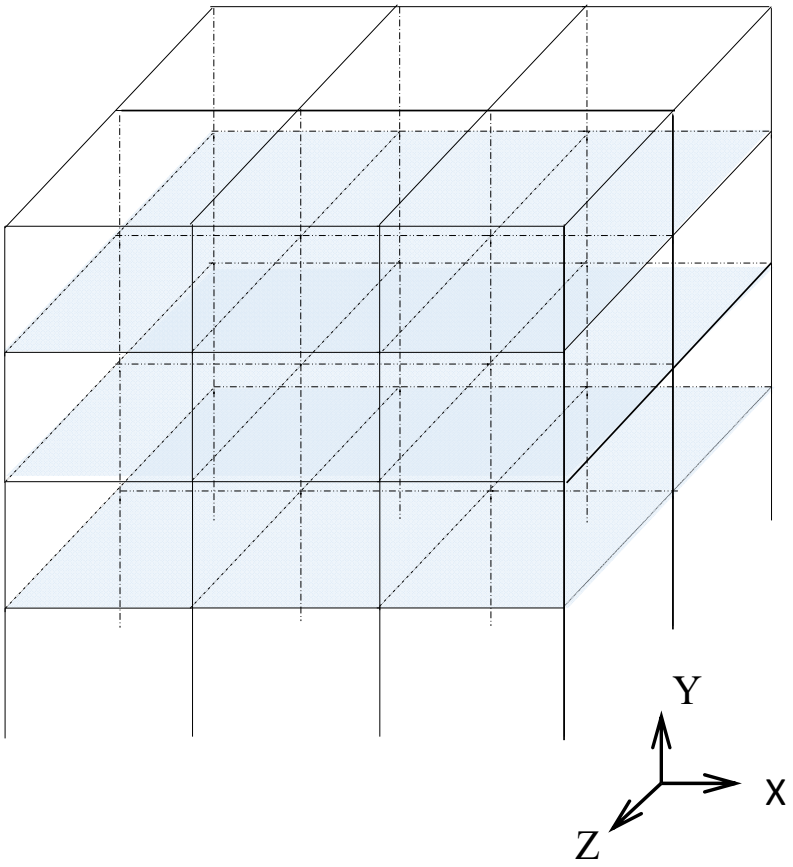


Figure 5.2. Regular 3D framed-structure with floor slabs.

The basic structural frame model available in SIFBuilder has also been extended to generate reduced models such as the X-Z grillage model (see Figure 5.3) with slabs and skewed angles, beneficial in representing asymmetric geometry or bridge models. Similarly, 2D plane frames, single beams, and single columns can also be generated, for the subsequent coupled fire structure analysis in an automatic manner.

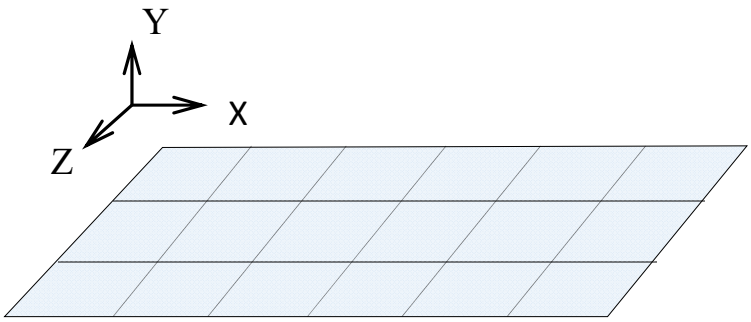


Figure 5.3. Grillage Bridge 3D Model.

5.2.2 Material Libraries

Based on the original 'Thermal' version of OpenSees, SIFBuilder has a rich source of material libraries for both steel and concrete. Moreover, its open source development makes it easy to include user defined materials. The thermal version of OpenSees consists of Eurocode and non-Eurocode based temperature dependent material models. A good choice of material models is essential in performing thermo-mechanical analysis.

To perform structural analysis at high temperatures, Eurocode 3 [12] and Eurocode 2 [13] stipulate uniaxial and multi-axial material models for steel and concrete. Some of the material models that have been incorporated in OpenSees material libraries are shown in Table 5.1. Details of these materials can be found in the 'OpenSees for fire' research group's web portal [14, 15].

Table 5.1. Implemented structural material types in OpenSees

ElasticThermal	Steel01Thermal
SteelECThermal	Steel02Thermal
ConcreteECThermal	Concrete02Thermal
DruckerPrager3DThermal	DruckerPragerConcreteThermal

For instance, *SteelECThermal* contains different steel types based on the Eurocode classification (see Table 5.2) [16].

Table 5.2. Steel Types of *SteelECThermal* Model.

Type tag	Steel Type
EC3	EN1993-1-2:2005 Structural Steel
EC2NH	EN1992-1-2:2004, Class N, Hot Rolled
EC2NC	EN1992-1-2:2004, Class N, Cold Worked
EC2X	EN1992-1-2:2004, Class X

To perform heat transfer analysis, four types of material are developed: *NWConcreteEC2* for normal weight concrete, *LWConcreteEC4* for lightweight concrete, *CarbonSteelEC3*, and *SteelASCE*, which are based the Eurocode and ASCE manual [17]. A user defined material model called *SimpleMaterial* is also provided. All these heat transfer materials hold the information on heat transfer parameters such as thermal conductivity, specific heat, density and enthalpy. More details of heat transfer material models can be found in Chapter 3 and Jiang's PhD thesis [7].

5.2.3 Element Libraries

Table 5.3. Implemented structural element types in OpenSees.

dispBeamColumn2dThermal	dispBeamColumn3dThermal
forceBeamColumn2dThermal	ShellMITC4Thermal

OpenSees offers an extensive element library for SIFBuilder. Due to its efficiency and high accuracy in thermo-mechanical analysis, dispBeamColumn3dThermal (displacement-based 3D beam/column element) and ShellMITC4Thermal are chosen as primary element types during the SIFBuilder model generation process. However, other options (shown in Table 5.3) may also be considered for one's own modelling purpose.

For both 2D and 3D heat transfer analysis utilised in SIFBuilder, the four-noded and eight-noded quadrilateral elements and the eight-noded brick element are used. However, there are more choices in the original 'Thermal' version of OpenSees, which can be found in Chapter 3 and Jiang's PhD thesis [7].

5.2.4 Fire Loads

Fire loads form the salient feature of SIFBuilder tool. A wide variety of well-established fire models are integrated into SIFBuilder, to provide its users the freedom of using different types of fire scenarios upon the structural model types (proposed in section 5.2.1) to assist their design calculations.

Based on the temperature-time curves presented in Eurocode 1 [18], post-flashover fires such as the standard fire (ISO-834), hydro-carbon fire, and empirical parametric fire are all available in SIFBuilder. Although these fire models are relatively simple, they are still widely used for both research and design purposes in fire safety engineering. A user defined external fire curve is also included for providing more flexibility. These idealised uniform fire models are all assumed to have the same temperature distribution in the entire compartment at a specific time. Compared to uniform fire models, more advanced non-uniform fire models are also provided in SIFBuilder. These non-uniform fire models are capable of producing spatially non-uniform gas temperatures in the compartment. It includes two pre-flashover localised fire models: the Hasemi localised fire model [19] and the SFPE localised fire model [20]. These two localised fire models are regarded to be more suitable for carrying out simulations such as vehicles burning in an open plan car park building. More details of fire models[2] in OpenSees can be found in Chapter 3 and Jiang's PhD thesis [7].

[2] It is worth to note that two zone models and an extended travelling fire method (ETFM) framework are also implemented into the current version of SIFBuilder by the author. The implementation details and relevant studies are presented in Chapter 6.

5.2.5 Heat Transfer Scheme

Fire load calculations are succeeded by the heat transfer analysis. SIFBuilder adopts an efficient member identification algorithm, where the heat transfer analysis activates on pre-specified members based on their exposure conditions. Figure 5.4 [3] depicts schematically a localised heat flux distribution adopted in a compartment or building sub-frame.

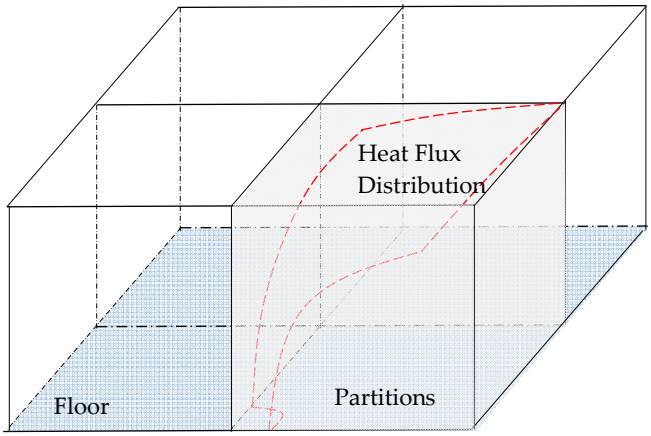


Figure 5.4. Heat flux distribution in a building sub-frame [10].

Figure 5.5 and Figure 5.6 show the implementation of fire exposure conditions in SIFBuilder, for a typical beam and column respectively. These exposure conditions are used by SIFBuilder to obtain time-temperature histories within each structural component. The key capability of SIFBuilder is enabling consideration of non-uniform heat flux, which is a far more realistic situation for large compartments, where performance-based engineering is usually required.

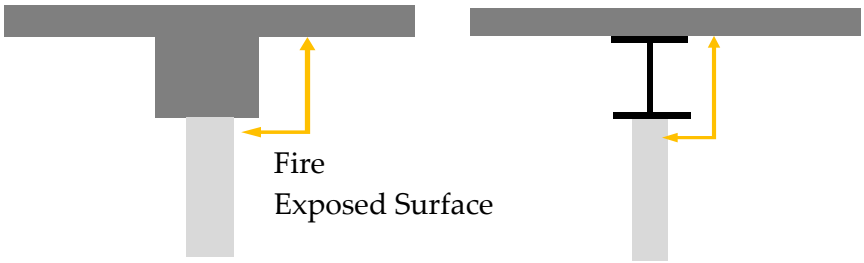


Figure 5.5. Typical beam cross-section fire exposure conditions in SIFBuilder [10].

[3] For the figures with reference number [10] in this Chapter, it means these figures are not generated by the author. These figures are from colleague Liming Jiang, who co-authored with the author in 2015 on a conference paper: "OpenSees-based Integrated Tool for Modelling Structures in Fire".

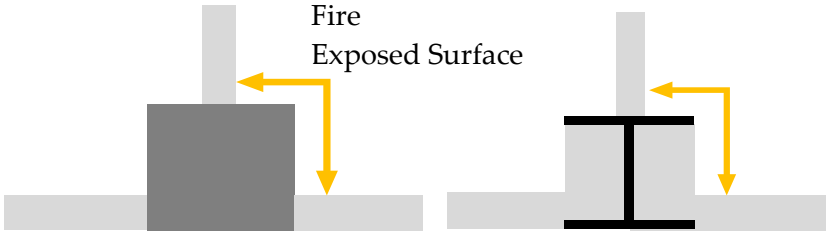


Figure 5.6. Typical column cross-section fire exposure conditions in SIFBuilder [10].

SIFBuilder is designed for simulating whole-frame structural behaviour in realistic fires. To achieve this, an efficient heat transfer strategy is employed for saving computational resources and offering additional flexibility to the end users. Repeated tests on the developed module has confirmed that for idealised uniform fires in a compartment, 1D heat transfer for the slab and 2D heat transfer for beam and column cross-sections holds good and hence, this strategy is adopted in SIFBuilder. Since heat flux is spatially invariant over structural component surfaces under idealised uniform fire scenarios this can be regarded as a reasonable approach without significant dispute.

For compartments under idealised non-uniform fires such as the Hasemi's localised fire model, the incident heat flux on structural members varies with location. The same approach as above is more questionable here, however numerical tests carried out by Jiang [16] show that even in this case this approach is highly feasible. Localised 1D heat transfer analysis for the slab and a series of 2D heat transfer for beam and column cross-sections is implemented and temperature-time histories between sections are obtained by interpolation, instead of running full 3D heat transfer analysis. Schematic representations of this strategy are illustrated in Figure 5.7 and Figure 5.8.

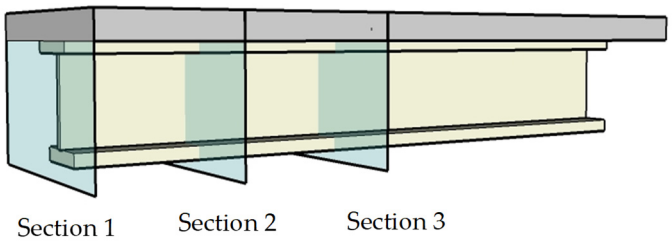


Figure 5.7. Efficient heat transfer strategy for 3D beam members in SIFBuilder [10].

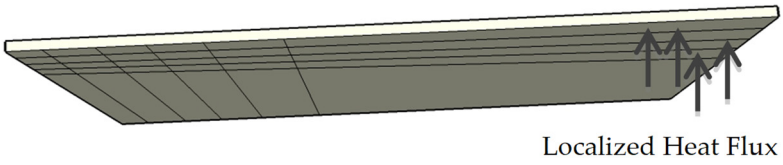


Figure 5.8. Efficient heat transfer strategy for 3D slab members in SIFBuilder [10].

5.2.6 Data Output and Transmission – the KEY Feature of SIFBuilder

As discussed above, both the fire loading calculation and the heat transfer analysis are implemented in a realistic, accurate, and efficient way. Moreover, as this convenient computational tool is not a black box, it is easy for users to customise their models and access the intermediate and final analysis information. The temperature history from the heat transfer module can be easily accessed by defining proper Tcl 'recorder' commands in the model script. Many other simple tasks such as monitoring the analysis by creating break points, obtaining the desirable node output by specifying parameters, simple debugging of the input scripts is made possible by adapting simple tips and tricks offered by Tcl.

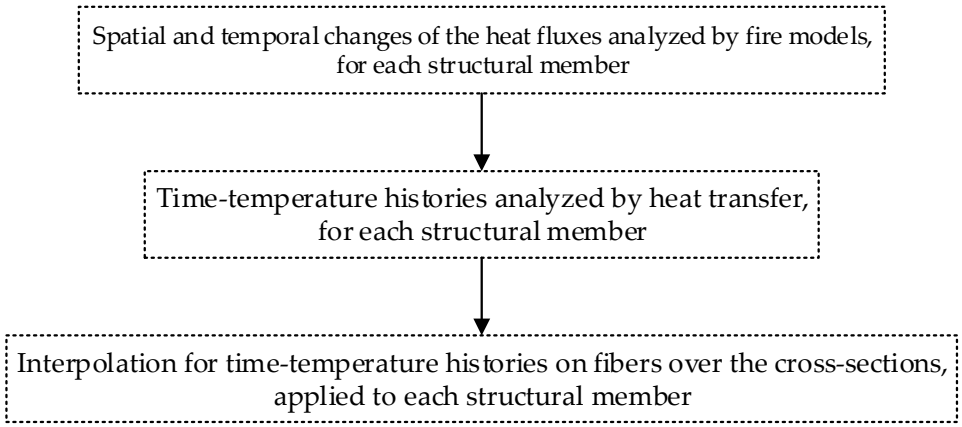


Figure 5.9. Data transmission from fire modelling to heat transfer, then to subsequent thermomechanical analysis.

The most prominent and important feature of SIFBuilder is that it can carry out the fire and heat transfer analysis for every fire-involved structural member sequentially and automatically, then map the heat transfer temperature results to the corresponding

structural elements during the global structural analysis. Figure 5.9 shows the flowchart of the data transmission scheme within the SIFBuilder.

This feature would reduce the modelling time expense to a great extent, compared with manually transferring the heat fluxes from the fire modelling to the heat transfer analysis, then interpolating the desired time-temperature histories at the fibres over the cross-section for each structural member. For modelling single structural members or simple portal frame structures under uniform fires, this feature would save time only in the scale of hours. However, once the modelling tasks upgrade to large model involved with realistic fire scenarios, such as full 3D steel-composite structure under travelling fires, this time expense saving feature would become more advantageous, in the scale of days and even months[4].

5.2.7 The Relationship between OpenSees and SIFBuilder

As explained in the previous sections, the OpenSees components (e.g. temperature-dependant material models, displacement-controlled beam column element considering geometrical nonlinearity, etc.) are actually the core of performing the structural fire simulations. SIFBuilder in nature is an efficient but novel way of combining all these OpenSees components together, to model structures in fires with different levels of complexities, such as a single cantilever beam under standard fire, or a plane frame with several compartments under parametric fires, or a 3D structure within a large compartment under travelling fires.

5.2.8 Documentations for SIFBuilder

All the source codes, examples, user and developer manuals on SIFBuilder can be found at the in the 'OpenSees for fire' research group's web portal [14, 15], so any interested researchers can review, download, and use it for free.

[4] This advantage has been approved by the parametric studies carried out in Chapter 7, which involves 29 travelling fire scenarios containing total number of 696 heat transfer analysis for the structural members, finished in a two weeks' time.

5.3 BENCHMARK PROBLEMS FOR OPENSEES

The relationship between OpenSees and SIFBuilder is explained in the previous section. To fully exploit the potential of SIFBuilder, the accuracy of the calculation components from OpenSees must be verified and validated. Verification and validation of the OpenSees heat transfer modelling capability was carried out by Y. Jiang [7], and the thermo-mechanical analysis capability was performed by J. Zhang [9] and J. Jiang [8]. However, there is still room for further verifications and validations to be performed, as the development of 'OpenSees for fire' is still an ongoing research project.

This section presents a series of benchmark problems for the verification and validation of OpenSees, spanning from analytical solutions, other software packages (ABAQUS and SAFIR), and experiments.

5.3.1 Heated Elastic Beams with Finite End Restraints

Conventional fire design codes are based on isolated single structural members under standard fire test with simply-supported boundary conditions [21, 22]. However, more and more criticism has been raised from the steel industry in recent decades as this approach to structural fire resistance design is too conservative and unrealistic. One of the seminal events that illustrated this was the Broadgate phase 8 fire in 1990. Its investigation report [23] suggested that the restraints may be beneficial to the deformation of a frame, and thermally induced forces due to restraints may not be neglected. Usmani *et al.* [24] confirmed the above drawbacks, and systematically presented a series of fundamental concepts of structural behaviour under thermal effects, involving the complex relationship among deflections, thermal induced strains, buckling, and different level of lateral and rotational restraints. Liu *et al*. [25] performed an experimental investigation on the effect of horizontal restraints to the steel beams in fire. They concluded that with higher axial restraints, the catenary action becomes more prominent, which may decrease the running-away effect with large deflections. Yin and Wang [26] carried out a numerical study on the large deflections of the restrained steel beams under high temperatures, emphasizing on the effect of different level of restraints in catenary action. They suggested that appropriate axial restraints will involve the steel beams into

catenary action without a collapse, even under very high temperature. Yin and Wang [27, 28] continued their research by developing a simplified hand calculation method for the catenary action of the restrained steel beams under uniform and non-uniform temperature distributions. This method is based on a simplified assumption that the beam's deflection profile is related to the loading condition and its rotational restraint, without considering the stress distribution in the beam. Tan and Huang [29, 30] found the critical temperature of restrained steel beams is decreased by higher horizontal restraint, and increased by rotational restraint. Kodur and Dwaikat [31] numerically studied the response of steel beam-columns under fire, with considering different factors such as the creep of steel material under high temperature, load level, effect of end restraints, and fire scenarios. One of the key findings is that the axial restraint is detrimental, but rotational restraint is beneficial for the steel beam fire resistance. Although the previous research has extensively investigated the end restraint effects on the steel beams behaviour, none of the above performed their research according to the basic beam theory under large deflections.

Hence, in this section, a mathematical model is first presented for determining the nonlinear response of beams with finite end restraints to transverse mechanical loads and through-depth thermal gradients[5]. Then the finite end restraints are modelled using translational and rotational springs at the end of the beam using OpenSees, thereby representing the cool surrounding structure. Six boundary conditions of the beam are investigated for the involved benchmark problems, to represent the most realistic boundary conditions including beams with translational restraint alone, with rotational restraint alone and with both translational and rotational restraints. Finally, software verification studies using OpenSees, ABAQUS, and this mathematical model are demonstrated.

[5] This mathematical model is developed by colleague Payam Khazaeinejad, who co-authored with the author in 2015 on this work, with a conference paper "Nonlinear analysis of heated elastic beams with finite end restraint" [35].

5.3.1.1 Mathematical model based on beam theory

To obtain analytical solutions for beam members considering geometrical nonlinearities, the coupling between the beam internal axial force and beam out-of-plane displacement must be taken into account in the beam theory. Since this mathematical model is not developed by the author, its derivation details are presented in the Appendix A.1.

5.3.1.2 Benchmark studies with six types of boundary conditions

To verify the OpenSees displacement-based beam column element (considering the geometrical nonlinearity at large deflections) and the simple material model (considering thermal expansion), a set of beam benchmark problems with six types of boundary conditions are investigated. It is worth to note that the material degradation at high temperature is not considered in these benchmark problems. The reason is because the motivation of the work in this section is for OpenSees verification, hence the thermal effects on the beam member has to be decoupled into simple components. Furthermore, according to the benchmark modelling investigations performed by Gillie [32], geometrical nonlinearity due to the thermal expansion has a more detrimental impact on the heated member, compared with the material degradation due to high temperatures. Therefore, in the following benchmark studies in this subsection, the beam is assumed to consider thermal expansion only, and material degradation is not taken into account.

These benchmark studies are based on a beam member with six types of boundary conditions, as shown in Table 5.4. The stiffness of the horizontal spring and rotational springs are $0.2\times EA/L$ and $8\times EI/L$ respectively, in which EA/L is the axial stiffness of the beam, EI/L is the bending stiffness of the beam. Moreover, the ratios (i.e. 0.2 and 8) for the stiffness of the end restraints are adopted to enable the investigated elastic beam to capture typical beam behaviours at high temperatures. However, the ratios can also be studied in a more extensive manner (i.e. 0.1 - ∞), which are not presented here.

The investigated elastic beam length, L, is 6 m, with cross-section width, d, 0.1 m, and cross-section depth, h, 0.2 m. The elastic modulus of the beam, E, is a constant value of 200×10^9 N/m^2. Moreover, its thermal expansion coefficient, α, is 12×10^{-6} /°C, and Poisson's ratio, ν, is 0.3.

The beam is subjected to a UDL of 1.2 kN/m. Since the beam is elastic, there is no yield bending moment capacity for estimating the design mechanical loadings under high temperature. Therefore, the deflection design criteria induced by UDL at ambient temperature is adopted, which should be less than the beam length over 400:

$$\frac{1}{400} \times 6 \geq \frac{5 \times q_d \times 6^4}{384 \times 2 \times 10^{11} \times \frac{0.1 \times 0.2^3}{12}} \tag{5.1}$$

Then it is obtained $q_d \leq 11.85$ kN/m, hence $0.1 \times q_d \approx 1.2$ kN/m is assumed to be the magnitude of UDL, q, as the applied mechanical loading for the heated beam.

A linear thermal gradient through the beam depth is applied as the thermal loading. The temperature at the top surface of the beam is assumed to be at ambient temperature (i.e. 20˚C), while the temperature at the bottom surface varies linearly from 20˚C - 1020˚C. Hence a through depth thermal gradient over the beam cross-section is formed.

Moreover, the heating regime setup for these benchmark studies, is based on the "fundamental principles of structural behaviour under thermal effects" done by Usmani *et al.* in 2001 [24]. According to this work, if temperature in the longitudinal direction of a beam is assumed to be uniformly distributed, then its structural behaviour would be dominated by the temperature distributions over its cross-section depth, and boundary conditions. Figure 5.10 schematically presents the thermal effects over the beam depth with 'a simplified equation': a realistic temperature distribution over the depth of a beam is decomposed into uniform temperature increase (ΔT) causing thermal expansion, and thermal gradient (Ty) causing thermal bowing effect.

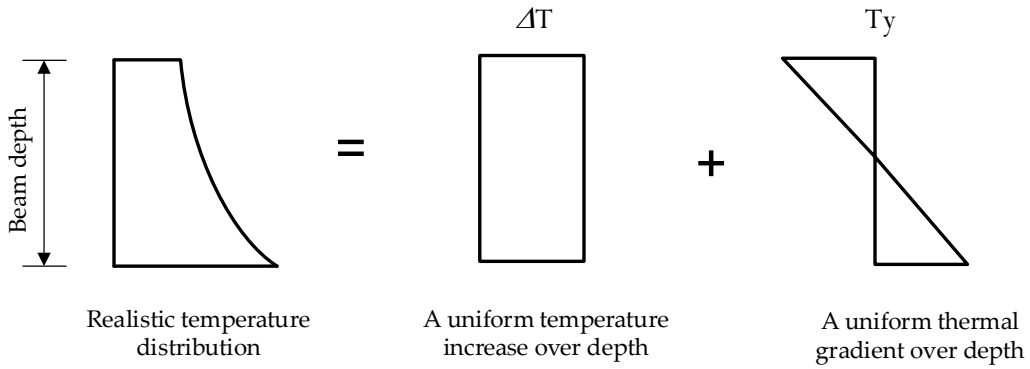


Figure 5.10. Decomposition of temperature effects over the depth of a beam section with thermal expansion and thermal bowing effects.

Table 5.4. Various types of boundary conditions for the heated elastic beam.

Boundary condition tags	Beam configurations	k_{tr}	k_r
BC1		0	0
BC2		$0.2\times EA/L = 1.34\times10^{8}$ N/m	0
BC3		∞	0
BC4		∞	$8\times EI/L = 1.76\times10^{7}$ Nm/rad
BC5		∞	∞
BC6		$0.2\times EA/L = 1.34\times10^{8}$ N/m	$8\times EI/L = 1.76\times10^{7}$ Nm/rad

If the heated beam is restrained by a certain level of boundary conditions (i.e. finite rotational restraints and horizontal restraints, as shown in Table 5.4), internal forces would be induced. The thermal expansion due to ΔT would be constrained by the horizontal restraints, hence an internal compressive axial force would be generated during the heating stage. Meanwhile, thermal bowing due to Ty would be constrained by rotational restraints, hence a hogging bending moment would be generated during the heating stage.

Figure 5.11 - Figure 5.14 show a pattern of development for beam mid-span deflections and axial forces with an increase in temperature in beams with elastically restrained ends.

In the case of a simply supported beam, denoted by BC1, the beam axial force is zero and the beam deflection increases linearly during the heating regime (see Figure 5.11 and Figure 5.12). This occurs because there is no horizontal restraint, so the beam can expand freely in the beam longitudinal direction. In the case of BC2, the presence of horizontal restraint generates larger beam deflection compared with BC1, as the thermal expansion in the beam longitudinal direction is constrained and accompanied with a compressive force as shown in Figure 5.11. BC3 stands for an extreme case when the stiffness of the horizontal restraint increases to infinity. The value of deflection is larger than that of the other cases (i.e. BC1 and BC2). Moreover, a larger value for the axial force is produced, which sharply increases and then decreases. The reason is because when the compressive axial force sharply builds up at around 150 ˚C, the beam is buckled then the compressive axial force is released.

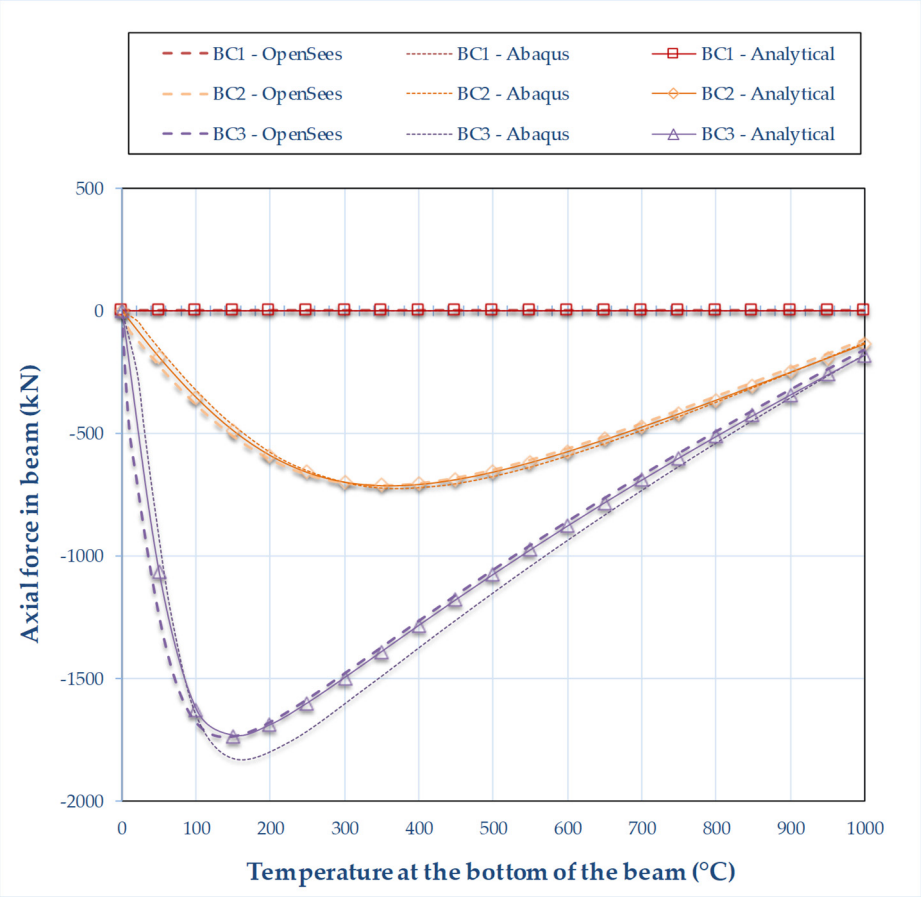


Figure 5.11. Axial force in the beam with boundary conditions BC1 - BC3, using OpenSees, ABAQUS, and the mathematical solutions.

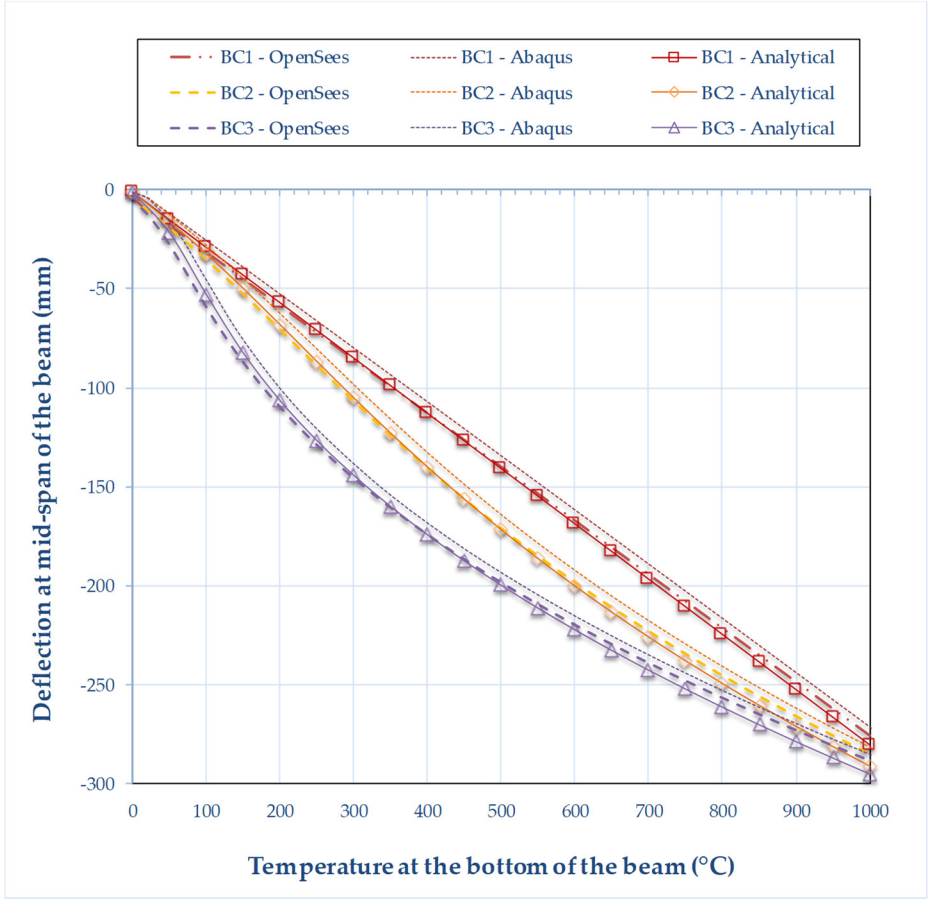


Figure 5.12. Deflections at beam mid-span with boundary conditions BC1 - BC3, using OpenSees, ABAQUS, and the mathematical solutions.

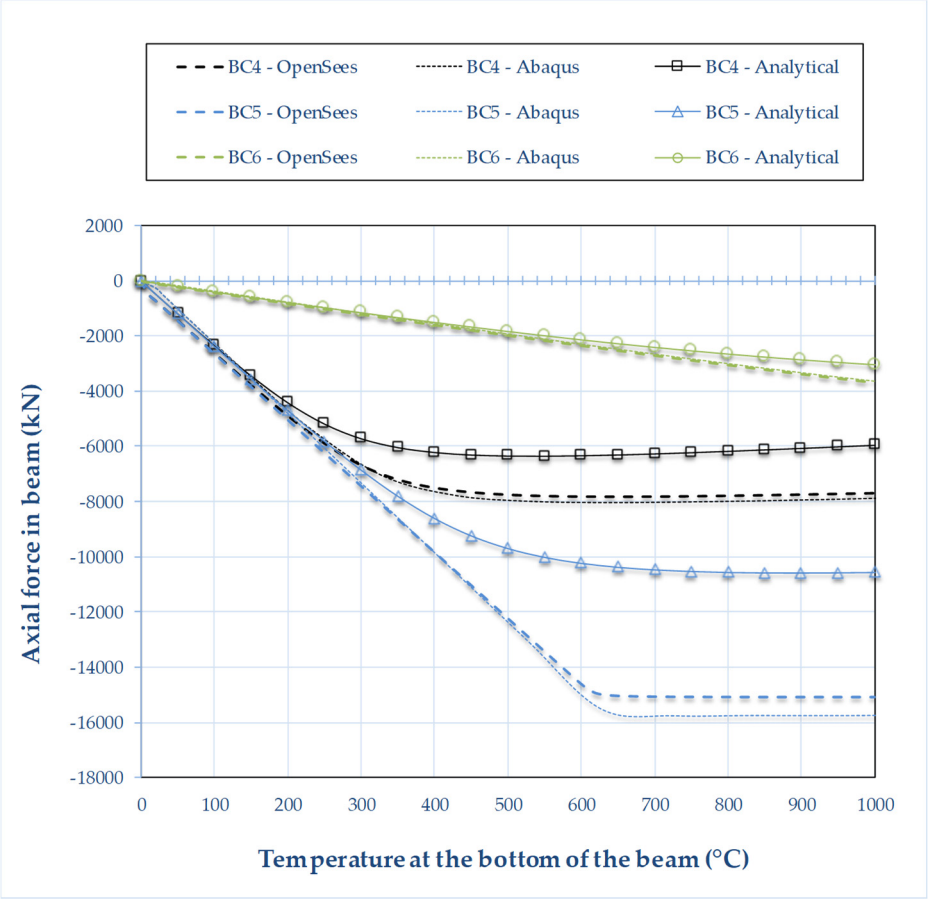


Figure 5.13. Axial force in the beam with boundary conditions BC4 - BC6, using OpenSees, ABAQUS, and the mathematical solutions.

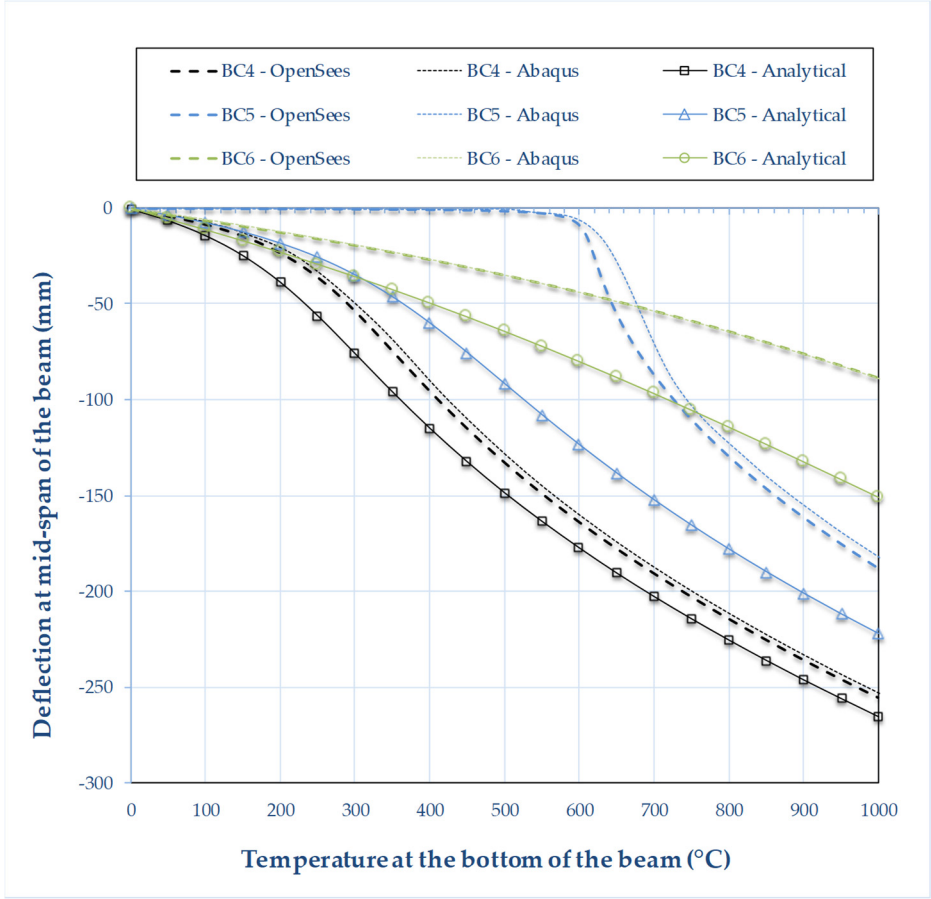


Figure 5.14. Deflections at beam mid-span with boundary conditions BC4 - BC6, using OpenSees, ABAQUS, and the mathematical solutions.

Compare with BC3, two rotational springs are added at the beam two ends for BC4. Due to the presence of these two rotational springs, the axial force in BC4 is much larger than BC3 and no obvious beam buckling is observed. Moreover, the beam deflects less in BC4 compared with BC3 (see Figure 5.12 and Figure 5.14). For example, at 200 °C, the beam deflection at BC3 is around 100 mm, but the deflection at BC4 is around 40 mm. These are due to the rotational springs, which would generate a certain level of hogging bending moment (depending of the stiffness of the rotational springs), when the beam is trying to deflect due to the thermal gradient. Hence this bending moment would enhance the beam which can bear larger axial forces without beam buckling, and also cancel the thermal induced large deflections. BC5 stands for another extreme case when the stiffness of the rotational restraints increases to infinity. The value of deflection in BC5 is less than BC4, experiencing with even larger axial force. Moreover, an axial force plateau is observed in Figure 5.13 for BC5, and a clear sign of beam buckling can be seen in Figure 5.14, at around 600 °C. It implies that this 'infinite stiffness' of the rotational springs in BC5 increases the beam buckling temperature from 150 °C at BC3 to 600 °C at BC5, and also cancels a certain amount of deflections due to the hogging bending moment. BC6 is the 'moderate' case among all the other boundary conditions (i.e. BC1 - BC5).

As shown in Figure 5.11 and Figure 5.12 for the boundary cases BC1 - BC3, there is a good agreement of the beam mid-span deflection and axial force results, obtained using OpenSees, ABAQUS, and the mathematical solutions. However, for the boundary conditions BC4 - BC6 which involve with the rotational restraints, the modelling results from OpenSees and ABAQUS agree well, but the analysis results from the mathematical solutions start to deviate. This deviation becomes clearer when larger stiffness of the rotational springs is used[6].

Nevertheless, good agreement between OpenSees and ABAQUS for all the boundary conditions (i.e. BC1-BC6), has shown the reliability of OpenSees, or SIFBuilder, which is capable of handling displacement-based beam column element (considering the

[6] It is important to note that the development of the mathematical model-based solutions is still ongoing, by the time when the author wrote this thesis.

geometrical nonlinearity at large deflections) and the simple material model (considering thermal expansion) under different boundary conditions with acceptable accuracy.

Finally, the benchmark problems presented in this section can be further used for verification of other software packages, or mathematically-based analytical solutions. Therefore, the relevant Tcl scripts for the OpenSees model, and the input files (.inp) for ABAQUS model are all attached in Appendix A.2.

5.3.2 A Cantilever Beam under Standard Fire Curve

To further validate OpenSees displacement-based beam column element (considering the geometrical nonlinearity at large deflections), and the material model (considering both thermal expansion and material degradation), a cantilever beam under standard fire curve with a concentrated mechanical load at the beam end is investigated. Moreover, this benchmark problem is built up using the integrated tool (i.e. SIFBuilder), rather than using the original 'Thermal version of OpenSees'. In addition, SAFIR is used for validating the results from SIFBuilder.

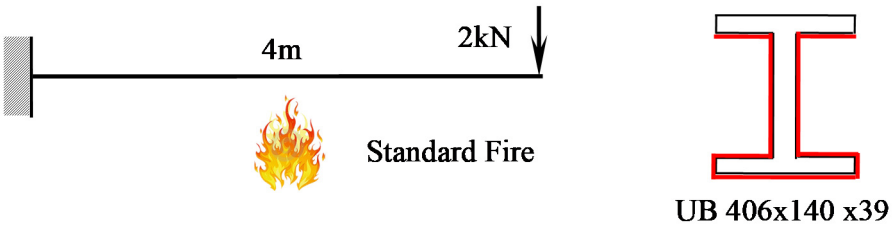


Figure 5.15. A cantilever beam under the standard fire with three sides fire exposure.

A 4 m long cantilever beam is subjected to a 3-sided exposure standard fire (ISO-834) for 1800 s, as shown in Figure 5.15. A carbon steel material which is associated with an I-section (UB 406x140x39), is adopted as the material of the beam. The yield stress and elasticity modulus at 20 °C are 3.55×10^8 N/m^2 and 200×10^9 N/m^2 respectively. The initial thermal expansion coefficient is assumed as 12×10^{-6} /°C. The properties of the material are taken to be temperature dependent according to Eurocode 3 [12]. Apart from the applied thermal loading, the self-weight of the beam and a 2 kN downward concentrated load are also applied. The details of this example are shown in Figure 5.15.

Figure 5.16 shows the temperature histories over the beam depth at top flange, mid-web, and bottom flange respectively. These results are obtained through the heat transfer analysis using SIFBuilder and SAFIR. It can be seen that the modelling results from both software agree well. The thermal gradient over the beam depth is induced by 3-sided fire exposure, which stands for a common situation when a concrete slab is above the steel beam with the heat sink effect.

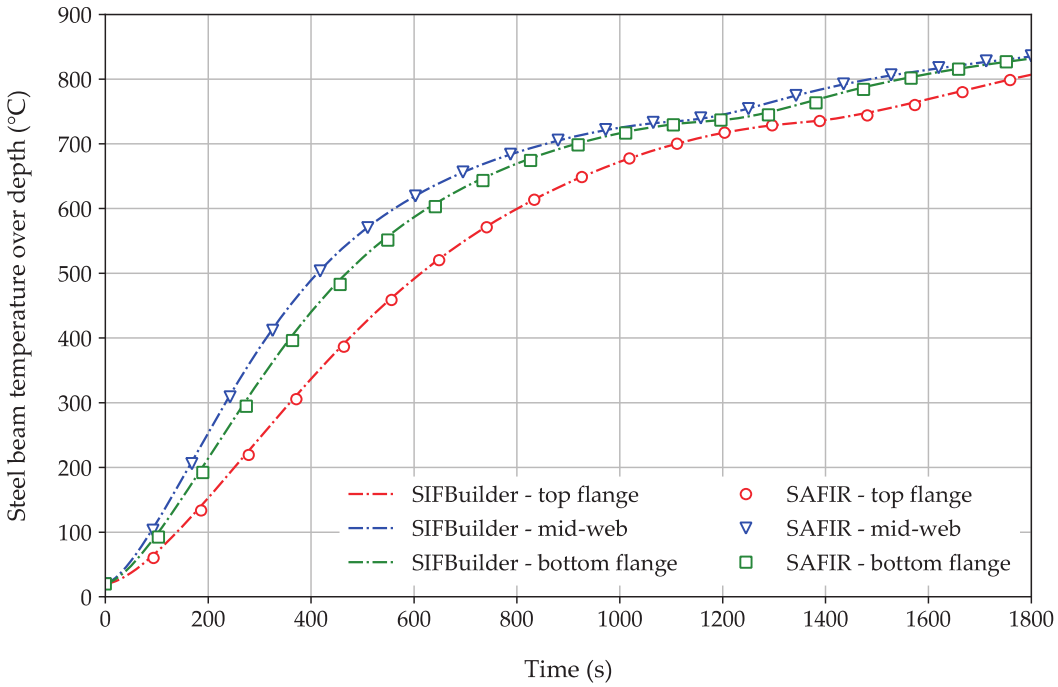


Figure 5.16. Heat transfer results comparison between SIFBuilder and SAFIR, for the temperature distribution over the beam depth.

Figure 5.17 demonstrates the vertical displacement histories of the beam free end, without applying the member self-weight, using SIFBuilder and SAFIR respectively. The reason of not applying the member self-weight is because in SIFBuilder the self-weight loading is applied on the structural member as a UDL automatically, depending on the user-defined beam member size and corresponding material density. To verify SIFBuilder in a step-by-step manner, this automatic self-weight function must be decoupled first, to check the structural response only under a concentrated mechanical load combined with 1800 s standard fire curve.

According to Figure 5.17, it can be seen that the modelling results agree well between SAFIR and SIFBuilder, when SteelECThermal material model is used. The Steel01Thermal material model was developed in the early development stage of the 'Thermal' version of OpenSees [9]. It can be regarded as a 'transitional' material model in the OpenSees material libraries. The Steel01Thermal_Princeton material model was developed based on the Steel01Thermal model, through incorporating better modelling capabilities for the material at unloading and cooling phases [33].

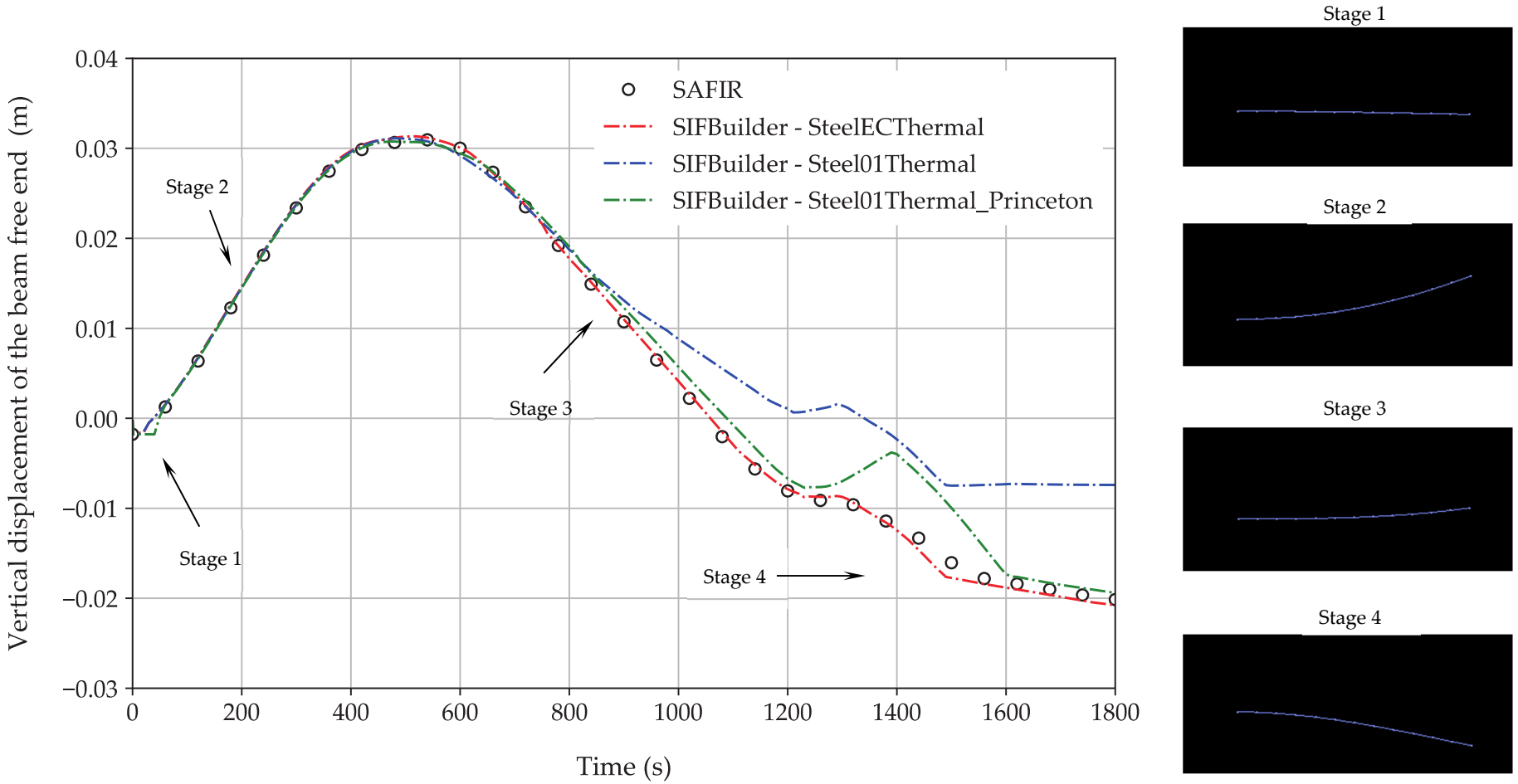


Figure 5.17. Vertical displacement of the beam free end, using SIFBuilder (with different material models) and SAFIR, without applying member self-weight; accompanying with four screenshots during SIFBuilder modelling.

However, in this benchmark problem which is under the pure standard fire heating, Steel01Thermal_Princeton does not show good modelling performance at higher temperatures (e.g. 750 ˚C at 1400 s), compared with the results from SAFIR and SIFBuilder (with SteelECThermal), as shown in Figure 5.17. This is probably because the Steel01Thermal_Princeton model inherits some existing problems from the original Steel01Thermal material model[7].

Furthermore, there are four typical stages of the behaviour of this cantilever beam in this benchmark problem, as shown in Figure 5.17 with four screenshots during SIFBuilder modelling. Stage 1 is the initial beam deflection due to the concentrated load at beam free end at ambient temperature. Then stage 2 launches represented by the beam right end moving upward due to the large thermal gradient, which is generated by the three-sided standard fire exposure (see Figure 5.16). In stage 3, the beam right end starts to move downward as the steel material loses half of its strength at around 500 ˚C at 500 s, under the concentrated loading at beam free end. Finally, in stage 4 the cantilever beam continues to deflect and exceeds its original mechanical load-induced deflection.

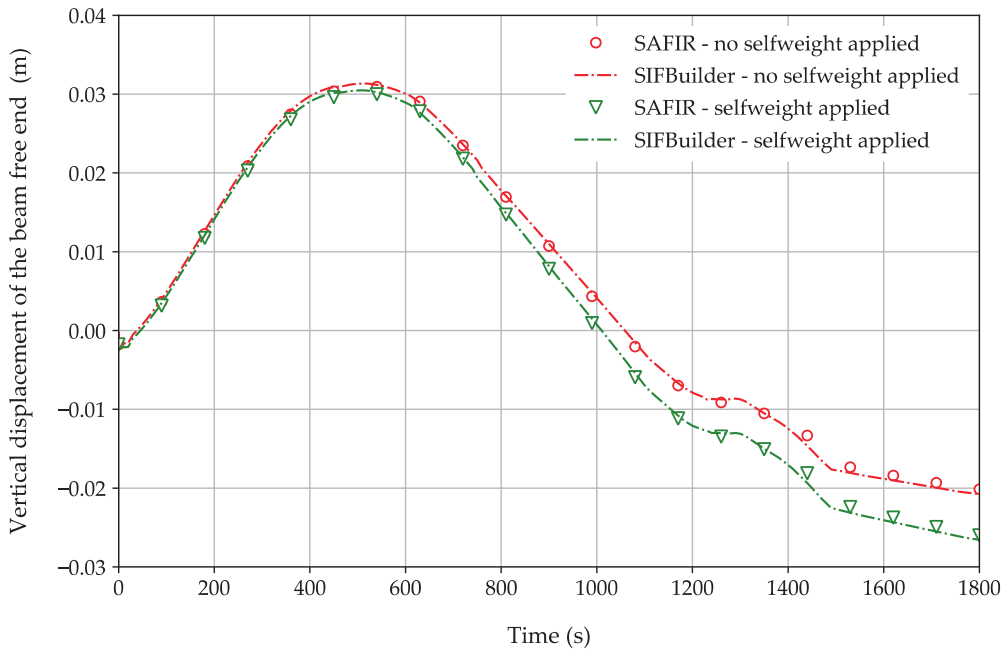


Figure 5.18. Vertical displacement of the beam free end, using SIFBuilder (SteelECThermal material model) and SAFIR, with and without applying member self-weight.

[7] It is worth to note that the development of improved material modelling capabilities for OpenSees/SIFBuilder is still ongoing, by the time when the author wrote this thesis.

Figure 5.18 demonstrates the vertical displacement histories of the beam free end, with and without applying the member self-weight, using SIFBuilder and SAFIR respectively. It can be seen that the modelling results agree well between SAFIR and SIFBuilder when the SteelECThermal material model is used. Hence, the automatic applying self-weight function in SIFBuilder is also verified in this benchmark problem[8].

[8] For thoroughly verifying and validating SIFBuilder, more benchmark problems are carried out. These additional benchmark problems are attached in the Appendix A.3 as a set of SIFBuilder Tcl scripts.

5.3.3 A Simply-supported Beam under Standard Fire Test

Section 5.3.1 and 5.3.2 demonstrate the verifications and validations of OpenSees/ SIFBuilder with a set of benchmark problems, spanning from the displacement-based beam column element (geometrical nonlinearity at large deflections), the material model (thermal expansion and material degradations), mechanical loadings (concentrated load and UDL), thermal loadings (through depth thermal gradient and three-sided standard fire exposure), and finite stiffness of end restraints, against mathematically-based analytical solutions, SAFIR, and ABAQUS.

This section presents a round robin modelling exercise lead by the University of Edinburgh, in parallel to a simply-supported steel beam subjected to a standard fire in a furnace, which was carried out by SP Technical Research Institute of Sweden [34]. OpenSees and ABAQUS are used to perform this modelling task[9]. There are two stages in this round robin modelling exercise. The difference between stage 1 and stage 2 is that for the second stage of the exercise the material model is modified to set the yield stress at 447.5 MPa, and the thermal loading is modified based on the supplied temperature history in the steel section based on the test data from SP.

The test beam is an HEB 300 steel beam with simply-supported boundary conditions. The geometry, loading and support conditions of the beam are shown in Figure 5.19.

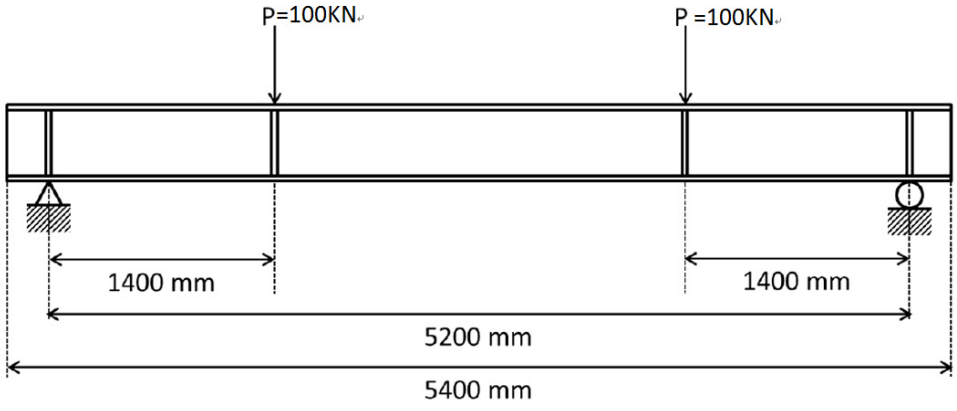


Figure 5.19. Configuration of the modelled steel beam.

[9] It is important to note that this modelling task was collaborated with another two colleagues, who are Jiayu Hu and Liming Jiang. The author was in charge of 3D ABAQUS modelling and drafting the final report. Jiayu Hu was in charge of 2D OpenSees modelling, and Liming Jiang provided the heat transfer results using OpenSees.

5.3.3.1 The FEM model setup for OpenSees and ABAQUS at stage 1

The analysis carried out in OpenSees is based on using beam elements to simulate the global response of the tested beam to the standard fire. The ABAQUS analysis uses shell elements in order to ensure that any possible local buckling deformations are accounted for. The ABAQUS shell element, S4R5, is employed and the mesh used for the cross-section is shown in Figure 5.20. As for the longitudinal direction of the beam, the length (5400 mm) is discretised into 270 elements along both flanges and the web. Therefore, the flange elements are 30 mm × 20 mm × 19 mm; and the web elements are 28.1 mm × 20 mm × 11mm. The OpenSees model consists of 54 *dispBeamColumn2dThermal* fibre-based elements, with the beam section modelled using 12 fibres (2 each for the flange and 8 for the web) as shown in Figure 5.21. Figure 5.22shows the full 3D mesh of the HEB 300 steel beam as analysed in ABAQUS.

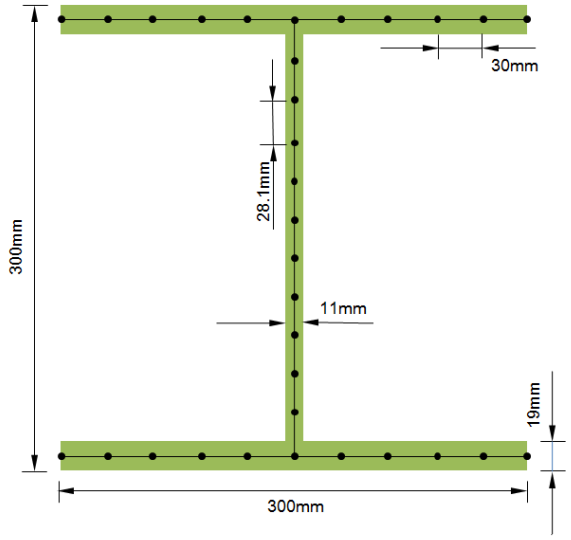


Figure 5.20. Mesh of the cross section in ABAQUS.

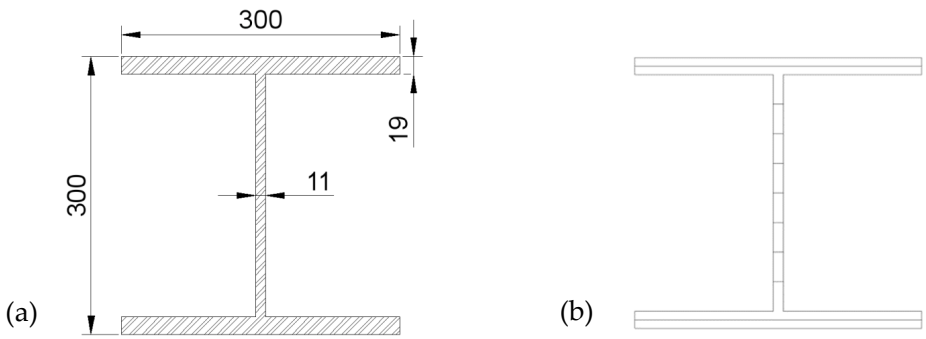


Figure 5.21. Dimensions (a) and fibres distribution (b) of steel section in OpenSees.

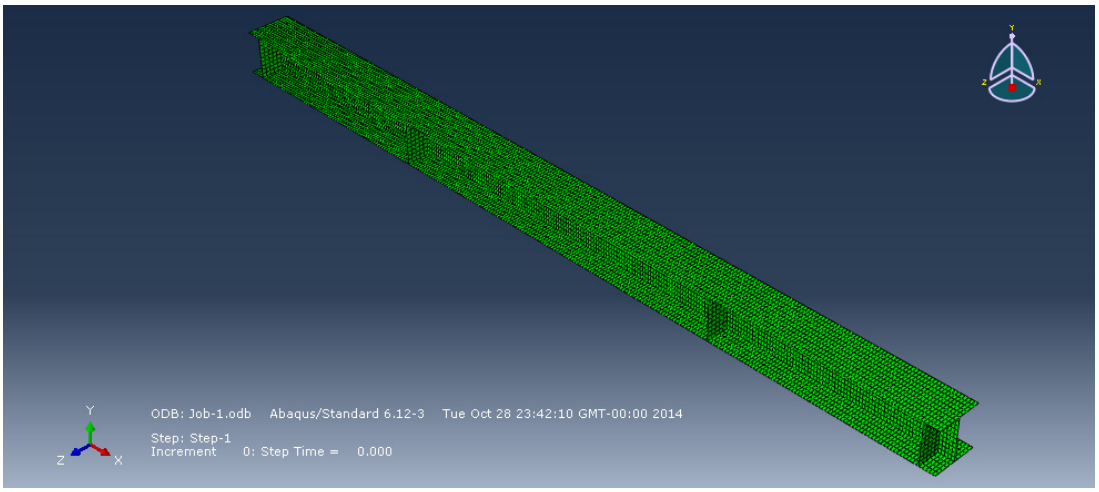


Figure 5.22. 3D mesh of HEB 300 beam in ABAQUS, using S4R5 shell element.

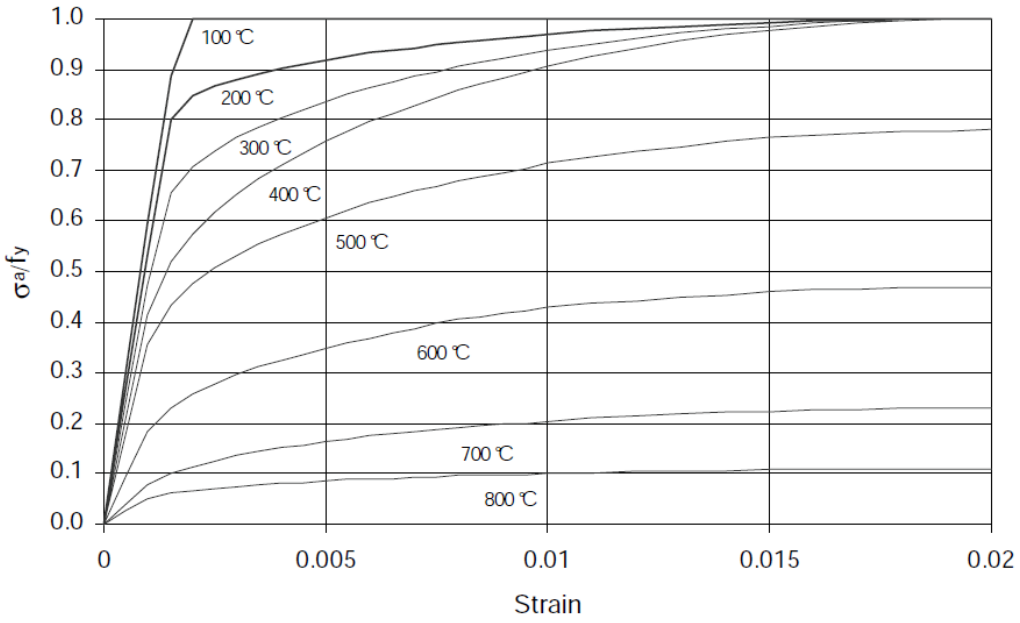


Figure 5.23. Variation of stress-strain relationship under temperatures for grade S355 steel of the HEB 300 beam.

Both OpenSees and ABAQUS use Eurocode 3 based temperature-dependent material properties [12] for steel as shown in Figure 5.23. The self-weight of the beam is applied as a UDL and the point loads are applied as indicated in the schematic supplied by SP shown in Figure 5.19. A heat transfer analysis of the HEB 300 steel beam subjected to a 3-hour standard fire with all sides exposed except the top of the upper flange was carried out using OpenSees in order to obtain the evolution of temperature in the beam. The time-temperature history along the section height is shown in Figure 5.24. The thermal loading

in the OpenSees and ABAQUS models are applied by interpolating the temperatures over the steel section assuming all the sections (in the longitudinal direction) have identical temperature evolutions.

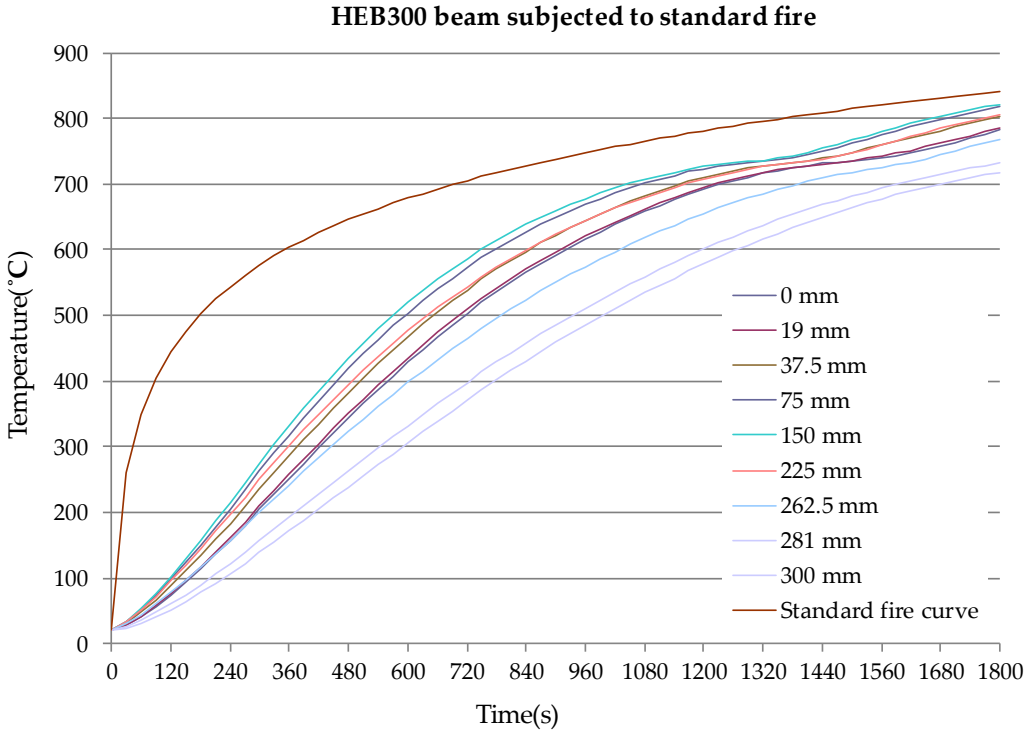


Figure 5.24. Time-temperature histories through the beam cross-section depth (heat transfer analysis performed by OpenSees).

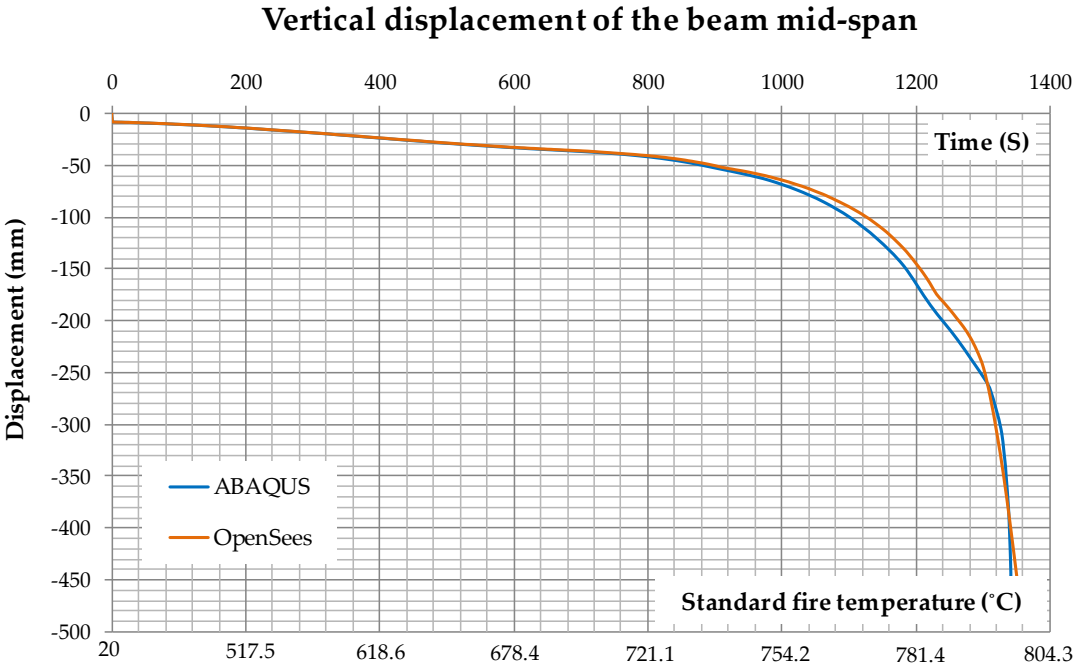


Figure 5.25. Beam mid-span deflection histories, using OpenSees and ABAQUS.

5.3.3.2 The modelling results from OpenSees and ABAQUS at stage 1

The deflection history of the beam at mid-span during standard fire exposure is shown in Figure 5.25. Both the OpenSees and ABAQUS modelling results show nearly identical behaviour with runaway failure occurring around 1270 seconds. The standard fire temperatures corresponding to the time are shown at the bottom axis of Figure 5.25. Figure 5.26 shows the final deflected shape of the beam from ABAQUS at 1340 s, and the corresponding stress distribution of the entire beam is shown in Figure 5.27.

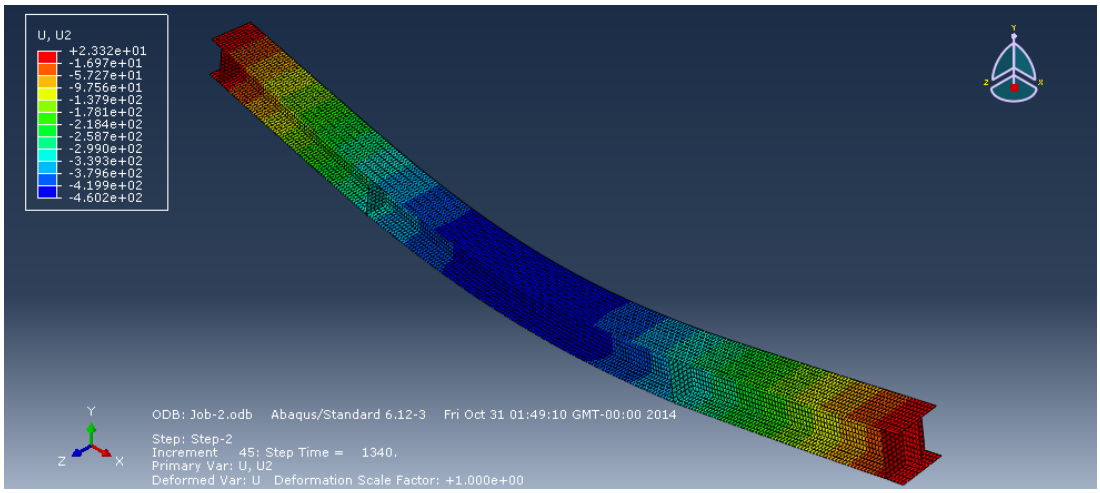


Figure 5.26. Final deflected shape of the beam from ABAQUS at 1340 s.

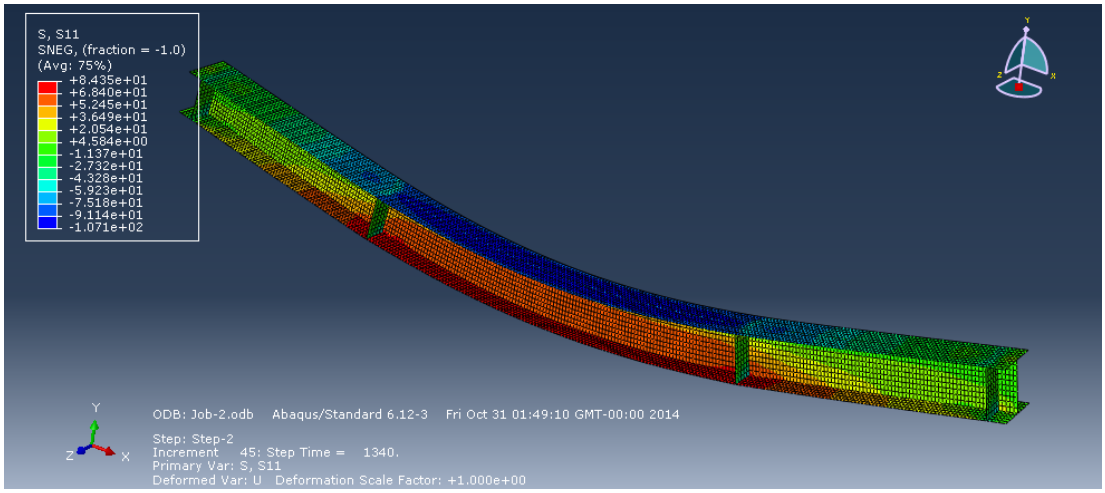


Figure 5.27. Stress distribution of the beam from ABAQUS at 1340 s.

The stress profile evolution over cross-section height at the beam mid-span using ABAQUS is shown in Figure 5.28. It can be seen that the stress is linearly distributed over the section height at the initial state, where the upper half is in compression and lower

half in tension due to the applied mechanical loads. From 0 s to 421 s, most of the cross-section area are distributed with compressive stress, especially at the web of the beam, which experiences higher temperatures, inducing larger thermal expansions. After this time period, the cross-sectional stress gradually shifts from compression to tension due to the increasing material softening, which further triggers the beam runaway failure, as shown at 1340 s at final state.

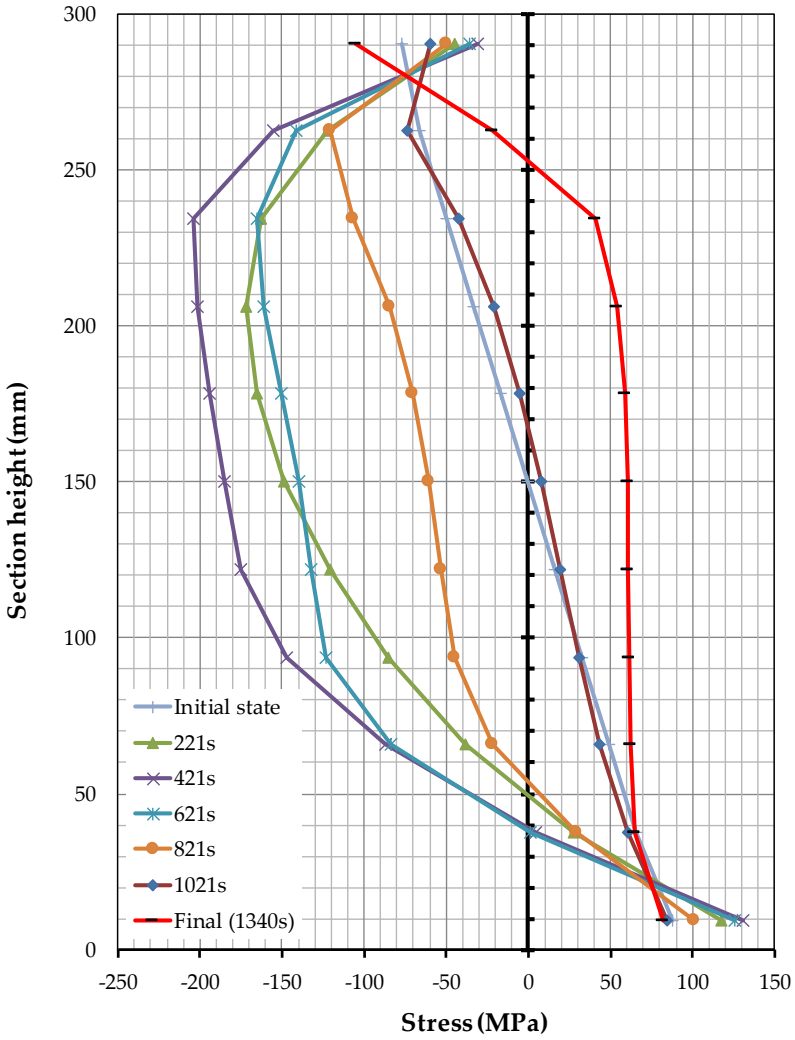


Figure 5.28. Stress profile evolution over cross-section height at the beam mid-span using ABAQUS.

Similar plots are also generated according to OpenSees modelling results, which are shown in Figure 5.29. It can be seen that both the ABAQUS and OpenSees present very similar stress shifting history, either qualitatively or quantitatively.

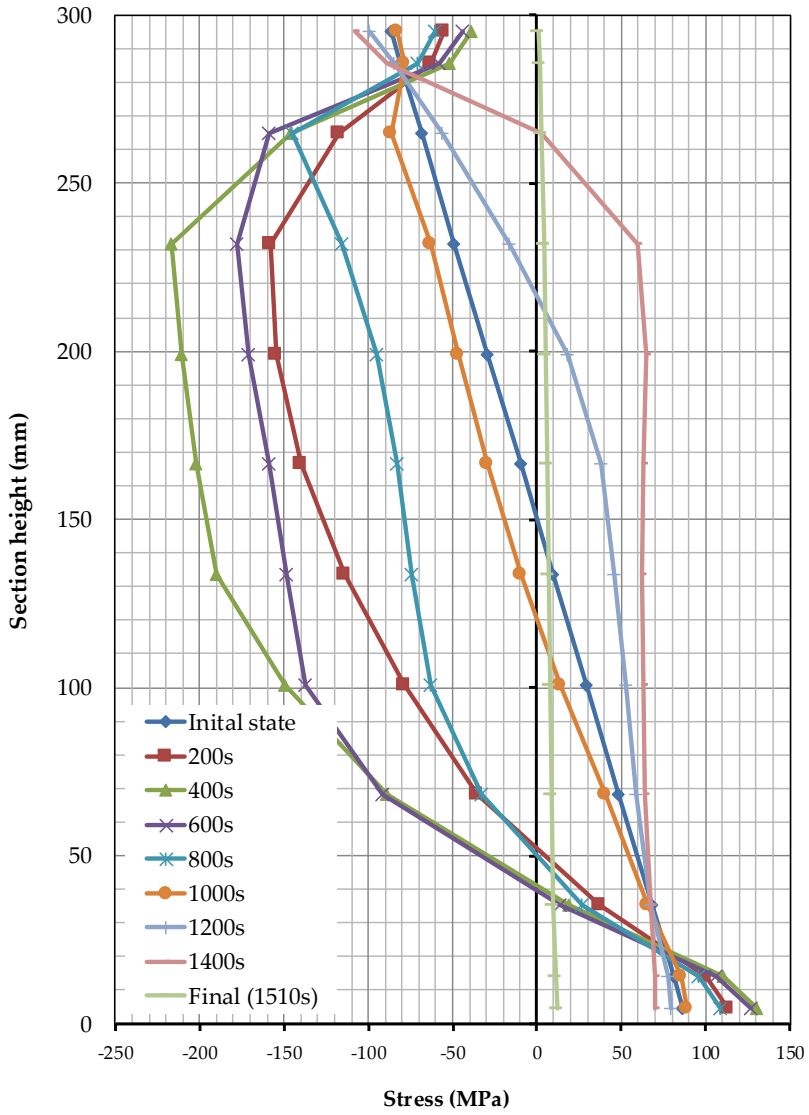


Figure 5.29. Stress profile evolution over cross-section height at the beam mid-span using OpenSees.

5.3.3.3 The modelling results from OpenSees and ABAQUS at stage 2

For the second stage of the modelling exercise, the material model is modified to set the steel yield stress at 447.5 MPa, and the thermal loading is changed according to the measured temperature history over the beam depth from SP. Furthermore, another ABAQUS analysis is performed considering the strain-hardening of the steel based on Eurocode 3. The temperature-dependent material properties of steel used in the modelling exercise are demonstrated in Figure 5.30 and Figure 5.31.

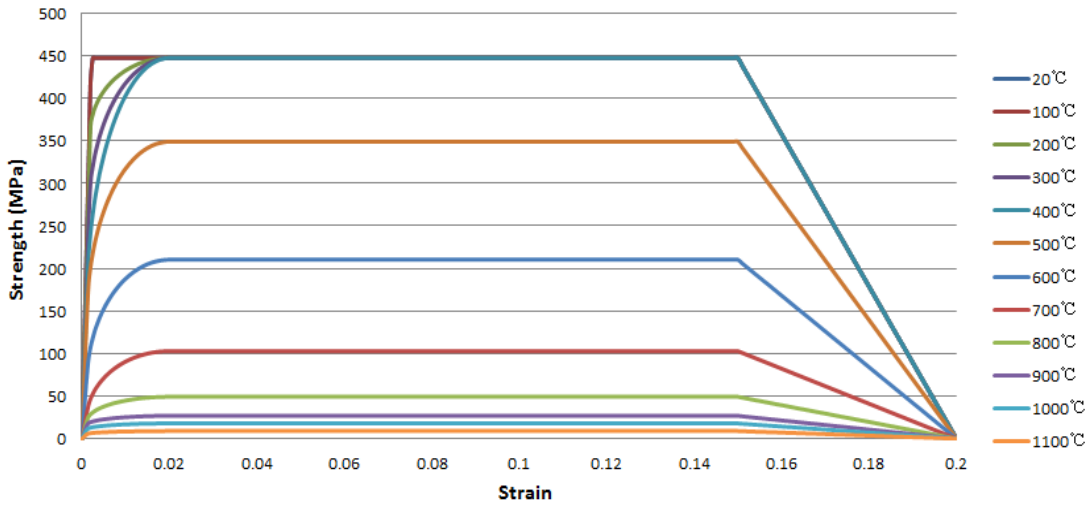


Figure 5.30. Variation of stress-strain relationship with temperature for 447.5 MPa steel (without strain-hardening).

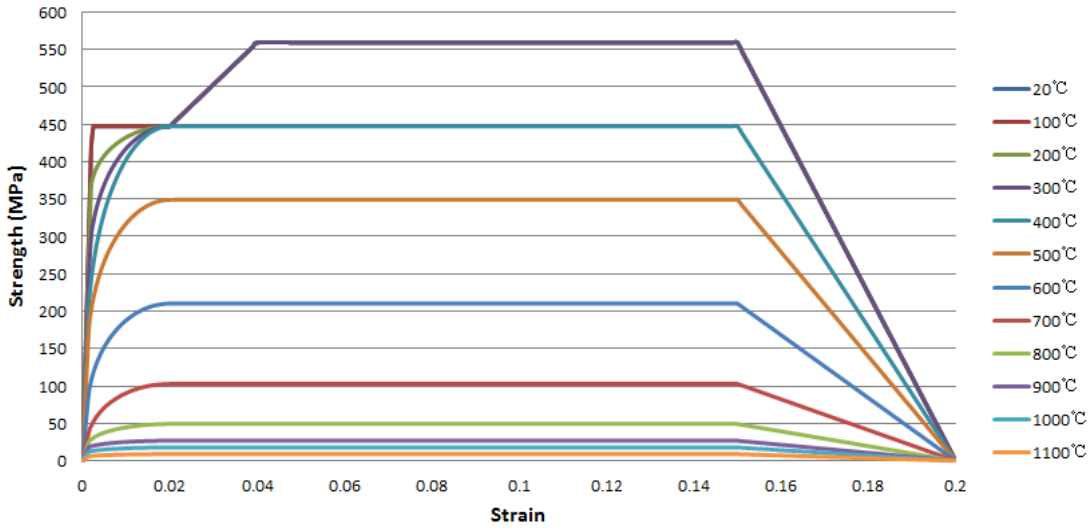


Figure 5.31. Variation of stress-strain relationship with temperature for 447.5 MPa steel (with strain-hardening).

The measured time-temperature history over the beam depth from SP, is used as the thermal loading input for both ABAQUS and OpenSees analyses (see Figure 5.32). For simplicity, the top flange and bottom flange temperatures are assumed to be the average of the values from measured location 1, 2, and measured location 4, 5, respectively. The temperature at the beam web is set to the temperature evolution provided from measured location 3.

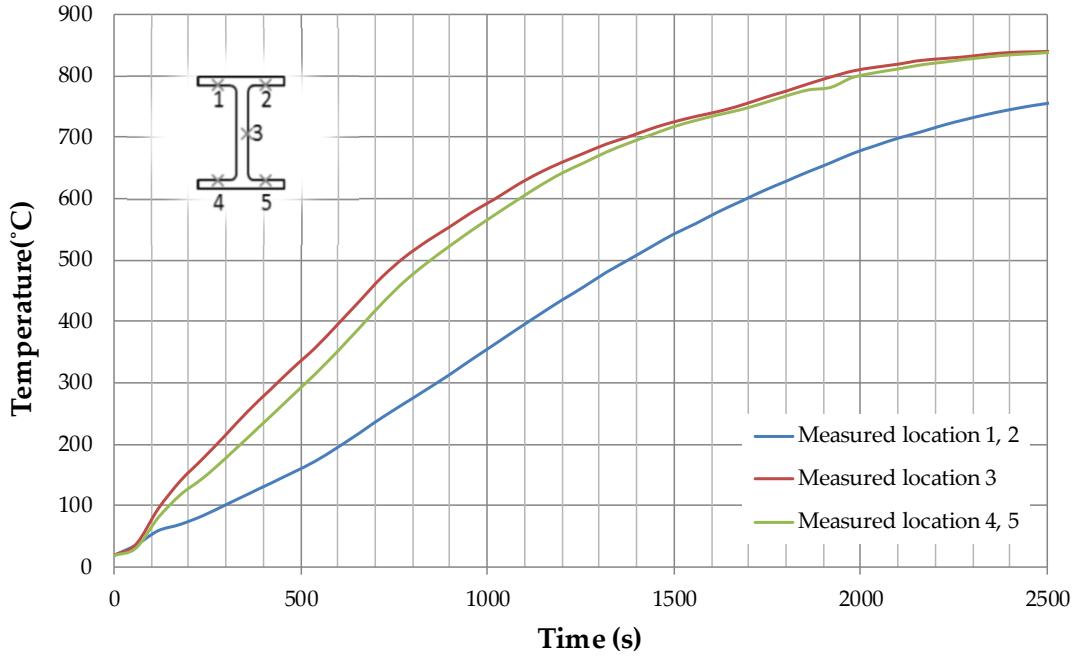


Figure 5.32. The measured time-temperature history over the beam depth, provided by SP.

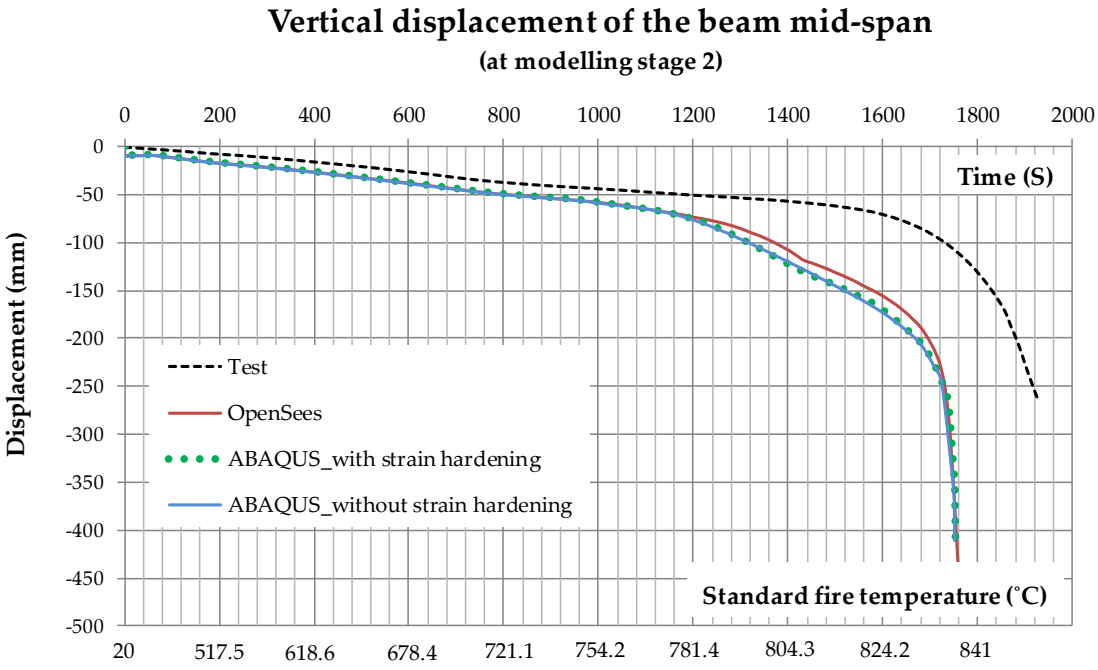


Figure 5.33. Beam mid-span deflection histories, modelled by OpenSees and ABAQUS (with and without strain hardening), and compared against the test data provided by SP.

At stage 2, the deflection history of the beam at mid-span during standard fire exposure is presented in Figure 5.33. Both the OpenSees and ABAQUS models (even the model considering strain hardening) show nearly identical behaviour, with runaway failure occurring around 1680 seconds. These modelling results are also compared against the test data provided by SP, which shows clear 'delayed' deflection histories.

Nevertheless, OpenSees still shows a good modelling performance against ABAQUS and other modelling approaches, according to Figure 5.34, which is the deflection histories reported by SP, where number 0 stands for the test data, and the rest of the numbers stand for round robin modelling results from other research institutions using different approaches [34]. It can be seen that the OpenSees modelling results 'fall' into the majority of other modelling predictions.

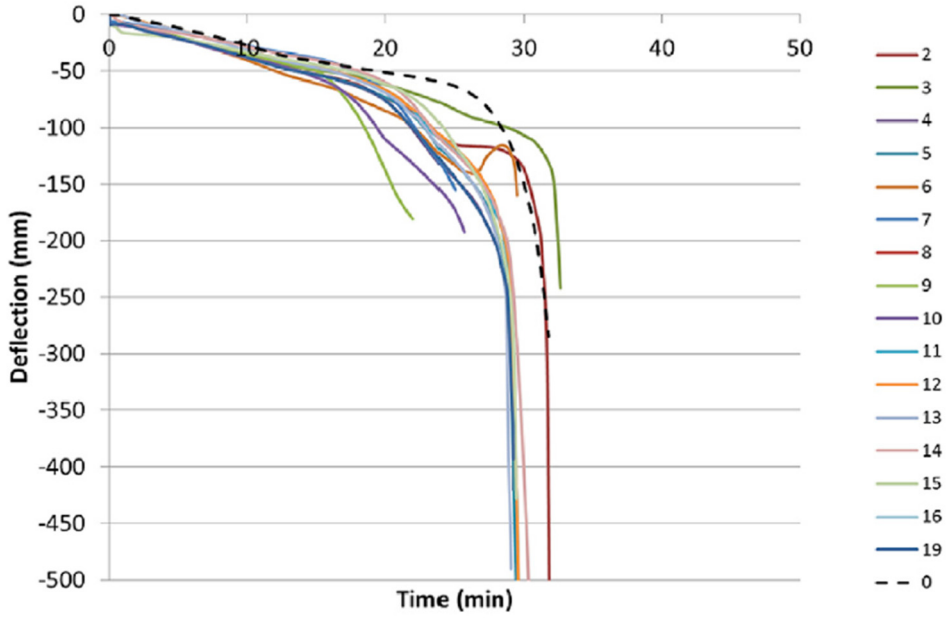


Figure 5.34. Deflection histories reported by SP, with the modelling results from various approaches (NO.2-NO.19), against the test data (NO. 0) [34].

5.3.4 A Modelling Benchmark for Cooling

To further validate the modelling capability and accuracy of OpenSees steel material model during the cooling phase, another benchmark problem published by Gillie [32] is re-analysed using OpenSees. The definition of this benchmark problem is illustrated in Figure 5.35. The investigated beam is 1 m long with cross-section dimension 35 mm x 35 mm. The beam is restrained by two pinned supports at both ends, under a UDL with magnitude of 4250 N/m. A linear temperature increase (from 0 ˚C to 800 ˚C) is first applied on the beam then linearly cooled again. The material temperature dependency used in this benchmark problem is also presented in Figure 5.35.

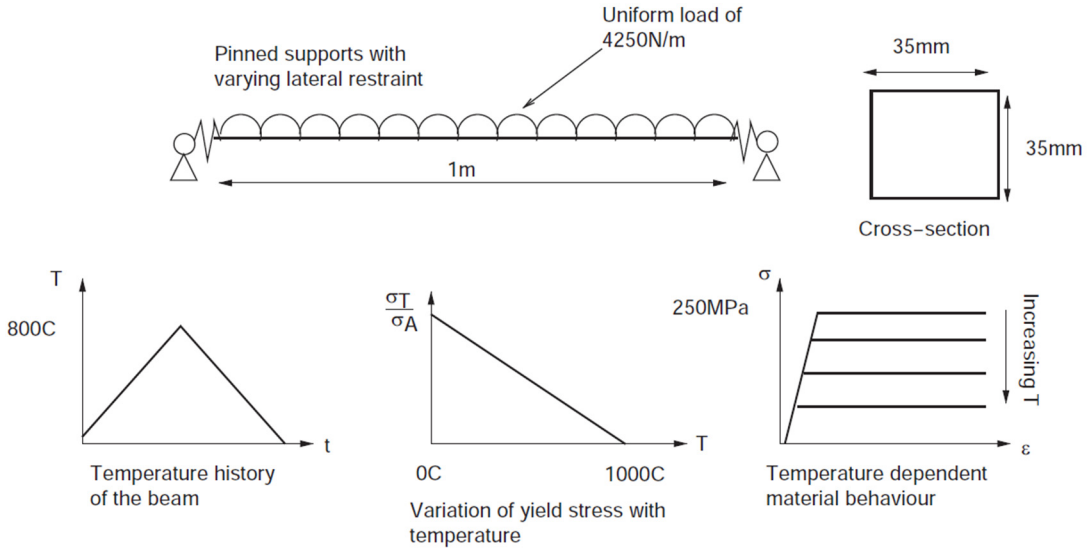


Figure 5.35. Benchmark problem definition by Gillie [32].

As illustrated in the previous sections, the temperature dependency of steel material model in OpenSees is mainly based on Eurocode 3 [12]. Hence, Steel01Thermal material model is adapted to fit the material properties defined in Figure 5.35. The details of the OpenSees model setup for this benchmark problem is attached in Appendix A.4 as a Tcl script. Figure 5.36 and Figure 5.37 are the beam axial force and mid-span deflection histories during the heating and cooling phase, using OpenSees with adapted Steel01Thermal material model compared against the original modelling results from Gillie.

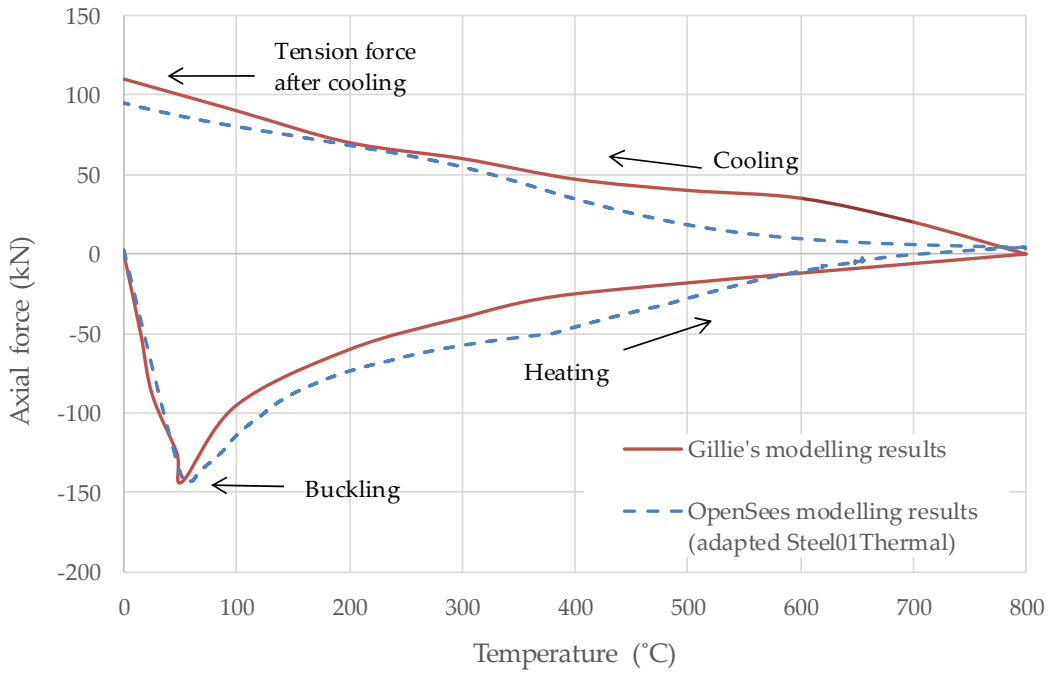


Figure 5.36. Axial force evolution during the heating and cooling phase using OpenSees, compared against original modelling results from Gillie.

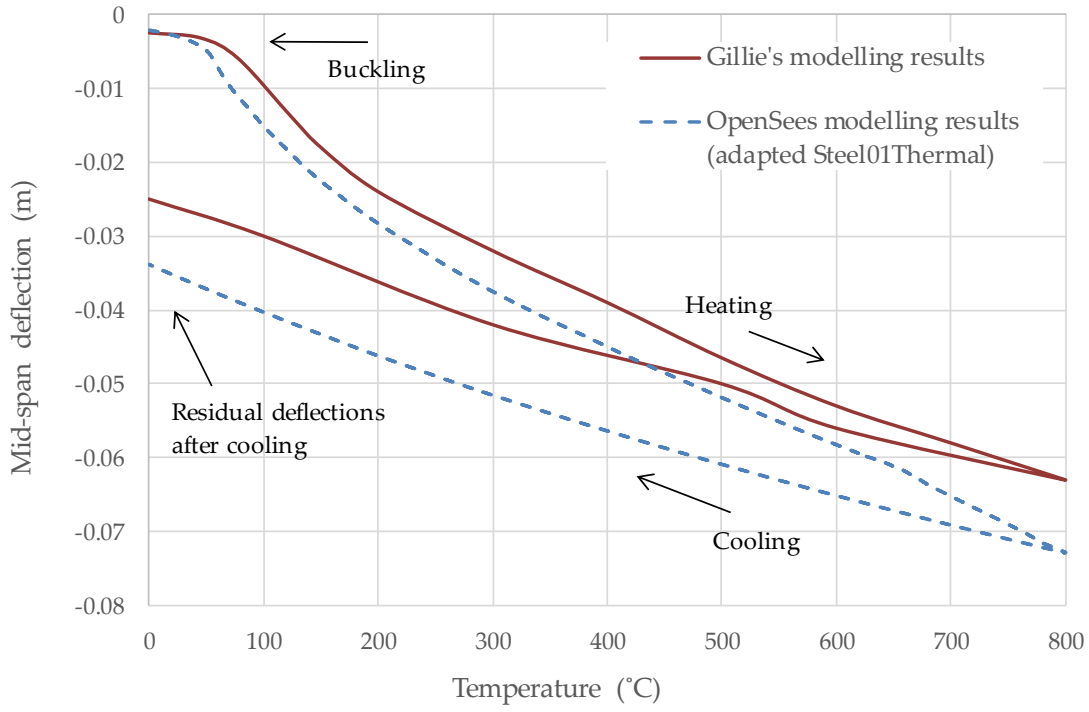


Figure 5.37. Mid-span deflection history during the heating and cooling phase using OpenSees, compared against original modelling results from Gillie.

In general, the axial force evolution shown in Figure 5.36 suggests good agreement between the OpenSees modelling results and Gillie's results. However, the mid-span deflection history shown in Figure 5.37 demonstrate a considerable difference between the two modelling approaches.

To be more specific, the beam axial force first starts to build up in compression, as the beam is pinned at both ends which constrains the thermally-induced expansion. Following that the beam gets buckled under -150 kN compressive force at around 60 °C. Hence the axial force gets released, and the beam steps into the mechanical unloading phase, in which the mid-span deflection modelled by OpenSees starts to deviate compared with the modelling results from Gillie. It suggests that the limitations of Steel01Thermal material model is due to the mechanical unloading rather than the increasing temperatures, as in this benchmark problem the deviation starts at a relative low temperature (at around 60 °C) which has not 'trigger' the material strength degradation yet, according to the Eurocode 3 [12]. In addition, similar deviation during the buckling-induced mechanical unloading phase has also been shown in the previous section (see Figure 5.11 - Figure 5.14), where the material degradation are not even considered in these benchmark problems (although the difference between OpenSees and ABAQUS modelling results is not apparent).

Actually, this modelling problem was also reported by the research group from Princeton University [33], after the work carried out by the author presented in this section. Their research group provided an improved material model (Steel01Thermal_Princeton). However, this model is also not perfect, which has been demonstrated in section 5.3.2. Therefore, a possible solution to improve the steel material model in OpenSees is adapting the unloading and cooling modelling capabilities from Steel01Thermal_Princeton to the SteelECThermal material model which shows good modelling performance during heating phase, but this model cannot handle cooling so far.

Nevertheless, according to Figure 5.36 and Figure 5.37, the beam axial force and mid-span deflection histories using OpenSees still generate very similar trends compared with the modelling results from Gillie, with capturing the beam buckling during heating phase, axial force reverses from compression to tension during cooling phase, and the residual

deflections and tensile forces after the beam cools back to the ambient temperature. It implies that under such situations the modelling results from OpenSees are still 'qualitatively' useful.

5.4 CONCLUSIONS

A comprehensive structures-in-fire analysis tool (i.e. SIFBuilder) is presented in this chapter, which can potentially revolutionise the way of research in the academia and the way of design in the industry, for the performance-based structural fire engineering. It is developed through integrating the modelling capabilities of fire analysis, heat transfer, and thermal-mechanical analysis in a single software platform, with an automatic data transmission manner to highly improve the structural fire modelling efficiency. Furthermore, SIFBuilder is entirely free to be used and improved by interested research users and developers due to its open source nature.

A series of benchmark problems are presented in this chapter, for further verifying and validating OpenSees/SIFBuilder under different situations, spanning from the displacement-based beam column element (geometrical nonlinearity at large deflections), the material model (thermal expansion and material degradations), mechanical loadings (concentrated load and UDL), thermal loadings (through depth thermal gradient, three-sided standard fire exposure, uniformly heating and cooling), and finite stiffness of end restraints, which are tested against analytical solutions, ABAQUS, SAFIR, and experimental data. It is found that the structural fire modelling capabilities of OpenSees using displacement-based beam column element with an appropriate steel material model can generate reliable modelling results in most situations, except for the buckling-induced unloading phase and cooling phase. However, even under such situations the modelling results from OpenSees are still 'qualitatively' useful.

5.5 REFERENCES

[1] W. Grosshandler, Ed., "Fire Resistance Determination and Performance Prediction Research Needs Workshop: Proceedings, NISTIR 6890," 2002.

[2] J. Franssen and T. Gernay, "Modeling structures in fire with SAFIR: theoretical background and capabilities," *Journal of Structural Fire Engineering*, vol. 8, no. 3, pp. 300–323, 2017.

[3] "Vulcan Software Package," *Vulcan Solutions Ltd.*, 2005. [Online]. Available: www.vulcan-solutions.com. [Accessed: 04-Sep-2017].

[4] B. Izzuddin, "Nonlinear dynamic analysis of framed structures," PhD Thesis, Department of Civil Engineering, Imperial College London, 1991.

[5] F. T. McKenna, "Object-Oriented Finite Element Programming: Frameworks for Analysis, Algorithms and Parallel Computing," PhD Thesis, University of California, 1997.

[6] A. Usmani, J. Zhang, J. Jiang, Y. Jiang, and I. May, "Using OpenSees for structures in fire," *Journal of Structural Fire Engineering*, vol. 3, no. 1, pp. 57–70, 2012.

[7] Y. Jiang, "Development and Application of A Thermal Analysis Framework in OpenSees for Structures in Fire," https://www.era.lib.ed.ac.uk/handle/1842/7941, PhD Thesis, School of Engineering, University of Edinburgh, 2012.

[8] J. Jiang, "Nonlinear Thermomechanical Analysis of Structures Using OpenSees," https://www.era.lib.ed.ac.uk/handle/1842/7749, PhD Thesis, School of Engineering, University of Edinburgh, 2012.

[9] J. Zhang, "Developing OpenSees Software Framework for Modelling Structures in Fire," PhD Thesis, School of Engineering, University of Edinburgh, 2014.

[10] L. Jiang, X. Dai, A. Usmani, and P. Kamath, "OpenSees-based integrated tool for modelling structures in fire," in *The First International Conference on Structural Safety under Fire & Blast, Glasgow, Scotland, UK*, 2015, pp. 461–468.

[11] B. B. Welch, K. Jones, and J. Hobbs, *Practical Programming in Tcl and TK*, 4th Ed. Prentice Hall, 2003.

[12] European Standard EN 1993-1-2, "Eurocode 3: Design of steel structures - Part 1-2: General rules - Structural fire design." CEN, Brussels, 2005.

[13] European Standard EN 1992-1-2, "Eurocode 2: Design of concrete structures - Part 1-2: General rules - Structural fire design." CEN, Brussels, 2004.

[14] "OpenSees for Fire - Main Website (Edinburgh Univristy)," 2016. [Online]. Available: https://www.wiki.ed.ac.uk/display/opensees/For+Users. [Accessed: 01-Nov-2017].

[15] "OpenSees for Fire - Main Website (HongKong Polytechnic Univristy)," 2017. [Online]. Available: http://openseesforfire.github.io/. [Accessed: 01-Nov-2017].

[16] L. Jiang, "Development of An Integrated Computational Tool for Modelling Structural Frames in Fire Considering Local Effects," https://www.era.lib.ed.ac.uk/handle/1842/19563, PhD Thesis, School of Engineering, University of Edinburgh, 2016.

[17] "Structural fire protection." Vol.78, American Society of Civil Engineers, 1992.

[18] European Standard EN 1991-1-2, "Eurocode 1: Actions on structures - Part 1-2: General actions - Actions on structures exposed to fire." CEN, Brussels, 2002.

[19] Y. Hasemi, Y. Yokobayashi, T. Wakamatsu, and A. V. Ptchelintsev, "Modeling of heating mechanism and thermal response of structural components exposed to localized fires: a new application of diffusion flame modeling to fire safety engineering," in *Thirteenth meeting of the UJNR planel on fire research and safety*, 1996, pp. 237–247.

[20] B. Y. Lattimer, "Heat fluxes from fires to surfaces," in *SFPE Handbook of Fire Protection Engineering*, Third Ed., National Fire Protection Association, 2002.

[21] "Standard Test Methods for Fire Tests of Building Construction and Materials, ASTM E119." American Society for Testing and Materials, West Conshohocken, PA, 2016.

[22] "BS EN 1363-1: Fire resistance tests, Part 1: General Requirements." European Commission, 2012.

[23] "Structural Fire Engineering Investigation of Broadgate Phase 8 Fire," The Steel Construction Institute, Berkshire, 1991.

[24] A. S. Usmani, J. M. Rotter, S. Lamont, A. M. Sanad, and M. Gillie, "Fundamental principles of structural behaviour under thermal effects," *Fire Safety Journal*, vol. 36, no. 8, pp. 721–744, 2001.

[25] T. C. H. Liu, M. K. Fahad, and J. M. Davies, "Experimental investigation of behaviour of axially restrained steel beams in fire," *Journal of Constructional Steel Research*, vol. 58, no. 9, pp. 1211–1230, 2002.

[26] Y. Z. Yin and Y. C. Wang, "A numerical study of large deflection behaviour of restrained steel beams at elevated temperatures," *Journal of Constructional Steel Research*, vol. 60, no. 7, pp. 1029–1047, 2004.

[27] Y. Z. Yin and Y. C. Wang, "Analysis of catenary action in steel beams using a simplified hand calculation method , Part 1◎: theory and validation for uniform temperature distribution," *Journal of Constructional Steel Research*, vol. 61, no. 2, pp. 183–211, 2005.

[28] Y. Z. Yin and Y. C. Wang, "Analysis of catenary action in steel beams using a simplified hand calculation method , Part 2◎: validation for non-uniform temperature distribution," *Journal of Constructional Steel Research*, vol. 61, no. 2, pp. 213–234, 2005.

[29] K. Tan and Z. Huang, "Structural Responses of Axially Restrained Steel Beams with Semirigid Moment Connection in Fire," *Journal of Structural Engineering*, vol. 131, no. 4, pp. 541–551, 2005.

[30] Z. Huang and K. Tan, "Fire resistance of compartments within a high-rise steel frame: New sub-frame and isolated member models," *Journal of Constructional Steel Research*, vol. 62, no. 10, pp. 974–986, 2006.

[31] V. K. R. Kodur and M. M. S. Dwaikat, "Response of steel beam-columns exposed to fire," *Engineering Structures*, vol. 31, no. 2, pp. 369–379, 2009.

[32] M. Gillie, "Analysis of heated structures: Nature and modelling benchmarks," *Fire Safety Journal*, vol. 44, no. 5, pp. 673–680, Jul. 2009.

[33] N. E. Khorasani, M. E. M. Garlock, and S. E. Quiel, "Modeling steel structures in OpenSees: Enhancements for fire and multi-hazard probabilistic analyses," *Computers and Structures*, vol. 157, pp. 218–231, 2015.

[34] D. Lange and L. Boström, "A round robin study on modelling the fire resistance of a loaded steel beam," *Fire Safety Journal*, vol. 92, pp. 64–76, 2017.

[35] P. Khazaeinejad, X. Dai, and A. Usmani, "Analysis of heated beams: modelling benchmarks," in *the First International Conference on Structural Safety under Fire & Blast, Glasgow, Scotland, UK*, 2015, pp. 469–473.

Chapter 6.

Implementation of Fire Models in an OpenSees-based Integrated Tool: SIFBuilder

6.1 INTRODUCTION

In the modern routine design of structures, computational modelling of the structural behaviour under natural and man-made hazards has become more and more important. The capabilities of analysing the structural performance under such hazards (e.g. snow, wind, earthquake, impact) have been widely utilized into the mainstream of nonlinear finite element method (NFEM) based software, such as SAP, ANSYS, ABAQUS etc. However, there are very limited software options to characterize fire impact on structures. In general, there are two types of computer programs for simulating structural behaviours in fire: research-oriented and commercially-oriented. The former such as SAFIR [1], VULCAN [2], and ADAPTIC [3] address specific modelling problems, because of a limited number of users and a small team of developers. These codes typically suffer from tightly bound architecture as a result of using procedural programming[1]. Furthermore, because they are often developed by a small dedicated team of researchers at the original host institution, they are not designed or suited for a devolved community of developers, and the codes are not open source [4]. They also typically have uncertain resourcing and great dependency on key individuals for support, maintenance and development. The latter such as ABAQUS, ANSYS and LS-DYNA are used commercially by researchers and industry across the world. Nevertheless, limited access to source codes, lack of transparency of the computational framework, high cost of purchase and maintenance are major limitations.

In 2009, OpenSees was adopted at the University of Edinburgh for further development to enable it to perform structural fire analysis. OpenSees is a C++ based open source software framework initially developed at the University of California, Berkeley by McKenna in the late 1990s [5]. Its original application for providing an advanced finite-element simulation tool to perform structural and geotechnical analysis under seismic loadings. It has now become a common platform, for researchers within the Pacific Earthquake Engineering Research Center (PEER, which is a multi-institutional research

[1] Procedural programming is a programming language paradigm designed to focus on procedures which gather input, process input data with series of functions, and deliver output. It offers ease and transparency for less complicated applications (popular languages such as C, Fortran, and Pascal).

and education centre in US), for the development, sharing, and dissemination of new ideas to further earthquake engineering research around the world. Three key references, i.e. the OpenSees user manual [6], the OpenSees main website [7], and McKenna's PhD thesis [5], have reached citation numbers to 764, 898, and 330 respectively, according to the latest records from Google Scholar in September 2017.

Facilitated by the nature of the OpenSees code, being open-source, a large number of thermal capabilities have been added to the framework by Usmani *et al.* [8]. Significant contributions in terms of heat transfer and fire modules have been made to the framework in developing the 'Thermal' version of OpenSees [9]. After verifying and validating the thermo-mechanical analysis of OpenSees [10, 11], users are able to model structures under extreme thermal actions (such as those resulting from fire conditions) through defining arbitrary non-uniform temperature distributions across and along an element. In 2014, an integrated tool named Structures in Fire Builder (abbrev. as SIFBuilder), was developed based on the original 'Thermal' version of OpenSees [12]. SIFBuilder aims to perform automated thermo-mechanical analysis for large structures under realistic fire scenarios, through integrating the fire modelling, heat transfer analysis, and thermo-mechanical analysis into a single software with 'minimum user input'. Currently, structural engineers do not usually perform structural fire analysis on large buildings considering various fire scenarios, but typically apply only code-based regulations for fire protection. This is mainly because of the limited availability of the numerical tools, and high time expense for performing such structural fire simulations. Nevertheless, SIFBuilder's 'minimum user input' strategy facilitates structural engineers to model and perform a quick structural fire analysis for a range of fire scenarios, such as standard fire, parametric fire, localised fire, etc. [13].

However, the available fire model options in the current version of SIFBuilder are still limited [12]. These options are either uniform fire models, assuming the same temperature distribution in the entire compartment with a time-temperature relationship (e.g. nominal fire curves [13]), or localised fire models, using correlational equations between incidental heat fluxes on the structural surfaces and radial distance from the fire source (e.g. Hasemi localised fire model [13]). However, in reality, the above-mentioned design fire models cannot guarantee the reliability of the structural fire design. Well

known examples of failures include the World Trade Center Towers in New York City in 2001 [14], Windsor Tower in Madrid in 2005 [15], the Faculty of TU Delft Architecture building in Netherlands in 2008 [16], and more recently the Plasco building in Tehran in 2017 [17]. These buildings were all designed according to the relevant structural fire design codes, but still experienced collapse or partial collapse during the fire challenge. Hence, in order to produce more optimised and robust structural fire design using SIFBuilder, more realistic design fire methodologies are needed to be implemented, validated, and investigated.

This chapter first reviews the existing fire models in OpenSees, including uniform compartment fire models and localised fire models. Secondly, the latest development of advanced fire models, i.e. smoke zone models [18] and travelling fire models [19] are also introduced, with their implementation details in SIFBuilder using C++. Finally, a case study using the zone models in OpenSees is investigated, for validating results against the original ASET-QB and FIRM-QB software packages [18].

6.2 A REVIEW OF EXISTING FIRE MODELS IN SIFBUILDER

Conventional structural fire design codes are based on isolated single structural members with simply-supported boundary conditions under standard fire test exposures, which refers to a heating curve such as ISO-834 standard fire [20], or ASTM-E119 fire [21]. The standard fire curve along with external fire curve, and hydrocarbon fire curve are categorized as "nominal fire curves" in the Eurocode [13, 22]. This type of fire curve is basically a time-temperature relationship, which stands for the case of a fully developed fire in a compartment (see Figure 6.1). All the above-mentioned fire curves are added in the OpenSees fire module as *NominalFireEC1* class[2] [9].

Different from the nominal fire curves, the parametric fire curves [13] which consider fire growth rate, fire load density, and characteristics of the compartment (e.g. thermal boundaries, openings, geometric quantities, etc.) are also added in OpenSees as *ParametricFireEC1* class [9].

[2] Class is a term in C++. It contains the definitions of the corresponding data and methods (also called functions).

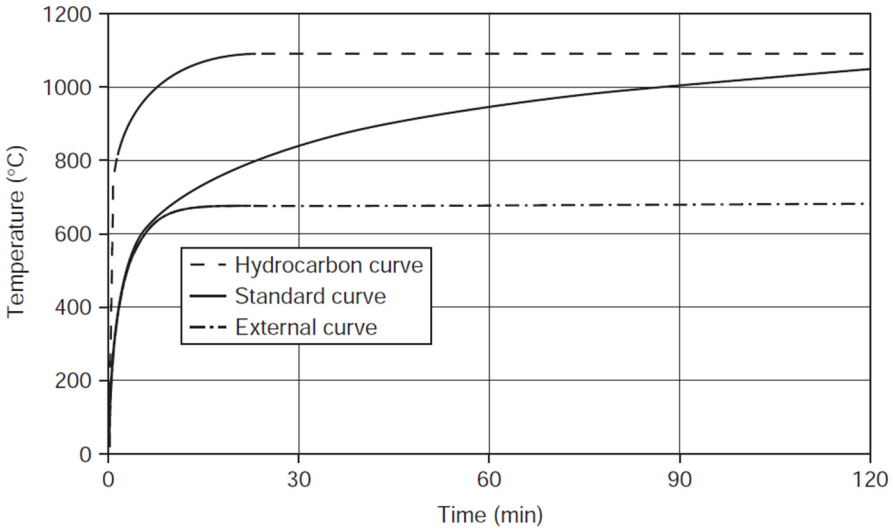


Figure 6.1. Three different nominal fire curves in Eurocode [22].

Although these fire models are relatively simple, they are still widely used for both research and design purposes in fire safety engineering. A user defined fire curve (*UserDefinedFire* class) [9] is also included in the OpenSees fire module for providing more flexibility. These idealised uniform fire models are all assumed to have the same temperature distribution in the entire compartment at a specific time.

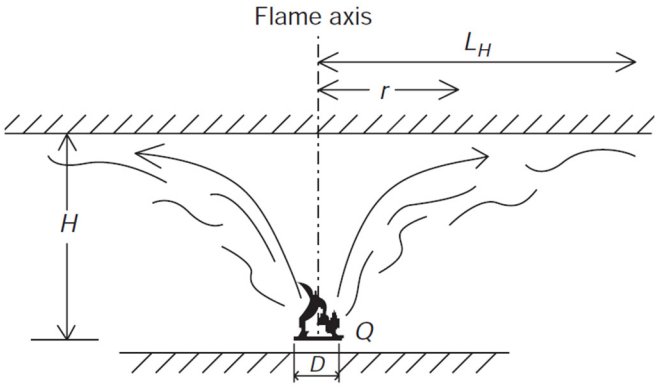


Figure 6.2. Localised fire model in Eurocode [22].

Further, in the case of isolated fuels burning in a large space (e.g. vehicles burning in a car park), localised fire models are regarded to be appropriate for simulating such burning scenarios. The Hasemi localised fire model (adopted in the Eurocode [13], see Figure 6.2), Alpert ceiling jet model [23], SFPE handbook-based localised fire model [24], and a user defined idealised local fire model, are added in OpenSees fire module with class names *LocalizedFireEC1*, *AlpertCeilingJetModel*, *LocalizedFireSFPE*, *Idealised_Local_Fire*

respectively [9, 25]. These localised fire models, in their mathematical nature, are all correlational equations between incidental heat fluxes on the structural surfaces and radial distance from the fire source.

6.3 ZONE MODELS IN SIFBUILDER

In a prescriptive structural fire design code, the fire exposure is usually constrained by the code with limited room for discussion (e.g. nominal fire curves in Eurocode). In a performance-based structural fire design code, the practitioners have greater flexibility and the fire is usually related to realistic fire loadings (e.g. localised fire models, zone models, and travelling fire models, etc.). The newly added two zone model classes, *ZoneModel_ASET*, *ZoneModel_FIRM*, enable the fire module in OpenSees to simulate the transient generation of a hot smoke layer upon the compartment ceiling. The transient height and temperature of the smoke layer are calculated according to a set of ordinary differential equations (ODEs) based on the mass and energy conservations [18].

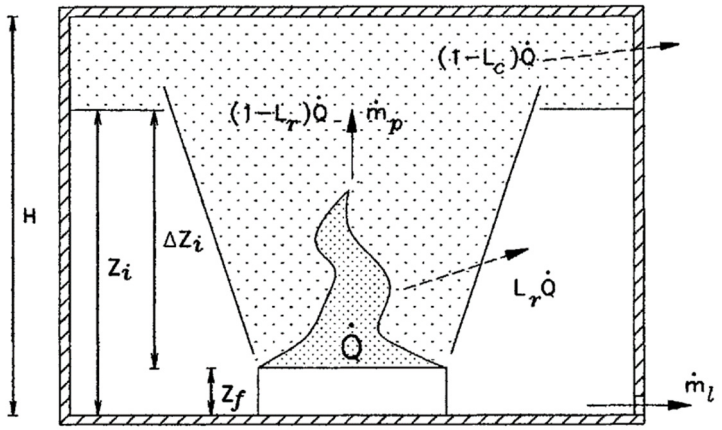


Figure 6.3. Fire problem modelled using ASET zone model [18].

The difference between *ZoneModel_ASET* and *ZoneModel_FIRM* is that, *ZoneModel_FIRM* can handle the compartment with vertical natural ventilations, however *ZoneModel_ASET* is basically filling the smoke in a 'box' without considering any significant openings (see Figure 6.3 and Figure 6.4). Hence, the *ZoneModel_FIRM* is more representative for solving practical engineering problems, and it is the first fire model which is fully documented, validated, verified, and evaluated following the ASTM

guidelines back in 2000 [18, 26]. Nevertheless, *ZoneModel_ASET* is still very useful for benchmarking simple zone modelling problems.

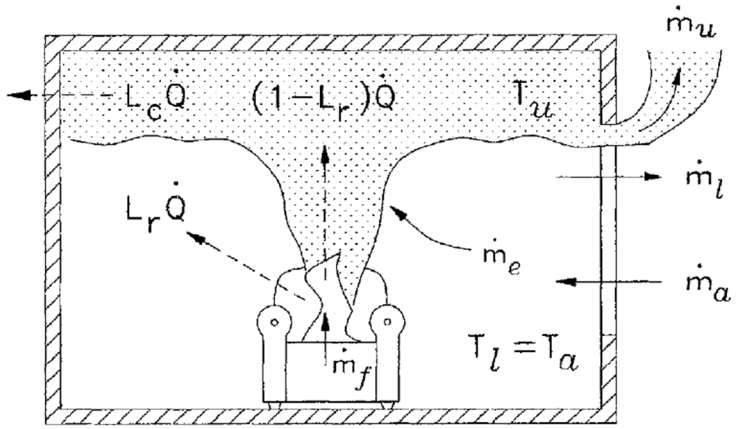


Figure 6.4. Fire problem modelled using FIRM zone model [18].

6.3.1 Mathematical formulations of the FIRM zone model

Two time-varying outputs are represented through the FIRM zone model, one being the transient upper smoke layer temperature, T_u (K), and the other one the evolution of the smoke layer interface height, Z_i (m). The determination of these two variables are via solving a set of ordinary differential equations (ODEs) based on the mass and energy conservations, such as the mass conservation of the lower ambient air [18]:

$$\frac{dZ_i}{dt} = \frac{\dot{m}_a - \dot{m}_l - \dot{m}_e}{\rho_a A} \tag{6.1}$$

where t (s) is the time, $\dot{m}_a$ (kg/s) is the vent flow rate of the ambient air entering the compartment, $\dot{m}_l$ (kg/s) is the lower layer vent flow rate leaving the compartment, $\dot{m}_e$ (kg/s) is the air entrainment mass flow rate, $\rho_a = 1.2$ (kg/m^3) is the density of the ambient air, and A (m^2) is the total compartment area.

Figure 6.5 schematically illustrates the mass balance of the design compartment. In addition to this figure, $\dot{m}_u$ (kg/s) is the smoke vent flow rate leaving the compartment, and H (m) is the clear height of the compartment. Another key ODE is about the upper smoke layer energy conservation [18], which is given as:

$$\frac{dT_u}{dt} = \frac{T_u\left[(1-L_c)\dot{Q} - \dot{m}_e c_p (T_u - T_a)\right]}{c_p \rho_a T_a A (H - Z_i)} \tag{6.2}$$

where $\dot{Q}$ (kW) is the heat release rate (HRR) of the fire, $T_a = 294.26$ (K) is the ambient air temperature, $c_p = 1.004$ (kJ/kgK) is the constant specific heat, L_c is the heat loss fraction ratio through the compartment boundaries (0.6 ~ 0.9 as recommended in [18]).

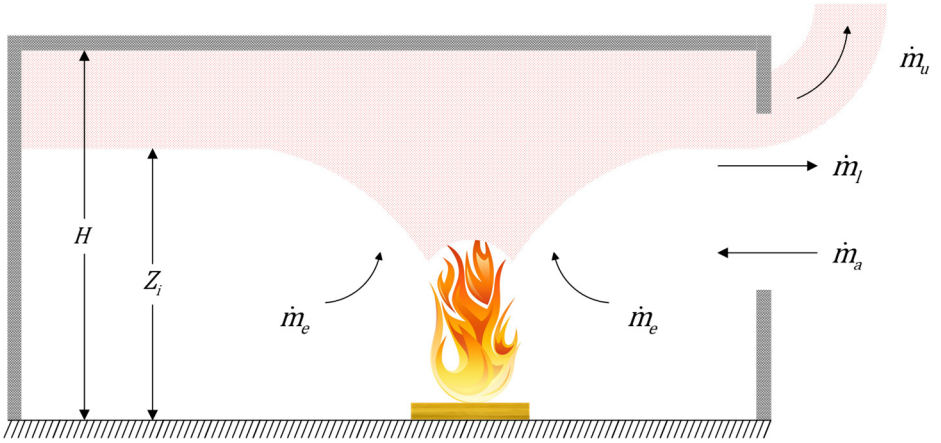


Figure 6.5. Schematic of the mass conservation of the FIRM zone model implemented in OpenSees.

Figure 6.6 schematically illustrates the energy balance of the design compartment. In addition to this figure, L_r is the radiative loss fraction of the fire plume (0.35 as recommended in [18]). It is worth noting that L_c and L_r are both empirical values but very fundamental to the resultant smoke layer temperature calculations.

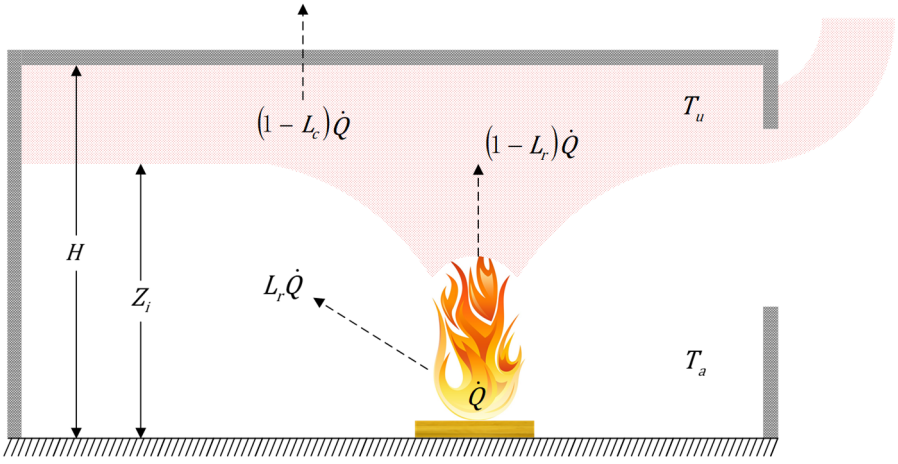


Figure 6.6. Schematic of the energy conservation of the FIRM zone model implemented in OpenSees.

There will be differences in terms of smoke temperature rises by using different air entrainment models to calculate $\dot{m}_e$, but these may be relatively minor [27]. Two air entrainment models can be selected in the current version of SIFBuilder. One is the Thomas model [28], which is widely used in the UK for venting calculations [29]:

$$\dot{m}_e = 0.188 W_{fi} (Z_i)^{3/2} \tag{6.3}$$

where W_{fi} (m) is the perimeter of the fire. The other one is Zukoski's model [30], which is given as:

$$\dot{m}_e = K(1 - L_r)^{1/3} \dot{Q}^{1/3} (\Delta Z_i)^{5/3} \tag{6.4}$$

where K = 0.076, ΔZ_i is the distance between the fuel top surface and the smoke layer interface. In the implementation of the FIRM zone model in the OpenSees fire module, the thickness of the fuel is ignored, hence $\Delta Z_i = Z_i$.

Furthermore, entrainment-controlled burning is included in the FIRM zone model, which means the upper bound values are assumed for the air mass flow rate $\dot{m}_e$ and corresponding HRR, $\dot{Q}$ [18]. Assuming Zukoski's plume model is employed, $\dot{m}_e$ is changed to:

$$(\dot{m}_e)_{max} = 55 K^{3/2} (1 - L_r)^{1/2} (\Delta Z_i)^{5/2} \tag{6.5}$$

and $\dot{Q}$ is changed to:

$$\left(\dot{Q}\right)_{max} = 3030 (\dot{m}_e)_{max} \tag{6.6}$$

Moreover, ventilation-controlled burning is also considered in the FIRM zone model. It is associated with a situation that the smoke layer interface drops to a certain height, causing the inflow of ambient air to reach its maximum under the ventilation-controlled burning [18]. This maximum value of the inflow of ambient air $\dot{m}_a$ is given as:

$$(\dot{m}_a)_{max} = 0.52 W_v (Z_t - Z_b)^{3/2} \tag{6.7}$$

where W_v (m) is the total vent widths, Z_t (m) is the average soffit height, Z_b (m) is the average sill height.

6.3.2 Vent flow regimes in the FIRM zone model

There are several typical vent flow regimes included in the *ZoneModel_FIRM* class according to [18], hereinafter referred to as: regime 1, regime 2, regime 2-3, regime 3, and regime 4. Regime 1 refers to the very early stage of the smoke accumulation beneath the ceiling, like a piston pushing down and expelling cool air to flow out of the compartment through the vents, as shown in Figure 6.7. During this regime, the pressure inside of the compartment, P_{in} (Pa), is larger than the pressure outside of the compartment, P_{out} (Pa).

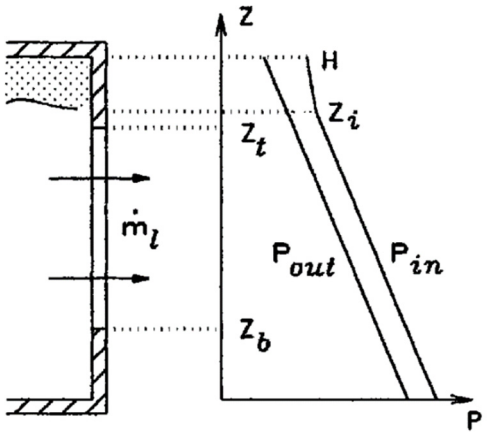


Figure 6.7. Flow regime 1 of the FIRM zone model implemented in the OpenSees fire module [18].

Regime 2 relates to a subsequent stage that the piston effect is still pushing down, but expelling both the hot smoke and cool air out of the compartment through the vents, at this regime the smoke layer interface is below the ceiling soffit, as shown in Figure 6.8. This regime is normally of short duration compared with other regimes.

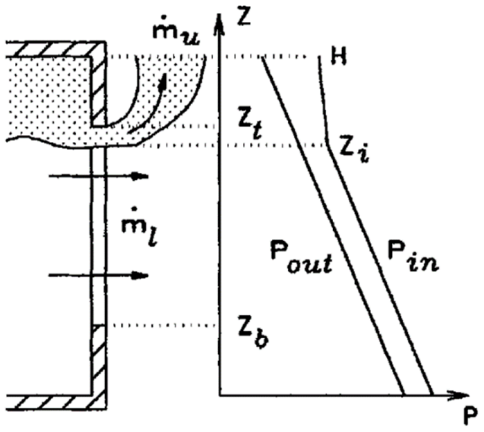


Figure 6.8. Flow regime 2 of the FIRM zone model implemented in the OpenSees fire module [18].

Regime 2-3 is a transit stage with a very short period of time, as the pressure inside of the compartment, P_{in}, is equal to the pressure outside of the compartment, P_{out}, as shown in Figure 6.9. It means that both the fresh air inflow and lower cool air out flow are zero, due to the pressure at the vents is transiently balanced. However, it is worth noting that the hot smoke from the upper layer is still flowing out of the compartment during this transient regime 2-3.

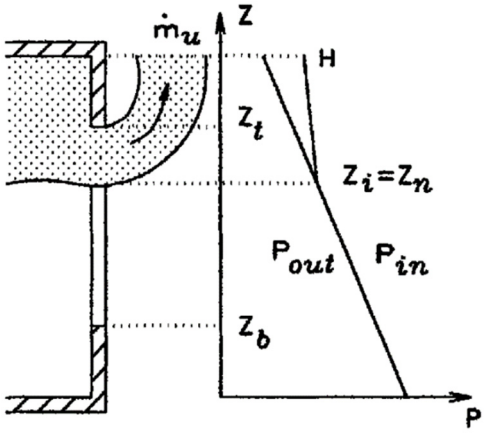


Figure 6.9. Flow regime 2-3 of the FIRM zone model implemented in the OpenSees fire module [18].

Regime 3 is the most important regime, when the pressure inside of the compartment, P_{in}, is lower than the pressure outside of the compartment, P_{out}, as shown in Figure 6.10. At this regime, the fresh cool air is 'pushed' into the compartment due to the pressure difference at the vents, and the hot smoke gases are venting out below the ceiling soffits.

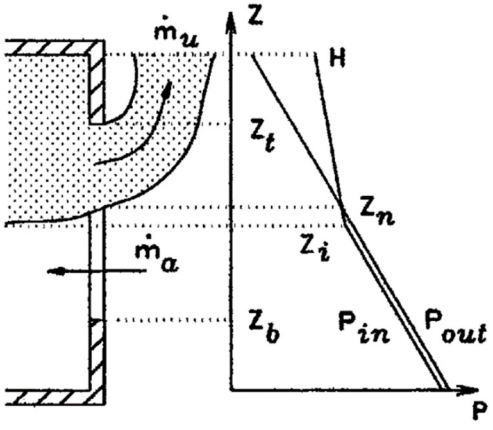


Figure 6.10. Flow regime 3 of the FIRM zone model implemented in the OpenSees fire module [18].

Regime 4 is an extreme case when the fire in the compartment is oxygen-starved, associated with a post-flashover fire under the ventilation-controlled burning. This regime is related to an upper bound value of the inflow of ambient air, $\dot{m}_a$, which is given as shown in Equation 6.7.

6.3.3 Implementation of ASET and FIRM zone models using C++

OpenSees is mainly developed by C++, which is an object-oriented programming (OOP) language. OOP is a programming paradigm designed to focus on manipulating 'objects' which are bundled with data and methods (popular programming languages such as Python, Java, and Ruby.) OOP is ideal for developing large-scale software framework, as it allows developers to reuse the developed codes much more easily and to secure the data without corruption. Moreover, it permits researchers to quickly view and gain an understanding of the workings of the codes, based on their own research interests on the specific model, instead of going through all the procedures and functions using procedural programming (popular programming languages such as C, Fortran, and Pascal).

In OOP, an object is instantiated by a class, which contains the definitions of the related data and methods. When the analysis is performed, the status of data and functions encapsulated with an object would be updated transiently. Moreover, a class definition can be used to instantiate several objects, and each object has its status updated accordingly when the analysis proceeds.

The definition of a class in C++, consists of a header (.h) file which declares the data and methods, and a source (.cpp) file which is the implementation detail of the declared methods from the corresponding header file. The source files of the *ZoneModel_ASET* class and *ZoneModel_FIRM* class are not presented in this thesis due to the limit on space. Nevertheless, corresponding header files are attached in the Appendix B.1, with all the related data variables and function methods explained with detailed annotations.

6.3.4 Limitations of the implemented zone models in SIFBuilder

The operation of the ASET zone model is basically to fill the smoke in a 'box' without considering any significant openings, hence its smoke layer temperature and depth prediction would normally rise up very quickly. The FIRM zone model is applicable for the ventilations with vertical openings through the walls only, neglecting horizontal openings through the ceiling, though the latter are much less common, and apply to scenarios with other complexities, e.g. basement fires. Furthermore, the venting is associated with natural ventilations, and forced ventilations are not considered [18].

6.4 ETFM FRAMEWORK IN SIFBUILDER

More recently, a very active research frontier on performance-based structural fire design named 'travelling fires', has begun to evolve [19]. This type of fire scenario is developed for characterising large compartment fires, which may burn locally and tend to move across entire floor plates over a period of time. Two main theoretical representations of travelling fire models can be found in previous literature, hereinafter referred to as: Clifton's model [31]; and Rein's model [32]. Clifton developed a fire model which divides the whole large compartment into several design areas, which are then subjected to modified parametric fires individually and sequentially. In Rein's original model, Alpert's ceiling jet model is adopted to calculate far field smoke temperature, and a uniform temperature (800°C-1200°C) is assumed for the near field which is adjacent to the structural members.

However, both models necessarily neglect some aspects of the fire dynamics. For instance, the accumulation of a hot smoke layer is ignored in both models. In Clifton's model, all elements in one 'firecell' (one design area) share the same fire exposure history. In Rein's model the uniform 800°C-1200°C assumption is very coarse. Furthermore, due to computational complexity for fully coupled analysis, neither of these models have thus far been coupled to the full 3D global structural response, although both travelling fire models have been developed for ultimate application to structural design.

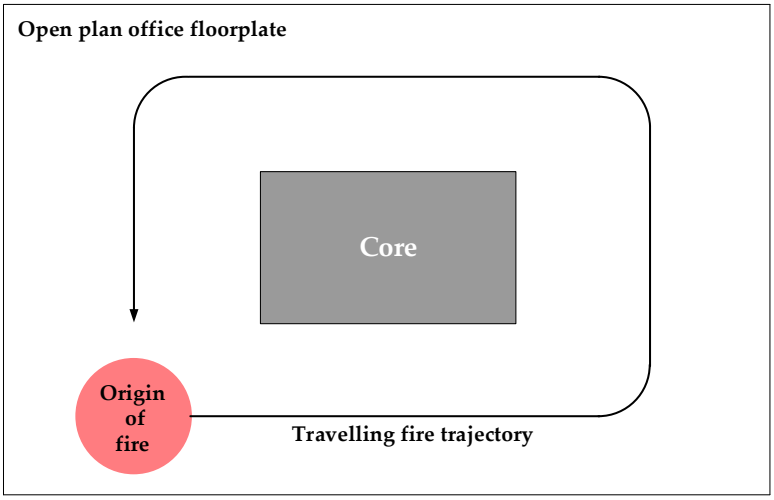


Figure 6.11. The ETFM framework in sectional plan view.

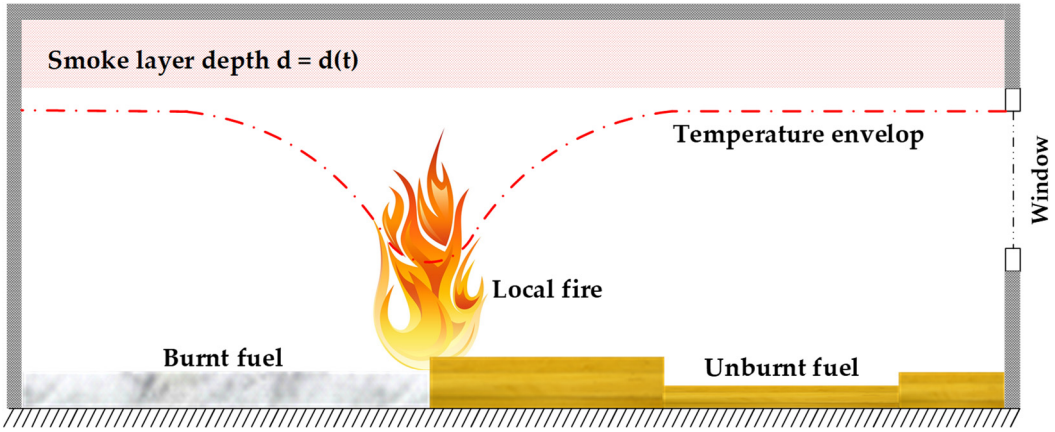


Figure 6.12. The ETFM framework in sectional elevation view.

In 2016, an extended travelling fire method (ETFM) framework[3] was proposed by the author and co-workers [19, 33, 34]. It is based on a 'mobilised version' of Hasemi's localised fire model, combined with a simple smoke layer calculation for the areas of the compartment away from the fire using FIRM zone model. This framework is idealized as a localised fire plume with characteristics that include: a predetermined plume propagation trajectory along which it travels; variable fuel load distribution along the trajectory; and consideration of smoke accumulation under the ceiling with energy and mass balance (see Figure 6.11 and Figure 6.12). A flashover scenario arises naturally in the

[3] This ETFM framework is detailed in Chapter 4.

ETFM framework, and the fire transitions from a localized travelling fire to a whole compartment fire when the temperature of the hot smoke layer reaches 500°C [29]. The key guiding principle in developing the ETFM framework for large compartments is to achieve a good balance between the enormous complexity and variability of such fires, against the practicability and simplicity required by a structural engineer in safely characterising this load. This combined fire model enables the analysis to capture both spatial and temporal changes of the thermal field, which is then automatically coupled to a thermomechanical analysis using SIFBuilder. The heat fluxes received by each structural member in a large compartment using this approach should provide greater fidelity with realistic conditions yet in a computationally tractable form.

6.4.1 Regulatory minimum fuel depth (RMFD) in the ETFM framework

In the implementation of the ETFM framework in SIFBuilder, the concept of a regulatory minimum fuel depth (RMFD) is introduced corresponding to a reference travelling fire spread rate and fuel load density. This RMFD is a layer of fuel uniformly distributed over the entire floor plate, and contributes to the total heat flux calculation. Moreover, an agreed quantity of additional lumped fuel is placed next to the most critical and/or most vulnerable parts of the structure identified in consultation with the structural engineer according to performance-based design principles. Figure 6.13 and Figure 6.14 illustrate how the travelling fire evolves based on the RMFD concept.

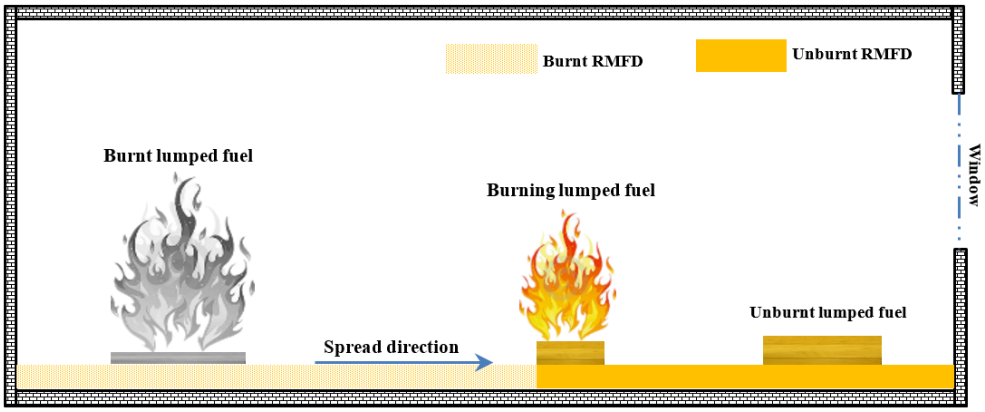


Figure 6.13. Elevation view - RMFD concept in 1D Travelling fire in the ETFM framework.

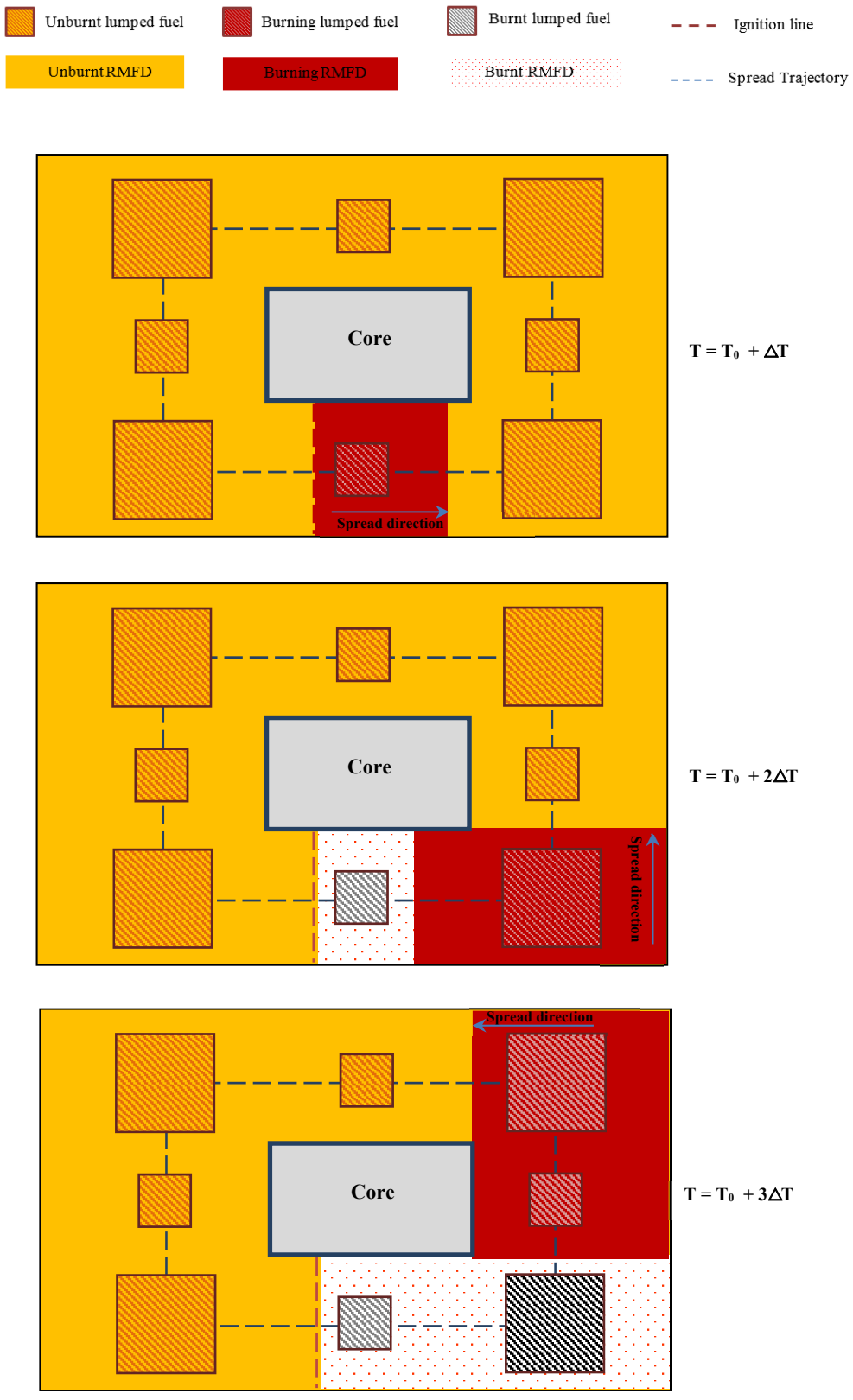


Figure 6.14. Plan view - RMFD concept in 1D Travelling fire in the ETFM framework.

6.4.2 Determination of the burning area of fuel in the ETFM framework

When the fire travels along the predefined trajectory, the burning area of fuel is determined by three variables: the travelling fire front edge location derived from the assumed constant fire spread rate, the travelling fire back edge location derived from the burn-out time (derived by the ratio of fuel load density to the maximum HRR per unit area [33]), and the compartment width derived from the floor plan dimensions. Figure 6.15 illustrates how the burning area of fuel, A_{fi}, is obtained schematically. For simplicity and clarity, the lumped fuel is not included in the drawing.

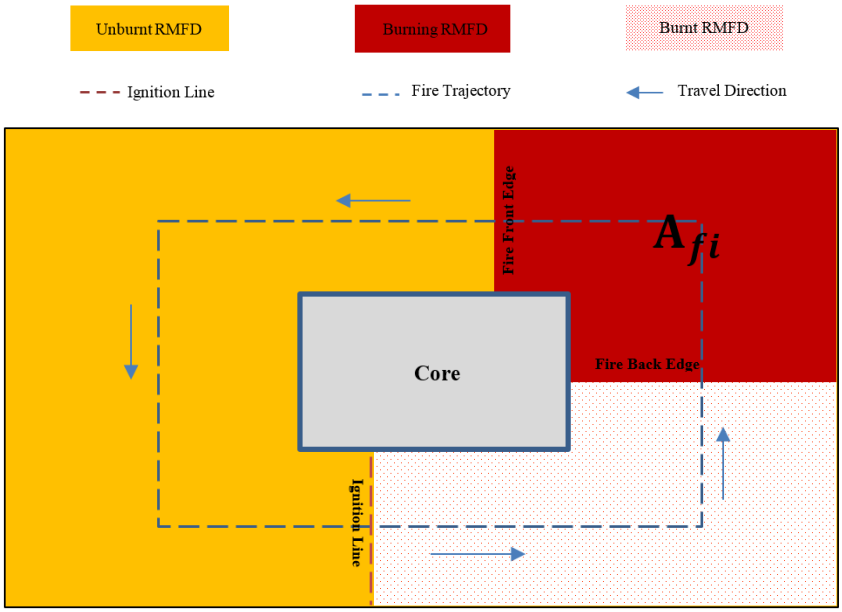


Figure 6.15. The determination of the burning area of fuel in the ETFM framework.

6.4.3 Implementation of the ETFM framework using C++

This ETFM framework is based on a 'mobile version' of Hasemi's localised fire model combined with a simple smoke layer calculation – the FIRM zone model. The *ETFM_MovingHasemi* class and *ETFM_ZoneModel_FIRM* class are added to the OpenSees fire module, to calculate the spatially and temporally non-uniform heat fluxes for different structural elements, produced from the 'summation' of the heat fluxes from the Hasemi localised fire model and the FIRM zone model (see Figure 6.16). The source (.cpp) files of the *ETFM_MovingHasemi* class and *ETFM_ZoneModel_FIRM* class are not

presented in this thesis due to the limit on space. Nevertheless, corresponding header (.h) files are attached in the Appendix B.2, with all the related data variables and function methods explained with detailed annotations.

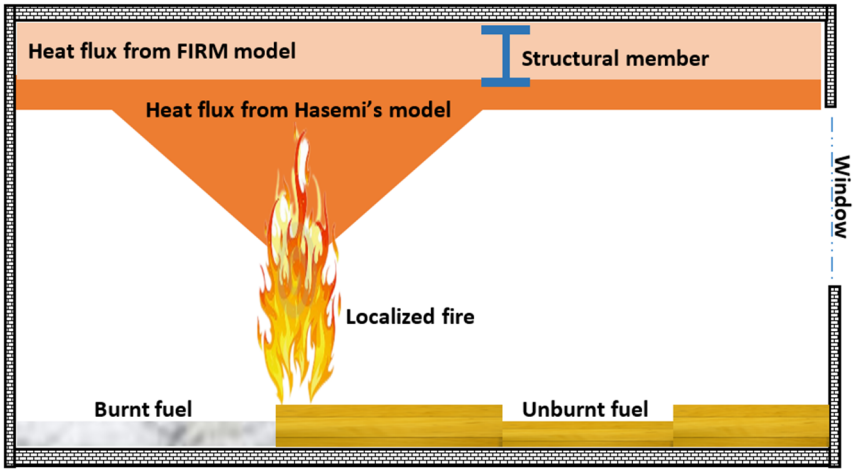


Figure 6.16. Heat flux at time t

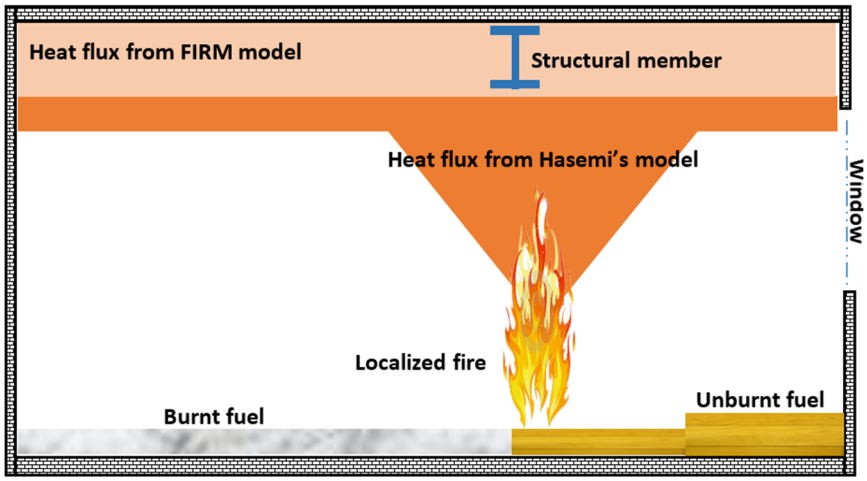


Figure 6.17. Heat flux at time $t + \Delta t$

The *ETFM_MovingHasemi* class is adapted from the *LocalizedFireEC1* class, which is the original class for the Hasemi's localised fire model implementation in OpenSees done by Jiang [9]. Moreover, the utilisation of the *ETFM_MovingHasemi* class in SIFBuilder is based on previous work [12, 35], which is mainly about integrating the heat transfer and thermo-mechanical analysis for modelling localised fire in large structures. It follows the same workflow as the localised fire model in SIFBuilder. After inputting basic structural information for generating the structural model, the user defines the structural loading and thereafter the fire loading information. The 'travelling fire' interacts with the heat

transfer module through their respective interfaces at each time step in order to determine the transient fire imposed boundary conditions adjacent to the structural surfaces. Nevertheless, unlike the localised fire model in SIFBuilder, both spatially and temporally non-uniform heat fluxes for different structural elements produced from the summation of the heat fluxes from the FIRM model and Hasemi's model, are updated at each time step according to the travelling fire location in the compartment (see Figure 6.16 and Figure 6.17, only one structural element shown for clarity). In addition, the *ETFM_ZoneModel_FIRM* class is adapted from the above-mentioned *ZoneModel_FIRM* class, which is the original class for the fire module in OpenSees to simulate the transient generation of a hot smoke layer upon the compartment ceiling with natural ventilations, through considering both the energy and mass conservations.

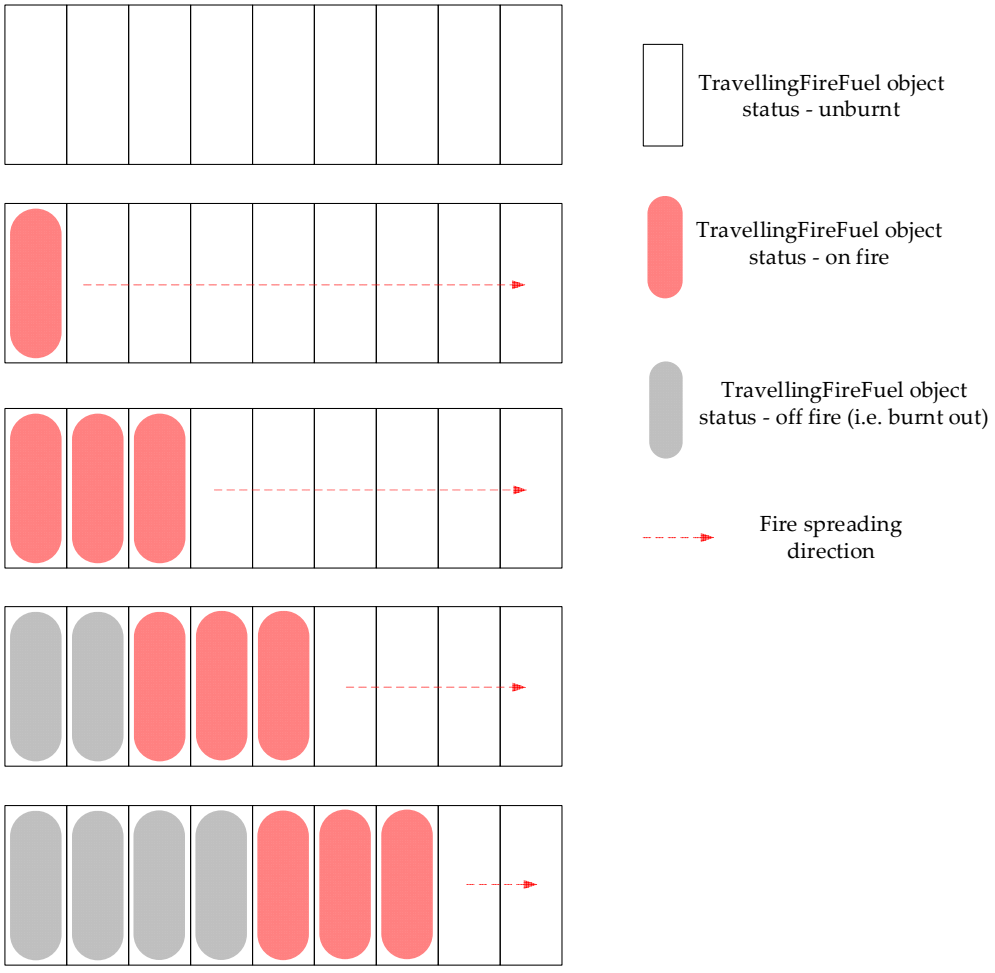


Figure 6.18. A slice of RMFD fuel bed meshed with *TravellingFireFuel* objects when the fire spreads.

In order to transiently determine the burning fuel status (e.g. transient fire location and fire size) as the fire is evolving in the large compartment, a *TravellingFireFuel* class is introduced in the OpenSees fire module. Figure 6.18 schematically demonstrates a slice of RMFD fuel bed meshed with *TravellingFireFuel* objects when the fire spreads. *TravellingFireFuel* is used to account for the distributed meshed fuel cells, and each instantiated *TravellingFireFuel* object has its own attributes. These attributes include the coordinates of the fuel cell in the compartment, size of the fuel cell, fire spread rate on the fuel cell, fuel load density, maximum HRR per unit area, and burning status (i.e. unburnt, burning, and burnt out), etc. This feature relies on the OOP using C++ in OpenSees, enabling each fuel cell (i.e. *TravellingFireFuel* object) to have its own transient status and identities. It further complies with the nature of the ETFM, as a framework. For instance, once a fire spread model is developed for the ETFM framework in the future, it can be easily associated with the fire spread rate variable for each *TravellingFireFuel* object, to 'propagate' the fire accordingly based on each fuel cell's identities (e.g. fuel load type, the relative location of the fuel cell and ventilation openings, etc.). Moreover, the *TravellingFireFuel* class also facilitates the convenience of implementation of the lumped fuel in the ETFM framework, as each fuel cell is 'consumed' independently. A *TravellingFireFuel_Iter* class is introduced for looping over all the meshed fuel cells (i.e. *TravellingFireFuel* objects) to check if the fuel is currently on fire, or not, at each time step, then aggregating them as the entire burning fuel area. The source (.cpp) files of the *TravellingFireFuel* class and *TravellingFireFuel_Iter* class are not presented in this thesis due to the limit on space. Nevertheless, corresponding header (.h) files are attached in the Appendix B.2, with all the related data variables and function methods explained with detailed annotations.

Along with the spatially and temporally changed heat fluxes induced by the ETFM framework, heat transfer analysis is performed to estimate the time-temperature histories of the structural members. Then the nodal temperature histories are automatically mapped to the fibres of the structural elements for each structural member. Following the heat transfer analysis, the thermo-mechanical analysis module is invoked to determine the structural response history for the whole frame, including all heating phases for each structural member. This may include the effects of preheating, direct heating, post-

heating and cooling. The implementation of the ETFM framework in SIFBuilder is schematically explained in Figure 6.19.

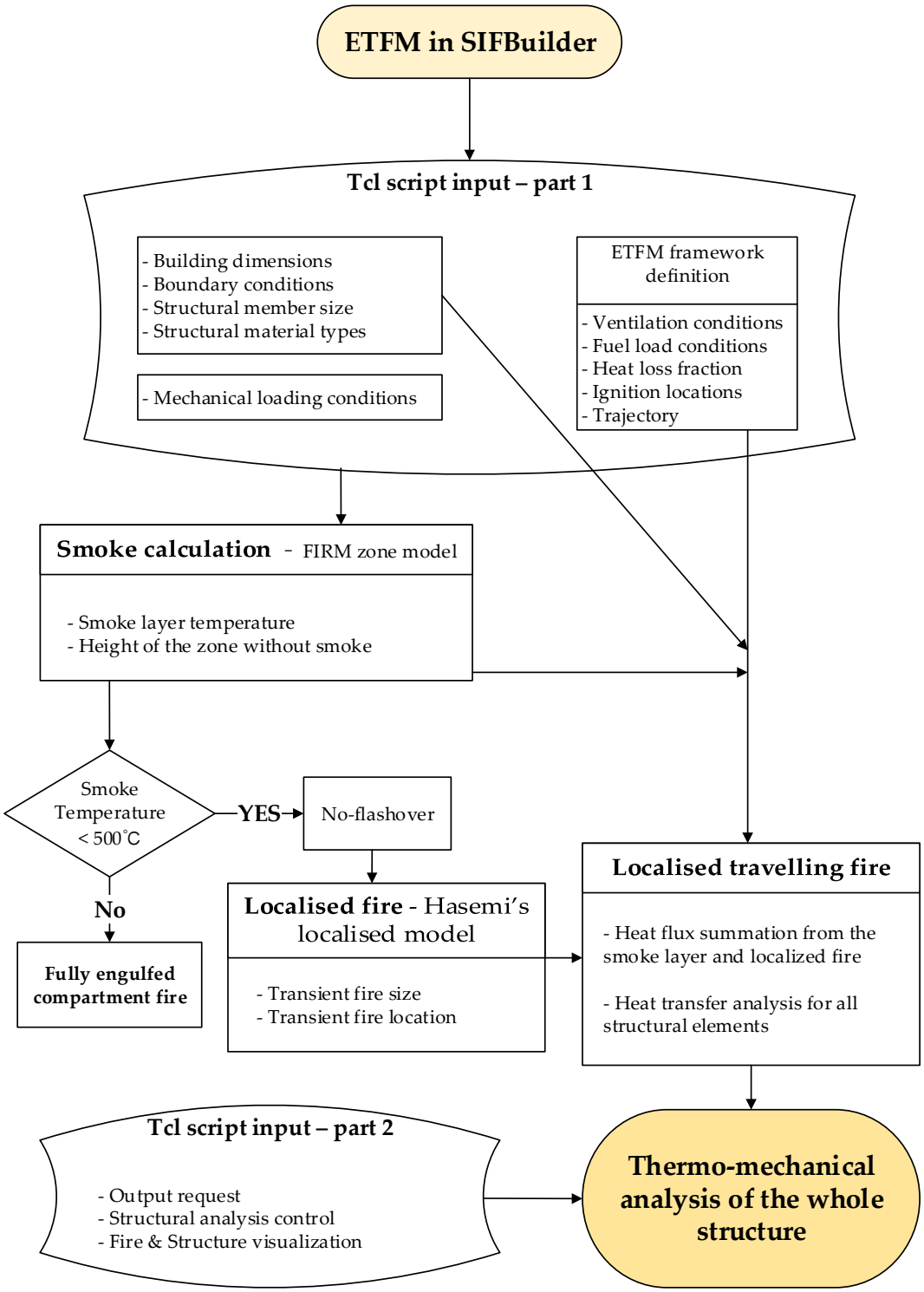


Figure 6.19. Flowchart of the ETFM implementation in SIFBuilder.

6.4.4 Limitations of the implemented ETFM framework in SIFBuilder

This ETFM framework is developed for providing a more realistic tool for structural design of fire resistance. There are several inevitable limitations in the model. Firstly, it is essentially a 1D trajectory-based travelling fire model, which is currently only applicable to floor plans with a core, or rectangular floorplan shape. Secondly, a potential limitation of the ETFM framework is the applicability of Hasemi's localized fire model, which is only strictly valid for fire diameters is less than 10m, and the rate of heat release less than 50 MW [13], though these are very large fire sizes for typical compartments. Finally, the utilised FIRM zone model in the ETFM framework, is applicable for the ventilations with vertical openings through the walls, rather than horizontal openings through the ceiling, though the latter are much less common, and apply to scenarios with other complexities, e.g. basement fires. Furthermore, the venting is assumed to be associated with natural ventilations, and forced ventilations are not considered [18].

6.5 VISUALISATION OF FIRE MODELS IN SIFBUILDER

During the fire and heat transfer analysis using SIFBuilder, the fire status would be rendered onto the screen monitor, for the users to check its fire modelling status transiently. The work of rendering fire modelling status in SIFBuilder is developed through utilising an existing class in original version of OpenSees. This class name is *OpenGLRenderer*, which is a class for displaying the Open Graphics Library (OpenGL) by rendering. Further, OpenGL is an application programming interface (API) for rendering 2D and 3D vector graphics [36]. Moreover, this visualisation capability is more important for the interested researchers to continue developing the fire module in SIFBuilder, as their proposed fire status can be visualized in the analysis during the code development debugging stage.

However, this effort (i.e. visualisation of fire models in SIFBuilder) is not direct relevant to the theme of this chapter, which is the implementation of fire models. Therefore, the relevant classes developed for this work with corresponding header (.h) files, are attached in Appendix B.3 accompanied with a code flowchart for the future developers to refer to.

6.6 VALIDATION OF ZONE MODELS IN SIFBUILDER

The zone modelling capability in the fire module of OpenSees is based on the ASET and FIRM zone models from Janssens' book [18]. The software of using ASET zone model and FIRM zone model are attached in the book, with names ASET-QB software and FIRM-QB software respectively. To validate the zone modelling results produced from OpenSees against the original software packages due to Janssens' book, a case study is performed as follows.

6.6.1 ASET in OpenSees vs. ASET-QB software

The floor area of the case study compartment is 468 m^2, with the clear floor height of 3.85 m. The fuel load density and maximum HRR per unit area are assumed to be 570 MJ/m^2 and 500 kW/m^2, respectively. The fire would spread in the compartment with a constant spread rate of 10 mm/s. Figure 6.20 and Figure 6.21 demonstrate the smoke layer temperature increase and smoke layer depth increase with the evolving time respectively, with OpenSees and ASET-QB software. It can be seen that the results from both code agree well.

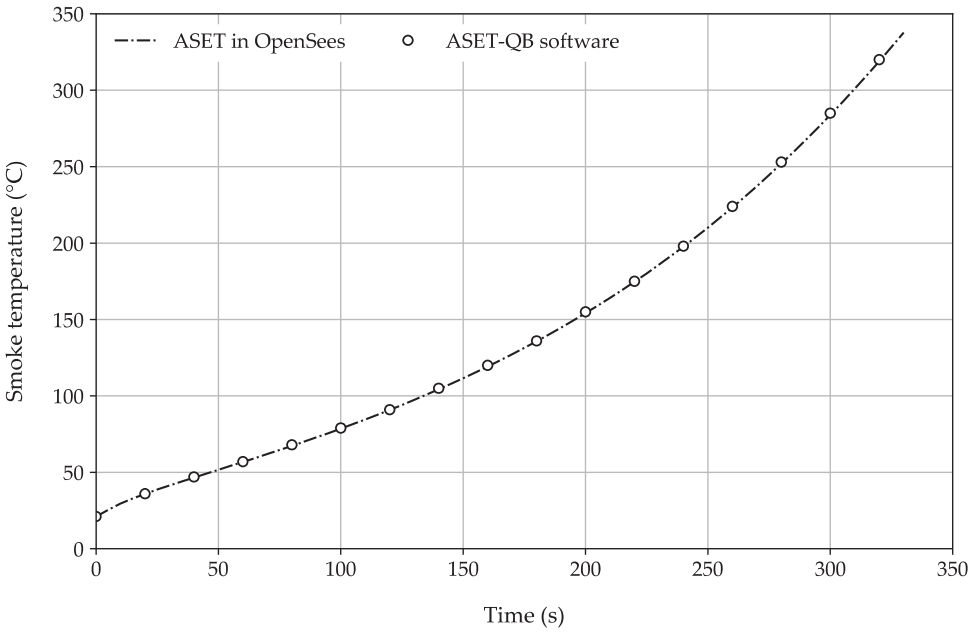


Figure 6.20. Smoke layer temperature evolution using OpenSees against ASET-QB software.

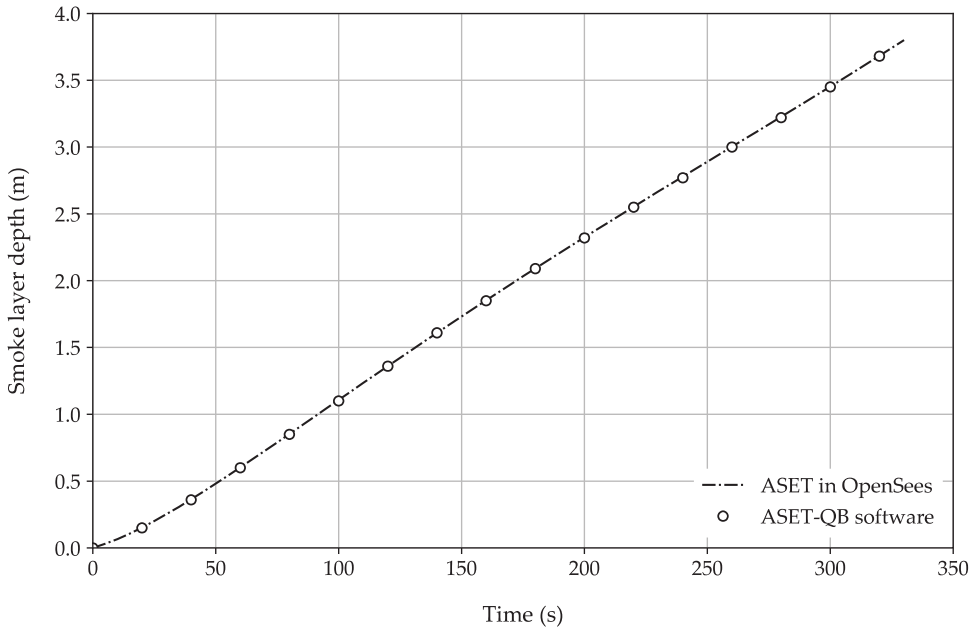


Figure 6.21. Smoke layer depth evolution using OpenSees against ASET-QB software.

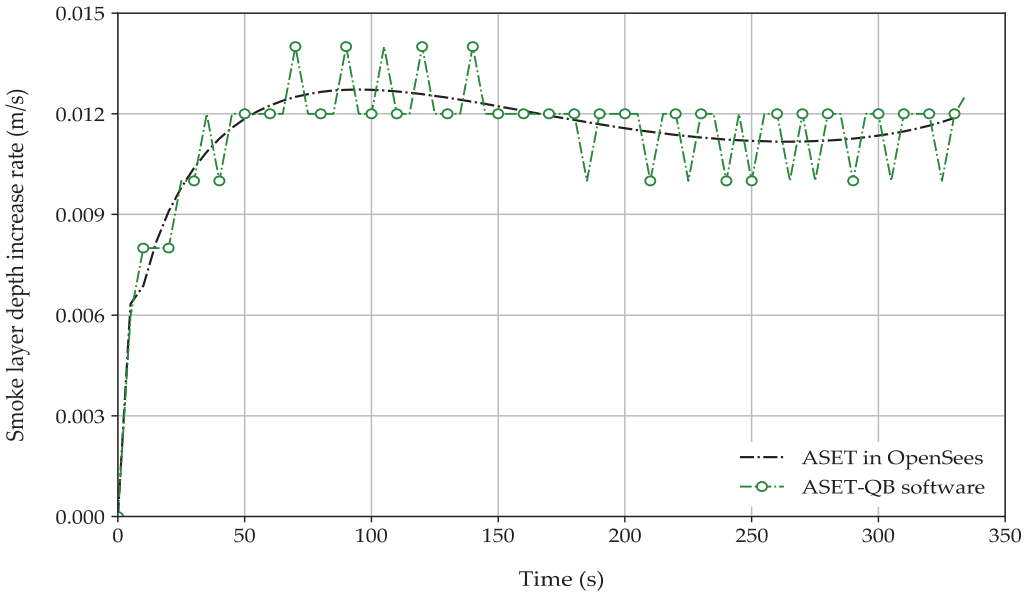


Figure 6.22. Smoke layer depth increase rate evolvement using OpenSees against ASET-QB software.

Nevertheless, it is worth to note that the smoke 'fills up' the whole compartment with a very short period of time (around 330 s), this is due to the nature of ASET zone model, which is "filling the smoke in a 'box' without considering any significant openings". Figure 6.22 shows the smoke layer depth increase rate evolution, which is the ratio of the increase of the smoke layer depth within an incremental time step, to the corresponding

time step length. Different from a smooth curve generated by OpenSees, a curve with turbulence is produced by ASET-QB software. This is due to the rounding decimal limitations of the software package back in year 2000. In addition, Figure 6.23 shows the comparison of the fire HRR evolution from the two codes.

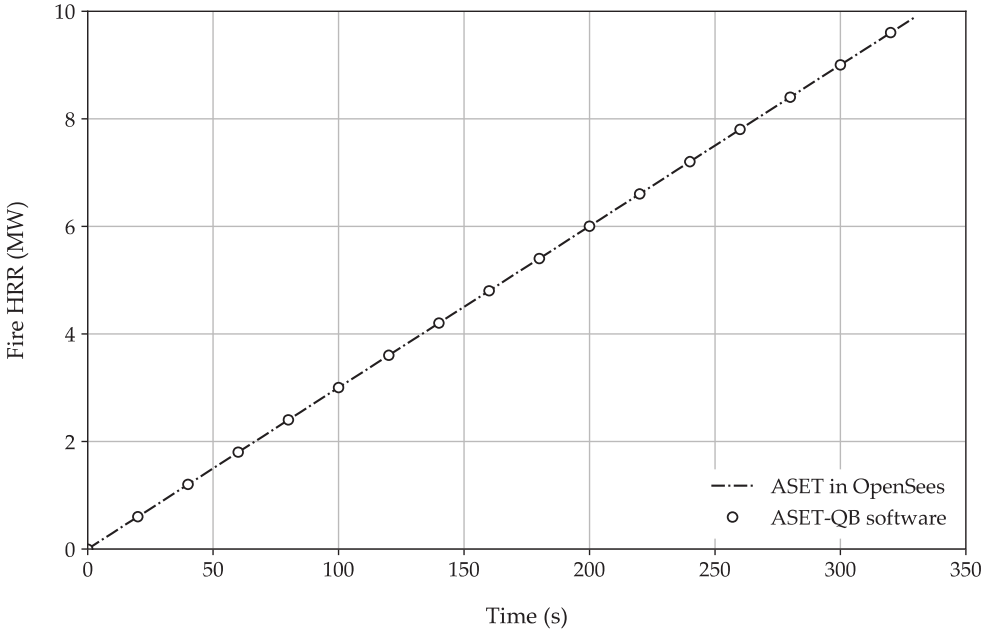


Figure 6.23. Fire HRR evolution using OpenSees against ASET-QB software.

6.6.2 FIRM in OpenSees vs. FIRM-QB software

As introduced in the previous sections, the main difference between the FIRM zone model and the ASET zone model is that the FIRM considers the vertical natural ventilations while ASET does not any significant ventilations. Therefore, the case study for FIRM zone model is the same as what has been investigated in the previous section, but including ventilations. The dimensions of the ventilations are 28 m for the total vent widths, 3 m for the ceiling soffit height, and 1 m for the sill height.

Figure 6.24, Figure 6.25 and Figure 6.26 demonstrate the evolution of the smoke layer temperature increase, smoke layer depth increase, and the height of zone free of smoke respectively, with the OpenSees and the ASET-QB software. It can be seen that the results from both codes agree well. Furthermore, the increase rate of the smoke layer depth evolution using both codes is presented in Figure 6.27. Another curve with turbulence is

produced by the FIRM-QB software. Again, this is also due to the same reason, that is the rounding decimal limitations of the software package back in year 2000. In addition, Figure 6.28 shows the comparison of the fire HRR evolution from the two codes.

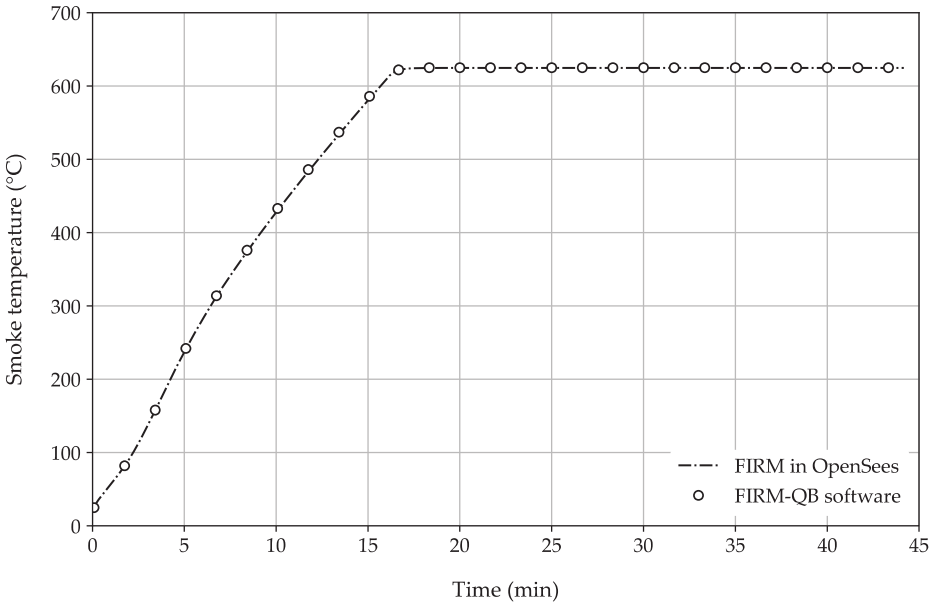


Figure 6.24. Smoke layer temperature evolution using OpenSees against FIRM-QB software.

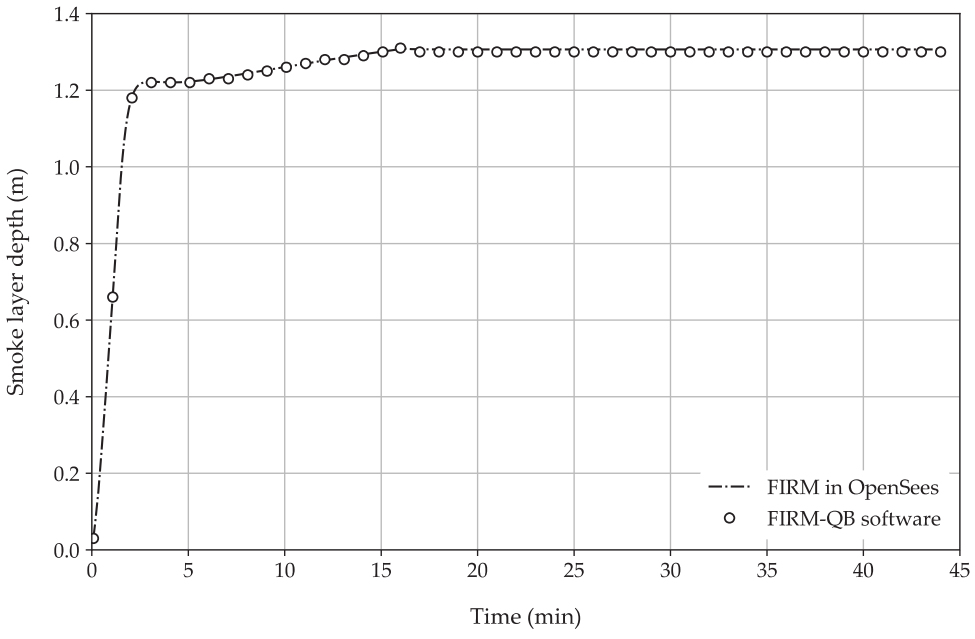


Figure 6.25. Smoke layer depth evolution using OpenSees against FIRM-QB software.

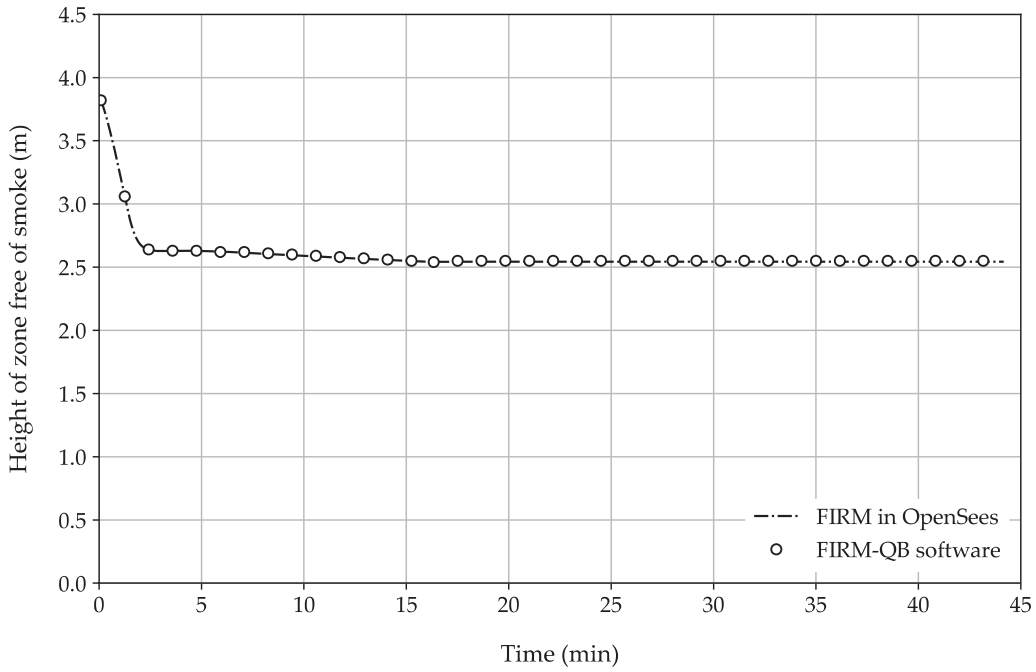


Figure 6.26. The evolution of height of zone free of smoke, using OpenSees against FIRM-QB software.

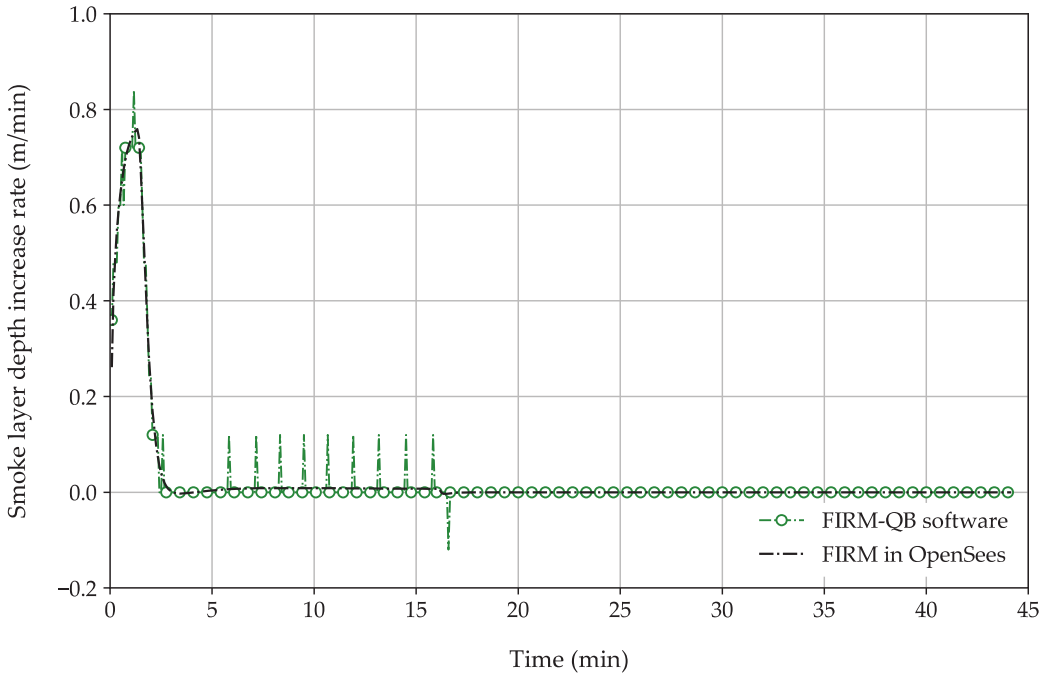


Figure 6.27. The evolution of the smoke layer depth increase rate, using OpenSees against FIRM-QB software.

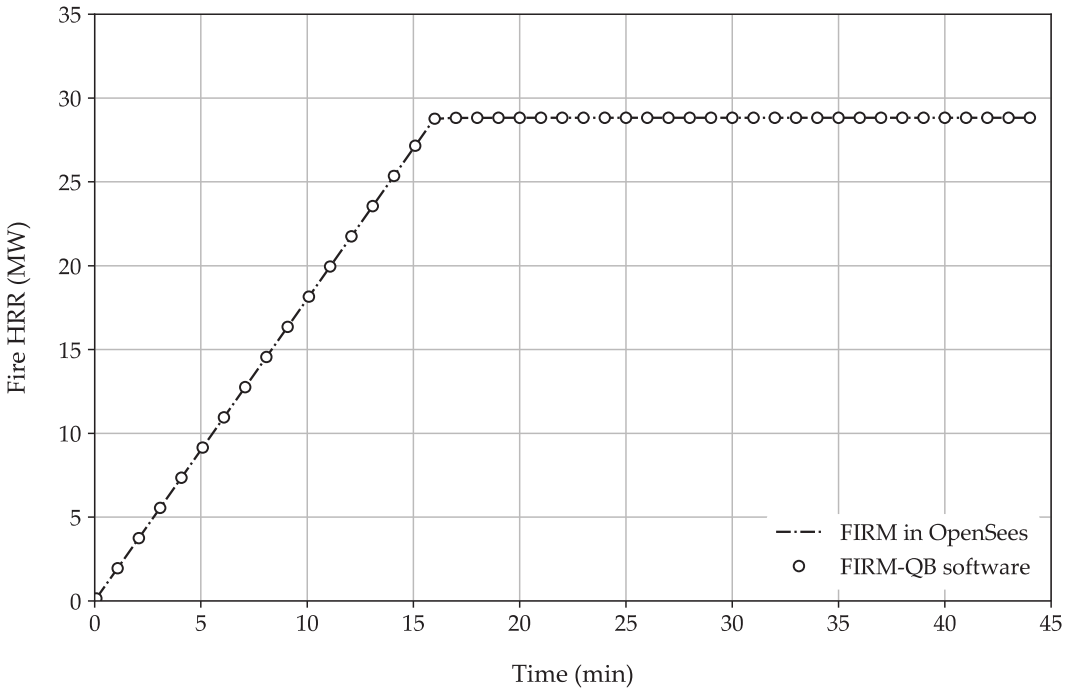


Figure 6.28. Fire HRR evolution using OpenSees against FIRM-QB software.

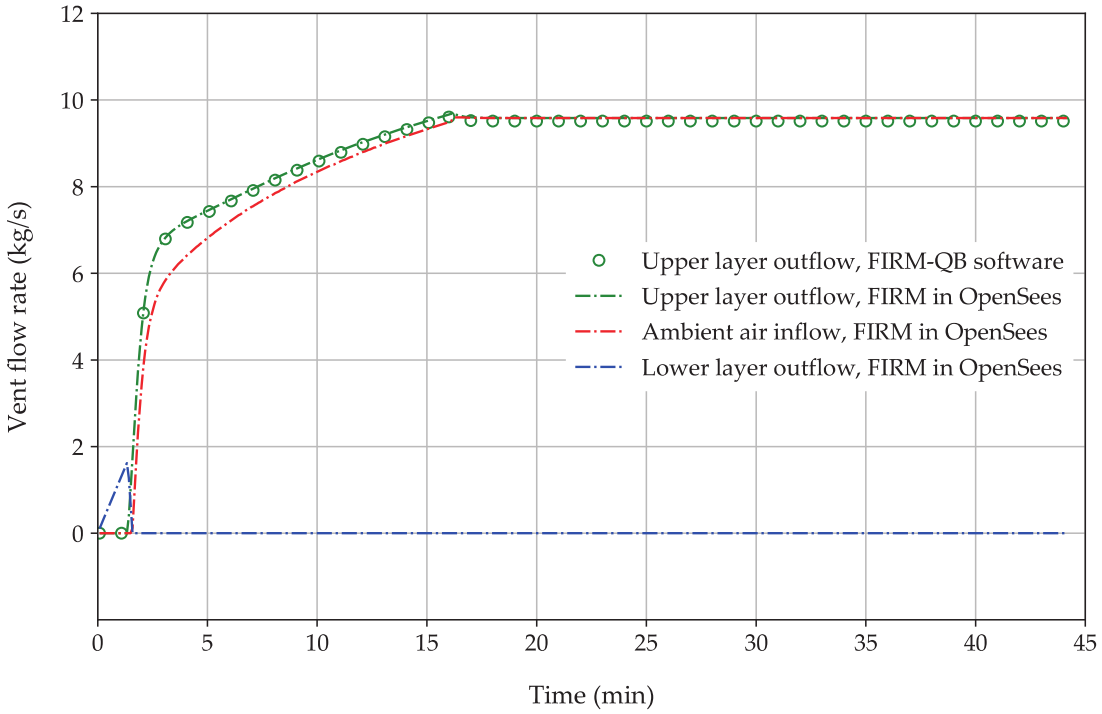


Figure 6.29. The evolution of vent flows using OpenSees against FIRM-QB software.

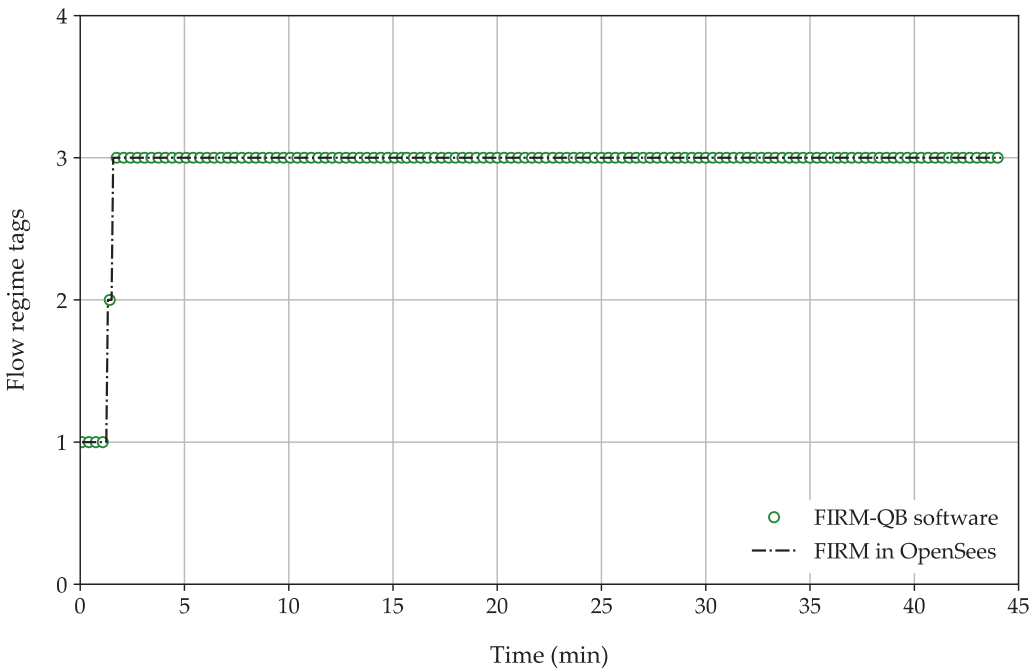


Figure 6.30. Flow regimes during the smoke evolution, using OpenSees against FIRM-QB software.

Figure 6.29 and Figure 6.30 illustrate the vent flow rates and corresponding vent flow regimes when the fire is developing in the case study compartment. According to the vent flow regimes of the FIRM zone model introduced in section 6.3.2, it can be seen that vent flow regime 1 first launches, with increasing lower layer (i.e. cool air) outflow rate due to the piston effect. Then the smoke upper layer outflow decreases to zero, instead the ambient air inflow and upper layer outflow start to increase. This is because the smoke vent flow regime quickly transits from regime 1 to the stabilized regime 3. Based on Figure 6.29 and Figure 6.30, it can be concluded that the flow regimes can be correctly estimated by OpenSees, compared against the original FIRM-QB software.

6.7 CONCLUSIONS

This chapter first reviews the existing fire models in OpenSees, including uniform compartment fire models (e.g. nominal fire curves, parametric fires) and localised fire models (e.g. Hasemi localised fire model, Alpert ceiling jet model). Secondly, the theoretical background of the ASET zone model, the FIRM zone model, and an existing travelling fire method (ETFM) framework are introduced. Their implementation details in SIFBuilder using C++ are also included, with corresponding header (.h) files attached in Appendix B. Finally, a case study using the ASET zone model and the FIRM zone model in OpenSees is investigated, for validating the results against the original ASET-QB software, and the FIRM-QB software respectively [18]. It shows the results obtained from the OpenSees agree well with those generated from the ASET-QB and FIRM-QB software packages.

Therefore, with the more advanced design fire methodologies implemented (i.e. zone models and the ETFM framework), SIFBuilder will provide a more flexible approach for examining the impact of fire on structural behaviour under realistic design fire scenarios, with a greatly reduced cost in terms of analysis time and user effort, than currently possible.

6.8 REFERENCES

[1] J. Franssen and T. Gernay, "Modeling structures in fire with SAFIR: theoretical background and capabilities," *Journal of Structural Fire Engineering*, vol. 8, no. 3, pp. 300–323, 2017.

[2] "Vulcan Software Package," *Vulcan Solutions Ltd.*, 2005. [Online]. Available: www.vulcan-solutions.com. [Accessed: 04-Sep-2017].

[3] B. Izzuddin, "Nonlinear dynamic analysis of framed structures," PhD Thesis, Department of Civil Engineering, Imperial College London, 1991.

[4] K. McGrattan, "Fire modeling: Where are we? Where are we going?," *Fire Safety Science*, pp. 53–68, 2005.

[5] F. T. McKenna, "Object-Oriented Finite Element Programming: Frameworks for Analysis, Algorithms and Parallel Computing," PhD Thesis, University of California, 1997.

[6] S. Mazzoni, F. Mckenna, M. H. Scott, and G. L. Fenves, "Open System for Earthquake Engineering Simulation (OpenSees): User Command-Language Manual," *Pacific Earthquake Engineering Research Center, University of California, Berkeley*, 2006. [Online]. Available: http://opensees.berkeley.edu/OpenSees/manuals/usermanual/index.html. [Accessed: 02-Sep-2017].

[7] F. McKenna, G. L. Fenves, and M. H. Scott, "Open System for Earthquake Engineering Simulation (OpenSees)," *Pacific Earthquake Engineering Research Center*, 2000. [Online]. Available: http://opensees.berkeley.edu. [Accessed: 02-Sep-2017].

[8] A. Usmani, J. Zhang, J. Jiang, Y. Jiang, and I. May, "Using OpenSees for structures in fire," *Journal of Structural Fire Engineering*, vol. 3, no. 1, pp. 57–70, 2012.

[9] Y. Jiang, "Development and Application of A Thermal Analysis Framework in OpenSees for Structures in Fire," https://www.era.lib.ed.ac.uk/handle/1842/7941, PhD Thesis, School of Engineering, University of Edinburgh, 2012.

[10] J. Jiang and A. Usmani, "Modeling of steel frame structures in fire using OpenSees," *Computers & Structures*, vol. 118, pp. 90–99, 2013.

[11] J. Jiang, L. Jiang, P. Kotsovinos, J. Zhang, A. Usmani, F. McKenna, and G.-Q. Li, "OpenSees software architecture for the analysis of structures in fire," *Journal of Computing in Civil Engineering*, vol. 29, no. 1, 2015.

[12] L. Jiang, X. Dai, A. Usmani, and P. Kamath, "OpenSees-based integrated tool for modelling structures in fire," in *The First International Conference on Structural Safety under Fire & Blast, Glasgow, Scotland, UK*, 2015, pp. 461–468.

[13] European Standard EN 1991-1-2, "Eurocode 1: Actions on structures - Part 1-2: General actions - Actions on structures exposed to fire." CEN, Brussels, 2002.

[14] R. G. Gann, A. Hamins, K. McGrattan, H. E. Nelson, T. J. Ohlemiller, K. R. Prasad, and W. M. Pitts, "Reconstruction of the fires and thermal environment in World Trade Center buildings 1, 2, and 7," *Fire Technology*, vol. 49, pp. 679–707, 2013.

[15] I. Fletcher, A. Borg, N. Hitchen, and S. Welch, "Performance of concrete in fire: A review of the state of the art, with a case study of the Windsor Tower fire," in *4th International Workshop in Structures in Fire*, pp. 779–790.

[16] D. M. Zannoni, J. G. H. Bos, D. K. E. Engel, and P. dr. U. Rosenthal, "Brand bij Bouwkunde," COT Instituut voor Veiligheids - en Crisismanagement, 2008.

[17] "Plasco Building Fire in Tehran," *Wikipedia*, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Plasco_Building. [Accessed: 08-Oct-2017].

[18] M. L. Janssens, *An Introduction to Mathematical Fire Modeling*, 2nd ed. CRC Press, 2000.

[19] X. Dai, S. Welch, and A. Usmani, "A critical review of 'travelling fire' scenarios for performance-based structural engineering," *Fire Safety Journal*, vol. 91, no. C, pp. 568–578, 2017.

[20] ISO 834-1:1999(E), "Fire-resistance tests - Elements of building construction - Part 1: General requirements." International Organization for Standardization, 1999.

[21] "Standard Test Methods for Fire Tests of Building Construction and Materials, ASTM E119." American Society for Testing and Materials, West Conshohocken, PA, 2016.

[22] J. Franssen, V. Kodur, and R. Zaharia, *Designing Steel Structures for Fire Safety*. CRC Press, 2009.

[23] R. L. Alpert, "Calculation of response time of ceiling-mounted fire detectors," *Fire Technology*, vol. 8, no. 3, pp. 181–195, 1972.

[24] B. Y. Lattimer, "Heat fluxes from fires to surfaces," in *SFPE Handbook of Fire Protection Engineering*, Third Ed., National Fire Protection Association, 2002.

[25] L. Jiang, "Development of An Integrated Computational Tool for Modelling Structural Frames in Fire Considering Local Effects," https://www.era.lib.ed.ac.uk/handle/1842/19563, PhD Thesis, School of Engineering, University of Edinburgh, 2016.

[26] "ASTM E1355-04, Standard Guide for Evaluating the Predictive Capabilities of Deterministic Fire Models." American Society for Testing and Materials, West Conshohocken, Pennsylvania, 2004.

[27] S. Kumar, R. Chitty, and S. Welch, "Development and validation of integrated fire modelling methodology for smoke ventilation design and hazard assessment in large buildings," in *Proceedings of the 4th International Seminar on Fire & Explosion Hazards*, 2003.

[28] P. H. Thomas, P. L. Hinkley, and C. R. Theobald, *Investigations into the Flow of Hot Gases in Roof Venting*. BRE Archive, 1963.

[29] D. Drysdale, *An Introduction to Fire Dynamics*, Third Edit. A John Wiley & Sons, Ltd., Publication, 2011.

[30] E. E. Zukoski, T. Kubota, and B. Cetegen, "Entrainment in fire plumes," *Fire Safety Journal*, vol. 3, no. 3, pp. 107–121, 1981.

[31] C. G. Clifton, "Fire Models for Large Firecells. HERA Report R4-83," HERA publications, New Zealand, 1996.

[32] E. Rackauskaite, C. Hamel, A. Law, and G. Rein, "Improved formulation of travelling fires and application to concrete and steel structures," *Structures*, vol. 3, pp. 250–260, 2015.

[33] X. Dai, L. Jiang, J. Maclean, S. Welch, and A. Usmani, "A conceptual framework for a design travelling fire for large compartments with fire resistant islands," in *Proceedings*

of the 14th International Interflam Conference, 2016, pp. 1039–1050.

[34] X. Dai, L. Jiang, J. Maclean, S. Welch, and A. Usmani, "Implementation of a new design travelling fire model for global structural analysis," in *the Proceedings of the 9th International Conference on Structures in Fire*, 2016, pp. 959–966.

[35] L. Jiang, Y. Jiang, J. Jiang, A. Usmani, and S. Chen, "An OpenSees-based integrated tool for modelling structures in realistic design fires," in *the 8th International Conference on Structures in Fire, Shanghai, China*, 2014, pp. 987–994.

[36] "Open Graphics Library (OpenGL)," *Wikipedia*, 2017. [Online]. Available: https://en.wikipedia.org/wiki/OpenGL. [Accessed: 29-Oct-2017].

Chapter 7.

A Case Study Using the ETFM Framework with SIFBuilder

7.1 INTRODUCTION

As discussed in Chapter 4, there are three design domains for the structural fire safety engineering: the time domain, the temperature domain, and the strength domain. Both Clifton's travelling fire model [1–3] and Rein's travelling fire model [4–11] have been well explored on all these three design domains in the past two decades. However, by nature both previous models are based on simply 'mobilizing' existing independent models (i.e. modified parametric fire curves in Clifton's model, 800°C-1200°C temperature block and Alpert's ceiling jet in Rein's model), without considering the essential energy and mass conservations like the ETFM framework. It means that the results and conclusions obtained through previous studies may neglect some important engineering implications, and potentially produce misleading results.

This chapter investigates the engineering implications of the travelling fires using the ETFM framework, mainly in the temperature design domain. The investigation is carried out through quantifying the cross-sectional time-temperature evolution of the steel beams with a case study. Their thermal response implications on the subsequent structural responses (e.g. the temperature rise induced thermal expansion axial force, the change of through depth induced thermal bowing bending moment) are also discussed. Furthermore, a demonstration of the ETFM framework for characterising structural response using SIFBuilder is also presented.

Both the fire and heat transfer analysis are performed with the ETFM framework using SIFBuilder [12–15], which is an OpenSees-based, open-source software framework with features of facilitating analysis of fire, heat transfer and thermo-mechanical response in one single software package. Its 'thermal' modelling robustness has been widely verified and validated by Jiang [16].

7.1.1 Input Parameters for the Parametric Studies

The ETFM framework involves a certain number of input parameters, such as the fuel load densities, ventilation dimensions, fire spread rates, etc. These parameters actually are used to evaluate the fire severity impact on the structure. Meanwhile, the parameters such as the building dimensions, structural element cross sections, material types etc., are to be used to evaluate the structural resistance ability. Along with the integrated tool (i.e. SIFBuilder), a variety of parameter types can be chosen in the following table:

Table 7.1. Available input parameters using ETFM framework along with SIFBuilder.

<table>
<tr><th colspan="2">Design fire severity</th><th colspan="2">Structural fire resistance</th></tr>
<tr><td rowspan="3">Fuel load conditions</td><td>Fire spread rate</td><td rowspan="4">Building dimensions</td><td>Each bay length</td></tr>
<tr><td>Fuel load density</td><td>Each bay width</td></tr>
<tr><td>HRR per unit area</td><td rowspan="2">Each bay height (i.e. compartment height)</td></tr>
<tr><td rowspan="3">Ventilation conditions</td><td>Sill height</td></tr>
<tr><td>Soffit height</td><td rowspan="3">Structural member size</td><td rowspan="2">I-beam section (e.g. section depth, flange thickness, etc.)</td></tr>
<tr><td>Total vent width</td></tr>
<tr><td rowspan="2">Trajectory</td><td>Pre-defined path</td><td rowspan="2">Flat concrete slab (e.g. slab thickness)</td></tr>
<tr><td>Travel direction</td></tr>
<tr><td colspan="2">Ignition location</td><td rowspan="2">Structural material</td><td>Steel</td></tr>
<tr><td rowspan="2">Air entrainment model</td><td>Zukoski’s model</td><td>Concrete</td></tr>
<tr><td>Thomas model</td><td rowspan="2">Mechanical loading conditions & combinations</td><td>Concentrated loads</td></tr>
<tr><td colspan="2">Radiative loss fraction of the fire plume</td><td>Uniformly distributed loads (UDL)</td></tr>
<tr><td colspan="2">Heat loss fraction ratio through walls & ceilings</td><td colspan="2" rowspan="4">Non-fire compartments definition (i.e. compartments with full fire protections presumably)</td></tr>
<tr><td colspan="2">Convection coefficient for heat transfer</td></tr>
<tr><td colspan="2">Emissivity of the material for heat transfer</td></tr>
<tr><td colspan="2">Lumped fuel distribution</td></tr>
</table>

Table 7.1 illustrates the varieties of the available input parameters that can be used to study the thermal impact on the structures due to the ETFM framework. To minimise the uncertainties and complexities, with better interpretable results, only the RMFD assumption with variation of two key design parameters are investigated in this chapter: the constant fire spread rate, v (mm/s), and the characteristic fuel load density, $q_{f,k}$ (MJ/m^2).

Table 7.2. Sampling points adopted to perform parametric studies for the ETFM framework.

Fire spread rate (mm/s) / **Fuel load density (MJ/m^2)**	1.6	2.0	5.0	8.0	10.0	15.0
100 (Transport)	•	•	•	•	•	•
230 (Hospital)	•	•	•		•	
300 (Theatre)	•	•	•		•	
420 (Office)	•			•	•	
570	•	•	•	•	•	•
600 (Shopping centre)	•				•	•
780 (Dwelling)	•				•	•

These two parameters essentially determine the travelling fire speed and the burning area, as explained in the previous sections. The range of these two values are 1.6-15 mm/s and 100-780 MJ/m^2, respectively, according to the values recommended in Chapter 4. A 'base line scenario' of the travelling fires is assumed, with fire spread rate of 10.0 mm/s, fuel load density of 570 MJ/m^2. The sampling data points are summarized in Table 7.2, where each black dot stands for a design fire scenario using ETFM framework. The total number of sampling data points is 29, which means 29 travelling fire scenarios are investigated altogether.

7.1.2 The Structure for Case Study

Figure 7.1 shows the plan view of an idealised structural layout for the case study (630 m^2 floor area), which is generic in modern tall buildings with a core (162 m^2 area) in the middle. Fire tends to 'travel' in this type of structural layout, similar to the WTC example represented in Chapter 4. A steel beam is investigated, with size UB 305×127×42, located at the top right of the floor plan. Concrete slabs are assumed at the top of the steel beams. The ignition line of the travelling fire is also shown as in Figure 7.1. The travelling fire trajectory is predefined to be under the mid-span of the main beams, which would normally represent the worst case for the structural response. The fire travel direction is assumed in the anti-clockwise direction.

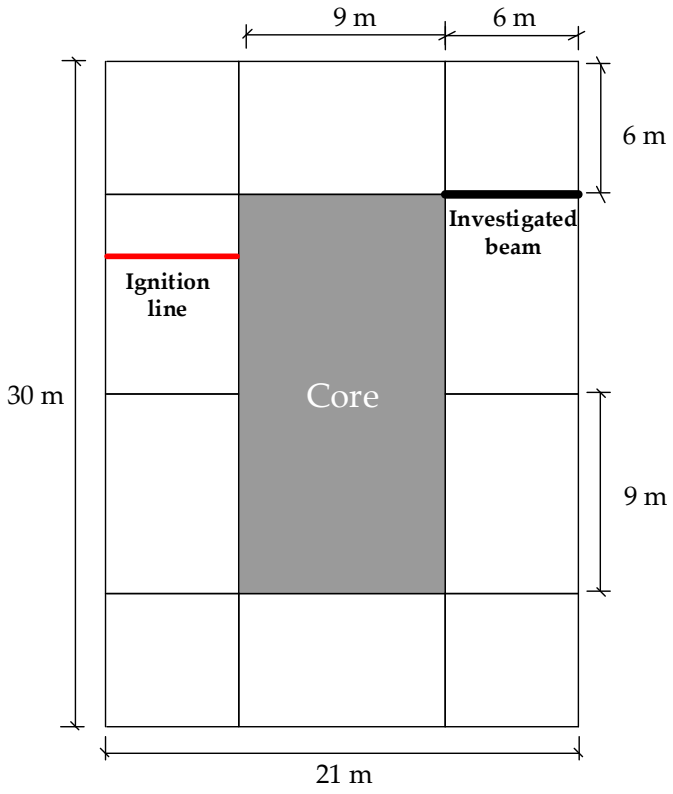


Figure 7.1. Case study plan view using the ETFM framework.

Figure 7.2 schematically shows the elevation of the investigated case study. The fire is ignited on the first floor of the building. Its clear compartment height, H, is 3.85 m. The total vent widths of this large compartment are 28 m. The soffit height and sill height are 3 m and 1 m respectively. The heat loss fraction ratio through the compartment boundaries, L_c , is assumed to be 0.8 (0.6 ~ 0.9 as recommended in [17]), and the radiative loss fraction of the fire plume, L_r , is assumed to be 0.35 (as recommended in [17]).

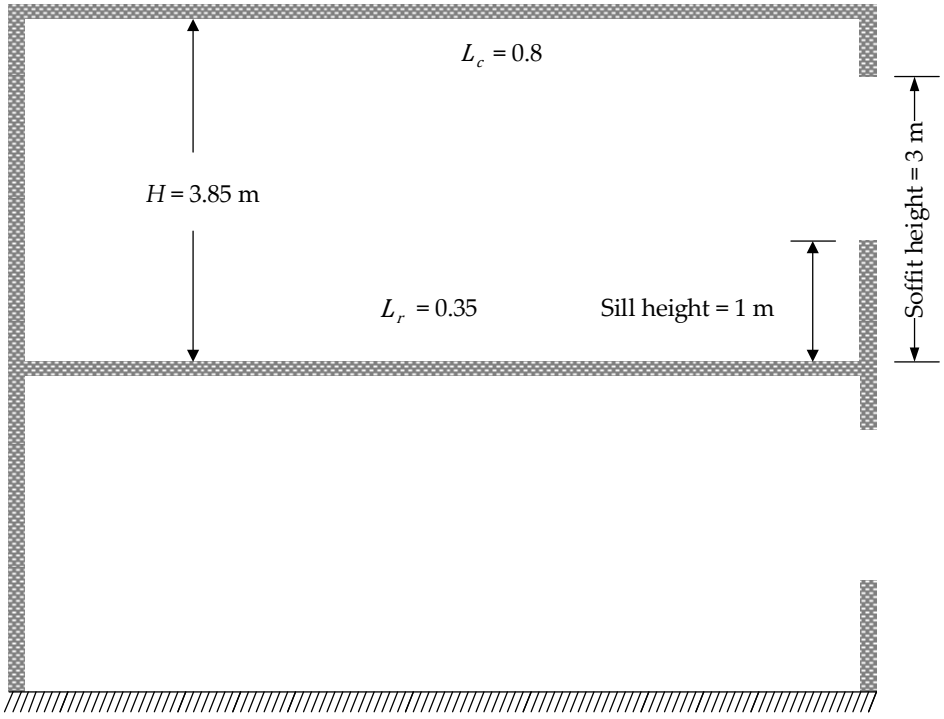


Figure 7.2. Case study elevation view using the ETFM framework.

7.1.3 The Travelling Fire Scenarios and Heat Transfer Modelling

A RMFD 'base line scenario' of the travelling fires is assumed with fuel load density, $q_{f,k}$, equal to 570 MJ/m^2, and fire spread rate, v, equal to 10 mm/s. Heat release rate per unit area, RHR_f, is assumed to be a constant, of 500 kW/m^2 (as recommended in Chapter 4) for all the travelling fire scenarios. Different fire scenarios would be generated with the combinations of varying v or $q_{f,k}$ as shown in Table 7.2. In addition, Zukoski's plume model is employed to calculate the transient air entrainment mass flow rate, $\dot{m}_e$.

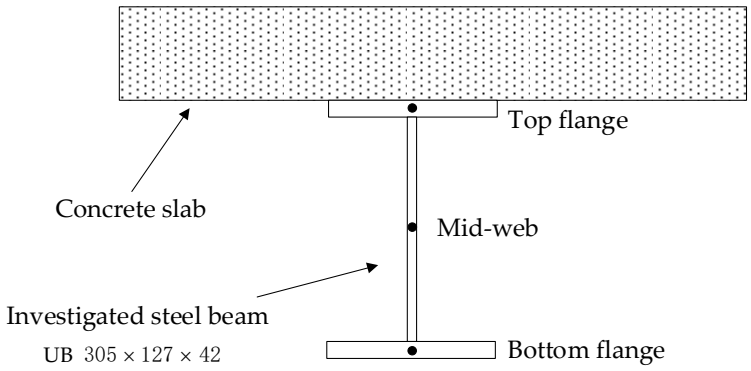


Figure 7.3. Schematic of the investigated beam cross-section for heat transfer.

Three sides of the investigated steel beam are exposed to the thermal impact in the ETFM framework, since a concrete slab is assumed to be at the top with a heat sink effect. Two-dimensional heat transfer analysis is carried out for the cross-section at the mid-span of the beam, using 35 W/m²K as the convection coefficient for fire-exposed surfaces and 0.7 as the emissivity of the steel (two coefficients recommended in [18]). Figure 7.3 is a schematic of the cross-section of the investigated steel beam, in which three temperature locations are recorded: the top flange, the mid-web, and the bottom flange.

7.2 THE 'BASE LINE SCENARIO' WITH ETFM FRAMEWORK

This section presents the case study with the 'moving Hasemi', the FIRM zone model, and the combined full version of ETFM framework under the 'base line scenario'. This scenario uses the fuel load density 570 MJ/m^2, the fire spread rate 10 mm/s, and the heat release rate per unit area 500 kW/m^2. The relevant SIFBuilder model script can be found in Appendix C.1.

Figure 7.4 - Figure 7.7 are the screenshots during the fire and heat transfer analysis in SIFBuilder for the 'base line scenario' with the combined full version of ETFM framework, showing four representative fire evolvement stages at corresponding specific time points. Figure 7.4 illustrates a stage when the fire is just ignited, the smoke starts to accumulate beneath the ceiling for the entire compartment. The smoke layer depth is 0.6 m with temperature 58 °C at 60 s. At this transient time point, the 0.6 m smoke layer depth is less than the 0.85 m barrier depth, which is the clear distance between the 3.85 m floor height and the 3 m soffit height (see Figure 7.2). Then the smoke continues to accumulate under the ceiling until it 'spills out' through the openings. Figure 7.5 shows a stabilized 'spilling out' status of the smoke layer at 2500 s, and its depth is 1.3 m which is larger than the 0.85 m barrier depth. At this stage the smoke mass flow out and the ambient air flow in through the openings are balanced. The temperature of this stabilized smoke layer is 625 °C. It is important to note that the 500 °C flashover threshold of the ETFM framework is switched off for this case study, since the theme of this section is investigating the 'travelling' thermal impact due to various fire spread rates and fuel load densities. Figure 7.6 demonstrates the fire is travelling beneath the investigated steel beam, which provides a direct exposure of the fire flames at 6000 s. Meanwhile, the smoke is still at the stabilized status with the same depth 1.3 m and temperature 625 °C. Figure 7.7 shows the time point at 8500s representing the decay stage of the travelling fire, where the fire size is decreasing, and the smoke layer depth drops to 1.2 m, smoke layer temperature decreases to 380 °C.

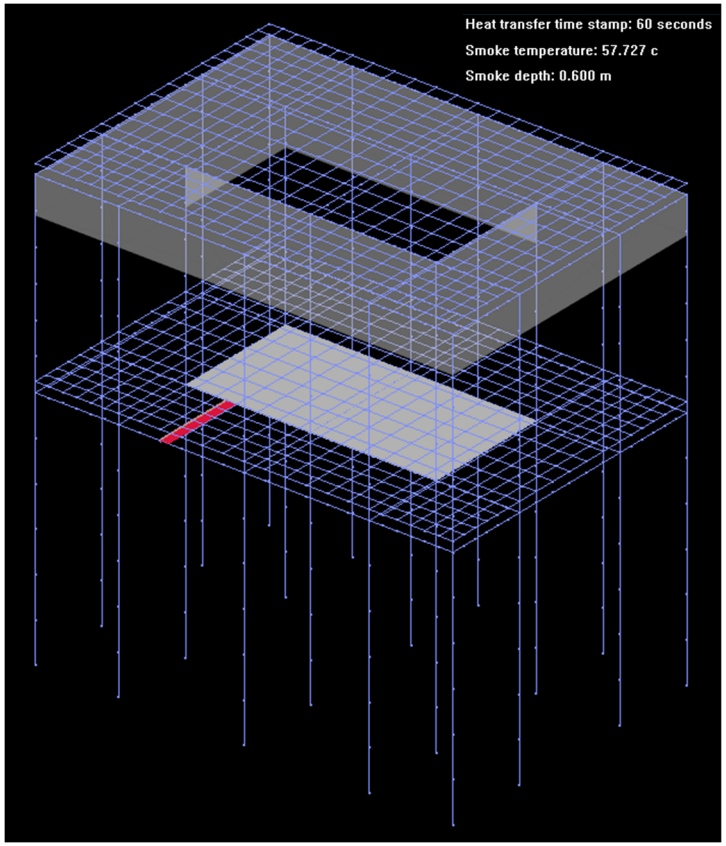


Figure 7.4. SIFBuilder visualization during fire & heat transfer analysis at 60 seconds for the 'base line scenario' of the travelling fires.

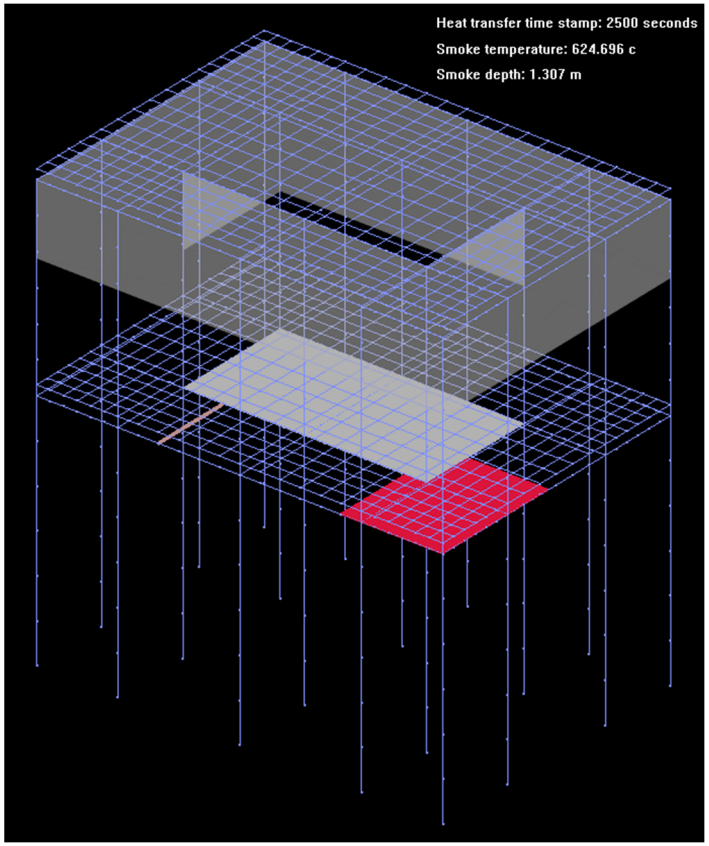


Figure 7.5. SIFBuilder visualization during fire & heat transfer analysis at 2500 seconds for the 'base line scenario' of the travelling fires.

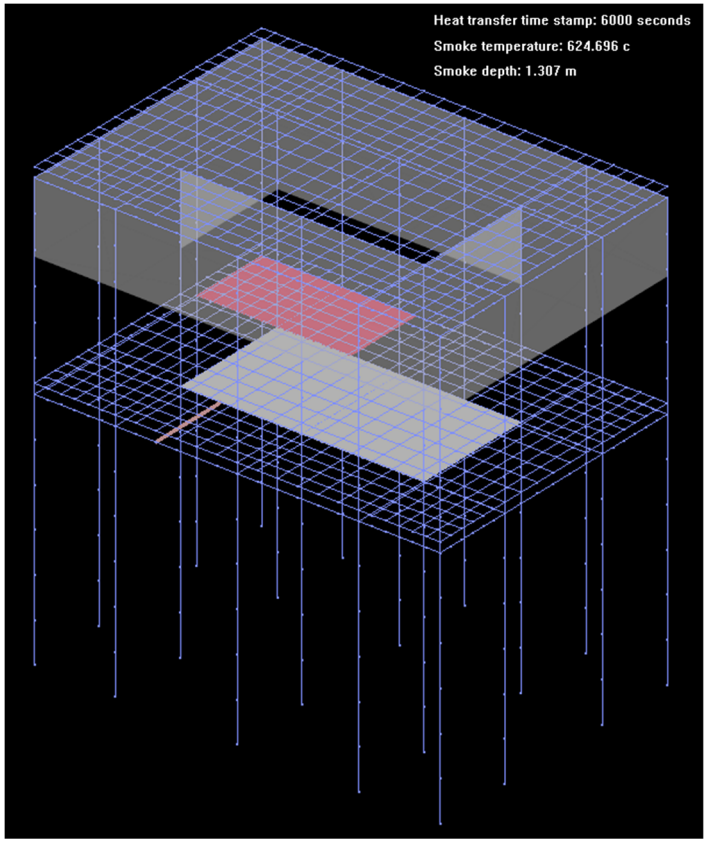


Figure 7.6. SIFBuilder visualization during fire & heat transfer analysis at 6000 seconds for the 'base line scenario' of the travelling fires.

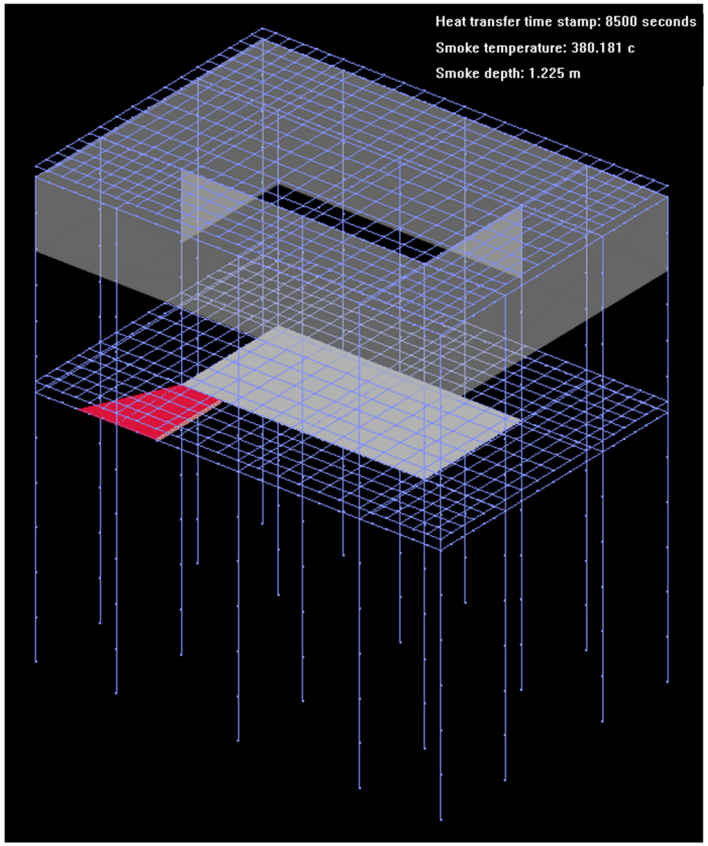


Figure 7.7. SIFBuilder visualization during fire & heat transfer analysis at 8500 seconds for the 'base line scenario' of the travelling fires.

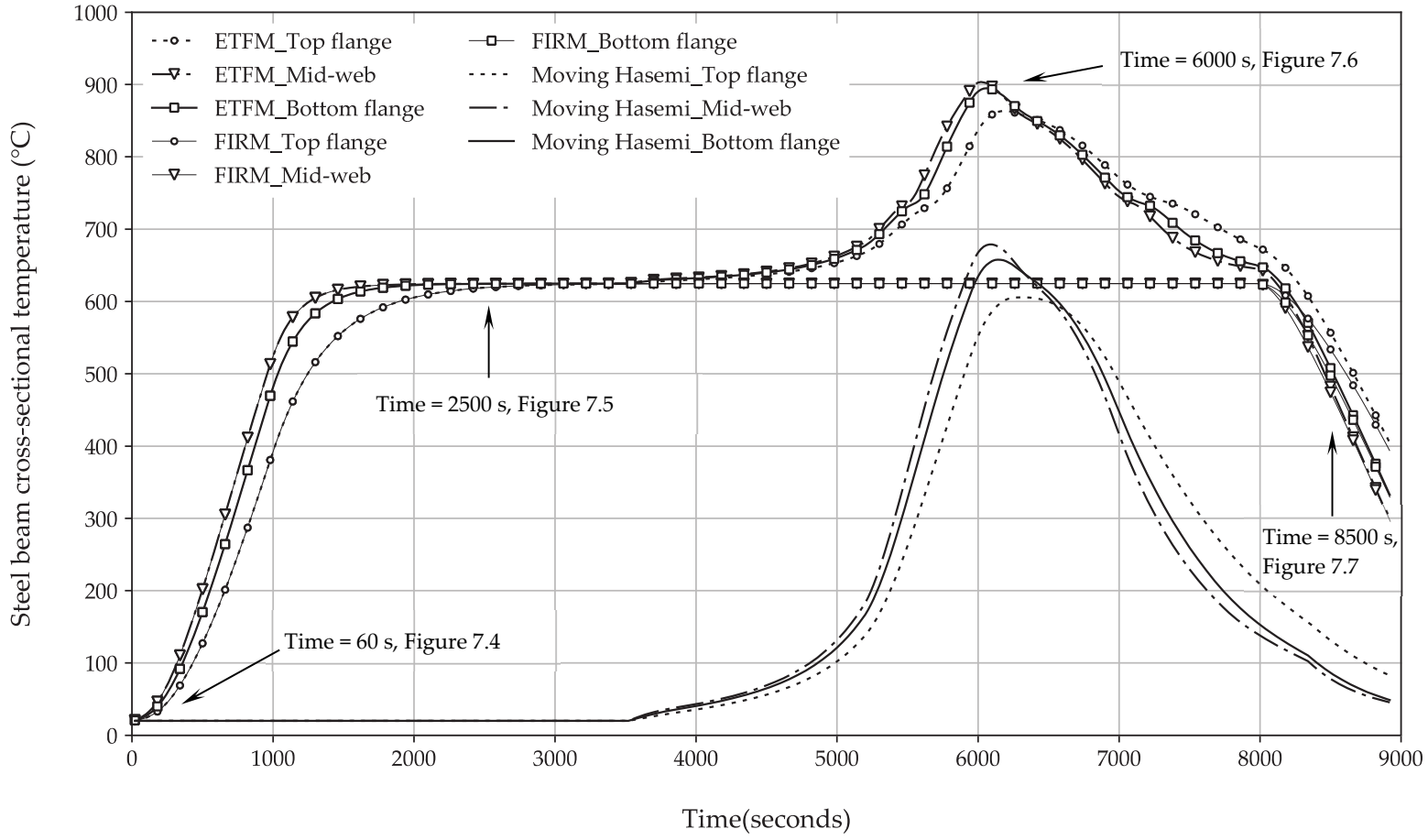


Figure 7.8. Cross-sectional temperature evolvement of the investigated steel beam, with full ETFM, FIRM zone model only, and moving Hasemi only, under the same 'base line scenario' of the travelling fires.

Figure 7.8 presents the temperature evolution of the investigated steel beam over the depth of the cross-section, including three locations at the top flange, the mid-web, and the bottom flange. The investigated beam is shown in Figure 7.1. Since the ETFM framework is essentially a combined model, with heat fluxes contribution from both the moving Hasemi's localised fire model and the FIRM zone model. Hence heat transfer results of the investigated steel member using mobilized Hasemi's model, and the FIRM zone model are also included in Figure 7.8 for comparison.

As shown in Figure 7.8, the investigated steel beam is initially preheated by the far-field smoke (e.g. time point at 60 s shown in Figure 7.4). Through-depth thermal gradient appears due to the heat sink effect of the concrete slab at this pre-heating stage. Further, the mid-web temperature is the highest compared with the temperatures at the bottom and the top flanges. Then, the thermal gradient starts to disappear due to the long duration of the stabilized smoke layer temperature (e.g. time point at 2500 s shown in Figure 7.5). The steel beam temperature is uniform as one single temperature at 625 °C at this stage, until the beam experiences near-field heating which again generates certain level of thermal gradient (e.g. time point at 6000 s shown in Figure 7.6), and the mid-web of the steel beam temperature reaches to its peak value 904 °C at 6020 s. After this time point, the steel beam temperature starts to decrease, since the fire travels away from the investigated beam, and the subsequent fire decay stage commences with decreasing fire size, smoke layer depth, and the smoke layer temperature (e.g. time point at 8500 s shown in Figure 7.7).

7.3 MORE SCENARIOS WITH SEPARATE 'MOVING HASEMI' MODEL AND FIRM ZONE MODEL

The fire development and the cross-sectional temperature evolvement of the investigated steel member, are presented in the previous section under the 'base line scenario' (fuel load density 570 MJ/m^2, fire spread rate 10 mm/s, and heat release rate per unit area 500 kW/m^2.). More fire scenarios would be generated with changing fire spread rate or fuel load density, in the range of 1.6-15 mm/s and 100-780 MJ/m^2 according to the values recommended in Chapter 4, but keeping the other value as a constant from the 'base line scenario'.

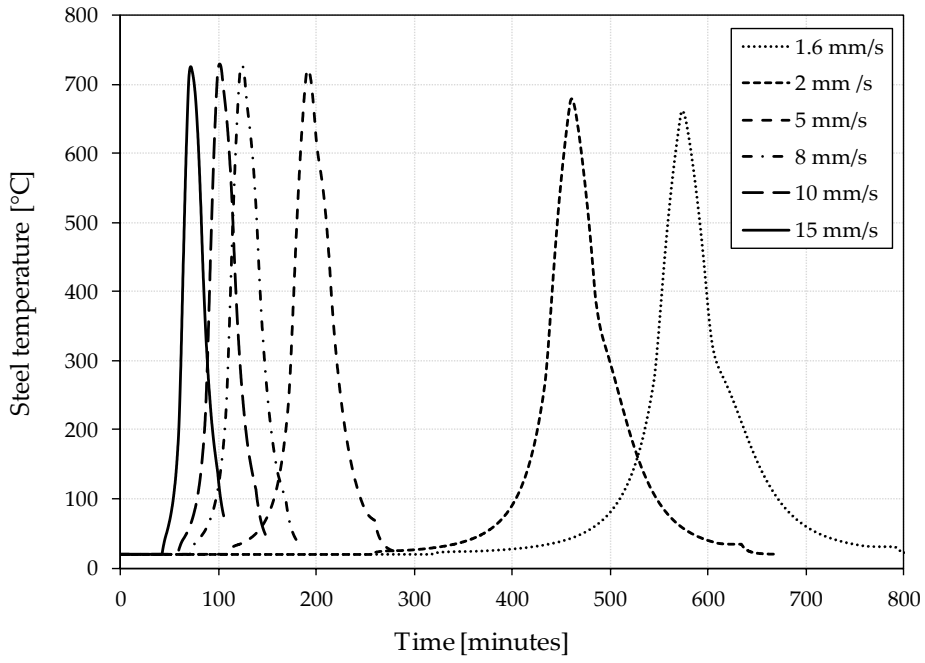


Figure 7.9. Heat transfer results from mobile Hasemi's fire model contribution, with various spread rates ranging from v = 1.6 mm/s to v = 15 mm/s.

In Figure 7.9, apart from the longer fire duration generated when the smaller fire spread rate v is used, the travelling fire scenarios with spread rates from 5 mm/s to 15 mm/s produce similar thermal impact in terms of the maximum steel temperatures. However, the two 'slow' fires with spread rates 1.6 mm/s and 2 mm/s produce relatively lower steel temperatures. The reason is because the fire HRR is calculated based on fire area, and fire area is a resultant of fire spread rate and burning rate of the fuel. Hence, although 'slow' fires have more time to heat up the steel member, they produce lower thermal impact due to smaller fire areas and HRR generated. Figure 7.10 illustrates that

longer fire durations and higher thermal impact are generated if higher fuel load densities are used.

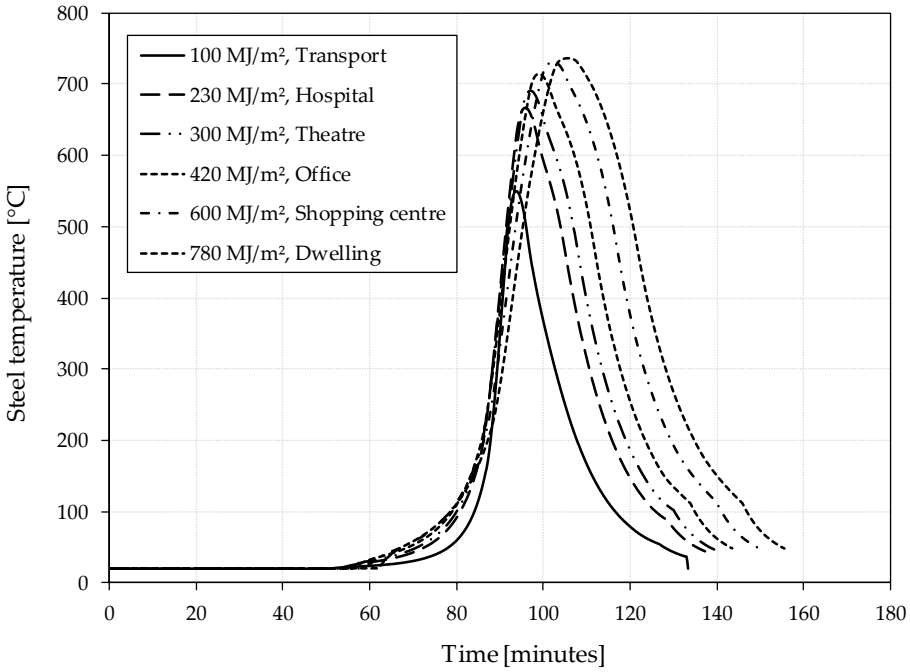


Figure 7.10. Heat transfer results from mobile Hasemi's fire model contribution, with various fuel load densities ranging from $q_{f,k}$ = 100 MJ/m² to $q_{f,k}$ = 780 MJ/m².

Figure 7.11 demonstrates that the travelling fire scenarios with higher fire spread rates (e.g. 10 mm/s, 15 mm/s) generate higher smoke layer temperature, with quicker temperature increase rate. The reason is because the energy conservation equation, from the FIRM zone model for calculating the transient smoke layer temperature increase, is directly dependent on the HRR, which decides the amount of energy to be 'pumped' into the smoke layer at each time step. The same as discussed earlier, the HRR is calculated based on fire area, and fire area is a resultant of fire spread rate and burning rate of the fuel. Therefore, 'fast' fires produce higher thermal impact due to bigger fire areas and HRR generated. Figure 7.12 illustrates that the smoke layer can become steady within 200 s for all the travelling fire scenarios. Figure 7.13 shows that the travelling fire scenarios with higher fuel load densities generate higher smoke layer temperatures. Again, it is directly dependent on the HRR which decides the amount of energy to be 'pumped' into the smoke layer at each time step, thus depending on the spread rate and burning rate. Larger fuel load densities would generate slower fire burning edge, thus a larger fire area would be produced. Therefore, 'dense' fires produce higher thermal impact due to bigger

fire areas and HRR generated. Figure 7.14 shows the smoke depth evolution, which is nearly independent of fuel load densities in the initial spread phase of the fire. Figure 7.15 and Figure 7.16 are the steel temperatures via heat transfer analysis in SIFBuilder.

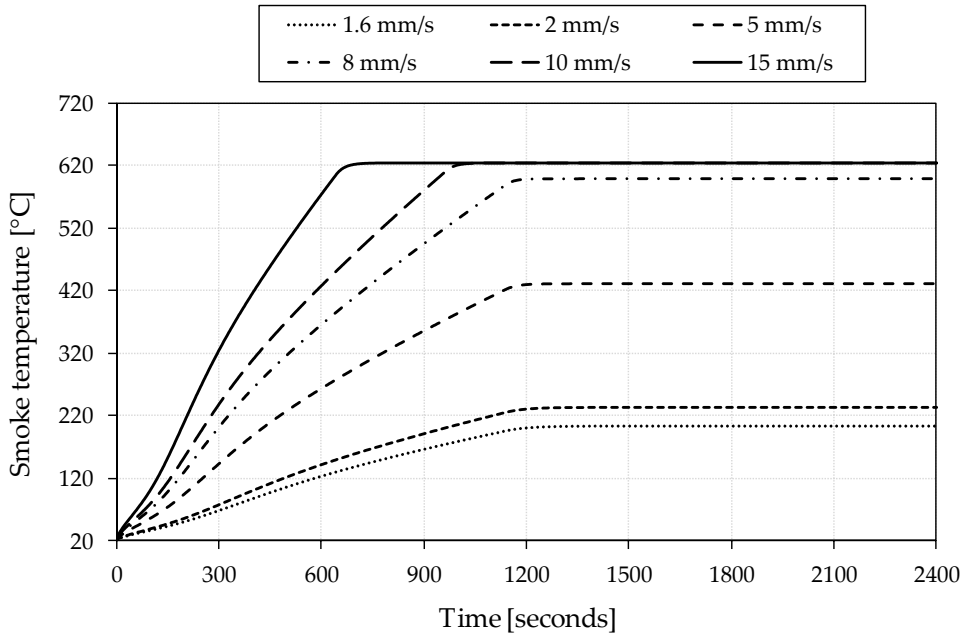


Figure 7.11. Smoke temperature evolvement with various spread rates range from v = 1.6 mm/s to v = 15 mm/s.

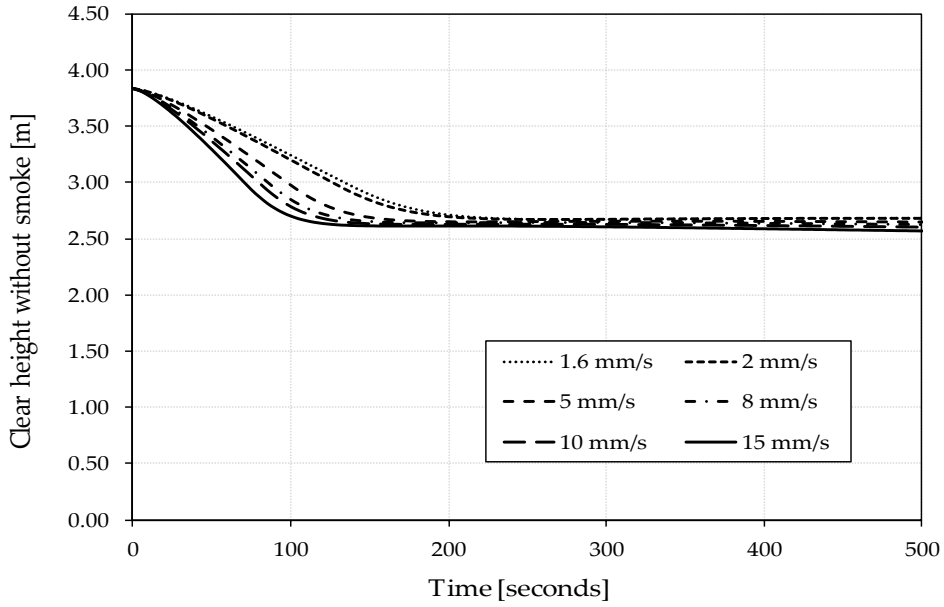


Figure 7.12. Height of zone free of smoke with various spread rates range from v = 1.6 mm/s to v = 15 mm/s.

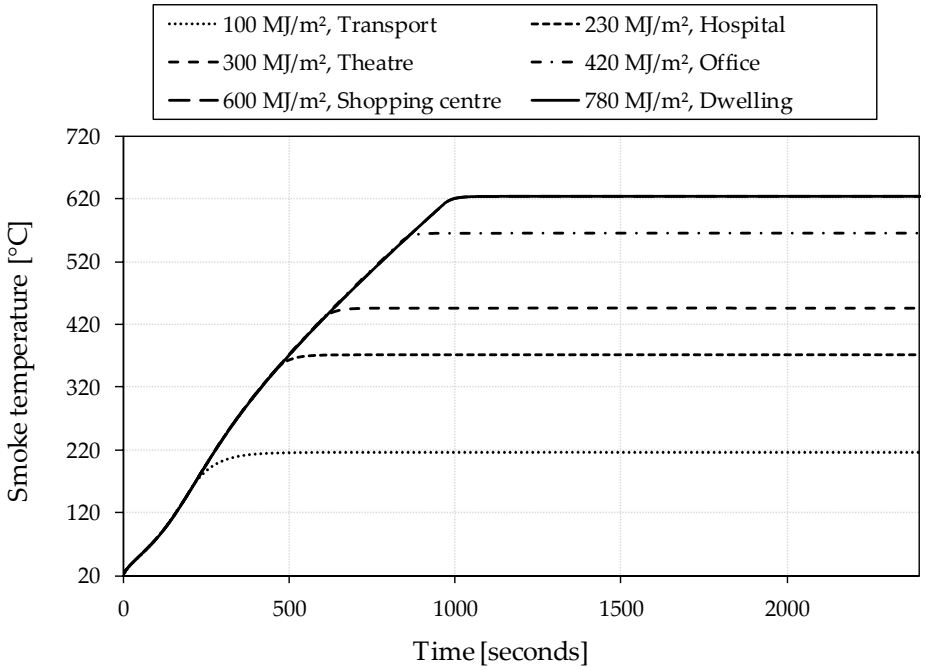


Figure 7.13. Smoke temperature evolvement with various fuel load densities range from $q_{f,k}$ = 100 MJ/m² to $q_{f,k}$ = 780 MJ/m².

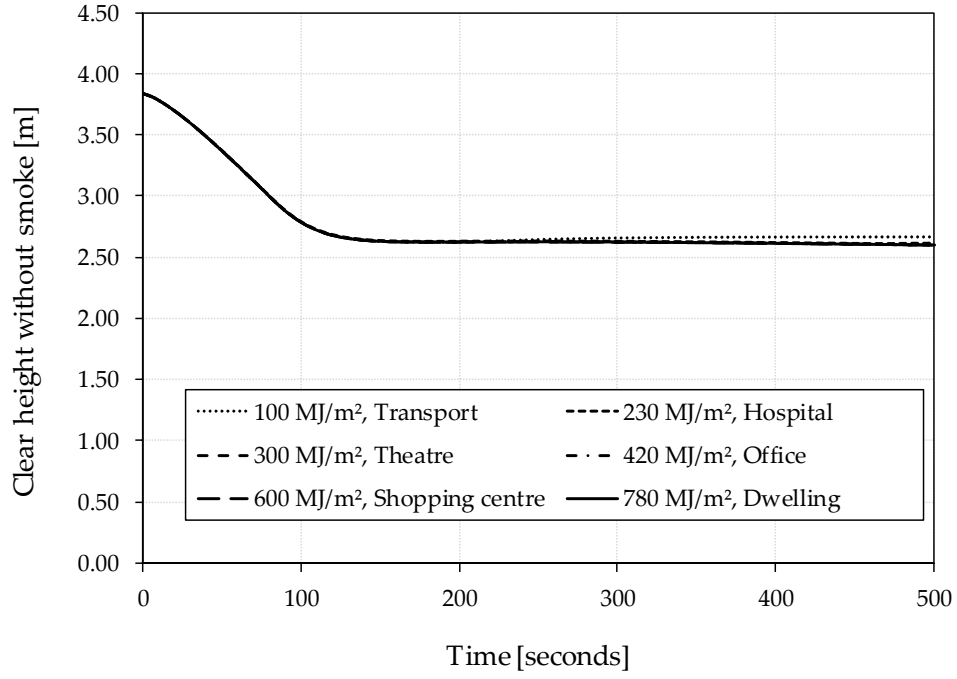


Figure 7.14. Height of zone free of smoke with various fuel load densities range from $q_{f,k}$ = 100 MJ/m² to $q_{f,k}$ = 780 MJ/m².

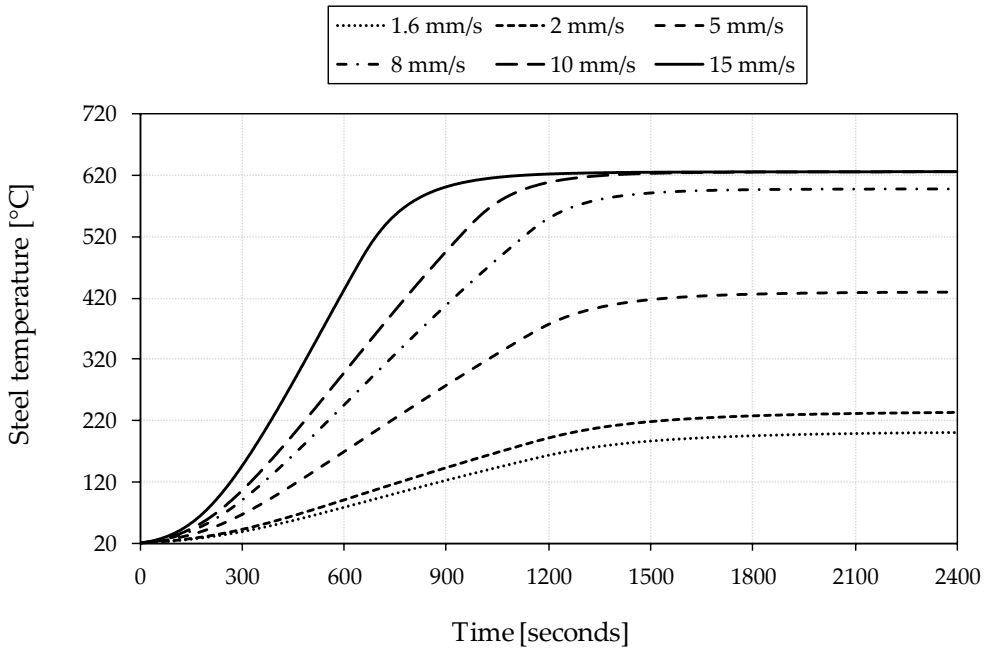


Figure 7.15. Heat transfer results from FIRM zone model contribution with various spread rates ranging from v = 1.6 mm/s to v =15 mm/s.

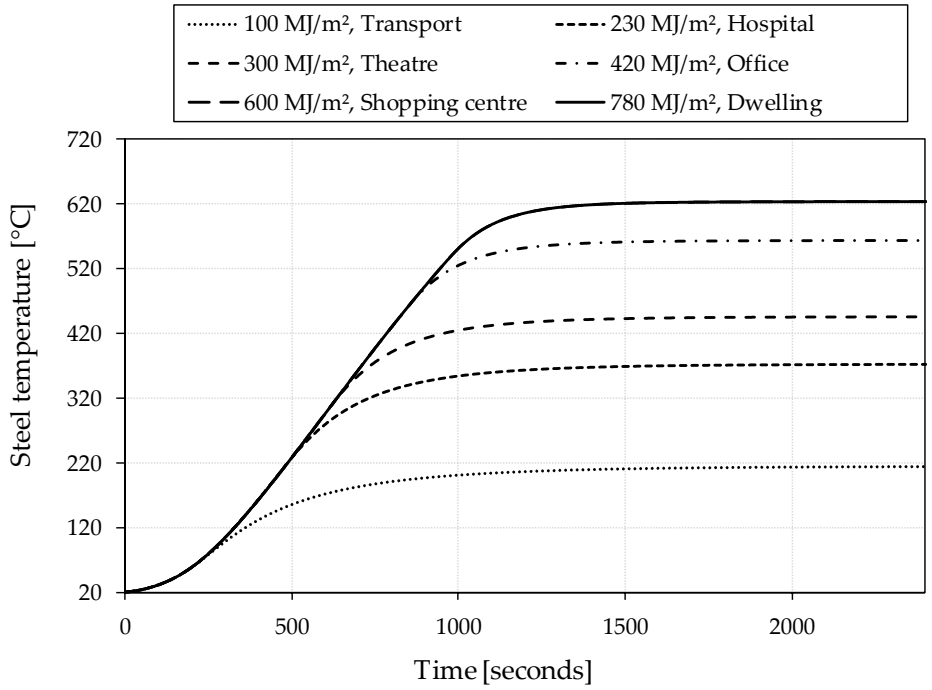


Figure 7.16. Heat transfer results from FIRM zone model contribution with various fuel load densities range from $q_{f,k}$ = 100 MJ/m² to $q_{f,k}$ = 780 MJ/m².

Table 7.3 presents the sensitivity of the results to selected design parameters relevant to performance-based structural fire design. Although these results are based on a single assumed scenario, they do provide some insights into how the key variables such as travelling fire spread rate and fuel load densities might affect the structural thermal response. In the ETFM framework, these two variables are the essential inputs for the determination of HRR [19] and given the uncertainty in their values it is important to characterise the sensitivity of the thermal and structural response to their assumed values.

Table 7.3. Summaries of the travelling fire thermal impact due to v, and $q_{f,k}$.

Thermal impact	v, $q_{f,k}$			
	Low fire spread Rate	**High fire spread Rate**	**Low fuel load density**	**High fuel load density**
Steel temperature from moving Hasemi's model	Slightly lower	Slightly higher	Lower	Higher
Thermal impact duration from moving Hasemi's model	Slightly longer	Slightly shorter	Shorter	Longer
Steel temperature from FIRM zone model	Much lower	Much higher	Much lower	Much higher
Smoke layer temperature from FIRM zone model	Much lower	Much higher	Much lower	Much higher
Time to form a steady smoke layer	Slightly slower	Slightly quicker	Not sensitive	Not sensitive
Smoke layer depth	Slightly thinner	Slightly thicker	Not sensitive	Not sensitive

7.4 MORE SCENARIOS WITH FULL ETFM FRAMEWORK

Unlike performing the same case study with 'moving Hasemi' and the FIRM zone model separately in the previous section, the full version of ETFM is utilised here. Their thermal response implications on the subsequent structural responses (e.g. the temperature rise induced thermal expansion axial force, the change of through depth induced thermal bowing bending moment) are also discussed this section.

Figure 7.17 shows that the travelling fire scenarios with higher fuel load densities generate higher smoke layer temperatures during the pre-heating stage. It is directly dependent on the HRR which decides the amount of energy to be 'pumped' into the smoke layer at each time step, thus depending on the spread rate and burning rate. Larger fuel load densities would generate slower fire burning edge, thus a larger fire area would be produced. Therefore, 'dense' fires produce higher thermal impact due to bigger fire areas and HRR generated. This reason is also applicable to the near-field heating stage, in which longer fire durations and higher peak temperatures are generated if higher fuel load densities are used, as shown in Figure 7.17.

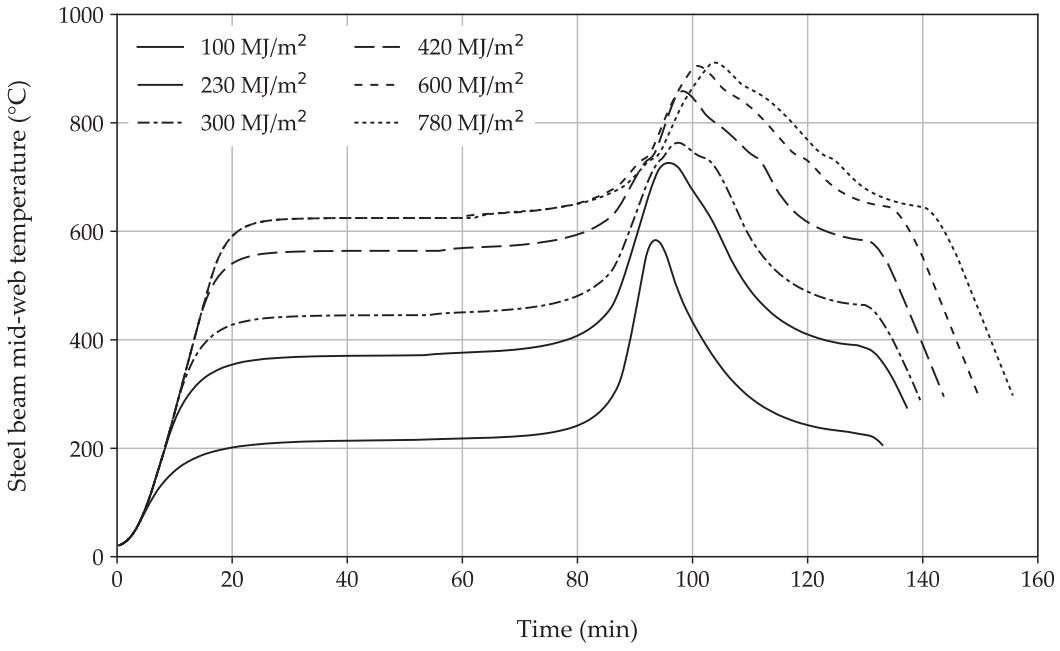


Figure 7.17. Temperature evolvement at steel beam mid-web with various fuel load densities ranging from $q_{f,k}$ = 100 MJ/m² to $q_{f,k}$ = 780 MJ/m², and constant v = 10 mm/s, RHR_f = 500 kW/m².

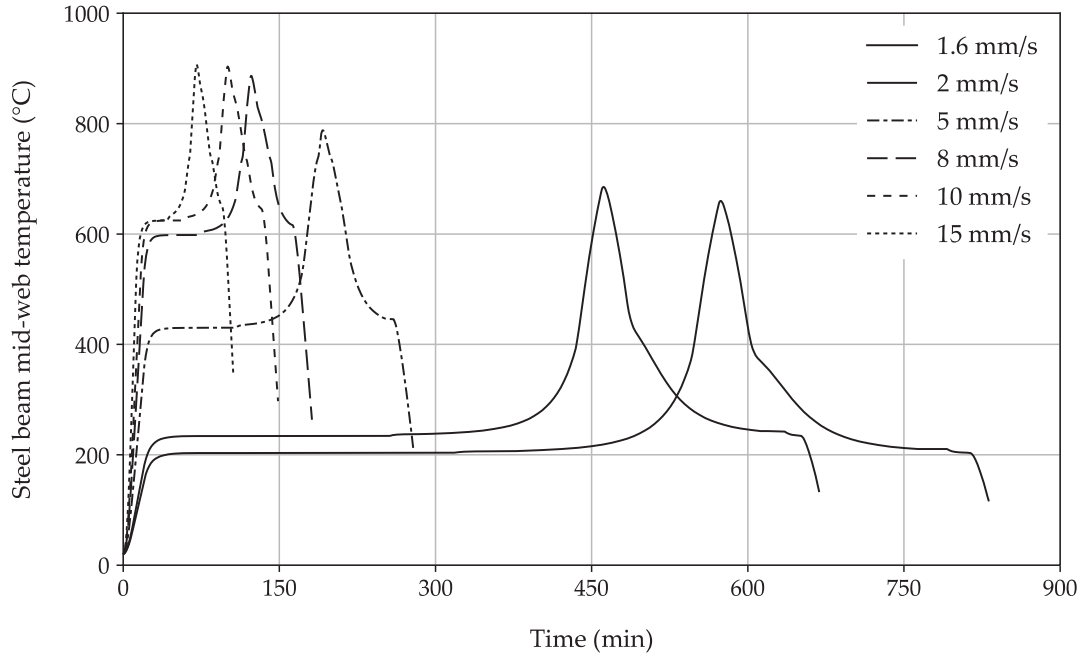


Figure 7.18. Temperature evolvement at steel beam mid-web with various fire spread rates ranging from v = 1.6 mm/s to v = 15 mm/s, and constant $q_{f,k}$ = 570 MJ/m², RHR_f = 500 kW/m².

Figure 7.18 demonstrates that the travelling fire scenarios with higher fire spread rates (e.g. 10 mm/s, 15 mm/s) generate higher smoke layer temperature, with quicker rates of temperature increase. Again, the reason is also because the energy conservation equation from the FIRM zone model for calculating the transient smoke layer temperature increase. It is directly dependent on the HRR, which decides the amount of energy to be 'pumped' into the smoke layer at each time step. The same as discussed earlier, the HRR is calculated based on fire area, and fire area is a resultant of fire spread rate and burning rate of the fuel. Therefore, 'fast' fires produce higher thermal impact due to bigger fire areas and greater HRR generated. Moreover, apart from the longer fire duration generated when the smaller fire spread rate, v, is used, the travelling fire scenarios with spread rates from 5 mm/s to 15 mm/s produce similar thermal impact, in terms of the 'net value' of steel temperature increase due to the near-field heating (all around 300 °C). In addition, the proportion of the near-field heating duration to the entire fire duration with 'fast' fires (e.g. 10 mm/s, 15 mm/s), is larger than the proportion with the 'slow' fires (e.g. 1.6 mm/s, 2 mm/s). For instance, the 15 mm/s near-field heating takes near half of the entire fire duration. This also results in 'fast' fires generating steeper temperature drop

during the fire decay stage, such as the cases with fire spread rates ranging from 8 mm/s - 15 mm/s generating the cooling part of the temperature curves with relatively sudden drop.

7.4.1 Temperature Histories of the Sequential Beams

Instead of only performing parametric studies on one single structural member (i.e. the investigated steel beam shown in Figure 7.1), more temperature histories of the steel beams of the large compartment are investigated in this section. For the total of 29 fire scenarios as mentioned in Table 7.2, heat transfer analysis is carried out for all the beam members in this compartment except the ones involved with the 'core' (see Figure 7.19). Hence each fire scenario is related to 24 steel beams, leading to the total number of heat transfer analysis of 696. To choose the most representative heat transfer results, the temperature histories of the structural members which are right above the fire trajectory are plotted in the following figures.

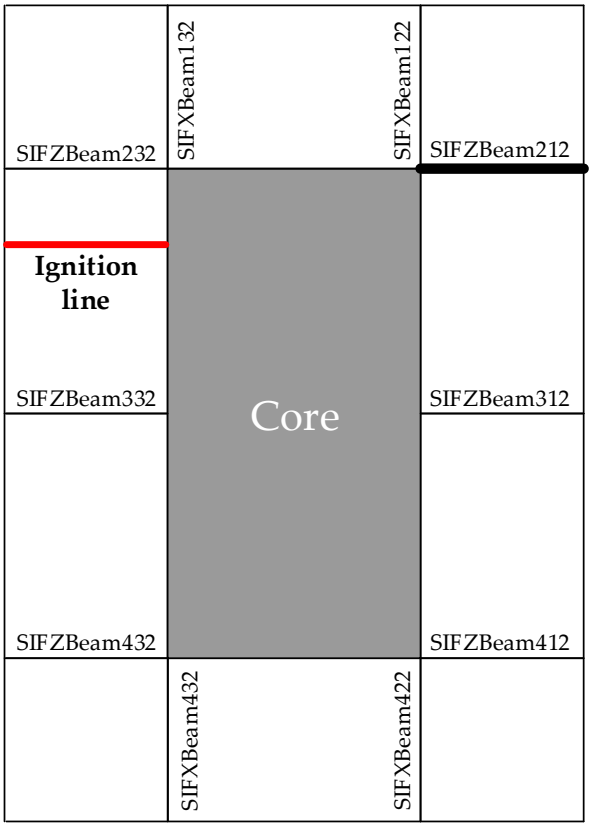


Figure 7.19. Schematic of the investigated sequential beams and their corresponding tags.

These sequential beam temperature histories for all 29 travelling scenarios can be found in Appendix C.2. Figure 7.19 schematically shows the tags of these investigated sequential beams and their locations in the large compartment. The way of naming these beam tags, follows the same naming pattern as the output of SIFBuilder software framework.

Figure 7.20 presents temperature histories at the mid-web of the sequential beams, which are right above the fire trajectory under the 'base line scenario' with v = 10 mm/s, $q_{f,k}$ = 570 MJ/m², RHR_f = 500 kW/m². Under this fire scenario the thermal impact on each structural member is very similar, in terms of the fire durations and peak temperatures (around 900 °C), except for the member SIFZBeam232 which is located at top left of the compartment. This member experiences relatively lower peak temperature, as the fire is at the decay stage when the near-field fire plume travels beneath the member, with decreasing fire size and HRR (see Figure 7.7). In addition, it can be seen that far-field smoke contributes more 'heat' compared with the temperature increase due to near-field fire plume, as a high temperature 'plateau' around 620 °C shown in Figure 7.20.

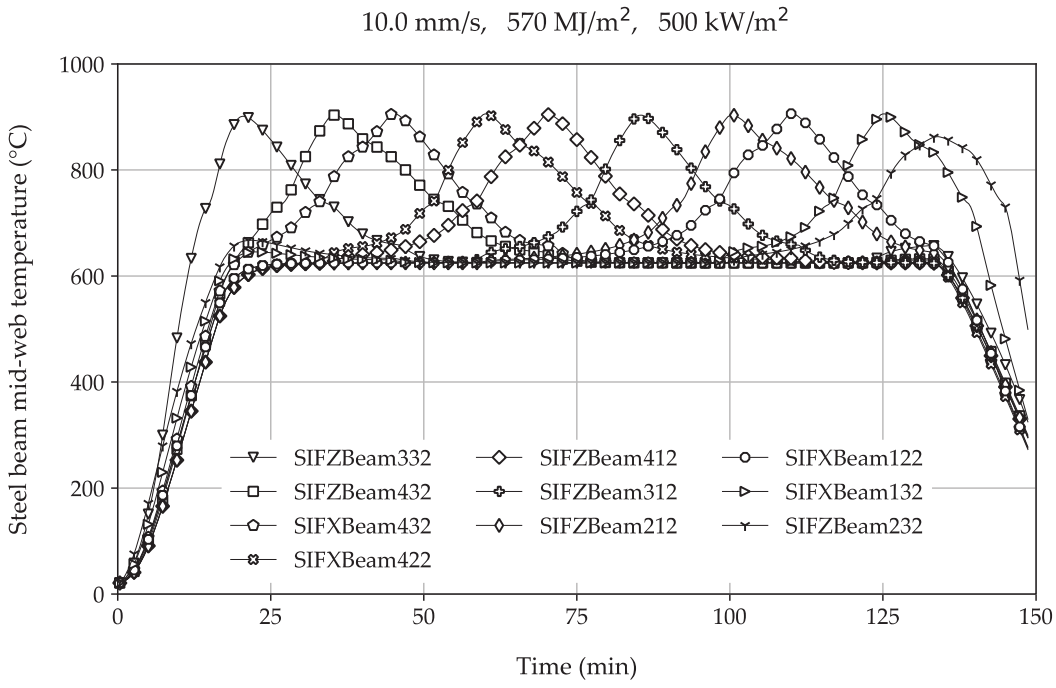


Figure 7.20. Temperature histories of the sequential beams right above the fire trajectory, under 'base line scenario' with v = 10 mm/s, $q_{f,k}$ = 570 MJ/m², RHR_f = 500 kW/m².

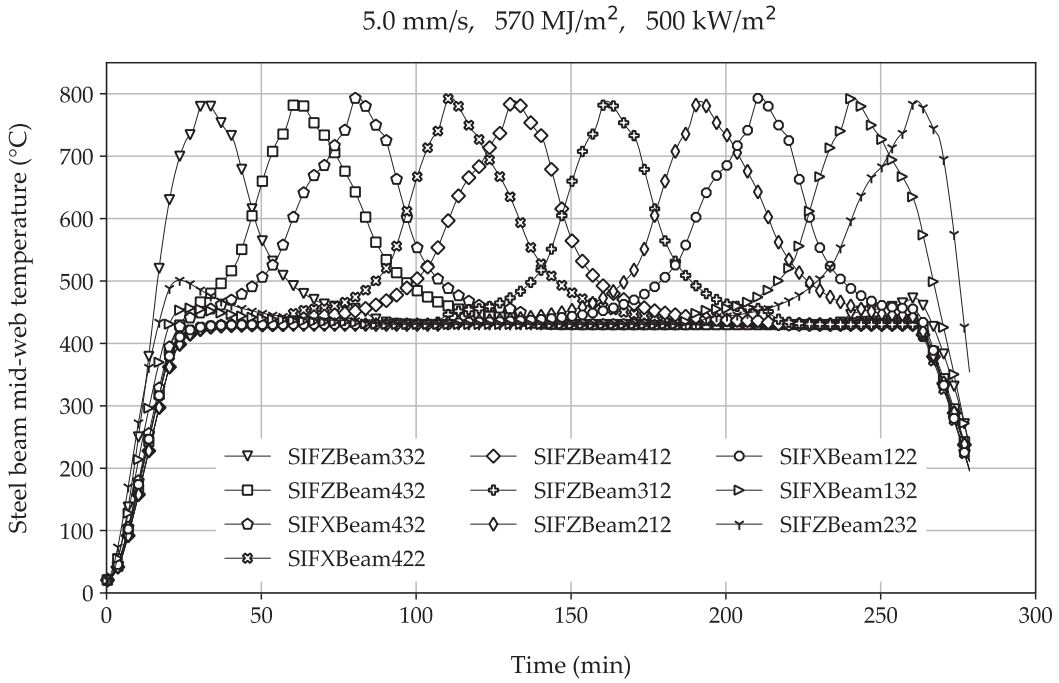


Figure 7.21. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 5.0 mm/s, $q_{f,k}$ = 570 MJ/m², RHR_f = 500 kW/m².

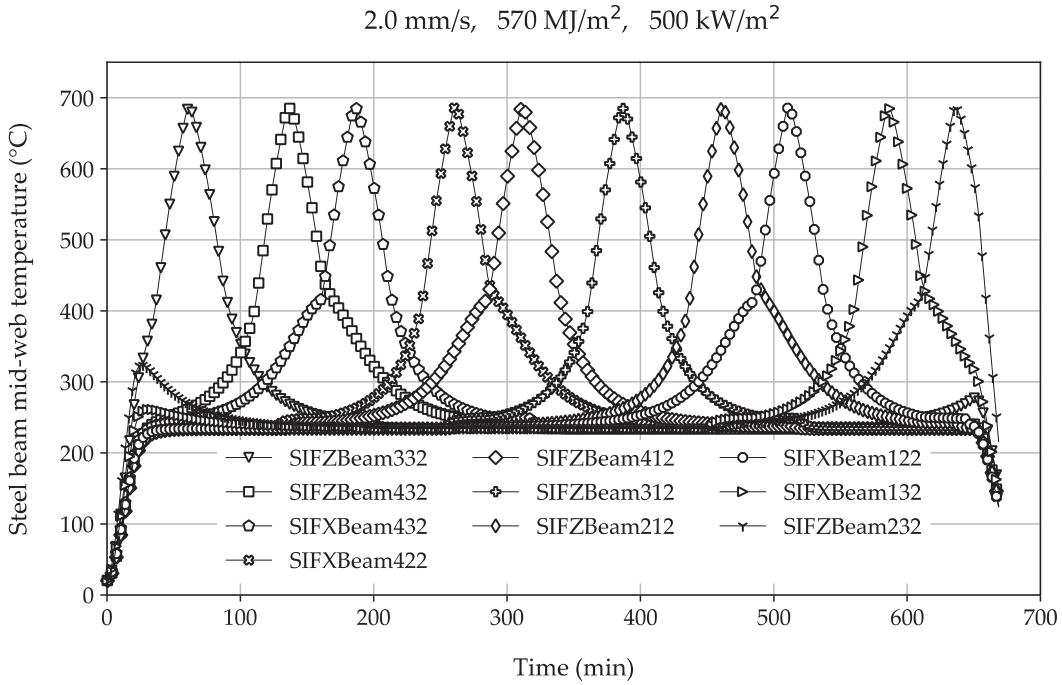


Figure 7.22. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 2.0 mm/s, $q_{f,k}$ = 570 MJ/m², RHR_f = 500 kW/m².

Figure 7.21 and Figure 7.22 show the temperature histories of these sequential beams with slower fire spread rates. The peak temperatures of these beams in each scenario are still very close. However, the 'smoke plateau' reduces when the fire spread rate decreases, as shown in Figure 7.20 - Figure 7.22. It implies that with 'slow' fires, the near-field fire plume brings more detrimental thermal impact compared with the impact from far-field smoke.

Figure 7.23 illustrates the temperature histories of these sequential beams under an extreme fire scenario, with v = 1.6 mm/s, $q_{f,k}$ = 100 MJ/m^2, RHR_f = 500 kW/m^2. Under this fire scenario, the fire would spread unusually slowly with a very small fire size. The peak temperatures of these beams in this scenario are still very close. It can be seen that the thermal impact due to the near-field fire plume is still larger than the impact from the far-field smoke. This situation would remain the same, even if with larger fuel load densities (i.e. larger fire sizes), as shown in Figure 7.24 and Figure 7.25. It can be concluded that for 'slow' fires, the near-field fire plume brings more detrimental thermal impact compared with the impact from far-field smoke. However, for 'fast' fires, the far-field smoke brings more detrimental thermal impact (see Figure 7.20 - Figure 7.22).

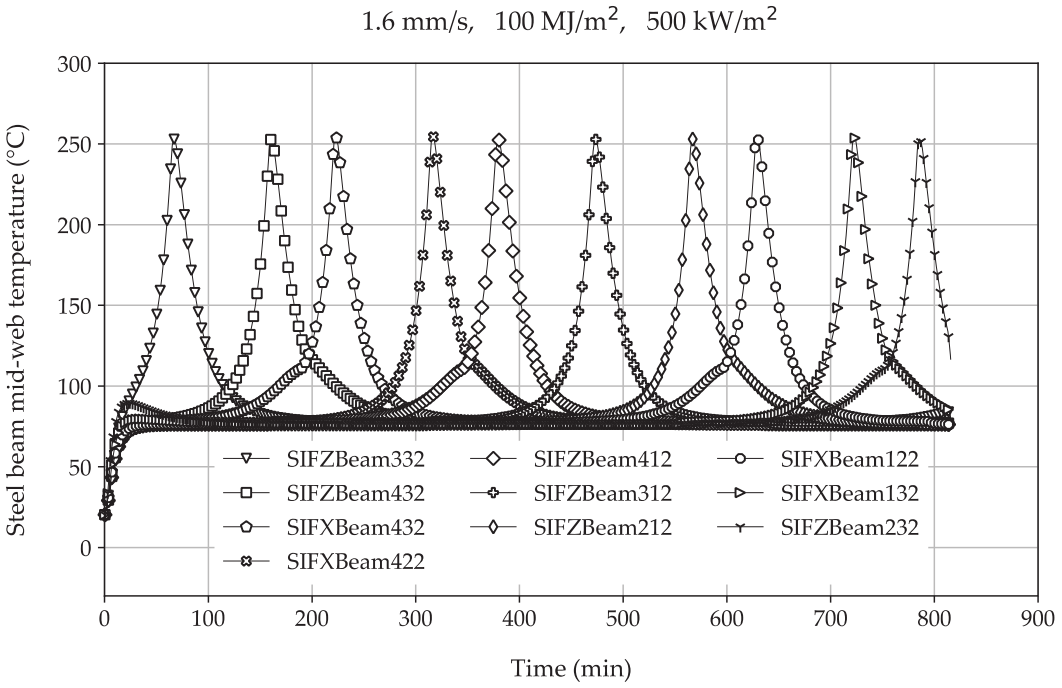


Figure 7.23. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 1.6 mm/s, $q_{f,k}$ = 100 MJ/m^2, RHR_f = 500 kW/m^2.

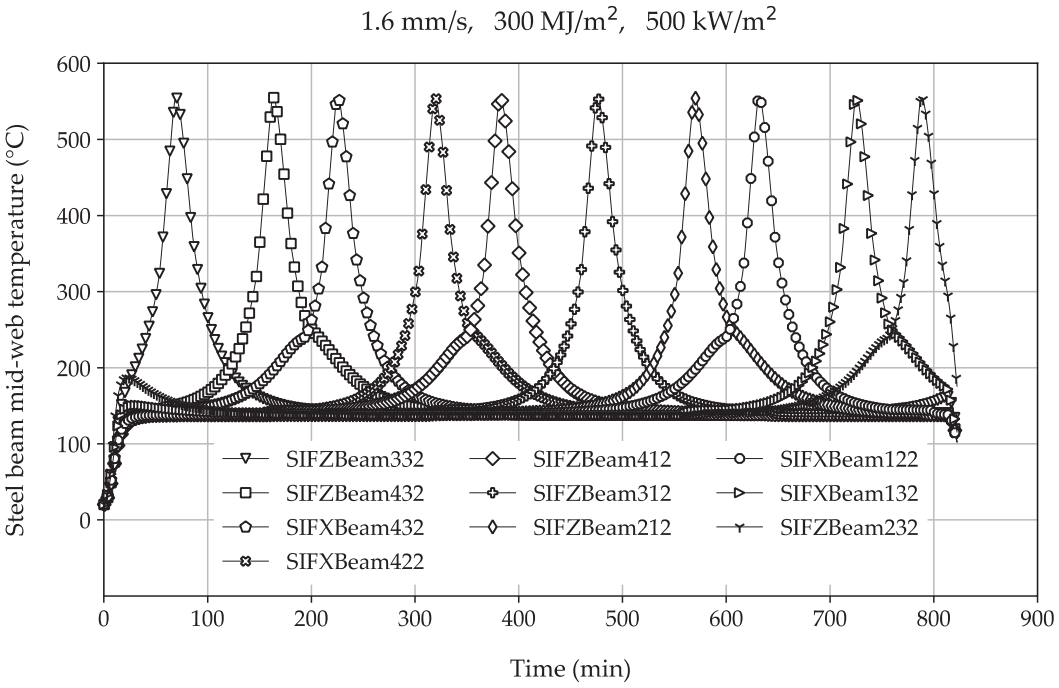


Figure 7.24. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 1.6 mm/s, $q_{f,k}$ = 300 MJ/m², RHR_f = 500 kW/m².

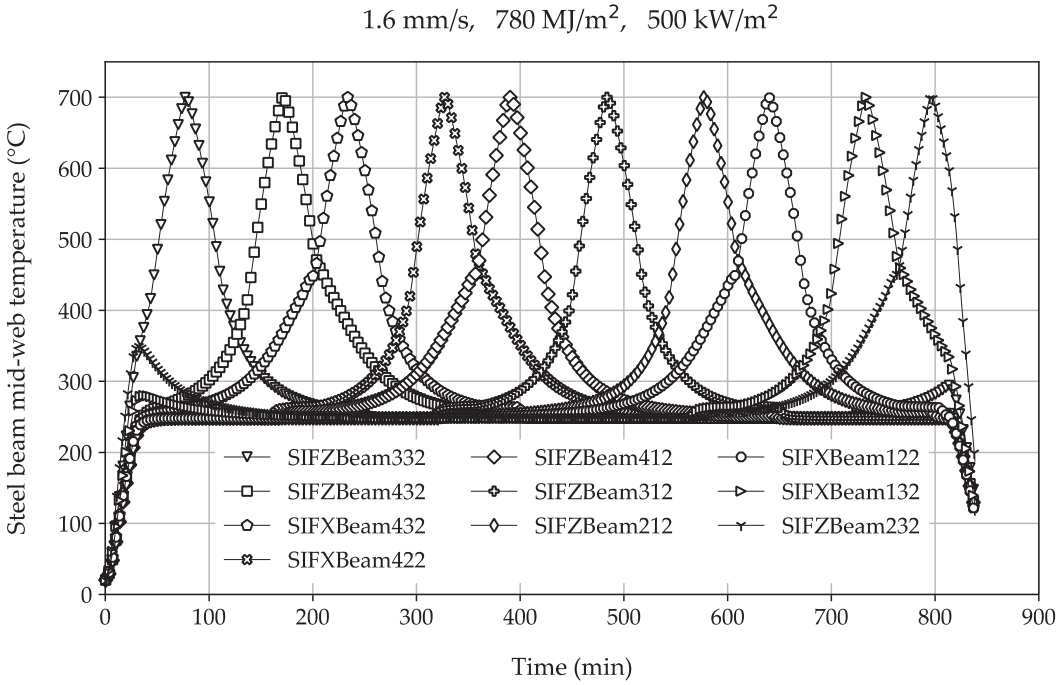


Figure 7.25. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 1.6 mm/s, $q_{f,k}$ = 780 MJ/m², RHR_f = 500 kW/m².

Figure 7.26 demonstrates the temperature histories of these sequential beams under another extreme fire scenario, with v = 15 mm/s, $q_{f,k}$ = 100 MJ/m², RHR_f = 500 kW/m². Under this fire scenario, the fire would spread unusually fast. The peak temperatures of these beams in this scenario are fluctuating. This is because some of the structural members are not given enough time to be heated up by the 'fast travelling' near-field fire plume. Hence the relative locations of the structural members in the compartment are having an impact on the results. Figure 7.27 and Figure 7.28 confirm this explanation, as this peak temperature fluctuation diminishes when the fire spread rates become lower.

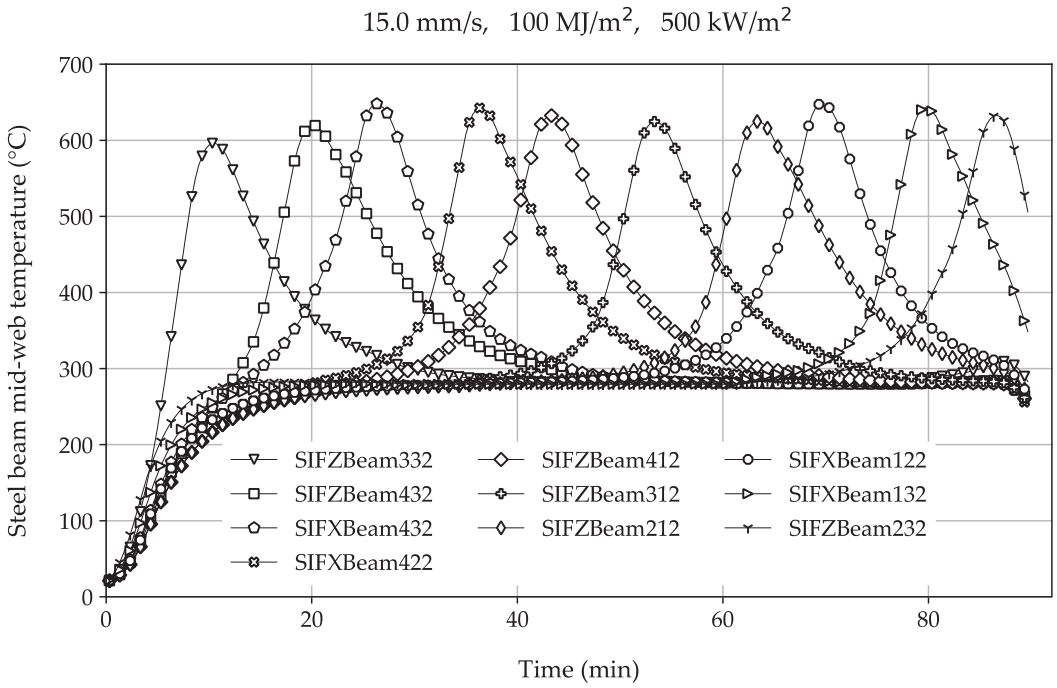


Figure 7.26. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 15.0 mm/s, $q_{f,k}$ = 100 MJ/m², RHR_f = 500 kW/m².

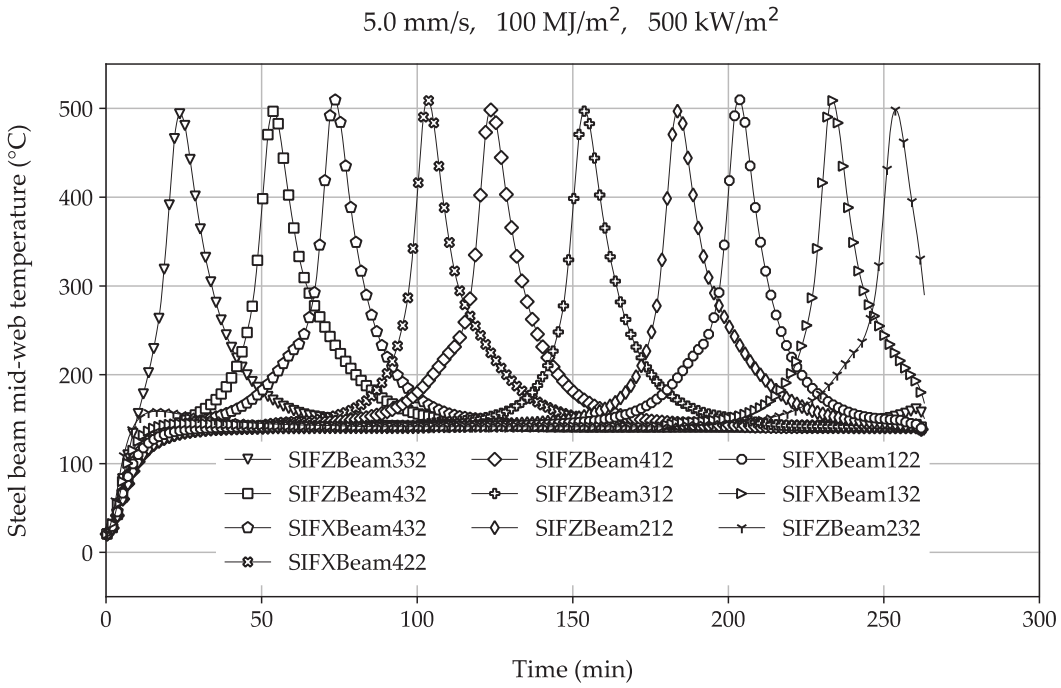


Figure 7.27. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 5.0 mm/s, $q_{f,k}$ = 100 MJ/m², RHR_f = 500 kW/m².

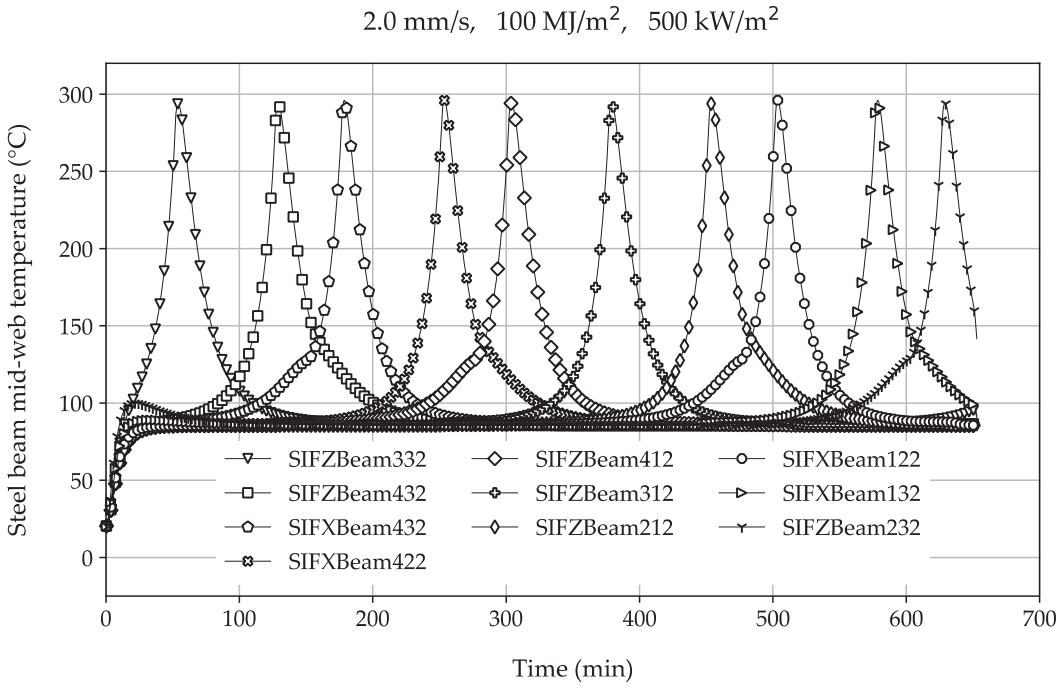


Figure 7.28. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 2.0 mm/s, $q_{f,k}$ = 100 MJ/m², RHR_f = 500 kW/m².

7.4.2 The ETFM Framework in the Time Design Domain

The structural fire design in the time domain generally refers to the criterion of failure time of a structural element under the standard fire being greater than the design fire duration (e.g. one hour, two hours, etc.). By this definition, studying the ETFM framework in the time design domain seems inappropriate, as the fire severity input is required to be a standard fire, rather than travelling fires. However, it is still worth to study the effect of various fire spread rates (v) and fuel load densities ($q_{f,k}$), on the time for a specified beam to reach its peak temperature, total travelling fire duration, and the ratio of these two values.

In the following sections a series of 3D plots are generated, to interpret the resultant impact due to various v and $q_{f,k}$. Hence Table 7.2 is replotted in a 3D view with 29 sampling points, representing the 29 travelling fire scenarios, for convenience to reference as shown in Figure 7.29.

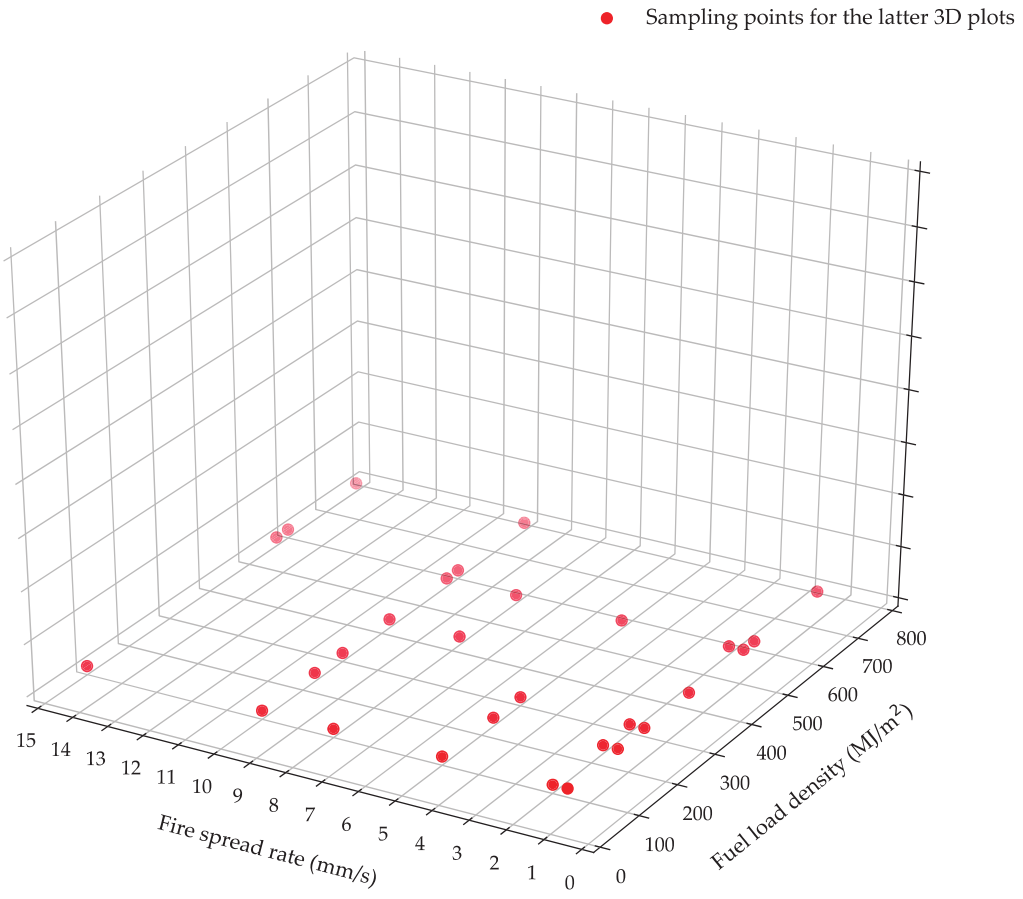


Figure 7.29. 3D view of the 29 travelling fire scenarios shown as 29 sampling points, with various fire spread rates and fuel load densities.

Figure 7.30 shows the time to reach peak temperature of the investigated steel beam (see Figure 7.1), under different travelling fire scenarios with various v and $q_{f,k}$. This figure demonstrates that around one third of the travelling fire scenarios (v = 10 mm/s - 15 mm/s) heat up the beam to its peak temperature, taking around 100 minutes. It also suggests that the fire spread rate is a more discriminating factor rather than the fuel load density, on affecting the time to reach the beam peak temperatures. Figure 7.31 is the total travelling fire duration under different travelling fire scenarios, which follows a very similar trend as Figure 7.30.

Figure 7.32 is the ratio of the time for reaching peak temperature to the fire total duration, of the investigated beam under different travelling fire scenarios with various v and $q_{f,k}$. This value ranges from 0.65 to 0.72, in which higher v and $q_{f,k}$ (i.e. larger fire sizes and HRR) would cause the beam to reach its peak temperature more quickly. This contour plot implies v and $q_{f,k}$ are 'equally' discriminating factors on the time to reach its peak, rather than what is suggested in Figure 7.30.

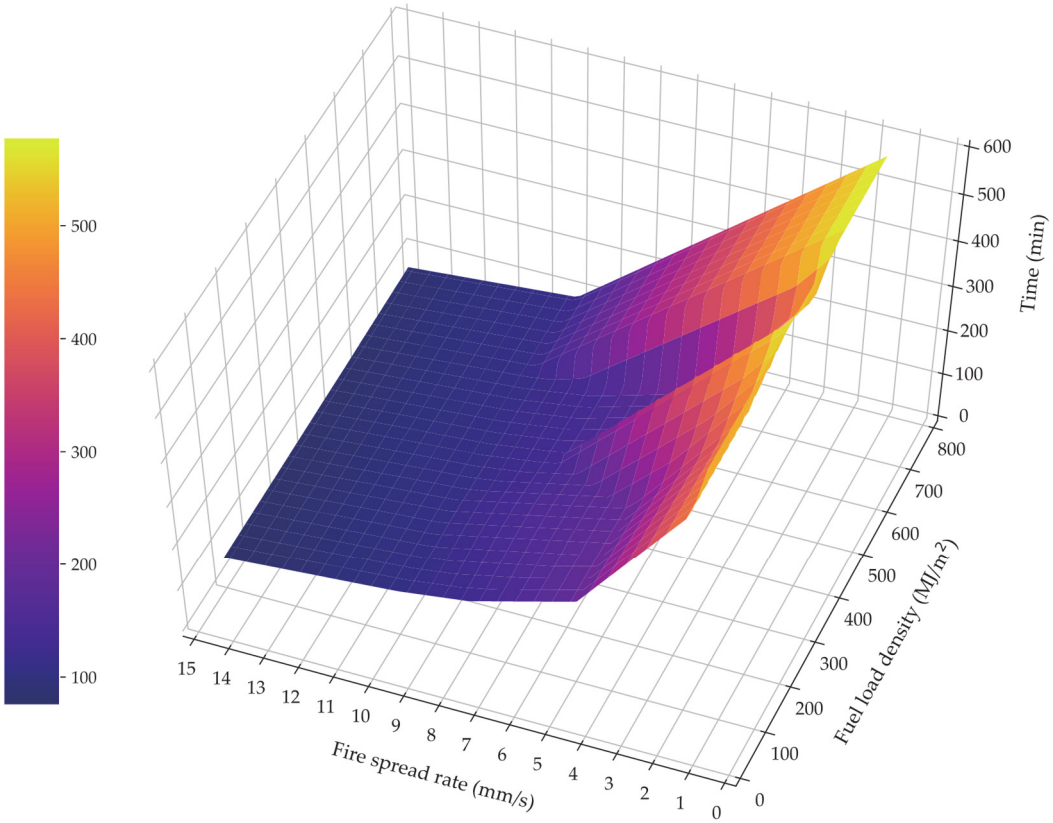


Figure 7.30. Time to reach the peak temperature of the investigated steel beam, with various fire spread rates and fuel load densities.

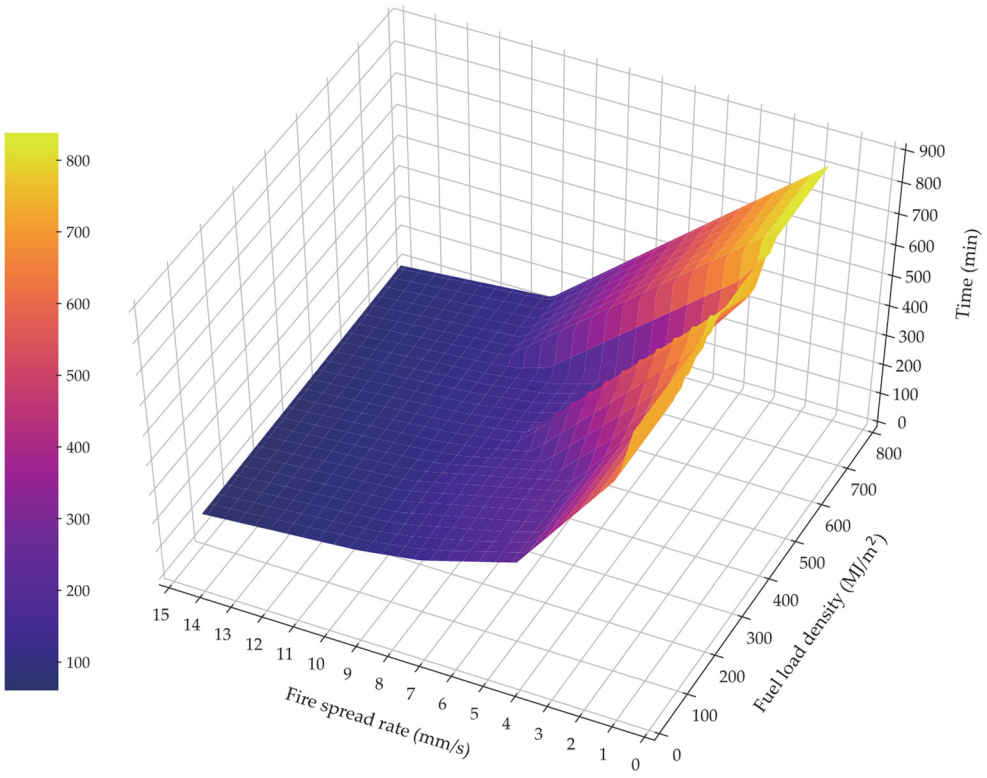


Figure 7.31. Total time durations of the travelling fire scenarios, with various fire spread rates and fuel load densities.

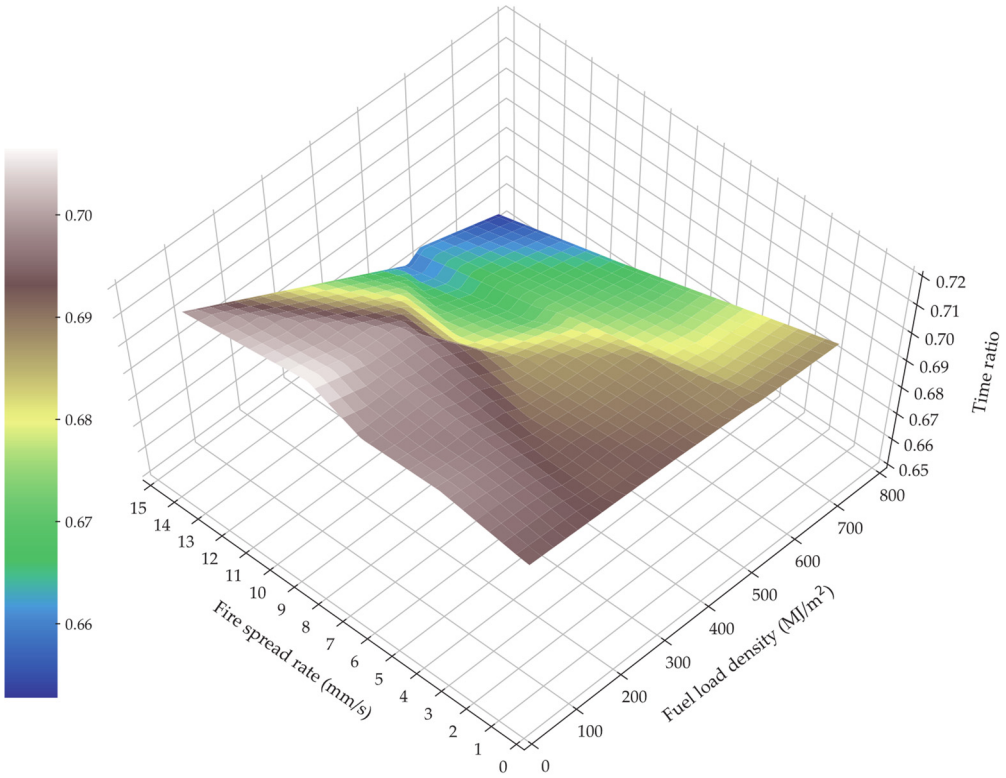


Figure 7.32. Ratio of the value shown in Figure 7.30 to the value shown in Figure 7.31, with various fire spread rates and fuel load densities.

7.4.3 The ETFM Framework in the Temperature Design Domain

The structural fire design in the temperature domain generally refers to the fact that the maximum temperature in the structural solid under the expected fire should be less than the temperature which induces the structural element to fail. Under this definition, the effect of various fire spread rates (v) and fuel load densities ($q_{f,k}$) on the maximum mid-web temperature of the investigated beam (as shown in Figure 7.1) is first studied as presented in Figure 7.33. Its corresponding top and bottom flange temperatures when the mid-web temperature reaches its maximum are also demonstrated in Figure 7.34 and Figure 7.35 respectively. These three locations over the depth of the beam cross-section is shown in Figure 7.3.

Figure 7.33 and Figure 7.35 demonstrate very similar mid-web and bottom flange temperature sensitivities due to various v and $q_{f,k}$, ranging from 200 °C to 900 °C. Higher maximum temperature are captured with higher v and $q_{f,k}$ (i.e. larger fire sizes and HRR). These two 3D contour plots imply v and $q_{f,k}$ are 'equally' influential on the structural design in the temperature design domain, as shown with 3 coloured rings (i.e. red: 700 °C to 900 °C, green: 400 °C to 700 °C, and blue: 200 °C to 400 °C). However, the top flange temperature sensitivities shown in Figure 7.34 suggest a different phenomenon. Apart from slightly smaller temperature range from 200 °C to 850 °C due to the heat sink effect from the concrete slab, more interestingly the 'green ring' in Figure 7.34 changes its 'uniformity' compared with the ones shown in Figure 7.33 and Figure 7.35. It implies that the fuel load densities produce a stronger impact than fire spread rates, which underlines that the resultant through-depth thermal gradient would be more dependent on fuel load densities, rather than fire spread rates.

Furthermore, all these three figures have a temperature plateau, in which Figure 7.33 and Figure 7.35 are at around 900 °C, and Figure 7.34 is at around 850 °C. This plateau is because the corresponding travelling fire scenarios are more severe fires (v = 10 mm/s - 15 mm/s, $q_{f,k}$ = 600 MJ/m^2 – 800 MJ/m^2). These fires are so large, to force the fire to step into the entrainment-controlled situation due to the oxygen starvation.

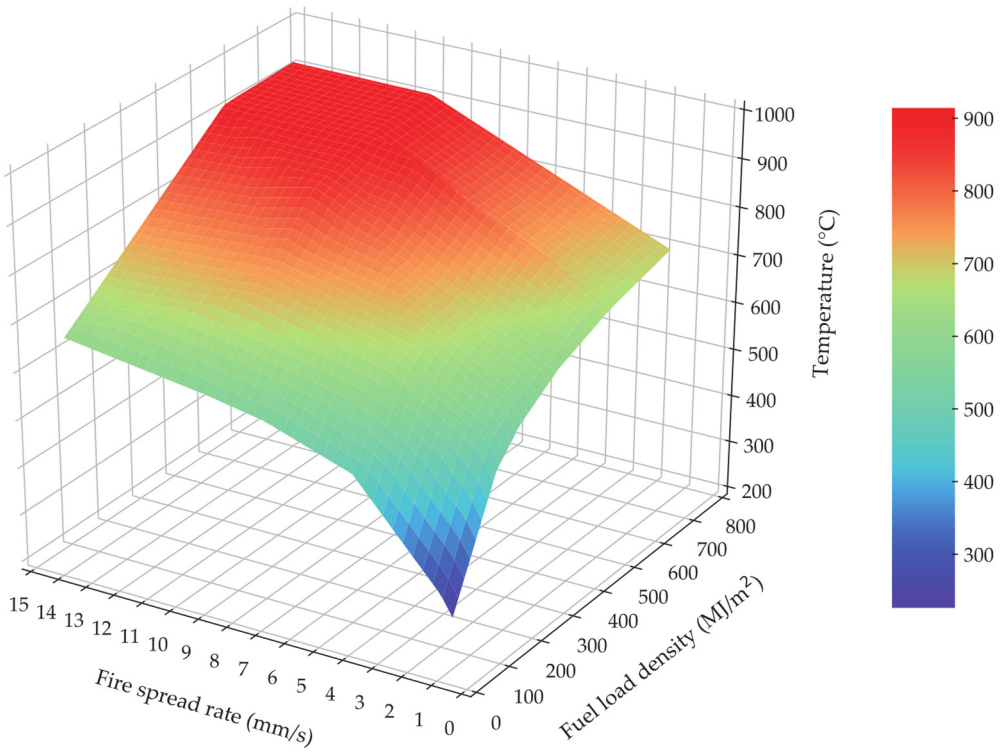


Figure 7.33. Maximum mid-web temperature of the investigated beam, with various fire spread rates and fuel load densities.

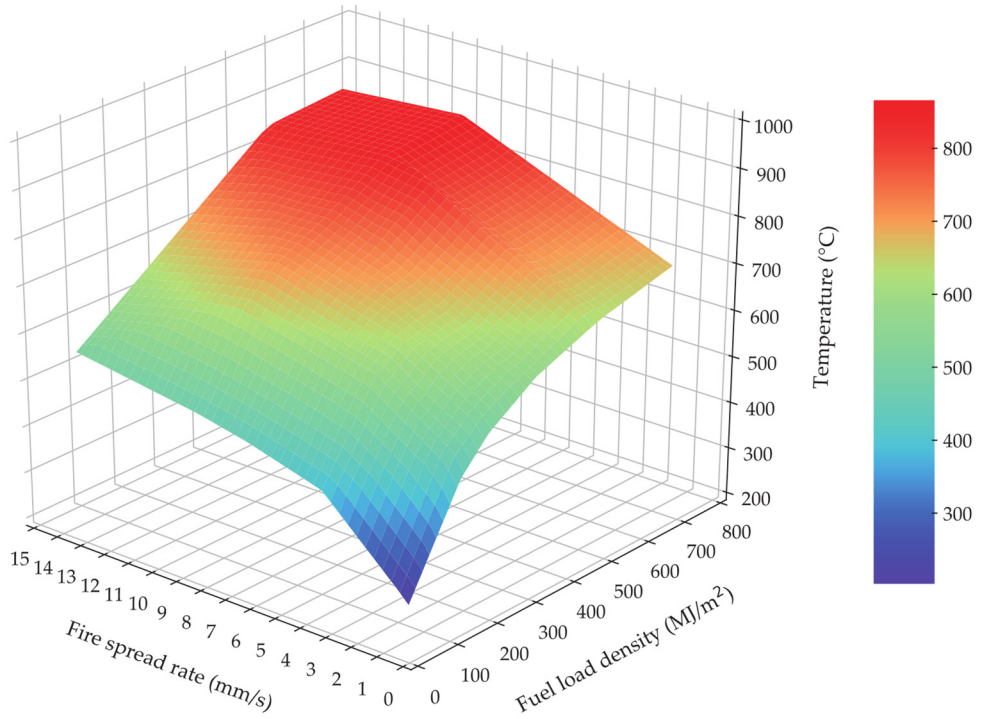


Figure 7.34. Corresponding top flange temperature when mid-web reaches its maximum, with various fire spread rates and fuel load densities.

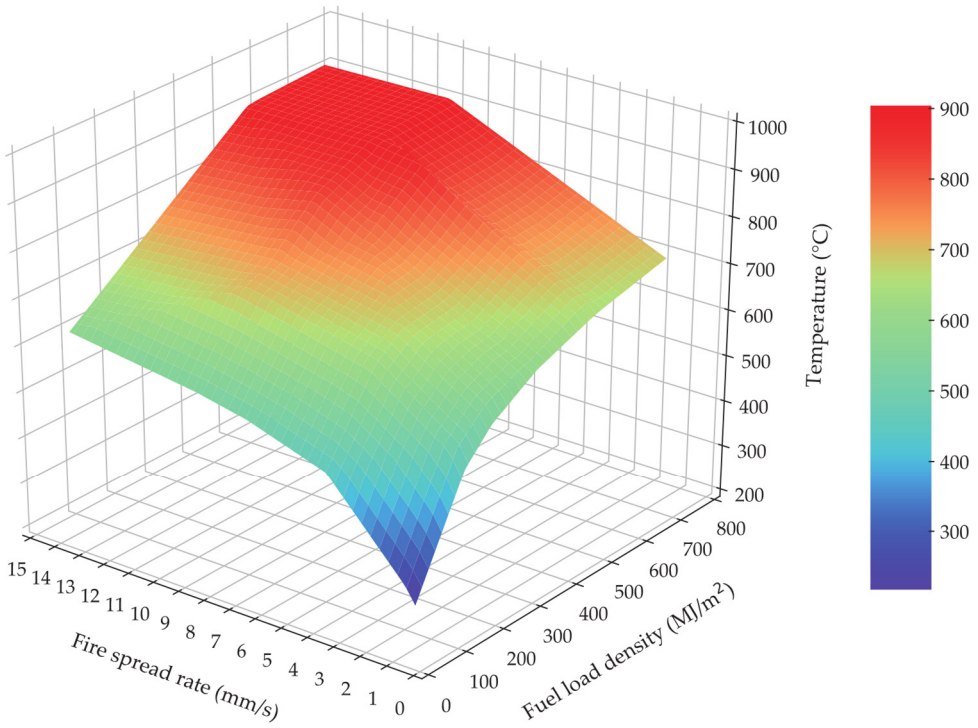


Figure 7.35. Corresponding bottom flange temperature when mid-web reaches its maximum, with various fire spread rates and fuel load densities.

To explain this plateau phenomenon more fundamentally, Figure 7.36, Figure 7.37, and Figure 7.38 are plotted. Figure 7.36 and Figure 7.37 show the maximum burning area and the maximum HRR respectively, when the fire is at the stabilized stage (see Figure 7.5) under different travelling fire scenarios. It can be seen that the maximum fire burning areas are linearly dependent on the v and $q_{f,k}$, which is consistent with the simple assumption made in the ETFM framework for the burning area of fuel A_{fi}. A_{fi} is determined by the travelling fire front edge location (derived from the assumed constant fire spread rate) and the travelling fire back edge location (derived from the burn-out time which is directly related with fuel load density). However, for the maximum HRR as shown in Figure 7.37, a similar plateau appears. This is due to entrainment-controlled burning being 'activated' under these more severe travelling fire scenarios, when the fire is large enough.

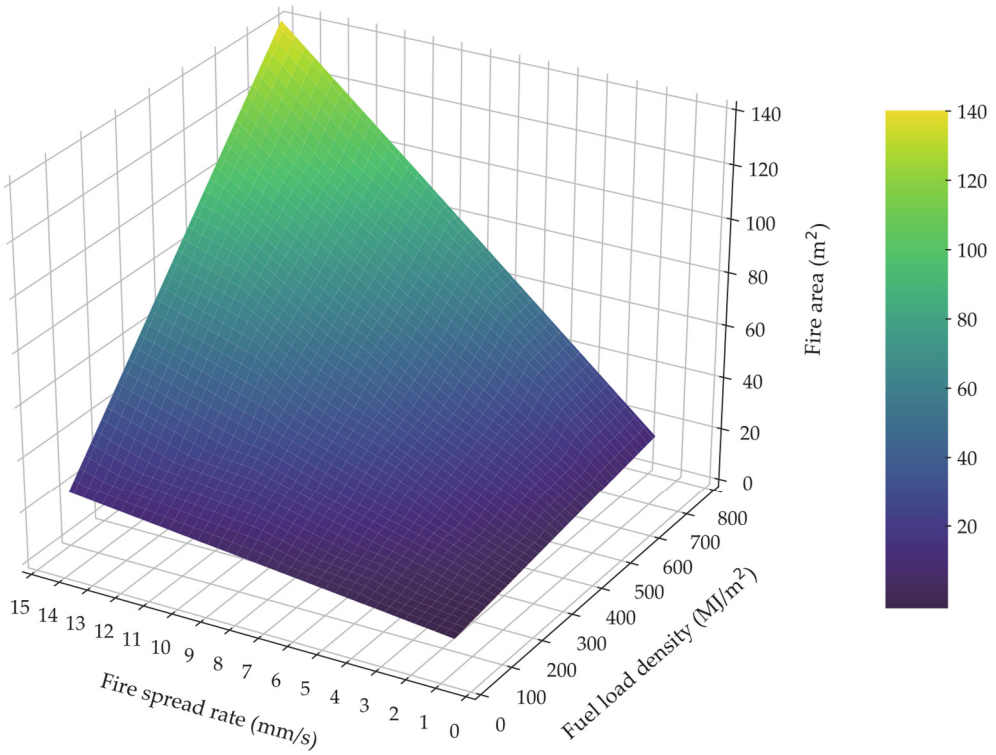


Figure 7.36. Maximum burning area when fire is at stabilized stage under different travelling fire scenarios.

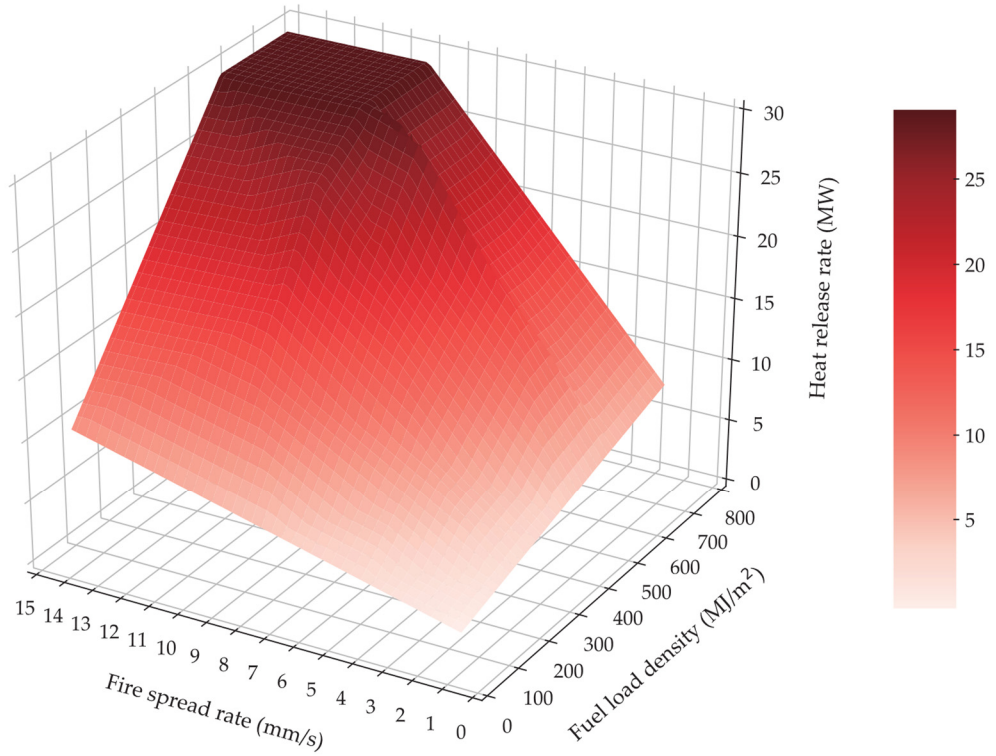


Figure 7.37. Maximum heat release rate when fire is at stabilized stage under different travelling fire scenarios.

The fires under these scenarios are no longer fuel-load controlled, instead they are entrainment-controlled. Maximum HRR is only related to the maximum air mass flow rate $(\dot{m}_e)_{max}$, which is calculated with ΔZ_i (height of zone free of smoke) and L_r. Figure 7.38 shows the stabilized height of the zone free of smoke under different travelling fire scenarios, to consider the effect of oxygen starvation. It can be seen that the smoke layer interface is always beneath the ceiling soffit barrier for all the travelling fire scenarios. This stabilized smoke layer interface 'drops' when more severe fire scenarios (i.e. larger v and $q_{f,k}$) are used, until the fire is large enough to step into the entrainment-controlled situation with a similar plateau. Solid red dots represent the entrainment-controlled fire scenarios, and the hollow ones represent the fuel-controlled fire scenarios.

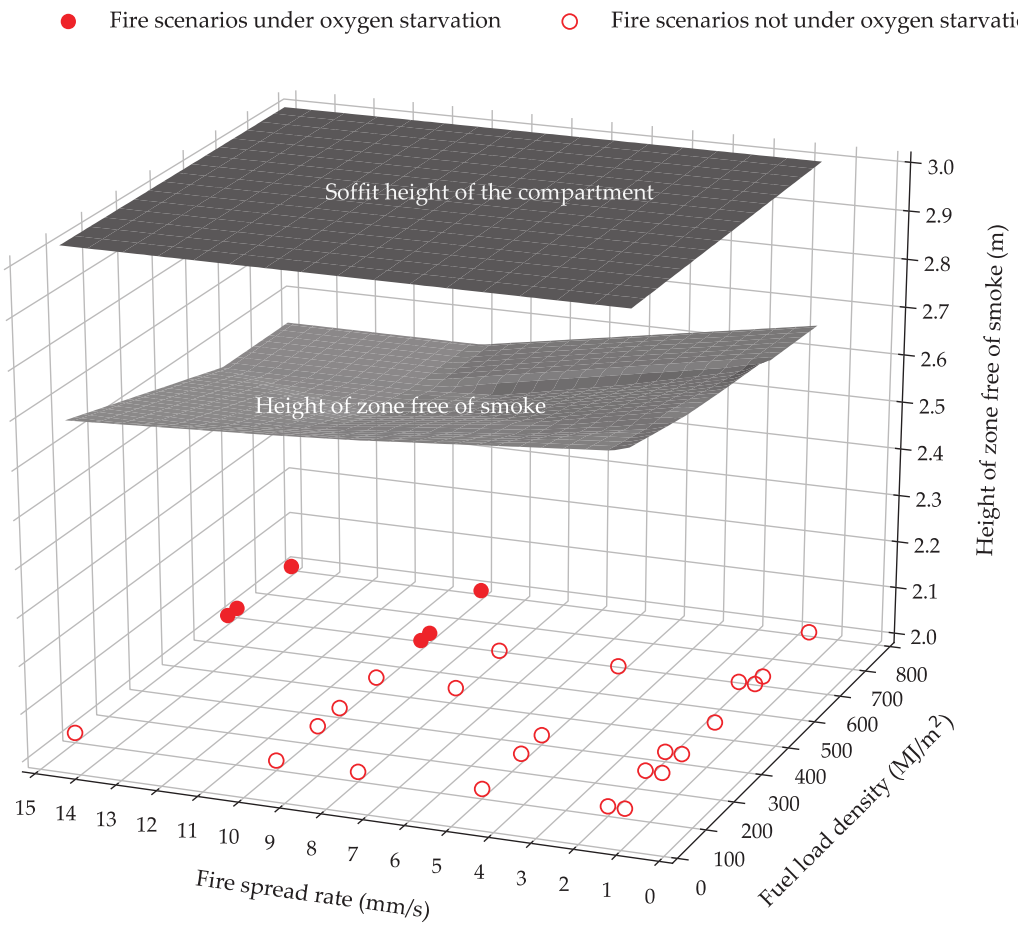


Figure 7.38. Height of zone free of smoke when fire is at stabilized stage, under different travelling fire scenarios to consider the effect of entrainment-controlled burning.

For this case study only six travelling fire scenarios are under the entrainment-controlled burning, out of the total of 29 fire scenarios. However, it is not appropriate to conclude that the entrainment-controlled burning is not likely to happen, and fuel load-controlled burning is more suitable for most cases. This is because only one building case is studied in the thesis, and the possibility for a fire if it is under the entrainment-controlled or fuel-controlled situation in a large compartment, also relies significantly on many other parameters (e.g. the size of the ventilation dimensions, heat loss fraction ratio through walls and ceilings, compartment dimensions, etc).

7.4.4 The Structural Implications Due to the ETFM Framework

The thermal effects on structural behaviour were fundamentally discussed by Usmani *et al.* in 2001 [20]. According to this work, if temperature in the longitudinal direction of a beam is assumed to be uniformly distributed, then its structural behaviour would be dominated by the temperature distributions over its cross-section depth, and boundary conditions. Figure 7.39 schematically presents the thermal effects over the beam depth with 'a simplified equation': a realistic temperature distribution over the depth of a beam is decomposed into uniform temperature increase (ΔT) causing thermal expansion, and thermal gradient (Ty)[1] causing thermal bowing effect.

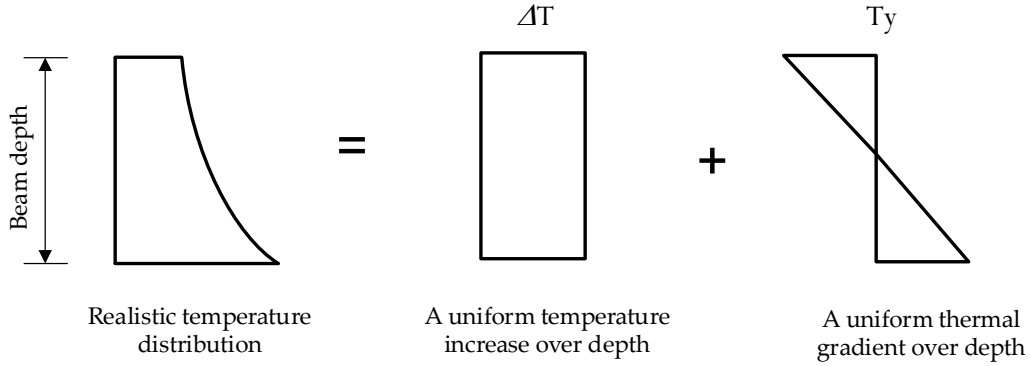


Figure 7.39. Decomposition of temperature effects over the depth of a beam section with thermal expansion and thermal bowing effects.

[1] It is worth to note that the actual definition of thermal gradient Ty from [20] is $(T2 - T1)/d$, where $T2$ is the beam bottom temperature, $T1$ is the beam top temperature, and d is the beam depth. Since the work presented in this Chapter has the same beam depth d for all the travelling fire scenarios, Ty is redefined as $(T2 - T1)/T1$ in this Chapter.

If the heated beam is restrained by a certain level of boundary conditions (i.e. finite rotational restraints and horizontal restraints), internal forces would be induced. As shown in Figure 7.40, the thermal expansion due to ΔT would be constrained by the horizontal restraints, hence an internal compressive axial force would be generated during the heating stage. Meanwhile, thermal bowing due to Ty would be constrained by rotational restraints, hence hogging bending moment would be generated during the heating stage.

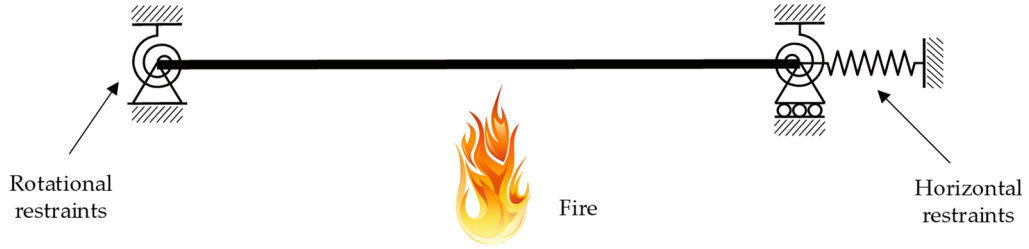


Figure 7.40. A beam member with finite boundary conditions.

Once the design fire severity upgrades from a simple model (e.g. ISO-834 standard fire curve with a single heating regime), to a more realistic fire model (e.g. ETFM framework with 'cyclic' heating and cooling regimes), the interpretation of the corresponding structural behaviour becomes more complicated.

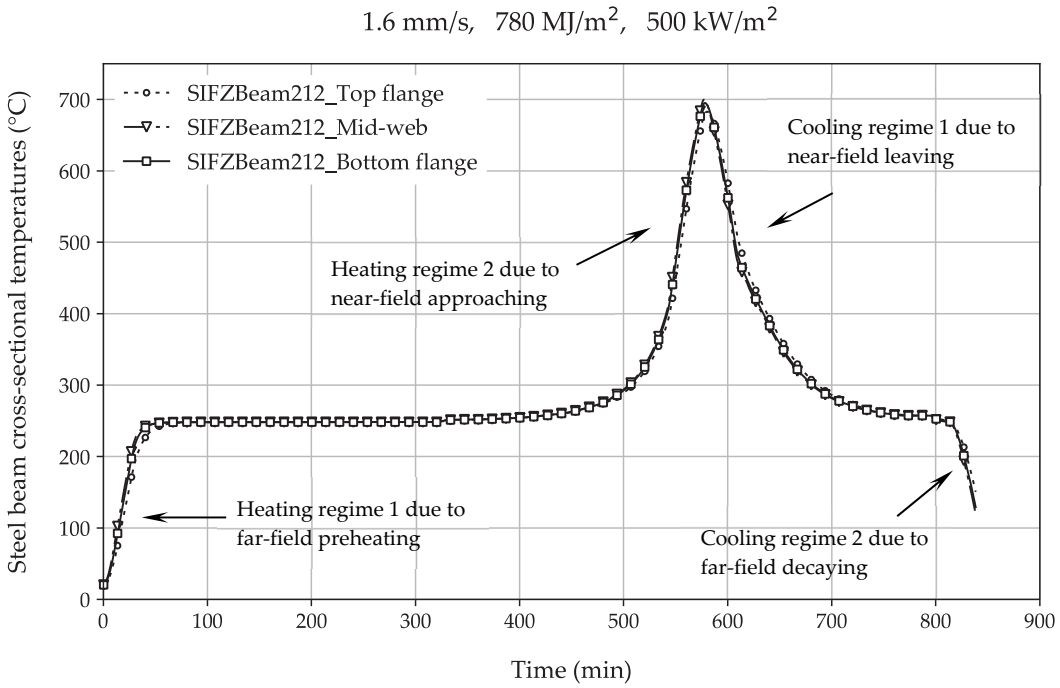


Figure 7.41. Temperature histories of the investigated beam, under travelling fire scenario with v = 1.6 mm/s, $q_{f,k}$ = 780 MJ/m², RHR_f = 500 kW/m².

For example, Figure 7.41 shows the cross-sectional temperature histories of the investigated steel beam in the previous case study, under the 'slow' but 'dense' fire scenario with v = 1.6 mm/s, $q_{f,k}$ = 780 MJ/m^2, RHR_f = 500 kW/m^2. It is assumed that the investigated beam is constrained with both rotational and horizontal restraints, and the temperature at the section mid-web can represent the temperature increase of the whole beam. A compressive force would be first induced by the far-field smoke preheating. After a certain duration of the stabilized smoke heating, larger compressive forces would be generated when the near-field fire plume is approaching the investigated beam (assuming the beam is not yielded, and no catenary action happens at this stage). Following this, a contraction force due to cooling would be generated, as the near-field fire plume is travelling away from the beam. Finally, more cooling induced contraction forces would be generated, during the decay stage of the far-field smoke.

Although both the axial force and bending moment histories due to the progress of travelling fires have been investigated using Rein's model [9, 11], the treatment of the thermal gradient in the beam cross-section is simplified, using lumped capacitance not numerical heat transfer, such as the finite element method (FEM) [9, 11]. In real buildings, a concrete slab is normally on top of these steel beams generating a heat sink effect, hence investigating the through-depth thermal gradient using numerical heat transfer (i.e. FEM) under this situation might generate more accurate distributions, especially for stocky members. Jiang *et al.* [7] investigated the maximum thermal gradient of a steel-composite structure using FEM and a fire boundary condition from Rein's model, concluding that higher through-depth thermal gradient in the beam would be generated with larger travelling fire sizes. However, the thermal gradient histories as the travelling progresses in the timeline were not studied. Further, maximum or peak thermal gradients due to different heating and cooling regimes were not investigated.

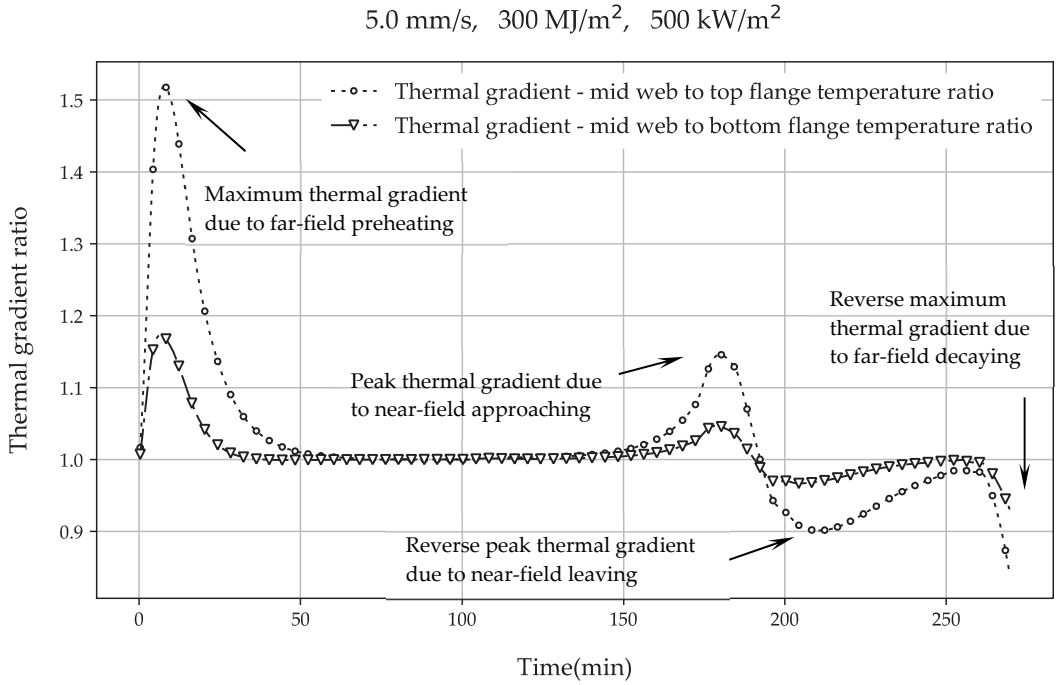


Figure 7.42. Thermal gradient histories of the investigated beam, under travelling fire scenario with v = 5.0 mm/s, $q_{f,k}$ = 300 MJ/m², RHR_f = 500 kW/m².

Figure 7.42 demonstrates the cross-sectional thermal gradient histories of the investigated steel beam, under a travelling fire scenario with v = 5.0 mm/s, $q_{f,k}$ = 300 MJ/m², RHR_f = 500 kW/m². Assuming the investigated beam is constrained with both rotational and horizontal restraints, a clear thermal gradient is first captured due to the far-field smoke preheating. This thermal gradient would generate hogging bending moment as the beam is rotationally constrained. This bending moment would then gradually disappear along with the smoke-induced thermal gradient reducing to 1.0, caused by a stabilized heating duration from far-field smoke. After this, hogging bending moment would be generated again, since the corresponding thermal gradient is induced when the near-field fire plume is approaching the investigated beam. More interestingly, when the near-field fire plume is travelling away from the beam, this thermal gradient would be reversed to 0.9 rather than reduced to 1.0. This 'thermal gradient reversal' means the bending moment would be reversed from hogging type to sagging type. Finally, this thermal gradient again gradually turns to 1.0 until the smoke temperature steps into its decay stage, where another thermal gradient which is below 1.0 shows up.

Figure 7.42 presents the thermal gradient histories under one single travelling fire scenario. It is worth to analyse the thermal gradient histories with more travelling fire scenarios, especially for the 'thermal gradient reversal' phenomena due to the near-field fire plume approaching and leaving the investigated beam period, as the entire thermal gradient histories are directly related to the corresponding hogging or sagging bending moments.

Figure 7.43 presents the thermal gradient histories of the investigated steel beam, under a 'slow' but 'thin' travelling fire, with v = 1.6 mm/s, $q_{f,k}$ = 100 MJ/m², RHR_f = 500 kW/m². Compared with the thermal gradient histories presented in Figure 7.42, it can be seen that maximum thermal gradient due to smoke pre-heating reduces from 1.5 to 1.3, as smaller travelling fire size is generated, with lower fire spread rate and fuel load density adopted in this travelling fire scenario. This confirms one of the findings by Jiang *et al.* [7] as mentioned above. However, it seems the peak thermal gradient due to the near-field fire plume approaching the beam is not sensitive to the fire size change, but the reverse thermal gradient increases a little bit when the fire size decreases.

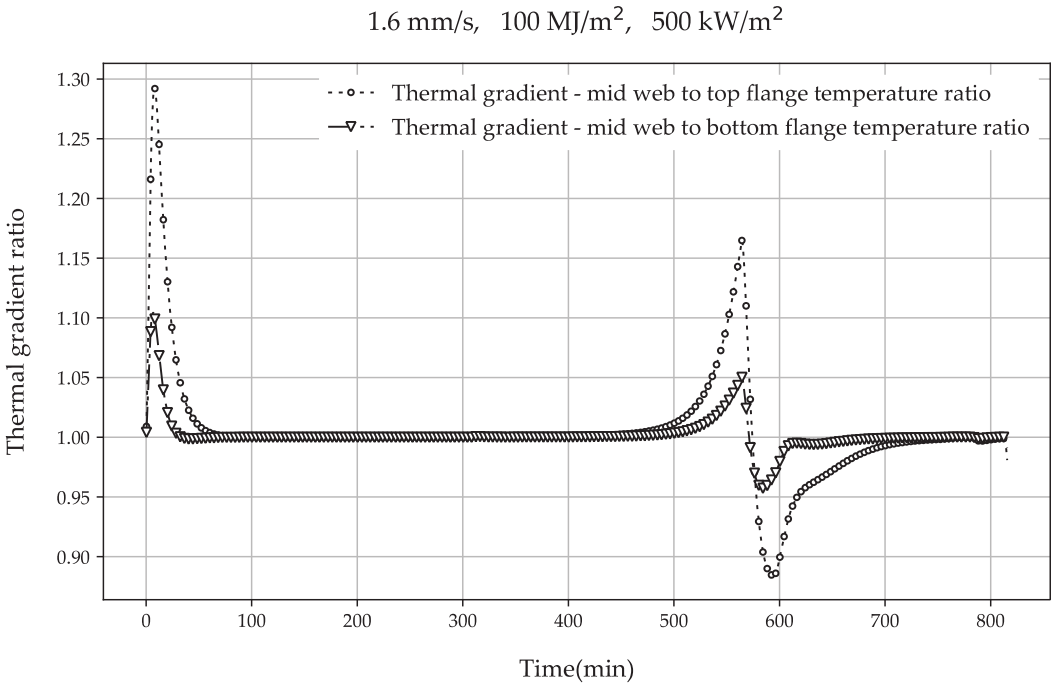


Figure 7.43. Thermal gradient histories of the investigated beam, under travelling fire scenario with v = 1.6 mm/s, $q_{f,k}$ = 100 MJ/m², RHR_f = 500 kW/m².

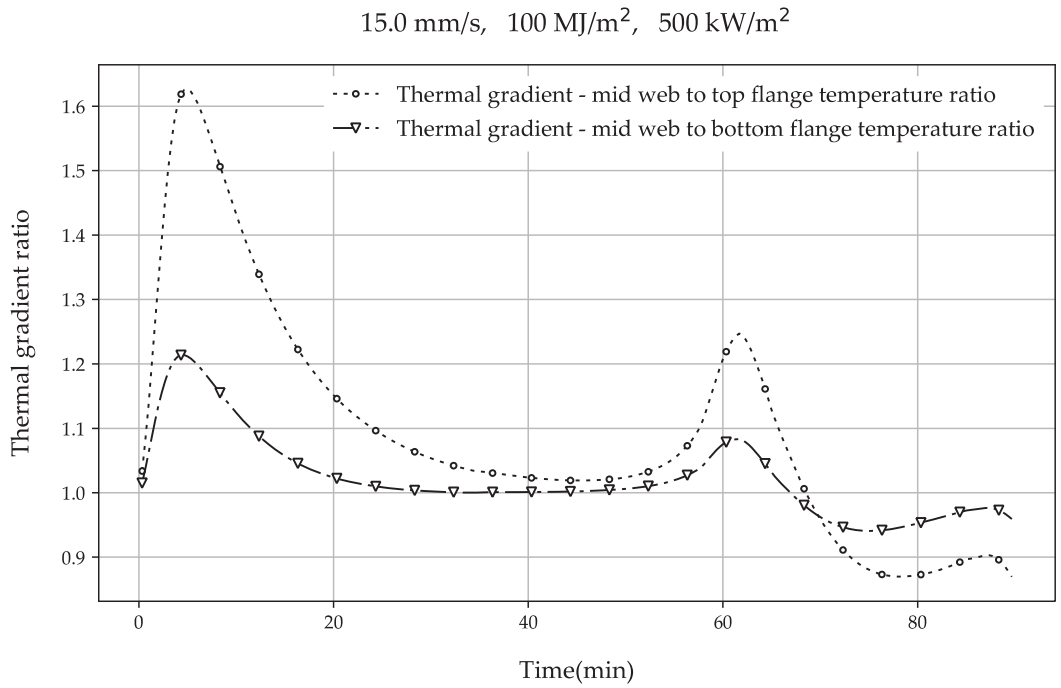


Figure 7.44. Thermal gradient histories of the investigated beam, under travelling fire scenario with v = 15.0 mm/s, $q_{f,k}$ = 100 MJ/m², RHR_f = 500 kW/m².

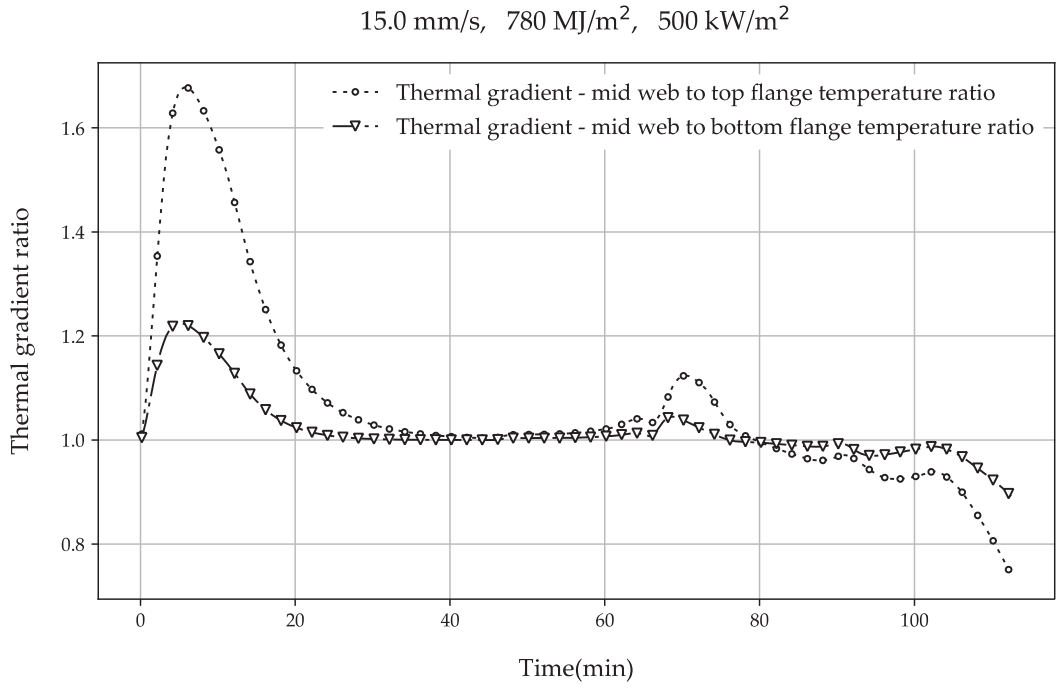


Figure 7.45. Thermal gradient histories of the investigated beam, under travelling fire scenario with v = 15.0 mm/s, $q_{f,k}$ = 780 MJ/m², RHR_f = 500 kW/m².

Compared with the thermal gradient histories presented in Figure 7.43 with a 'slow' but 'thin' travelling fire scenario, Figure 7.44 presents the thermal gradient histories under a 'fast' but 'thin' travelling fire with v = 15.0 mm/s, $q_{f,k}$ = 100 MJ/m^2, RHR_f = 500 kW/m^2. It shows that both the maximum thermal gradient due to smoke pre-heating and peak thermal gradient due to near-field approaching increase when a higher fire spread rate is used. However, it seems the reverse thermal gradient due to near-field leaving is not sensitive to the fire spread rate increase. Sticking to the same fire spread rate (15.0 mm/s) but increasing the fuel load density to an extreme case of 780 MJ/m^2, Figure 7.45 implies the maximum thermal gradient due to smoke pre-heating increases slightly. More interestingly, the 'thermal gradient reversal' due to the near-field approaching and leaving the beam, diminish under this extremely 'fast' and 'thick' travelling fire scenario.

Figure 7.46 demonstrates the comparison of thermal gradient histories of the investigated steel beam, with various fuel load densities ranging from $q_{f,k}$ = 100 MJ/m^2 to $q_{f,k}$ = 780 MJ/m^2, and constant v = 10 mm/s, RHR_f = 500 kW/m^2. This thermal gradient is the ratio of mid-web temperature to top flange temperature. It can be seen that both the maximum thermal gradient due to smoke pre-heating (1.6 approximately), and reverse maximum thermal gradient due to smoke decaying (0.8 approximately), are not sensitive to various fuel load densities. However, both the peak thermal gradient due to near-field approaching and the reverse peak thermal gradient due to near-field leaving the investigated beam, change dramatically with various fuel load densities. In general, these peak and reverse peak thermal gradients decrease with increasing fuel load densities.

Figure 7.47 demonstrates the comparison of thermal gradient histories of the investigated steel beam, with various fire spread rates ranging from v = 1.6 mm/s to v = 15 mm/s, and constant $q_{f,k}$ = 570 MJ/m^2, RHR_f = 500 kW/m^2. This thermal gradient is the ratio of mid-web temperature to top flange temperature. It can be seen that both the peak thermal gradient due to near-field approaching (1.1 approximately), and reverse peak thermal gradient due to near-field leaving (0.9 approximately), are not sensitive to various fire spread rates. However, both the maximum thermal gradient due to smoke pre-heating and the reverse maximum thermal gradient due to smoke decaying, are sensitive to various fire spread rates. In general, these maximum and reverse maximum thermal gradients increase with higher fire spread rates.

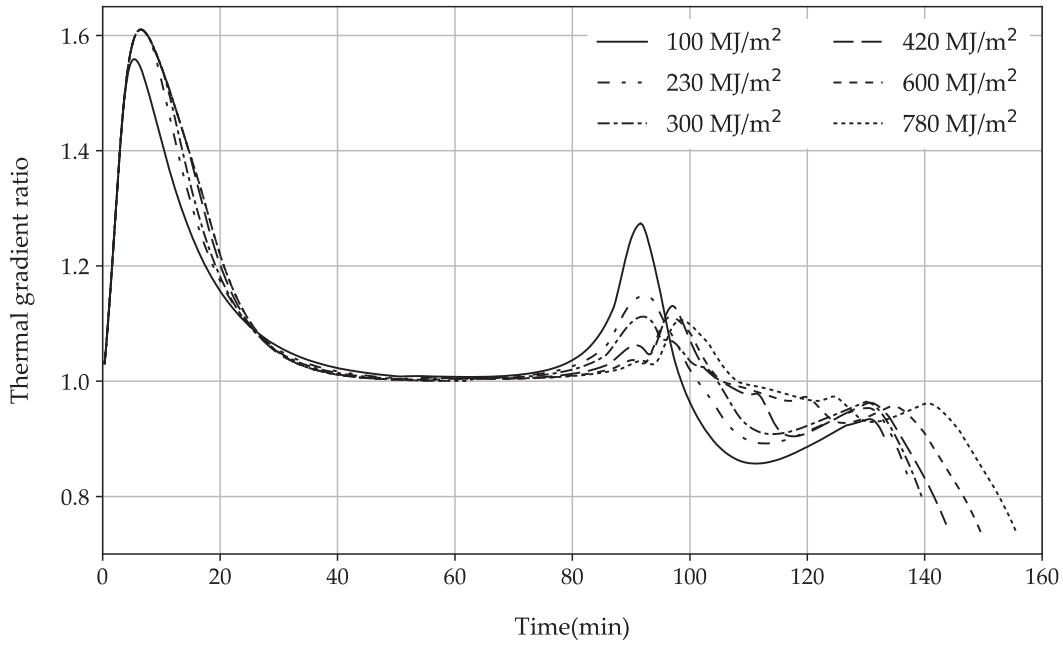


Figure 7.46. Thermal gradient histories of the investigated steel beam, with various fuel load densities ranging from $q_{f,k}$ = 100 MJ/m² to $q_{f,k}$ = 780 MJ/m², and constant v = 10 mm/s, RHR_f = 500 kW/m².

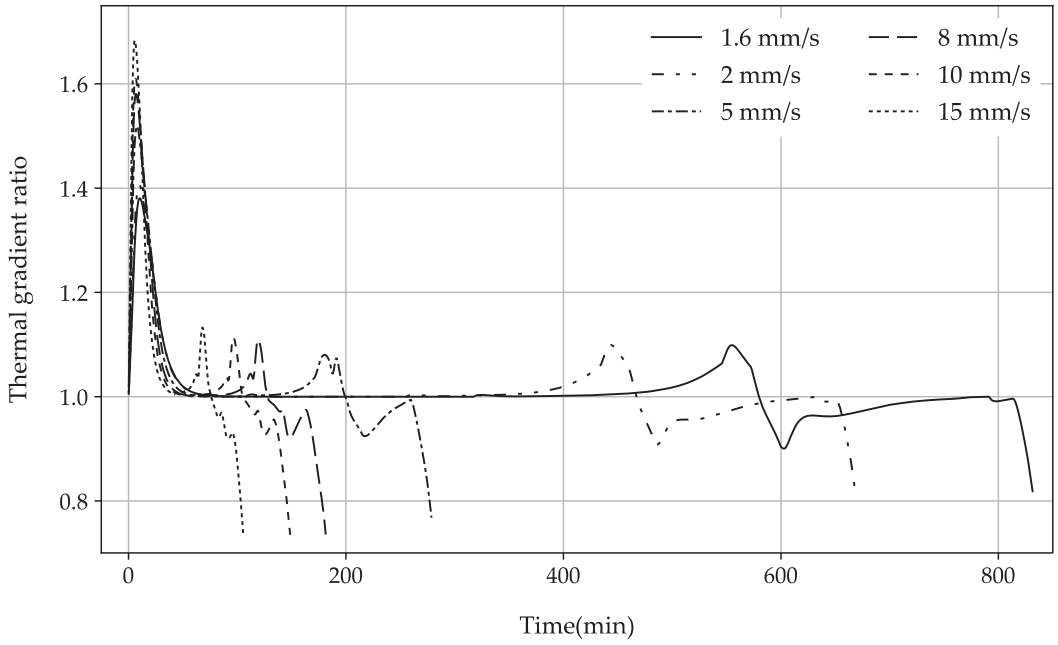


Figure 7.47. Thermal gradient histories of the investigated steel beam, with various fire spread rates ranging from v = 1.6 mm/s to v = 15 mm/s, and constant $q_{f,k}$ = 570 MJ/m², RHR_f = 500 kW/m².

Figure 7.42 to Figure 7.45 present the thermal gradient histories under individual travelling fire scenarios in the timeline. Figure 7.46 and Figure 7.47 present the thermal gradient histories under a series of travelling fire scenarios in the timeline, with various fuel load densities and fire spread rates respectively. However, it is worth to investigate the maximum values and peak values of the thermal gradients (due to far-field smoke and near-field fire plume), based on all 29 travelling fire scenarios.

Figure 7.48 shows the maximum thermal gradient due to smoke pre-heating under all 29 travelling fire scenarios. It suggests this maximum value is proportional to the fire spread rate, and less sensitive to the fuel load density. Figure 7.49 shows the peak thermal gradient due to near-field approaching the investigated beam, under all 29 travelling fire scenarios. It implies this peak value is more sensitive to the fuel load density rather than fire spread rate, where larger peak values are captured with lower fuel load densities.

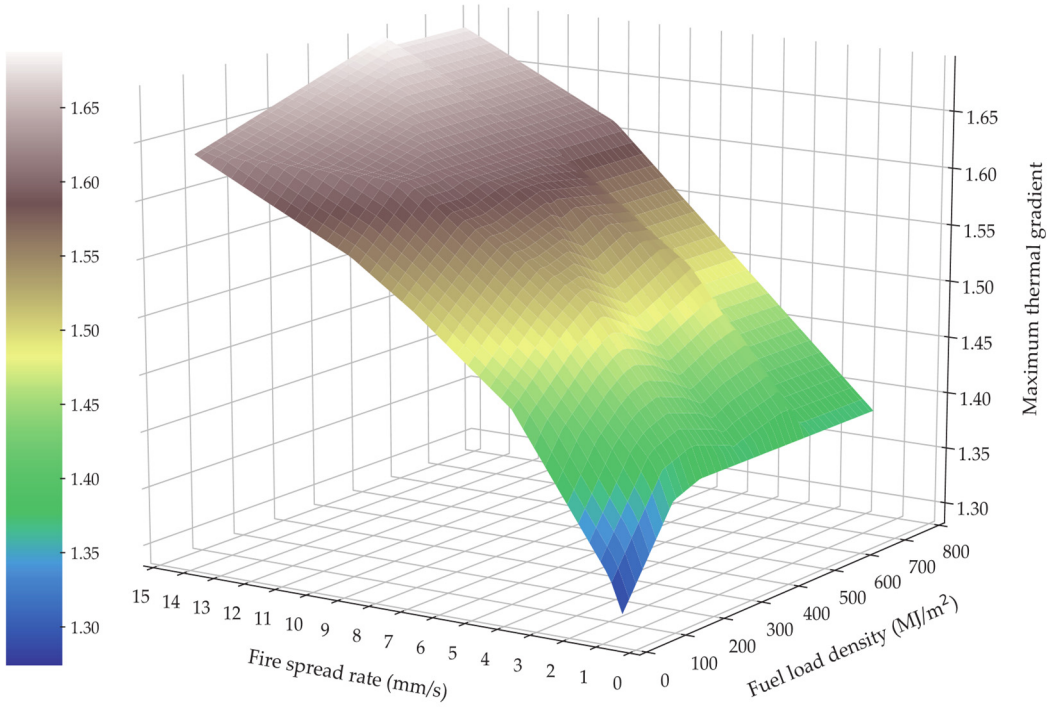


Figure 7.48. Maximum thermal gradient due to the smoke pre-heating investigated beam, under all the 29 travelling fire scenarios.

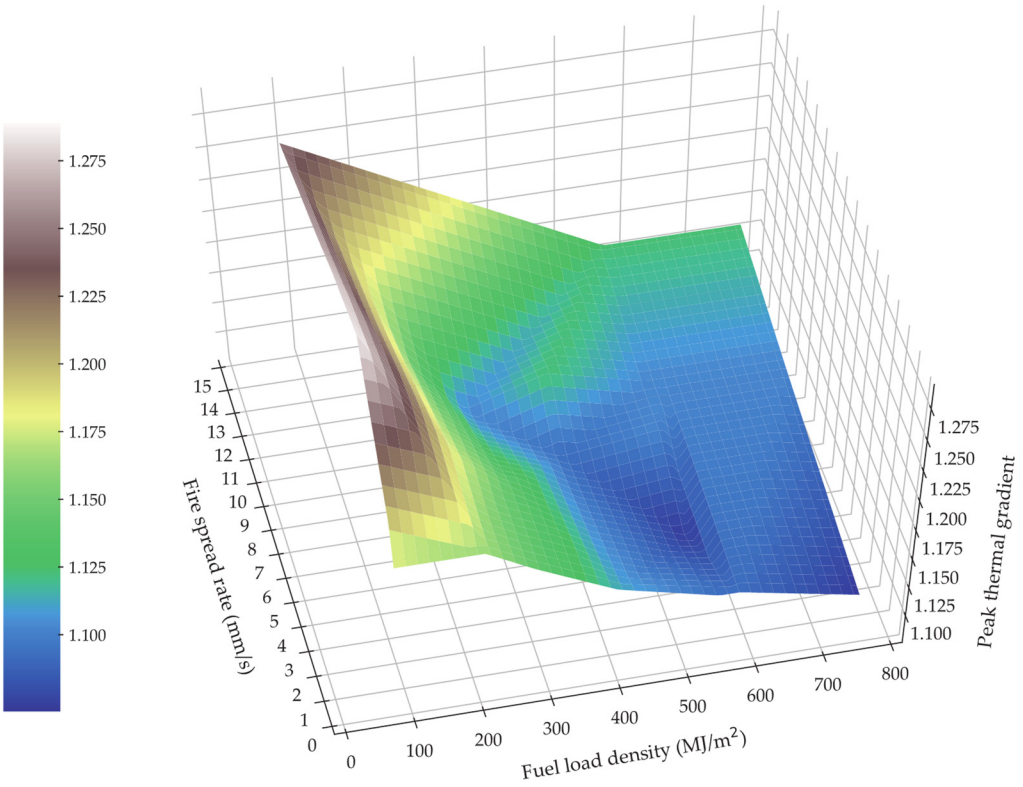


Figure 7.49. Peak thermal gradient due to the near-field approaching investigated beam, under all the 29 travelling fire scenarios.

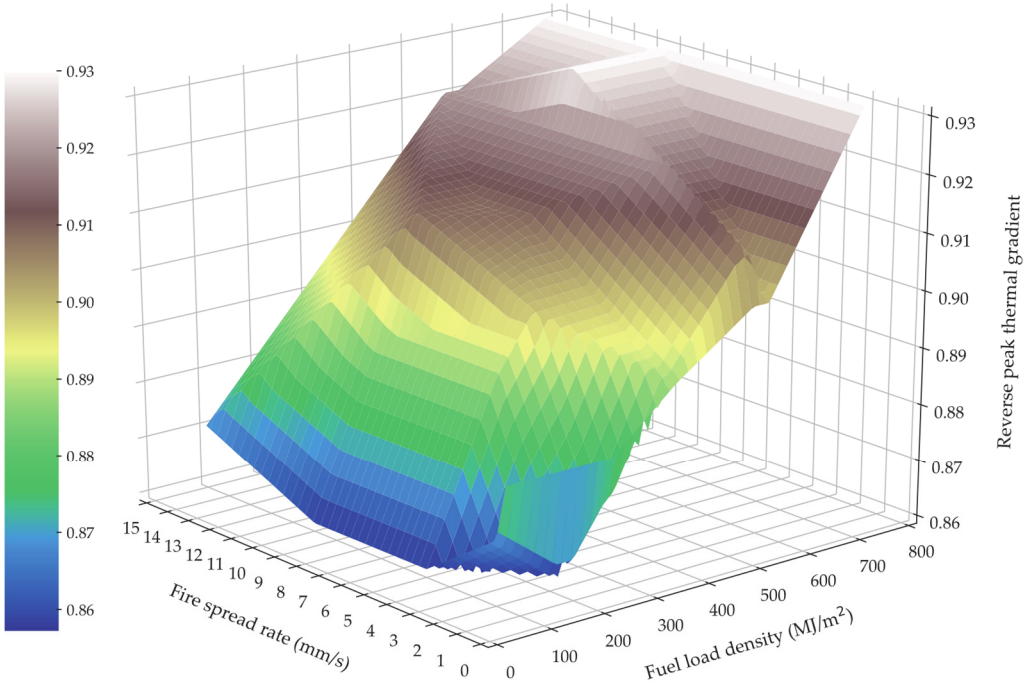


Figure 7.50. Reverse peak thermal gradient due to the near-field leaving investigated beam, under all the 29 travelling fire scenarios.

Figure 7.50 shows the reverse peak thermal gradient due to the near-field leaving the investigated beam under all 29 travelling fire scenarios. Similar to what is suggested in Figure 7.49, Figure 7.50 implies this reverse peak value is also sensitive to the fuel load density rather than fire spread rate. However, this reverse peak value is inversely proportional to the fuel load densities. Figure 7.51 shows the ratio of the difference between the peak thermal gradient and reverse peak thermal gradient, to further investigate this 'thermal gradient reversal' phenomena. This figure presents a very similar trend as that shown in Figure 7.49. It implies this difference value is more sensitive to the fuel load density rather than fire spread rate, where larger difference values are captured with lower fuel load densities.

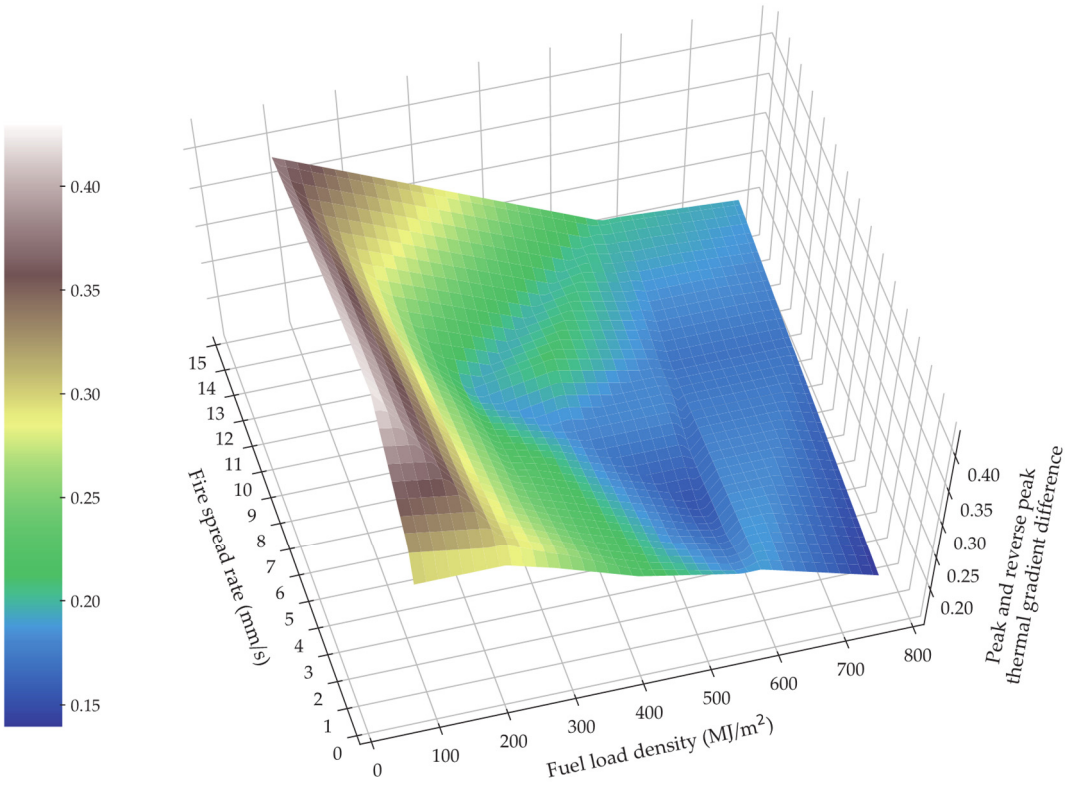


Figure 7.51. Peak and reverse peak thermal gradient difference, due to near-field approaching and leaving the investigated beam, under all the 29 travelling fire scenarios.

7.5 A DEMONSTRATION OF THE ETFM FRAMEWORK FOR CHARACTERISING STRUCTURAL RESPONSE

This section presents a demonstration of the ETFM framework for characterising structural response. It is assumed that the beams in the global x direction have size UB 305x127x42, and in the global z direction have size UB 254x102x28 (x, and z directions defined in Figure 7.52). All columns are assumed fire-proved with size UC 356x406x235. The concrete slabs are not considered in this case, for simplifying the representation of whole structural response in a clearer manner. All beams are applied with a 2 kN/m UDL and self-weight. The design travelling fire parameters are kept the same as presented in the case study from section 7.1, but assuming the fire spread rate is 5 mm/s, fuel load density 300 MJ/m^2, heat release rate per unit area is 500 kW/m^2.

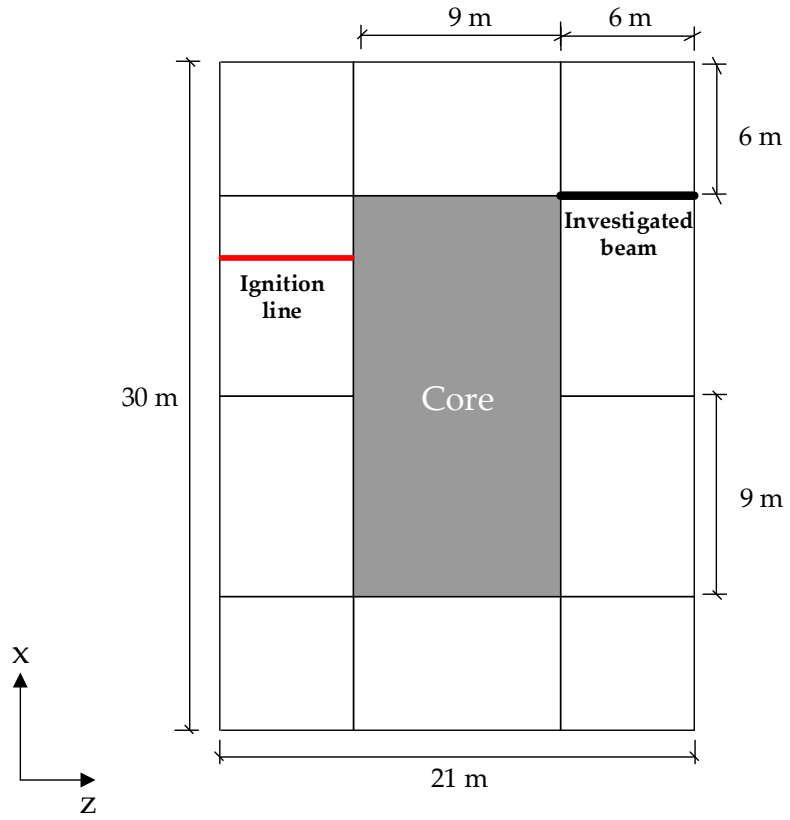


Figure 7.52. Case study plan view with coordinates definition.

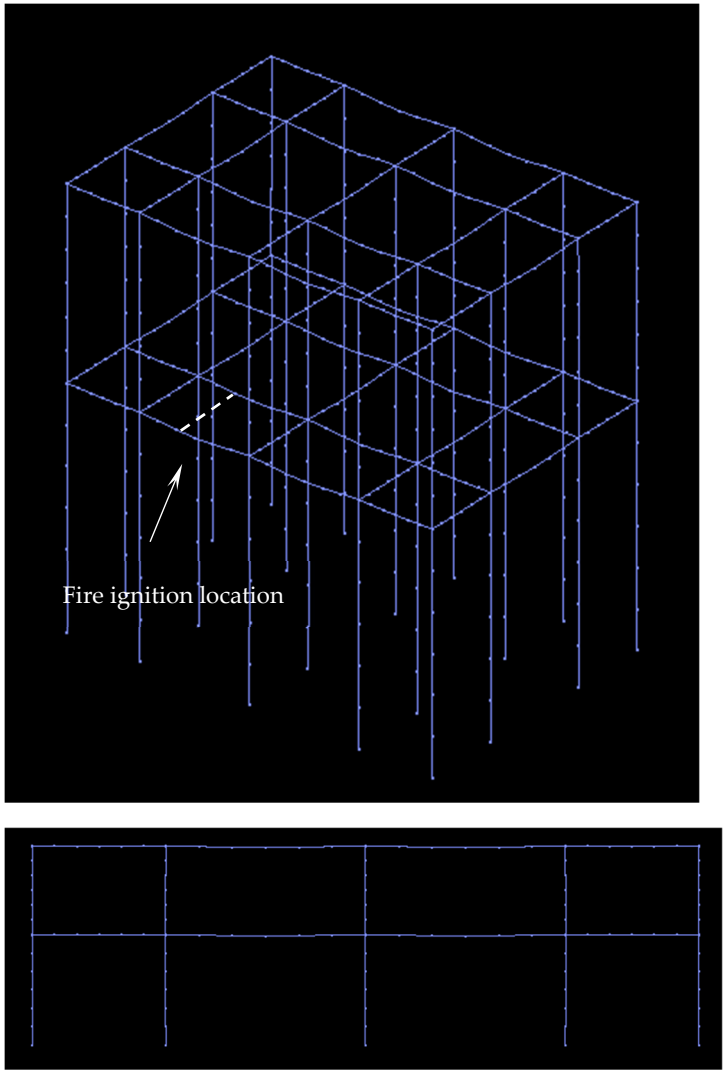


Figure 7.54. SIFBuilder visualization during thermo-mechanical analysis after applying 2 kN/m UDL and self-weight (3D and elevation view).

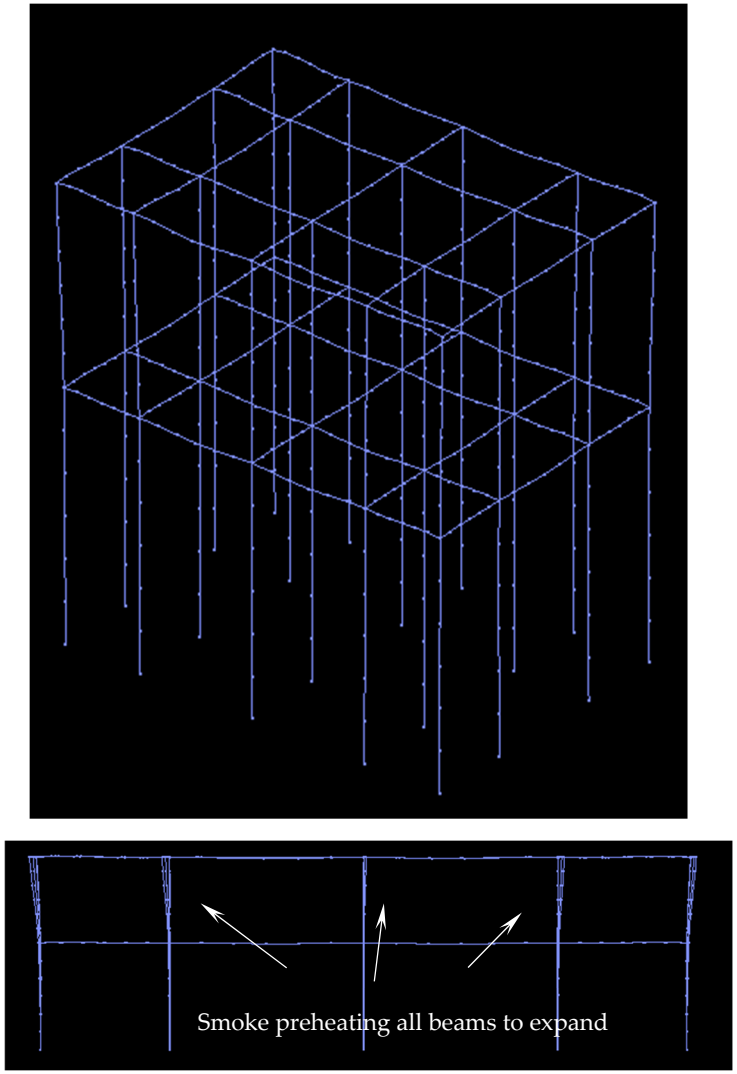


Figure 7.53. SIFBuilder visualization during thermo-mechanical analysis at 10.7 mins under the travelling fire (3D and elevation view).

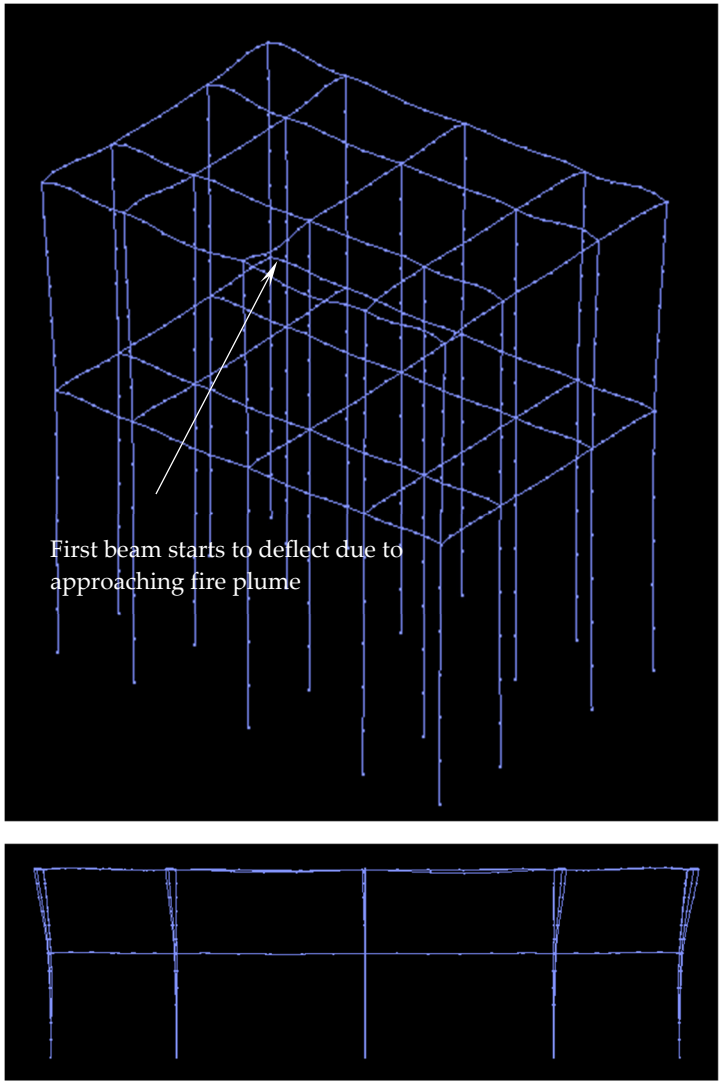


Figure 7.56. SIFBuilder visualization during thermo-mechanical analysis at 46.3 mins under the travelling fire (3D and elevation view).

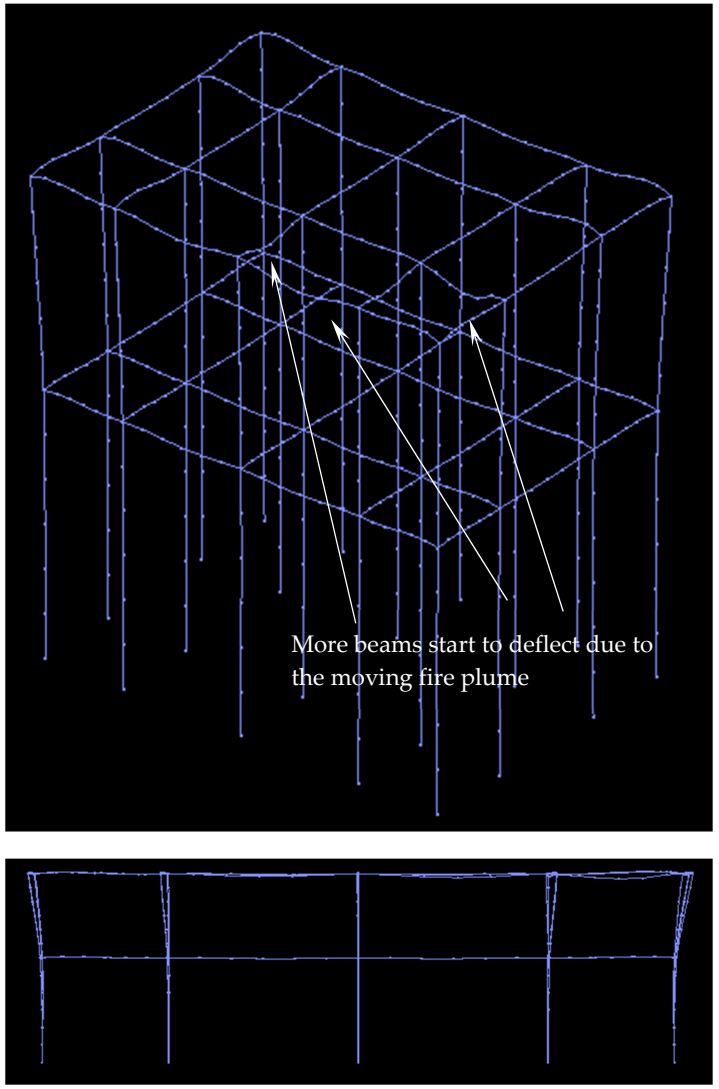


Figure 7.55. SIFBuilder visualization during thermo-mechanical analysis at 75.7 mins under the travelling fire (3D and elevation view).

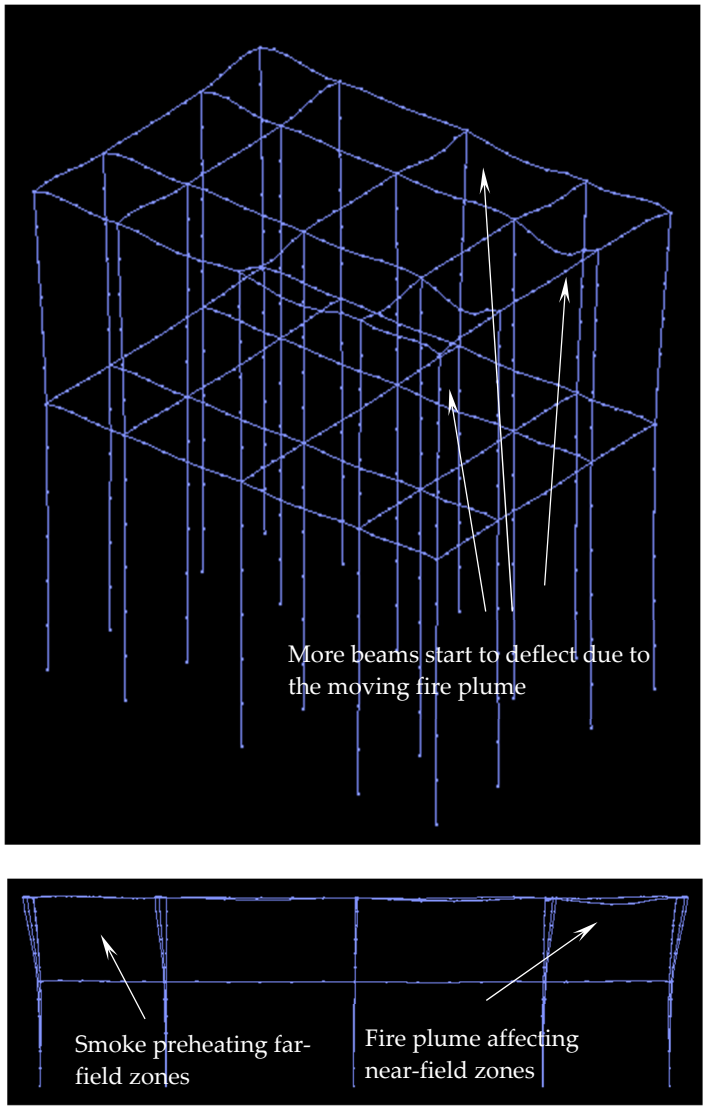


Figure 7.57. SIFBuilder visualization during thermo-mechanical analysis at 106.3 mins under the travelling fire (3D and elevation view).

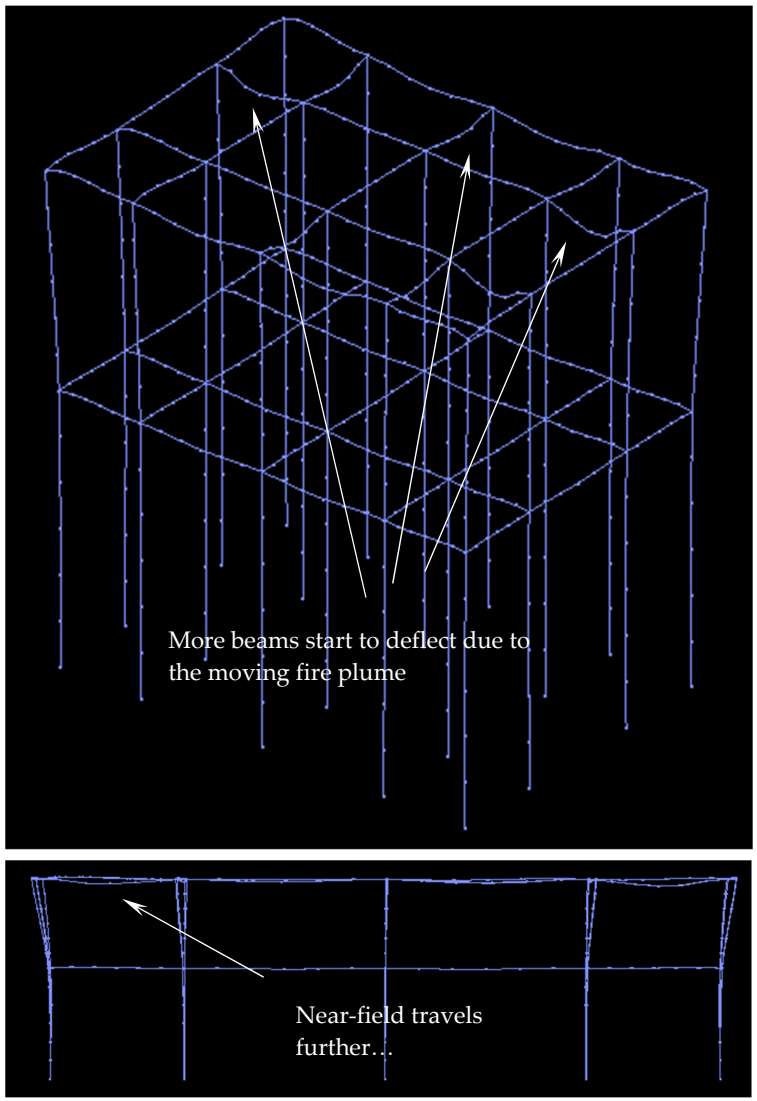


Figure 7.58. SIFBuilder visualization during thermo-mechanical analysis at 205 mins under the travelling fire (3D and elevation view).

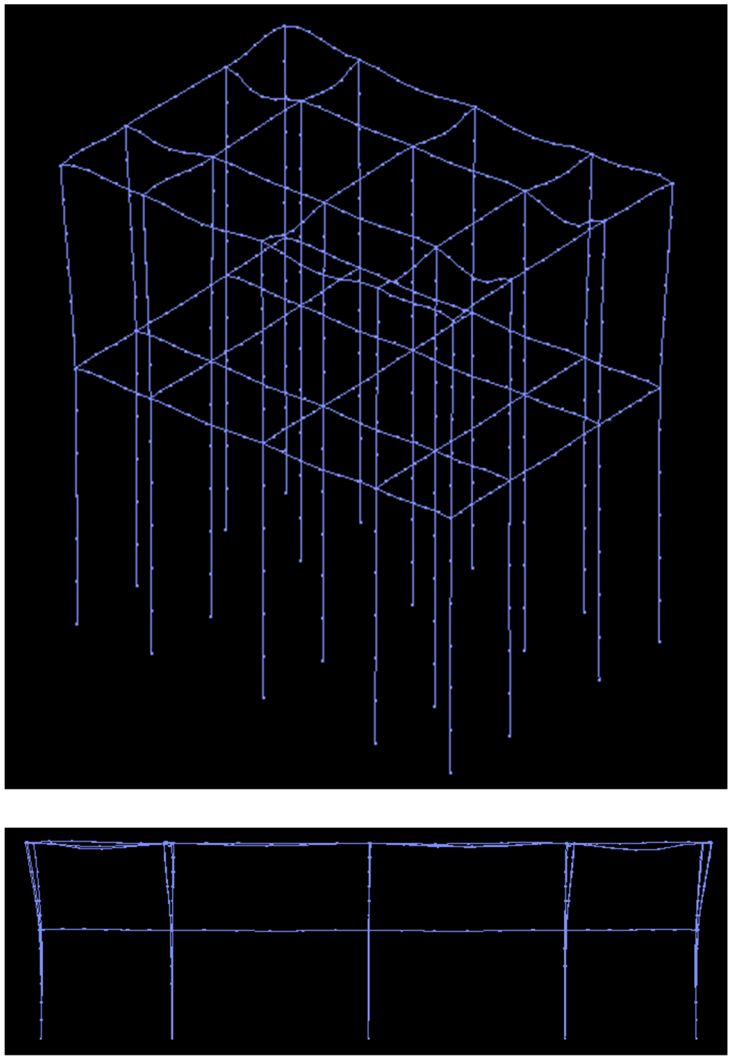


Figure 7.59. SIFBuilder visualization during thermo-mechanical analysis at 234 mins under the travelling fire (3D and elevation view).

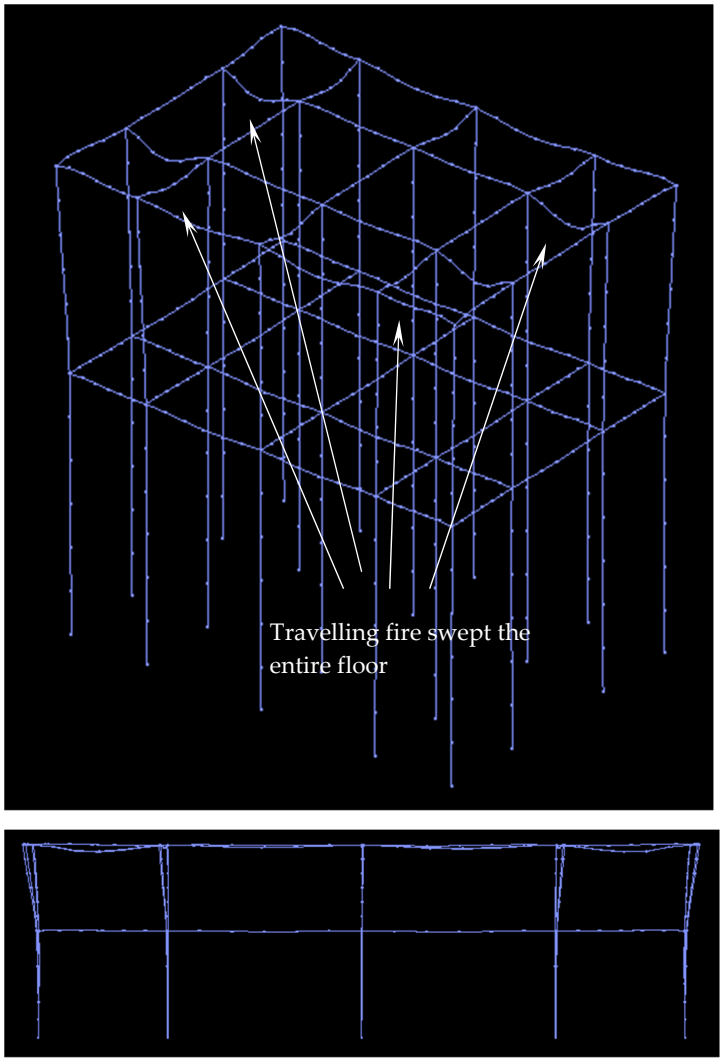


Figure 7.60. SIFBuilder visualization during thermo-mechanical analysis at 266.3 mins under the travelling fire (3D and elevation view).

7.6 CONCLUSIONS

This chapter investigates the impact of the ETFM parameters (i.e. fire spread rate, fuel load density) in the time and temperature design domain, through quantifying the cross-sectional time-temperature evolution of an investigated steel beam with a case study. Moreover, the temperature histories of the sequential beams which are right above the fire trajectory are analysed. In general, 29 travelling fire scenarios are investigated, with total number of 696 heat transfer analysis are performed in this chapter. The complete modelling results are attached in Appendix C. Finally, the thermal response implications on the subsequent structural responses: the temperature rise induced thermal expansion axial force, and the change of through depth induced thermal bowing bending moment, are also discussed in this chapter.

The following are some of the key findings: (1) For 'slow' fires (i.e. low fire spread rate), the near-field fire plume brings more detrimental thermal impact compared with the impact from far-field smoke. However, for 'fast' fires (i.e. high fire spread rate), the far-field smoke brings more detrimental thermal impact. (2) For the sequential beams (right above the fire trajectory) under a 'fast' but 'thin' travelling fire scenario (i.e. high spread rate and low fuel load density), the peak temperatures of these beams would be fluctuating. This is because some of the structural members are not given enough time to be heated up by the 'fast travelling' near-field fire plume. Hence the relative locations of the structural members in the compartment are having an impact. This peak temperature fluctuation would diminish when the fire spread rates become lower. (3) Fire spread rate and fuel load density are 'equally' discriminating factors on the time for a structural member to reach its peak temperature. Nevertheless, the fire spread rate is a more discriminating factor in affecting the total travelling fire time duration in the ETFM framework. (4) Fire spread rate and fuel load density are 'equally' influential on the maximum temperature that a design beam would reach. (5) For more severe travelling fire scenarios (i.e. high fire spread rates and large fuel load densities), the fire would be entrainment-controlled rather than fuel-controlled. Hence the ventilation cannot be neglected for a design fire severity using travelling fires. (6) The through depth thermal gradients due to different travelling fire scenarios are extensively explored, especially

with regards to the 'thermal gradient reversal' due to the near-field fire plume approaching and leaving the design structural member. It suggests that the maximum thermal gradient due to smoke-preheating, is proportional to the fire spread rate and less sensitive to the fuel load density. The peak thermal gradient due to near-field approaching is more sensitive to the fuel load density than fire spread rate, where larger peak values are captured with lower fuel load densities. Moreover, the reverse peak thermal gradient due to near-field leaving is also sensitive to the fuel load density rather than the fire spread rate, but this reverse peak value is inversely proportional to the fuel load densities.

7.7 REFERENCES

[1] C. G. Clifton, "Fire Models for Large Firecells. HERA Report R4-83," HERA publications, New Zealand, 1996.

[2] P. J. Moss and G. C. Clifton, "Modelling of the cardington LBTF steel frame building fire tests," *Fire and Materials*, vol. 28, no. 2–4, pp. 177–198, 2004.

[3] E. Ellobody and C. G. Bailey, "Structural performance of a post-tensioned concrete floor during horizontally travelling fires," *Engineering Structures*, vol. 33, no. 6, pp. 1908–1917, 2011.

[4] A. Law, J. Stern-Gottfried, M. Gillie, and G. Rein, "The influence of travelling fires on a concrete frame," *Engineering Structures*, vol. 33, no. 5, pp. 1635–1642, 2011.

[5] J. Stern-Gottfried and G. Rein, "Travelling fires for structural design–Part I: Literature review," *Fire Safety Journal*, vol. 54, pp. 74–85, 2012.

[6] J. Stern-Gottfried and G. Rein, "Travelling fires for structural design-Part II: Design methodology," *Fire Safety Journal*, vol. 54, pp. 96–112, 2012.

[7] Y. Jiang, P. Kotsovinos, A. Usmani, G. Rein, and J. Stern-Gottfried, "Numerical investigation of thermal responses of a composite structure in horizontally travelling fires using OpenSees," *Procedia Engineering*, vol. 62, pp. 736–744, 2013.

[8] E. Rackauskaite, C. Hamel, A. Law, and G. Rein, "Improved formulation of travelling fires and application to concrete and steel structures," *Structures*, vol. 3, pp. 250–260, 2015.

[9] F. H. Rezvani and H. R. Ronagh, "Structural response of a MRF exposed to travelling fire," *Proceedings of the Institution of Civil Engineers - Structures and Buildings*, vol. 168, no. 9, pp. 619–635, 2015.

[10] E. Rackauskaite, P. Kotsovinos, A. Jeffers, and G. Rein, "Structural analysis of multi-storey steel frames exposed to travelling fires and traditional design fires," *Engineering Structures*, vol. 150, pp. 271–287, 2017.

[11] E. Rackauskaite, P. Kotsovinos, and G. Rein, "Structural response of a steel-frame building to horizontal and vertical travelling fires in multiple floors," *Fire Safety Journal*, vol. 91, pp. 542–552, 2017.

[12] X. Dai, L. Jiang, J. Maclean, S. Welch, and A. Usmani, "Implementation of a new design travelling fire model for global structural analysis," in *the Proceedings of the 9th International Conference on Structures in Fire*, 2016, pp. 959–966.

[13] X. Dai, Y. Jiang, L. Jiang, S. Welch, and A. S. Usmani, "Implementation of fire models in OpenSees," in *Proceedings of the 1st European Conference on OpenSees*, 2017, pp. 47–50.

[14] F. T. McKenna, "Object-Oriented Finite Element Programming: Frameworks for Analysis, Algorithms and Parallel Computing," PhD Thesis, the University of California, 1997.

[15] L. Jiang, "Development of An Integrated Computational Tool for Modelling Structural Frames in Fire Considering Local Effects," https://www.era.lib.ed.ac.uk/handle/1842/19563, PhD Thesis, School of Engineering, University of Edinburgh, 2016.

[16] Y. Jiang, "Development and Application of a Thermal Analysis Framework in OpenSees for Structures in fire," https://www.era.lib.ed.ac.uk/handle/1842/7941, PhD Thesis, School of Engineering, the University of Edinburgh, 2012.

[17] M. L. Janssens, *An Introduction to Mathematical Fire Modeling*, 2nd ed. CRC Press, 2000.

[18] European Standard EN 1991-1-2, "Eurocode1: Actions on Structures - Part 1-2: General Actions - Actions on Structures Exposed to Fire." CEN, Brussels, 2002.

[19] X. Dai, L. Jiang, J. Maclean, S. Welch, and A. Usmani, "A conceptual framework for a design travelling fire for large compartments with fire resistant islands," in *Proceedings of the 14th International Interflam Conference*, 2016, pp. 1039–1050.

[20] A. S. Usmani, J. M. Rotter, S. Lamont, A. M. Sanad, and M. Gillie, "Fundamental principles of structural behaviour under thermal effects," *Fire Safety Journal*, vol. 36, no. 8, pp. 721–744, 2001.

Chapter 8.

Conclusions and Future Work

8.1 CONCLUSIONS

Structural failure events keep happening in recent years and typically involve a high fire inhomogeneity in large compartments, while relevant structural fire research still relies on highly simplified travelling fire models (i.e. Clifton's model and Rein's model). Meanwhile, no equivalent numerical tools can perform simulations of high computationally efficiency under such fire threats for performance-based structural design, which involves analysis of travelling fires, heat transfer and thermo-mechanical response in one single software package with an automatic coupled manner. This research was carried out on this basis, focusing on developing an extended travelling fire method (ETFM) framework along with an integrated comprehensive tool with high computational efficiency. These two outcomes can be readily used by the community for ensuring the relevant structural safety under the so-called threat due to "travelling fires".

This research reviews the experiments conducted for characterizing travelling fires, in conjunction with the current analytical travelling fire models. It is found that the experiments in which structural response was recorded did not register the mass loss rate of the fuel (which is a key factor to estimate heat release rate of the fire), and vice versa. It suggests more reasonable large scale travelling fire experiments should be set up and carried out, with detailed measurements of required parameters for both the structure and the fire. The features of different travelling fire models are summarized and compared, with particular emphasis on the inherent limitations from each model. It is recognised that the current travelling fire methods are still at an early phase of development, which need to be further improved with 'coupled' new travelling fire experiments under the close collaboration between structural and fire engineers' teams.

An overview of the development of OpenSees software framework for modelling structures in fire over the past decade is also presented. It is found that the research outcomes resulting from thousands of lines of codes for the thermal version of OpenSees are 'scattered' in different conference papers, journal papers, internal reports, and PhD theses, without being systematically summarised and published, since developing and maintaining a software is not regarded as an original scientific

contribution. This research summarizes the 'OpenSees in fire' code development over the past ten years, with addressing its theoretical background, fundamental assumptions, and inherent limitations. It provides a resource for users to cite this software with full current updated information, and for developers to redirect to the corresponding literature.

This research put forward an extended travelling fire method (ETFM) framework, which can predict the fire severity under the travelling fire concept with an upper bound. This upper bound in the framework is achieved through considering the energy and mass conservation, rather than simply forcing other independent models to 'travel' in the large compartment (i.e. modified parametric fire curves in Clifton's model, 800°C-1200°C temperature block and the Alpert's ceiling jet in Rein's model). It is developed based on combining Hasemi's localized fire model for the moving fire plume, and a simple smoke layer calculation by utilising the FIRM zone model for the areas of the compartment away from the fire. The most important aspect of this ETFM framework is bringing the key design elements altogether for performance-based structural design, including fire spread rate, fuel load density, maximum HRR per unit area, ventilation conditions, flashover threshold, fire trajectory, ignition location, heat loss fraction through the compartment boundaries, un-uniform fuel load distributions, and so on. It would gear the structural engineers to generate a series of bounded fire severity scenarios, to ensure the robustness and reliability of their structural design under such fire threats. In addition, design instructions with relevant information which can be readily used by the structural fire engineers are also provided. Finally, the key assumptions of the ETFM framework are rationalised and its limitations are emphasized.

This research developed an OpenSees-based integrated tool called SIFBuilder, which can perform fire modelling, heat transfer analysis, and thermo-mechanical analysis in one single software with an automatic coupled manner and high computational expediency. As explained in the main chapters, the original thermal version of OpenSees consists of a series of calculation components (e.g. various empirical fire models, heat transfer analysis module, and structural analysis module considering both the material nonlinearity and geometrical nonlinearity), which are

independent to each other. The nature of SIFBuilder means that it is a way of constructing all these OpenSees components together in a single analysis, with high computational expediency, to model structures in fires with different level of complexities, such as a single cantilever beam under standard fire, or a plane frame with several compartments under parametric fires, or a 3D structure with large compartment under travelling fires. This manner would facilitate structural engineers to apply fire loading on their design structures like other mechanical loading types (e.g. seismic loading, gravity loading, concentrated loading, etc.), without transferring the fire and heat transfer modelling results to each structural element manually. Furthermore, it would reduce the modelling time expense to a great extent, compared with manually transferring the heat fluxes from the fire modelling to the heat transfer analysis, then interpolating the desired time-temperature histories at the fibres over the cross-section for each structural member. For modelling single structural members or simple portal frame structures under uniform fires, this feature would save time only in the scale of hours on conventional hardware. However, once the modelling tasks upgrade to large structural models involved with realistic fire scenarios, such as full 3D steel-composite structure under travelling fires, this time expense saving feature would become more significant, in the scale of days and even months. This advantage has been approved by the parametric studies carried out in Chapter 7, which involves 29 travelling fire scenarios containing total number of 696 heat transfer analysis for the structural members, which was performed with very modest resources.

This research investigated a series of benchmark problems for further verifying and validating OpenSees/SIFBuilder, against mathematically-based analytical solutions, ABAQUS, SAFIR, and experimental data. These benchmark problems span from: (1) an elastic beam with finite end restraints under through-depth thermal gradient and a UDL, considering both the geometrical nonlinearities and thermal expansions but no material degradations, verifying against analytical solutions and ABAQUS; (2) an elastic-plastic cantilever beam under three-sided standard fire exposure, a UDL, and a concentrated load at beam end, considering both the geometrical and material nonlinearities with different material models of OpenSees/SIFBuilder, validating against SAFIR; (3) an simply-supported beam under a standard fire test, considering both the

geometrical and material nonlinearities, validating against ABAQUS and experimental data; (4) an elastic-plastic beam with pinned supports at two ends under a heating cooling cycle and a UDL, considering both the geometrical and material nonlinearities, validating against the modelling results from other researchers. It is found that the structural fire modelling capabilities of OpenSees/SIFBuilder using displacement-based beam column element with an appropriate steel material model can generate reliable modelling results in most situations (benchmark problem 1-3), except for the buckling-induced unloading phase and cooling phase (benchmark problem 4). However, even under such situations the modelling results from OpenSees/SIFBuilder are still 'qualitatively' useful. Furthermore, these benchmark problems can be used for interested researchers to verify and validate their own numerical or analytical models, and can be also treated as an induction guide for the OpenSees/SIFBuilder software.

This research further implemented more realistic design fires in SIFBuilder, including the two zone models (i.e. ASET zone model & FIRM zone model) and the ETFM framework, in order to produce more optimised and robust structural fire design using SIFBuilder. Both the theoretical background of these fire models and their implementation details in SIFBuilder using C++ are included. In addition, a case study using the ASET zone model and the FIRM zone model in SIFBuilder is investigated, which shows good results agreement between the SIFBuilder and the original ASET-QB and FIRM-QB software.

This research investigates the thermal impact of the ETFM parameters (i.e. fire spread rate, fuel load density) in the temperature design domain, through quantifying the cross-sectional time-temperature evolution of the steel beams with a case study. A series of 29 travelling fire scenarios are investigated, with total number of 696 heat transfer analysis performed using SIFBuilder. The following are some of the key findings: (1) For 'slow' travelling fires (i.e. low fire spread rates), the near-field fire plume brings more dominant thermal impact compared with the impact from far-field smoke. However, for 'fast' travelling fires (i.e. high fire spread rates), the far-field smoke brings more dominant thermal impact. (2) For the sequential beams (right above the fire trajectory) under a 'fast' but 'thin' travelling fire scenario (i.e. high spread rate and low fuel load density), the peak temperatures of these beams would be fluctuating. This is

because some of the structural members are not given enough time to be heated up by the 'fast travelling' near-field fire plume. (3) Fire spread rate and fuel load density are 'equally' discriminating factors on the time for a structural member to reach its peak temperature. Nevertheless, the fire spread rate is a more discriminating factor in affecting the total travelling fire time duration in the ETFM framework. (4) Fire spread rate and fuel load density are 'equally' influential on the maximum temperature that a design beam would reach. (5) For more severe travelling fire scenarios (i.e. high fire spread rates and large fuel load densities), the fire would be entrainment-controlled rather than fuel-controlled. It suggests the ventilation cannot be neglected for a design fire severity using travelling fires. (6) The through depth thermal gradients due to different travelling fire scenarios are extensively explored, especially with regards to the 'thermal gradient reversal' due to the near-field fire plume approaching and leaving the design structural member, as this 'thermal gradient reversal' would fundamentally reverse the thermally-induced bending moment from hogging to sagging. It suggests that the maximum thermal gradient due to smoke-preheating is proportional to the fire spread rate and less sensitive to the fuel load density. The peak thermal gradient due to near-field approaching is more sensitive to the fuel load density than fire spread rate, where larger peak values are captured with lower fuel load densities. Moreover, the reverse peak thermal gradient due to the near-field leaving is also sensitive to the fuel load density rather than the fire spread rate, but this reverse peak value is inversely proportional to the fuel load densities.

This work also presents a demonstration of a full 3D steel-framed structure under the fire impact due to the ETFM framework using SIFBuilder. The impact of the travelling fires on this structure is investigated, through qualitatively demonstrating the structural response of several steel beams in the large compartment.

To conclude, an extended travelling fire method (ETFM) framework along with an integrated comprehensive tool (i.e. SIFBuilder) is developed in this work. The ETFM framework can be regarded as a 'driver', which generates a series of reasonable and diverse fire severity scenarios according to the various design input parameters. SIFBuilder can be regarded as a 'vehicle', which provides the community a tool with high computational expediency to try, test, and further improve this ETFM framework,

and also the SIFBuilder itself. Hence more optimised and robust structural design under such fire threat can be generated and guaranteed, where we believe these efforts will advance the performance-based structural and fire safety engineering.

8.2 FUTURE WORK

To further improve the ETFM framework as a design methodology for ensuring the structural safety under travelling fires, and enhance the availability of SIFBuilder as a design computational tool for structural engineers can utilise in a day-to-day design basis, a series of recommendations are summarized as below.

8.2.1 The ETFM Framework

- It is apparent that none of the travelling fire design methods have been properly validated against any travelling fire experiments so far. To improve the reliability and identify the defects of the current version of the ETFM framework, the thermal impact on the structures due to the ETFM framework should be compared against existing well-documented travelling fire experiments (e.g. the Veselí travelling fire test).
- A 'coupled' new travelling fire experiment under the close collaboration between structural and fire engineers' teams should be carried out, characterising both the structural response and corresponding fire status (i.e. mass loss rate of the fuel) in a single test, to further provide sufficient experimental data to improve the current design travelling fire methods (e.g. the ETFM framework).
- A set of engineering simplifications exist in the current version of the ETFM framework, such as the constant travelling fire spread rate and 1D travelling trajectory. A simplistic fire spread model should be further developed and imbedded to the ETFM framework, to further predict the travelling fire severity in more complex realistic building layouts with considering the impact from ventilations.

- This thesis mainly investigates the impact of the ETFM parameters (i.e. fire spread rate, fuel load density) in the temperature design domain using SIFBuilder; more design parameters should be investigated, such as the ventilation conditions, heat loss fraction through the compartment boundaries, etc.

8.2.2 SIFBuilder

- More benchmark problems for verifying and validating OpenSees/SIFBuilder should be carried out, with a particular focus on the current available temperature dependant concrete material models and corresponding shell elements. Further, the problems of the steel material model under the unloading and cooling phases should be addressed and resolved.
- The impact of the ETFM parameters should be further investigated in the strength design domain using SIFBuilder, which requires more coding work to be done for utilising the dynamic implicit/explicit analysis from OpenSees to SIFBuilder. It will improve the analysis convergence ability while sacrificing a certain level of accuracy, to sustain the thermo-mechanical analysis covering the whole travelling fire period, unlike the current situation, in which the analysis terminates due to the numerical failure of one single structural member as the static implicit analysis is utilised.
- A user manual and a technical guide with a set of examples for SIFBuilder should be made available online, along with all the source codes including the zone models and the ETFM framework.
- A Graphical User Interface (GUI) should be further developed for the thermal version of OpenSees, and SIFBuilder, especially for the post-processor to visualize the structural response under various fire scenarios.

Appendix A

For Chapter 5

A.1 Heated elastic beams with finite end restraints – analytical mathematical model

The attached models in this section refer to the work presented in Chapter 5, section 5.3.1.1. To obtain analytical solutions for beam members considering geometrical nonlinearities, the coupling between the beam internal axial force and beam out-of-plane displacement must be taken into account in the beam theory. For an elastic geometrically nonlinear beam, the strain and curvature of the beam can be obtained through the equations as below:

$$\epsilon_{xx} = \frac{\partial u}{\partial x} + \frac{1}{2}\left(\frac{d\omega}{dx}\right)^2$$
$$\kappa = \frac{d^2\omega}{dx^2} \tag{A.1}$$

where ϵ_{xx} is the beam axial strain, u is the beam axial displacement, ω is the beam transverse displacement, x is the Cartesian coordinates in the beam longitudinal direction, and κ is the beam curvature. The target problem is schematically presented in Figure A.1, where z is the Cartesian coordinates in the beam transverse direction, k_{rl} is the stiffness of the left hand rotational spring, k_{rr} is the stiffness of the right hand rotational spring, and k_t is the stiffness of the right-hand side horizontal spring.

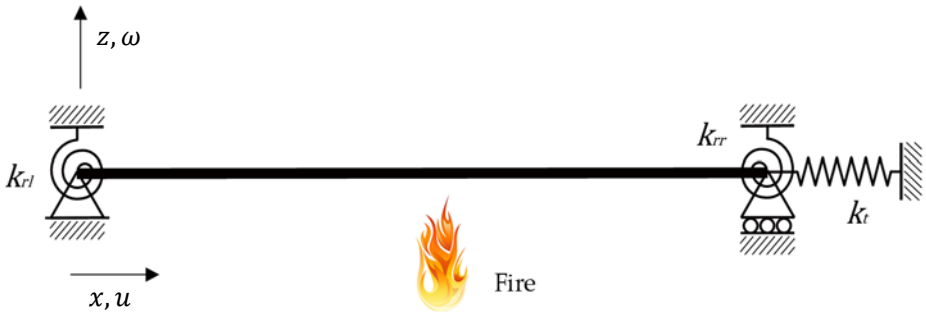


Figure A.1. The schematic of the target beam problem with finite end restraints.

The resultant axial force, N, and bending moment, M, are given as:

$$\begin{aligned} N &= EA\epsilon_{xx} - N^{\theta} \\ M &= -EI\kappa - M^{\theta} \end{aligned} \tag{A.2}$$

where E is the beam elastic modulus, A is the beam cross-sectional area, N^{θ} is the thermal axial force, M^{θ} is the thermal bending moment, and I is the beam moment of inertia which is defined by:

$$I = \int_A z^2 \, dA \tag{A.3}$$

Hence, the beam transverse displacement under high temperatures must satisfy the differential equation as follows:

$$\frac{d^4\omega}{dx^4} - \frac{N}{EI}\frac{d^2\omega}{dx^2} + \frac{1}{EI}\left(\frac{d^2M^{\theta}}{dx^2} + q\right) = 0 \tag{A.4}$$

According to the target problem which is schematically presented in Figure A.1, the boundary conditions on both ends of the beam can be mathematically represented as follows:

$$\begin{aligned} u|_{x=0} &= 0 \\ \omega|_{x=0} &= 0 \\ M_l = EI\left[\frac{d^2\omega}{dx^2}\right]_{x=0} &= k_{rl}\left[\frac{d\omega}{dx}\right]_{x=0} \end{aligned} \tag{A.5}$$

and

$$\begin{aligned} k_t u|_{x=L} &= N \\ \omega|_{x=L} &= 0 \\ M_r = EI\left[\frac{d^2\omega}{dx^2}\right]_{x=L} &= -k_{rl}\left[\frac{d\omega}{dx}\right]_{x=0} \end{aligned} \tag{A.6}$$

where M_l is the bending moment at the left end of the beam, M_r is the bending moment at the right end of the beam, and L is the length of the beam member. In addition to Equation A.5 and Equation A.6, for the beam with special boundary conditions, when the beam is horizontally unrestrained the beam axial force would be zero and horizontal displacement is nonzero; when the beam is fixed at both ends the horizontal displacement would be zero and axial force is nonzero.

Due to the absence of in-plane forces, the beam in-plane equation is obtained through:

$$\frac{\partial N}{\partial x} = 0\,, \qquad or \qquad N = constant \tag{A.7}$$

Since the axial force in the beam is constant along its longitudinal direction, Equation A.2 is integrated, which yields the following equation:

$$\frac{NL}{EA} = u|_{x=L} - u|_{x=0} - \frac{N^{\theta}L}{EA} + \frac{1}{2}\int_0^L \left(\frac{d\omega}{dx}\right)^2 dx \tag{A.8}$$

After substituting the first equation of Equation A.6 to the above Equation A.8, the axial force of the beam can be obtained as:

$$N = \frac{EAk_t}{Lk_t - EA}\left[-\frac{N^{\theta}L}{EA} + \frac{1}{2}\int_0^L \left(\frac{d\omega}{dx}\right)^2 dx\right] \tag{A.9}$$

The reaction forces at the beam supports can be further calculated based on the force equilibrium equations:

$$\begin{gathered} P_l + P_r - qL = 0 \\ M_l - M_r - P_r L + \frac{qL^2}{2} = 0 \end{gathered} \tag{A.10}$$

where P_l is the shear force at the beam left end, P_r is the shear force at the beam right end, and q is the uniformly distributed load (UDL) applied on the whole length of the beam. In general, transverse mechanical loading can be expressed in Fourier series, such as the equation below:

$$q(x) = \sum_{m=1}^{\infty} q_m \sin\frac{m\pi x}{L} \tag{A.11}$$

Then q_m can be obtained by:

$$q_m = \frac{2}{L}\int_0^L q(x) \sin\frac{m\pi x}{L} dx \tag{A.12}$$

Since the mechanical loading type on the beam is transverse UDL, then q_m becomes to:

$$q_m = q \tag{A.13}$$

For sinusoidal loading of magnitude q where $m = 1$:

$$q_m = \frac{4q}{\pi} \tag{A.14}$$

For q applying at a specific point x_0 in the beam longitudinal direction:

$$q_m = \frac{2q}{L}\sin\frac{m\pi x_0}{L} \tag{A.15}$$

The resultant thermal bending moment can be calculated according to the form of transverse mechanical loading:

$$M^{\theta}(x) = \sum_{m=1}^{\infty} M_m^{\theta} \sin\frac{m\pi x}{L} \tag{A.16}$$

where M_m^{θ} can be obtained through Fourier integration:

$$M_m^{\theta} = \frac{4M^{\theta}}{\pi} \tag{A.17}$$

After substituting Equation A.11 and Equation A.16 into Equation A.4, the differential governing equation becomes to:

$$\frac{d^4\omega}{dx^4} - \frac{N}{EI}\frac{d^2\omega}{dx^2} + \frac{1}{EI}\left(-\frac{\pi^2 M_m^{\theta}}{L^2} + q_m\right)\sin\frac{m\pi x}{L} = 0 \tag{A.18}$$

The solution of Equation A.18 contains two parts, one is the homogeneous solution, ω_h , and the other one is the particular solution, ω_p.

$$\omega = \omega_h + \omega_p \tag{A.19}$$

The particular solution can be estimated as follows:

$$\omega_p = C_0 \sin\frac{m\pi x}{L} \tag{A.20}$$

where constant C_0 can be calculated:

$$C_0 = \frac{\pi^2 L^2 M_m^\theta - q_m L^4}{EIm^4\pi^4 + m^2\pi^2 L^2 N} \tag{A.21}$$

And the homogeneous solution can be represented as:

$$\omega_h = C_1 + C_2 x + C_3 \sinh\sqrt{\frac{N}{EI}}x + C_4 \cosh\sqrt{\frac{N}{EI}}x \tag{A.22}$$

Hence the transverse displacement of the beam is given by:

$$\omega = C_0 \sin\frac{m\pi x}{L} + C_1 + C_2 x + C_3 \sinh\sqrt{\frac{N}{EI}}x + C_4 \cosh\sqrt{\frac{N}{EI}}x \tag{A.23}$$

where C_1, C_2, C_3, C_4 are four constants, which can be calculated with the beam boundary conditions due to Equation A.5 and Equation A.6,

$$C_1 + C_4 = 0 \tag{A.24}$$

$$C_4 N - k_{rl}\left(C_0\frac{\pi}{L} + C_2 + C_3\sqrt{\frac{N}{EI}}\right) = 0 \tag{A.25}$$

$$C_1 + C_2 L + C_3 \sinh\sqrt{\frac{N}{EI}}L + C_4 \cosh\sqrt{\frac{N}{EI}}L = 0 \tag{A.26}$$

$$C_3 N \sinh\sqrt{\frac{N}{EI}}L + C_4 N \cosh\sqrt{\frac{N}{EI}}L + k_{rr}\left(-C_0\frac{\pi}{L} + C_2 + C_3\sqrt{\frac{N}{EI}}\cosh\sqrt{\frac{N}{EI}}L + C_4\sqrt{\frac{N}{EI}}\sinh\sqrt{\frac{N}{EI}}L\right) = 0 \quad \text{(A.27)}$$

After substituting Equation A.24 in to Equation A.25 – Equation A.27, then these equations become to:

$$-\frac{k_{rl}\pi}{L}C_0 - NC_1 - k_{rl}C_2 - k_{rl}\sqrt{\frac{N}{EI}}C_3 = 0 \quad \text{(A.28)}$$

$$\left(1 - \cosh\sqrt{\frac{N}{EI}}L\right)C_1 + C_2 L + C_3 \sinh\sqrt{\frac{N}{EI}}L = 0 \quad \text{(A.29)}$$

$$-\frac{k_{rr}\pi}{L}C_0 - \left(N\cosh\sqrt{\frac{N}{EI}}L + k_{rr}\sqrt{\frac{N}{EI}}\sinh\sqrt{\frac{N}{EI}}L\right)C_1 + k_{rr}C_2 + \left(N\sinh\sqrt{\frac{N}{EI}}L + k_{rr}\sqrt{\frac{N}{EI}}\cosh\sqrt{\frac{N}{EI}}L\right)C_3 = 0 \quad \text{(A.30)}$$

Through solving Equations A.28 – A.30, along with Equation A.9 simultaneously, constants C_0, C_1, C_2, C_3 and beam axial force, N, can be obtained. Then the beam transverse displacement, ω , can be calculated using Equation A.23.

Finally, the beam through-depth temperature increase, θ, can be represented as follow:

$$\theta = \left(\frac{1}{2} - \frac{z}{h}\right)\theta_b + \left(\frac{1}{2} + \frac{z}{h}\right)\theta_t \quad \text{(A.31)}$$

where z is the Cartesian coordinates in the beam transverse direction, h is the beam depth, θ_b is the beam bottom surface temperature and θ_t is the beam top surface temperature.

A.2 Heated elastic beams with finite end restraints – numerical model

The attached models in this section refer to the work presented in Chapter 5, section 5.3.1.2. Due to the page limit, only BC6 model using ABAQUS and OpenSees are attached, other models (i.e. BC1-BC5) can be generated by modifying the provided script.

OpenSees_BC6.tcl

```
#############################################################################################
#####################                                                   ####################
#####################                  Types BC6 Beam                   ####################
#####################                                                   ####################
#############################################################################################
# Single steel beam subjected to UDL and Ty and deltaT;
# total 8 elements for 6m beam;
# distributed load UDL=1200N/m; thermal gradient is linear along cross section;
# thermal gradient is defined by Tbot and Ttop;
# 8 fibers along the height 0.2m of the beam section;
# Young's modulus is constant E=2e11 at elevated temperature;
# Thermal expansion coefficient is constant with value of 12e-6;
# SI unit i.e. meter, newton
# Written: Xu Dai
# June 2015, University of Edinburgh;
#--------------------------------------------------------------------------------------------
# Geometric model                                                      ____
#                                                                     |    |
#           |-----------------------------------------|               |    |
#           1    2    3    4    5   ............   n                |____|
#           |<-----------------Lm------------------->|
#
# SET UP -------------------------------------------------------------------------------------

wipe;

model BasicBuilder -ndm 2 -ndf 3;
set dataDir BenchmarkBC6;       # set up name of data directory
file mkdir $dataDir;            # create data directory
source DisplayPlane.tcl;        # procedure for displaying a plane in model
source DisplayModel2D.tcl;      # procedure for displaying 2D perspective of model

# GEOMETRY INFO
set beamSecWidth 0.1;           # beam section width
set beamSecHeight 0.2;          # beam section height

set BeamLen 6.0;                # beam length
set MeshNumBeam 8;              # number of elements across the length of the beam
set EleBeamLen [expr $BeamLen/$MeshNumBeam];                # single beam element
length

# NODES DEFINITION:
# define nodal coordinates for beams
for {set level 1} {$level <=[expr $MeshNumBeam+1]} {incr level 1} {
    set X [expr ($level-1)*$EleBeamLen];
    set nodeID $level;
    node $nodeID $X 0.0;
}

# SINGLE POINT CONSTRAINTS -- BOUNDARY CONDITIONS
set EndNodetag [expr $MeshNumBeam+1];               # single beam right end node
tag
fix 1 1 1 0;                                        # pinned
fix $EndNodetag 0 1 0;                              # roller

# SET UP SPRINGS -- BOUNDARY CONDITIONS (zerolength element to model spring
restraint )
set LSNodetag [expr $EndNodetag+1];                 # left spring node tag
set RSNodetag [expr $EndNodetag+2];                 # right spring node tag
node $LSNodetag 0 0;                                # left spring node definition
node $RSNodetag $BeamLen 0.;                        # right spring node definition
fix $LSNodetag 1 1 1;
fix $RSNodetag 1 1 1;

# DEFINE MATERIAL: with Tag=1 and E=2e11.
set matTag 1;
#uniaxialMaterial Steel01Thermal $matTag 2e30 2e11 1;
uniaxialMaterial ElasticThermal $matTag 2e11 1.2e-5;

# DEFINE MATERIAL: for zeroLength element;
uniaxialMaterial Elastic 2 17.6e6;                  #rotational spring (8*EI/L)
```

```
uniaxialMaterial Elastic 3 1.34e8;                  #translational spring (0.2*EA/L)

# DEFINE FIBRED SECTION
# for beam: patch rect $matTag $numSubdivY $numSubdivZ $yI $zI $yJ $zJ
# origin of section is the center of rectangular
set SecTag 1;
section FiberThermal $SecTag {
# 8 fibres in beam
    set yI [expr -$beamSecHeight/2];
    set zI [expr -$beamSecWidth/2];
    set yJ [expr $beamSecHeight/2];
    set zJ [expr $beamSecWidth/2];
    patch rect 1 8 2 $yI $zI $yJ $zJ;
};

# GEOMTRANSF CHOICE is Linear, Corotational, PDelta ;
set TransfTag 1;
geomTransf Corotational $TransfTag;

# DEFINE ELEMENTS: dispBeamColumnThermal $eleTag $iNode $jNode $numIntgrPts $secTag $TransfTag;
# eles for beams
# "numIntgrPts" is the number of integration points along the element;
# "TransfTag" and "SecTag" are pre-defined coordinate-transformation;
set numIntgrPts 5;
for {set level 1} {$level <=[expr $MeshNumBeam]} {incr level 1} {
    set node1 $level
    set node2 [expr $node1+1]
    set eleID [expr $node1]
    element dispBeamColumnThermal $eleID $node1 $node2 $numIntgrPts $SecTag $TransfTag;
}

# DEFINE ELEMENT: for translational & rotational springs
set Spring1 [expr $MeshNumBeam+1];
set Spring2 [expr $MeshNumBeam+2];
set Spring3 [expr $MeshNumBeam+3];
element zeroLength $Spring1 1 $LSNodetag -mat 2 -dir 6;             #rotational spring
element zeroLength $Spring2 $EndNodetag $RSNodetag -mat 2 -dir 6;   #rotational spring
element zeroLength $Spring3 $EndNodetag $RSNodetag -mat 3 -dir 1;   #translational spring

# DEFINE OUTPUT
set MidNodetag [expr ($MeshNumBeam/2)+1];               # single beam mid-span node tag
recorder Node -file $dataDir/DFreeDisp$MidNodetag.out -time -node $MidNodetag -dof 2 disp;              # displacements of free nodes
recorder Node -file $dataDir/DFreeForce$EndNodetag.out -time -node $EndNodetag -dof 1 reaction;         # axial force in the support of beam - from node output
recorder Element -file $dataDir/Elelocal$MeshNumBeam.out -time -ele $MeshNumBeam localForce;            # axial force in the support of beam - from element output
recorder Element -file $dataDir/Eleglobal$MeshNumBeam.out -time -ele $MeshNumBeam globalForce;          # axial force in the support of beam - from element output

# VIEW THE DEFORMED SHAPE
set ViewScale 0.0001;
DisplayModel2D DeformedShape $ViewScale ;   # display deformed shape, the scaling factor needs to be adjusted for each model

#APPLY UDL
puts "UDL";
set UDL -1200;
pattern Plain 1 Linear {
    for {set level 1} {$level <=[expr $MeshNumBeam]} {incr level 1} {
        set eleID $level ;
        eleLoad -ele $eleID -type -beamUniform $UDL;
    }
};
constraints Plain;
numberer Plain;
system BandGeneral;
```

```
131  test NormDispIncr 1e-8 100;
132  algorithm Newton;
133  integrator LoadControl 1;
134  analysis Static;
135  analyze 1;
136  loadConst -time 0.0;
137
138  #APPLY THERMAL GRADIENT
139  puts "Thermal";
140  set T1 1000;
141  set T2 0;
142  set Y1 [expr -$beamSecHeight/2];
143  set Y2 [expr $beamSecHeight/2];
144  pattern Plain 2 Linear {
145
146  for {set level 1} {$level <=[expr $MeshNumBeam]} {incr level 1} {
147      set eleID $level
148      eleLoad -ele $eleID -type -beamThermal $T1 $Y1 $T2 $Y2 ;
149  }
150  }
151  integrator LoadControl 0.01
152  analysis Static;
153  analyze 100;
154  loadConst -time 0.0;
155  puts "Done!"
156  wipe;
157
158
159
160
```

ABAQUS_BC6.inp

```
**This model is for Types BC6 Beam, which is a single steel beam subjected to UDL
and Ty and deltaT. Total 10 elements for 6m beam, with distributed load UDL=1200N/m.
Thermal gradient is linear along cross section. Young's modulus is constant E=2e11
at elevated temperature. Thermal expansion coefficient is constant with value of
12e-6.
**SI units: kg, m, s, N.
**Written: Xu Dai, September 2017, University of Edinburgh.
**
**********************************************************************
**********************************************************************
**********           MESH FOR THE MDOEL            **************
**********************************************************************
**********************************************************************
**
*****NODE GENERATION*****
**
*NODE
1,0,0,0
101,6,0,0
1000,0,0,0
1001,6,0,0
1002,6,0,0
*NGEN,NSET=BEAM_NODES
1,101,10
*NSET, NSET=ALLNODES
BEAM_NODES,1000,1001,1002
**
*****ELEMENT GENERATION*****
**
*ELEMENT,TYPE=B31
1,1,11
*ELGEN,ELSET=BEAM_ELES
1,10,10,10
**
*****SPRING ELEMENT DEFINITION*****
**
*ELEMENT,TYPE=CONN3D2,ELSET=ROTATIONAL_SPRING_BOTH_ENDS
1000,1000,1
1001,101,1001
*ELEMENT,TYPE=CONN3D2,ELSET=TRANS_SPRING_RIGHT_END
1002,101,1002
*ELSET,ELSET=ALL_BOUNDARY_ELEMENTS
ROTATIONAL_SPRING_BOTH_ENDS,TRANS_SPRING_RIGHT_END
**
*****BEAM SECTION DEFINITION*****
**
*BEAM SECTION,ELSET=BEAM_ELES,MATERIAL=ELASTIC_CONST,SECTION=RECT,TEMPERATURE=VALUES
0.1,0.2
0,0,-1
**
*****SPRING ELEMENT SECTION DEFINITION*****
**
*CONNECTOR SECTION,ELSET=ROTATIONAL_SPRING_BOTH_ENDS,
BEHAVIOR=BOTH_ENDS_ROTATIONAL_SPRING
ROTATION
*CONNECTOR BEHAVIOR,NAME=BOTH_ENDS_ROTATIONAL_SPRING
*CONNECTOR ELASTICITY,COMPONENT=6
17.6e+6,
*CONNECTOR SECTION,ELSET=TRANS_SPRING_RIGHT_END, BEHAVIOR=RIGHT_END_TRANS_SPRING
AXIAL
*CONNECTOR BEHAVIOR,NAME=RIGHT_END_TRANS_SPRING
*CONNECTOR ELASTICITY,COMPONENT=1
1.34e+8,
**
**********************************************************************
**********************************************************************
**  DEFINITION OF MATERIAL CARACTERISTICS UNDER TEMPERATURE EFFECT **
**                      ( MATERIAL DEFINITION )                     **
**********************************************************************
**********************************************************************
**
```

```
*MATERIAL, NAME=ELASTIC_CONST
**Density
**7.85e-06,
*ELASTIC
2.1e+11,0.3,   20.0
2.1e+11,0.3,  100.0
2.1e+11,0.3,  200.0
2.1e+11,0.3,  300.0
2.1e+11,0.3,  400.0
2.1e+11,0.3,  500.0
2.1e+11,0.3,  600.0
2.1e+11,0.3,  700.0
2.1e+11,0.3,  800.0
2.1e+11,0.3,  900.0
2.1e+11,0.3, 1000.0
2.1e+11,0.3, 1100.0
2.1e+11,0.3, 1200.0
*EXPANSION
1.2e-5,20.
1.2e-5,100.
1.2e-5,200.
1.2e-5,300.
1.2e-5,400.
1.2e-5,500.
1.2e-5,600.
1.2e-5,700.
1.2e-5,800.
1.2e-5,900.
1.2e-5,1000.
1.2e-5,1100.
1.2e-5,1200.
**********************************************************************
**********************************************************************
**                APPLLYING BOUNDARY CONDITIONS                     **
**********************************************************************
**********************************************************************
**
*****BOUNDARY CONDITIONS*****
**
*BOUNDARY
1,1,5
101,2,5
1000,1,6
1001,1,6
1002,1,6
**
*****PREDEFINED FIELDS FOR INITIAL TEMPERATURE*****
**
*INITIAL CONDITIONS,TYPE=TEMPERATURE
BEAM_NODES,20,20,20,20
**********************************************************************
**********************************************************************
**                     APPLLYING STATIC LOADS                       **
**********************************************************************
**********************************************************************
*STEP, NAME=STATIC-LOADS, NLGEOM=YES
*STATIC
0.1, 1., 1e-05, 1.
**
*****APPLY UDL*****
**
*DLOAD
BEAM_ELES, P2, -1200
**
*****OUTPUT REQUEST*****
**
*OUTPUT, FIELD
*NODE OUTPUT, NSET=ALLNODES
NT, U, CF, RF
*ELEMENT OUTPUT, ELSET=BEAM_ELES
TEMP, SF, S, E
```

```
*ELEMENT OUTPUT, ELSET=ALL_BOUNDARY_ELEMENTS
CU
*OUTPUT, HISTORY
*NODE OUTPUT, NSET=ALLNODES
NT, U, CF, RF
*ELEMENT OUTPUT, ELSET=BEAM_ELES
TEMP, SF, S, E
*ELEMENT OUTPUT, ELSET=ALL_BOUNDARY_ELEMENTS
CU
*End Step
**********************************************************************
**********************************************************************
**              APPLLYING TEMPERATURE ON THE BEAM                   **
**********************************************************************
**********************************************************************
*STEP, NAME=FIRE-LOADS, NLGEOM=YES
*STATIC
1, 100, 0.1, 10
**
** PREDEFINED FIELDS
**
*Temperature
BEAM_NODES,1020,1020,20,20
**
*****OUTPUT REQUEST*****
**
*OUTPUT, FIELD
*NODE OUTPUT, NSET=ALLNODES
NT, U, CF, RF
*ELEMENT OUTPUT, ELSET=BEAM_ELES
TEMP, SF, S, E
*ELEMENT OUTPUT, ELSET=ALL_BOUNDARY_ELEMENTS
CU
*OUTPUT, HISTORY
*NODE OUTPUT, NSET=ALLNODES
NT, U, CF, RF
*ELEMENT OUTPUT, ELSET=BEAM_ELES
TEMP, SF, S, E
*ELEMENT OUTPUT, ELSET=ALL_BOUNDARY_ELEMENTS
CU
*End Step
```

A.3 SIFBuilder examples

The attached models in this section refer to the work presented in Chapter 5, section 5.3.2. Its corresponding SIFBuilder example is presented in *SIFBuilder_Example_15.tcl*. More benchmark problems generated by SIFBuilder are also attached. Their model descriptions are annotated with the model scripts.

SIFBuilder_Example_11.tcl

```
#########################################################################
#####                                                               #####
##### M 0 N (X-Z grillage with out of plane displacements - 3D)     #####
#####                                                               #####
#########################################################################

# SI unit i.e. meter, newton, second
# Written by: Xu Dai (x.dai@ed.ac.uk)
# Aug 2015, University of Edinburgh

wipe;
source DisplayPlane.tcl;
source DisplayModel2D.tcl;
source DisplayModel3D.tcl;
set dataDir 11-Grillage;
file mkdir $dataDir;
file mkdir 11-HTData;

# Define STRUCTURAL MODEL
SIFBuilder Frame;
SIFZBay 1 1;
SIFXBay 2 2 2 2;
AddMaterial steel 1 -type EC3 3e8  2e11;
AddSection ISection 1 1 0.203 0.102 0.0054 0.009;
AssignSection beams 1;
SetBC fixedJoint -Locx 0;
SetBC fixedJoint -Locx 8;

# Define LOADING
AddLoad -joint 311 321 331 -load 0 -1000 0;
AddFire -compartment 211 311 -type standard;

# Mesh MODEL
BuildModel -MeshCtrl 6 0 6;

# Define DISPLAY
set  xPixels 300;
set  yPixels 1000;
set  xLoc1 10;
set  yLoc1 10;
set ViewScale 0.001;
DisplayModel3D DeformedShape $ViewScale $xLoc1 $yLoc1 $xPixels $yPixels 0;

# Define OUTPUT RESULTS
SIFRecorder Joint -file $dataDir/SIFJoint111.out -time -joint 111 disp;
SIFRecorder Joint -file $dataDir/SIFJoint131.out -time -joint 131 disp;
SIFRecorder Joint -file $dataDir/SIFJoint511.out -time -joint 511 disp;
SIFRecorder Joint -file $dataDir/SIFJoint311.out -time -joint 311 disp;
SIFRecorder Joint -file $dataDir/SIFJoint321.out -time -joint 321 disp;
SIFRecorder Joint -file $dataDir/SIFJoint331.out -time -joint 331 disp;

# Apply LOADS & Define ANALYSIS & Define HT OUTPUT
time {
SIFAnalyze SelfWeight -dt 0.5 Load -dt 0.1 Fire -dt 10 -duration 1800 -output 11-
HTData;
}

# Print KEY INFO
print $dataDir/domain.out

wipe;

```

SIFBuilder_Example_12.tcl

```
1  ######################################################################
2  #####                                                            #####
3  #####    M N 0 (X-Y frame with in plane displacements - 3D)    #####
4  #####                                                            #####
5  ######################################################################
6
7  # SI unit i.e. meter, newton, second
8  # Written by: Xu Dai (x.dai@ed.ac.uk)
9  # Aug 2015, University of Edinburgh
10
11 wipe;
12 source DisplayPlane.tcl;
13 source DisplayModel2D.tcl;
14 source DisplayModel3D.tcl;
15 set dataDir 12.1-PlaneFrame;
16 file mkdir $dataDir;
17 file mkdir 12.1-HTData;
18
19 # Define STRUCTURAL MODEL
20 SIFBuilder Frame;
21 SIFXBay 3 3 3 3;
22 SIFStorey 2 2;
23 AddMaterial steel 1 -type EC3 3e8  2e11;
24 AddSection ISection 1 1 0.203 0.102 0.0054 0.009;
25 AddSection ISection 2 1 0.203 0.203 0.007 0.011;
26 AssignSection beams 1;
27 AssignSection columns 2;
28 SetBC fixedJoint -Locy 0;
29
30 # Define LOADING
31 AddLoad -joint 111 112 511 512 -load 20000 0 0;
32 AddFire -compartment 111 211 -type standard;
33
34 # Mesh MODEL
35 BuildModel -MeshCtrl 6 6 0;
36
37 # Define DISPLAY
38 set  xPixels 1500;
39 set  yPixels 600;
40 set  xLoc1 10;
41 set  yLoc1 10;
42 set ViewScale 10;
43 DisplayModel3D DeformedShape $ViewScale $xLoc1 $yLoc1 $xPixels $yPixels 0;
44
45 # Define OUTPUT RESULTS
46 SIFRecorder Joint -file $dataDir/SIFJoint110.out -time -joint 110 disp;
47 SIFRecorder Joint -file $dataDir/SIFJoint210.out -time -joint 210 disp;
48 SIFRecorder Joint -file $dataDir/SIFJoint310.out -time -joint 310 disp;
49 SIFRecorder Joint -file $dataDir/SIFJoint410.out -time -joint 410 disp;
50 SIFRecorder Joint -file $dataDir/SIFJoint111.out -time -joint 111 disp;
51 SIFRecorder Joint -file $dataDir/SIFJoint112.out -time -joint 112 disp;
52
53 # Apply LOADS & Define ANALYSIS & Define HT OUTPUT
54 time {
55 SIFAnalyze SelfWeight -dt 0.5 Load -dt 0.1 Fire -dt 30 -duration 1800 -output 12.1-
   HTData;
56 }
57
58 # Print KEY INFO
59 print $dataDir/domain.out
60
61 wipe;
62
```

SIFBuilder_Example_13.tcl

```
#######################################################################
#####                                                             #####
#####  0 M N (Y-Z frame with out of plane displacements - 3D) #####
#####                                                             #####
#######################################################################

# SI unit i.e. meter, newton, second
# Written by: Xu Dai (x.dai@ed.ac.uk)
# Aug 2015, University of Edinburgh

wipe;
source DisplayPlane.tcl;
source DisplayModel2D.tcl;
source DisplayModel3D.tcl;
set dataDir 13.2-PlaneFrame;
file mkdir $dataDir;
file mkdir 13.2-HTData;

# Define STRUCTURAL MODEL
SIFBuilder Frame;
SIFZBay 3 3 3 3;
SIFStorey 2 2;
AddMaterial steel 1 -type EC3 3e8  2e11;
AddSection ISection 1 1 0.203 0.102 0.0054 0.009;
AddSection ISection 2 1 0.203 0.203 0.007 0.011;
AssignSection beams 1;
AssignSection columns 2;
SetBC fixedJoint -Locy 0;

# Define LOADING
AddLoad -joint 112 -load 2000 0 0;
AddFire -compartment 111 121 -type standard;

# Mesh MODEL
BuildModel -MeshCtrl 0 6 6;

# Define DISPLAY
set  xPixels 1500;
set  yPixels 600;
set  xLoc1 10;
set  yLoc1 10;
set ViewScale 0.001;
DisplayModel3D DeformedShape $ViewScale $xLoc1 $yLoc1 $xPixels $yPixels 0;

# Define OUTPUT RESULTS
SIFRecorder Joint -file $dataDir/SIFJoint110.out -time -joint 110 disp;
SIFRecorder Joint -file $dataDir/SIFJoint111.out -time -joint 111 disp;
SIFRecorder Joint -file $dataDir/SIFJoint112.out -time -joint 112 disp;

# Apply LOADS & Define ANALYSIS & Define HT OUTPUT
time {
SIFAnalyze SelfWeight -dt 0.5 Load -dt 0.1 Fire -dt 30 -duration 1800 -output 13.2-
HTData;
}

# Print KEY INFO
print $dataDir/domain.out

wipe;

```

SIFBuilder_Example_14.tcl

```
#######################################################################
#####                                                             #####
#####   0 N 0 (Y single storey or multi-storey column - 3D)       #####
#####                                                             #####
#######################################################################

# SI unit i.e. meter, newton, second
# Written by: Xu Dai (x.dai@ed.ac.uk)
# July 2015, University of Edinburgh

wipe;
source DisplayPlane.tcl;
source DisplayModel2D.tcl;
source DisplayModel3D.tcl;
set dataDir 14-SingleColumn;
file mkdir $dataDir;
file mkdir 14-HTData;

# Define STRUCTURAL MODEL
SIFBuilder Frame;
SIFStorey 1.5 1.5 1.5;
AddMaterial steel 1 -type EC3 3e8  2e11;
AddSection ISection 1 1 0.203 0.203 0.007 0.011;
AssignSection columns 1;
SetBC fixedJoint -Locy 0;
SetBC fixedJoint -Locy 4.5;

# Define LOADING
AddLoad -joint 112 -load -1000 0 0;
AddFire -storey 111 112 113 -type standard;

# Mesh MODEL
BuildModel -MeshCtrl 0 6 0;

# Define DISPLAY
set  xPixels 500;
set  yPixels 1000;
set  xLoc1 10;
set  yLoc1 10;
set ViewScale 0.01;
DisplayModel3D DeformedShape $ViewScale $xLoc1 $yLoc1 $xPixels $yPixels 0;

# Define OUTPUT RESULTS
SIFRecorder Joint -file $dataDir/SIFJoint111.out -time -joint 111 disp;
recorder Node -file $dataDir/Node2.out -time -node 2 -dof 1 2 3 disp;

# Apply LOADS & Define ANALYSIS & Define HT OUTPUT
time {
SIFAnalyze SelfWeight -dt 1 Load -dt 0.1 Fire -dt 5 -duration 1800 -output 14-HTData;
}

# Print KEY INFO
print $dataDir/domain.out;

wipe;

```

SIFBuilder_Example_15.tcl

```
#######################################################################
#####                                                             #####
#####      M 0 0 (X single Beam or continuous beams - 3D)         #####
#####                                                             #####
#######################################################################

# This file is for verifying SIFBuilder - 7 examples - No.15
# Written by: Xu Dai (x.dai@ed.ac.uk), Jan 2017, University of Edinburgh
# SI unit i.e. meter, newton, second

wipe;
set dataDir 7Examples-No.15-SingleBeam;
file mkdir $dataDir;

source DisplayPlane.tcl;
source DisplayModel2D.tcl;
source DisplayModel3D.tcl;

# Define STRUCTURAL MODEL
SIFBuilder Frame;
SIFXBay 2 2;
#AddMaterial steel 1 -type Steel01Thermal 3.55e8  2e11;
AddMaterial steel 1 -type EC3 3.55e8  2e11;
AddSection ISection 1 1 0.3973 0.1418 0.0063 0.0086;    #UB 406x140x39
AssignSection beams 1;

# Set BOUNDARY CONDITION
SetBC fixedJoint -Locx 0;
#SetBC fixedJoint -Locx 4;

# Define LOADING
AddLoad -joint 311 -load 0 -2000 0;

# FIRE DEFINITION
AddFire -compartment 111 211 -type standard;

# Mesh MODEL
BuildModel -MeshCtrl 6 0 0;

# Define DISPLAY
set  xPixels 800;
set  yPixels 800;
set  xLoc1 10;
set  yLoc1 10;
set ViewScale 0.001;
DisplayModel3D DeformedShape $ViewScale $xLoc1 $yLoc1 $xPixels $yPixels 0;

# Define OUTPUT RESULTS
SIFRecorder Joint -file $dataDir/Joint111.out -time -joint 111 disp;
SIFRecorder Joint -file $dataDir/Joint311.out -time -joint 311 disp;
recorder Node -file $dataDir/Node1.out -time -node 1 -dof 1 2 3 disp;
recorder Node -file $dataDir/Node3.out -time -node 3 -dof 1 2 3 disp;

# Apply LOADS & Define ANALYSIS & Define HT OUTPUT
time {
#SIFAnalyze Load -dt 0.1 Fire -dt 10 -duration 1800 -output $dataDir;
SIFAnalyze SelfWeight -dt 0.1 Load -dt 0.1 Fire -dt 10 -duration 1800 -output
$dataDir;
}

# Print KEY INFO
print $dataDir/domain15.out

wipe;
# For debugging: puts "The codes above are fine!";
```

SIFBuilder_Example_16.tcl

```
#######################################################################
#####                                                             #####
#####      0 0 N (Z single Beam or continuous beams - 3D)         #####
#####                                                             #####
#######################################################################

# SI unit i.e. meter, newton, second
# Written by: Xu Dai (x.dai@ed.ac.uk)
# Aug 2015, University of Edinburgh

wipe;
source DisplayPlane.tcl;
source DisplayModel2D.tcl;
source DisplayModel3D.tcl;
set dataDir 16-SingleBeam;
file mkdir $dataDir;
file mkdir 16-HTData;

# Define STRUCTURAL MODEL
SIFBuilder Frame;
SIFZBay 2 2;
AddMaterial steel 1 -type EC3 3e8  2e11;
AddSection ISection 1 1 0.203 0.203 0.007 0.011;
AssignSection beams 1;
SetBC fixedJoint -Locz 0;
SetBC pinnedJoint -Locz 2;

# Define LOADING
AddLoad -joint 131 -load 0 -2000 0;
AddFire -bay 111 121 -type standard;

# Mesh MODEL
BuildModel -MeshCtrl 0 0 6;

# Define DISPLAY
set  xPixels 800;
set  yPixels 800;
set  xLoc1 10;
set  yLoc1 10;
set ViewScale 0.001;
DisplayModel3D DeformedShape $ViewScale $xLoc1 $yLoc1 $xPixels $yPixels 0;

# Define OUTPUT RESULTS
SIFRecorder Joint -file $dataDir/Joint131.out -time -joint 131 disp;

# Apply LOADS & Define ANALYSIS & Define HT OUTPUT
time {
SIFAnalyze Load -dt 0.1 Fire -dt 10 -duration 1800 -output 16-HTData;
}

# Print KEY INFO
print $dataDir/domain16.out

wipe;

```

A.4 A Modelling Benchmark for Cooling

The attached model refers to the work presented in Chapter 5, section 5.3.4.

```
##########################################################################################
# Martin Gillie's benchmark 1st test for cooling;
# Single steel beam subjected to UDL (4250N/m) and uniformly heated then cooled;
# total 8 elements for 1m beam;
# thermal load is defined by Tbot and Ttop;
# 8 fibers along the height 0.035m of the beam section;
# Temperature-independent elasto-plastic steel is used with E=2.07e11;
# Yield strength reducing linearly from 250MPa at 0C to 0 at 1000C;
# expansion coefficient is constant with value of 12e-6;
# SI unit i.e. meter, newton;
# Written: Xu Dai
# Feb 2015,  University of Edinburgh
#-----------------------------------------------------------------------------------------
# Geometric model                                                    _______
#                                                                   |       |
#           |---------------------------------------|               |       | 0.035
#        21&1    2     3     4     5     6     7     8     9&22     |_______|
#           |<-----------------1m------------------>|               0.035
#
# SET UP ----------------------------------------------------------------------------------
# units: m,N
wipe;

model BasicBuilder -ndm 2 -ndf 3;
set dataDir Data;                # set up name of data directory (can remove this)
file mkdir $dataDir;             # create data directory

source DisplayPlane.tcl
source DisplayModel2D.tcl

# nodal coordinates:
node 1 0 0;
node 2 0.125 0;
node 3 0.25 0;
node 4 0.375 0;
node 5 0.5 0;
node 6 0.625 0;
node 7 0.75 0;
node 8 0.875 0;
node 9 1 0;

# assign lumped nodal mass for applying HHT integrator:
# (35*35e-6*1*7850)/8=1.202kg; 1.202/2=0.601kg
#mass 1 0.601 0.601 0;
#mass 9 0.601 0.601 0;
#for {set node 2} {$node <=8} {incr node 1} {
#   set value 1.202;
#   mass $node $value $value 0;
#   }

#node 21,22 for zerolength element to model spring(restraint)
#node 21 0 0;
#node 22 1 0;

# Single point constraints -- Boundary Conditions
fix 1 1 1 0;
fix 9 1 1 0;

#fix 21 1 1 1;
#fix 22 1 1 1;

#define an elastic material with Tag=1 and E=2.07e11.
set matTag 1;
uniaxialMaterial Steel01Thermal $matTag 2.5e8 2.07e11 0.0000001;

#define fibred section: fiber $yLoc $zLoc $A $matTag
#origin of section is the center of rectangular
set SecTag 1;
section FiberThermal $SecTag  {
#8fibers
        fiber 0.0021875 0 0.000153125 $matTag;
```

```
72          fiber 0.0065625 0 0.000153125 $matTag;
73          fiber 0.0109375 0 0.000153125 $matTag;
74          fiber 0.0153125 0 0.000153125 $matTag;
75          fiber -0.0021875 0 0.000153125 $matTag;
76          fiber -0.0065625 0 0.000153125 $matTag;
77          fiber -0.0109375 0 0.000153125 $matTag;
78          fiber -0.0153125 0 0.000153125 $matTag;
79
80   };
81
82   # define geometric transformation: Linear, PDelta, Corotational
83   set TransfTag 1;
84   geomTransf Corotational $TransfTag;
85
86   #define beam element: dispBeamColumnThermal $eleTag $iNode $jNode $numIntgrPts
     $secTag $TransfTag <-mass $massDens>;
87   #"numIntgrPts" is the number of integration points along the element;
88   #"TransfTag" is pre-defined coordinate-transformation;
89   #"massDens" is the element mass density (per unit length), from which a lumped-mass
     matrix is formed (optional, default=0.0)
90   set numIntgrPts 3;
91   element dispBeamColumnThermal 1 1 2 $numIntgrPts $SecTag $TransfTag -mass 7850;
92   element dispBeamColumnThermal 2 2 3 $numIntgrPts $SecTag $TransfTag -mass 7850;
93   element dispBeamColumnThermal 3 3 4 $numIntgrPts $SecTag $TransfTag -mass 7850;
94   element dispBeamColumnThermal 4 4 5 $numIntgrPts $SecTag $TransfTag -mass 7850;
95   element dispBeamColumnThermal 5 5 6 $numIntgrPts $SecTag $TransfTag -mass 7850;
96   element dispBeamColumnThermal 6 6 7 $numIntgrPts $SecTag $TransfTag -mass 7850;
97   element dispBeamColumnThermal 7 7 8 $numIntgrPts $SecTag $TransfTag -mass 7850;
98   element dispBeamColumnThermal 8 8 9 $numIntgrPts $SecTag $TransfTag -mass 7850;
99
100  # first define material for zeroLength element;
101  # elasticity modulus is equal to stiffness of spring
102  # dir 6 is the rotation around z; dir 1 is the translational dof
103  #uniaxialMaterial Elastic 3 2.54e8;         #translational spring
104  #uniaxialMaterial Elastic 2 25885.8;        #rotational spring
105
106  #define element for translational&rotational spring
107  #element zeroLength 9 9 22 -mat 3 -dir 1;   #translational spring
108  #element zeroLength 10 1 21 -mat 3 -dir 1;  #translational spring
109  #element zeroLength 121 1 21 -mat 2 -dir 6; #rotational spring
110  #element zeroLength 922 9 22 -mat 2 -dir 6; #rotational spring
111
112  # Define RECORDERS ----------------------------------------------------------------
113  recorder Node -file Data/DFree5.out -time -node 5 -dof 2 disp;          # mid-span
     deflection
114  recorder Element -file Data/Ele4.out -time -ele 4 localForce;           # axial force
115
116  # view the deformed shape
117  set ViewScale 0.001;
118  DisplayModel2D DeformedShape $ViewScale ;   # display deformed shape, the scaling
     factor needs to be adjusted for each model
119
120  #define load
121  #first add a UDL
122  set UDL -4250;
123  pattern Plain 1 Linear {
124  eleLoad -ele 1 -type -beamUniform $UDL 0. 0.;
125  eleLoad -ele 2 -type -beamUniform $UDL 0. 0.;
126  eleLoad -ele 3 -type -beamUniform $UDL 0. 0.;
127  eleLoad -ele 4 -type -beamUniform $UDL 0. 0.;
128  eleLoad -ele 5 -type -beamUniform $UDL 0. 0.;
129  eleLoad -ele 6 -type -beamUniform $UDL 0. 0.;
130  eleLoad -ele 7 -type -beamUniform $UDL 0. 0.;
131  eleLoad -ele 8 -type -beamUniform $UDL 0. 0.;
132  }
133  constraints Plain;
134  numberer Plain;
135  system BandGeneral;
136  test NormDispIncr 1e-3 180;
137  algorithm Newton;
138  integrator LoadControl 1;
```

```
analysis Static;
analyze 1;
loadConst -time 0.0

# define thermal load (i.e. temperature distribution in section)
# define thermal load path
set Path "Series -time {0   10  20  30  40  50  60  70  80  90  100 110 120 130 140
150 160 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330 340 350
360 370 380 390 400 410 420 430 440 450 460 470 480 490 500 510 520 530 540 550 560
570 580 590 600 610 620 630 640 650 660 670 680 690 700 710 720 730 740 750 760 770
780 790 800 810 820 830 840 850 860 870 880 890 900 910 920 930 940 950 960 970 980
990 1000    1010    1020    1030    1040    1050    1060    1070    1080    1090
1100    1110    1120    1130    1140    1150    1160    1170    1180    1190
1200    1210    1220    1230    1240    1250    1260    1270    1280    1290
1300    1310    1320    1330    1340    1350    1360    1370    1380    1390
1400    1410    1420    1430    1440    1450    1460    1470    1480    1490
1500    1510    1520    1530    1540    1550    1560    1570    1580    1590
1600} -values {0    0.0125  0.025   0.0375  0.05    0.0625  0.075   0.0875  0.1
0.1125  0.125   0.1375  0.15    0.1625  0.175   0.1875  0.2 0.2125  0.225   0.2375
0.25    0.2625  0.275   0.2875  0.3 0.3125  0.325   0.3375  0.35    0.3625  0.375
0.3875  0.4 0.4125  0.425   0.4375  0.45    0.4625  0.475   0.4875  0.5 0.5125
0.525   0.5375  0.55    0.5625  0.575   0.5875  0.6 0.6125  0.625   0.6375  0.65
0.6625  0.675   0.6875  0.7 0.7125  0.725   0.7375  0.75    0.7625  0.775   0.7875
0.8 0.8125  0.825   0.8375  0.85    0.8625  0.875   0.8875  0.9 0.9125  0.925
0.9375  0.95    0.9625  0.975   0.9875  1   0.9875  0.975   0.9625  0.95    0.9375
0.925   0.9125  0.9 0.8875  0.875   0.8625  0.85    0.8375  0.825   0.8125  0.8
0.7875  0.775   0.7625  0.75    0.7375  0.725   0.7125  0.7 0.6875  0.675   0.6625
0.65    0.6375  0.625   0.6125  0.6 0.5875  0.575   0.5625  0.55    0.5375  0.525
0.5125  0.5 0.4875  0.475   0.4625  0.45    0.4375  0.425   0.4125  0.4 0.3875
0.375   0.3625  0.35    0.3375  0.325   0.3125  0.3 0.2875  0.275   0.2625  0.25
0.2375  0.225   0.2125  0.2 0.1875  0.175   0.1625  0.15    0.1375  0.125   0.1125
0.1 0.0875  0.075   0.0625  0.05    0.0375  0.025   0.0125  0}"

# -beamThermal $T1 $LocY1 $T2 LocY2....; two temperature means uniform or linear
temperature distribution
# T1 is the bottom temp of beam and T2 is the top
# Y1 is the coordinate of bottom of beam section; Y2 is for top
# the temperature will be interpolated along the section
pattern Plain 2 $Path $Path $Path $Path $Path $Path $Path $Path $Path {
set T1 800;set T2 800;
set Y1 -0.0175;set Y2 0.0175;
eleLoad -ele 1 -type -beamThermal $T1 $Y1 $T2 $Y2 ;eleLoad -ele 2 -type -beamThermal
$T1 $Y1 $T2 $Y2 ;
eleLoad -ele 3 -type -beamThermal $T1 $Y1 $T2 $Y2 ;eleLoad -ele 4 -type -beamThermal
$T1 $Y1 $T2 $Y2 ;
eleLoad -ele 5 -type -beamThermal $T1 $Y1 $T2 $Y2 ;eleLoad -ele 6 -type -beamThermal
$T1 $Y1 $T2 $Y2 ;
eleLoad -ele 7 -type -beamThermal $T1 $Y1 $T2 $Y2 ;eleLoad -ele 8 -type -beamThermal
$T1 $Y1 $T2 $Y2 ;
}

constraints Transformation;
#integrator LoadControl 1;
integrator HHT 0.7;
analysis Transient;                     # define type of analysis: time-dependent
#analysis Static;
#analyze 1600;
#analyze $Nstep $time-step_increment;
analyze 16000 0.1;
loadConst -time 0.0;
wipe;
```

Appendix B

For Chapter 6

B.1 Header files for the zone models in C++

The definition of a class in C++ consists of a header (.h) file, which declares the data and methods, and a source (.cpp) file, which is the implementation detail of the declared methods from the corresponding header file. The source files of the *ZoneModel_ASET* class and *ZoneModel_FIRM* class are not presented in this thesis due to the limit on space. Nevertheless, corresponding header files are attached in here, with all the related data variables and function methods explained with detailed annotations. The theoretical background of these two implemented zone models are discussed in Chapter 6, section 6.3.

ZoneModel_ASET.h

```
/* ****************************************************************** **
**    OpenSees - Open System for Earthquake Engineering Simulation    **
**          Pacific Earthquake Engineering Research Center             **
**                                                                     **
**                                                                     **
** (C) Copyright 2001, The Regents of the University of California    **
** All Rights Reserved.                                               **
**                                                                     **
** Commercial use of this program without express permission of the   **
** University of California, Berkeley, is strictly prohibited.  See   **
** file 'COPYRIGHT'  in main directory for information on usage and   **
** redistribution,  and for a DISCLAIMER OF ALL WARRANTIES.           **
**                                                                     **
** Fire & Heat Transfer modules developed by:                         **
**   Yaqiang Jiang (y.jiang@ed.ac.uk)                                **
**   Liming Jiang (Liming.Jiang@ed.ac.uk)                            **
**   Xu Dai (x.dai@ed.ac.uk)                                         **
**   Asif Usmani (asif.usmani@ed.ac.uk)                              **
**                                                                     **
** ****************************************************************** */

// $Date: 2017-01-20
// $Source: /usr/local/cvs/OpenSees/SRC/fire/ZoneModel/ZoneModel_ASET.h

// Written by: Xu Dai (x.dai@ed.ac.uk)
// Description: This file contains the class implementation of ASET Zone Model,
//              according to the book called 'An Introduction to Mathematical Fire
//              Modeling' book name is referenced in the codes as: AITMFM


#ifndef ZoneModel_ASET_h
#define ZoneModel_ASET_h

#include <FireModel.h>
#include <TravellingFireFuel.h>

class ZoneModel_ASET : public FireModel
{

public:
    ZoneModel_ASET(int tag, int theSmokeGenModelTypeTag, double theconLc,
        double theStoreyH, double theTotalFloorArea, TravellingFireFuel*
        theIgnitionTFfuel, bool theFireTravelDirection, double startTime = 0.0);
    virtual ~ZoneModel_ASET();

    void applyFluxBC(HeatFluxBC* theFlux, double time);

    //linearly interpolate the fire parameters at a certain time point
    double Interpolate_FirePars(double time, const Vector& TriTimePnts,
        const Vector& FirePars);

    //calculate the derivatives of smoke status with time
    //SmokeStatus(0): height of the zone free of smoke
    //SmokeStatus(1): upper hot layer temperature
    //theDerivatives(0): dZi/dt, theDerivatives(1): dTu/dt
    // - according to Eqn 3.2 and Eqn 3.7 in book AITMFM
    int SmokePars_Derivatives(double time, const Vector& SmokeStatus, Vector&
        Derivatives);

    //Basic 4th order Runge-Kutta ODE solver without step size control
    //theSmokeStatus(0): height of the zone free of smoke
    //theSmokeStatus(1): upper hot layer temperature
    int FourthOrder_RungeKutta(const Vector& theDerivatives,
        double theTrialTime_StepSize, double theTrialTime_Initial,
        Vector& theSmokeStatus);

    //Driver for 4th order Runge-Kutta solution of system of ODEs with step
    //size control
```

```
69      int ODESolver_forUpdateSmokePars(const Vector& theMaxError,
70          int theNum_SubSteps, double theOldTimeStep, double theNewTimeStep,
71          Vector& theSmokeStatus);
72
73      //Actually update the smoke status when a new time step launched
74      int UpdateSmokePars(double theTimeMarker, double theNewTimeStep);
75
76      //plume mass flow rate calculation according to different models
77      //Thomas' model
78      double PlumeFlowRate_Thomas(double FirePerimeter, double FreeZoneHeight) const;
79      //Zukoski's model
80      double PlumeFlowRate_Zukoski(double TotalHRR, double FreeZoneHeight) const;
81
82      //get fire current info: fire area, fire perimeter, fire total HRR
83      int CalculateFire_CurrentInfo(double& theFireArea, double& theFirePerimeter,
84          double& theFireTotalHRR, TravellingFireFuel* theCurrentFrontFuel,
85          TravellingFireFuel* theCurrentBackFuel) const;
86
87      //function added for SIFfireAction::displaySelf() use
88      double getTemperature(double time, double &Temperature);
89
90      //for initialising the output file path & data stream
91      int InitializeFilePath(void);
92
93      //get the front edge of the fire spread related travelling fire fuel
94      TravellingFireFuel* UpdateFireFrontEdge(TravellingFireFuel* theCurrentFrontFuel,
95          double theNewTimeStep);
96
97      //get the back edge of the fire related travelling fire fuel
98      TravellingFireFuel* UpdateFireBackEdge(TravellingFireFuel* theCurrentBackFuel,
99          double theNewTimeStep);
100
101     //update the fire parameters with three time points when a new time step
102     //is trying to launch
103     int UpdateFirePars(double theTimeMarker, double theNewTimeStep);
104
105     //for saving the smoke status at each time step into a .dat file
106     int ResultsTxtProducer(double theNewTimeStep);
107
108     virtual void  Print(OPS_Stream&, int = 0) { return; };
109
110 protected:
111
112 private:
113     double getGasTemperature(double time);
114
115     double StartTime;
116     bool FireTravelDirection;       //false: anti-clockwise       true: clockwise
117     const double StoreyH;           //storey height of the fire floor (unit: m)
118     const double FloorArea;         //total floor area (unit: m^2)
119
120     TravellingFireFuel* IgnitionTFfuel; //the ignition travelling fire fuel pointer
121     TravellingFireFuel* FireFrontFuel;  //the travelling fire front edge related
        fuel pointer
122     TravellingFireFuel* FireBackFuel;   //the travelling fire back edge related fuel
        pointer
123
124     Vector Afi;                     //fire area (unit: m^2) at three time points
125     Vector Wfi;                     //fire perimeter (unit: m) at three time points
126     Vector Qf;                      //total HRR (unit: KW) at three time points
127     Vector ThreeTimePnts;           //three time points
128                                     //1st time point: time = TimeMarker
129                                     //3rd time point: time = theNewTimeStep
130                                     //2nd time point: time = (TimeMarker +
                                        theNewTimeStep) / 2
131
132     const int SmokeGenModelTypeTag; //1.Thomas' model   2.Zukoski's model
133     const double conLc;     //total heat loss fraction from plume
```

```
134                             //& compartment bounding surfaces
135       double Mf;            //plume mass flow rate (unit: kg/s)
136       Vector SmokeStatusInfo; //SmokeStatusInfo(0): height of the zone free of smoke
          (unit: m)
137                             //SmokeStatusInfo(1): upper smoke layer temperature
                                (unit: k)
138
139       Vector SmokeAndFireData;//vector to save the smoke current status data
140       bool initialRecording;  //to see if the .dat file has been written in any results
141       bool initializationDone;//to see if the initialization of the output file is done
142       OPS_Stream *theOutputStream;
143   };
144
145   #endif
146
```

ZoneModel_FIRM.h

```
/* ****************************************************************** **
**    OpenSees - Open System for Earthquake Engineering Simulation    **
**          Pacific Earthquake Engineering Research Center             **
**                                                                     **
**                                                                     **
** (C) Copyright 2001, The Regents of the University of California    **
** All Rights Reserved.                                               **
**                                                                     **
** Commercial use of this program without express permission of the   **
** University of California, Berkeley, is strictly prohibited.  See   **
** file 'COPYRIGHT'  in main directory for information on usage and   **
** redistribution,  and for a DISCLAIMER OF ALL WARRANTIES.           **
**                                                                     **
** Fire & Heat Transfer modules developed by:                         **
**   Yaqiang Jiang (y.jiang@ed.ac.uk)                                **
**   Liming Jiang (Liming.Jiang@ed.ac.uk)                            **
**   Xu Dai (x.dai@ed.ac.uk)                                         **
**   Asif Usmani (asif.usmani@ed.ac.uk)                              **
**                                                                     **
** ****************************************************************** */

// $Date: 2017-02-10
// $Source: /usr/local/cvs/OpenSees/SRC/fire/ZoneModel/ZoneModel_FIRM.h

// Written by: Xu Dai (x.dai@ed.ac.uk)
// Description: This file contains the class implementation of FIRM Zone Model,
//              according to the book called 'An Introduction to Mathematical Fire
//              Modeling' book name is referenced in the codes as: AITMFM


#ifndef ZoneModel_FIRM_h
#define ZoneModel_FIRM_h

#include <FireModel.h>
#include <TravellingFireFuel.h>

class ZoneModel_FIRM : public FireModel
{

public:
    ZoneModel_FIRM(int tag, int theSmokeGenModelTypeTag, double theconLc,
        double theconLr, double theStoreyH, double theTotalFloorArea,
        const Vector& theVentDimensions, TravellingFireFuel* theIgnitionTFfuel,
        bool theFireTravelDirection, bool theVent_Conrolled = true,
        bool theEntrainment_Controlled = true, double startTime = 0.0);
    virtual ~ZoneModel_FIRM();

    void applyFluxBC(HeatFluxBC* theFlux, double time);

    //linearly interpolate the fire parameters at a certain time point
    double Interpolate_FirePars(double time, const Vector& TriTimePnts,
        const Vector& FirePars);

    //calculate the derivatives of smoke status with time
    //SmokeStatus(0): height of the zone free of smoke
    //SmokeStatus(1): upper hot layer temperature
    //theDerivatives(0): dZi/dt, theDerivatives(1): dTu/dt
    // - according to Eqn 5.2 and Eqn 5.1 in book AITMFM
    int SmokePars_Derivatives(double time, const Vector& SmokeStatus,
        Vector& Derivatives);

    //calculate the outflow of lower or upper layer gases at any time point
    //SmokeStatus(0): height of the zone free of smoke
    //SmokeStatus(1): upper hot layer temperature
    int CalVentFlows(double time, const Vector& SmokeStatus, double& Interpolated_Qf,
        Vector& VentFlows, int& OxygenStarvedFlag, int& VentFlowRegimeFlag);

    //Basic 4th order Runge-Kutta ODE solver without step size control
```

```
69      //theSmokeStatus(0): height of the zone free of smoke
70      //theSmokeStatus(1): upper hot layer temperature
71      int FourthOrder_RungeKutta(const Vector& theDerivatives, double
72          theTrialTime_StepSize, double theTrialTime_Initial, Vector& theSmokeStatus);
73
74      //Driver for 4th order Runge-Kutta solution of system of ODEs
75      //with step size control
76      int ODESolver_forUpdateSmokePars(const Vector& theMaxError, int theNum_SubSteps,
77          double theOldTimeStep, double theNewTimeStep, Vector& theSmokeStatus);
78
79      //Actually update the smoke status when a new time step launched
80      int UpdateSmokePars(double theTimeMarker, double theNewTimeStep);
81
82      //plume mass flow rate calculation according to different models
83      //Thomas' model
84      double PlumeFlowRate_Thomas(double FirePerimeter, double FreeZoneHeight) const;
85      //Zukoski's model
86      double PlumeFlowRate_Zukoski(double TotalHRR, double FreeZoneHeight) const;
87
88      //get fire current info: fire area, fire perimeter, fire total HRR
89      int CalculateFire_CurrentInfo(double& theFireArea, double& theFirePerimeter,
90          double& theFireTotalHRR, TravellingFireFuel* theCurrentFrontFuel,
91          TravellingFireFuel* theCurrentBackFuel) const;
92
93      //functions added for SIFfireAction::displaySelf() use
94      double getTemperature(double time, double &Temperature);
95      const Vector& getCurrentSmokeInfo(void) const;
96
97      //for initialising the output file path & data stream
98      int InitializeFilePath(void);
99
100     //get the front edge of the fire spread related travelling fire fuel
101     TravellingFireFuel* UpdateFireFrontEdge(TravellingFireFuel* theCurrentFrontFuel,
102     double theNewTimeStep);
103
104     //get the back edge of the fire related travelling fire fuel
105     TravellingFireFuel* UpdateFireBackEdge(TravellingFireFuel* theCurrentBackFuel,
106     double theNewTimeStep);
107
108     //update the fire parameters with three time points when
109     //a new time step is trying to launch
110     int UpdateFirePars(double theTimeMarker, double theNewTimeStep);
111
112     //for saving the smoke status at each time step into a .dat file
113     int ResultsTxtProducer(double theNewTimeStep);
114
115     virtual void  Print(OPS_Stream&, int = 0) { return; };
116
117 protected:
118
119 private:
120     double getGasTemperature(double time);
121
122     double StartTime;
123     bool FireTravelDirection;       //false: anti-clockwise       true: clockwise
124     const double StoreyH;           //storey height of the fire floor (unit: m)
125     const double FloorArea;         //total floor area (unit: m^2)
126
127     TravellingFireFuel* IgnitionTFfuel; //the ignition travelling fire fuel pointer
128     TravellingFireFuel* FireFrontFuel;  //the travelling fire front edge related
        fuel pointer
129     TravellingFireFuel* FireBackFuel;   //the travelling fire back edge related fuel
        pointer
130
131     Vector Afi;             //fire area (unit: m^2) at three time points
132     Vector Wfi;             //fire perimeter (unit: m) at three time points
133     Vector Qf;              //total HRR (unit: KW) at three time points
134     Vector ThreeTimePnts;   //three time points
```

```
135                             //1st time point: time = TimeMarker
136                             //3rd time point: time = theNewTimeStep
137                             //2nd time point: time = (TimeMarker + theNewTimeStep) / 2
138
139     const int SmokeGenModelTypeTag; //1.Thomas' model   2.Zukoski's model
140     const double conLc;             //total heat loss fraction from plume
141                                     //& compartment bounding surfaces
142     const double conLr;             //radiative heat loss fraction
143     const Vector ventDimensions;    //vector for storing the ventilation dimensions
144                                     //(unit: m)
145                                     //ventDimensions(0): total vent widths
146                                     //ventDimensions(1): average sill height
147                                     //ventDimensions(2): average soffit height
148
149     const bool Vent_Controlled;         //if considering the ventilation-controlled
        burning in FIRM
150     const bool Entrainment_Controlled;  //if considering the entrainment-controlled
        burning in FIRM
151
152     double maxVentFlow;         //maximum inflow of ambient air (kg/s)
153     double Mf;                  //plume mass flow rate (unit: kg/s)
154     Vector SmokeStatusInfo;     //SmokeStatusInfo(0): height of the zone free of
        smoke
155                                 //(unit: m)
156                                 //SmokeStatusInfo(1): upper smoke layer temperature
157                                 //(unit: k)
158
159     Vector SmokeAndFireData;//vector to save the smoke current status data
160     bool initialRecording;  //to see if the .dat file has been written in any results
161     bool initializationDone;//to see if the initialization of the output file is done
162     OPS_Stream *theOutputStream;
163     int VentFlowRegimeFlag; //1: regime 1    2: regime 2    3: regime 3    4: regime
        4
164                             //23: transition between regime 2 and regime 3
165                             //5: regime that smoke layer interface dropped below sill
166                             //all regimes definition can be found in AITMFM -
                                chapter 4
167
168     int OxygenStarvedFlag;  //0:  not under oxygen starvation
169                             //1:  under oxygen starvation
170                             //-1: entrainment-controlled burning is not considered
171
172     Vector VentFlowsInfo;   //VentFlowsInfo(0):vent flow of ambient air
173                             //entering the compartment (unit: kg/s)
174                             //VentFlowsInfo(1):vent flow of lower layer gas
175                             //leaving the compartment (unit: kg/s)
176                             //VentFlowsInfo(2):vent flow of upper layer gas
177                             //leaving the compartment (unit: kg/s)
178
179     double Qf_Info;         //total HRR (unit: KW) which is actually used in
180                             //the calculation - when considering
181                             //entrainment-controlled burning
182 };
183
184 #endif
185
```

B.2 Header files for the ETFM framework in C++

This section contains the header (.h) files of the related classes for implementing the ETFM framework in SIFBuilder. These classes are *ETFM_MovingHasemi* class, *ETFM_ZoneModel_FIRM* class, *TravellingFireFuel* class, and *TravellingFireFuel_Iter* class. The implementation of the ETFM framework using these four classes is discussed in Chapter 6, section 6.4. The theoretical background of the ETFM framework itself can be found in Chapter 4.

ETFM_MovingHasemi.h

```
/* ****************************************************************** **
**    OpenSees - Open System for Earthquake Engineering Simulation    **
**          Pacific Earthquake Engineering Research Center             **
**                                                                     **
**                                                                     **
** (C) Copyright 2001, The Regents of the University of California    **
** All Rights Reserved.                                                **
**                                                                     **
** Commercial use of this program without express permission of the   **
** University of California, Berkeley, is strictly prohibited.  See   **
** file 'COPYRIGHT'  in main directory for information on usage and   **
** redistribution,  and for a DISCLAIMER OF ALL WARRANTIES.           **
**                                                                     **
** Fire & Heat Transfer modules developed by:                          **
**   Yaqiang Jiang (y.jiang@ed.ac.uk)                                 **
**   Liming Jiang (Liming.Jiang@ed.ac.uk)                             **
**   Xu Dai (x.dai@ed.ac.uk)                                          **
**   Asif Usmani (asif.usmani@ed.ac.uk)                               **
**                                                                     **
** ****************************************************************** */

// $Date: 2016-10-11
// $Source: /usr/local/cvs/OpenSees/SRC/fire/TravellingFire/ETFM_MovingHasemi.h

// Written by: Xu Dai (x.dai@ed.ac.uk)
// Description: This file contains the class implementation of
//              the moving Hasemi localised fire model in ETFM framework

#ifndef ETFM_MovingHasemi_h
#define ETFM_MovingHasemi_h

#include <FireModel.h>
#include <TravellingFireFuel.h>

class HeatTransferNode;

class ETFM_MovingHasemi : public FireModel
{
public:
    ETFM_MovingHasemi(int tag, int theMemberTypeTag, double theStoreyH,
        const Matrix& theIslandOutline, TravellingFireFuel* theIgnitionTFfuel,
        bool theFireTravelDirection, double startTime = 0.0);

    virtual ~ETFM_MovingHasemi();

    void applyFluxBC(HeatFluxBC* theFlux, double time);
    double getFlux(HeatTransferNode* the_node, double time);

    //reset the FireFrontFuel and FireBackFuel pointers back to initial status,
    //i.e. equal to the IgnitionTFfuel pointer
    //function only called in SIFHTforMember::applyFire(), for the travelling fire
    //model pointer which is not deleted but recycled used
    int ResetFrontandBackPtr(void);

protected:


private:

    //update the fire to fit Hasemi localized fire model parameters
    int UpdateHasemiPars(double theNewTimeStep);

    //get the front edge of the fire spread related travelling fire fuel
    TravellingFireFuel* UpdateFireFrontEdge(TravellingFireFuel* theCurrentFrontFuel,
    double theNewTimeStep);

    //get the back edge of the fire related travelling fire fuel
    TravellingFireFuel* UpdateFireBackEdge(TravellingFireFuel* theCurrentBackFuel,
```

```
69          double theNewTimeStep);
70
71      //get fire current info: fire centre coordinates, fire area, fire total HRR
72      int CalculateFire_CurrentInfo(Vector& theFireCentreCrds, double& theFireArea,
73          double& theFireTotalHRR, TravellingFireFuel* theCurrentFrontFuel,
74          TravellingFireFuel* theCurrentBackFuel) const;
75
76      //judge if the moving Hasemi centre 'sees' the heat transfer node due to
77      //the fire resistant island
78      bool ShadeView(HeatTransferNode* the_node) const;
79
80      int MemberTypeTag;          //1.SIFXBeam 2.SIFYBeam 3.SIFColumn
81                                  //10.SIFSlab 21.SecXBeam
82      double StartTime;
83      double StoreyH;             //storey height of the fire floor (unit: m)
84      bool FireTravelDirection;   //false: anti-clockwise       true: clockwise
85
86      TravellingFireFuel* IgnitionTFfuel; //the ignition travelling fire fuel pointer
87      TravellingFireFuel* FireFrontFuel;  //the travelling fire front edge
88                                          //related fuel pointer
89      TravellingFireFuel* FireBackFuel;   //the travelling fire back edge
90                                          //related fuel pointer
91
92      double x1, x2, x3;  //current fire origin location in HT coordinate system
93      double d;           //current fire diameter (unit: m)
94      double q;           //current fire HRR (unit: W)
95      int centerLine;     //center line tag of the localized hasemi's fire:
96                          //corresponding to the current heat transfer analysis
97                          //member under the HT coordinate system
98      const Matrix IslandOutline; //fire-resistant island four corner points
99                                  //to bring this Matrix is only for checking
100                                 //the ShadeViews
101
102     bool Ratio_85;  //for considering different MemberTypes (i.e. beams, or slabs)
103                     //with 0.85 ratio, details can be found at the "Annex C of
104                     //prEN1991-1-2" - Page 7: last paragraph
105 };
106
107 #endif
108
109
```

ETFM_ZoneModel_FIRM.h

```
/* ****************************************************************** **
**    OpenSees - Open System for Earthquake Engineering Simulation    **
**          Pacific Earthquake Engineering Research Center             **
**                                                                     **
**                                                                     **
** (C) Copyright 2001, The Regents of the University of California    **
** All Rights Reserved.                                                **
**                                                                     **
** Commercial use of this program without express permission of the   **
** University of California, Berkeley, is strictly prohibited.  See   **
** file 'COPYRIGHT'  in main directory for information on usage and   **
** redistribution,  and for a DISCLAIMER OF ALL WARRANTIES.           **
**                                                                     **
** Fire & Heat Transfer modules developed by:                          **
**   Yaqiang Jiang (y.jiang@ed.ac.uk)                                 **
**   Liming Jiang (Liming.Jiang@ed.ac.uk)                             **
**   Xu Dai (x.dai@ed.ac.uk)                                          **
**   Asif Usmani (asif.usmani@ed.ac.uk)                               **
**                                                                     **
** ****************************************************************** */

// $Date: 2017-02-10
// $Source: /usr/local/cvs/OpenSees/SRC/fire/TravellingFire/ETFM_ZoneModel_FIRM.h

// Written by: Xu Dai (x.dai@ed.ac.uk)
// Description: This file contains the class implementation of FIRM Zone Model,
//              according to the book named 'An Introduction to Mathematical Fire
//              Modeling' book name is referenced in the codes as: AITMFM


#ifndef ETFM_ZoneModel_FIRM_h
#define ETFM_ZoneModel_FIRM_h

#include <FireModel.h>
#include <TravellingFireFuel.h>

class ETFM_ZoneModel_FIRM : public FireModel
{

public:

    ETFM_ZoneModel_FIRM(int tag, int theSmokeGenModelTypeTag, double theconLc,
        double theconLr, double theStoreyH, double theTotalFloorArea,
        const Vector& theVentDimensions, TravellingFireFuel* theIgnitionTFfuel,
        bool theFireTravelDirection, bool theVent_Conrolled = true,
        bool theEntrainment_Controlled = true, double startTime = 0.0);

    virtual ~ETFM_ZoneModel_FIRM();

    void applyFluxBC(HeatFluxBC* theFlux, double time);

    //functions added for SIFfireAction::displaySelf() use
    double getTemperature(double time, double &Temperature);
    const Vector& getCurrentSmokeInfo(void) const;

    virtual void  Print(OPS_Stream&, int = 0) { return; };

protected:

private:

    double getGasTemperature(double time);

    //plume mass flow rate calculation according to different models
    //Thomas' model
    double PlumeFlowRate_Thomas(double FirePerimeter, double FreeZoneHeight) const;
    //Zukoski's model
    double PlumeFlowRate_Zukoski(double TotalHRR, double FreeZoneHeight) const;
```

```
69
70      //update the fire parameters with three time points when a new time step
71      //is trying to launch
72      int UpdateFirePars(double theTimeMarker, double theNewTimeStep);
73
74      //get the front edge of the fire spread related travelling fire fuel
75      TravellingFireFuel* UpdateFireFrontEdge(TravellingFireFuel* theCurrentFrontFuel,
76          double theNewTimeStep);
77      //get the back edge of the fire related travelling fire fuel
78      TravellingFireFuel* UpdateFireBackEdge(TravellingFireFuel* theCurrentBackFuel,
79          double theNewTimeStep);
80
81      //get fire current info: fire area, fire perimeter, fire total HRR
82      int CalculateFire_CurrentInfo(double& theFireArea, double& theFirePerimeter,
83          double& theFireTotalHRR, TravellingFireFuel* theCurrentFrontFuel,
84          TravellingFireFuel* theCurrentBackFuel) const;
85
86      //Actually update the smoke status when a new time step launched
87      int UpdateSmokePars(double theTimeMarker, double theNewTimeStep);
88
89      //linearly interpolate the fire parameters at a certain time point
90      double Interpolate_FirePars(double time, const Vector& TriTimePnts,
91          const Vector& FirePars);
92
93      //calculate the derivatives of smoke status with time
94      //SmokeStatus(0): height of the zone free of smoke
95      //SmokeStatus(1): upper hot layer temperature
96      //theDerivatives(0): dZi/dt, theDerivatives(1): dTu/dt -
97      //according to Eqn 5.2 and Eqn 5.1 in book AITMFM
98      int SmokePars_Derivatives(double time, const Vector& SmokeStatus,
99          Vector& Derivatives);
100
101     //calculate the outflow of lower or upper layer gases at any time point
102     //SmokeStatus(0): height of the zone free of smoke
103     //SmokeStatus(1): upper hot layer temperature
104     int CalVentFlows(double time, const Vector& SmokeStatus, double& Interpolated_Qf,
105         Vector& VentFlows, int& OxygenStarvedFlag, int& VentFlowRegimeFlag);
106
107     //Basic 4th order Runge-Kutta ODE solver without step size control
108     //theSmokeStatus(0): height of the zone free of smoke
109     //theSmokeStatus(1): upper hot layer temperature
110     int FourthOrder_RungeKutta(const Vector& theDerivatives,
111         double theTrialTime_StepSize, double theTrialTime_Initial,
112         Vector& theSmokeStatus);
113
114     //Driver for 4th order Runge-Kutta solution of system of ODEs with
115     //step size control
116     int ODESolver_forUpdateSmokePars(const Vector& theMaxError, int theNum_SubSteps,
117         double theOldTimeStep, double theNewTimeStep, Vector& theSmokeStatus);
118
119     //for initialising the output file path & data stream
120     int InitializeFilePath(void);
121     //for saving the smoke status at each time step into a .dat file
122     int ResultsTxtProducer(double theNewTimeStep);
123
124     double StartTime;
125     bool FireTravelDirection;//false: anti-clockwise       true: clockwise
126     const double StoreyH;    //storey height of the fire floor (unit: m)
127     const double FloorArea;  //total floor area (unit: m^2)
128
129     TravellingFireFuel* IgnitionTFfuel; //the ignition travelling fire fuel pointer
130     TravellingFireFuel* FireFrontFuel;  //the travelling fire front edge related
131                                         //fuel pointer
132     TravellingFireFuel* FireBackFuel;   //the travelling fire back edge related
133                                         //fuel pointer
134
135     Vector Afi;          //fire area (unit: m^2) at three time points
136     Vector Wfi;          //fire perimeter (unit: m) at three time points
```

```
137       Vector Qf;           //total HRR (unit: KW) at three time points
138       Vector ThreeTimePnts;//three time points
139                            //1st time point: time = TimeMarker
140                            //3rd time point: time = theNewTimeStep
141                            //2nd time point: time = (TimeMarker + theNewTimeStep) / 2
142
143       const int SmokeGenModelTypeTag; //1.Thomas' model   2.Zukoski's model
144       const double conLc;             //total heat loss fraction from plume
145                                       //& compartment bounding surfaces
146       const double conLr;             //radiative heat loss fraction
147       const Vector ventDimensions;    //vector for storing the ventilation dimensions
148                                       //(unit: m)
149                                       //ventDimensions(0): total vent widths
150                                       //ventDimensions(1): average sill height
151                                       //ventDimensions(2): average soffit height
152
153       const bool Vent_Controlled;       //if considering the ventilation-controlled
154                                         //burning in FIRM
155       const bool Entrainment_Controlled;//if considering the entrainment-controlled
156                                         //burning in FIRM
157
158       double maxVentFlow;     //maximum inflow of ambient air (kg/s)
159       double Mf;              //plume mass flow rate (unit: kg/s)
160       Vector SmokeStatusInfo; //SmokeStatusInfo(0): height of the zone free of smoke
161                               //(unit: m)
162                               //SmokeStatusInfo(1): upper smoke layer temperature
163                               //(unit: k)
164
165       Vector SmokeAndFireData;    //vector to save the smoke current status data
166       bool initialRecording;      //to see if the .dat file has been written
167                                   //in any results
168       bool initializationDone;    //to see if the initialization of the output
169                                   //file is done
170       OPS_Stream *theOutputStream;
171       int VentFlowRegimeFlag; //1: regime 1    2: regime 2    3: regime 3
172                               //4: regime 4
173                               //23: transition between regime 2 and regime 3
174                               //5: regime that smoke layer interface dropped below sill
175                               //all regimes definition can be found in AITMFM -
                                  chapter 4
176
177       int OxygenStarvedFlag;      //0:  not under oxygen starvation
178                                   //1:  under oxygen starvation
179                                   //-1: entrainment-controlled burning is not considered
180
181       Vector VentFlowsInfo;   //VentFlowsInfo(0):vent flow of ambient air entering
182                               //the compartment (unit: kg/s)
183                               //VentFlowsInfo(1):vent flow of lower layer gas leaving
184                               //the compartment (unit: kg/s)
185                               //VentFlowsInfo(2):vent flow of upper layer gas leaving
186                               //the compartment (unit: kg/s)
187
188       double Qf_Info;         //total HRR (unit: KW) which is actually used in the
189                               //calculation - when considering
190                               //entrainment-controlled burning
191   };
192
193   #endif
194
```

TravellingFireFuel.h

```
/* ****************************************************************** **
**    OpenSees - Open System for Earthquake Engineering Simulation    **
**          Pacific Earthquake Engineering Research Center            **
**                                                                    **
**                                                                    **
** (C) Copyright 2001, The Regents of the University of California    **
** All Rights Reserved.                                               **
**                                                                    **
** Commercial use of this program without express permission of the   **
** University of California, Berkeley, is strictly prohibited.  See   **
** file 'COPYRIGHT'  in main directory for information on usage and   **
** redistribution,  and for a DISCLAIMER OF ALL WARRANTIES.           **
**                                                                    **
** Fire & Heat Transfer modules developed by:                         **
**   Yaqiang Jiang (y.jiang@ed.ac.uk)                                **
**   Liming Jiang (Liming.Jiang@ed.ac.uk)                            **
**   Xu Dai (x.dai@ed.ac.uk)                                         **
**   Asif Usmani (asif.usmani@ed.ac.uk)                              **
**                                                                    **
** ****************************************************************** */

// $Date: 2016-10-04
// $Source: /usr/local/cvs/OpenSees/SRC/fire/TravellingFire/TravellingFireFuel.h

// Written by: Xu Dai (x.dai@ed.ac.uk)
// Description: This file contains the 'fuel' class, which is developed for
// implementing travelling fire models. Each fuel object has its own identities,
// such as HRR/m2, fire load density, burning status, etc.

#ifndef TravellingFireFuel_h
#define TravellingFireFuel_h

#include <Matrix.h>
#include <Vector.h>
#include <TaggedObject.h>
#include <MovableObject.h>
#include <Renderer.h>

class TravellingFireFuel : public TaggedObject
{

public:
    TravellingFireFuel(int tag, const Matrix& Corner_Nodes,
        const Vector& Centre_Nodes_OnTrajectory, double Spread_Rate,
        double Fuel_Load_Density, double HRR_perArea, bool Fire_Status = false);

    ~TravellingFireFuel();

    //reset the fuel to initial status for the next time SIFBuilder - HT analysis
    void ResetStatus(void);

    void SetFuelOnFire(void);
    void SetFuelOffFire(void);
    bool getFuelFireStatus(void) const;     //to know if the fuel is on fire or not

    //set the existing two neighbour travelling fire fuels, linked with 'this' fuel
    int SetNeighbourFuels(TravellingFireFuel* NeighbourFuel_AntiClk,
        TravellingFireFuel* NeighbourFuel_Clk);

    //return corresponding neighbour travelling fire fuel pointer
    TravellingFireFuel* getNeighbourFuelPtr(const TravellingFireFuel* thisFuel,
        bool FireTravelDirection) const;

    //set when the fuel would be on fire and when would be off fire
    int SetTimeFeatures(double T_onFire);

    const Vector& getFuelCentre_OnTrajec(void) const;
    int setFuelLength_OnTraject(double theFuelLength);
```

```
69      double getFuelLength_OnTraject(void) const;
70
71      //calculate fuel perimeter
72      double CalculateFuelPerimeter(void);
73
74      double getFuel_Area(void) const;
75      double getFuel_Perimeter(void) const;
76      double getFuel_HRR(void) const; //(unit: KW)
77      double getFuel_SpreadRate(void) const;
78      double getFuel_OnFireTime(void) const;
79      double getFuel_OffFireTime(void) const;
80
81      //display TravellingFireFuel if it's on fire
82      int displaySelf(Renderer &theRenderer);
83
84      virtual void  Print(OPS_Stream&, int = 0) { return; };
85
86  private:
87
88      double CalculateFuelArea(void); //calculate fuel area
89
90      int GetFuel_OnFire_Color(void);
91
92      const Matrix CornerNodes;   // stores the coordinates of the corner nodes
93                                  // for one fuel object (unit: m)
94      const Vector CentreOnTrajec;// stores the coordinates of the fuel centre point
95                                  // on the travelling fire trajectory (unit: m)
96
97      double FuelLength_OnTraject;// the trajectory length covered by one
98                                  // isolated fuel (unit: m)
99
100     double SpreadRate;          // fire spread rate on the fuel (unit: mm/s)
101     double FuelLoadDensity;     // fuel load density (unit: MJ/m2)
102     double HRRperArea;          // heat release rate per area (unit: KW/m2)
103
104     double FuelArea;            // fuel area (unit: m2)
105     double FuelPerimeter;       // fuel perimeter (unit: m)
106
107     bool FireStatus;            // false: not on fire
108                                 // true: currently on fire
109
110     // the fuel which is adjacent to 'this' one, in the AntiClockWise direction
111     TravellingFireFuel* NeighbourFuel_AntiClockWise;
112     // the fuel which is adjacent to 'this' one, in the ClockWise direction
113     TravellingFireFuel* NeighbourFuel_ClockWise;
114
115     double TimeOnFire;      //on what time the fuel would be on fire    (unit: s)
116     double TimeOffFire;     //on what time the fuel would be off fire   (unit: s)
117
118     //polygon RGB colour
119     Matrix *FuelPolygonRGB;
120 };
121
122 #endif
123
124
```

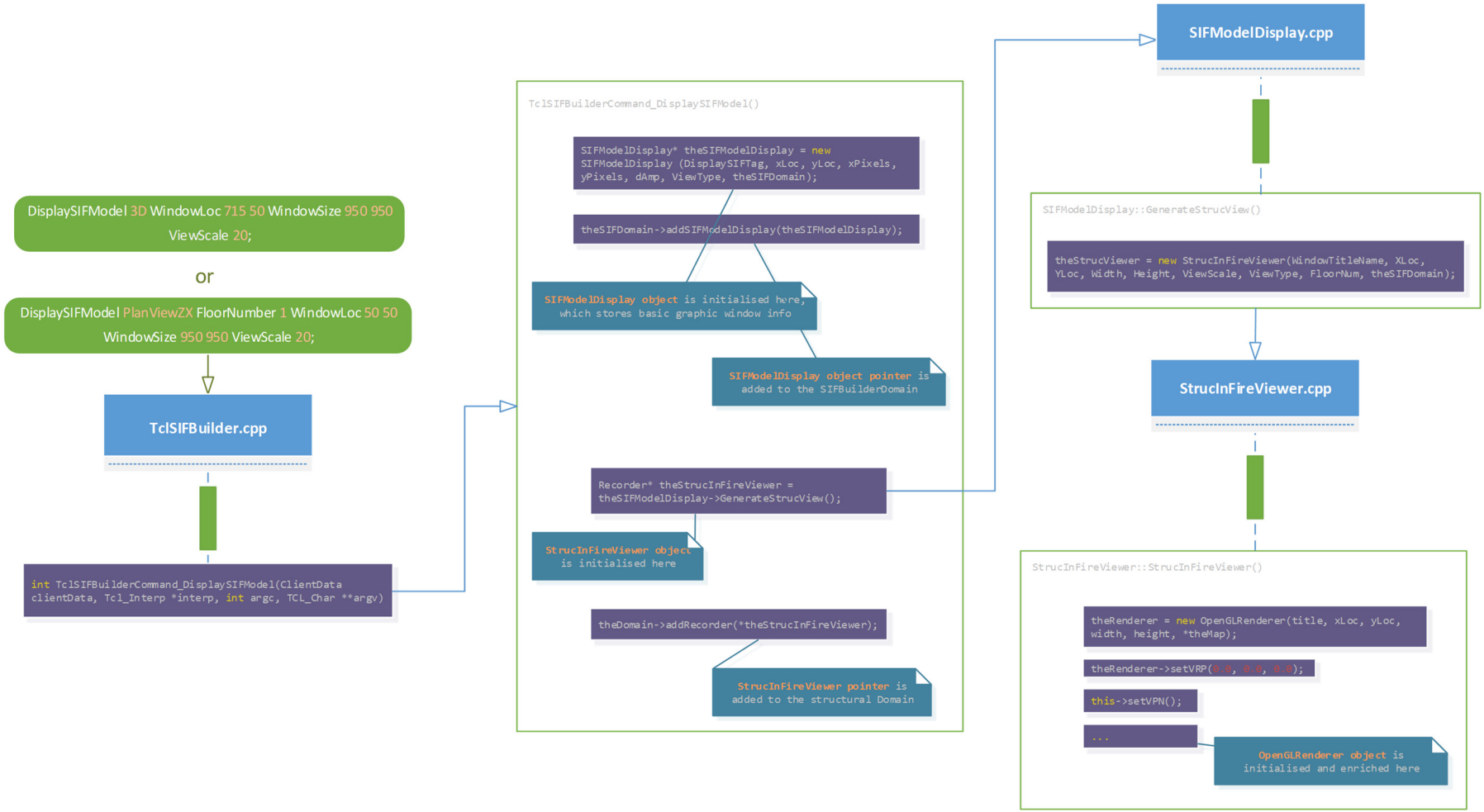
TravellingFireFuel_Iter.h

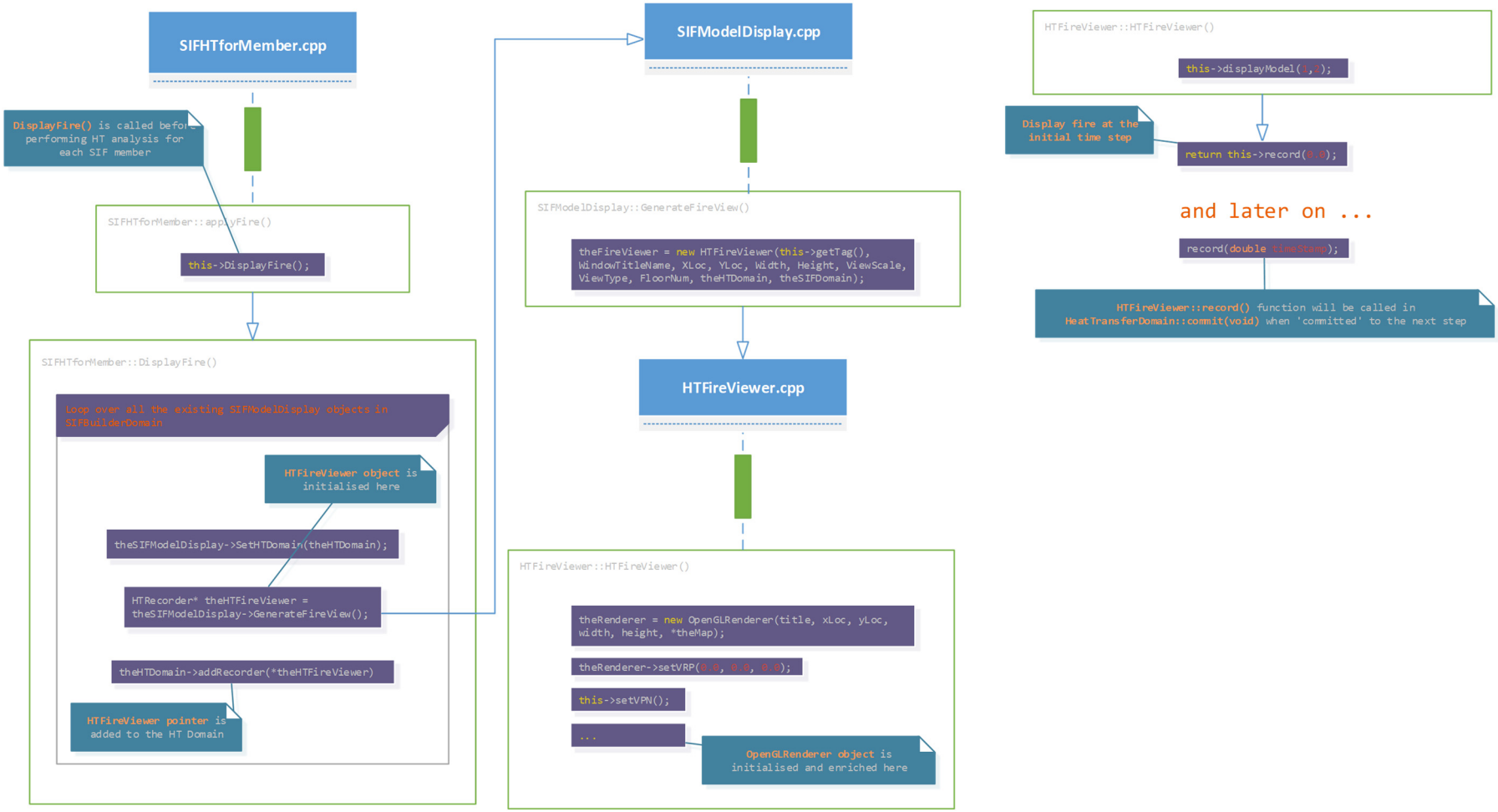
```
/* ****************************************************************** **
**    OpenSees - Open System for Earthquake Engineering Simulation    **
**          Pacific Earthquake Engineering Research Center            **
**                                                                    **
**                                                                    **
** (C) Copyright 2001, The Regents of the University of California    **
** All Rights Reserved.                                               **
**                                                                    **
** Commercial use of this program without express permission of the   **
** University of California, Berkeley, is strictly prohibited.  See   **
** file 'COPYRIGHT'  in main directory for information on usage and   **
** redistribution,  and for a DISCLAIMER OF ALL WARRANTIES.           **
**                                                                    **
** Fire & Heat Transfer modules developed by:                         **
**   Yaqiang Jiang (y.jiang@ed.ac.uk)                                 **
**   Liming Jiang (Liming.Jiang@ed.ac.uk)                             **
**   Xu Dai (x.dai@ed.ac.uk)                                          **
**   Asif Usmani (asif.usmani@ed.ac.uk)                               **
**                                                                    **
** ****************************************************************** */

// $Date: 2016-10-05
// $Source: /usr/local/cvs/OpenSees/SRC/fire/TravellingFire/TravellingFireFuel_Iter.h

// Written by: Xu Dai (x.dai@ed.ac.uk)
// Description: This file contains the class definition for TravellingFireFuel_Iter
// An TravellingFireFuel_Iter is an iter for returning the TravellingFireFuels


#ifndef TravellingFireFuel_Iter_h
#define TravellingFireFuel_Iter_h

class TravellingFireFuel;
class TaggedObjectStorage;
class TaggedObjectIter;

class TravellingFireFuel_Iter
{
public:
    TravellingFireFuel_Iter(TaggedObjectStorage* theStorage);
    virtual ~TravellingFireFuel_Iter();
    virtual TravellingFireFuel* operator()(void);
    virtual void reset(void);

private:
    TaggedObjectIter& myIter;
};

#endif

```

B.3 Header files for the visualisation of fire models in C++

This section contains the header (.h) files of the related classes for visualising the fire model analysis in SIFBuilder. During the fire and heat transfer analysis using SIFBuilder, the fire status would be rendered onto the screen monitor, for the users to check its fire modelling status transiently. The work of rendering fire modelling status in SIFBuilder is developed through utilising an existing class in the original version of OpenSees. This class name is *OpenGLRenderer*, which is a class for displaying the Open Graphics Library (OpenGL) by rendering.

Four additional classes are added for the visualisation of fire models in SIFBuilder: *SIFModelDisplay* class, *SIFModelDisplayIter* class, *HTFireViewer* class, and *StrucInFireViewer* class. The relationship of these four classes, and how these four classes interact with each other in SIFBuilder, are schematically presented with a two-page code flowchart (see Figure B.1 and Figure B.2). Moreover, this two-page code flowchart is more useful for interested developers, who can continue developing their own fire models in SIFBuilder by adapting the above-mentioned rendering classes, to visualise their proposed fire modelling status during the code development debugging stage.


DisplaySIFModel 3D WindowLoc 715 50 WindowSize 950 950 ViewScale 20;
or
DisplaySIFModel PlanViewZX FloorNumber 1 WindowLoc 50 50 WindowSize 950 950 ViewScale 20;
TclSIFBuilder.cpp
int TclSIFBuilderCommand_DisplaySIFModel(ClientData clientData, Tcl_Interp *interp, int argc, TCL_Char **argv)
TclSIFBuilderCommand_DisplaySIFModel()
SIFModelDisplay* theSIFModelDisplay = new SIFModelDisplay (DisplaySIFTag, xLoc, yLoc, xPixels, yPixels, dAmp, ViewType, theSIFDomain);
theSIFDomain->addSIFModelDisplay(theSIFModelDisplay);
SIFModelDisplay object is initialised here, which stores basic graphic window info
SIFModelDisplay object pointer is added to the SIFBuilderDomain
Recorder* theStrucInFireViewer = theSIFModelDisplay->GenerateStrucView();
StrucInFireViewer object is initialised here
theDomain->addRecorder(*theStrucInFireViewer);
StrucInFireViewer pointer is added to the structural Domain
SIFModelDisplay.cpp
SIFModelDisplay::GenerateStrucView()
theStrucViewer = new StrucInFireViewer(WindowTitleName, XLoc, YLoc, Width, Height, ViewScale, ViewType, FloorNum, theSIFDomain);
StrucInFireViewer.cpp
StrucInFireViewer::StrucInFireViewer()
theRenderer = new OpenGLRenderer(title, xLoc, yLoc, width, height, *theMap);
theRenderer->setVRP(0.0, 0.0, 0.0);
this->setVPN();
...
OpenGLRenderer object is initialised and enriched here


SIFHTforMember.cpp
DisplayFire() is called before performing HT analysis for each SIF member
SIFHTforMember::applyFire()
this->DisplayFire();
SIFHTforMember::DisplayFire()
Loop over all the existing SIFModelDisplay objects in SIFBuilderDomain
HTFireViewer object is initialised here
theSIFModelDisplay->SetHTDomain(theHTDomain);
HTRecorder* theHTFireViewer = theSIFModelDisplay->GenerateFireView();
theHTDomain->addRecorder(*theHTFireViewer)
HTFireViewer pointer is added to the HT Domain
SIFModelDisplay.cpp
SIFModelDisplay::GenerateFireView()
theFireViewer = new HTFireViewer(this->getTag(), WindowTitleName, XLoc, YLoc, Width, Height, ViewScale, ViewType, FloorNum, theHTDomain, theSIFDomain);
HTFireViewer.cpp
HTFireViewer::HTFireViewer()
theRenderer = new OpenGLRenderer(title, xLoc, yLoc, width, height, *theMap);
theRenderer->setVRP(0.0, 0.0, 0.0);
this->setVPN();
...
OpenGLRenderer object is initialised and enriched here
HTFireViewer::HTFireViewer()
this->displayModel(1,0);
Display fire at the initial time step
return this->record(0.0);
and later on ...
record(double timeStamp);
HTFireViewer::record() function will be called in HeatTransferDomain::commit(void) when 'committed' to the next step

SIFModelDisplay.h

```
/* ****************************************************************** **
**    OpenSees - Open System for Earthquake Engineering Simulation    **
**          Pacific Earthquake Engineering Research Center            **
**                                                                    **
**                                                                    **
** (C) Copyright 1999, The Regents of the University of California    **
** All Rights Reserved.                                               **
**                                                                    **
** Commercial use of this program without express permission of the   **
** University of California, Berkeley, is strictly prohibited.  See   **
** file 'COPYRIGHT'  in main directory for information on usage and   **
** redistribution,  and for a DISCLAIMER OF ALL WARRANTIES.           **
**                                                                    **
** Developed by:                                                      **
**   Frank McKenna (fmckenna@ce.berkeley.edu)                         **
**   Gregory L. Fenves (fenves@ce.berkeley.edu)                       **
**   Filip C. Filippou (filippou@ce.berkeley.edu)                     **
**                                                                    **
** ****************************************************************** */

/* ****************************************************************** **
**                          SIFBUILDER PROJECT                        **
**                                                                    **
**  This project is aiming to provide an interface to define models   **
**  for simulating structural behaviours under fire action.           **
**  Developed by:                                                     **
**                                                                    **
**   Liming Jiang (liming.jiang@ed.ac.uk)                             **
**   Xu Dai(X.Dai@ed.ac.uk)                                           **
**   Praven Kamath(Praveen.Kamath@ed.ac.uk)                           **
**   Asif Usmani(asif.usmani@ed.ac.uk)                                **
**                                                                    **
** ******************************************************************/

// $Date: 2015-12-15
// $Source: /usr/local/cvs/OpenSees/SRC/SIFBuilder/SIFModelDisplay/SIFModelDisplay.h

// Written by: Xu Dai (x.dai@ed.ac.uk)
// Description: This file contains the class implementation for preparing
//              SIFModel display - fire & structural response

#ifndef SIFModelDisplay_h
#define SIFModelDisplay_h

#include <string.h>
#include <iostream>
#include <TaggedObject.h>
#include <OPS_Globals.h>
#include <HTFireViewer.h>
#include <StrucInFireViewer.h>

using namespace std;
class SIFBuilderDomain;

class SIFModelDisplay : public TaggedObject
{
public:
    SIFModelDisplay(int tag, int XLoc, int YLoc, int Width, int Height, double
    ViewScale,
        int ViewType, SIFBuilderDomain* theSifDomain);
    ~SIFModelDisplay();

    int SetHTDomain(HeatTransferDomain* theHTdomain);
    int SetFloorNum(int theFloorNumber);

    // creating HTfireViewer objects for HT recorders
    HTRecorder* GenerateFireView();

```

```
68      // creating StrucInFireViewer objects for structural recorders
69      Recorder* GenerateStrucView();
70
71      virtual void  Print(OPS_Stream&, int = 0) { return; };
72
73  private:
74
75      // determining graphical window title according to different modelling stages
76      // ModellingStage == 1: fire & heat transfer modelling
77      // ModellingStage == 2: structural modelling
78      const string& DetermineWindowTitle(int ModellingStage) const;
79
80      int XLoc;           //horizontal location in pixels of graphical window
81      int YLoc;           //vertical location in pixels of graphical window (0,0=upper
        left-most corner)
82      int Width;          //xPixels: width of graphical window in pixels
83      int Height;         //yPixels: height of graphical window in pixels
84      double ViewScale;   //scaling factor for viewing deformed shape
85      int ViewType;       //1:ElevationViewZY 2:ElevationViewXY 3:PlanViewZX 4:3D
86      int FloorNum;       //floor number for displaying, 0: ground floor, 1: first
        floor ...
87      HeatTransferDomain* theHTDomain;
88      SIFBuilderDomain* theSIFDomain;
89  };
90
91  #endif
```

SIFModelDisplayIter.h

```
/* ****************************************************************** **
**    OpenSees - Open System for Earthquake Engineering Simulation    **
**          Pacific Earthquake Engineering Research Center            **
**                                                                    **
**                                                                    **
** (C) Copyright 1999, The Regents of the University of California    **
** All Rights Reserved.                                               **
**                                                                    **
** Commercial use of this program without express permission of the   **
** University of California, Berkeley, is strictly prohibited.  See   **
** file 'COPYRIGHT'  in main directory for information on usage and   **
** redistribution,  and for a DISCLAIMER OF ALL WARRANTIES.           **
**                                                                    **
** Developed by:                                                      **
**   Frank McKenna (fmckenna@ce.berkeley.edu)                         **
**   Gregory L. Fenves (fenves@ce.berkeley.edu)                       **
**   Filip C. Filippou (filippou@ce.berkeley.edu)                     **
**                                                                    **
** ****************************************************************** */

/* ****************************************************************** **
**                          SIFBUILDER PROJECT                        **
**                                                                    **
**  This project is aiming to provide an interface to define models   **
**  for simulating structural behaviours under fire action.           **
**  Developed by:                                                     **
**                                                                    **
**   Liming Jiang (liming.jiang@ed.ac.uk)                             **
**   Xu Dai(X.Dai@ed.ac.uk)                                           **
**   Praven Kamath(Praveen.Kamath@ed.ac.uk)                           **
**   Asif Usmani(asif.usmani@ed.ac.uk)                                **
**                                                                    **
** ******************************************************************/

// $Date: 2015-12-17
// $Source:
/usr/local/cvs/OpenSees/SRC/SIFBuilder/SIFModelDisplay/SIFModelDisplayIter.h

// Written by: Xu Dai (x.dai@ed.ac.uk)
// Description: This file contains the class definition for SIFModelDisplayIter.
// A SIFModelDisplayIter is an iter for returning the SIFModelDisplays

#ifndef SIFModelDisplayIter_h
#define SIFModelDisplayIter_h

class SIFModelDisplay;
class TaggedObjectStorage;
class TaggedObjectIter;

class SIFModelDisplayIter
{
public:
    SIFModelDisplayIter(TaggedObjectStorage *theStorage);
    virtual ~SIFModelDisplayIter();
    virtual SIFModelDisplay *operator()(void);
    virtual void reset(void);

protected:

private:
    TaggedObjectIter &myIter;
};

#endif
```

HTFireViewer.h

```
/* ****************************************************************** **
**    OpenSees - Open System for Earthquake Engineering Simulation    **
**          Pacific Earthquake Engineering Research Center            **
**                                                                    **
**                                                                    **
** (C) Copyright 2001, The Regents of the University of California    **
** All Rights Reserved.                                               **
**                                                                    **
** Commercial use of this program without express permission of the   **
** University of California, Berkeley, is strictly prohibited.  See   **
** file 'COPYRIGHT'  in main directory for information on usage and   **
** redistribution,  and for a DISCLAIMER OF ALL WARRANTIES.           **
**                                                                    **
** Fire & Heat Transfer modules developed by:                         **
**   Yaqiang Jiang (y.jiang@ed.ac.uk)                                 **
**   Liming Jiang (Liming.Jiang@ed.ac.uk)                             **
**   Xu Dai (x.dai@ed.ac.uk)                                          **
**   Asif Usmani (asif.usmani@ed.ac.uk)                               **
**                                                                    **
** ****************************************************************** */

// $Date: 2015-12-14
// $Source: /usr/local/cvs/OpenSees/SRC/HeatTransfer/HTRecorder/HTFireViewer.h

// Written by: Xu Dai (x.dai@ed.ac.uk)
// Description: This file contains the class implementation of displaying fire shape
//              for visualization, it is initially written for SIFBuilder fire
display,
//              but it can be rewritten for normal heat transfer - fire display.

#ifndef HTFireViewer_h
#define HTFireViewer_h

#include <HTRecorder.h>
#include <Renderer.h>
#include <Vector.h>
class ColorMap;
class SIFBuilderDomain;

extern "C" {
#include <tcl.h>
}

class HTFireViewer : public HTRecorder
{
public:
    HTFireViewer(int tag, const char *title, int xLoc, int yLoc, int width,
        int height, double displayfact, int viewType, int floorNum,
        HeatTransferDomain *theHTdomain, SIFBuilderDomain *theSIFdomain);
    ~HTFireViewer();

    //function to loop over the existing nodes, eles, fires to display
    int record(double timeStamp);

    //methods invoked on the ViewingSystem
    int setVPN(void);               //direction of outward normal to view plane
    int setVUP(void);               //direction defining up direction of view plane
    int setPRP(void);               //eye location in local coordinate system
                                    //defined by viewing system

    int setViewWindow(void);        //view bounds definition
    int setPlaneDist(void);         //distance to front and back clipping
                                    //planes from eye

    int setFillMode(const char *);  //1 = wire, otherwise fill

    //set uMid & vMid values for setPRP() function
    int setuvMid(float uMin, float uMax, float vMin, float vMax);
```

```
68
69      //set parameters for setViewWindow() function
70      int setViewWindowPars(float uMin, float uMax, float vMin, float vMax);
71
72      // methods invoked on the FE_Viewer
73      int displayModel(int eleFlag, int nodeFlag);
74
75  protected:
76
77  private:
78      ColorMap *theMap;
79      Renderer *theRenderer;
80      SIFBuilderDomain* theSIFDomain;
81      HeatTransferDomain* theHTDomain;
82      Vector uvMidPars;
83      Vector ViewWindowPars;
84
85      int FloorNum;   //floor number for displaying, 0: ground floor, 1: first floor ...
86      int ViewType;   //1:ElevationViewZY     2:ElevationViewXY   3:PlanViewZX    4:3D
87      int theEleMode; //determine what pixel size would be used for element display
88      int theNodeMode;//determine what pixel size would be used for node display
89      int wipeFlag;
90      double theDisplayFact;  //relative amplification factor for deformations
91  };
92
93  #endif
```

StrucInFireViewer.h

```
/* ****************************************************************** **
**    OpenSees - Open System for Earthquake Engineering Simulation    **
**          Pacific Earthquake Engineering Research Center            **
**                                                                    **
**                                                                    **
** (C) Copyright 1999, The Regents of the University of California    **
** All Rights Reserved.                                               **
**                                                                    **
** Commercial use of this program without express permission of the   **
** University of California, Berkeley, is strictly prohibited.  See   **
** file 'COPYRIGHT'  in main directory for information on usage and   **
** redistribution,  and for a DISCLAIMER OF ALL WARRANTIES.           **
**                                                                    **
** Developed by:                                                      **
**   Frank McKenna (fmckenna@ce.berkeley.edu)                         **
**   Gregory L. Fenves (fenves@ce.berkeley.edu)                       **
**   Filip C. Filippou (filippou@ce.berkeley.edu)                     **
**                                                                    **
** ****************************************************************** */

// $Date: 2017-07-24
// $Source: /usr/local/cvs/OpenSees/SRC/SIFBuilder/SIFModelDisplay/StrucInFireViewer.h

// Written by: Xu Dai (x.dai@ed.ac.uk) [University of Edinburgh]
// Description: This file contains the class implementation of displaying whole
//              structural response for visualization. It is initially written for
//              displaying structural response using SIFBuilder under fire, but it
//              can be rewritten for normal structural response display under other
//              load types in the future.

#ifndef StrucInFireViewer_h
#define StrucInFireViewer_h

#include <SIFBuilderDomain.h>
#include <Recorder.h>
#include <Renderer.h>
#include <Vector.h>

class ColorMap;

extern "C" {
#include <tcl.h>
}

class StrucInFireViewer : public Recorder
{
public:
    StrucInFireViewer(const char *title, int xLoc, int yLoc, int width, int height,
        double displayfact, int viewType, int floorNum,
        SIFBuilderDomain *theSIFdomain);
    ~StrucInFireViewer();

    //function to loop over the existing nodes & eles to display
    int record(int commitTag, double timeStamp);

    //methods invoked on the ViewingSystem
    int setVPN(void);//direction of outward normal to view plane
    int setVUP(void);//direction defining up direction of view plane
    int setPRP(void);//eye location in local coordinate system defined
                     //by viewing system

    int setViewWindow(void);//view bounds definition
    int setPlaneDist(void); //distance to front and back clipping planes from eye
    int setFillMode(const char *);  //1 = wire, otherwise fill

    //set uMid & vMid values for setPRP() function
    int setuvMid(float uMin, float uMax, float vMin, float vMax);

```

```
69      //set parameters for setViewWindow() function
70      int setViewWindowPars(float uMin, float uMax, float vMin, float vMax);
71
72      //methods invoked on the FE_Viewer
73      int displayModel(int eleFlag, int nodeFlag);
74
75  protected:
76
77  private:
78      ColorMap *theMap;
79      Renderer *theRenderer;
80      SIFBuilderDomain* theSIFDomain;
81      Vector uvMidPars;
82      Vector ViewWindowPars;
83
84      int FloorNum;   //floor number for displaying, 0: ground floor, 1: first floor ...
85      int ViewType;   //1:ElevationViewZY     2:ElevationViewXY   3:PlanViewZX    4:3D
86      int theEleMode; //determine what pixel size would be used for element display
87      int theNodeMode;//determine what pixel size would be used for node display
88      int wipeFlag;
89      double theDisplayFact;  //relative amplification factor for deformations
90  };
91
92  #endif
```

Appendix C

For Chapter 7

C.1 SIFBuilder Tcl script using the ETFM framework

The selected fire and heat transfer modelling results with SIFBuilder using the ETFM framework are discussed in Chapter 7. This modelling capability[1] is developed by the author using C++ with customized Tcl commands for the users, hence a SIFBuilder Tcl script is attached in this section for the interested researchers to further exploit the potential of the ETFM framework and SIFBuilder software. This Tcl script builds up the model of the 'base line scenario' using the ETFM framework, assuming the fuel load density of 570 MJ/m^2, the fire spread rate of 10 mm/s, and the heat release rate per unit area of 500 kW/m^2. The modelling results of this 'base line scenario' are presented in Chapter 7, section 7.2. For the convenience of referring directly in this appendix the schematic structural layout of the case study is shown in Figure C.1 and Figure C.2, which are the same figures presented in Chapter 7: Figures 7.1 and 7.2.

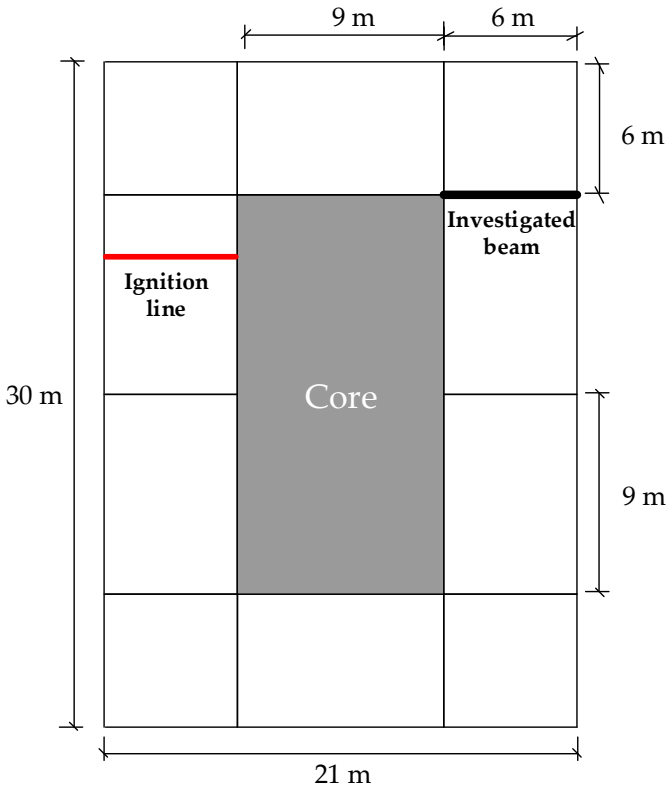


Figure C.1. Case study plan view using the ETFM framework.

[1] The implementation of the ETFM framework in SIFBuilder can be found in Chapter 6.

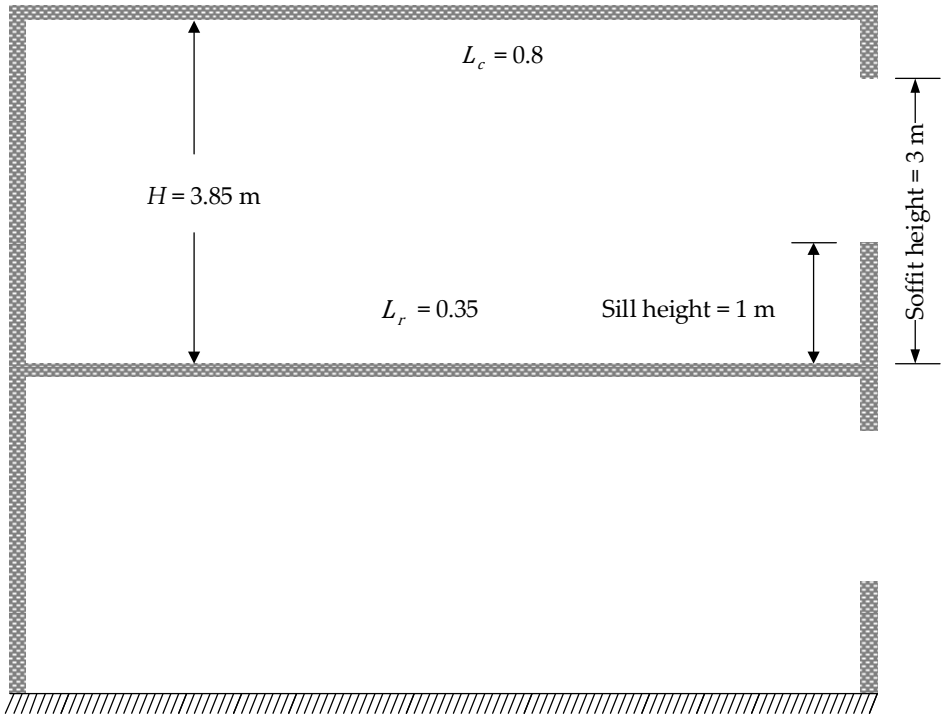


Figure C.2. Case study elevation view using the ETFM framework.

It is strongly recommended the users should understand the limitations of the ETFM framework (see Chapter 4), and OpenSees-based SIFBuilder software (see Chapter 5 and Chapter 6) before they run the relevant simulations – as "*cowboys with computers*" are not encouraged in any sense, especially with the coupled structure fire simulation tool – SIFBuilder.

Moreover, the ETFM framework involves a certain number of input parameters, such as the fuel load densities, ventilation dimensions, fire spread rates, etc. These parameters are actually used to evaluate the fire severity impact on the structure. Meanwhile, the parameters such as the building dimensions, structural element cross sections, material types etc., are to be used to evaluate the structural resistance ability. Along with SIFBuilder using the ETFM framework, a variety of parameter types can be modified based on this attached 'base line scenario' Tcl script. The list of modifiable parameters is summarized in Table C.1, which is the same as presented in Chapter 7, Table 7.1.

```
#####################################################################
#####                                                           #####
#####      (OpenSees 3D - 4*3*2 steel-composite structure)      #####
#####                                                           #####
#####################################################################

# Written by: Xu Dai (x.dai@ed.ac.uk), September 2017, University of Edinburgh
# This Tcl model works in SVN commit version 61
# Full ETFM framework - base line scenario
# SI unit i.e. meter, newton, second

wipe;
set dataDir Case_ETFM_S10.0_F570_H500;
file mkdir $dataDir;

# Define STRUCTURAL MODEL
SIFBuilder -type wholeStruc -ndm 3;
SIFXBay 6 9 9 6;
SIFZBay 6 9 6;
SIFStorey 5 4;

# ASSIGN SECTION
AddMaterial steel 1 -type EC3 2.75e8  2e11;
AddMaterial concrete 2 -type EC2 0 30;
AddSection ISection 1 1 0.2604 0.1022 0.0063 0.01;  # UB 254x102x28
AddSection ISection 2 1 0.3072 0.1243 0.008 0.0121; # UB 305x127x42
AddSection ISection 3 1 0.381 0.3948 0.0184 0.0152; # UC 356x406x235
AddSection SlabSection 4 2 0.15;
AssignSection XBeams 1;
AssignSection ZBeams 2;
AssignSection columns 3;
AssignSection slabs 4;

# Set BOUNDARY CONDITION
SetBC fixedJoint -Locy 0;

# Define LOADING
AddLoad -SIFMember allBeams -load 0 -2000 0;

# FIRE DEFINITION
AddFire -floor 1 -type ETFM -nonFireCompartment 222 322 -IgnitionLine point1 9 5 15
point2 9 5 21 -fireTravelDirection AntiClockWise;

# MORE FIRE INFO
AddFirePars -floor 1 -type ETFM -combination 1 1 -plumeModel Zukoski
-totalHeatLossFraction 0.8 -radiativeHeatLossFraction 0.35 -ventWidth 28 -sillHeight
1 -soffitHeight 3;

# FUEL LOAD DISTRIBUTION DEFINITION
# Unit: mm/s, MJ/m^2, kW/m^2
AddFuel -RMFD 1 -Floor 1 -SpreadRate 10.0 -FuelLoadDensity 570 -HRRperArea 500;

# BUILD MODEL
BuildModel -MeshCtrl 6 6 6 -geomTransf PDelta;

# Define DISPLAY FOR TWO SCREENS
# Unit: pixels
DisplaySIFModel PlanViewZX FloorNumber 1 WindowLoc -1850 50 WindowSize 665 950
ViewScale 20;
DisplaySIFModel 3D WindowLoc -1190 50 WindowSize 950 950 ViewScale 20;
DisplaySIFModel ElevationViewZY WindowLoc 50 50 WindowSize 950 450 ViewScale 20;
DisplaySIFModel ElevationViewXY WindowLoc 50 500 WindowSize 1300 450 ViewScale 20;

# Define OUTPUT RESULTS
SIFRecorder SIFJoint -file $dataDir/SIFJoint112_Disp.out -time -joint 112 disp;
SIFRecorder SIFJoint -file $dataDir/SIFJoint222_Disp.out -time -joint 222 disp;
SIFRecorder SIFMember -file $dataDir/SIFXBeam422_Mid_Deflect.out -time -xBeam 422 mid
-deflect;
```

```
64  SIFRecorder SIFMember -file $dataDir/SIFYBeam332_Mid_Deflect.out -time -yBeam 332 mid
    -deflect;
65  SIFRecorder SIFMember -file $dataDir/SIFColumn542_Mid_Deflect.out -time -column 542
    mid-disp;
66  #SIFRecorder SIFMember -file $dataDir/Slab111.out -time -slab 111 mid-deflect;
67
68  # Apply LOADS & Define ANALYSIS & Define HT OUTPUT
69  SIFAnalyze Fire -dt 20 -output $dataDir -datapoints 9;
70  #SIFAnalyze selfWeight -dt 0.2 Load -dt 0.1 Fire -dt 20 -output $dataDir -datapoints
    9;
71
72  # Print KEY INFO
73  print $dataDir/domain.out
74
75  wipe;
76  #wipeSIFBuilder;
```

Table C.1. Modifiable parameters using the ETFM framework along with SIFBuilder, based on the above Tcl script.

<table>
<tr><th colspan="2">Design fire severity</th><th colspan="2">Structural fire resistance</th></tr>
<tr><td rowspan="3">Fuel load conditions</td><td>Fire spread rate</td><td rowspan="4">Building dimensions</td><td>Each bay length</td></tr>
<tr><td>Fuel load density</td><td>Each bay width</td></tr>
<tr><td>HRR per unit area</td><td rowspan="2">Each bay height (i.e. compartment height)</td></tr>
<tr><td rowspan="3">Ventilation conditions</td><td>Sill height</td></tr>
<tr><td>Soffit height</td><td rowspan="4">Structural member size</td><td rowspan="2">I-beam section (e.g. section depth, flange thickness, etc.)</td></tr>
<tr><td>Total vent width</td></tr>
<tr><td rowspan="2">Trajectory</td><td>Pre-defined path</td><td rowspan="2">Flat concrete slab (e.g. slab thickness)</td></tr>
<tr><td>Travel direction</td></tr>
<tr><td colspan="2">Ignition location</td><td rowspan="2">Structural material</td><td>Steel</td></tr>
<tr><td rowspan="2">Air entrainment model</td><td>Zukoski’s model</td><td>Concrete</td></tr>
<tr><td>Thomas model</td><td rowspan="2">Mechanical loading conditions & combinations</td><td>Concentrated loads</td></tr>
<tr><td colspan="2">Radiative loss fraction of the fire plume</td><td>Uniformly distributed loads (UDL)</td></tr>
<tr><td colspan="2">Heat loss fraction ratio through walls & ceilings</td><td colspan="2" rowspan="4">Non-fire compartments definition (i.e. compartments with full fire protections presumably)</td></tr>
<tr><td colspan="2">Convection coefficient for heat transfer</td></tr>
<tr><td colspan="2">Emissivity of the material for heat transfer</td></tr>
<tr><td colspan="2">Lumped fuel distribution</td></tr>
</table>

C.2 Temperature histories of the sequential beams for all 29 travelling fire scenarios

There are 29 travelling fire scenarios using the ETFM framework which have been analysed for Chapter 7. The temperature histories of the steel beams (right above the fire trajectory) in the case study are presented in full in this section. For the convenience of referring to it in this appendix, Table 7.2 in Chapter 7 is represented here as Table C.2. It is worth to note that for all the travelling fire scenarios, the maximum heat release rate per unit area is assumed to be as a constant: 500 kW/m^2.

Table C.2. Travelling fire scenarios adopted for the ETFM framework.

Fire spread rate (mm/s) / **Fuel load density (MJ/m^2)**	1.6	2.0	5.0	8.0	10.0	15.0
100 (Transport)	•	•	•	•	•	•
230 (Hospital)	•	•	•		•	
300 (Theatre)	•	•	•		•	
420 (Office)	•			•	•	
570	•	•	•	•	•	•
600 (Shopping centre)	•				•	•
780 (Dwelling)	•				•	•

Figure C.3 (same as Figure 7.19 in Chapter 7) schematically shows the tags of these investigated sequential beams and their locations in the case study compartment under all the 29 travelling fire scenarios. The way of naming these beam tags follows the same naming pattern as the SIFBuilder output.

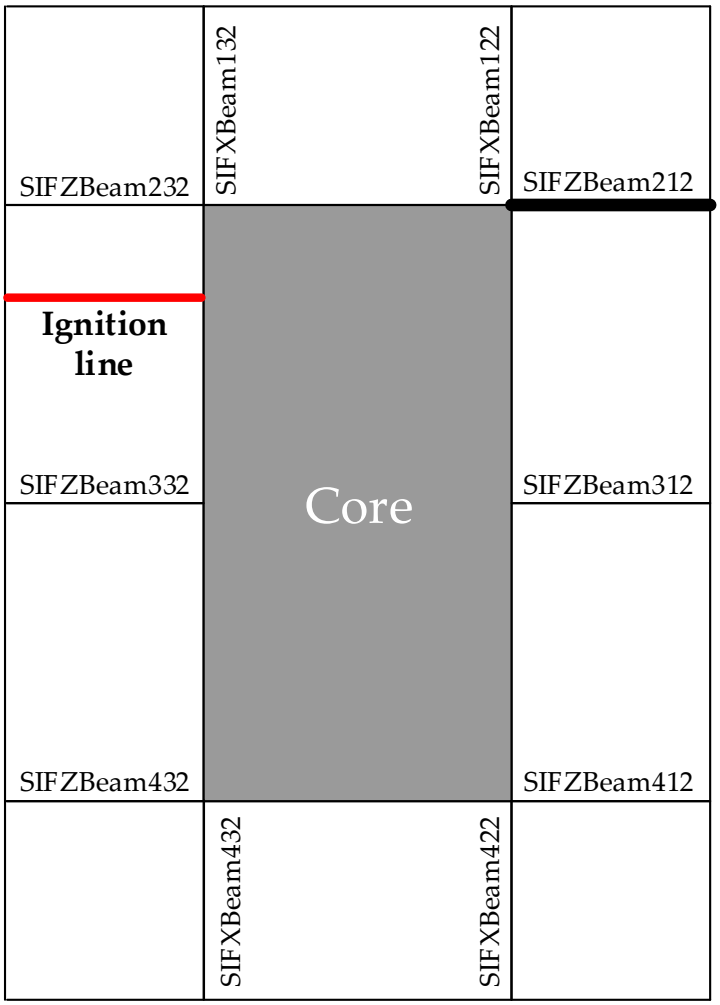


Figure C.3. Schematic of the investigated sequential beams and their corresponding tags.

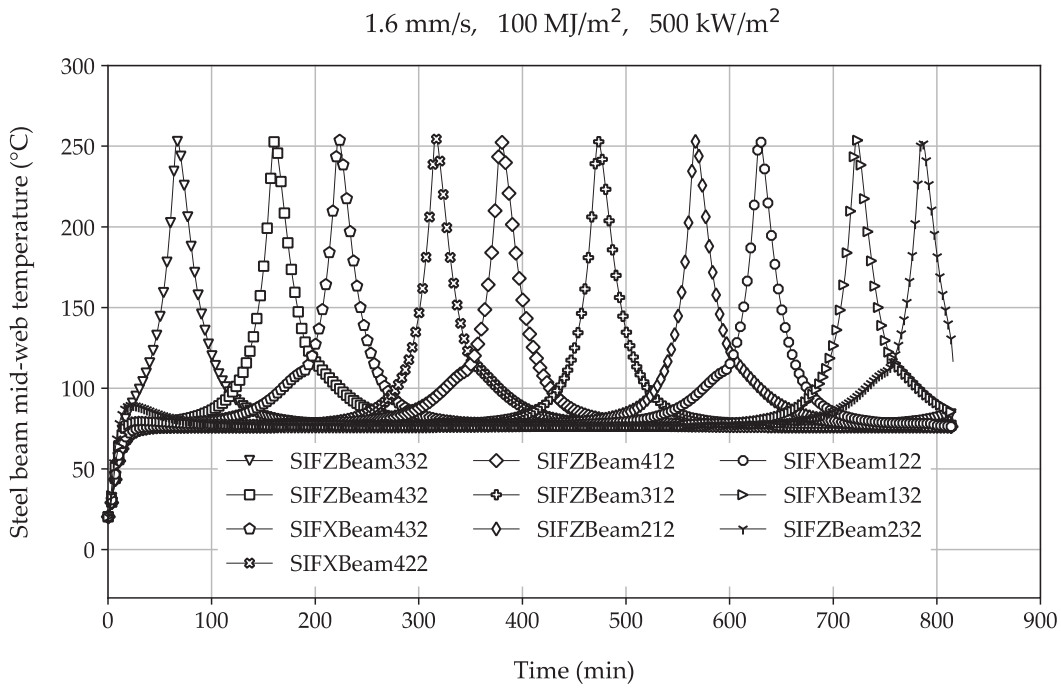


Figure C.4. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 1.6 mm/s, $q_{f,k}$ = 100 MJ/m², RHR_f = 500 kW/m².

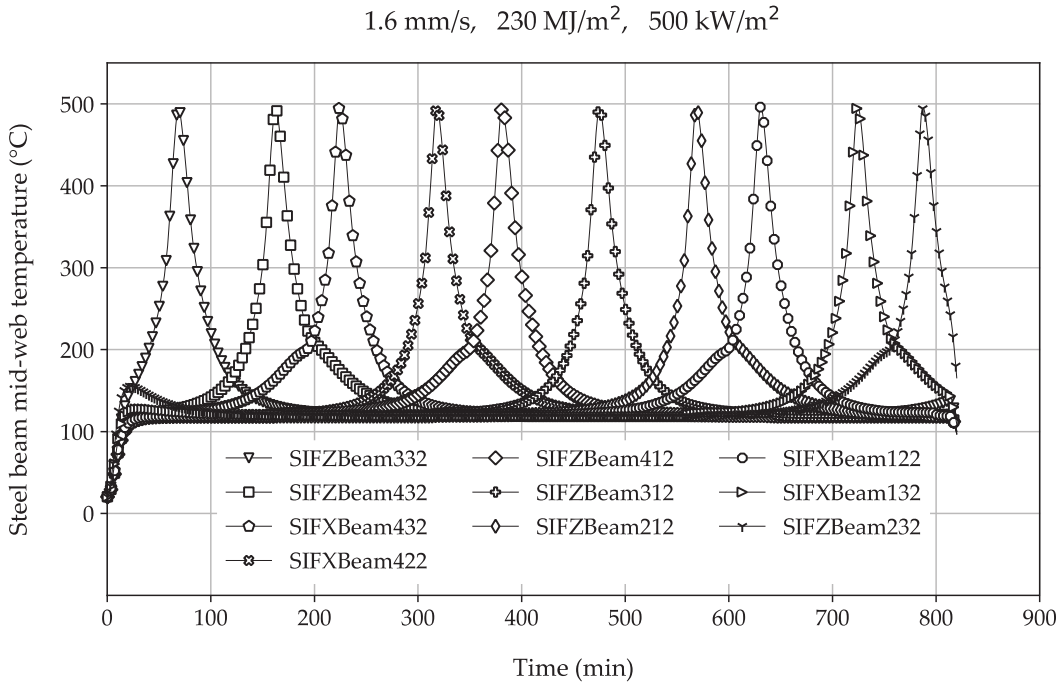


Figure C.5. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 1.6 mm/s, $q_{f,k}$ = 230 MJ/m², RHR_f = 500 kW/m².

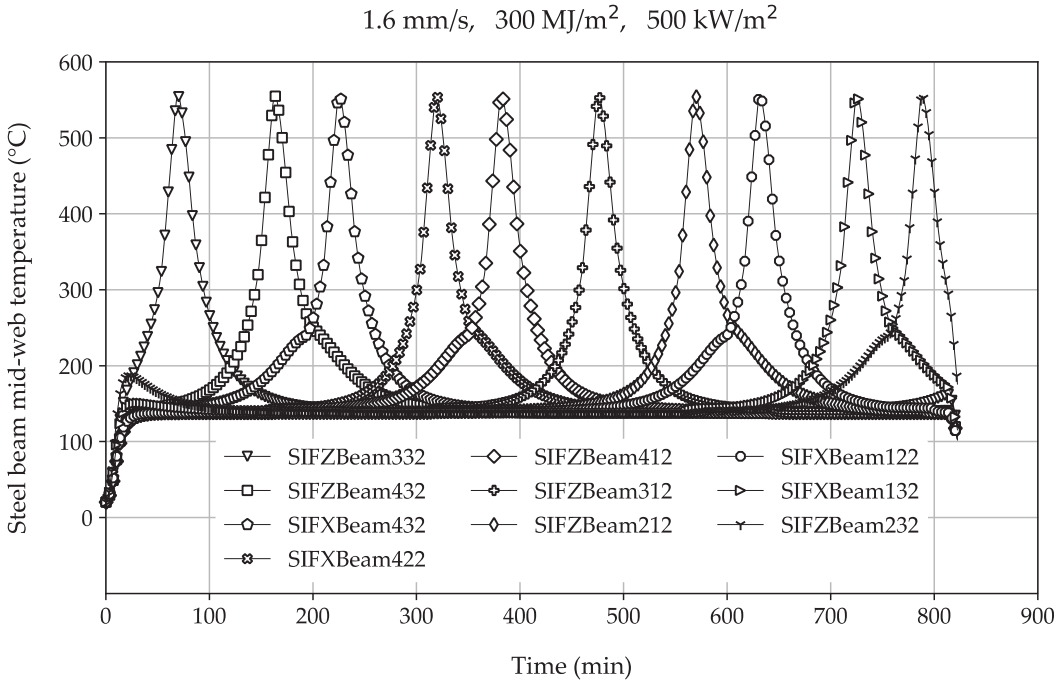


Figure C.6. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 1.6 mm/s, $q_{f,k}$ = 300 MJ/m², RHR_f = 500 kW/m².

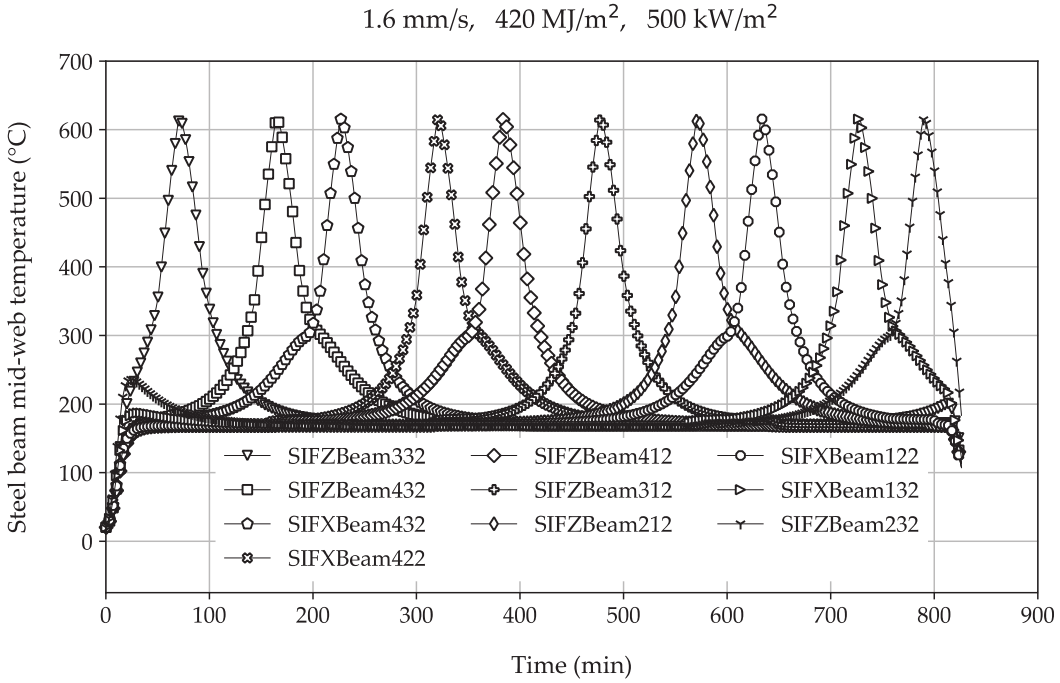


Figure C.7. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 1.6 mm/s, $q_{f,k}$ = 420 MJ/m², RHR_f = 500 kW/m².

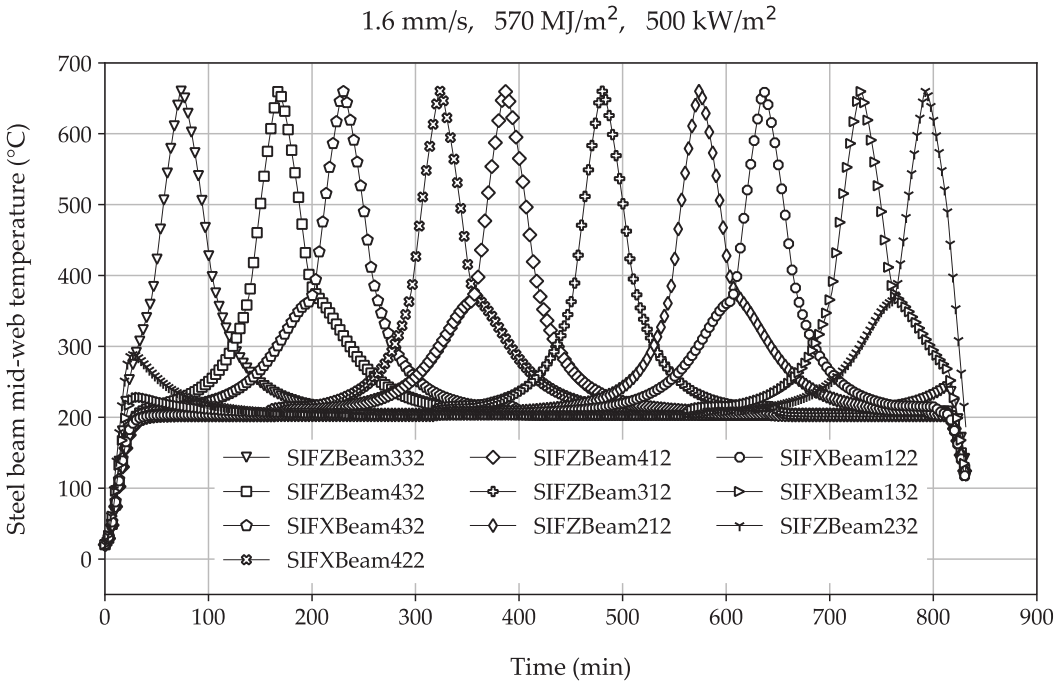


Figure C.8. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 1.6 mm/s, $q_{f,k}$ = 570 MJ/m², RHR_f = 500 kW/m².

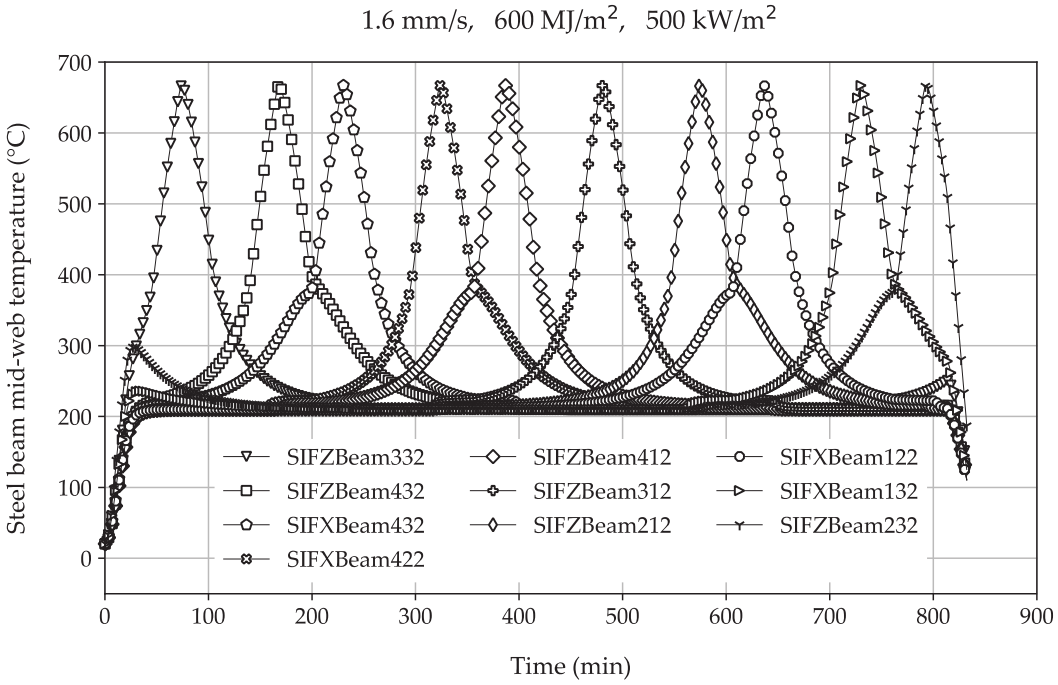


Figure C.9. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 1.6 mm/s, $q_{f,k}$ = 600 MJ/m², RHR_f = 500 kW/m².

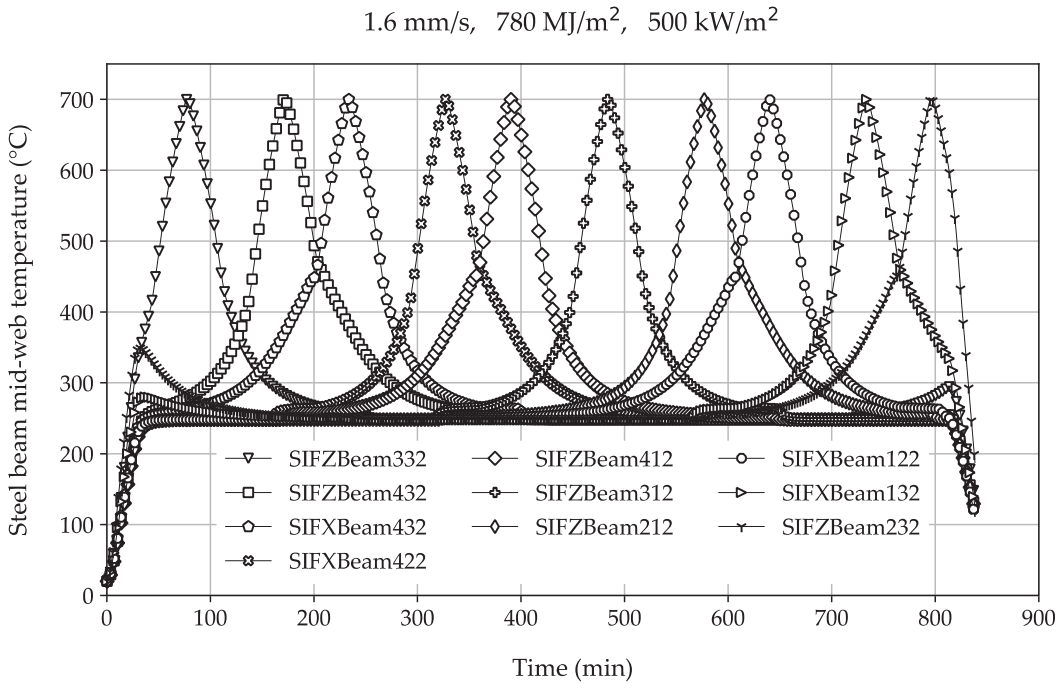


Figure C.10. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 1.6 mm/s, $q_{f,k}$ = 780 MJ/m², RHR_f = 500 kW/m².

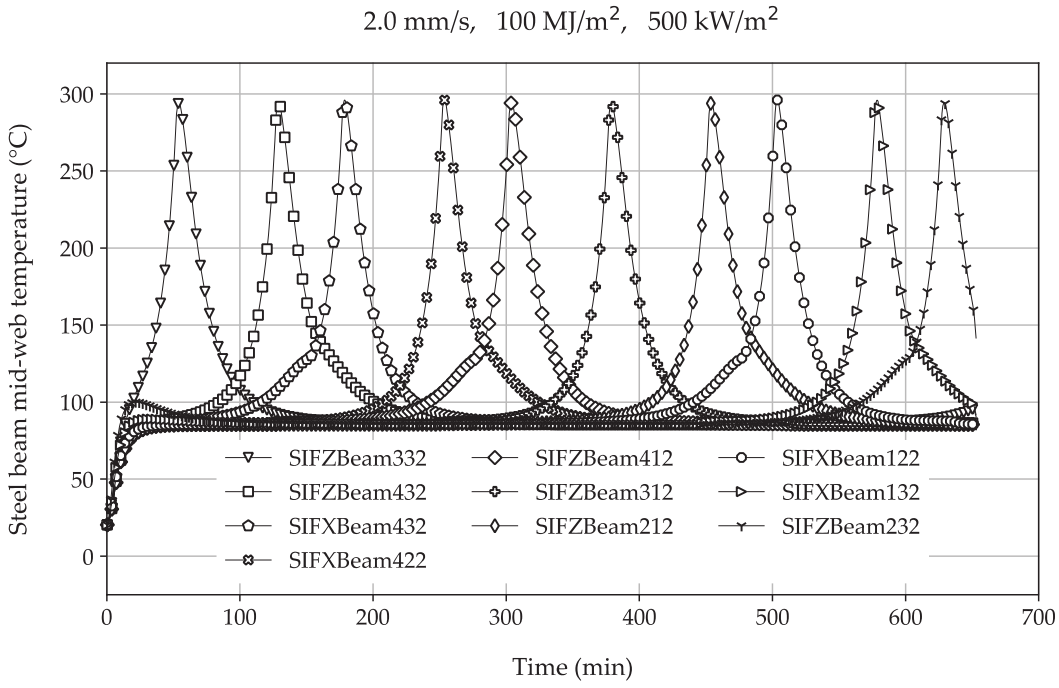


Figure C.11. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 2.0 mm/s, $q_{f,k}$ = 100 MJ/m², RHR_f = 500 kW/m².

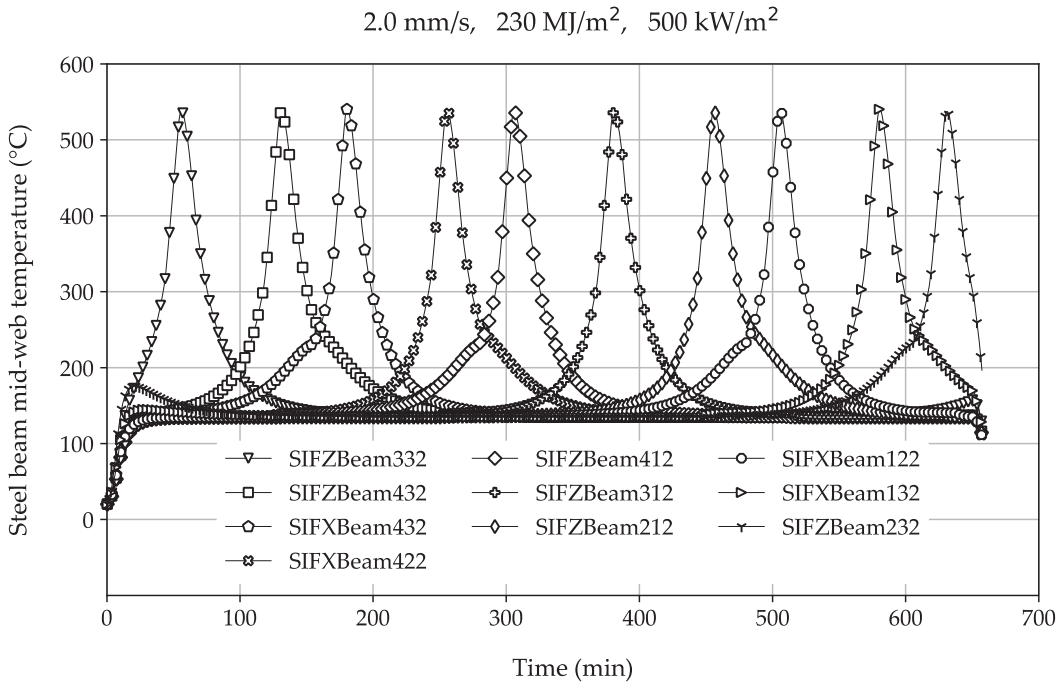


Figure C.12. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 2.0 mm/s, $q_{f,k}$ = 230 MJ/m², RHR_f = 500 kW/m².

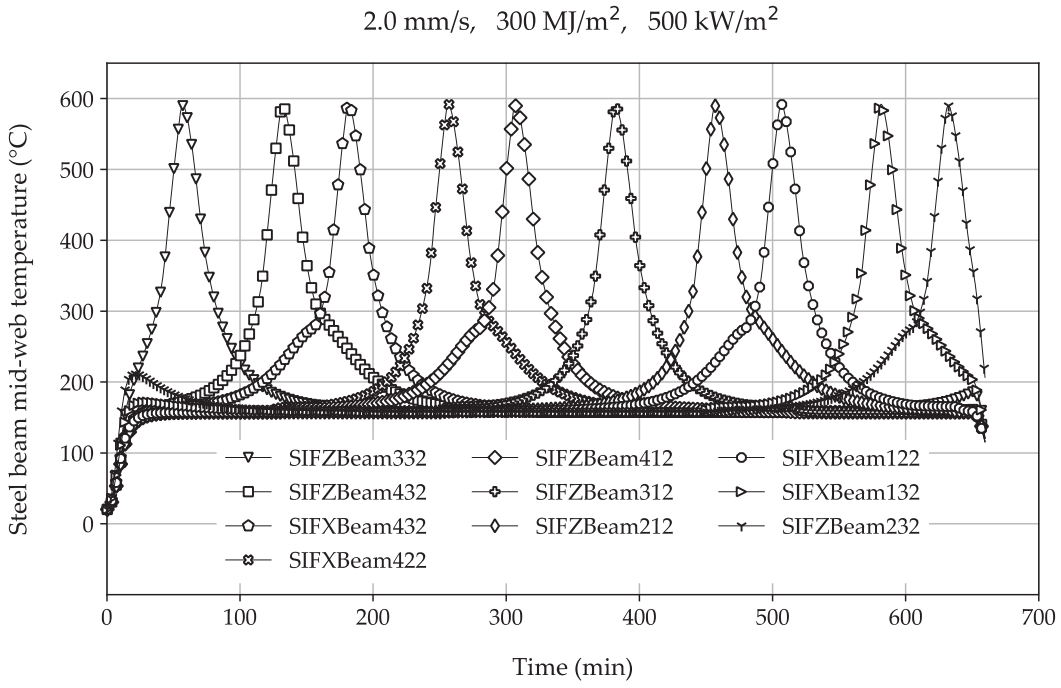


Figure C.13. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 2.0 mm/s, $q_{f,k}$ = 300 MJ/m², RHR_f = 500 kW/m².

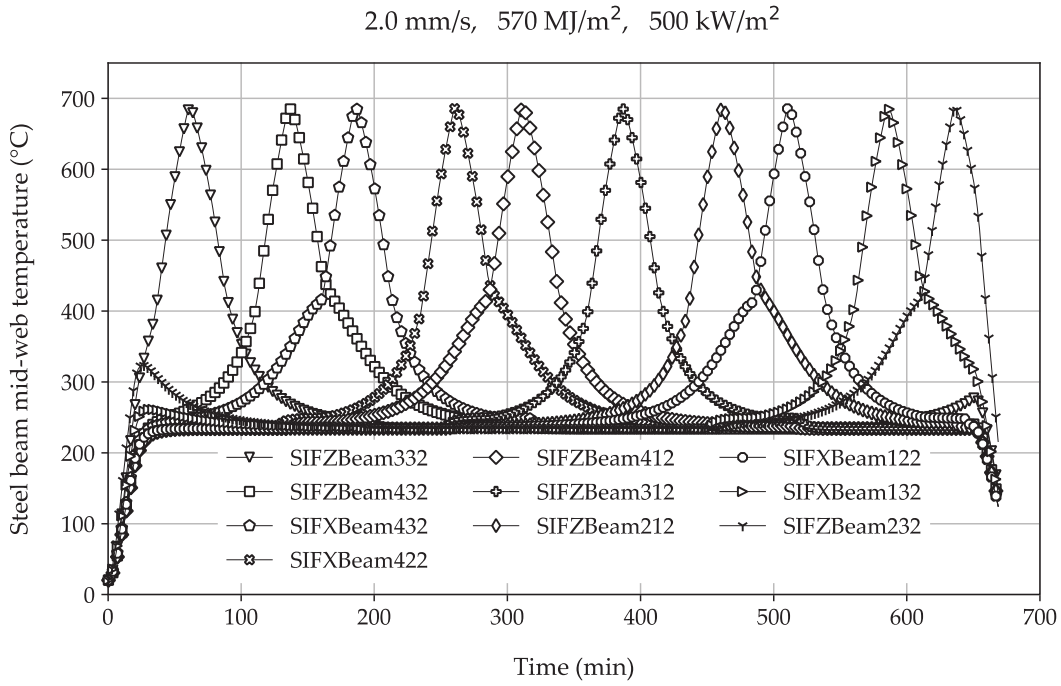


Figure C.14. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 2.0 mm/s, $q_{f,k}$ = 570 MJ/m², RHR_f = 500 kW/m².

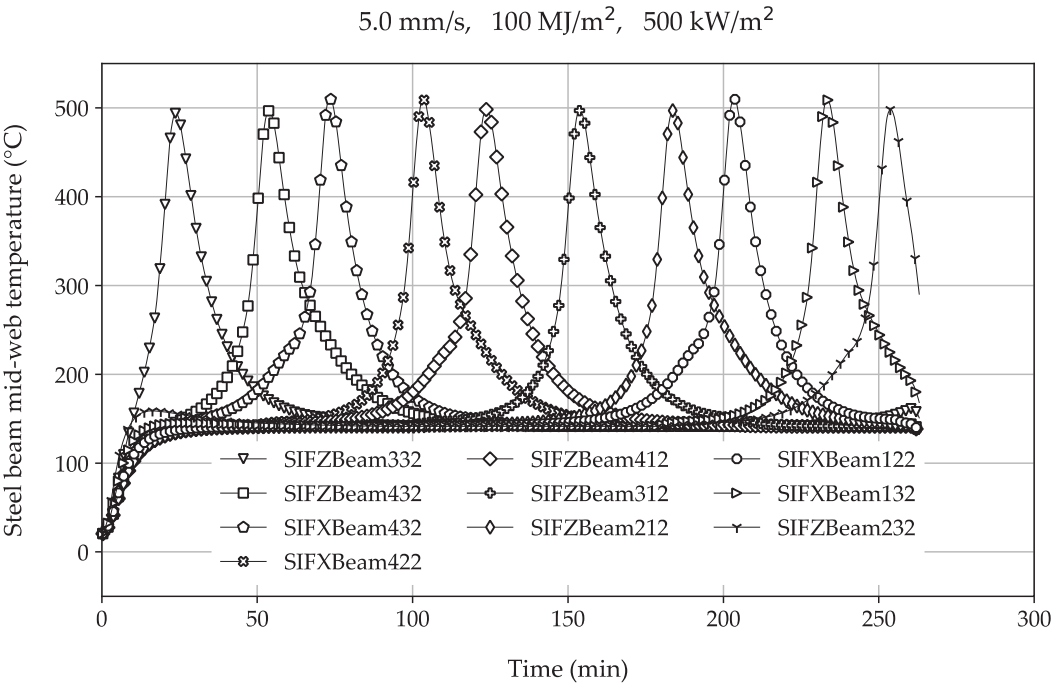


Figure C.15. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 5.0 mm/s, $q_{f,k}$ = 100 MJ/m², RHR_f = 500 kW/m².

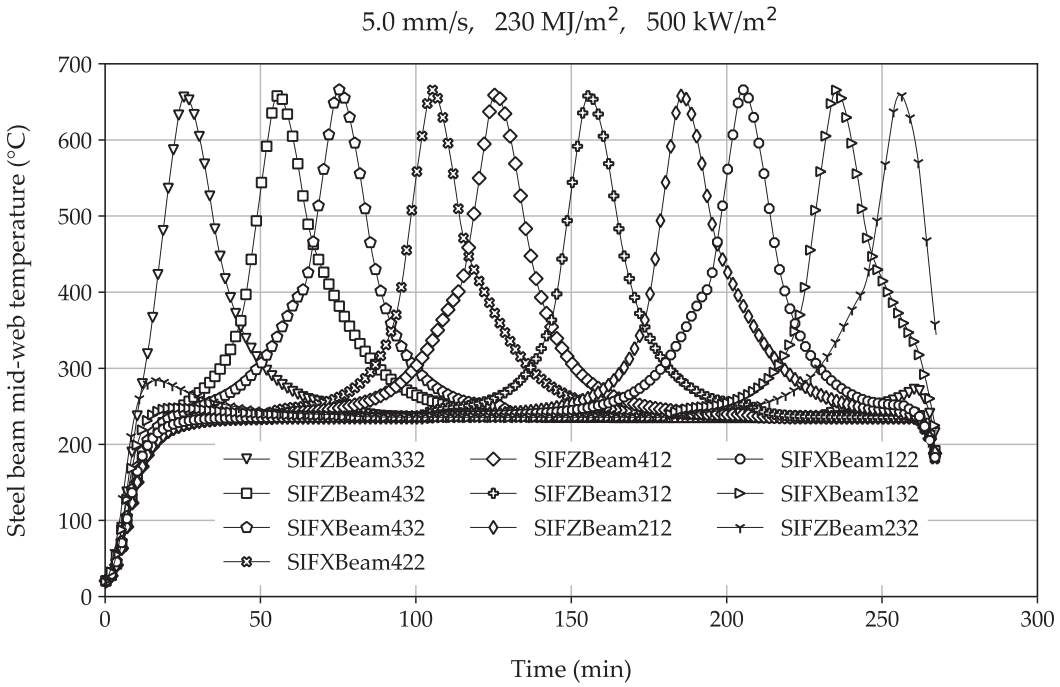


Figure C.16. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 5.0 mm/s, $q_{f,k}$ = 230 MJ/m², RHR_f = 500 kW/m².

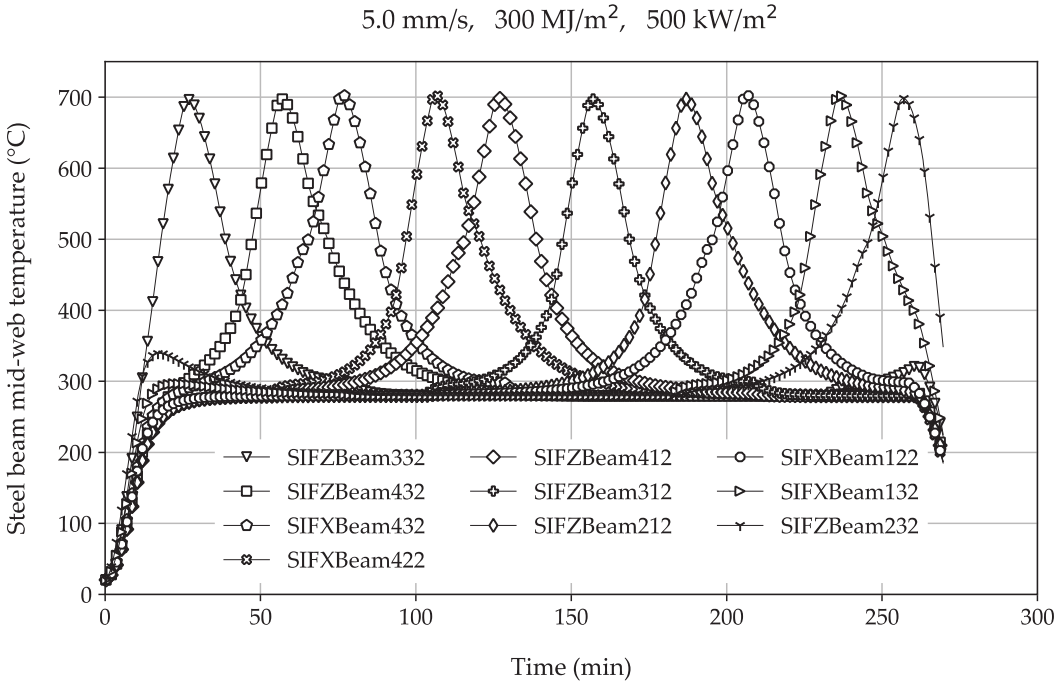


Figure C.17. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 5.0 mm/s, $q_{f,k}$ = 300 MJ/m², RHR_f = 500 kW/m².

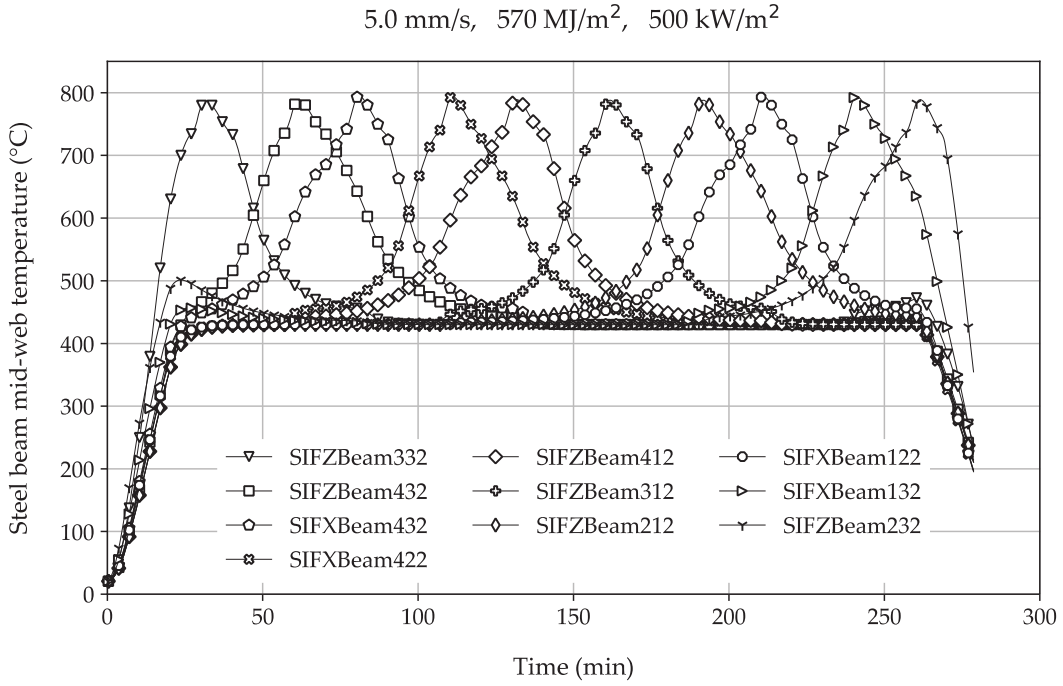


Figure C.18. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 5.0 mm/s, $q_{f,k}$ = 570 MJ/m², RHR_f = 500 kW/m².

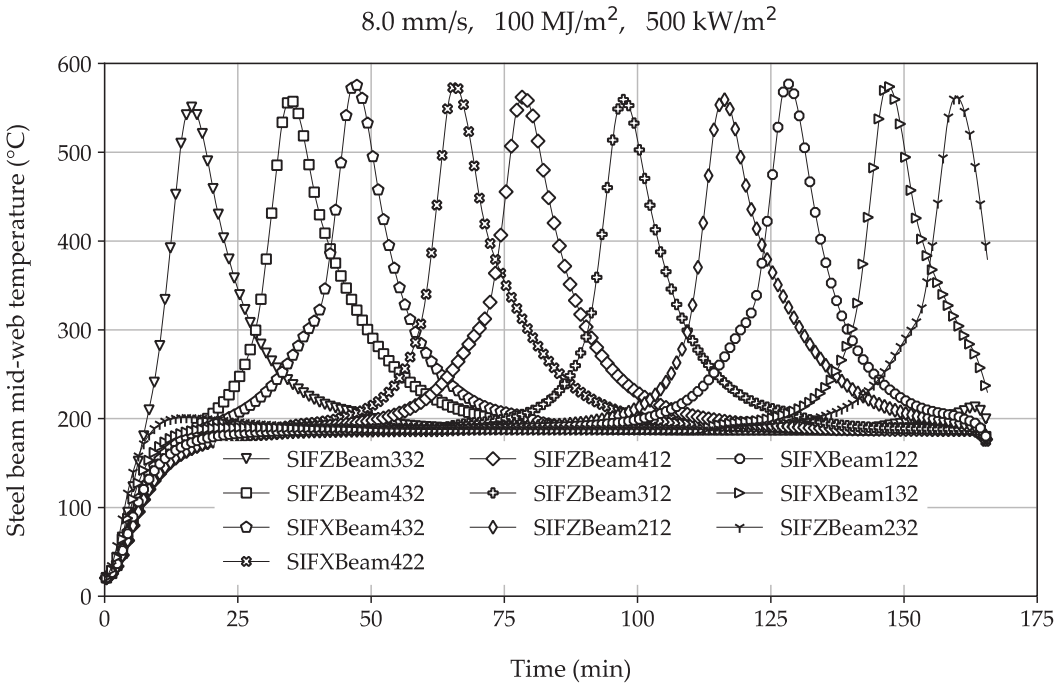


Figure C.19. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 8.0 mm/s, $q_{f,k}$ = 100 MJ/m², RHR_f = 500 kW/m².

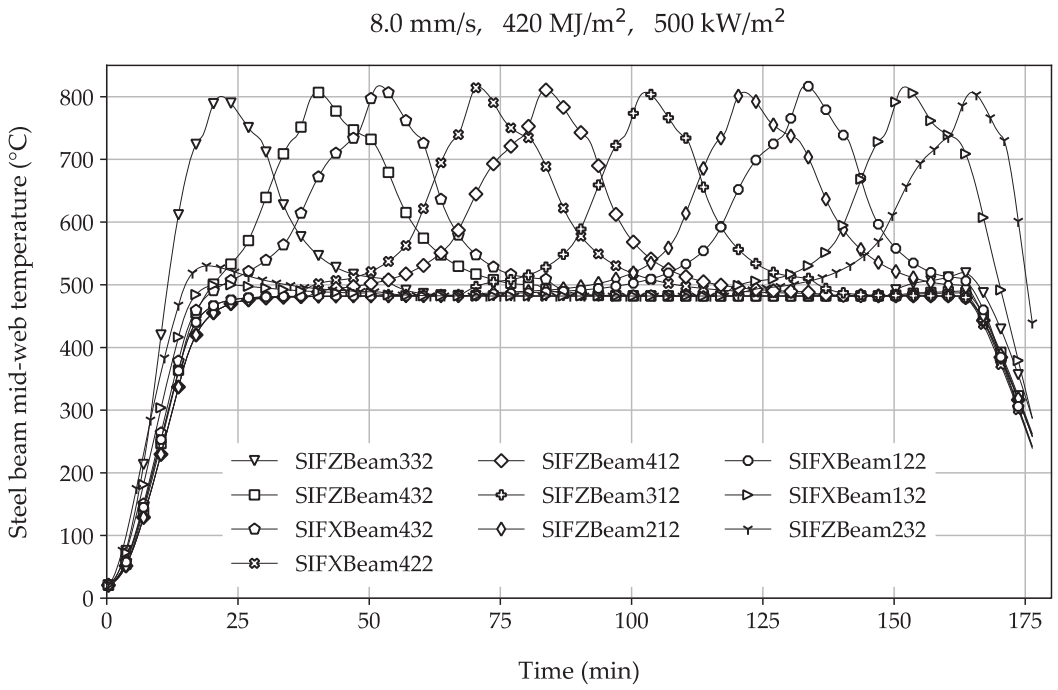


Figure C.20. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 8.0 mm/s, $q_{f,k}$ = 420 MJ/m², RHR_f = 500 kW/m².

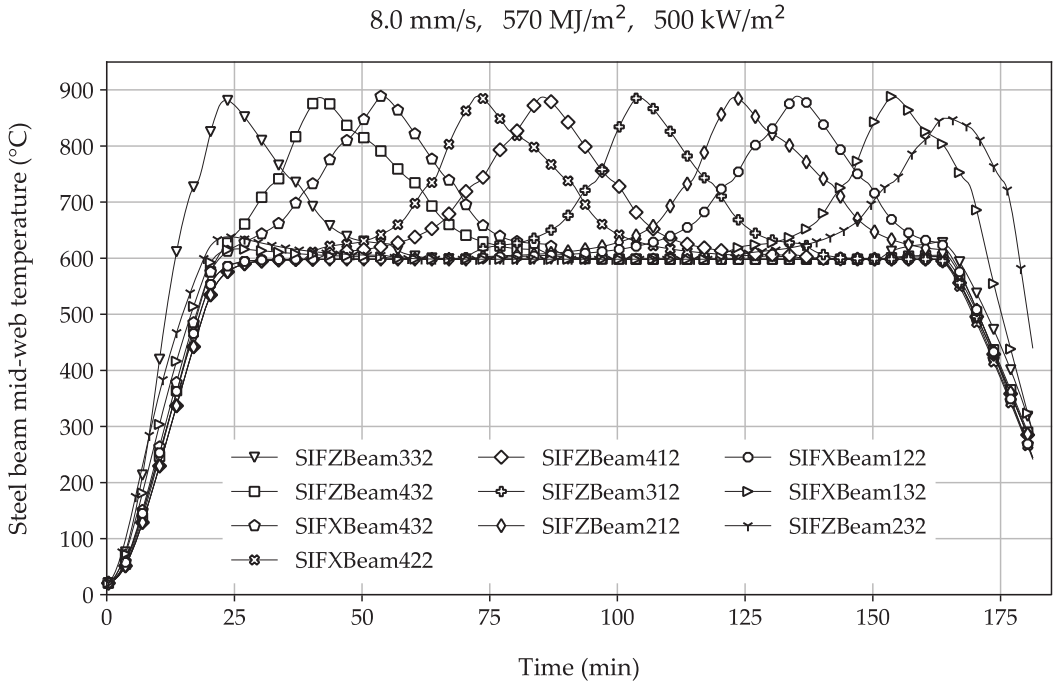


Figure C.21. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 8.0 mm/s, $q_{f,k}$ = 570 MJ/m², RHR_f = 500 kW/m².

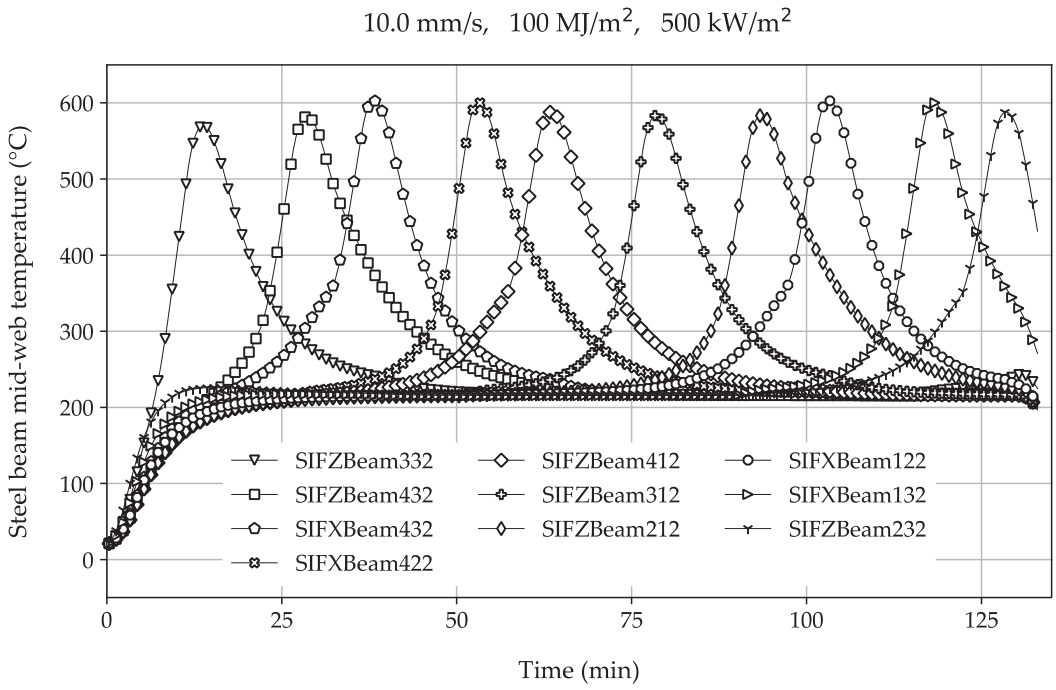


Figure C.22. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 10.0 mm/s, $q_{f,k}$ = 100 MJ/m², RHR_f = 500 kW/m².

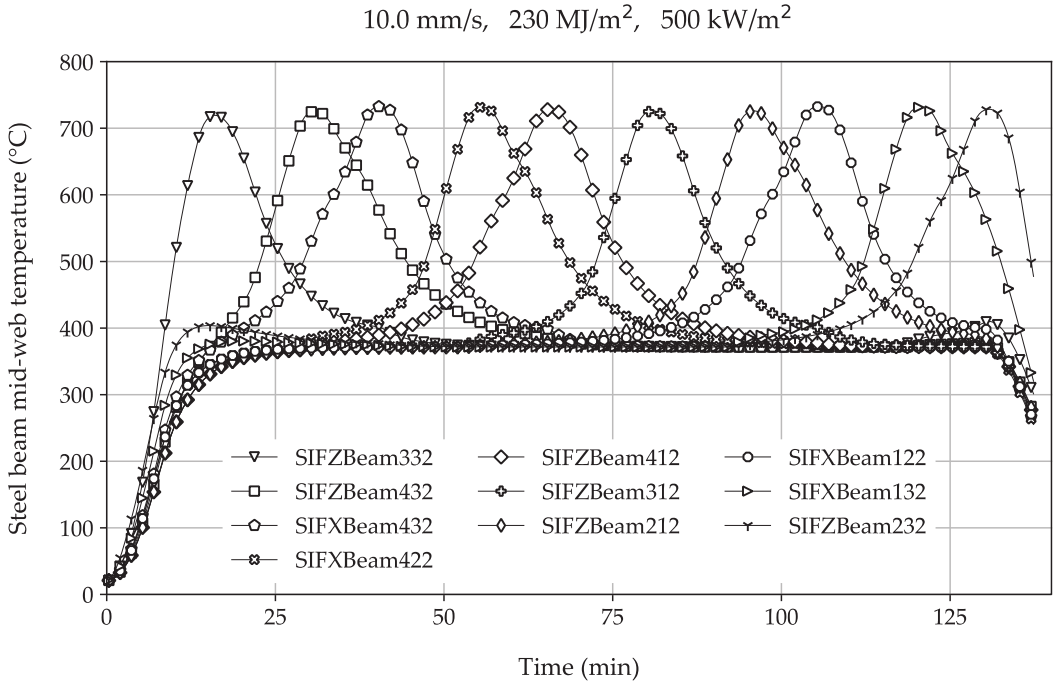


Figure C.23. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 10.0 mm/s, $q_{f,k}$ = 230 MJ/m², RHR_f = 500 kW/m².

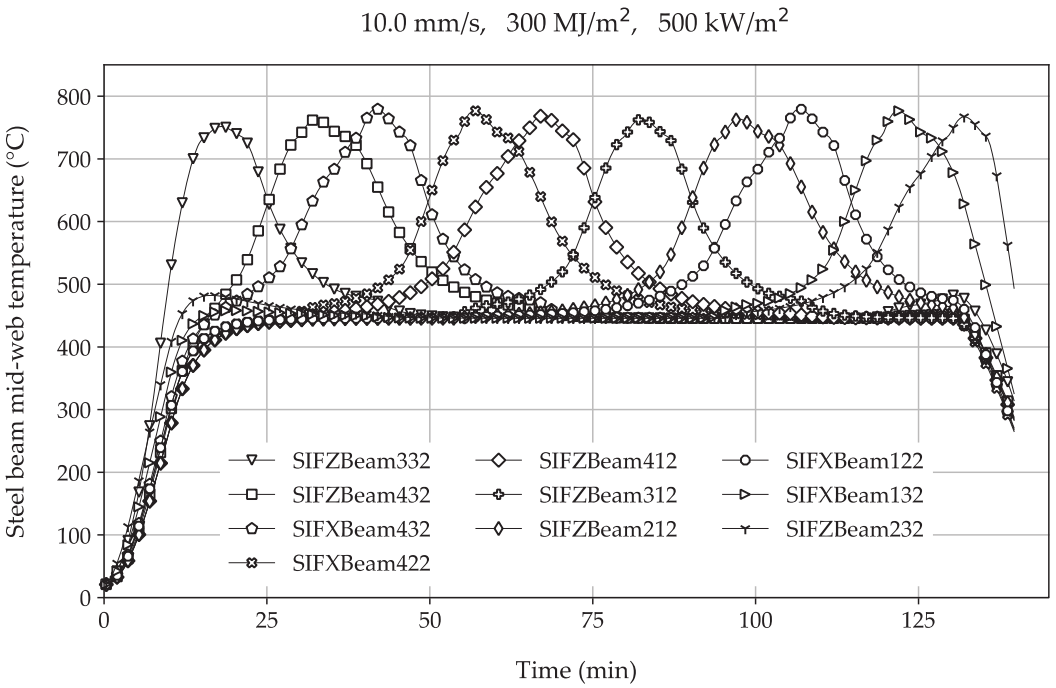


Figure C.24. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 10.0 mm/s, $q_{f,k}$ = 300 MJ/m², RHR_f = 500 kW/m².

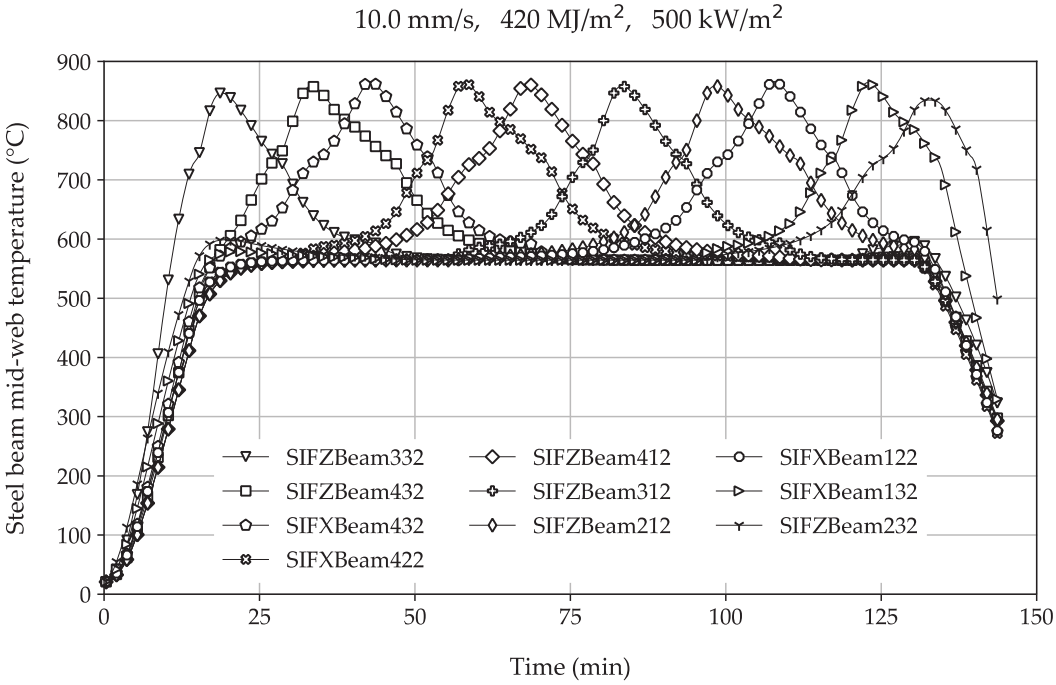


Figure C.25. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 10.0 mm/s, $q_{f,k}$ = 420 MJ/m², RHR_f = 500 kW/m².

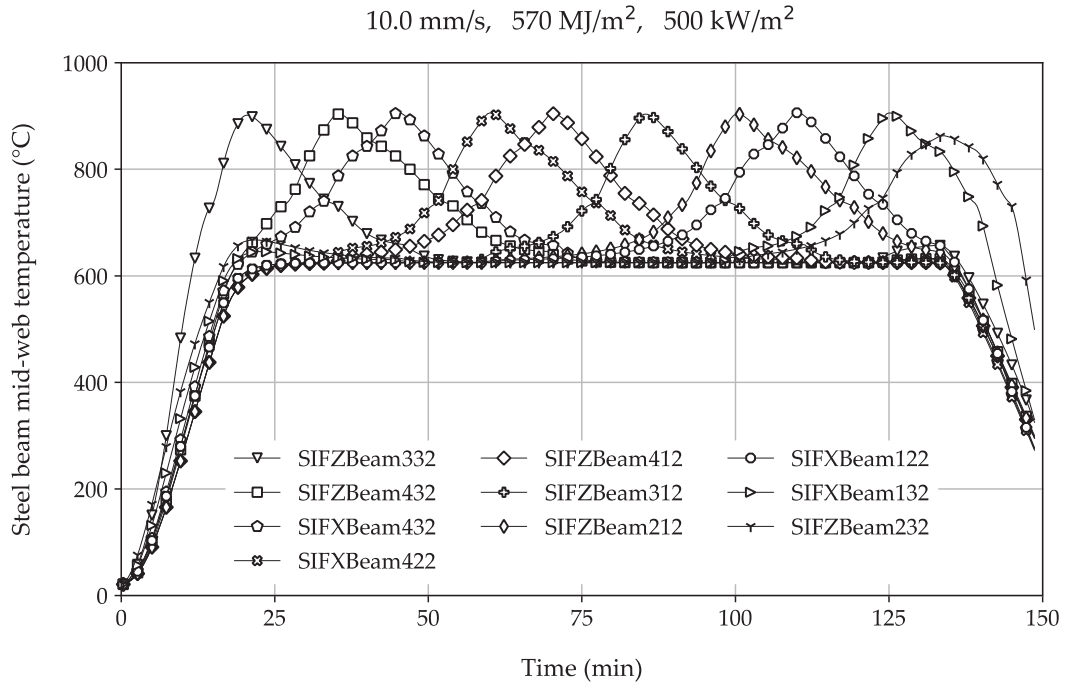


Figure C.26. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 10.0 mm/s, $q_{f,k}$ = 570 MJ/m², RHR_f = 500 kW/m².

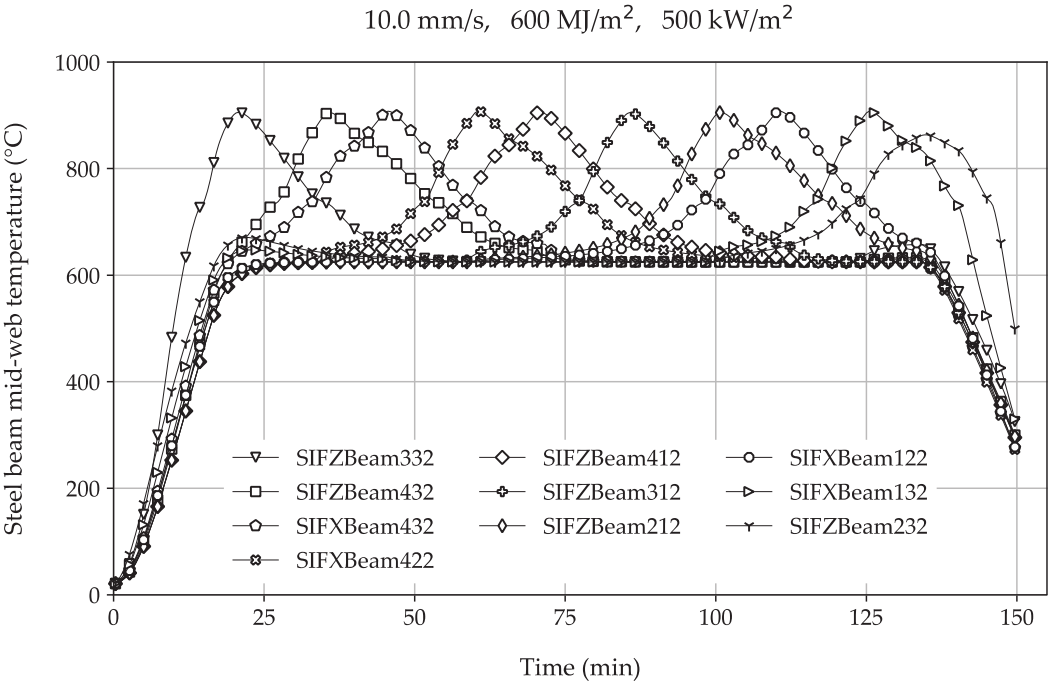


Figure C.27. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 10.0 mm/s, $q_{f,k}$ = 600 MJ/m², RHR_f = 500 kW/m².

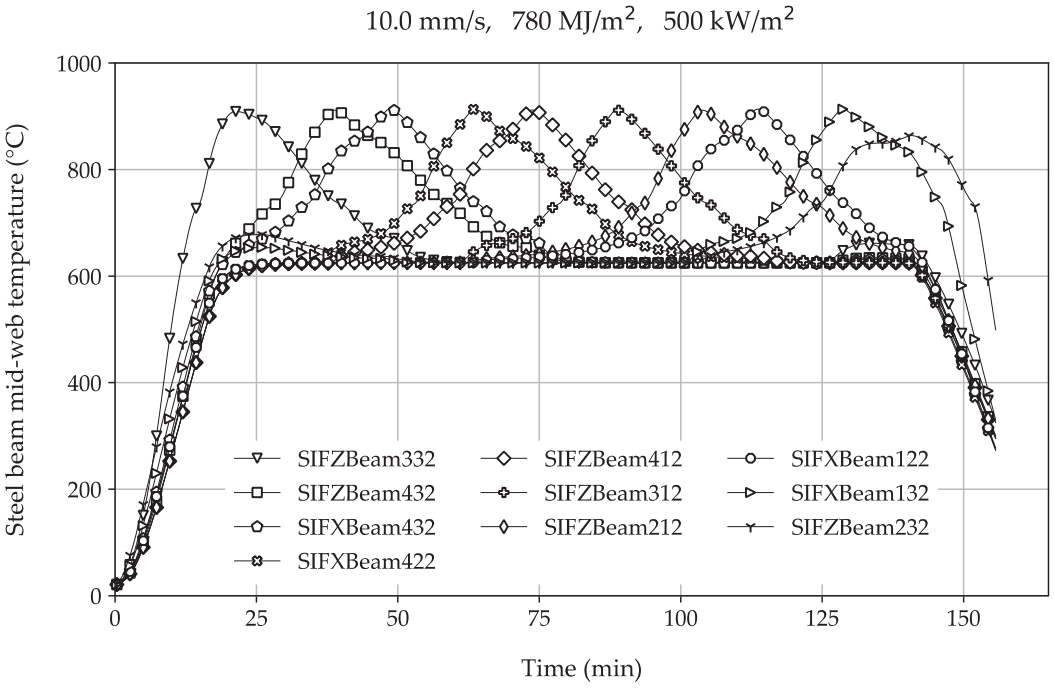


Figure C.28. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 10.0 mm/s, $q_{f,k}$ = 780 MJ/m², RHR_f = 500 kW/m².

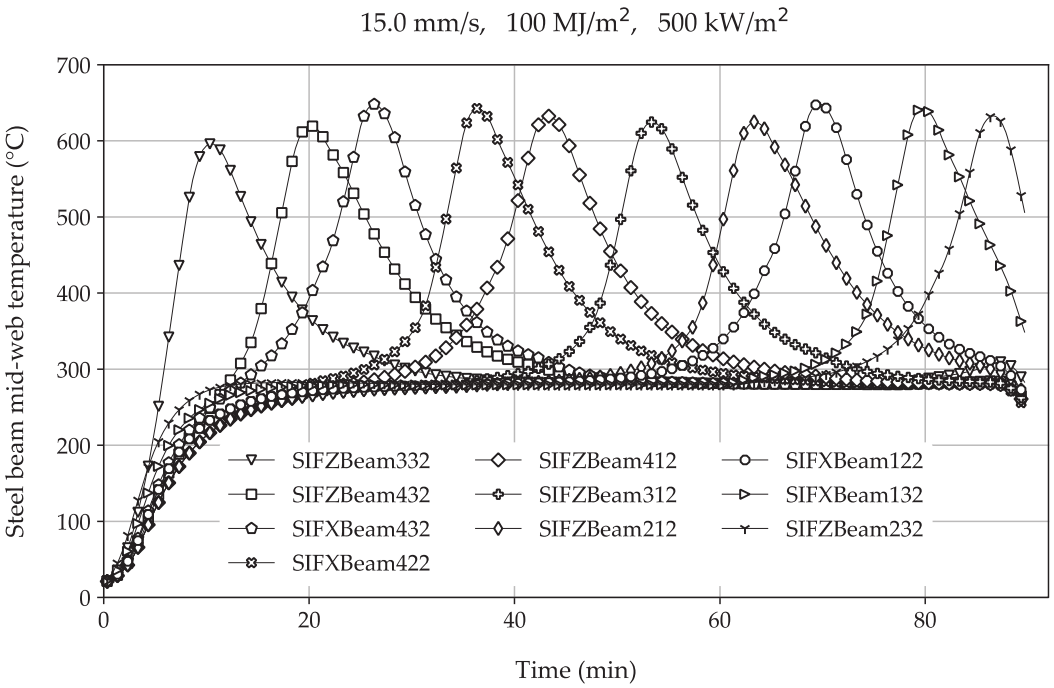


Figure C.29. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 15.0 mm/s, $q_{f,k}$ = 100 MJ/m², RHR_f = 500 kW/m².

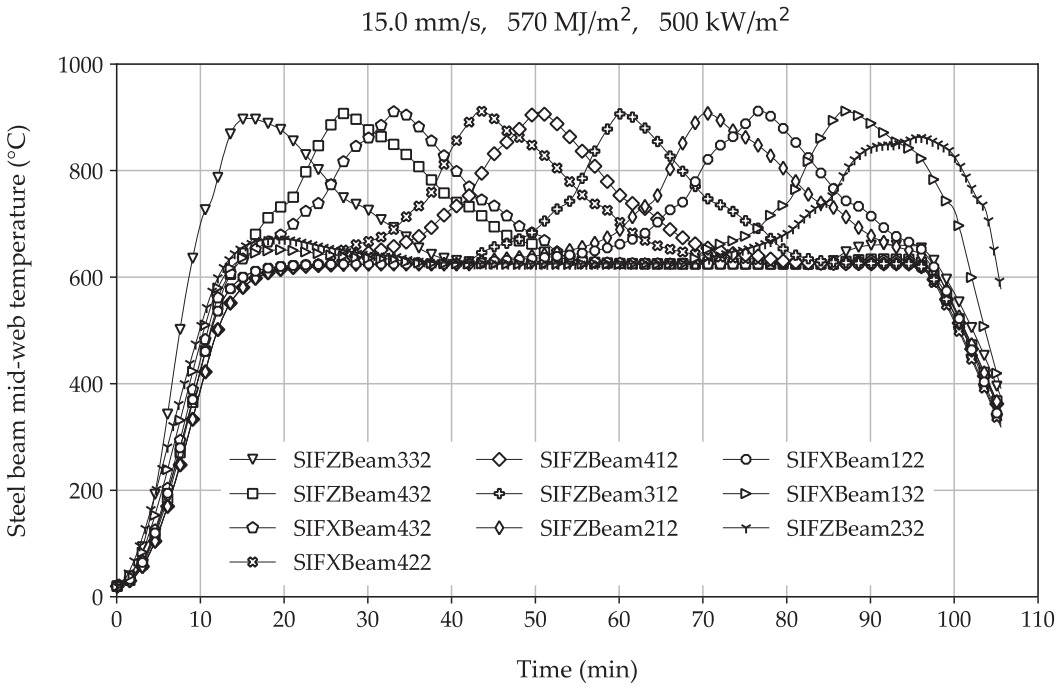


Figure C.30. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 15.0 mm/s, $q_{f,k}$ = 570 MJ/m², RHR_f = 500 kW/m².

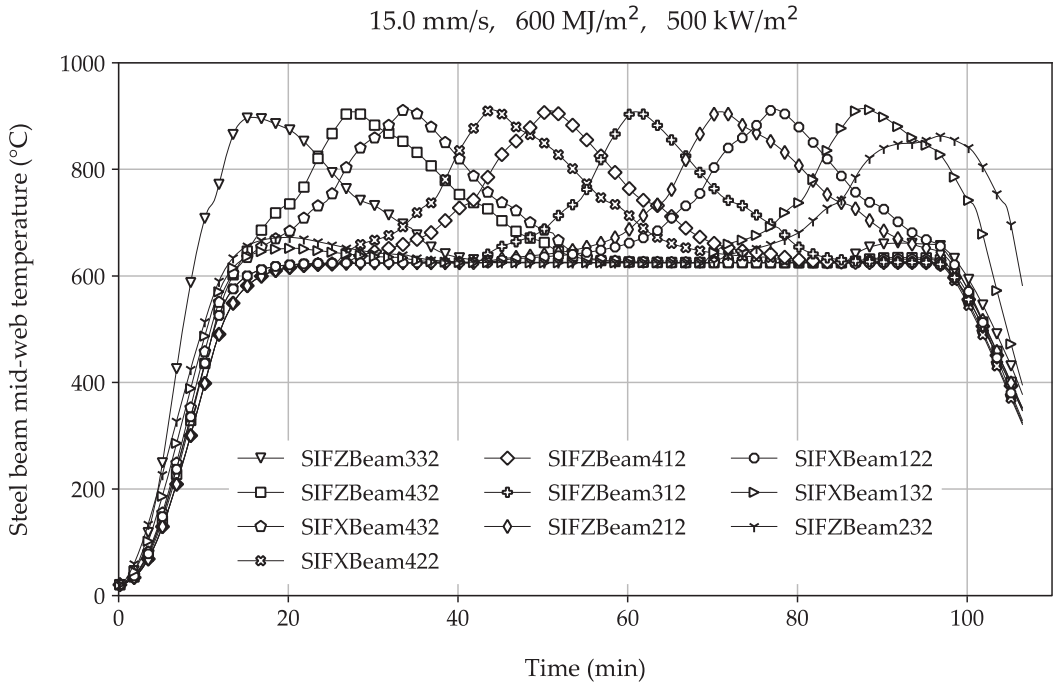


Figure C.31. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 15.0 mm/s, $q_{f,k}$ = 600 MJ/m², RHR_f = 500 kW/m².

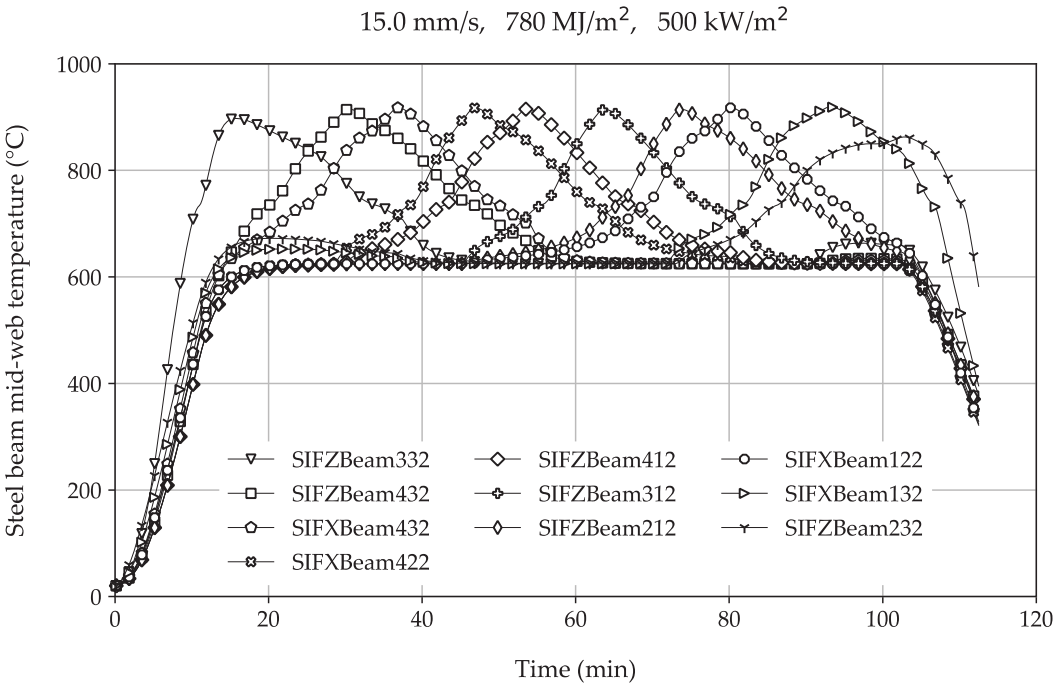


Figure C.32. Temperature histories of the sequential beams right above the fire trajectory, under fire scenario with v = 15.0 mm/s, $q_{f,k}$ = 780 MJ/m², RHR_f = 500 kW/m².