



**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS**

**LOURENÇO JERÔNIMO DE REZENDE VIEIRA**

**ANÁLISE TÉRMICA DE SEÇÕES TRANSVERSAIS EM SITUAÇÃO  
DE INCÊNDIO VIA MÉTODO DOS ELEMENTOS FINITOS**

**BELO HORIZONTE**

**2022**

***PPGEC – Programa de Pós Graduação em Engenharia Civil***

**LOURENÇO JERÔNIMO DE REZENDE VIEIRA**

**ANÁLISE TÉRMICA DE SEÇÕES TRANSVERSAIS EM SITUAÇÃO DE INCÊNDIO  
VIA MÉTODO DOS ELEMENTOS FINITOS**

**Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Civil do Centro Federal de Educação Tecnológica de Minas Gerais como requisito parcial para obtenção do título de Mestre em Engenharia Civil.**

**Orientadora: Profa. Dra. Renata Gomes Lanna da Silva**

**BELO HORIZONTE**

**2022**

V658a      Vieira, Lourenço Jerônimo de Rezende  
Análise térmica de seções transversais em situação de incêndio via  
método dos elementos finitos / Lourenço Jerônimo de Rezende Vieira. –  
2022.  
141 f. : il., gráfs, tabs.

Dissertação apresentada ao Programa de Pós-Graduação em  
Engenharia de Civil.  
Orientadora: Renata Gomes Lanna da Silva.  
Bibliografia: f. 95-99.  
Dissertação (mestrado) – Centro Federal de Educação Tecnológica de  
Minas Gerais, Departamento de Engenharia Civil.

1. Incêndios – Temperatura – Teses. 2. Análise térmica – Teses.  
3. Transferência de calor – Teses. 4. Métodos de elementos finitos  
– Teses. I. Silva, Renata Gomes Lanna da. II. Centro Federal de Educação  
Tecnológica de Minas Gerais. Departamento de Engenharia Civil.  
III. Título.

CDD 621.4021

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS**  
**DEPARTAMENTO DE ENGENHARIA CIVIL**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA CIVIL**

**LOURENÇO JERÔNIMO DE REZENDE VIEIRA**

**AVALIAÇÃO NUMÉRICA AVANÇADA DO DESEMPENHO DE ESTRUTURAS  
PLANAS DE AÇO SOB TEMPERATURAS ELEVADAS**

**Dissertação apresentada ao Programa de  
Pós-Graduação em Engenharia de Civil do  
Centro Federal de Educação Tecnológica de  
Minas Gerais como requisito parcial para  
obtenção do título de Mestre em Engenharia  
Civil.**

**BANCA EXAMINADORA**

---

**Prof. Dra. Renata Gomes Lanna da Silva – Orientadora**

---

**Prof. Dr. Lineker Max Goulart Coelho – CEFET MG**

---

**Prof. Dr. Rodrigo Barreto Caldas – UFMG**

**Belo Horizonte, 31 de março de 2022**

*“Uma vida satisfeita é melhor que uma vida bem sucedida. Porque o nosso sucesso é medido pelos outros, mas nossa satisfação é medida pela sua própria alma, mente e coração”.*

**Dalai Lama**

*“Long Live Rock ‘N’ Roll”*

**Ritchie Blackmore/Ronnie James Dio**

*Aos meus pais, ao meu filho, a minha filha, a minha esposa e as minhas irmãs.*

## **AGRADECIMENTOS**

Aos meus pais, Ângela e Miguel, pelo incentivo, o amor, a dedicação, pela educação e valores ensinados. Não sou nada sem vocês.

Ao meu filho Lorenzo e minha filha Marcela, vocês são a minha luz.

A minha esposa, Josiane, pelo grande incentivo, apoio e motivação.

As minhas irmãs Tatiana, Lara e Jéssica, pela eterna fidelidade, companheirismo e também pelos ótimos momentos de descontração.

A minha avó materna, Maria de Lourdes, sempre estará comigo, em todo tempo e lugar.

A meu sogro, Raimundo, e minha sogra, Efigênia, por me tratarem como filho.

Aos amigos, Guilherme, Danilo, Lucas, Jefferson, Fábio, Leonardo, Gleyson, Braulio, Wallace, Warlei, Luck e tantos outros (as) que não foram listados (as) por simples economia de espaço, mas que não foram esquecidos e nunca serão, sintam-se lembrados.

A minha orientadora Prof. Dra. Renata Gomes Lanna da Silva por todos os ensinamentos compartilhados, confiança, incentivo, disponibilidade, apoio e paciência, imprescindíveis para a conclusão deste trabalho.

A todos os professores que passaram pela minha vida, em especial ao Prof. Dr. Paulo Anderson Santana Rocha. Muito obrigado, por tudo.

Aos colegas do IFMG - Campus Ouro Branco pela excelente convivência.

Ao CEFET-MG e ao PPGEC, todos os professores e funcionários do departamento.

## RESUMO

Nos projetos de edificações, independentemente do material utilizado, os elementos estruturais, além de serem dimensionados para resistir aos esforços atuantes à temperatura ambiente, devem ser avaliados também em situação de incêndio. Em temperaturas elevadas, ocorre a degradação progressiva das propriedades físicas e mecânicas dos materiais, resultando na perda de resistência e rigidez das peças estruturais. A distribuição e a evolução da temperatura nas seções transversais podem ser determinadas por métodos avançados de análise, baseados nas hipóteses da transferência de calor. O trabalho tem como objetivo realizar o estudo da evolução da temperatura nas seções transversais de elementos estruturais de aço, concreto e mistos aço/concreto, submetidas à ação do fogo por meio de uma análise térmica avançada. Dessa forma, foi desenvolvido um programa computacional, escrito em linguagem Fortran 90/95, capaz de realizar a análise térmica em regimes permanente (estacionário) e transiente de elementos bidimensionais, considerando o elemento finito triangular linear com 3 nós. Para validar a eficiência do programa de análise térmica e sua futura inserção no programa para análise termo-estrutural, exemplos numéricos são apresentados, sem e com revestimento térmico, sujeitos a diferentes condições de exposição ao fogo. Os resultados da distribuição da temperatura em função do tempo foram satisfatórios quando comparados com os resultados disponíveis da literatura e pelos softwares Ansys 2021 R1 e SAFIR 2014.a.1, evidenciando o bom desempenho do programa computacional desenvolvido.

Palavras-chave: Análise térmica, Incêndio, Temperatura, Transferência de calor, Método dos elementos finitos.

## **ABSTRACT**

In building projects, regardless of the material used, the structural elements, in addition to being designed to resist the forces acting at room temperature, must also be evaluated in fire situations. At high temperatures, there is a progressive degradation of the physical and mechanical properties of materials, resulting in the loss of strength and rigidity of structural parts. The distribution and development of temperature in cross-sections can be determined by advanced analysis methods, based on the assumptions of heat transfer. The objective of this work is to study the temperature evolution in cross sections of steel, concrete, and steel/concrete composite structural elements subjected to the action of fire by means of an advanced thermal analysis. Thus, a computational program was developed, written in Fortran 90/95 language, capable of performing thermal analysis in steady and transient regimes on two-dimensional elements, considering the linear triangular finite element with 3 nodes. To validate the efficiency of the thermal analysis program and its future insertion in a thermo-structural analysis program, numerical examples subjected to different fire exposure conditions, with and without thermal protection, are presented. The results of the temperature distribution as a function of time were satisfactory when compared with the results available in the literature and obtained by Ansys 2021 R1 and SAFIR 2014.a.1 software programs, showing the good performance of the developed computer program.

**Keywords:** Thermal analysis, Fire, Temperature, Heat transfer, Finite element method



## LISTA DE FIGURAS

Figura 2.1 - Triângulo do fogo .....	28
Figura 2.2 - Curva de incêndio real .....	29
Figura 2.3 - Comparação entre as curvas nominais de incêndio.....	33
Figura 2.4 - Condutividade térmica do aço.....	35
Figura 2.5 - Calor específico do aço.....	37
Figura 2.6 - Deformação térmica específica do aço .....	39
Figura 2.7 - Fatores de redução da relação tensão-deformação dos aços laminados .....	40
Figura 2.8 - Condutividade térmica do concreto em função da temperatura .....	43
Figura 2.9 - Calor específico do concreto em função da temperatura e umidade .....	45
Figura 2.10 - Deformação térmica específica do concreto em função da temperatura .....	47
Figura 2.11 - Massa específica relativa do concreto em função da temperatura .....	48
Figura 2.12 - Coeficiente de redução da resistência à compressão do concreto em função da temperatura .....	50
Figura 2.13 - Coeficiente de redução da resistência à tração do concreto em função da temperatura .....	51
Figura 2.14 - Coeficiente de redução do módulo de elasticidade do concreto em função da temperatura .....	52
Figura 2.15 - Calor específico da Blaze Shield II em função da temperatura.....	56
Figura 2.16 - Condutividade térmica da Blaze Shield II em função da temperatura..	57
Figura 3.1 - Princípio da conservação da energia aplicada ao elemento infinitesimal .....	62
Figura 3.2 - Condições de contorno em um problema de domínio sólido .....	65
Figura 3.3 - Elemento finito implementado .....	68
Figura 3.4 - Variação da temperatura no intervalo de tempo $\Delta t$ .....	69
Figura 3.5 - Algoritmo incremental simples .....	71
Figura 4.1 - Arquivo de entrada.....	73
Figura 4.2 - Arquivo de saída .....	74
Figura 4.3 - Fluxograma geral do PPLANLEP/AI .....	77
Figura 5.1 - Chaminé industrial: (a) Dimensões e condições de contorno, .....	79
Figura 5.2 - Perfil I Laminado IPE 360 .....	80
Figura 5.3 - Nós considerados na análise do perfil/malha de elementos finitos .....	81

Figura 5.4 - Curva temperatura x tempo: Sem revestimento.....	82
Figura 5.5 - Curva temperatura x tempo: Com revestimento.....	83
Figura 5.6 - Curva temperatura x tempo - Ponto D: com e sem revestimento.....	84
Figura 5.7 - Campo de temperaturas para o perfil IPE360 sem a proteção térmica..	85
Figura 5.8 - Campo de temperaturas o perfil IPE360 com a proteção térmica.....	86
Figura 5.9 - Seção transversal da viga mista e pontos estudados ao longo da seção .....	87
Figura 5.10 - Malha de elementos finitos triangulares lineares .....	87
Figura 5.11 - Curva temperatura x tempo: Viga mista.....	88
Figura 5.12 - Campo de temperaturas para a viga mista de aço-concreto.....	89
Figura 5.13 - Seção transversal do pilar T e pontos estudados ao longo da seção ..	90
Figura 5.14 - Malha de elementos finitos triangulares lineares .....	91
Figura 5.15 - Curva temperatura x tempo: Pilar de concreto T.....	91
Figura 5.16 - Campo de temperaturas para o pilar de concreto T.....	92

## LISTA DE TABELAS

Tabela 2.1 - Temperatura dos gases em função do tempo.....	31
Tabela 2.2 - Valores dos fatores de redução em função da temperatura.....	40
Tabela 2.3 - Relação tensão-deformação do aço em temperaturas elevadas .....	41
Tabela 2.4 - Condutividade térmica dos materiais referentes ao concreto à temperatura ambiente .....	42
Tabela 2.5 - Valores de pico para o calor específico do concreto situado entre 100°C e 200°C .....	44
Tabela 2.6 - Coeficiente de redução da resistência à compressão do concreto em função da temperatura .....	49
Tabela 2.7 - Propriedades da argamassa de vermiculita .....	54
Tabela 2.8 - Calor específico da argamassa Blaze Shield II em função da temperatura .....	55
Tabela 2.9 - Condutividade térmica da argamassa Blaze Shield II com a temperatura .....	56
Tabela 3.1 - Matrizes e vetores para o elemento T3 .....	68
Tabela 5.1 - Temperaturas nodais: Chaminé industrial.....	79
Tabela 5.2 - Temperatura em cada ponto verificado por tempo - com e sem revestimento.....	85

## LISTA DE QUADROS

Quadro 4.1 - Subrotinas do PPLANLEP/AI .....	76
--	----

## LISTA DE SÍMBOLOS

<b>A</b>	Área do elemento finito triangular linear [m <sup>2</sup> ]
<b>B</b>	Matriz que contém as derivadas das funções interpoladoras
<b>c</b>	Calor específico do material [J/kg°C]
<b>c<sub>s,θ</sub></b>	Calor específico do aço em função da temperatura [J/kg°C]
<b>c<sub>c,θ</sub></b>	Calor específico do concreto em função da temperatura [J/kg°C]
<b>c<sub>pico</sub></b>	Valor de pico do calor específico em função da umidade do concreto [J/kg°C]
<b>c<sub>BS,θ</sub></b>	Calor específico da argamassa Blaze Shield [J/kg°C]
<b>C</b>	Matriz de capacitância (capacidade térmica)
<b>C<sub>n+β</sub></b>	Matriz de capacitância (capacidade térmica) dentro do intervalo de tempo
<b>D</b>	Matriz que contém as condutividades térmicas
<b>E<sub>s,θ</sub></b>	Módulo de elasticidade do aço em função da temperatura [MPa]
<b>E<sub>s</sub></b>	Módulo de elasticidade do aço à temperatura ambiente [MPa]
<b>E<sub>c,θ</sub></b>	Módulo de elasticidade do concreto em função da temperatura [Mpa]
<b>E<sub>c</sub></b>	Módulo de elasticidade do concreto à temperatura ambiente [MPa]
<b>f<sub>sy,θ</sub></b>	Resistência ao escoamento do aço em função da temperatura [MPa]
<b>f<sub>sy</sub></b>	Resistência ao escoamento do aço à temperatura ambiente [MPa]
<b>f<sub>sp,θ</sub></b>	Limite de proporcionalidade do aço em função da temperatura [MPa]
<b>f<sub>c,θ</sub></b>	Resistência à compressão do concreto em função da temperatura [Mpa]
<b>f<sub>ck</sub></b>	Resistência característica à compressão do concreto [MPa]
<b>f<sub>ct,θ</sub></b>	Resistência à tração do concreto em função da temperatura [Mpa]
<b>f<sub>ctk</sub></b>	Resistência característica à tração do concreto [MPa]
<b>h</b>	Coeficiente de transmissão de calor por convecção e radiação [W/m <sup>2</sup> °C]
<b>k<sub>sy,θ</sub></b>	Fator de redução da resistência ao escoamento do aço
<b>k<sub>sp,θ</sub></b>	Fator de redução do limite de proporcionalidade do aço
<b>k<sub>sE,θ</sub></b>	Fator de redução do módulo de elasticidade do aço
<b>k<sub>c,θ</sub></b>	Coeficiente de redução da resistência à compressão
<b>k<sub>ct,θ</sub></b>	Coeficiente de redução da resistência à tração

$k_{cE,0}$	Coeficiente de redução do módulo de elasticidade
$\mathbf{K}$	Matriz de condutividade térmica
$\mathbf{K}_{n+\beta}$	Matriz de condutividade térmica dentro do intervalo de tempo
$\mathbf{k}_c$	Segunda parcela da matriz de condutividade térmica
$\mathbf{k}_T$	Primeira parcela da matriz de condutividade térmica
$l$	Comprimento inicial do elemento [m]
$L$	Comprimento da aresta do elemento finito triangular [m]
$\mathbf{N}$	Matriz das funções interpoladoras
$q_x$	Fluxo de calor por unidade de área e tempo na direção x [W/sm <sup>2</sup> ]
$q_y$	Fluxo de calor por unidade de área e tempo na direção y [W/sm <sup>2</sup> ]
$q_c$	Fluxo de calor na convecção por unidade de área [W/m <sup>2</sup> ]
$q_r$	Fluxo de calor na radiação por unidade de área [W/m <sup>2</sup> ]
$q_{cr}$	Fluxo de calor na convecção e radiação por unidade de área [W/m <sup>2</sup> ]
$\mathbf{q}_{cr}$	Vetor de fluxo de calor na convecção e radiação
$q_0$	Fluxo de calor prescrito por unidade de área [W/m <sup>2</sup> ]
$\mathbf{q}_0$	Vetor de fluxo de calor prescrito
$\mathbf{q}$	Vetor de fluxos de calor no domínio
$Q$	Calor gerado no interior do elemento por unidade de volume e tempo [J/sm <sup>3</sup> ]
$\mathbf{q}_Q$	Vetor de calor gerado no interior do elemento
$\mathbf{R}$	Vetor de fluxos de calor nodais
$\mathbf{R}_{n+\beta}$	Vetor de fluxos de calor nodais dentro do intervalo de tempo $\Delta t$ [min] ou [s]
$t_0$	Tempo inicial da análise [min] ou [s]
$t_n$	Instante inicial do intervalo de tempo [s]
$t_{n+\beta}$	Instante dentro do intervalo de tempo [s]
$t_{n+1}$	Instante final do intervalo de tempo [s]
$T_0$	Temperatura inicial da análise [°C]
$T$	Temperatura [°C]
$T_\infty$	Temperatura do fluido [°C]
$T_s$	Temperatura na superfície do sólido [°C]

$T_n$	Temperatura no instante inicial do intervalo de tempo [°C]
$T_{n+\beta}$	Temperatura no instante dentro do intervalo de tempo [°C]
$T_{n+1}$	Temperatura no instante final do intervalo de tempo [°C]
$\mathbf{T}$	Vetor de temperaturas nodais
$U$	Teor de umidade do concreto [%]
$\alpha_c$	Coeficiente de transmissão de calor por convecção [W/m <sup>2</sup> °C]
$\alpha_r$	Coeficiente de transmissão de calor por radiação [W/m <sup>2</sup> °C]
$\beta$	Parâmetro de integração no tempo
$\Delta t$	Intervalo de tempo [s]
$\Delta l$	Expansão térmica do elemento devido à temperatura [m]
$\varepsilon_s$	Deformação do aço
$\varepsilon_{sp,\theta}$	Deformação do aço na tensão de proporcionalidade em função da temperatura
$\varepsilon_{sy,\theta}$	Deformação do aço no início do patamar de escoamento em função da temperatura
$\varepsilon_{st,\theta}$	Deformação do aço no final do patamar de escoamento em função da temperatura
$\varepsilon_{su,\theta}$	Deformação do aço na ruptura em função da temperatura
$\varepsilon$	Emissividade
$\theta_g$	Temperatura dos gases [°C]
$\theta_s$	Temperatura no aço [°C]
$\theta_c$	Temperatura no concreto [°C]
$\theta_{BS}$	Temperatura da argamassa Blaze Shield [°C]
$\lambda_{s,\theta}$	Condutividade térmica do aço em função da temperatura [W/m°C]
$\lambda_{c,\theta}$	Condutividade térmica do concreto [W/(m°C)]
$\lambda_{BS,\theta}$	Condutividade térmica da argamassa Blaze Shield [W/m°C]
$\lambda_x$	Condutividade térmica na direção x [W/(m°C)]
$\lambda_y$	Condutividade térmica na direção y [W/(m°C)]
$\rho$	Massa específica do material [kg/m <sup>3</sup> ]
$\rho_{s,\theta}$	Massa específica do aço [kg/m <sup>3</sup> ]

$\rho_{c,\theta}$  Massa específica do concreto em função da temperatura [kg/m<sup>3</sup>]

$\sigma$  Coeficiente de Stefan-Boltzmann [W/m<sup>2</sup>°C<sup>4</sup>]



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>20</b>
1.1	CONTEXTUALIZAÇÃO DA PESQUISA .....	20
1.2	PERGUNTA DE PESQUISA .....	24
1.3	OBJETIVOS .....	24
<b>1.3.1</b>	<b>Objetivo geral.....</b>	<b>24</b>
<b>1.3.2</b>	<b>Objetivos específicos.....</b>	<b>24</b>
1.4	JUSTIFICATIVA E RELEVÂNCIA.....	25
1.5	RESTRIÇÕES E LIMITAÇÕES DA PESQUISA .....	25
1.6	ESTRUTURA DO TRABALHO .....	26
<b>2</b>	<b>REVISÃO TEÓRICA .....</b>	<b>27</b>
2.1	CARACTERÍSTICAS DOS INCÊNDIOS.....	27
<b>2.1.1</b>	<b>Incêndio real .....</b>	<b>28</b>
<b>2.1.2</b>	<b>Incêndio padrão .....</b>	<b>30</b>
2.1.2.1	Curva ISO-834-1:1999.....	30
2.1.2.2	Curva ASTM E119:2000.....	31
2.1.2.3	Curva de hidrocarbonetos .....	32
2.1.2.4	Curva de incêndio para elementos exteriores .....	32
2.1.2.5	Comparação entre curvas nominais .....	33
2.2	PROCESSOS DE TRANSFERÊNCIA DE CALOR.....	33
2.3	PROPRIEDADES DOS MATERIAIS EM ELEVADAS TEMPERATURAS....	34
<b>2.3.1</b>	<b>Propriedades térmicas do aço em elevadas temperaturas.....</b>	<b>34</b>
2.3.1.1	Condutividade térmica .....	34
2.3.1.2	Calor específico .....	36
<b>2.3.2</b>	<b>Propriedades físicas do aço em elevadas temperaturas .....</b>	<b>37</b>
2.3.2.1	Massa específica .....	37
2.3.2.2	Deformação térmica específica .....	37
<b>2.3.3</b>	<b>Propriedades mecânicas do aço em elevadas temperaturas .....</b>	<b>39</b>
<b>2.3.4</b>	<b>Propriedades térmicas do concreto em elevadas temperaturas.....</b>	<b>41</b>
2.3.4.1	Condutividade térmica .....	41
2.3.4.2	Calor específico .....	43
<b>2.3.5</b>	<b>Propriedades físicas do concreto em elevadas temperaturas .....</b>	<b>45</b>

2.3.5.1	Deformação térmica específica .....	45
2.3.5.2	Massa específica .....	47
<b>2.3.6</b>	<b>Propriedades mecânicas do concreto em elevadas temperaturas .....</b>	<b>49</b>
2.3.6.1	Resistência à compressão .....	49
2.3.6.2	Resistência à tração .....	50
2.3.6.3	Módulo de elasticidade .....	52
<b>2.3.7</b>	<b>Materiais de proteção contra o fogo .....</b>	<b>53</b>
2.3.7.1	Argamassas à base de vermiculita .....	54
2.3.7.2	Blaze Shield II .....	54
2.3.7.3	Mantas .....	57
2.3.7.4	Tinta intumescente .....	57
<b>2.3.8</b>	<b>Spalling .....</b>	<b>58</b>
<b>3</b>	<b>ANÁLISE TÉRMICA VIA MEF .....</b>	<b>60</b>
3.1	EQUAÇÃO DIFERENCIAL DA CONDUÇÃO DE CALOR EM SÓLIDOS .....	60
3.2	CONDIÇÕES INICIAIS E DE CONTORNO .....	64
3.3	EQUAÇÃO DE EQUILÍBRIO VIA MEF .....	66
3.4	SOLUÇÃO DO PROBLEMA TRANSIENTE DE CONDUÇÃO DE CALOR .....	68
<b>4</b>	<b>IMPLEMENTAÇÃO COMPUTACIONAL .....</b>	<b>72</b>
4.1	GENERALIDADES .....	72
4.2	CARACTERÍSTICAS DO PROGRAMA PPLANLEP/AI .....	74
<b>4.2.1</b>	<b>Curvas de incêndio .....</b>	<b>74</b>
<b>4.2.2</b>	<b>Propriedade dos materiais .....</b>	<b>75</b>
<b>4.2.3</b>	<b>Elemento finito utilizável .....</b>	<b>75</b>
<b>4.2.4</b>	<b>Condição inicial .....</b>	<b>75</b>
<b>4.2.5</b>	<b>Condições de contorno .....</b>	<b>75</b>
<b>4.2.6</b>	<b>Método de solução do sistema de equações .....</b>	<b>76</b>
4.3	SUBROTINAS .....	76
<b>5</b>	<b>EXEMPLOS NUMÉRICOS .....</b>	<b>78</b>
5.1	CHAMINÉ INDUSTRIAL .....	78
5.2	PERFIL I LAMINADO .....	80
5.3	VIGA MISTA DE AÇO E CONCRETO .....	86
5.4	PILAR DE CONCRETO EM T .....	90

<b>6</b>	<b>CONCLUSÕES .....</b>	<b>93</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>96</b>
	<b>APÊNDICE .....</b>	<b>101</b>
	APÊNDICE A - PROGRAMA PRINCIPAL.....	101
	APÊNDICE B - MÓDULOS .....	101
	APÊNDICE C - SUBROTINA INPUT.....	102
	APÊNDICE D - SUBROTINA PREPROCESSOR .....	104
	APÊNDICE E - SUBROTINA SOLVER .....	105
	APÊNDICE F - SUBROTINA MATRIX .....	109
	APÊNDICE G - SUBROTINA FLUXPRESCVAR .....	122
	APÊNDICE H - SUBROTINA ADDBAN2 .....	136
	APÊNDICE I - SUBROTINA RHS .....	137
	APÊNDICE J - SUBROTINA COLSOL.....	138
	APÊNDICE K - SUBROTINA FLUX_VECTOR.....	139
	APÊNDICE L - SUBROTINA ENERGY .....	141
	APÊNDICE M - SUBROTINA ENSG_SCALNODE .....	142
	APÊNDICE N - SUBROTINA ENSG_VECNODE.....	142

## 1 INTRODUÇÃO

Neste capítulo, apresenta-se, uma concisa contextualização da pesquisa, a justificativa de escolha do tema, os objetivos deste trabalho, limitações e restrições, bem como a organização do texto.

### 1.1 CONTEXTUALIZAÇÃO DA PESQUISA

Nos projetos de edificações, independentemente do material utilizado, os elementos estruturais, além de serem dimensionados para resistir aos esforços atuantes à temperatura ambiente, devem ser avaliados também em situação de incêndio, ou seja, verificar a estrutura quanto à capacidade de resistir aos esforços solicitantes em temperatura elevada. O conhecimento dos efeitos do crescimento da temperatura no comportamento das peças estruturais é fundamental para a análise estrutural sob incêndio e sabe-se que a ação térmica provoca a degradação progressiva das suas propriedades físicas e mecânicas, resultando na perda de resistência e rigidez. Assim, a resistência ao fogo pode ser definida como a capacidade do elemento estrutural permanecer, durante um prazo determinado, desempenhando as funções para as quais foi projetado, sob a ação do incêndio. O tempo de exposição ao fogo, desse modo, é um parâmetro de controle bastante significativo na análise de estruturas sob altas temperaturas.

O projeto de estruturas de aço e de estruturas mistas de aço e concreto em situação de incêndio, no Brasil, é regido pela norma ABNT NBR 14323:2013 e o projeto de estruturas de concreto em situação de incêndio é orientado pela norma ABNT NBR 15200:2012. Essas normas, baseadas nas normas europeias (Eurocódigos), apresentam métodos simplificados, além de permitir o uso de ensaios e também de métodos avançados para a análise dos elementos estruturais em situação de incêndio.

O método de análise térmica realizado por ensaios experimentais é economicamente dispendioso, uma vez que para a sua realização são necessários fornos de grandes dimensões, equipamentos tecnológicos de grande precisão e produção de corpos de prova. Os resultados obtidos por ensaios fornecem parâmetros de entrada e de comparação para a validação de métodos teóricos e numéricos.

Os métodos simplificados apesar de garantirem a segurança da estrutura por serem conservadores, muitas vezes não garantem a economia do projeto. A utilização dos métodos simplificados é prática corrente dos projetistas, uma vez que produzem resultados satisfatórios, de fácil utilização e compreensão. No entanto, apresentam simplificações, como a distribuição de temperaturas uniforme ao longo da seção transversal do elemento estrutural e da propagação uniforme de temperatura no ambiente de incêndio. Para uma análise mais apurada, a utilização dos métodos avançados de estudos térmicos é devida, pode levar a soluções mais seguras e econômicas do ponto de vista do dimensionamento estrutural.

A utilização dos métodos avançados de verificação da segurança das estruturas em situação de incêndio exige modelos para determinar a distribuição e a evolução da temperatura nas peças estruturais. O modelo empregado deve considerar as ações térmicas relevantes, a variação das propriedades térmicas do material em função temperatura, também deve ser fundamentado em princípios reconhecidos e hipóteses da transferência de calor. Neste cenário, a modelagem computacional é uma área que teve um crescimento expressivo nas últimas décadas devido à disponibilidade de novos recursos computacionais. Consequentemente, pesquisas que empregam técnicas numéricas têm sido realizadas, visando um melhor conhecimento e entendimento do desempenho estrutural e dos materiais construtivos sob temperaturas elevadas.

Existem vários métodos aproximados para a solução de equações diferenciais que governam problemas físicos, como o térmico. Desses, destacam-se o método das diferenças finitas (MDF) e o método dos elementos finitos (MEF). Embora possua algumas restrições de solução, o MDF mostra-se uma técnica eficaz, caracterizado por uma aplicação didática associada a uma formulação considerada simples, conforme apresentado por Lie (1994), Xiong et al. (2016) e Neves (2021). O MEF é bastante usado em problemas da engenharia, por ser uma formulação robusta e consistente. Aplicações do MEF para análise térmica, podem ser vistas nas pesquisas de Xu e Wu (2009), Zandi et al. (2012), Behnam e Rezvani (2014) e Neves et al. (2019). Grande parte das pesquisas no Brasil relacionadas à análise das estruturas em situação de incêndio, tem tido cunho predominantemente numérico, como pode ser constatado nos trabalhos brevemente descritos a seguir.

Ribeiro (2004) desenvolveu um programa para análise térmica tridimensional transiente não linear via MEF. Em seu trabalho, o autor comparou os resultados da

análise numérica com os procedimentos prescritos pela norma brasileira ABNT NBR14323:2003 e os Eurocodes. O software denominado THERSYS, implementado na pesquisa, foi validado por meio de comparações com softwares já consagrados.

Caldas (2008) apresentou dois modelos numéricos para a análise térmica transiente, sendo o primeiro baseado no Método das Diferenças Finitas (MDF) e o segundo baseado no Método dos Elementos Finitos (MEF). Para o modelo baseado no MDF, o autor utiliza a formulação proposta em Blomberg (1996), e descreve que o campo de temperaturas, aproximado por valores em pontos discretos que formam uma malha, é considerado em passos de tempo consecutivos, caracterizando um método explícito. Os modelos numéricos apresentados se mostraram adequados para a análise de estruturas em situação de incêndio conforme normas internacionais (EN 1992-1-2:2004; EN 1993-1-2:2005) e com base na comparação dos resultados obtidos com resultados numéricos e experimentais encontrados na literatura. O autor concluiu que os modelos numéricos são capazes de simular, de forma precisa e robusta, o comportamento de estruturas de aço, concreto e mistas em situação de incêndio.

Rigobello (2011) desenvolveu um programa computacional com base no MEF, para análises termo-estruturais de pórtico de aço quando expostas a ações térmicas típicas de situações de incêndio. O programa permite que sejam realizadas análises térmicas transientes com base no método dos elementos finitos para se determinar campos de temperatura nas seções transversais dos elementos estruturais sujeitos ao fogo. A influência da temperatura nas propriedades dos materiais é levada em consideração para se avaliar o desempenho da estrutura em cada instante da análise em situação de incêndio, até que o colapso estrutural seja verificado.

Landesmann (2012) realizou uma análise transiente não linear de transferência de calor na seção transversal de estruturas mistas de aço e concreto, por meio da discretização em fibras. Cada fibra era definida pela coordenada de seu centroide e área. A temperatura, a tensão e a deformação associadas a cada fibra foram consideradas uniformes e, calculadas no centroide de cada elemento de área da fibra.

Pierin et al. (2015) desenvolveram um programa de computador denominado de ATERM, que efetua a análise térmica de estruturas bidimensionais em regime

transiente por meio do MEF. A estrutura verificada pode ser constituída de qualquer material e os resultados foram comparados aos *softwares* SuperTempcalc e ANSYS.

Pires et al. (2018) realizaram uma análise térmica em regime transiente através do MEF para seções transversais mistas aço-concreto por meio da expansão e utilização do Sistema Computacional para Análise Estrutural Avançada/Análise de Fogo (CS-ASA/FA). Duas seções transversais foram analisadas e os resultados da distribuição de temperatura foram satisfatórios. Além disso, diferentes processos de solução iterativa foram adotados na análise. Estudos paramétricos também foram realizados, relacionados à variação da malha dos elementos finitos e aumento do tempo. A partir dos resultados, os autores concluíram que o módulo computacional implementado, CS-ASA/FA, pode fornecer as informações necessárias quando é realizada uma análise termo-estrutural para avaliação das perdas de resistência e rigidez do material estrutural quando exposto ao fogo.

A análise de estruturas em situação de incêndio pode ser realizada por uma análise termo-estrutural, interligando-se as duas análises, a térmica e a estrutural. A análise térmica é caracterizada pela determinação do campo de temperaturas ao longo da seção transversal dos elementos afetados pelo incêndio. Para cada instante de tempo estabelecido do incêndio, a temperatura em determinados pontos da seção transversal é obtida através de método numérico de cálculo de transferência de calor. Esse procedimento possibilita considerar a variação das propriedades térmicas e mecânicas do material em função do aumento de temperatura. A análise estrutural consiste em verificar a estrutura com os efeitos procedentes da análise térmica devidamente considerados. Cabe ressaltar que, no problema termo-estrutural, a consideração dos efeitos inelásticos e de segunda ordem é fundamental.

No presente trabalho, a atenção será direcionada especificamente à etapa de análise térmica, visando realizar o estudo da evolução da temperatura nas seções transversais submetidas à ação do fogo. Dessa forma, foi desenvolvido um módulo computacional capaz de realizar a análise térmica em regimes permanente (estacionário) e transiente em elementos estruturais. No regime permanente, as ações térmicas no contorno não variam com o tempo e, portanto, o campo de temperaturas resultante é constante. No regime transiente, a temperatura não é constante, sendo dependente do tempo. O módulo PPLANLEP/AI (Programa de Pórticos Planos de Aço considerando a Análise Não Linear Elasto-Plástica/Análise

sob Incêndio), implementado em linguagem Fortran 90/95 (Chapman, 2003) e baseado no Método dos Elementos Finitos (MEF) (Cook et al., 1989), fará parte do programa PPLANLEP (Lavall, 1996), que realiza atualmente análise avançada estática (Silva et al., 2018) e dinâmica (Viana et al., 2020) de estruturas de aço em temperatura ambiente.

## 1.2 PERGUNTA DE PESQUISA

Como é a distribuição e evolução da temperatura na seção transversal de elementos estruturais de aço, concreto e mistos aço/concreto durante o incêndio, empregando um método avançado de análise térmica?

## 1.3 OBJETIVOS

### 1.3.1 Objetivo geral

Simular numericamente a distribuição bidimensional de temperatura, resultante da curva de incêndio padrão, na seção transversal de elementos estruturais de aço, concreto, e mistos de aço e concreto, com e sem proteção térmica, utilizando-se um procedimento baseado no Método dos Elementos Finitos.

### 1.3.2 Objetivos específicos

Para o alcance do objetivo principal, foi necessário definir alguns objetivos específicos, a saber:

- Implementar o programa PPLANLEP/AI referente à análise térmica bidimensional da seção transversal de elementos estruturais com e sem proteção térmica;
- Validar a formulação proposta por meio da comparação entre os resultados obtidos na presente pesquisa com as soluções numéricas disponíveis na literatura.



## 1.4 JUSTIFICATIVA E RELEVÂNCIA

O presente trabalho insere-se no contexto da análise avançada de estruturas em situação de incêndio. É importante destacar que, inicialmente, o conceito de temperatura crítica, pelo qual as estruturas de aço eram protegidas limitando a temperatura desse material, juntamente com ensaios experimentais, foram os principais procedimentos de verificação da capacidade resistente das estruturas em situação de incêndio, conforme afirma Caldas (2008).

Atualmente, modelos numéricos e experimentais cada vez mais sofisticados permitem um melhor conhecimento e entendimento do comportamento estrutural sob a ação do fogo. Contudo, ensaios experimentais exigem um alto custo e poucas universidades brasileiras disponibilizam condições para ensaios, em escala real, de uma estrutura nessas condições.

Nunes (2014) destaca que as análises experimentais de estruturas submetidas a ações térmicas características de um incêndio, apesar de proporcionar grande quantidade de dados sobre o comportamento dos elementos estruturais, ou até mesmo de toda a estrutura, têm como notáveis particularidades um elevado grau de complexidade e um custo considerável de tempo, de materiais e de equipamentos.

Diante disso, conforme afirma Pires (2018), os programas computacionais desenvolvidos a partir de pesquisas científicas, como é o caso deste trabalho de pesquisa, se tornam importantes alternativas para a análise estrutural em situação de incêndio. Uma vez calibrado, é possível realizar análises relevantes, várias vezes, para se chegar à melhor forma de representar o desempenho estrutural quando ocorre o incêndio.

O presente trabalho possui também a relevância de minimizar a dependência de pacotes computacionais comerciais. Assim, o código desenvolvido neste trabalho terá por fim possibilitar análises de estruturas de aço em situação de incêndio.

## 1.5 RESTRIÇÕES E LIMITAÇÕES DA PESQUISA

Como restrições a serem mencionadas, dentro do contexto do presente trabalho, citam-se:

- No módulo computacional foi implementado somente o elemento finito triangular linear (T3) com 3 nós.
- Neste trabalho foi implementado apenas o algoritmo iterativo simples para a solução do problema transiente.
- Para a solução do sistema de equações foi implementado somente o método direto  $LDL^T$  proposto por Bathe (1996) com a matriz de rigidez armazenada em esquema *skyline*.
- A análise termo-estrutural não é contemplada

## 1.6 ESTRUTURA DO TRABALHO

O trabalho apresenta-se organizado em seis capítulos: introdução, revisão teórica, metodologia, visão geral do programa, aplicações numéricas com discussão dos resultados e conclusão.

Os conceitos básicos para o entendimento do problema térmico serão apresentados no Capítulo 2. As características de um incêndio, os mecanismos de transferência de calor, o comportamento do aço e do concreto sob elevadas temperaturas, de acordo com as prescrições das normas atuais, serão abordados.

No Capítulo 3 será apresentada a formulação de elementos finitos para a obtenção da distribuição de temperaturas, em nível da seção transversal do elemento, bem como a estratégia de solução do problema transiente de condução de calor, por meio de uma estratégia numérica de integração no tempo, baseada no MDF.

No Capítulo 4 será apresentado o sistema computacional PPLANLEP/AI com uma breve descrição de cada rotina. O Capítulo 5 apresenta as aplicações numéricas desenvolvidas nesta pesquisa e as suas respectivas discussões de resultados. As conclusões e considerações finais serão apresentadas no Capítulo 6, referentes ao desenvolvimento desta pesquisa, finalizando com as sugestões para trabalhos futuros.

## 2 REVISÃO TEÓRICA

A ação térmica associada ao incêndio é considerada excepcional, visto que incêndios são eventos raros na vida útil de uma estrutura. O efeito da ação térmica, durante o evento, acontece quando o calor é transferido dos gases quentes do compartimento, por convecção e radiação, para a superfície dos elementos estruturais e, posteriormente, para o interior da estrutura, por condução.

Logo, o adequado entendimento das características dos incêndios, como também, dos processos de transferência de calor (condução, convecção e radiação) faz-se indispensável. Outro fator fundamental é a alteração, em função da temperatura, das propriedades térmicas, físicas e mecânicas dos materiais que compõem as peças estruturais. No escopo deste trabalho, o aço e o concreto.

A seguir, são apresentadas as curvas que descrevem a dinâmica do incêndio segundo prescrições normativas, a definição dos processos de transferência de calor e a variação das propriedades térmicas, físicas e mecânicas do aço e do concreto, também de acordo com as indicações da normalização pertinente.

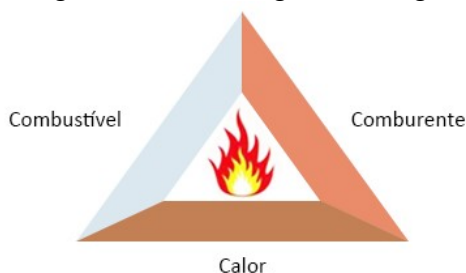
### 2.1 CARACTERÍSTICAS DOS INCÊNDIOS

O incêndio pode ser considerado uma ocorrência de fogo não controlado. Em termos simples, os incêndios não se iniciam sem combustível, fonte de ignição (calor) e comburente (oxigênio). Estes três fatores constituem o que na literatura se esquematiza como sendo o triângulo do fogo, apresentado na Figura 2.1.

A evolução do incêndio depende de vários fatores, como as propriedades térmicas dos materiais que compõe os elementos de compartimentação, a geometria e as dimensões do compartimento, a existência e os tipos de ventilação, dos materiais combustíveis presentes no ambiente (carga de incêndio), a existência e o uso de medidas de combate a incêndio, dentre outros.

Dessa forma, a curva, temperatura x tempo, de um incêndio real não é determinada facilmente, visto os diversos fatores que exercem influência no desenvolvimento do incêndio. Pode-se verificar também que o incêndio real possui características distintas de um ambiente para outro, pois é sensivelmente impactado pelo cenário de incêndio.

Figura 2.1 - Triângulo do fogo



Fonte: Autor

Assim, para auxiliar nos procedimentos de ensaios e projetos de estruturas, o incêndio foi padronizado por curvas, temperatura x tempo, chamadas de nominais, termo genérico utilizado para representar os incêndios padronizados. As curvas nominais são determinadas por equações simples ou tabelas, de aplicação direta e generalizada a qualquer tipo de compartimento, independente dos fatores, anteriormente, citados.

A curva de incêndio real e as curvas padronizadas que modelam a dinâmica do incêndio segundo as normas ISO-834-1:1999, EN 1991-1-2:2002, ABNT NBR 14432:2001 e ASTM E119:2000, são apresentadas a seguir.

### 2.1.1 Incêndio real

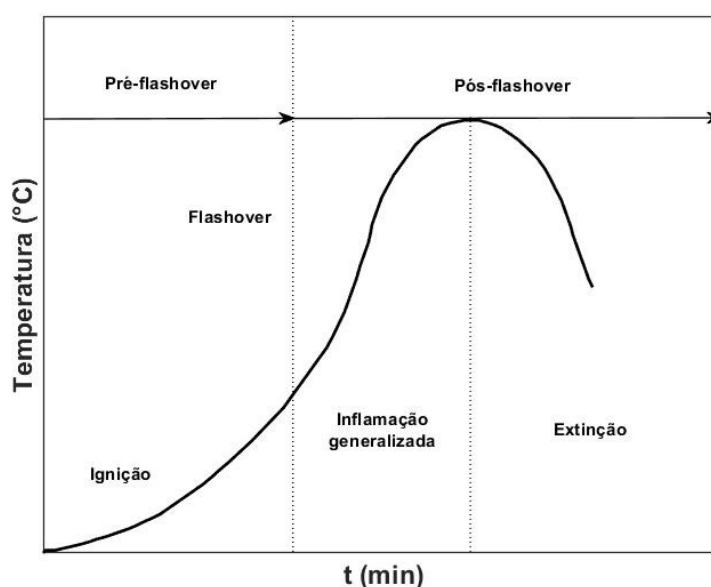
O incêndio real é caracterizado por uma curva temperatura versus tempo que possui um trecho ascendente que representa a elevação da temperatura e um descendente que corresponde ao estágio de resfriamento do ambiente. Em geral, essa curva exibe três fases distintas, conforme mostra a Figura 2.2.

As etapas da curva de incêndio real são definidas por:

- **Ignição:** Representa o estágio de aquecimento no início da inflamação, com crescimento de temperatura. No início da ignição, com duração entre 2 a 5 minutos, a temperatura ambiente aumenta de 20°C a 250-350°C e o incêndio já envolveu algum objeto, mas não todo o compartimento. Nesta etapa, os equipamentos de proteção ativa, como sprinklers, hidrantes, extintores, detectores de fumaça e calor, dentre outros, podem ser suficientes para controlar e extinguir o incêndio, sem prejudicar a segurança estrutural. No final dessa fase é grande o volume de fumaça, e a visibilidade se reduz.

- **Incêndio generalizado:** Pouco antes da segunda fase, verifica-se um crescimento brusco da temperatura, na faixa entre 250 e 350°C, em que todo o compartimento é tomado pelas chamas e o incêndio deixa de ser controlado pelos meios de proteção ativa. Esse instante é denominado de ponto de *flashover*, na qual se inicia a generalização do incêndio, definido por elevadas temperaturas, grandes volumes de fumaça e rápida propagação. Neste segundo estágio, todo o material combustível no compartimento entra em combustão. A duração dessa fase depende de vários fatores, como a carga de incêndio específica, o grau de ventilação e as características térmicas do material componente da vedação. Essa fase de incêndio generalizado dura de 20 a 40 minutos ou até que cerca de 60 a 80% da carga combustível seja consumida. A temperatura pode atingir um valor de 1000°C a 1200°C.
- **Resfriamento ou extinção:** Etapa de redução gradativa da temperatura do ambiente, após a completa extinção do material combustível presente no compartimento. Conforme Gouveia (2006), esta fase pode durar de 1 a 3 horas e podem ocorrer eventos importantes, como o colapso de elementos estruturais e de vedação. Quanto mais rápido for o resfriamento, mais prejudicial será para a resistência do elemento estrutural.

Figura 2.2 - Curva de incêndio real



Adaptado de Pires (2018)

### 2.1.2 Incêndio padrão

A norma ABNT NBR 14432:2001 define o incêndio padrão como a elevação padronizada de temperatura em relação ao tempo, representada por meio de curvas, chamadas também de curvas nominais. Essas curvas são caracterizadas por possuírem somente um ramo ascendente, admitindo que a temperatura dos gases sempre aumente com o tempo, desconsiderando assim os fatores que influenciam no incêndio real. As curvas padronizadas são geradas por meio de equações, com exceção da ASTM E119:2000 (tabela), cuja única variável é o tempo. Logo, cabe destacar, que as informações fornecidas nas curvas nominais devem ser analisadas com bastante cuidado, pois não representam um incêndio real, ou seja, o valor de resistência ao fogo obtido por meio dessas curvas não representa o tempo real que a estrutura resistirá ao incêndio.

As normas nacionais e internacionais propõem a segurança dos elementos estruturais submetidos ao incêndio por um determinado tempo, associado à curva-padrão. Esse tempo, conforme a ABNT 14432:2001, é chamado de tempo requerido de resistência ao fogo (TRRF), o tempo mínimo de resistência ao fogo de um elemento estrutural, quando sujeito ao incêndio padrão. Seguem nos tópicos seguintes as curvas de incêndio padronizadas, nominais.

#### 2.1.2.1 Curva ISO-834-1:1999

A orientação da *International Organization for Standardization* – ISO-834-1:1999 relaciona a elevação da temperatura com o tempo de incêndio para materiais celulósicos, por meio de uma curva logarítmica, utilizada para representar um incêndio completamente desenvolvido num compartimento, expressa da seguinte forma:

$$\theta_g = 20 + 345 \log(8t + 1) \quad (2.1)$$

Na qual  $\theta_g$  é temperatura dos gases em °C e  $t$  é o tempo em minutos.

O código EN 1991-1-2:2002 e a norma nacional ABNT NBR 14432:2001 recomendam para a determinação da resistência de elementos construtivos

submetidos ao fogo, o uso da curva nominal ISO-834-1:1999. Essa curva também serve de base para métodos prescritivos de projeto de estruturas apresentados nas normas brasileiras ABNT NBR 14323:2013 e ABNT NBR 15200:2012.

#### 2.1.2.2 Curva ASTM E119:2000

A *American Society of Test Materials* – ASTM E119:2000 recomenda uma curva padronizada, também para materiais celulósicos, que difere pouco da especificada pela ISO-834-1:1999. É bastante utilizada nos Estados Unidos e inicialmente foi apresentada de forma tabelada. Seguem os valores na Tabela 2.1.

Tabela 2.1 - Temperatura dos gases em função do tempo			
Tempo (min)	Temperatura (°C)	Tempo (min)	Temperatura (°C)
0	20	70	946
5	538	75	955
10	704	80	963
15	760	85	971
20	795	90	978
25	821	95	985
30	843	100	991
35	862	120	1010
40	878	150	1031
45	892	200	1066
50	905	250	1100
55	916	300	1135
60	927	400	1204
65	937	480	1260

Adaptado de ASTM E119:2000

Várias equações para descrever esta curva foram desenvolvidas, sendo que a Equação 2.2 é a que melhor se adéqua aos valores da Tabela 2.1.

$$\theta_g = 20 + 750 \left( 1 - e^{-3,79553\sqrt{t}} \right) + 170,41\sqrt{t} \quad (2.2)$$

Onde  $\theta_g$  é a temperatura dos gases em °C e  $t$  é o tempo em horas.

### 2.1.2.3 Curva de hidrocarbonetos

Nas ocorrências em que o compartimento apresenta materiais derivados de petróleo (hidrocarbonetos), o incêndio é mais agressivo. Assim, de forma diferente da curva de incêndio padrão, que evolui progressivamente ao longo do tempo, a curva de incêndio de hidrocarbonetos aumenta de maneira muito veloz, levando aproximadamente meia hora para chegar à temperatura de 1100°C, mantendo-se constante a partir desse momento. O EN 1991-1-2:2002 define a seguinte expressão para a curva de incêndio de hidrocarbonetos:

$$\theta_g = 20 + 1080 \left( 1 - 0,325e^{-0,167t} - 0,675e^{-2,5t} \right) \quad (2.3)$$

Na qual  $\theta_g$  é a temperatura dos gases em °C e  $t$  é o tempo em minutos.

Os materiais celulósicos e os hidrocarbonetos possuem, respectivamente, o poder calorífico em torno de 20 MJ/kg e 40 MJ/kg, de acordo com o anexo E da EN 1991-1-2:2002.

### 2.1.2.4 Curva de incêndio para elementos exteriores

As curvas de incêndio para elementos exteriores consideram algumas hipóteses no cenário do incêndio, como a carga de incêndio equivalente à de materiais celulósicos e a temperatura no exterior do compartimento onde se localiza a estrutura ser inferior àquela no interior do compartimento. Logo, conforme orientado pelo EN 1991-1-2:2002, essa curva é utilizada para elementos estruturais exteriores expostos ao fogo diretamente a partir do interior do compartimento em chamas por meio das aberturas, como janelas, nas fachadas do edifício. É representada pela seguinte equação:



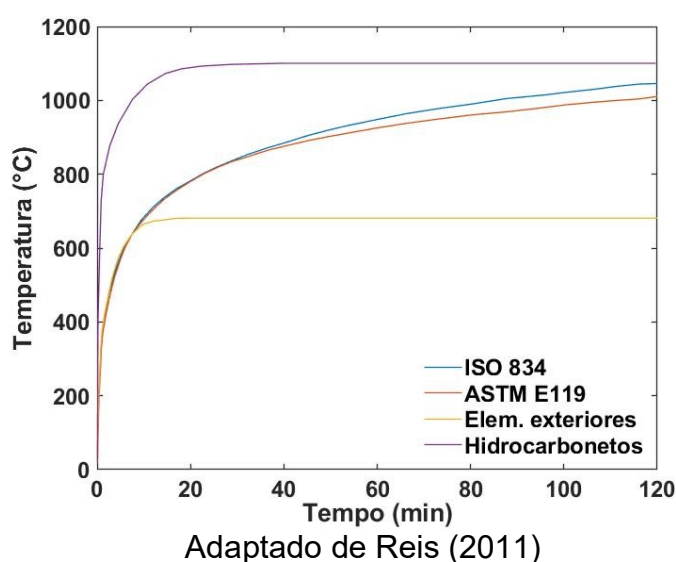
$$\theta_g = 20 + 600 \left( 1 - 0,687e^{-0,32t} - 0,313e^{-3,8t} \right) \quad (2.4)$$

Na qual  $\theta_g$  é a temperatura dos gases em °C e  $t$  é o tempo em minutos.

#### 2.1.2.5 Comparação entre curvas nominais

Na Figura 2.3 podemos verificar as diferenças das curvas nominais apresentadas. Observa-se que enquanto as curvas de incêndio-padrão e ASTM E119:2000 permanecem crescentes, as curvas de incêndio para elementos exteriores e de hidrocarboneto convergem para um patamar de temperatura.

Figura 2.3 - Comparação entre as curvas nominais de incêndio



## 2.2 PROCESSOS DE TRANSFERÊNCIA DE CALOR

A transferência de calor ocorre quando existe uma diferença de temperaturas em um dado sistema. O calor flui de uma região com a temperatura mais elevada para uma região de temperatura mais baixa. A condução é descrita como um mecanismo de transferência de calor essencialmente entre materiais sólidos por contato. A transferência de energia térmica ocorre com maior ou menor facilidade conforme a constituição atômica do material, a qual faz com que ele seja classificado como condutor ou isolante de calor. Os metais são bons condutores de calor.

Durante o incêndio, a transferência de calor por radiação é gerada pela diferença de temperatura entre as chamas e a superfície dos elementos estruturais e de compartimentação.

A transferência de calor por convecção é gerada pela diferença entre os gases do ambiente em chamas. Os gases quentes são menos densos e tendem a ocupar a atmosfera superior, enquanto os gases frios são mais densos e tendem a se movimentar para a atmosfera inferior do ambiente.

## 2.3 PROPRIEDADES DOS MATERIAIS EM ELEVADAS TEMPERATURAS

Quando os elementos estruturais são submetidos ao fogo, ocorre uma série de fenômenos físicos e químicos que alteram as propriedades dos materiais que os constituem. O aço, por exemplo, é formado por diferentes ligas metálicas que possuem comportamentos térmicos diversos em função da temperatura. O aquecimento das peças estruturais formadas pelo material aço acarreta uma modificação na fase de equilíbrio das ligas constituintes, alterando assim sua composição microscópica.

O aumento da temperatura, nos elementos de concreto, muda o estado físico da água intersticial, que provoca um aumento da poro-pressão, consequência da convecção forçada do vapor d'água para o exterior do material. Várias reações químicas acontecem entre a pasta de cimento e os agregados ao longo da elevação da temperatura do elemento estrutural de concreto.

Nesta seção será apresentado o comportamento das propriedades físicas, térmicas e mecânicas para os materiais aço e concreto, de acordo com os valores simplificados e as expressões analíticas recomendados nas normas brasileiras e também nas indicações dos códigos de normas europeu (Eurocodes).

### 2.3.1 Propriedades térmicas do aço em elevadas temperaturas

#### 2.3.1.1 Condutividade térmica

A condutividade térmica pode ser definida como a característica do material que expressa a velocidade com que o calor se propaga no seu interior.

De acordo com as normas ABNT NBR 14323:2013 e EN 1994-1-2:2005, a condutividade térmica do aço pode ser calculada por meio das seguintes expressões:

Para  $20^{\circ}\text{C} \leq \theta_s < 800^{\circ}\text{C}$

$$\lambda_{s,\theta} = 54 - 3,33 \times 10^{-2} \theta_s \quad (2.5a)$$

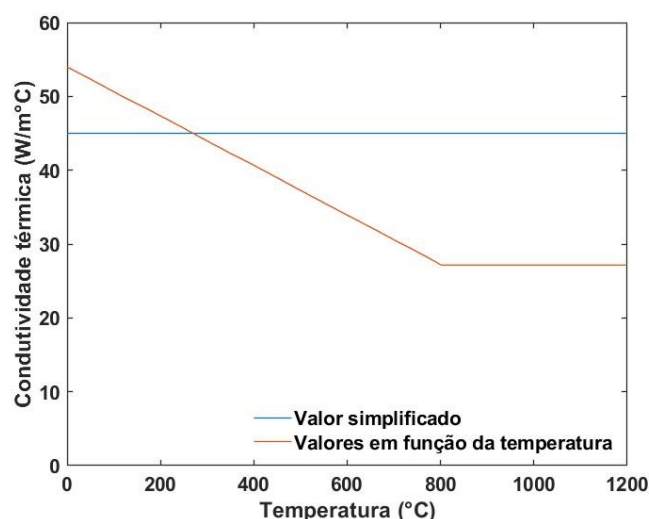
Para  $800^{\circ}\text{C} \leq \theta_s \leq 1200^{\circ}\text{C}$

$$\lambda_{s,\theta} = 27,3 \quad (2.5b)$$

Onde  $\lambda_{s,\theta}$  é a condutividade térmica do aço [ $\text{W/m}^{\circ}\text{C}$ ] em função da temperatura ( $\theta_s$ ).

As duas normas observam que de forma simplificada, o valor da condutividade térmica pode ser considerado constante, independente da temperatura do aço e igual a  $45 \text{ W/m}^{\circ}\text{C}$ . Apresenta-se, na Figura 2.4, a variação da condutividade térmica do aço em função da temperatura.

Figura 2.4 - Condutividade térmica do aço



Adaptado de ABNT NBR 14323:2013 e EN 1994-1-2:2005

Como se pode observar no gráfico da Figura 2.4, a condutividade térmica do aço é grande, contudo seu valor diminui com o aumento da temperatura, até chegar

a um comportamento constante para temperaturas acima de 800°C.

### 2.3.1.2 Calor específico

O calor específico é uma propriedade que representa a quantidade de energia calorífica necessária para elevar em um grau um metro cúbico de algum material, neste caso, o aço.

De acordo com as normas ABNT NBR 14323:2013 e EN 1994-1-2:2005, o calor específico do aço pode ser calculado por meio das seguintes expressões:

Para  $20^{\circ}\text{C} \leq \theta_s < 600^{\circ}\text{C}$

$$c_{s,\theta} = 425 + 7,73 \times 10^{-1} \theta_s - 1,69 \times 10^{-3} \theta_s^2 + 2,22 \times 10^{-6} \theta_s^3 \quad (2.6a)$$

Para  $600^{\circ}\text{C} \leq \theta_s < 735^{\circ}\text{C}$

$$c_{s,\theta} = 666 + 13002 / (738 - \theta_s) \quad (2.6b)$$

Para  $735^{\circ}\text{C} \leq \theta_s < 900^{\circ}\text{C}$

$$c_{s,\theta} = 545 + 17820 / (\theta_s - 731) \quad (2.6c)$$

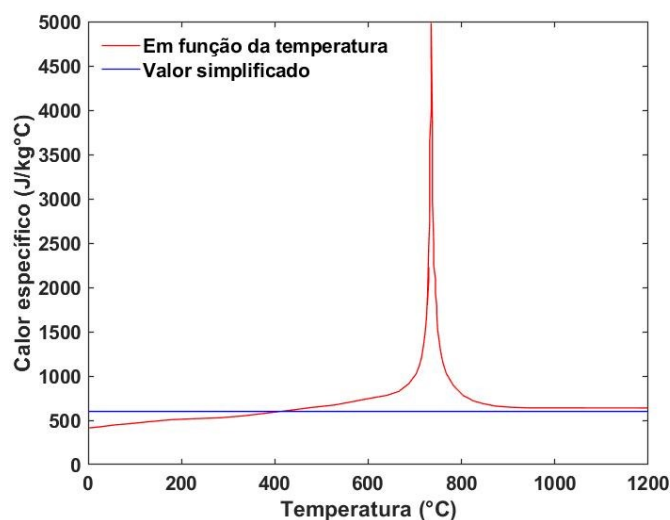
Para  $900^{\circ}\text{C} \leq \theta_s \leq 1200^{\circ}\text{C}$

$$c_{s,\theta} = 650 \quad (2.6d)$$

Onde  $c_{s,\theta}$  é o calor específico do aço [J/kg°C] em função da temperatura ( $\theta_s$ ).

O valor do calor específico, de forma simplificada, pode ser considerado independente da temperatura do aço. As normas ABNT NBR 14323:2013 e EN 1994-1-2:2005 recomendam o valor de 600 J/kg°C. Apresenta-se, na Figura 2.5, a variação do calor específico do aço em função da temperatura.

Figura 2.5 - Calor específico do aço



Adaptado de ABNT NBR 14323:2013 e EN 1994-1-2:2005

Como se pode observar no gráfico da Figura 2.5, o calor específico é crescente inicialmente, atinge um pico entre 700°C e 800°C e decresce até um valor constante.

### 2.3.2 Propriedades físicas do aço em elevadas temperaturas

#### 2.3.2.1 Massa específica

O aço é composto de uma microestrutura bem definida e estável, mesmo em temperaturas elevadas. Com isso, a sua massa específica ( $\rho_{s,0}$ ) é considerada independente da temperatura, constante, de valor igual a 7850 kg/m<sup>3</sup>.

#### 2.3.2.2 Deformação térmica específica

A ação da temperatura provoca deformação nos materiais. As normas ABNT NBR 14323:2013, EN 1992-1-2:2004 e EN 1994-1-2:2005 recomendam as seguintes expressões para a deformação térmica específica do aço:

Para  $20^{\circ}\text{C} \leq \theta_s \leq 750^{\circ}\text{C}$

$$\frac{\Delta l}{l} = -2,416 \times 10^{-4} + 1,2 \times 10^{-5} \theta_s + 0,4 \times 10^{-8} \theta_s^2 \quad (2.7a)$$

Para  $750^{\circ}\text{C} < \theta_s \leq 860^{\circ}\text{C}$

$$\frac{\Delta l}{l} = 1,1 \times 10^{-2} \quad (2.7b)$$

Para  $860^{\circ}\text{C} < \theta_s \leq 1200^{\circ}\text{C}$

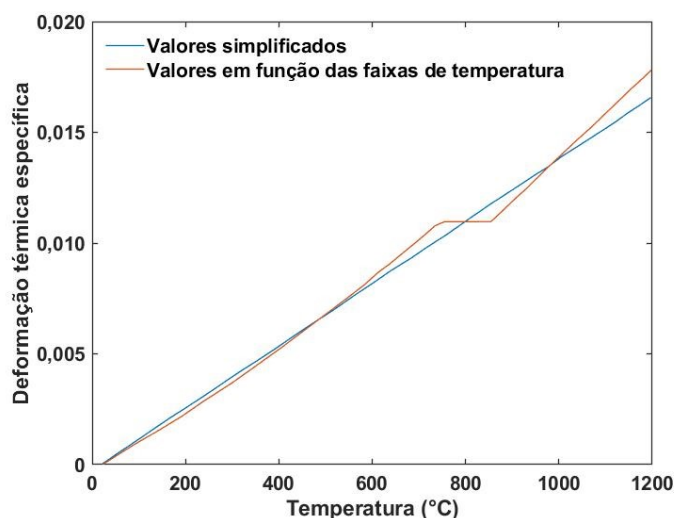
$$\frac{\Delta l}{l} = -6,2 \times 10^{-3} + 2 \times 10^{-5} \theta_s \quad (2.7c)$$

Em que  $\Delta l$  é a expansão térmica do aço devido à temperatura ( $\theta_s$ ) e  $l$  é o comprimento inicial da peça de aço.

O valor da deformação térmica específica também pode ser considerado pela Equação 2.8, de forma simplificada, de acordo com as normas ABNT NBR 14323:2013 e EN 1994-1-2:2005. A Figura 2.6 ilustra a variação da deformação térmica específica do aço em função da temperatura.

$$\frac{\Delta l}{l} = 14 \times 10^{-6} (\theta_s - 20^{\circ}\text{C}) \quad (2.8)$$

Figura 2.6 - Deformação térmica específica do aço



Adaptado de ABNT NBR 14323:2013, EN 1992-1-2:2004 e EN 1994-1-2:2005

### 2.3.3 Propriedades mecânicas do aço em elevadas temperaturas

A Tabela 2.2 e a Figura 2.7 apresentam os fatores de redução à temperatura elevada, relativos aos valores à temperatura ambiente (20°C), para a resistência ao escoamento, o limite de proporcionalidade e o módulo de elasticidade dos aços laminados, respectivamente definidos pelas seguintes expressões:

$$k_{sy,\theta} = \frac{f_{sy,\theta}}{f_{sy}} \quad (2.9a)$$

$$k_{sp,\theta} = \frac{f_{sp,\theta}}{f_{sy}} \quad (2.9b)$$

$$k_{sE,\theta} = \frac{E_{s,\theta}}{E_s} \quad (2.9c)$$

Onde  $f_{sy,\theta}$  é a resistência ao escoamento dos aços laminados a uma temperatura  $\theta_s$ ,  $f_{sy}$  é a resistência ao escoamento do aço a temperatura ambiente,  $f_{sp,\theta}$  é o limite de proporcionalidade a uma temperatura  $\theta_s$ ,  $E_{s,\theta}$  é o módulo de elasticidade a uma

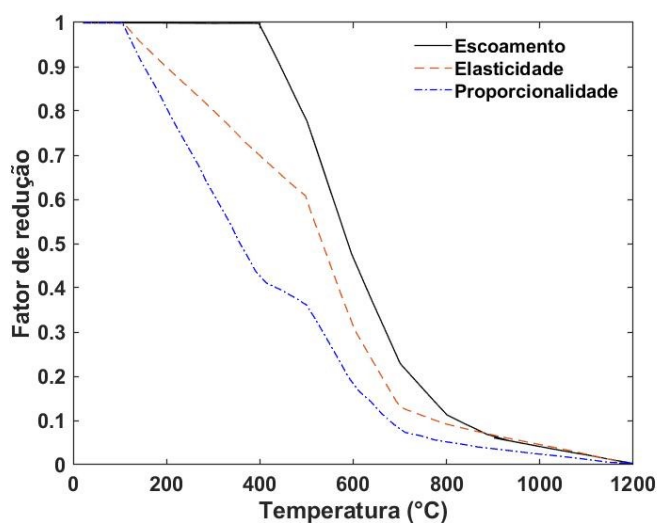
temperatura  $\theta_s$  e  $E_s$  é o módulo de elasticidade à temperatura ambiente[ MPa].

Tabela 2.2 - Valores dos fatores de redução em função da temperatura

$\theta_s$ (°C)	Fator de redução para a resistência ao escoamento	Fator de redução para o limite de proporcionalidade	Fator de redução para o módulo de elasticidade
	$k_{sy,\theta}$	$k_{sp,\theta}$	$k_{sE,\theta}$
20	1,0000	1,0000	1,0000
100	1,0000	1,0000	1,0000
200	1,0000	0,8070	0,9000
300	1,0000	0,6130	0,8000
400	1,0000	0,4200	0,7000
500	0,7800	0,3600	0,6000
600	0,4700	0,1800	0,3100
700	0,2300	0,0750	0,3100
800	0,1100	0,0500	0,0900
900	0,0600	0,0375	0,0675
1000	0,0400	0,0250	0,0450
1100	0,0200	0,0125	0,0225
1200	0,0000	0,0000	0,0000

Adaptado de EN 1993-1-2:2005

Figura 2.7 - Fatores de redução da relação tensão-deformação dos aços laminados



Adaptado de EN 1993-1-2:2005



A relação tensão-deformação do aço em temperatura elevada, para taxas de aquecimento entre 2 e 50°C/min, é dada pelas expressões apresentadas na Tabela 2.3.

Tabela 2.3 - Relação tensão-deformação do aço em temperaturas elevadas

Deformação	Tensão	Módulo de elasticidade
$\varepsilon_s \leq \varepsilon_{sp,\theta}$	$\varepsilon_s E_{s,\theta}$	$E_{s,\theta}$
$\varepsilon_{sp,\theta} \leq \varepsilon_s \leq \varepsilon_{sy,\theta}$	$f_{sp,\theta} - c + (b/a) \left[ a^2 - (\varepsilon_{sy,\theta} - \varepsilon_s)^2 \right]^{0,5}$	$\frac{b(\varepsilon_{sy,\theta} - \varepsilon_s)}{a \left[ a^2 - (\varepsilon_{sy,\theta} - \varepsilon_s)^2 \right]^{0,5}}$
$\varepsilon_{sy,\theta} \leq \varepsilon_s \leq \varepsilon_{st,\theta}$	$f_{sy,\theta}$	0
$\varepsilon_{st,\theta} \leq \varepsilon_s \leq \varepsilon_{su,\theta}$	$f_{sy,\theta} \left[ 1 - \frac{(\varepsilon_s - \varepsilon_{st,\theta})}{\varepsilon_{su,\theta} - \varepsilon_{st,\theta}} \right]^{0,5}$	-
$\varepsilon_s = \varepsilon_{su,\theta}$	0	-
<b>Parâmetros</b>	$\varepsilon_{sp,\theta} = \frac{f_{sp,\theta}}{E_{s,\theta}} \quad \varepsilon_{sy,\theta} = 0,02 \quad \varepsilon_{st,\theta} = 0,15 \quad \varepsilon_{su,\theta} = 0,20$	
<b>Funções</b>	$a^2 = (\varepsilon_{sy,\theta} - \varepsilon_{sp,\theta})(\varepsilon_{sy,\theta} - \varepsilon_{sp,\theta} + c/E_{s,\theta})$ $b^2 = c(\varepsilon_{sy,\theta} - \varepsilon_{sp,\theta})E_{s,\theta} + c^2$ $c = \frac{(f_{sy,\theta} - f_{sp,\theta})^2}{(\varepsilon_{sy,\theta} - \varepsilon_{sp,\theta})E_{s,\theta} - 2(f_{sy,\theta} - f_{sp,\theta})}$	

Adaptado de EN 1993-1-2:2005

### 2.3.4 Propriedades térmicas do concreto em elevadas temperaturas

#### 2.3.4.1 Condutividade térmica

De acordo com o Boletim FIB 38 (2007), mesmo que a condutividade térmica do concreto dependa de todos os constituintes de sua massa, ela é determinada, sobretudo, pelo tipo de agregado utilizado, pois esse componente representa cerca de 60% a 80% do volume do concreto. Outro parâmetro importante é a umidade do concreto, já que a condutividade térmica da água, embora de pequeno valor, é bem

maior que a do ar. A Tabela 2.4 apresenta valores de condutividade térmica de alguns materiais referentes ao concreto.

Tabela 2.4 - Condutividade térmica dos materiais referentes ao concreto à temperatura ambiente

<b>Material</b>	$\lambda_{c,\theta}$
Agregados	0,7 – 4,2
Concretos saturados	1,0 – 3,6
Pasta de cimento saturada endurecida	1,1 – 1,6
Água	0,515
Ar	0,0034

Adaptado de FIB 38 (2007)

A ABNT NBR 15200:2012 recomenda a Equação 2.10a para a condutividade térmica do concreto com agregado silicoso, com temperatura variando entre 20°C e 1200°C. Para o mesmo intervalo de temperatura, a EN 1992-1-2:2004 especifica a Equação 2.10b para o concreto produzido com agregado calcário.

$$\lambda_{c,\theta} = 2 - 0,2451 \frac{\theta_c}{100} + 0,0107 \left( \frac{\theta_c}{100} \right)^2 \quad (2.10a)$$

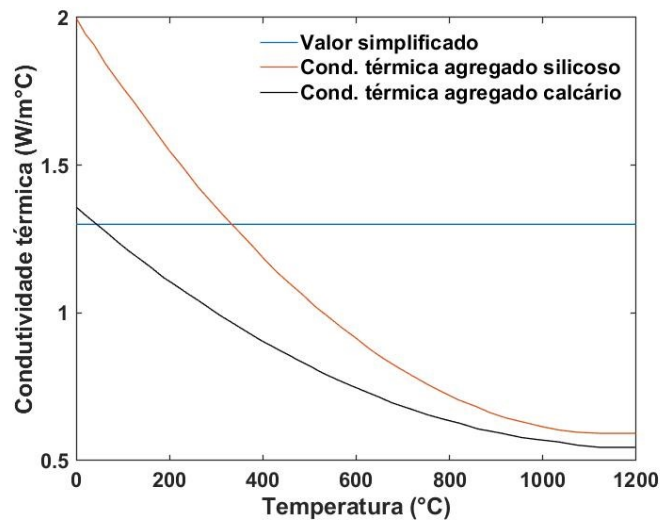
$$\lambda_{c,\theta} = 1,36 - 0,136 \frac{\theta_c}{100} + 0,0057 \left( \frac{\theta_c}{100} \right)^2 \quad (2.10b)$$

Onde  $\lambda_{c,\theta}$  é a condutividade térmica do concreto [W/(m°C)] em função da temperatura  $\theta_c$ .

A ABNT NBR 15200:2012, de forma simplificada, assume que a relação entre a condutividade térmica do concreto e a temperatura pode ser considerada constante e com valor igual a 1,3 W/m°C.

A Figura 2.8 representa graficamente as expressões indicadas anteriormente para o concreto de agregados silicosos e calcários. Pode-se observar que a condutividade térmica do concreto é reduzida com o aumento da temperatura.

Figura 2.8 - Condutividade térmica do concreto em função da temperatura



Adaptado de ABNT NBR 15200:2012 e EN 1992-1-2:2004

#### 2.3.4.2 Calor específico

O calor específico, no concreto, é função do teor da umidade e da temperatura, sendo independente do tipo de agregado utilizado. Para a determinação do calor específico do concreto, a EN 1992-1-2:2004 recomenda as seguintes expressões:

Para  $20^{\circ}\text{C} \leq \theta_c \leq 100^{\circ}\text{C}$

$$c_{c,\theta} = 900 \quad (2.11a)$$

Para  $100^{\circ}\text{C} < \theta_c \leq 115^{\circ}\text{C}$

$$c_{c,\theta} = c_{\text{pico}} \quad (2.11b)$$

Para  $115^{\circ}\text{C} < \theta_c \leq 200^{\circ}\text{C}$

$$c_{c,\theta} = c_{\text{pico}} - \frac{(c_{\text{pico}} - 1000)}{85} \quad (2.11c)$$

Para  $200^{\circ}\text{C} < \theta_c \leq 400^{\circ}\text{C}$

$$c_{c,\theta} = 1000 + \frac{(\theta_c - 200)}{2} \quad (2.11d)$$

Para  $400^{\circ}\text{C} < \theta_c \leq 1200^{\circ}\text{C}$

$$c_{c,\theta} = 1100 \quad (2.11e)$$

Em que  $c_{c,\theta}$  é o calor específico do concreto em função da temperatura,  $c_{\text{pico}}$  é o valor de pico do calor específico em função da umidade relativa do concreto, os dois dados em  $[\text{J}/\text{kg}^{\circ}\text{C}]$ .

Os valores de pico para o calor específico do concreto são fornecidos na EN 1992-1-2:2004, para umidades de 0%; 1,5% e 3,0%. Para os demais valores pode-se interpolar e extrapolar linearmente, conforme é apresentado na Tabela 2.5.

Tabela 2.5 - Valores de pico para o calor específico do concreto situado entre  $100^{\circ}\text{C}$  e  $200^{\circ}\text{C}$

$U(\%)$ em peso de concreto	$c_{\text{pico}}$
0	900
1,5	1470
2,0	1875
3,0	2020
4,0	2750
10,0	5600

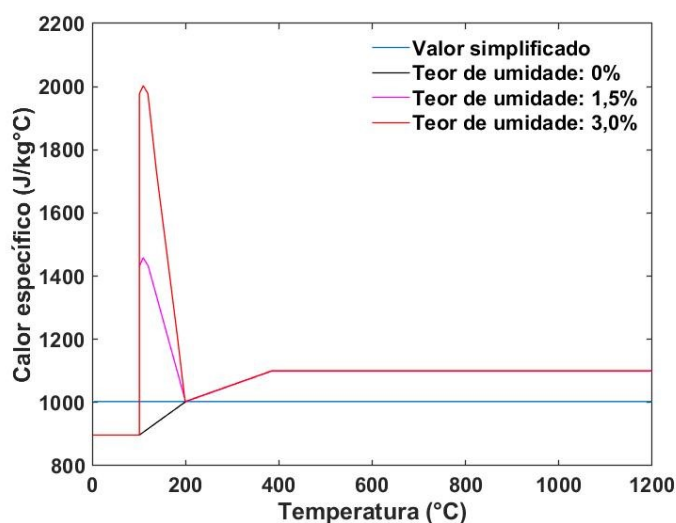
Adaptado de EN 1992-1-2:2004

Segundo Costa (2008), o teor de umidade  $U \leq 4,0\%$  em peso possui maior representatividade nas estruturas de concreto armado, enquanto  $4,0\% \leq U \leq 10,0\%$  nas estruturas mistas de aço e concreto. Ainda de acordo com o pesquisador, os elementos estruturais de concreto armado dentro de ambientes internos às edificações e secos tendem a ter  $U \leq 1,5\%$  e  $U \leq 3,0\%$  para ambientes externos.

De maneira simplificada, a ABNT NBR 15200:2012 recomenda o valor constante de 1000 J/kg°C para o calor específico do concreto, independente da variação de temperatura.

Na Figura 2.9 é apresentada a variação do calor específico do concreto com a temperatura para os três teores de umidade citados na EN 1992-1-2:2004.

Figura 2.9 - Calor específico do concreto em função da temperatura e umidade



Adaptado de ABNT NBR 15200:2012 e EN 1992-1-2:2004

### 2.3.5 Propriedades físicas do concreto em elevadas temperaturas

#### 2.3.5.1 Deformação térmica específica

Assim como o aço, o concreto também se deforma com a variação de temperatura. Para o concreto de densidade normal com agregado silicoso, as normas ABNT NBR 15200:2012 e EN 1992-1-2:2004 indicam as seguintes equações:

Para  $20^{\circ}\text{C} \leq \theta_c \leq 700^{\circ}\text{C}$

$$\frac{\Delta l}{l} = -1,8 \times 10^{-4} + 9 \times 10^{-6} \theta_c + 2,3 \times 10^{-11} \theta_c^3 \quad (2.12a)$$

Para  $700^{\circ}\text{C} < \theta_c \leq 1200^{\circ}\text{C}$

$$\frac{\Delta l}{l} = 1,4 \times 10^{-2} \quad (2.12b)$$

A EN 1992-1-2:2004 recomenda para o concreto de densidade normal com agregado calcário, e mesmo intervalo de temperatura, as seguintes expressões:

Para  $20^{\circ}\text{C} \leq \theta_c \leq 805^{\circ}\text{C}$

$$\frac{\Delta l}{l} = -1,2 \times 10^{-4} + 6 \times 10^{-6} \theta_c + 1,4 \times 10^{-11} \theta_c^3 \quad (2.13a)$$

Para  $805^{\circ}\text{C} < \theta_c \leq 1200^{\circ}\text{C}$

$$\frac{\Delta l}{l} = 1,2 \times 10^{-2} \quad (2.13b)$$

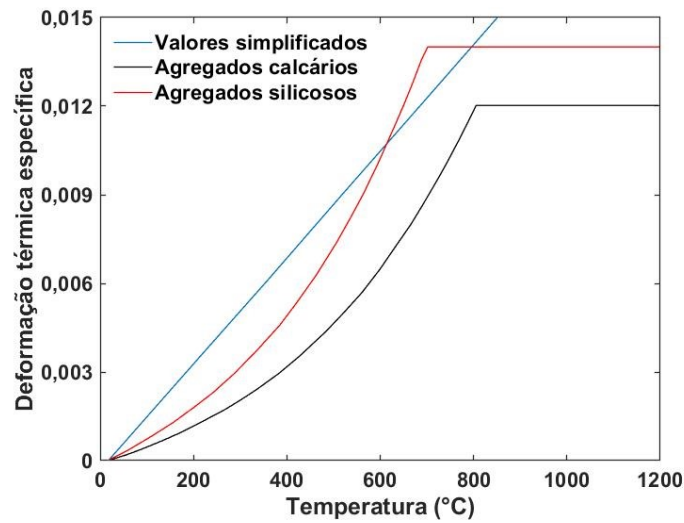
Onde  $\Delta l$  é a expansão térmica do concreto devido à temperatura ( $\theta_c$ ) e  $l$  é o comprimento inicial da peça de concreto.

A ABNT NBR 15200:2012 recomenda, de forma simplificada, a Equação 2.14 para a determinação da deformação térmica específica do concreto, independentemente das faixas de temperatura e do tipo de agregado.

$$\frac{\Delta l}{l} = 18 \times 10^{-3} (\theta_c - 20) \quad (2.14)$$

A Figura 2.10 ilustra a variação da deformação térmica específica do concreto com a temperatura.

Figura 2.10 - Deformação térmica específica do concreto em função da temperatura



Adaptado de ABNT NBR 15200:2012 e EN 1992-1-2:2004

### 2.3.5.2 Massa específica

A variação da massa específica do concreto com a temperatura pode ser determinada pelas seguintes equações:

Para  $20^{\circ}\text{C} \leq \theta_c \leq 115^{\circ}\text{C}$

$$\rho_{c,\theta} = \rho(20^{\circ}\text{C}) \quad (2.15a)$$

Para  $115^{\circ}\text{C} < \theta_c \leq 200^{\circ}\text{C}$

$$\rho_{c,\theta} = \rho(20^{\circ}\text{C}) \left[ 1 - 0,02 \left( \frac{\theta_c - 115}{85} \right) \right] \quad (2.15b)$$

Para  $200^{\circ}\text{C} < \theta_c \leq 400^{\circ}\text{C}$

$$\rho_{c,\theta} = \rho(20^{\circ}\text{C}) \left[ 0,98 - 0,03 \left( \frac{\theta_c - 200}{200} \right) \right] \quad (2.15c)$$

Para  $400^{\circ}\text{C} < \theta_c \leq 1200^{\circ}\text{C}$

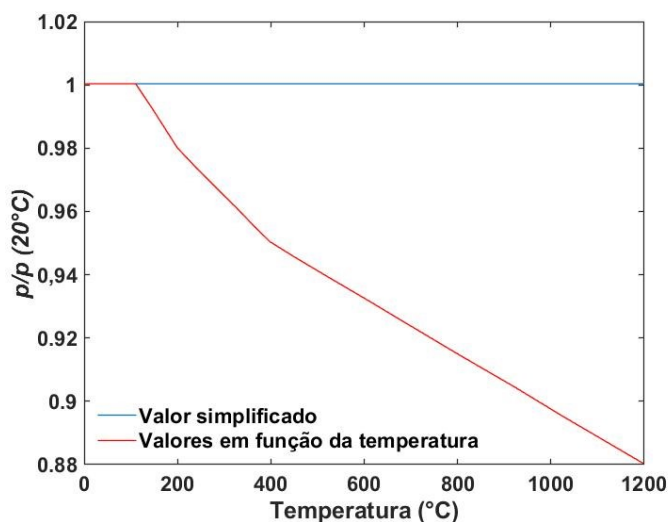
$$\rho_{c,\theta} = \rho(20^{\circ}\text{C}) \left[ 0,95 - 0,07 \left( \frac{\theta_c - 200}{800} \right) \right] \quad (2.15d)$$

Onde  $\rho_{c,\theta}$  é a massa específica do concreto em função da temperatura  $[\text{kg}/\text{m}^3]$ .

Para os projetos de estruturas de concreto submetidas à temperatura ambiente, as normas ABNT NBR 6120:1980 e ABNT NBR6118:2014 indicam os valores de massa específica  $\rho_{c,\theta} = 2400 \text{ kg}/\text{m}^3$  para concreto simples e  $\rho_{c,\theta} = 2500 \text{ kg}/\text{m}^3$  para concreto armado.

A variação da massa específica relativa do concreto em função da temperatura é apresentada na Figura 2.11.

Figura 2.11 - Massa específica relativa do concreto em função da temperatura



Adaptado de ABNT NBR 15200:2012 e EN 1992-1-2:2004



### 2.3.6 Propriedades mecânicas do concreto em elevadas temperaturas

#### 2.3.6.1 Resistência à compressão

A norma EN 1992-1-2:2004 e a norma ABNT NBR 15200:2012 recomendam a Equação 2.16 para determinar a resistência à compressão do concreto em função da temperatura.

$$f_{c,\theta} = k_{c,\theta} f_{ck} \quad (2.16)$$

Em que  $k_{c,\theta}$  é o coeficiente de redução da resistência à compressão,  $f_{c,\theta}$  e  $f_{ck}$  são a resistência à compressão do concreto em função da temperatura e a resistência característica à compressão do concreto em temperatura ambiente, ambas em MPa.

Os valores do coeficiente de redução da resistência à compressão em função da temperatura são apresentados na Tabela 2.6 para os concretos produzidos com agregados silicosos e calcários, de acordo com as normas citadas.

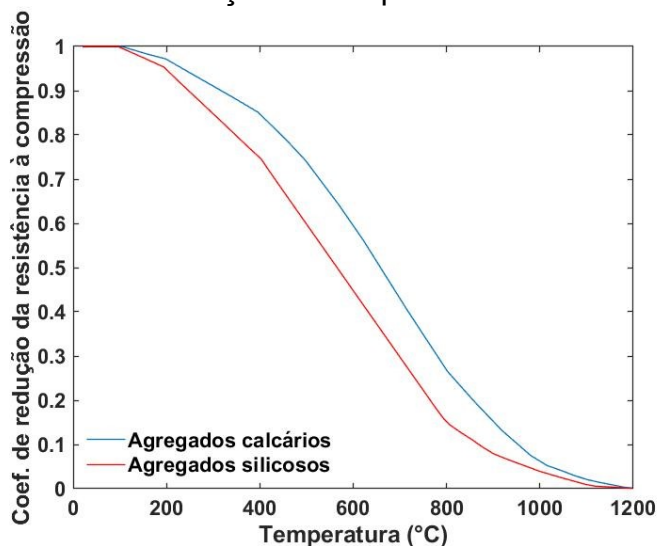
Tabela 2.6 - Coeficiente de redução da resistência à compressão do concreto em função da temperatura

$\theta_c$ (°C)	<b>Agregados silicosos   Agregados calcários</b>	
	$f_{c,\theta}/f_{ck}$	$f_{c,\theta}/f_{ck}$
20	1,00	1,00
100	1,00	1,00
200	0,95	0,97
300	0,85	0,91
400	0,75	0,85
500	0,60	0,74
600	0,45	0,60
700	0,30	0,43
800	0,15	0,27
900	0,08	0,15
1000	0,04	0,06
1100	0,01	0,02
1200	0,00	0,00

Adaptado de ABNT NBR 15200:2012 e EN 1992-1-2:2004

A Figura 2.12 mostra a variação do coeficiente de redução da resistência à compressão do concreto com o crescimento da temperatura.

Figura 2.12 - Coeficiente de redução da resistência à compressão do concreto em função da temperatura



Adaptado de ABNT NBR 15200:2012 e EN 1992-1-2:2004

### 2.3.6.2 Resistência à tração

A resistência à tração do concreto em função da temperatura não é abordada pela ABNT NBR 15200:2012, contudo, a EN 1992-1-2:2004 indica a seguinte expressão para o cálculo dessa propriedade:

$$f_{ct,\theta} = k_{ct,\theta} f_{ctk} \quad (2.17)$$

Em que  $k_{ct,\theta}$  é o coeficiente de redução da resistência à tração,  $f_{ct,\theta}$  e  $f_{ctk}$  são a resistência à tração do concreto em função da temperatura e a resistência característica à tração do concreto em temperatura ambiente, ambas em MPa.

A EN 1992-1-2:2004 recomenda as seguintes equações para obter o coeficiente de redução do concreto sujeito à tração, em função da temperatura, independentemente do tipo de agregado.

Para  $20^{\circ}\text{C} \leq \theta_c \leq 100^{\circ}\text{C}$

$$k_{ct,\theta} = 1,0 \quad (2.18a)$$

Para  $100^{\circ}\text{C} < \theta_c \leq 600^{\circ}\text{C}$

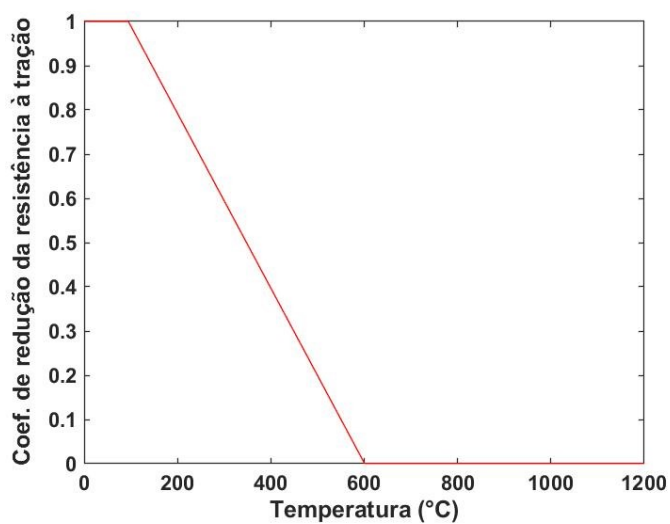
$$k_{ct,\theta} = 1,0 - \frac{(\theta_c - 100)}{500} \quad (2.18b)$$

Para  $600^{\circ}\text{C} < \theta_c \leq 1200^{\circ}\text{C}$

$$k_{ct,\theta} = 0 \quad (2.18c)$$

A Figura 2.13 apresenta a representação gráfica da variação do coeficiente de redução da resistência à tração do concreto, em função da temperatura.

Figura 2.13 - Coeficiente de redução da resistência à tração do concreto em função da temperatura



Adaptado de EN 1992-1-2:2004

### 2.3.6.3 Módulo de elasticidade

O módulo de elasticidade do concreto em função da temperatura pode ser calculado por meio da seguinte equação:

$$E_{c,\theta} = k_{cE,\theta} E_c \quad (2.19)$$

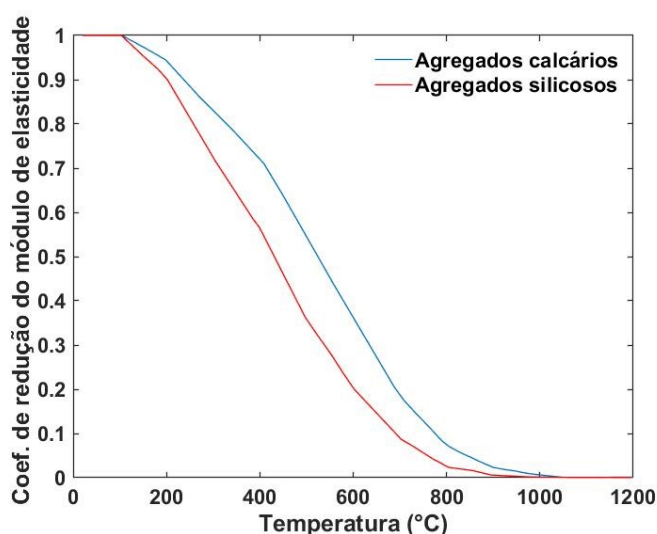
Onde  $k_{cE,\theta}$  é o coeficiente de redução do módulo de elasticidade,  $E_{c,\theta}$  e  $E_c$  são o módulo de elasticidade do concreto em função da temperatura e o módulo de elasticidade do concreto à temperatura ambiente, os dois em MPa.

De acordo com o EN 1992-1-2:2004 o coeficiente de redução do módulo de elasticidade pode ser obtido através da Equação 2.20.

$$k_{cE,\theta} = k_{c,\theta}^2 \quad (2.20)$$

Em que  $k_{c,\theta}$  é o coeficiente de redução da resistência à compressão em função da temperatura. A variação do coeficiente de redução do módulo de elasticidade com a temperatura é apresentada na Figura 2.14, para concretos de agregados silicosos e calcários.

Figura 2.14 - Coeficiente de redução do módulo de elasticidade do concreto em função da temperatura



Adaptado de EN 1992-1-2:2004

### 2.3.7 Materiais de proteção contra o fogo

Os valores das propriedades mecânicas do aço e do concreto, como mostrado nos tópicos anteriores, sofrem reduções com o aumento da temperatura, possibilitando o colapso prematuro do elemento estrutural e com isso causar danos patrimoniais e perda de vidas humanas.

Para evitar a ruína, em situação de incêndio, pode-se dimensionar a estrutura para resistir às altas temperaturas, como aumentar a seção dos pilares e vigas de um edifício para obter a segurança desejada. Contudo, alguns recursos podem ser empregados para que elementos estruturais de aço e de concreto elevem sua resistência no cenário de incêndio. Um dos recursos são os materiais de revestimento, que ao envolverem as peças estruturais, atuam como proteção contra o fogo. Outro meio, utilizado para elementos de concreto, se dá por adição de certos materiais durante o processo de fabricação, que melhoram a sua resistência em temperaturas elevadas, como é o caso das fibras de polipropileno. Esses artifícios possuem o objetivo, sobretudo, de evitar o colapso prematuro do elemento estrutural.

Os materiais utilizados para proteção térmica, de acordo com Mendes (2004), devem apresentar as seguintes características:

- Baixa massa específica;
- Baixa condutividade térmica;
- Alto calor específico;
- Adequada resistência mecânica;
- Garantia de integridade durante a evolução do incêndio;
- Custo compatível com a realidade da obra.

Serão apresentados a seguir alguns materiais de proteção térmica para elementos estruturais de aço e concreto.

### 2.3.7.1 Argamassas à base de vermiculita

As argamassas fabricadas com vermiculita são constituídas de agregados leves à base desse mineral, cimento ou gesso e aglomerantes minerais. A vermiculita, pertencente ao grupo dos minerais micáceos, é considerada incombustível (dificulta a propagação das chamas) e possui ponto de fusão em torno de 1370°C. A vermiculita também pode ser empregada como agregado fino na produção de concretos, no lugar da areia, fornecendo características como leveza, isolamento acústico e térmico e grande resistência mecânica. A aplicação pode ser realizada por meio de jateamento ou uso de espátulas.

Os valores das propriedades térmicas da argamassa de vermiculita e, também, de sua massa específica são apresentados na Tabela 2.7.

Tabela 2.7 - Propriedades da argamassa de vermiculita

<b>Propriedade</b>	<b>Valor</b>	<b>Unidade</b>
Condutividade térmica	0,12	W/m°C
Calor específico	1200	J/kg°C
Massa específica	350	Kg/m <sup>3</sup>

Adaptado de Rigobello (2007)

### 2.3.7.2 Blaze Shield II

O Blaze Shield II é uma argamassa projetada, desenvolvida pela Isolatek International. A sua massa específica pode ser considerada, de forma simplificada, com o valor constante de 240 kg/m<sup>3</sup>. O calor específico desse produto varia de acordo com os dados da Tabela 2.8.

Tabela 2.8 - Calor específico da argamassa Blaze Shield II em função da temperatura

Temperatura (°C)	Calor específico [J/kg°C]
96	2093
104	837
150	1675
200	1770
400	2148
482	2303
600	2343
800	2411
1093	2512
1200	2512

Adaptado de Silva (2001)

Pode-se determinar o calor específico do Blaze Shield II, por meio do ajuste por regressão numérica dos valores da Tabela 2.8, pelas equações a seguir:

Para  $20^{\circ}\text{C} \leq \theta_{\text{BS}} \leq 100^{\circ}\text{C}$

$$c_{\text{BS},\theta} = 2093 \quad (2.21a)$$

Para  $100^{\circ}\text{C} < \theta_{\text{BS}} < 104^{\circ}\text{C}$

$$c_{\text{BS},\theta} = -1,74444\theta_{\text{BS}}^3 + 497,167\theta_{\text{BS}}^2 - 47229,1\theta_{\text{BS}} + 1497570 \quad (2.21b)$$

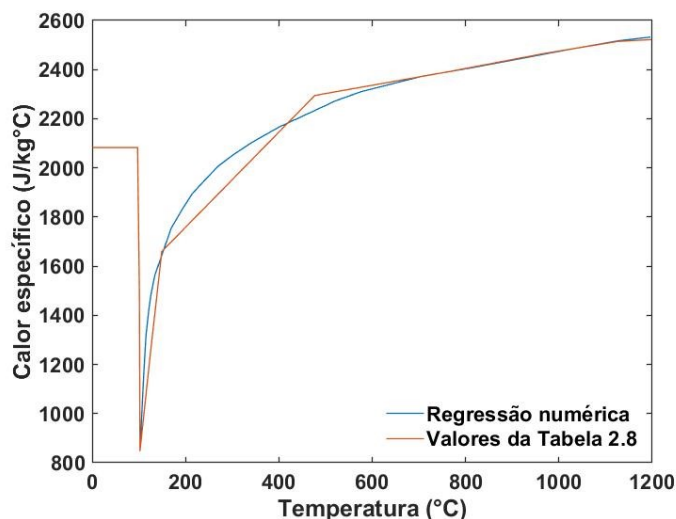
Para  $104^{\circ}\text{C} \leq \theta_{\text{BS}} \leq 1200^{\circ}\text{C}$

$$c_{\text{BS},\theta} = 287,510 \log_e (\theta_{\text{BS}} - 101,1) + 530,6 \quad (2.21c)$$

Onde  $c_{\text{BS},\theta}$  é o calor específico da argamassa Blaze Shield II [J/kg°C] em função da temperatura ( $\theta_{\text{BS}}$ ).

A Figura 2.15 demonstra a variação do calor específico com a temperatura, para os resultados da Tabela 2.8 e para o ajuste dos valores por regressão.

Figura 2.15 - Calor específico da Blaze Shield II em função da temperatura



Adaptado de Silva (2001)

Os valores da condutividade térmica da argamassa Blaze Shield II, em função da temperatura, são apresentados na Tabela 2.9.

Tabela 2.9 - Condutividade térmica da argamassa Blaze Shield II com a temperatura

Temperatura (°C)	Condutividade térmica [W/m°C]
100	0,061
200	0,080
400	0,112
482	0,147
600	0,173
1093	0,208

Adaptado de Silva (2001)

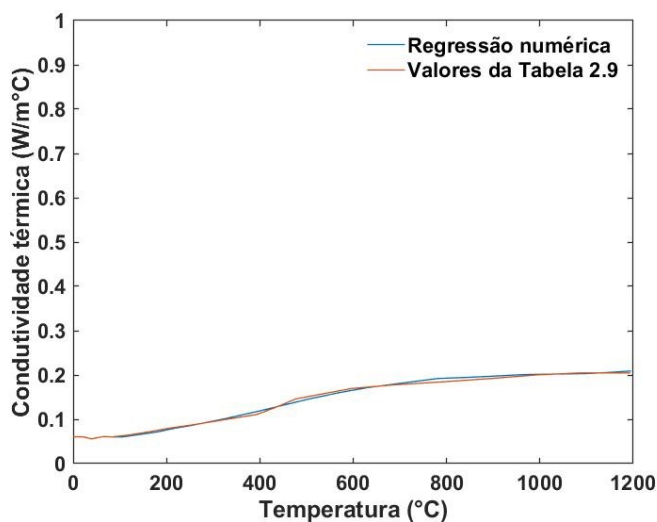
Ao ajustar os valores da Tabela 2.9 por regressão, pode-se determinar a condutividade térmica pela Equação 2.22. A representação gráfica da variação da condutividade térmica em função da temperatura é apresentada na Figura 2.16.

$$\lambda_{BS,\theta} = 4,143 \times 10^{-13} \theta_{BS}^4 - 1,164 \times 10^{-9} \theta_{BS}^3 + 9,666 \times 10^{-7} \theta_{BS}^2 - 7,547 \times 10^{-5} \theta_{BS} + 0,062 \quad (2.22)$$



Em que  $\lambda_{BS,\theta}$  é a condutividade térmica da argamassa Blaze Shield [W/m°C] em função da temperatura ( $\theta_{BS}$ ).

Figura 2.16 - Condutividade térmica da Blaze Shield II em função da temperatura



Adaptado de Silva (2001)

### 2.3.7.3 Mantas

As mantas são uma medida de proteção adequada para edificações em uso. Consistem em materiais compostos por aglomerados de fibra cerâmica, lã de rocha ou algum outro material fibroso. São aplicadas na superfície do elemento, envolvendo-o.

De acordo com Mendes (2004), possuem baixa massa específica, em torno de 64 kg/m³, condutividade térmica de 0,087 W/mK e calor específico de 1130 J/kg°C. Contudo, para fins de estudos, é recomendado obter esses valores em função do material que constitui a manta.

### 2.3.7.4 Tinta intumescente

A tinta intumescente é um produto de proteção térmica próprio para proteção passiva, que permanece inativo no elemento estrutural, como tinta usual, até que seja exposta a temperatura superior a 200 °C. Quando a tinta atinge esse valor de temperatura, origina-se um processo de expansão volumétrica e forma-se uma

espuma rígida de proteção, devido às resinas componentes da tinta. Esse processo retarda a elevação rápida de temperatura nas peças estruturais. A aplicação se dá por meio de spray, pincel ou rolo, várias demãos, até chegar à espessura adequada para a proteção. Segundo Fakury et al. (2016), tal tinta, aplicada em película de 0,25 a 0,6 mm de espessura, ao ser submetida ao calor, tem sua espessura aumentada entre 20 e 30 vezes. Além disso, permite que sobre ela seja aplicada uma pintura de acabamento, de modo que a estrutura fique com a cor final desejada.

### 2.3.8 Spalling

O lascamento ou *spalling* é um fenômeno natural que ocorre nas estruturas de concreto submetidas a temperaturas elevadas. De forma resumida, os vapores originados do aquecimento do concreto que possui certa umidade interna não são liberados com facilidade. Isto faz com que o concreto se desintegre, podendo até explodir.

Soares (2003) enumera outras razões para que este fenômeno ocorra:

- O valor do coeficiente de dilatação térmica da pasta de cimento é o dobro do valor do coeficiente de dilatação térmica dos agregados. Logo, os materiais constituintes do concreto dilatam-se de maneira diferente, inicializando um processo de desagregação;
- A água livre e a água de hidratação do cimento evaporaram-se, criando locais com elevada pressão interna;
- Apesar do aço e do concreto possuírem praticamente o mesmo coeficiente de dilatação térmica, o aço dilata-se mais cedo por apresentar maior coeficiente de condutividade térmica que o concreto. Portanto, surgem tensões entre os dois materiais e por consequência a perda de aderência, o que pode favorecer o *spalling*.

O principal risco deste fenômeno é a perda da camada de cobrimento da armadura principal, o aço eleva sua temperatura mais rapidamente e, conseqüentemente, diminui sua resistência. Observa-se que pode haver, também, a

perda da estabilidade da estrutura, pois o *spalling* reduz a seção transversal dos elementos estruturais.

De acordo com Caldas (2008), o lascamento das superfícies dos elementos estruturais pode comprometer as estruturas de concreto de alta resistência, logo no início do incêndio. O concreto de alta resistência é mais perigoso quando submetido a um incêndio de grandes proporções porque este tipo de concreto permite a construção de estruturas mais esbeltas, onde o calor se propaga mais rapidamente para o interior das peças, degradando com maior rapidez as propriedades do concreto. As normas internacionais estabelecem dimensões mínimas para evitar essa rápida degradação estrutural. A adição de fibras de polipropileno na mistura do concreto também pode minimizar os riscos do *spalling*, pois em caso de incêndio as fibras se derretem e formam sulcos, criando micro-canais por onde a pressão de vapor interna pode ser liberada, mitigando a desagregação e explosão do material.

### 3 ANÁLISE TÉRMICA VIA MEF

A análise térmica desenvolvida neste trabalho visa basicamente determinar a distribuição de temperatura nos elementos estruturais em situação de incêndio por meio de um procedimento de solução de equações não lineares do tipo incremental simples. Tal procedimento possibilita a consideração da variação das propriedades térmicas e físicas dos materiais em função do aumento da temperatura, observando recomendações estabelecidas por normas para modelos de análise avançada. Além disso, cuidou-se em realizar a análise térmica de forma que uma análise estrutural tenha os subsídios necessários para a avaliação da resistência e da rigidez do material estrutural, pois, dentre os efeitos provocados nas propriedades do material devido às altas temperaturas, os mais significativos estão associados à modificação (degradação) da resistência e rigidez.

A seguir será apresentado o problema de transferência de calor, bem como a formulação para análise térmica via Método dos Elementos Finitos (MEF).

#### 3.1 EQUAÇÃO DIFERENCIAL DA CONDUÇÃO DE CALOR EM SÓLIDOS

Partindo-se de um referencial cartesiano em duas dimensões, pode-se considerar que o fluxo de calor por condução é definido a partir da Lei de Fourier. Tal lei estabelece que o fluxo de calor depende da área da seção transversal do sólido, da condutividade térmica e do gradiente térmico. Assim, o fluxo de calor por unidade de área e por unidade de tempo é dado por (Bathe, 1996):

$$q_x = -\lambda_x \frac{\partial T}{\partial x} \quad (3.1a)$$

$$q_y = -\lambda_y \frac{\partial T}{\partial y} \quad (3.1b)$$

Sendo  $\lambda_x$  e  $\lambda_y$  as condutividades térmicas nas direções x e y, T é a temperatura.

Na transferência de calor por convecção, o fluxo de calor por unidade de área é governado pela Lei de Newton, ou seja:

$$q_c = \alpha_c (T_s - T_\infty) \quad (3.2)$$

Em que  $\alpha_c$  é o coeficiente de transmissão de calor por convecção, chamado também de coeficiente de película médio,  $T_\infty$  é temperatura do fluido e  $T_s$  é a temperatura na superfície do sólido.

O fluxo de calor por unidade de área na transferência por radiação é dado pela seguinte expressão:

$$q_r = \sigma \varepsilon (T_s^4 - T_\infty^4) \quad (3.3)$$

Em que  $\sigma$  é o coeficiente de Stefan-Boltzmann ( $\sigma = 5,67 \times 10^{-8} \text{ W / m}^2 \text{ K}^4$ ),  $\varepsilon$  é a emissividade ( $0 < \varepsilon < 1$ ),  $T_\infty$  é temperatura do fluido e  $T_s$  é a temperatura na superfície do sólido.

A Equação 3.3 pode ser representada da seguinte forma:

$$q_r = \sigma \varepsilon (T_s - T_\infty)(T_s + T_\infty)(T_s^2 + T_\infty^2) \quad (3.4)$$

E considerando-se que:

$$\alpha_r = \sigma \varepsilon (T_s + T_\infty)(T_s^2 + T_\infty^2) \quad (3.5)$$

Obtém-se:

$$q_r = \alpha_r (T_s - T_\infty) \quad (3.6)$$

Em que  $\alpha_r$  é o coeficiente de transmissão de calor por radiação.

Somando-se as Equações 3.2 e 3.6 pode-se obter um fluxo de calor equivalente, dado por:

$$q_{cr} = q_c + q_r \quad (3.7)$$

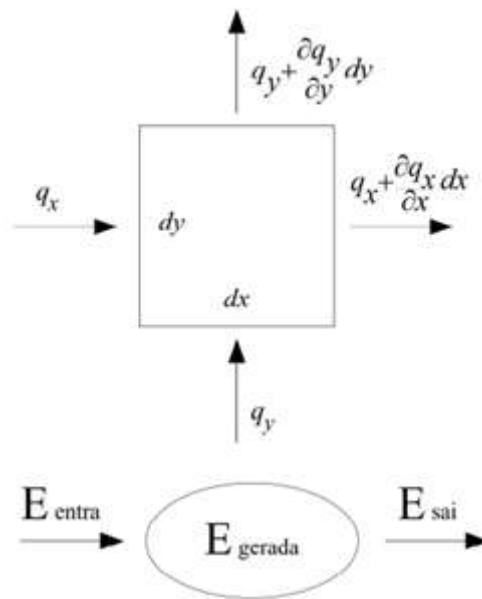
Substituindo-se as Equações 3.4 e 3.6 na Equação 3.7, tem-se:

$$q_{cr} = h(T_s - T_\infty) \quad (3.8)$$

Em que  $h$  é o coeficiente de transmissão de calor por convecção e radiação, definido pela soma  $(\alpha_c + \alpha_r)$ .

Considerando agora um elemento infinitesimal de um corpo bidimensional (espessura constante) conforme a Figura 3.1, pode-se determinar a equação que governa o problema de condução de calor em um meio em equilíbrio aplicando-se a primeira lei da termodinâmica, Lei de Conservação da Energia (Figura 3.1), no elemento infinitesimal bidimensional.

Figura 3.1 - Princípio da conservação da energia aplicada ao elemento infinitesimal



Fonte: Pires (2018)

A diferença entre o fluxo que entra e o fluxo que sai no elemento é dada por:

$$(q_x + q_y) - \left[ \left( q_x + \frac{\partial q_x}{\partial x} dx \right) + \left( q_y + \frac{\partial q_y}{\partial y} dy \right) \right] \quad (3.9)$$

A energia interna gerada no sistema e a energia armazenada no corpo, considerando que a espessura do corpo é unitária, são definidas, respectivamente, por:

$$Q dx dy \quad (3.10)$$

$$\rho c dx dy \frac{\partial T}{\partial t} \quad (3.11)$$

Em que  $Q$  é o calor gerado no interior do elemento por unidade de volume e tempo,  $\rho$  é a massa específica do material,  $c$  é o calor específico do material e  $T(x, y, t)$  é a distribuição de temperatura no espaço e no tempo.

Substituindo as Equações 3.9, 3.10 e 3.11 na lei de conservação de energia e aplicando-se a lei de Fourier, Equação 3.1, obtém-se a expressão correspondente à propagação de calor por condução, ou seja:

$$\left( \lambda_x \frac{\partial^2 T}{\partial x^2} + \lambda_y \frac{\partial^2 T}{\partial y^2} \right) + Q = \rho c \frac{\partial T}{\partial t} \quad (3.12)$$

Supondo as hipóteses de homogeneidade térmica e isotropia, a condutividade térmica  $\lambda$  é constante em qualquer ponto do material e em qualquer direção. Consequentemente, a equação anterior fica:

$$\lambda \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + Q = \rho c \frac{\partial T}{\partial t} \quad (3.13)$$

Ou ainda,

$$\nabla^2 T + \frac{Q}{\lambda} = \frac{\rho c}{\lambda} \frac{\partial T}{\partial t} = 0 \quad (3.14)$$

Em que  $\nabla$  é o operador Laplaciano e  $\frac{\lambda}{\rho c}$  é a difusividade térmica do material. No caso do regime permanente (ou estacionário), a temperatura não varia com o tempo, ou seja,  $\partial T / \partial t = 0$  e as Equações 3.13 e 3.14 resultam na equação de Poisson que governa vários dos problemas de campo importantes na engenharia. Além disso, em regime permanente e não havendo geração interna de calor ( $Q=0$ ) tem-se nas expressões 3.13 e 3.14 a equação conhecida como equação de Laplace.

### 3.2 CONDIÇÕES INICIAIS E DE CONTORNO

O campo de temperaturas que satisfaz a equação de condução de calor deve satisfazer as condições iniciais e de contorno do problema, isto é, para obter a solução da Equação 3.12, é necessário especificar as condições iniciais ( $t = t_0$ ) no domínio ( $\Omega$ ), bem como condições de contorno associadas à superfície ( $\Gamma$ ) para o problema de interesse. Para os corpos sólidos, as trocas de calor no volume do corpo (domínio) ocorrem apenas por condução. Estando um corpo sólido envolto por um fluido, como mostra a Figura 3.2, pode-se empregar a convecção em conjunto com a radiação como condição de contorno para o problema de domínio sólido (contorno convectivo radiativo), além de um fluxo de calor prescrito.

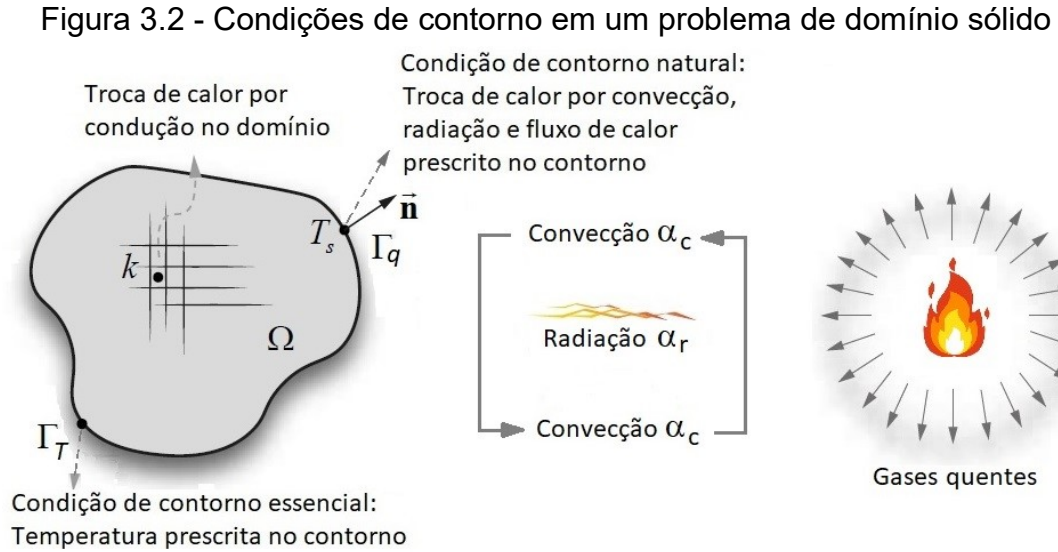
Em relação às condições iniciais a serem consideradas, essas consistem de um campo de temperaturas no início da análise, especificadas conforme a seguir, partindo de um instante de tempo  $t_0$ , ou seja:

$$T = T_0(x, y, z, 0) \text{ em } \Omega, \text{ para } t = t_0 \quad (3.15)$$

Adicionalmente, as condições de contorno a serem estabelecidas podem ser divididas em essenciais e naturais. As condições de contorno essenciais, ou também chamadas de Dirichlet, correspondem a temperaturas prescritas numa parte do contorno, e são dadas por:

$$T = T_0(x, y, z, t) \text{ em } \Gamma_T \quad (3.16)$$





Adaptado de Neves (2021)

As condições de contorno naturais, ou de Neumann, correspondem à troca de calor por convecção, radiação e fluxos de calor prescritos em uma parte do contorno, saindo do domínio na direção normal ( $n$ ) ao contorno. São representadas genericamente a seguir:

$$q = -\lambda \frac{\partial T}{\partial n} = q_0(x, y, l, t) + h(T_s - T_\infty) \text{ em } \Gamma_q \quad (3.17)$$

Com  $q_0(x, y, l, t)$  representando o fluxo de calor prescrito no instante  $t$ ,  $h$  é o coeficiente de película combinado de transmissão de calor por convecção e radiação ( $h = \alpha_c + \alpha_r$ ),  $T_s$  é a temperatura na superfície do elemento e  $T_\infty$  é a temperatura dos gases. No caso de uma superfície perfeitamente isolada (ou adiabática), o fluxo de calor é considerado igual a zero.

Como apontado anteriormente, o módulo implementado para realizar a análise térmica foi desenvolvido com base no MEF. Dessa forma, na subseção seguinte são apresentados os procedimentos para aplicação do MEF em problemas de condução de calor, associados aos sólidos.

### 3.3 EQUAÇÃO DE EQUILÍBRIO VIA MEF

O método dos elementos finitos (MEF) é um procedimento numérico bastante utilizado na análise de estruturas. Ele fundamenta-se no conceito de discretização dos meios contínuos e, a partir daí, na obtenção de soluções numéricas aproximadas. Assim, procura-se dividir o meio contínuo em subdomínios, referidos como elementos, que são interligados por meio dos pontos nodais onde são definidos os graus de liberdade a serem determinados. A ideia básica consiste em transformar um problema complexo na soma de diversos problemas simples.

O problema de condução de calor, de acordo com o apresentado na seção anterior, consiste em resolver uma equação diferencial parcial, considerando a relação constitutiva do material (Lei de Fourier), e satisfazendo as condições de contorno. Vários problemas de engenharia são regidos por uma equação diferencial válida em um domínio e sujeita a condições de contorno na superfície. Entretanto, de forma geral, só para alguns casos simples se conhecem soluções analíticas para essas equações diferenciais. Contudo, por meio da aplicação de métodos numéricos, é possível obter os valores da função desconhecida (solução do problema) em alguns pontos pré-determinados. Soluções aproximadas para equações diferenciais podem ser obtidas considerando-se o Método dos Resíduos Ponderados (MRP). Tal método é aplicável a uma grande variedade de problemas e, por isso, se faz desnecessário buscar uma formulação variacional para a aplicação do MEF nesses problemas.

Neste trabalho, o MRP é empregado em conjunto com o método de Galerkin. A partir disso, no contexto do MEF, chega-se a equação de equilíbrio que governa o problema transiente de condução de calor, que é descrita a seguir na forma matricial:

$$\mathbf{C} \left\{ \frac{\partial \mathbf{T}}{\partial t} \right\} + \mathbf{K} \{ \mathbf{T} \} = \mathbf{R} \quad (3.18)$$

Em que:

$\mathbf{C} = \int_{\Omega} \rho c \mathbf{N}^T \mathbf{N} d\Omega$  é a matriz de capacitância (capacidade térmica);

$\mathbf{K} = \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} d\Omega + h \int_{\Gamma} \mathbf{N}^T \mathbf{N} d\Gamma$  é a matriz de condutividade térmica;

$\mathbf{R} = Q \int_{\Omega} \mathbf{N}^T d\Omega + h T_{\infty} \int_{\Gamma} \mathbf{N}^T d\Gamma - q_0 \int_{\Gamma} \mathbf{N}^T d\Gamma$  é o vetor de fluxos de calor nodais.

$\mathbf{T}$  é o vetor de temperaturas nodais que se deseja determinar.

As matrizes  $\mathbf{N}$ ,  $\mathbf{B}$  e  $\mathbf{D}$  nas equações anteriores são, respectivamente: a matriz de funções interpoladoras; a matriz que contém as derivadas das funções interpoladoras em relação à  $x$  e  $y$ ; e a matriz que contém as condutividades térmicas,  $\lambda_x$  e  $\lambda_y$ , ou simplesmente constante,  $\lambda$ .

O vetor de fluxos de calor no domínio pode ser calculado em função dos valores nodais de temperatura pela expressão:

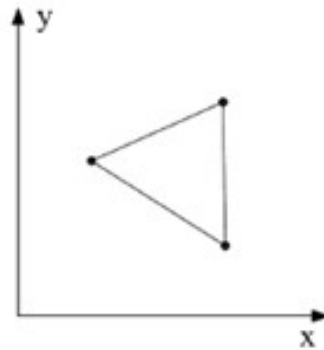
$$\mathbf{q} = -\mathbf{D} \mathbf{B} \mathbf{T}^e \quad (3.19)$$

Os efeitos de ações térmicas podem ser analisados considerando os problemas térmicos em regime permanente. Com isso as ações térmicas no contorno não variam com o tempo e, assim, o campo de temperaturas resultante é constante. Logo, a equação de equilíbrio do problema estacionário de condução de calor é dada fazendo-se o termo  $\frac{\partial \mathbf{T}}{\partial t}$  na Equação 3.18 igual a zero, portanto:

$$\mathbf{K} \{\mathbf{T}\} = \mathbf{R} \quad (3.20)$$

O elemento finito implementado no módulo de análise térmica e utilizado nas verificações deste trabalho é apresentado na Figura 3.3. Implementou-se o elemento triangular linear (T3) com três nós. Na Tabela 3.1 é apresentada todos os vetores e matrizes e obtidos para o elemento triangular linear.

Figura 3.3 - Elemento finito implementado



Fonte: Autor

Tabela 3.1 - Matrizes e vetores para o elemento T3

Variável	Matrizes e Vetores		
	Aresta 1-2	Aresta 2-3	Aresta 1-3
$\mathbf{q}_{cr}$	$hT_s \frac{L_{1-2}}{2} \begin{Bmatrix} 1 \\ 1 \\ 0 \end{Bmatrix}$	$hT_s \frac{L_{2-3}}{2} \begin{Bmatrix} 0 \\ 1 \\ 1 \end{Bmatrix}$	$hT_s \frac{L_{1-3}}{2} \begin{Bmatrix} 1 \\ 0 \\ 1 \end{Bmatrix}$
$\mathbf{q}_0$	$q_0 \frac{L_{1-2}}{2} \begin{Bmatrix} 1 \\ 1 \\ 0 \end{Bmatrix}$	$q_0 \frac{L_{2-3}}{2} \begin{Bmatrix} 0 \\ 1 \\ 1 \end{Bmatrix}$	$q_0 \frac{L_{1-3}}{2} \begin{Bmatrix} 1 \\ 0 \\ 1 \end{Bmatrix}$
$\mathbf{q}_Q$	$Q \frac{A}{3} \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix}$		
$\mathbf{k}_c$	$\frac{hL_{1-2}}{6} \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\frac{hL_{2-3}}{6} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}$	$\frac{hL_{1-3}}{6} \begin{bmatrix} 2 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 2 \end{bmatrix}$
$\mathbf{k}_T$	$\mathbf{B}^T \mathbf{D} \mathbf{B}$		
$\mathbf{C}$	$\rho c \frac{A}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$		

Fonte: Autor

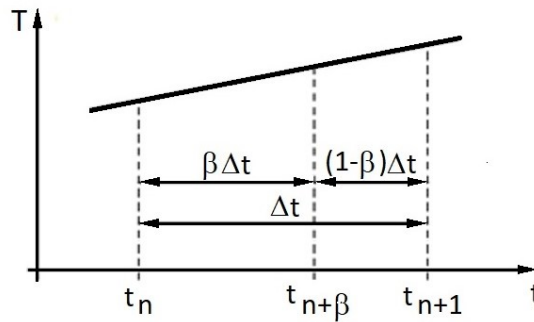
### 3.4 SOLUÇÃO DO PROBLEMA TRANSIENTE DE CONDUÇÃO DE CALOR

A Equação 3.18 é resultante apenas da discretização do espaço, representando um sistema de equações diferenciais ordinárias de primeira ordem. Na solução desse sistema de equações utilizam-se métodos numéricos que

requerem a discretização no tempo. A discretização do tempo adotada corresponde a uma aplicação do método das diferenças finitas, com base nas seguintes hipóteses:

- A Equação 3.18 é satisfeita apenas em pontos discretos  $t_{n+\beta}$  de cada intervalo de tempo  $\Delta t$  discretizado, conforme apresentado na Figura 3.4.
- As temperaturas variam de forma linear ao longo do intervalo de tempo  $\Delta t$ , desde o instante  $t_n$  até  $t_{n+1} = t_n + \Delta t$ .

Figura 3.4 - Variação da temperatura no intervalo de tempo  $\Delta t$



Adaptado de Ribeiro (2004)

As temperaturas  $T_{n+\beta}$  no instante  $t_{n+\beta} = t_n + \beta\Delta t$  podem ser dadas então, por:

$$T_{n+\beta} = T_n + \beta\Delta t(T_{n+1} - T_n)/\Delta t \text{ com } \beta \in (0,1) \text{ e } n \in (0,1,...N-1) \quad (3.21)$$

Sendo  $N$  o número total de intervalos de tempo e  $\beta$  é um parâmetro que define dentro de cada intervalo de tempo o instante em que a Equação 3.18 é satisfeita.

Pela segunda hipótese, a derivada da temperatura em relação ao tempo é constante dentro de cada intervalo é dada por:

$$\frac{\partial T_{n+\beta}}{\partial t} = \frac{T_{n+1} - T_n}{\Delta t} \quad (3.22)$$

Substituindo-se a Equação 3.21 e a Equação 3.22 na Equação 3.18 chega-se à seguinte equação:

$$\left( \frac{\mathbf{C}_{n+\beta}}{\beta \Delta t} + \mathbf{K}_{n+\beta} \right) \mathbf{T}_{n+\beta} = \mathbf{R}_{n+\beta} + \frac{\mathbf{C}_{n+\beta}}{\beta \Delta t} \mathbf{T}_n \quad (3.23)$$

As matrizes  $\mathbf{C}_{n+\beta}$ ,  $\mathbf{K}_{n+\beta}$ ,  $\mathbf{R}_{n+\beta}$  são avaliadas no instante  $t_n$ . Reescrevendo-se a Equação 3.23 de outra forma, obtém-se:

$$\hat{\mathbf{K}}_{n+\beta} \mathbf{T}_{n+\beta} = \hat{\mathbf{R}}_{n+\beta} \quad (3.24)$$

Onde:

$$\hat{\mathbf{K}}_{n+\beta} = \frac{\mathbf{C}_{n+\beta}}{\beta \Delta t} + \mathbf{K}_{n+\beta} \quad (3.25a)$$

$$\hat{\mathbf{R}}_{n+\beta} = \frac{\mathbf{C}_{n+\beta}}{\beta \Delta t} \mathbf{T}_n + \mathbf{R}_{n+\beta} \quad (3.25b)$$

Resolvido o sistema de equações para  $\mathbf{T}_{n+\beta}$ , as temperaturas no final do intervalo de tempo  $t_{n+1}$  são dadas por:

$$\mathbf{T}_{n+1} = \frac{1}{\beta} \mathbf{T}_{n+\beta} + \left( 1 - \frac{1}{\beta} \right) \mathbf{T}_n \quad (3.26)$$

Sendo essas as temperaturas  $\mathbf{T}_n$  quando se avança para o intervalo de tempo seguinte.

Como pode ser observada, a Equação 3.23 é fortemente não linear devido à dependência estabelecida das propriedades do material com a temperatura. Cabe destacar que não há um método geral para resolver tal sistema não linear de equações. Contudo, existem diversas técnicas numéricas essencialmente baseadas numa integração linear no tempo e utilizando processos iterativos. Neste trabalho foi implementado o procedimento incremental simples apresentado no esquema da Figura 3.5. Assim, os valores da temperatura no passo de tempo corrente,  $t_{n+1}$  são

obtidos por meio das temperaturas calculadas no passo de tempo anterior  $t_n$  e usando os fluxos de calor nodais no passo de tempo corrente e anterior. O parâmetro  $\beta$  define, dentro de cada intervalo de tempo, o instante em que a Equação 3.18 será satisfeita. Através da variação deste parâmetro, diferentes esquemas de integração no tempo podem ser obtidos.

No presente trabalho,  $\beta$  é adotado igual a  $2/3$  (Esquema Implícito de Galerkin) nas análises. Mais informações sobre os esquemas de integração no tempo e a solução detalhada do problema transiente de condução de calor podem ser encontradas em Real (1988), Bathe(1996), Hughes (1987), Lewis et al. (2004), Rigobello (2011) e Nunes (2014).

Figura 3.5 - Algoritmo incremental simples

---

**1. Solução do problema transiente de condução de calor:**

1a. Definir os dados de entrada, condições iniciais e de contorno

1b. Fazer:  $T_n = T_{n+1} = T_0 = 20^\circ\text{C}$  (Temperatura ambiente)

**2. Processo incremental:** (Incremento de tempo = 1, 2, 3,..., nmáx)

2a. Calcular:  $C_{n+\beta} = \int_{\Omega} \rho c N^T N d\Omega$  (Matriz de capacitância)

2b. Calcular:  $K_{n+\beta} = \int_{\Omega} B^T D B d\Omega + h \int_{\Gamma} N^T N d\Gamma$  (Matriz de condutividade)

2c. Calcular:  $R_{n+\beta} = Q \int_{\Omega} N^T d\Omega + h T_{\infty} \int_{\Gamma} N^T d\Gamma - q_0 \int_{\Gamma} N^T d\Gamma$  (Vetor de fluxo de calor)

2d. Obter:  $\hat{K}_{n+\beta} = \frac{C_{n+\beta}}{\beta \Delta t} + K_{n+\beta}$

2e. Obter:  $\hat{R}_{n+\beta} = \frac{C_{n+\beta}}{\beta \Delta t} T_n + R_{n+\beta}$

2f. Resolver o sistema de equações:  $\hat{K}_{n+\beta} T_{n+\beta} = \hat{R}_{n+\beta}$

2g. Calcular:  $T_{n+1} = \frac{1}{\beta} T_{n+\beta} + \left(1 - \frac{1}{\beta}\right) T_n$  (Temperatura no instante  $T_{n+1}$ )

**3. Realizar um novo incremento de tempo e voltar ao passo 2**

---

Fonte: Autor

## 4 IMPLEMENTAÇÃO COMPUTACIONAL

### 4.1 GENERALIDADES

O programa computacional desenvolvido neste trabalho, para a determinação numérica da distribuição e evolução da temperatura nas seções transversais de elementos estruturais, será apresentado neste capítulo. O módulo, denominado PPLANLEP/AI (Programa de Pórticos Planos de Aço considerando a Análise Não Linear Elasto-Plástica/Análise sob Incêndio) é baseado no MEF, como demonstrado no Capítulo 3, e realiza as análises térmicas, permanente e transiente, para seções transversais bidimensionais de geometria qualquer.

O PPLANLEP/AI foi implementado na considerada linguagem de baixo nível, Fortran 90/95 (Chapman, 2003). As linguagens de programação de baixo nível, apesar de menos amigáveis que as linguagens de alto nível, possuem as vantagens de aproveitarem da melhor forma a arquitetura do computador e também são mais rápidas para compilar e processar os dados, benefícios básicos na otimização computacional.

A entrada de dados no PPLANLEP/AI é realizada por meio de documento de texto, de extensão (.geo). É um arquivo exportado pela versão acadêmica do *software* Mechanical APDL 2021 R1, onde contém todas as informações da malha como as coordenadas cartesianas dos pontos nodais e conectividade dos elementos finitos. O arquivo de entrada ainda deve conter as seguintes informações:

- Número de materiais;
- Número de elementos;
- Número de nós;
- Tempo total de análise em segundos;
- Intervalo de tempo em segundos.
- Temperatura prescrita (valor da temperatura em °C, número de nós expostos à temperatura prescrita e os nós discriminados);
- Fluxo prescrito (valor do fluxo em W/m<sup>2</sup>, número de nós expostos ao fluxo de calor prescrito e os nós discriminados);
- Fluxo de calor do incêndio (número de nós expostos ao fluxo de calor do incêndio e os nós discriminados).



Na Figura 4.1 é apresentado um exemplo do arquivo de entrada.

Figura 4.1 - Arquivo de entrada

```

2 1248 720.00000 10.00000 720.00000 (Nº total de mat.; Nº total de elem.; Tempo da análise; Intervalo de tempo; Nº total de informações de saída)
726 (Nº total de nós)
  1 0.00000E+00 0.00000E+00 0.00000E+00 (Nº do nó; Coordenadas x, y, z)
  2 1.00000E-01 0.00000E+00 0.00000E+00
  3 2.00000E-01 0.00000E+00 0.00000E+00
  .
  .
  .
Mat 1 (Nº do material)
  332 (Nº de elementos)
    1 85 64 65 (Nº do elemento; Numeração dos nós que compõe o elemento triangular)
    2 85 84 64
    3 3 636 73
    .
    .
    .
Mat 2 (Nº do material)
  916 (Nº elementos)
    333 651 281 280 (Nº do elemento; Numeração dos nós que compõe o elemento triangular)
    334 651 262 281
    335 652 289 266
    .
    .
    .
Temperature
point
  2 (Nº de nós)
100.0000 1 (Valor da temperatura prescrita; Nº do nó exposto à temperatura prescrita)
100.0000 2
Flux
Point
  2 (Nº de nós)
500.0000 3 (Valor do fluxo prescrita; Nº do nó exposto ao fluxo prescrita)
500.0000 4
Flux
Point
  2 (Nº de nós)
  5 (Nº do nó exposto ao incêndio)
  6

```

Fonte: Autor

Os resultados são também impressos num arquivo de texto, de extensão (.txt). As respostas apresentadas nesse arquivo de saída dependem do nó escolhido, em que se podem verificar o tempo de análise (passo de tempo), em segundos, e a sua correspondente temperatura (°C) para o nó. A Figura 4.2 apresenta um exemplo do arquivo de resultados.

A discretização das seções transversais (malha de elementos finitos) e a etapa de pós-processamento são efetuadas pela versão acadêmica dos *softwares* Mechanical APDL 2021 R1 e EnSight Standard 2021 R1, respectivamente. Ambos os *softwares* pertencem ao pacote ANSYS 2021 R1. A interação entre o módulo computacional PPLANLEP/AI e os *softwares* Mechanical APDL 2021 R1 e EnSight Standard 2021 R1 foi solucionada por meio de programação em linguagem própria desses dois últimos programas.

Figura 4.2 - Arquivo de saída

(Tempo)	(Temperatura)
00.000000000000	20.000000000000
10.000000000000	23.5134909458664
20.000000000000	28.6289047940631
30.000000000000	39.8842435768807
40.000000000000	55.8913646082244
50.000000000000	70.9322815118499
60.000000000000	126.058556797435
70.000000000000	130.082308474093
80.000000000000	134.382273742142
90.000000000000	138.934785390101
100.000000000000	143.729712319622
110.000000000000	148.754656507254
120.000000000000	153.993810588549
130.000000000000	159.433587076631
140.000000000000	165.059910761474
150.000000000000	170.859722098557
160.000000000000	188.843201314798
170.000000000000	198.883945706410
180.000000000000	209.042025870789
190.000000000000	215.176284831525
200.000000000000	221.406281879407
210.000000000000	231.960434246656
220.000000000000	238.311666625676
230.000000000000	244.734901413060
240.000000000000	251.223191431193
250.000000000000	283.409224230654
260.000000000000	289.628886419273
270.000000000000	295.873051010244
280.000000000000	302.140985172014
290.000000000000	308.430452054929
300.000000000000	314.739837313945
.	.
.	.
.	.

Fonte: Autor

No Apêndice deste trabalho é disponibilizado o código fonte completo do módulo PPLANLEP/AI.

## 4.2 CARACTERÍSTICAS DO PROGRAMA PPLANLEP/AI

A seguir, são mostradas as características do programa implementado PPLANLEP/AI, referentes às curvas de incêndio, às propriedades dos materiais, tipos de elementos utilizáveis, condições de contorno e método de solução do sistema de equações.

### 4.2.1 Curvas de incêndio

O programa PPLANLEP/AI foi elaborado com a possibilidade de uso das seguintes curvas de incêndio:

- Curva ISO-834-1:1999;
- Curva ASTM E119:2000;
- Curva de hidrocarbonetos;
- Curva de incêndio para elementos exteriores.

#### **4.2.2 Propriedade dos materiais**

O programa utiliza todas as propriedades variáveis com a temperatura para os seguintes materiais:

- Aço estrutural, de acordo com a ABNT NBR 14323:2013;
- Concreto de densidade normal, de acordo com a ABNT NBR 14323:2013;
- Blaze Shield II, de acordo com Silva (2001).

#### **4.2.3 Elemento finito utilizável**

Como informado no Capítulo 3, o programa foi implementado com o elemento finito triangular linear.

#### **4.2.4 Condição inicial**

O programa admite como condição inicial um único valor de temperatura prescrita em todo o domínio no início da análise, ou seja,  $t = 0$ .

#### **4.2.5 Condições de contorno**

O programa admite três formas distintas de condições de contorno:

- Nós com temperatura prescrita;
- Superfícies com fluxo de calor prescrito;
- Superfícies com incêndio por convecção e radiação.

#### 4.2.6 Método de solução do sistema de equações

O programa resolve o sistema de equações lineares por eliminação Gaussiana direta, com a utilização do método  $LDL^T$  de escalonamento proposto por Bathe (1996). Procurou-se melhorar o desempenho do algoritmo de solução por meio do armazenamento das matrizes de rigidez em esquema *skyline*.

#### 4.3 SUBROTINAS

A seguir, no Quadro 4.1, são descritas de forma sucinta as principais subrotinas que compõem o módulo computacional PPLANLEP/AI.

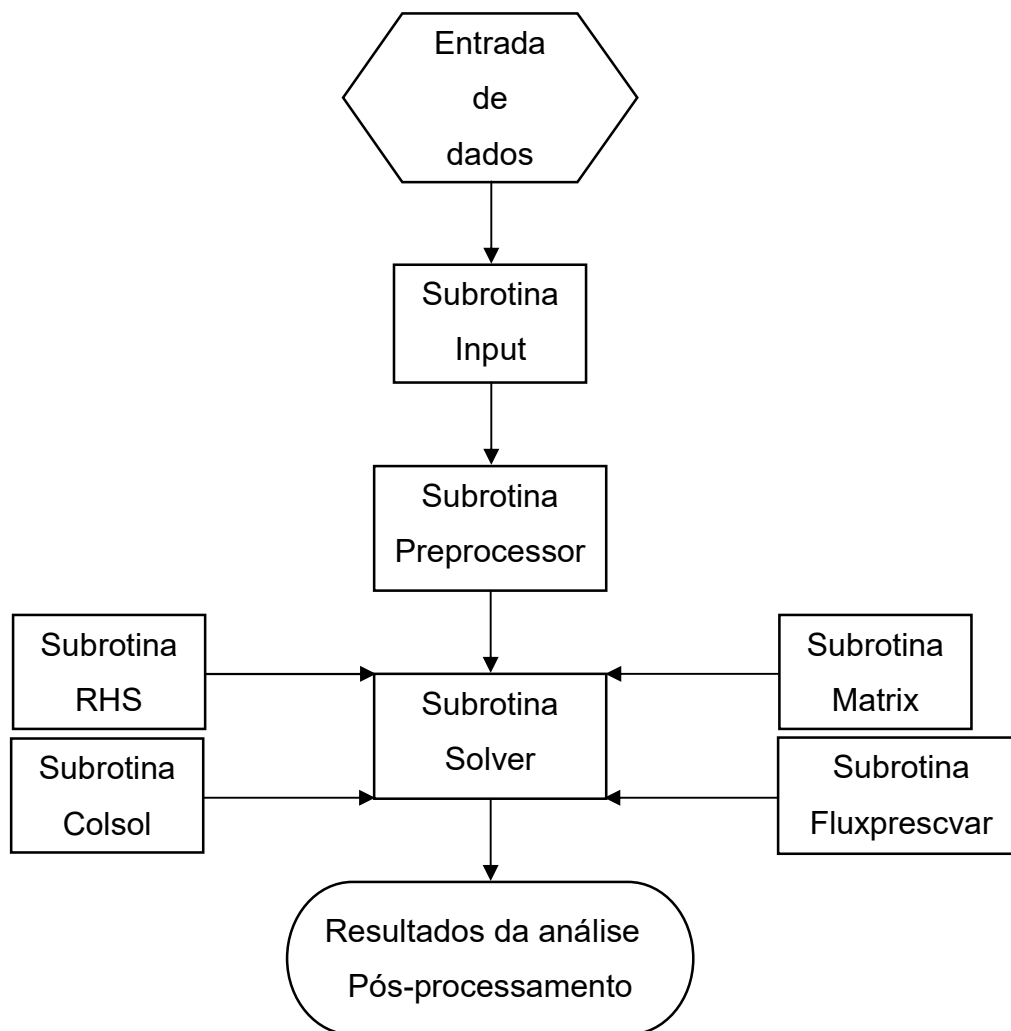
Quadro 4.1 - Subrotinas do PPLANLEP/AI

Subrotinas	Descrição
Input	<ul style="list-style-type: none"> <li>• Abre arquivo de entrada de dados</li> <li>• Leitura dos dados de entrada gerais do programa</li> <li>• Salva os dados em constantes e variáveis declaradas</li> </ul>
Preprocessor	Organiza os dados nas seguintes matrizes: <ul style="list-style-type: none"> <li>• ID (vetor que identifica a equação associada a cada nó não prescrito)</li> <li>• LM (matriz que associa os pontos nodais de cada elemento ao número da equação correspondente)</li> </ul>
Solver	<ul style="list-style-type: none"> <li>• Resolve o sistema de equações algébricas</li> <li>• Imprime os valores da temperatura em cada passo de tempo para o nó escolhido</li> </ul>
Matrix	Monta as matrizes de capacitância, de condutividade térmica e o vetor de fluxo de calor prescrito
Fluxprescvar	Monta o vetor de fluxo de calor variável correspondente ao incêndio
RHS	Organiza os vetores de fluxo de calor e temperatura prescrita num vetor de carga
Colsol	Resolve o sistema de equações algébricas dentro da subrotina Solver

Fonte: Autor

Na Figura 4.3 é apresentado um esquema geral do programa PPLANLEP/AI.

Figura 4.3 - Fluxograma geral do PPLANLEP/AI



Fonte: Autor

## 5 EXEMPLOS NUMÉRICOS

Neste capítulo, o programa computacional desenvolvido para análise térmica é avaliado considerando os resultados obtidos em outros trabalhos disponíveis na literatura e por softwares conhecidos. Quatro exemplos são analisados, com diferentes características físicas e geométricas, condições de contorno distintas, por meio das análises térmicas, permanente e transiente. Admitiu-se o incremento de tempo de 10s.

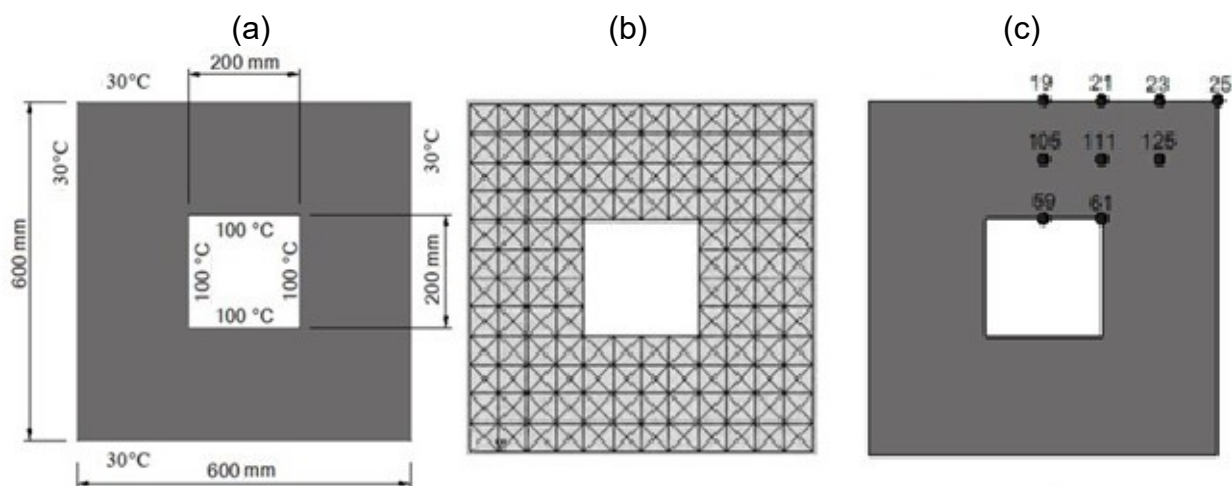
### 5.1 CHAMINÉ INDUSTRIAL

A Figura 5.1 apresenta a seção transversal de uma chaminé industrial feita de concreto. Este exemplo foi estudado por Moaveni (1999) e, posteriormente, por Rigobello (2011), que realizaram a análise térmica permanente (estacionária). As dimensões e as condições de contorno da chaminé são apresentadas na Figura 5.1a, em que  $T_{\infty} = 30^{\circ}\text{C}$  e  $T = 100^{\circ}\text{C}$  são, respectivamente, a temperatura ambiente no contorno externo e a temperatura interna da chaminé, assumida uniforme. O coeficiente de transferência de calor por convecção  $h$  da superfície é igual a  $20 \text{ W/m}^2\text{C}$ . Na Figura 5.1b é apresentada a discretização da malha e na Figura 5.1c são mostrados os nós que são verificados. A condutividade térmica do concreto é assumida igual a  $1,4 \text{ W/m}^{\circ}\text{C}$ .

Os resultados obtidos pelo programa desenvolvido no presente trabalho são apresentados na Tabela 5.1, onde são comparados aos resultados obtidos pela versão acadêmica do *software* Mechanical APDL 2021 R1 e pelos trabalhos de Moaveni (1999) e Rigobello (2011). Observa-se boa concordância entre os resultados. Os valores das temperaturas nodais da análise térmica permanente, obtidos pelo programa computacional foram praticamente iguais aos valores obtidos pelo Mechanical APDL 2021 R1, como esperado, pois a mesma malha de elementos finitos foi utilizada em ambos os programas. Os resultados também foram bem próximos aos obtidos por Moaveni (1999) e Rigobello (2011). A pequena diferença entre os valores se deve à discretização distinta das malhas adotadas na modelagem e ao método de resolução do sistema de equações. No presente

trabalho empregou-se um método direto, enquanto os outros autores utilizaram métodos numéricos aproximados.

Figura 5.1 - Chaminé industrial: (a) Dimensões e condições de contorno, (b) Discretização da malha, (c) Nós verificados



Fonte: Autor

Tabela 5.1 - Temperaturas nodais: Chaminé industrial  
TEMPERATURAS (°C)

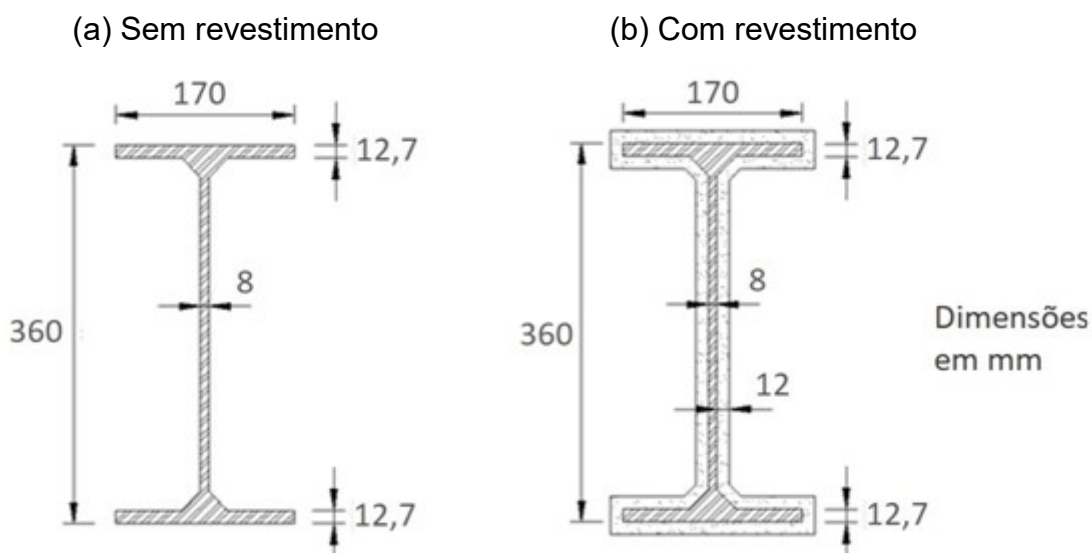
Nó	PPLANLEP/AI	ANSYS	MOAVENI			RIGOBELLO	
	(A)	(B)	(A)/(B)	(C)	(A)/(C)	(D)	(A)/(D)
19	45,02	44,92	1,002	45,88	0,981	45,94	0,980
21	43,32	43,25	1,002	43,67	0,992	43,73	0,991
23	38,98	38,98	1,000	40,10	0,972	40,13	0,971
25	33,59	34,08	0,986	32,73	1,026	32,72	1,027
59	100,00	100,00	1,000	100,00	1,000	100,00	1,000
61	100,00	100,00	1,000	100,00	1,000	100,00	1,000
105	68,99	68,95	1,001	70,83	0,974	70,95	0,972
111	64,45	64,42	1,000	67,02	0,962	67,12	0,960
125	52,27	52,26	1,000	51,56	1,014	51,64	1,012

Fonte: Autor

## 5.2 PERFIL I LAMINADO

Apresenta-se agora a análise térmica de um perfil I laminado do tipo IPE 360 sem e com o revestimento de proteção térmica, conforme mostrado na Figura 5.2.

Figura 5.2 - Perfil I Laminado IPE 360

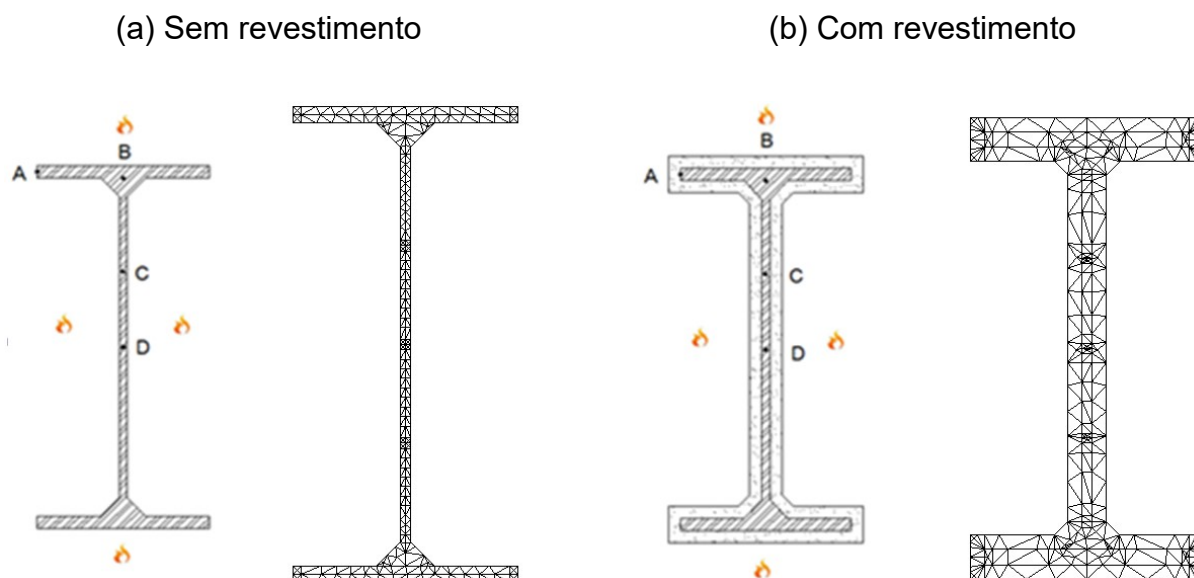


Fonte: Autor

O perfil, estudado também por Rigobello (2011) e Pires et al. (2015), foi considerado exposto ao fogo por todos os lados. Modelou-se o perfil sem e com a proteção térmica do tipo contorno, utilizando-se uma malha de elementos triangulares lineares (T3) como mostrado na Figura 5.3. O objetivo é validar o programa por meio dos resultados fornecidos pelo *software* SAFIR (Franssen, 2005) e pelo programa CS-ASA/FA (*Computational System for Advanced Structural Analysis/Fire Analysis*) apresentados no trabalho de Pires et al. (2015). O revestimento considerado na proteção térmica foi a argamassa projetada, Blaze Shield II, de espessura igual a 12 mm. As propriedades térmicas desse material foram consideradas constantes para a comparação, sendo: o calor específico igual a 2093 J/kg°C, a condutividade térmica igual a 0,043 W/m°C, a massa específica igual a 240 kg/m³ (fonte: Isolatek) e a emissividade resultante igual a 0,8 de acordo com recomendação do EN 1991-1-2:2002. As duas seções são expostas ao incêndio-padrão, sendo o coeficiente de transferência de calor por convecção adotado igual a 25 W/m²°C. Para as análises do perfil, foram considerados os quatro pontos destacados na Figura 5.3.



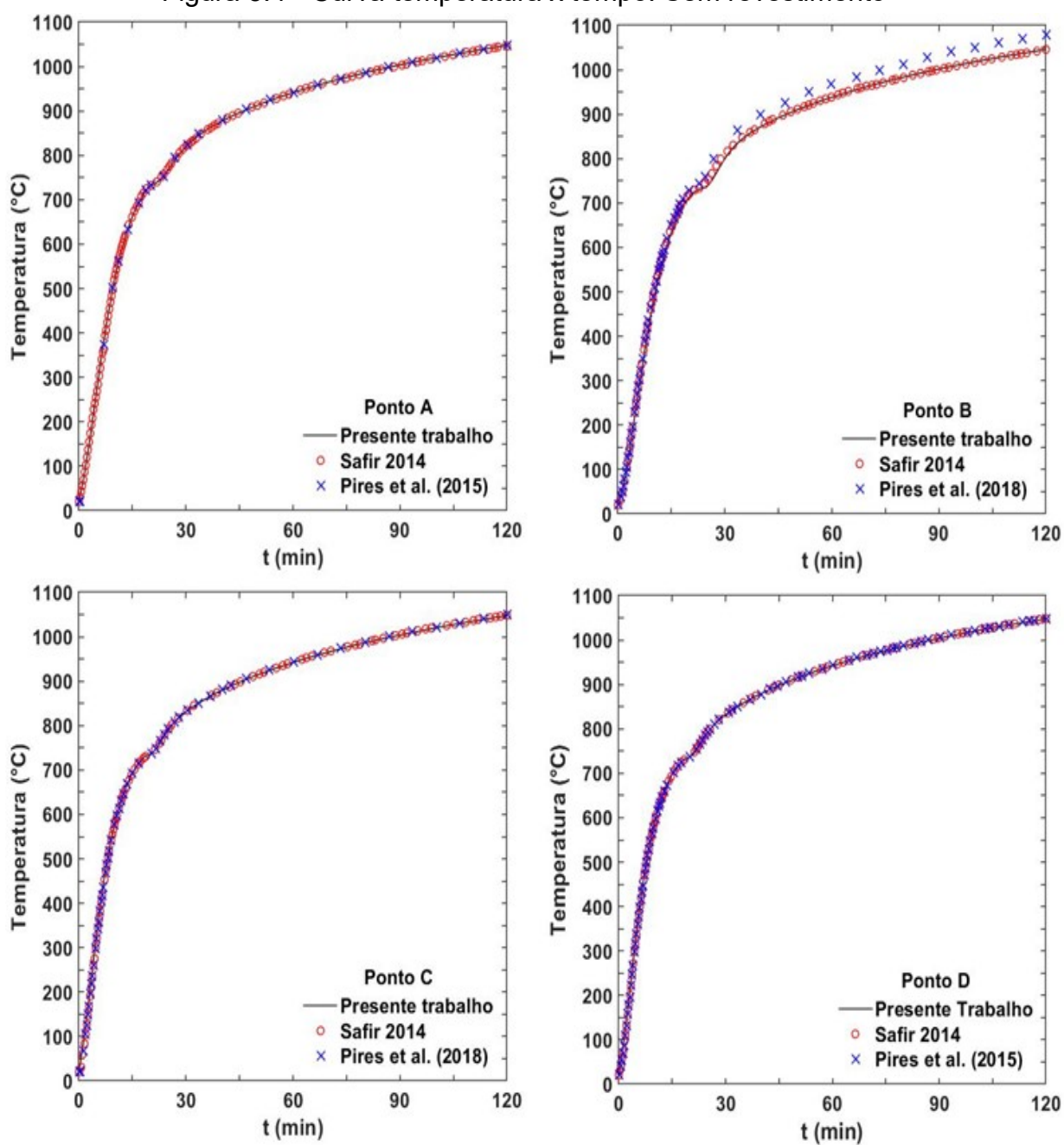
Figura 5.3 - Nós considerados na análise do perfil/malha de elementos finitos



Fonte: Autor

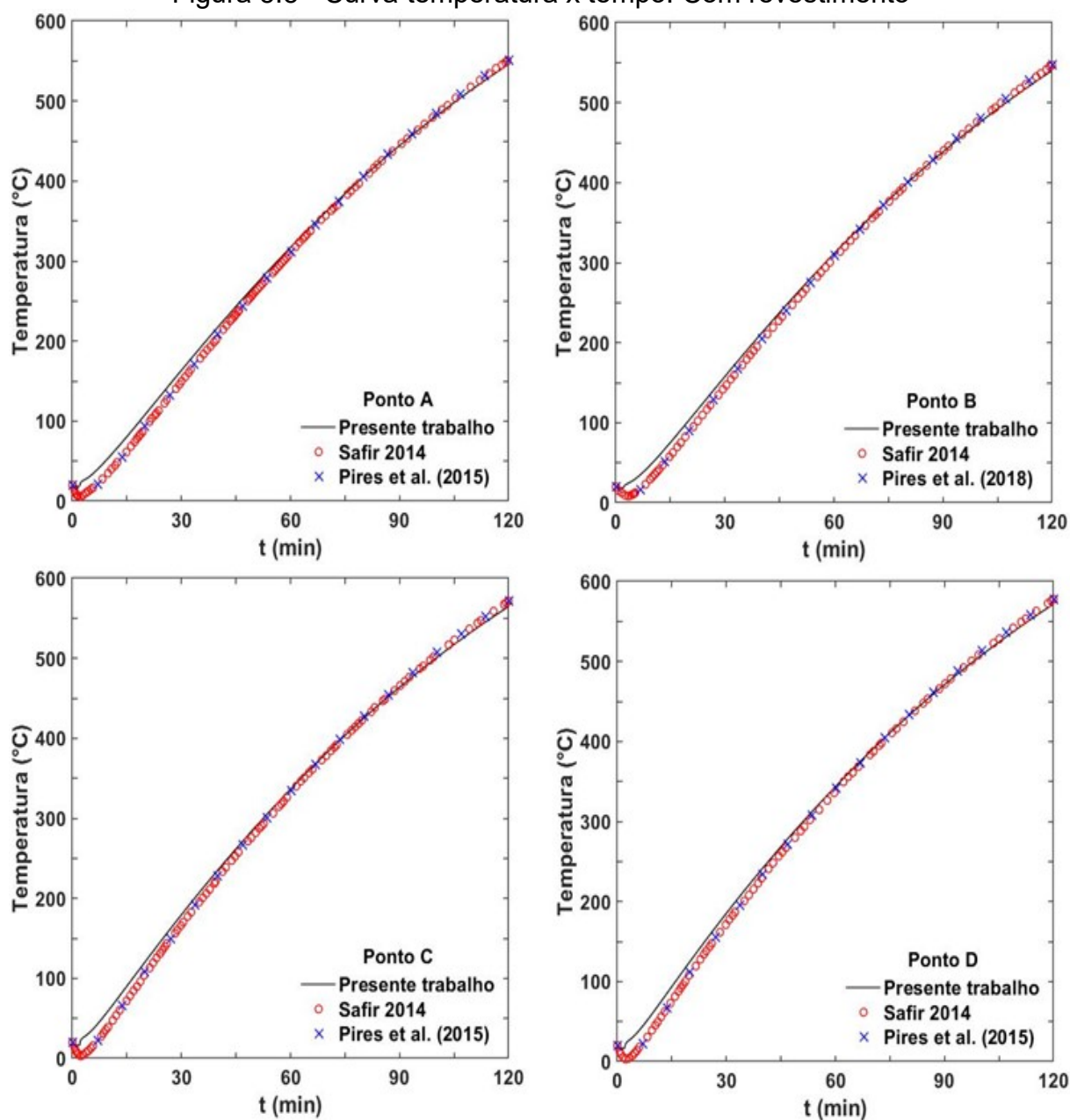
As curvas temperatura versus tempo, obtidas para os pontos A, B, C e D do perfil sem e com revestimento são ilustradas nas Figuras 5.4 e 5.5, respectivamente. Na presente pesquisa o perfil IPE360 sem revestimento foi discretizado em 393 elementos finitos triangulares lineares e com revestimento de proteção foi discretizado em 450 elementos finitos triangulares lineares. No trabalho de Pires et al. (2015), para ambos os casos, os perfis foram discretizados em elementos triangulares lineares com três nós (T3) e quadrilaterais lineares com 4 nós (Q4). Pode-se perceber que os resultados alcançados apresentam uma concordância satisfatória com as respostas obtidas na pesquisa de Pires et al. (2015) para os perfis sem e com revestimento.

Figura 5.4 - Curva temperatura x tempo: Sem revestimento



Fonte: Autor

Figura 5.5 - Curva temperatura x tempo: Com revestimento



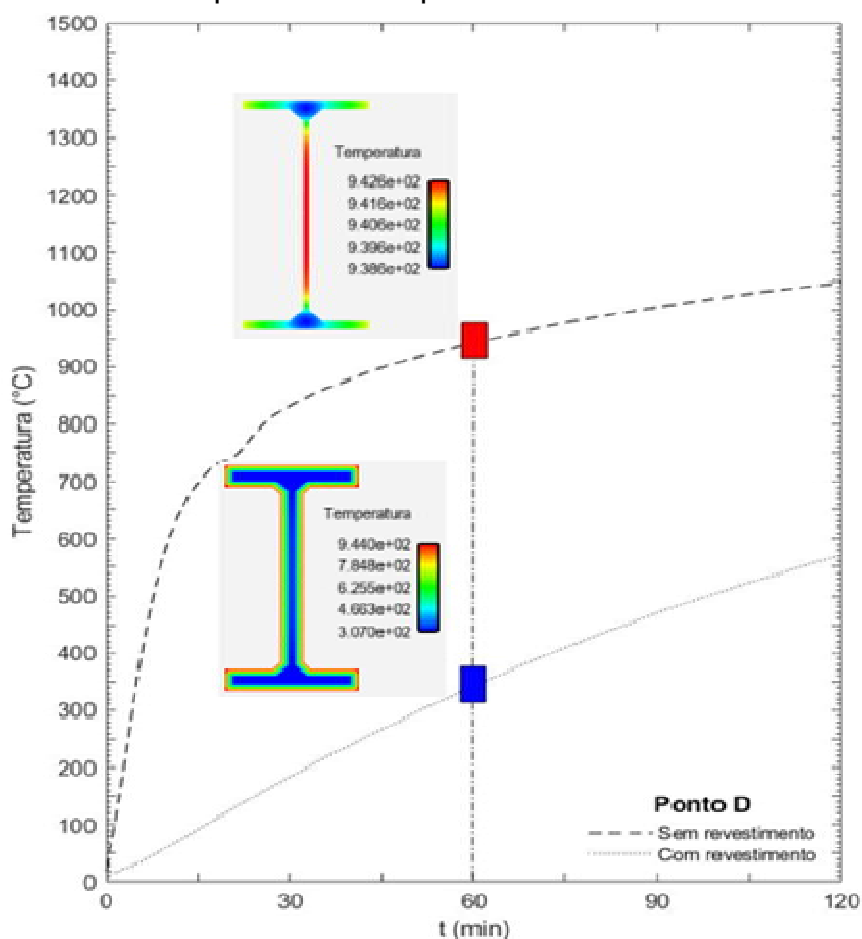
Fonte: Autor

Diferentes métodos de solução do problema transiente foram adotados nas implementações computacionais, bem como diferentes valores do parâmetro de integração no tempo  $\beta$ . Conforme destacado anteriormente, no presente trabalho foi adotado o algoritmo incremental simples com  $\beta = 2/3$ , mesmo algoritmo utilizado por Pires et al. (2015) mas com  $\beta = 0,9$ , enquanto o programa SAFIR 2014.a.1 utiliza um esquema incremental-iterativo baseado no método de Newton-Raphson e parâmetro de integração no tempo  $\beta = 0,9$ . É importante destacar que o método de solução do

sistema de equações implementado no programa foi o método direto de eliminação gaussiana com escalonamento  $LDL^T$  proposto por Bathe (1996), com a matriz de rigidez armazenada em esquema *skyline*.

Considerando somente o ponto D para comparação, é possível observar o comportamento distinto do aquecimento da seção transversal quando da utilização do material de proteção. Nota-se uma diferença significativa no valor da temperatura nesse ponto, com redução de aproximadamente 64% da temperatura no tempo de 60 minutos, como apresentado na Figura 5.6. A Tabela 5.2 apresenta a diferença para todos os pontos verificados nos tempos de 30, 60, 90 e 120 minutos. Portanto, os resultados apresentados justificam o emprego desse recurso, que visa aumentar a resistência do elemento estrutural de aço submetido a altas temperaturas.

Figura 5.6 - Curva temperatura x tempo - Ponto D: com e sem revestimento



Fonte: Autor

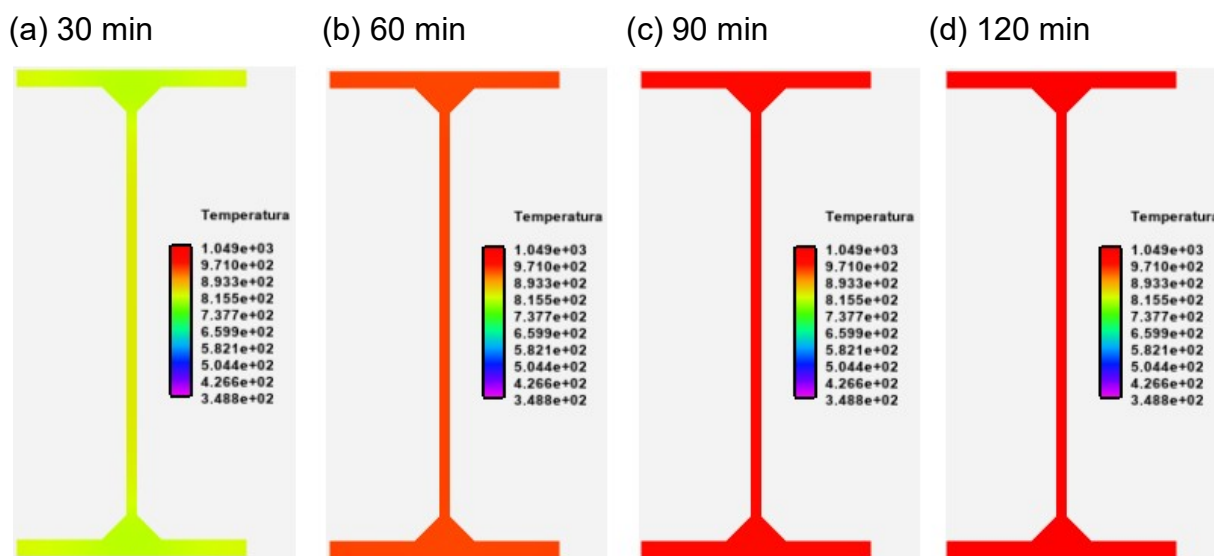
Tabela 5.2 - Temperatura em cada ponto verificado por tempo - com e sem revestimento

Temperatura (°C)								
Tempo	30 min		60 min		90 min		120 min	
Pontos	SR	CR	SR	CR	SR	CR	SR	CR
A	819,43	163,49	941,20	317,11	1003,50	444,35	1047,26	545,87
B	800,08	157,77	938,31	311,26	1001,60	438,50	1045,80	540,06
C	830,69	179,44	942,21	336,41	1004,07	463,74	1047,65	564,63
D	831,95	185,29	942,25	343,56	1004,09	470,98	1047,67	571,65

Fonte: Autor

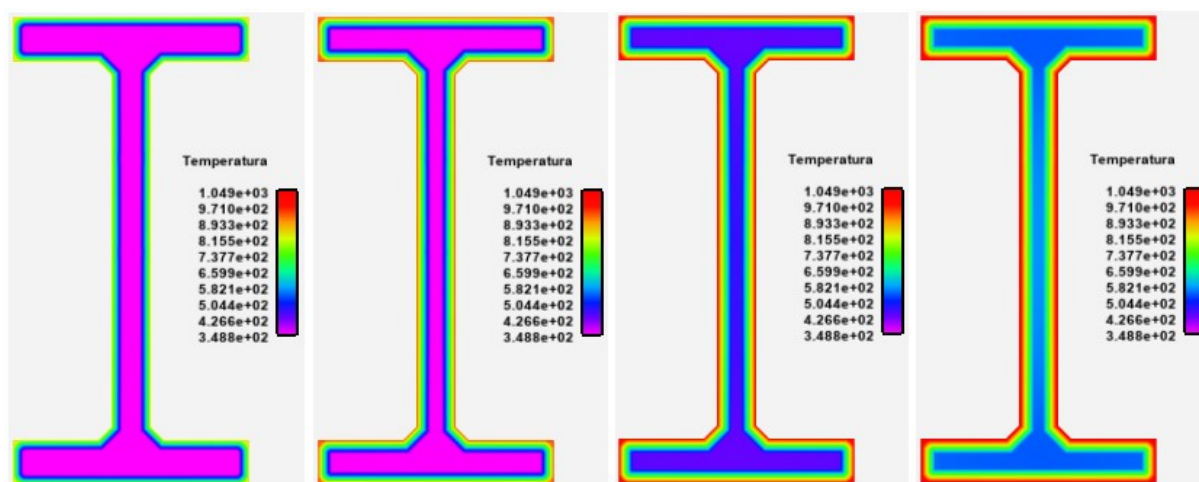
Em projetos e análises de estruturas em situação de incêndio, uma forma de visualizar os resultados de uma análise térmica é por meio do campo bidimensional de temperatura. Essa forma de pós-processamento auxilia na avaliação das regiões críticas da seção transversal do elemento estrutural expostas aos elevados gradientes térmicos provocados pelo incêndio. Nas Figuras 5.7 e 5.8 pode-se observar o campo de temperaturas nos tempos de 30, 60, 90 e 120 minutos, para o perfil IPE360 sem revestimento e com o revestimento de proteção térmica, respectivamente.

Figura 5.7 - Campo de temperaturas para o perfil IPE360 sem a proteção térmica



Fonte: Autor

Figura 5.8 - Campo de temperaturas o perfil IPE360 com a proteção térmica  
(a) 30 min (b) 60 min (c) 90 min (d) 120 min



Fonte: Autor

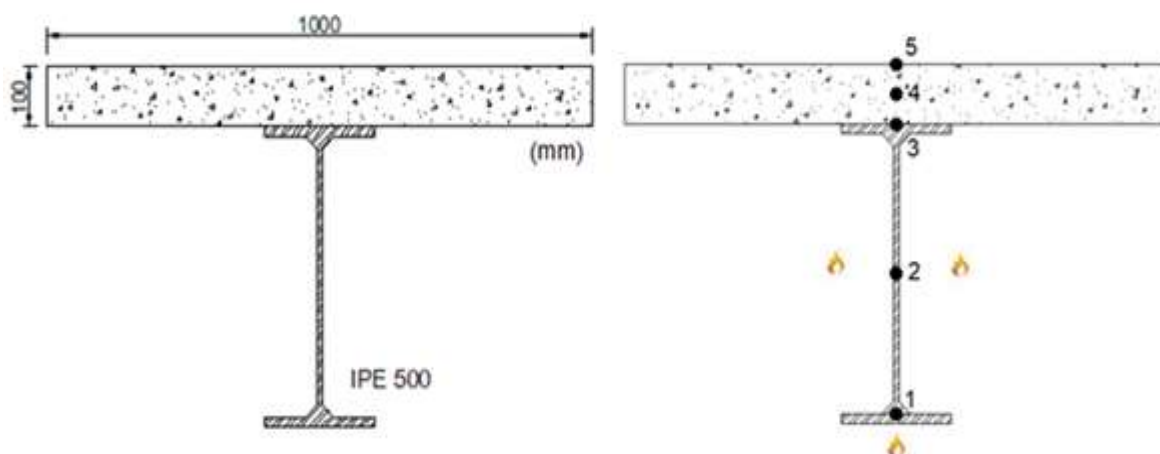
De acordo com os campos de temperaturas mostrados nas Figuras 5.7 e 5.8, pode-se mostrar novamente que a utilização de argamassa projetada como proteção passiva contribui para impedir o aumento rápido da temperatura e para aumentar a resistência do elemento estrutural de aço submetido ao fogo.

### 5.3 VIGA MISTA DE AÇO E CONCRETO

A viga mista ilustrada pela Figura 5.9 é formada pelo perfil IPE 500 e compõe uma estrutura de sete andares estudada inicialmente por Behnam e Rezvani (2014). Os autores investigaram a estrutura sujeita à ação do incêndio padrão (ISO-834-1:1999) por meio do software SAFIR 2014.a.1. A mesma viga mista também foi analisada por Pires et al. (2018) utilizando-se o módulo computacional CS-ASA/FA. A ação uniforme do fogo representada pela curva de incêndio padrão também foi considerada. Os resultados expostos pelos autores dos dois trabalhos são utilizados para comparação.

As propriedades térmicas do concreto, neste exemplo constituído de agregado silicoso, e emissividade resultante foram consideradas com base na norma NBR 15200:2012. Para as análises adotou-se o coeficiente de transferência de calor igual a 35 W/m°C, a umidade de 3% e o topo da laje em condição ambiente de temperatura (20 °C).

Figura 5.9 - Seção transversal da viga mista e pontos estudados ao longo da seção  
 (a) Seção transversal (b) Pontos avaliados

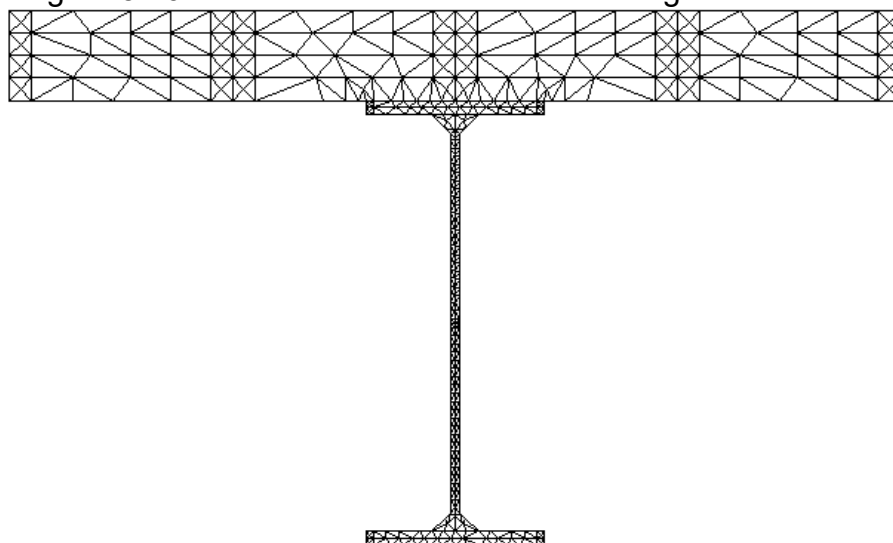


Fonte: Autor

Na Figura 5.10 é apresentada a discretização da malha estruturada com 1248 elementos finitos triangulares lineares na seção transversal da viga mista aço-concreto.

A curva temperatura versus tempo para os cinco pontos em estudo (Figura 5.9) é mostrada na Figura 5.11. Novamente é possível observar a boa concordância entre os resultados obtidos no presente trabalho e os resultados fornecidos por Behnam e Rezvani (2014) e Pires et al. (2018).

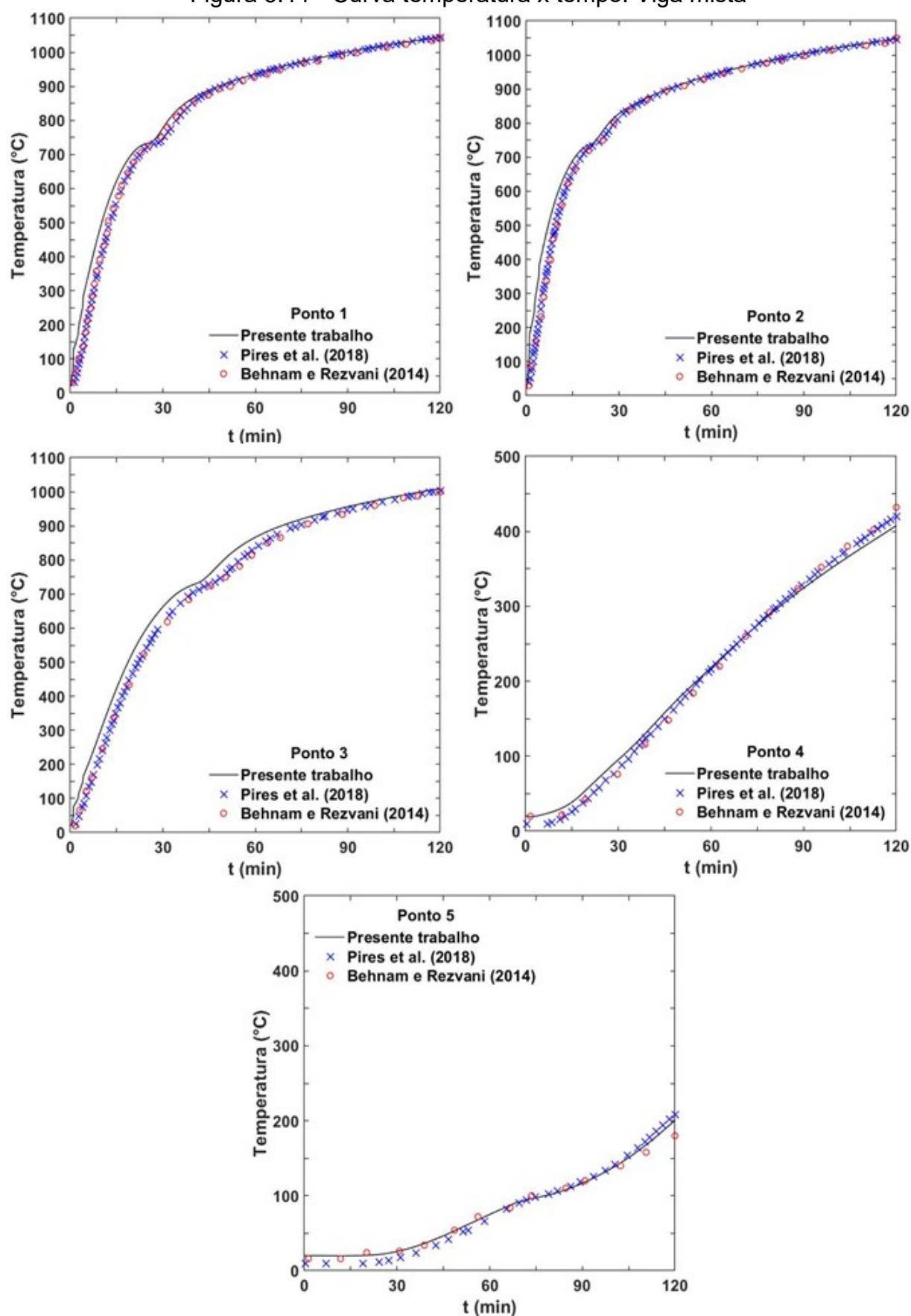
Figura 5.10 - Malha de elementos finitos triangulares lineares



Fonte: Autor



Figura 5.11 - Curva temperatura x tempo: Viga mista

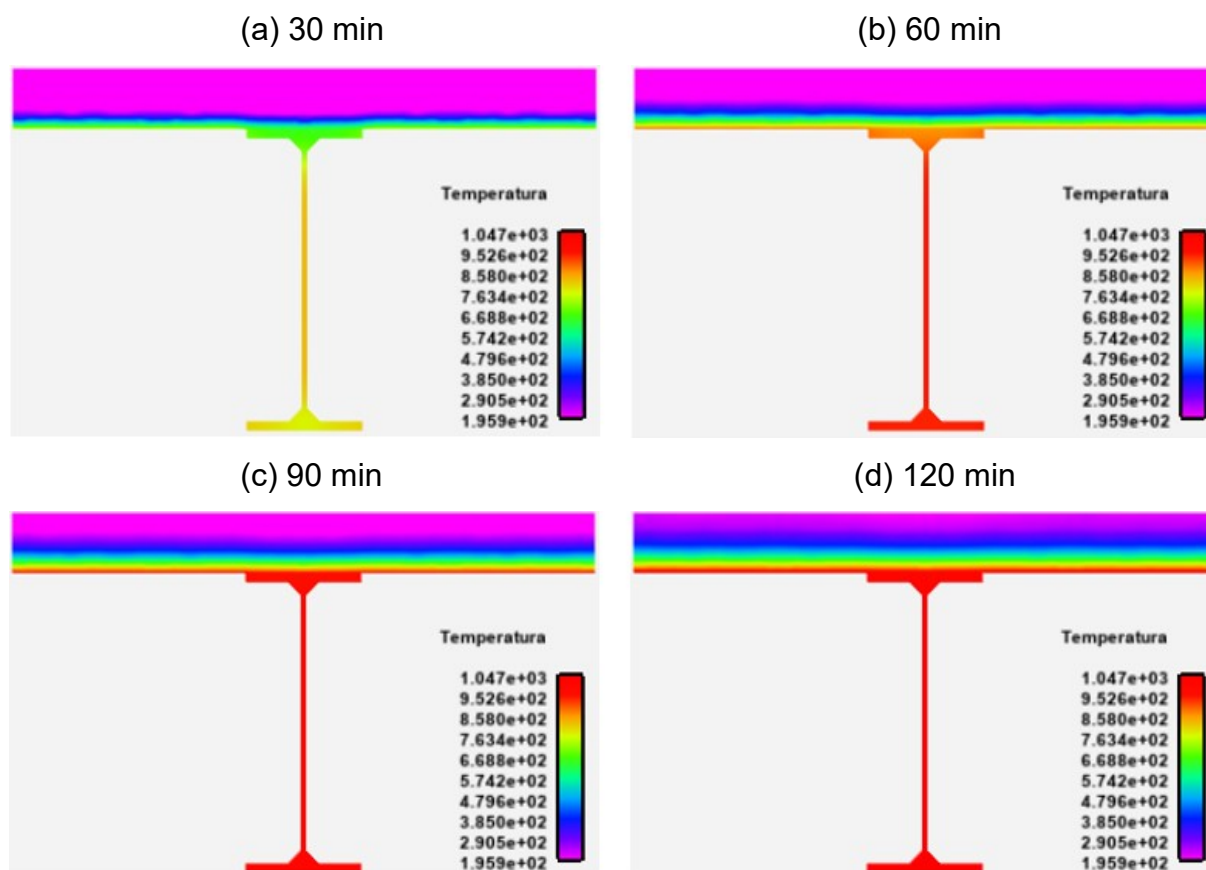


Fonte: Autor



Para facilitar a compreensão e visualização de resultados, na Figura 5.12 é ilustrado o campo de temperaturas nos tempos de 30, 60, 90 e 120 minutos, para a viga mista de aço-concreto.

Figura 5.12 - Campo de temperaturas para a viga mista de aço-concreto



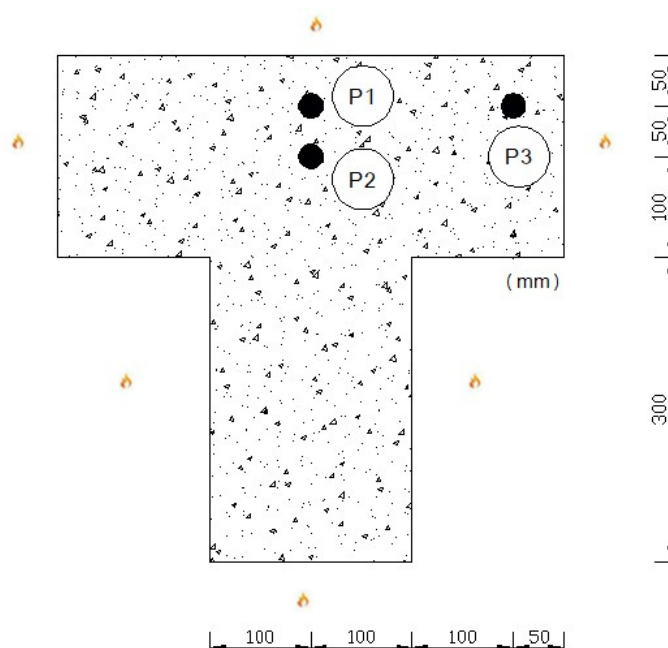
Fonte: Autor

De acordo com as curvas de temperatura versus tempo apresentadas na Figura 5.11 para pontos ao longo da seção transversal e dos campos de temperaturas mostrados na Figura 5.12, observa-se que, de um modo geral, a temperatura de todo o sistema misto é crescente com o tempo. Nota-se que, a temperatura na área do perfil I de aço varia pouco e esse comportamento se deve a sua elevada condutividade térmica. Em contrapartida, devido à baixa condutividade térmica do concreto, as temperaturas na laje são menores em comparação com as temperaturas medidas no perfil de aço.

#### 5.4 PILAR DE CONCRETO EM T

O quarto exemplo trata-se de uma seção transversal de um pilar de concreto em forma de T analisado inicialmente no trabalho de Neves et al. (2021) por meio do *software* NASEN. Todas as faces da estrutura estão expostas ao incêndio padrão, conforme ilustra a Figura 5.13. As propriedades térmicas do concreto foram consideradas com base na norma NBR 15200:2012. Adota-se uma emissividade resultante no valor de 0,8 e o coeficiente de convecção para as faces expostas ao incêndio igual a  $12 \text{ W/m}^2\text{°C}$ . Observa-se que Neves et al. (2021) não disponibilizaram o tipo de agregado empregado na análise e a porcentagem de umidade do concreto. No presente trabalho é utilizado o agregado silicoso e umidade de 3%. Para o estudo do pilar foram considerados os três pontos apresentados na Figura 5.13. Os resultados publicados por Neves et al. (2021) são utilizados para comparação.

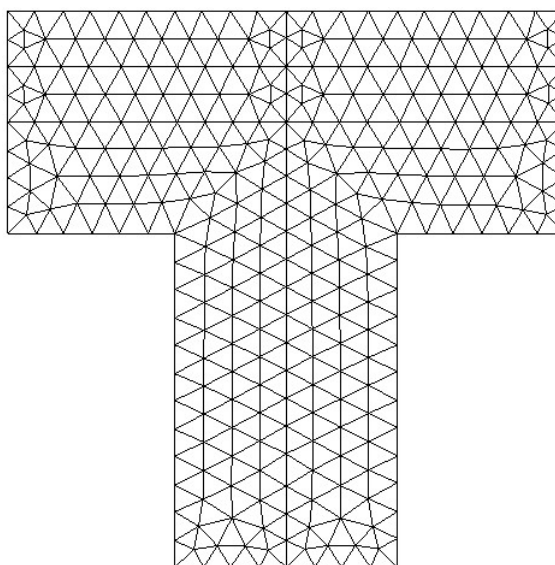
Figura 5.13 - Seção transversal do pilar T e pontos estudados ao longo da seção



Fonte: Autor

Na Figura 5.14 é mostrada a discretização da malha estruturada com 560 elementos finitos triangulares lineares na seção transversal do pilar de concreto.

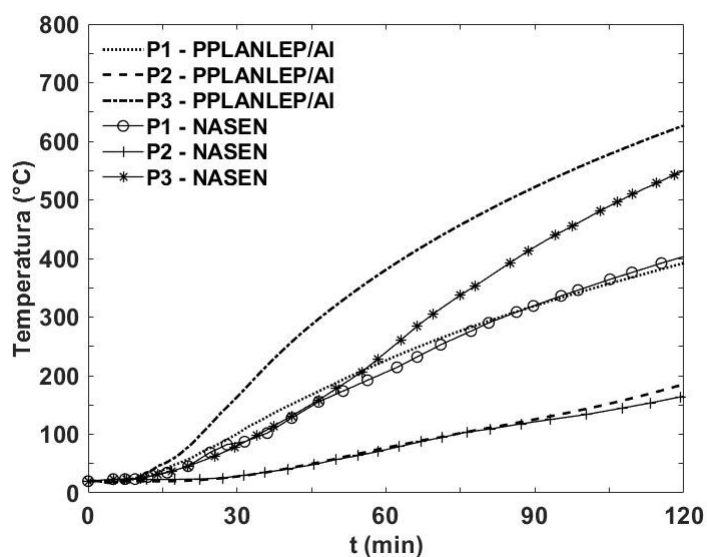
Figura 5.14 - Malha de elementos finitos triangulares lineares



Fonte: Autor

A curva temperatura versus tempo para os três pontos em estudo (Figura 5.13) é apresentada na Figura 5.15. Novamente é possível observar a boa concordância entre os resultados obtidos no presente trabalho e os resultados fornecidos por Neves et al. (2021), com exceção do ponto P3, onde observa-se certa divergência no resultado.

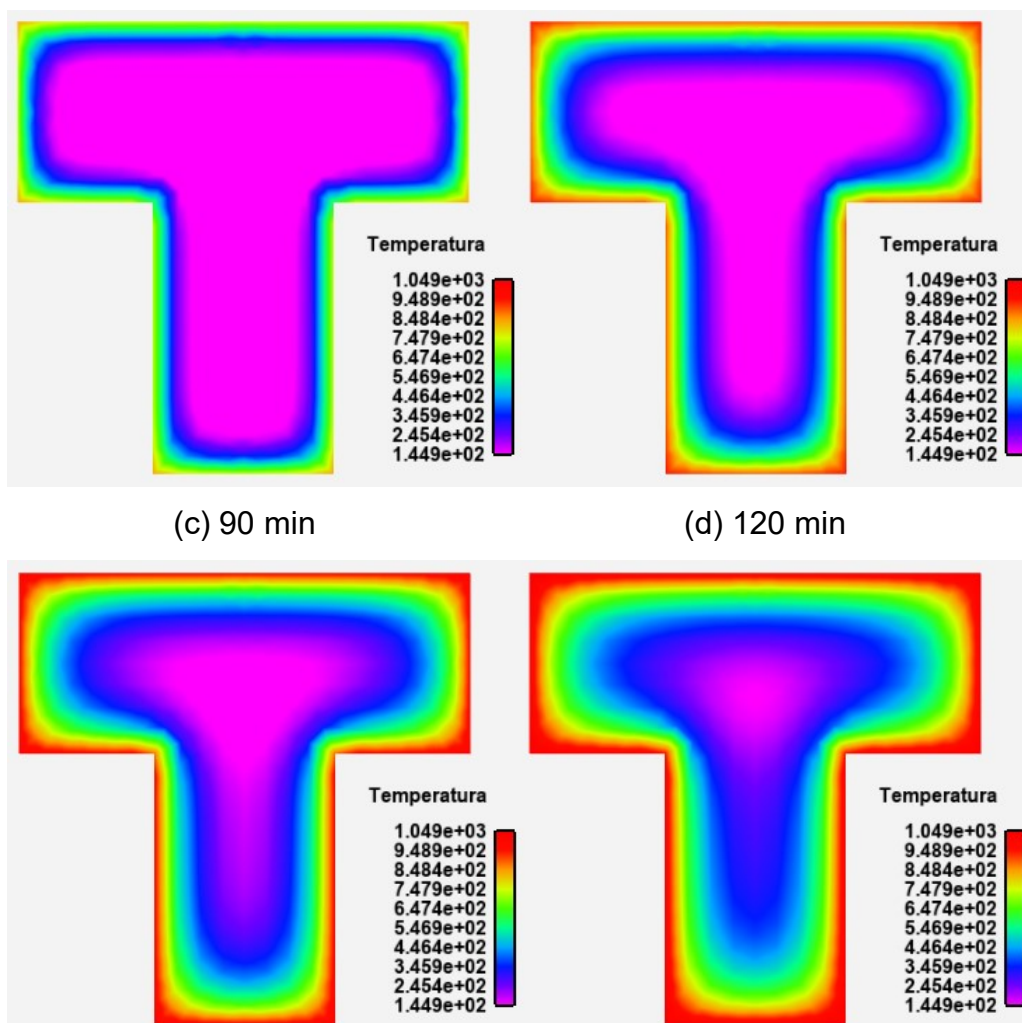
Figura 5.15 - Curva temperatura x tempo: Pilar de concreto T



Fonte: Autor

Para uma melhor compreensão do comportamento térmico do pilar de concreto e também maior clareza na visualização dos resultados, é ilustrado na Figura 5.16 o campo de temperaturas nos tempos de 30, 60, 90 e 120 minutos.

Figura 5.16 - Campo de temperaturas para o pilar de concreto T  
(a) 30 min (b) 60 min



Fonte: Autor

De acordo com os campos de temperaturas apresentados pela Figura 5.16 observa-se que no tempo analisado, 120 minutos, o núcleo da seção transversal do pilar T de concreto permanece com temperaturas bem mais amenas se comparado com a superfície. Isso é devido, principalmente, ao baixo valor de condutividade térmica e ao elevado valor de calor específico do material concreto.

## 6 CONCLUSÕES

Os materiais que constituem os elementos estruturais (aço e concreto) quando submetido a elevadas temperaturas, apresentam uma degradação progressiva das suas propriedades mecânicas, havendo probabilidade elevada de ocorrer o colapso prematuro da estrutura. Por conseguinte, é fundamental realizar um controle e prevenção de elementos que compõem a estrutura em relação ao incêndio.

A análise dos elementos estruturais sob ação térmica exige o conhecimento da evolução dos campos de temperatura em seu interior para avaliação transiente da perda de resistência e rigidez devido ao aumento da temperatura. Dessa forma, o presente trabalho teve como objetivo a determinação do campo de temperaturas em seções transversais, com e sem revestimento de proteção, quando submetidas à situação de incêndio, por meio de uma análise térmica em regimes permanente e transiente. Para isso, foi desenvolvido um programa computacional denominado PPLANLEP/AI, escrito em linguagem Fortran 90/95. A discretização das seções transversais (malha de elementos finitos) e a etapa de pós-processamento foram efetuadas pela versão acadêmica dos softwares Mechanical APDL 2021 R1 e EnSight Standard 2021 R1, respectivamente. Ambos os softwares pertencem ao pacote ANSYS 2021 R1.

Para validar a eficiência do programa de análise térmica, exemplos numéricos foram apresentados considerando-se as seções transversais de uma chaminé industrial em regime permanente, de um perfil IPE360 sem e com revestimento térmico e de uma viga mista IPE500 sujeitos a diferentes condições de exposição ao fogo em regime transiente.

Para determinar a distribuição de temperatura não uniforme nas seções transversais, diferentes pontos nodais foram escolhidos com o objetivo de traçar a curva do aumento da temperatura em função do tempo de aquecimento. Os resultados da distribuição da temperatura em função do tempo nos exemplos apresentados foram bastantes satisfatórios quando comparados com os resultados disponíveis da literatura e pelos programas Mechanical APDL 2021 R1 e SAFIR 2014.a.1, evidenciando o bom desempenho do programa computacional desenvolvido. Além disso, foi possível verificar a versatilidade do programa em

analisar diferentes geometrias, materiais e condições de contorno por meio dos exemplos apresentados.

Para o perfil I laminado sem e com proteção térmica do tipo contorno, exposto ao fogo por todos os lados, pôde-se notar que a camada de revestimento de proteção ao fogo apresentou um bom desempenho no que diz respeito à sua função de retardar os efeitos induzidos pelo incêndio. A baixa condutividade térmica e o elevado calor específico do material de revestimento de proteção utilizado (Blaze Shield II) contribuíram para impedir o aumento rápido da temperatura no elemento estrutural e para a conservação de sua resistência quando submetido a altas temperaturas.

Para a determinação da distribuição não uniforme de temperatura na seção, a seção transversal da viga mista de aço e concreto foi exposta ao carregamento de incêndio em três faces (com exceção da face superior), e pôde-se observar que, de um modo geral, a temperatura de todo o sistema misto é crescente com o tempo. No entanto, cada parte da viga mista apresentou um campo de temperaturas não uniforme distinto.

Com relação aos resultados apresentados para o pilar de concreto de seção transversal T e comparados com os resultados obtidos no trabalho de Neves et al. (2021) percebe-se uma diferença mais acentuada no ponto P3, que é um ponto próximo da quina superior direita da seção transversal do pilar, ou seja, recebe com maior rapidez e facilidade o fluxo de calor tanto da face direita quanto da face superior. Neste cenário, observa-se que no resultado apresentado por Neves et al. (2021) para o ponto P3, a curva de temperatura x tempo leva aproximadamente 45 minutos para se diferenciar e distanciar da curva temperatura x tempo do ponto P1, que é um ponto próximo a face superior e distante das faces esquerda e direita da seção transversal do pilar de concreto, o que inicialmente configura-se uma certa inconsistência nesses resultados. Contudo, é difícil o confronto das respostas obtidas, pois no trabalho de Neves et al. (2021) não foram disponibilizados alguns dados como a porcentagem de umidade no concreto e o tipo de agregado utilizado na análise.

Por fim, conclui-se que o programa desenvolvido é eficiente para realizar a análise térmica de seções bidimensionais e pode ser utilizado como uma alternativa a outros programas computacionais difundidos na literatura.

Ao final deste trabalho, considerando o programa de análise térmica desenvolvido, sugere-se a realização de estudos relacionados com a discretização da malha, o tipo de elemento finito, o esquema de integração no tempo e o incremento de tempo. Dessa forma, com a finalidade de aprimorar ainda mais os resultados do campo de temperaturas nas seções transversais, pode-se sugerir:

- A implementação de elemento finito quadrilateral linear (Q4) com 4 nós e elementos finitos quadráticos, triangular (T6) e quadrilateral (Q8), com 6 e 8 nós, respectivamente.
- A implementação do algoritmo incremental-iterativo de Newton-Raphson para a solução do problema transiente.
- A implementação de métodos iterativos para a solução do sistema de equações, como o método de Gauss-Seidel.

Com relação à análise termo-estrutural, sugere-se implementar o módulo PPLANLEP/AI no programa computacional PPLANLEP, desenvolvido por Lavall (1996) e aprimorado por Silva (2010), para realizar a análise avançada de pórticos planos de aço com ligações semirrígidas sem situação de incêndio.

## REFERÊNCIAS BIBLIOGRÁFICAS

ASTM, 2000 - E119-200: standard methods for fire tests of building construction and materials. American Society for Testing and Materials.

Bathe, K. J., 1996. Finite Element Procedures. Prentice-Hall. New Jersey, EUA.

Behnam, B., Rezvani, F. H. Structural evaluation of tall steel moment-resisting structures in simulated horizontally traveling post earthquake fire. Journal of Performance of Constructed Facilities, 04014207(12), 2014.

Caldas, R.B., 2008. Análise Numérica de Estruturas de Aço, Concreto e Mistas em Situação de Incêndio. Tese de Doutorado, Programa de Pós-Graduação em Engenharia de Estruturas, EE/UFMG, Belo Horizonte, MG, Brasil.

Cook, R., Malkus, D., Plesha, M., 1989. Concepts and Applications of Finite Element Analysis. 3rd ed. New York: John Willey & Sons.

European Committee For Standardization – CEN, 2002. EN 1991-1-2:2002 Eurocode 1 –Basis of design and actions on structures. Part 1-2: Actions on structures – Actions on structures exposed to fire. Brussels.

European Committee for Standardization – CEN, 2004. EN 1992-1-2:2004 Eurocode 2 – Design of Concrete Structures. Part 1-2: General Rules – Structural Fire Design. Brussels.

European Committee for Standardization - EN 1993-1-2:2005. Eurocode 3: Design of Steel Structures, Parte 1.2: General Rules, Structural Fire Design.

European Committee for Standardization – CEN, 2005. EN 1994-1-2:2005 Eurocode 4 – Design of Composite Steel and Concrete Structures. Part 1-2: General Rules – Structural Fire Design. Brussels.



Fakury, Ricardo H.; Silva, Ana Lydia R. Castro e; Caldas, Rodrigo B. Dimensionamento de Elementos Estruturais de Aço e Mistos de Aço e Concreto. São Paulo: Pearson, 2016. 496 p.

FIB, F. I. D. B. Bulletin 38 - Fire Design of Concrete Structures. Lausanne, Suíça: EPFL, 2007.

Figueiredo, A. D. Concreto Reforçado com Fibras . Tese livre-docência – Escola Politécnica da Universidade de São Paulo. São Paulo, 2011.

Franssen, J. -M., 2005. SAFIR - A Thermal/Structural Program Modelling Structures under Fire. Engineering Journal AISC, 42(3):143–158.

Hughes, T. J. R., 1987. The Finite Element Method: Linear Static and Dynamics Finite Element Analysis, Prentice Hall, New Jersey, EUA.

ISO 834-1, 1999. Fire resistance tests – Elements of buildings construction, Part 1: General requirements. ISO – International Organization for Standardization. Geneva.

Landesmann, A., 2012. Refined plastic-hinge model for analysis of steel-concrete structures exposed to fire. Journal of Constructional Steel Research, vol. 71, pp. 202-209.

LAVALL, A. C. C. (1996): Uma Formulação Teórica Consistente para a Análise Não linear de Pórticos Planos pelo Método dos Elementos Finitos Considerando Barras com Imperfeições Iniciais e Tensões Residuais nas Seções Transversais. Tese de Doutorado. Escola de Engenharia de São Carlos, USP, p. 265.

Lewis, R. W., Nithiarasu, P., Seetharamu, K. N., 2004. Fundamentals of the Finite Element Method for Heat and Fluid Flow. Chichester: John Wiley & Sons.

Lie, T., 1994. Fire resistance of circular steel columns filled with bar-reinforced concrete. Journal of Structural Engineering, v. 120, n. 5, p. 1489-1509.

Mendes, C. L. Estudo Teórico sobre Perfis formados a frio em Situação de Incêndio. Dissertação de Mestrado – Escola de Engenharia de São Carlos da Universidade de São Paulo. São Paulo, 2004.

Moaveni, S., 1999. Finite Element Analysis: Theory and Application with ANSYS. New Jersey: Prentice Hall.

NBR-14323, 2013. Projeto de estruturas de aço e de estruturas mistas de aço e concreto de edifícios em situação de incêndio, ABNT – Associação Brasileira de Normas Técnicas, Rio de Janeiro.

NBR-14432, 2000. Exigências de resistência ao fogo de elementos construtivos das edificações, ABNT – Associação Brasileira de Normas Técnicas, Rio de Janeiro.

NBR-15200, 2012. Projeto de estruturas de concreto em situação de incêndio, ABNT – Associação Brasileira de Normas Técnicas, Rio de Janeiro.

NBR-6118, 2014. Projeto de Estruturas de Concreto - Procedimento, ABNT – Associação Brasileira de Normas Técnicas, Rio de Janeiro, Brasil.

NBR-6120, 1980. Cargas para o cálculo de estruturas de edificações - Procedimento, ABNT – Associação Brasileira de Normas Técnicas, Rio de Janeiro, Brasil.

Neves, N. S., Azevedo, M. S., Camargo, R. S., & Pinheiro, V. P., 2019. Análise térmica bidimensional de perfil de aço sujeita a elevadas temperaturas. In: X Encontro Científico de Física Aplicada, Vitória. Blucher.

Neves, N. S., 2021. Aplicação da formulação numérica de elementos finitos e diferenças finitas na análise térmica transiente não linear. The Journal of Engineering and Exact Sciences – jCEC, Vol. 07 N. 02 (2021).

Nunes, N. E. M., 2014. Código Computacional para Análise Térmica Tridimensional de Estruturas em Situação de Incêndio. Dissertação (Mestrado - Programa de Pós-

Graduação em Engenharia de Estruturas) – Escola de Engenharia de São Carlos da Universidade de São Paulo. São Carlos.

Pierin, I., Silva, V. P., La Rovere, H. L., 2015. Análise térmica de estruturas bidimensionais sem situação de incêndio. Revista IBRACON de Estruturas e Materiais, vol. 8, pp. 25-48.

Pierin, I., Silva, V. P., 2014. Análise térmica de estruturas bidimensionais em situação de incêndio por meio do método dos elementos finitos. Jornadas Sul Americanas de Engenharia Estrutural.

Pires, D., Barros, R.C., Lemes, I.J.M., Silveira, R.A.M., Rocha, P.A.S. Análise térmica de seções transversais via método dos elementos finitos. In: CILAMCE, 36. Anais... Rio de Janeiro, RJ, Brasil, v. 1, p. 1-19, 2015.

Pires, D., Barros, R.C., Silveira, R.A.M., Rocha, P.A.S., 2018. Behavior of steel-concrete composite cross sections exposed to fire. Revista Escola de Minas, vol. 71(2), pp.149-157

Ribeiro, J. C. L. 2004. Simulação via método dos elementos finitos da distribuição tridimensional de temperatura em estruturas em situação de incêndio. Dissertação (Mestrado em Engenharia de Estruturas). Escola de Engenharia da UFMG, Universidade Federal de Minas Gerais. Belo Horizonte.

Rigobello, R., 2007. Análise Numérica de Seções Transversais e de Elementos Estruturais de Aço e Mistos de Aço e Concreto em Situação de Incêndio. Dissertação (Mestrado em Engenharia de Estruturas). Escola de Engenharia de São Carlos, Universidade de São Paulo. São Carlos, SP, Brasil.

Rigobello, R. 2011. Desenvolvimento e aplicação de código computacional para análise de estruturas de aço aporticadas em situação de incêndio. Tese (Doutorado - Programa de Pós-Graduação em Engenharia de Estruturas) – Escola de Engenharia de São Carlos da Universidade de São Paulo. São Carlos.

Real, P. M. M. 1988. Modelação por elementos finitos do comportamento térmico termo-elástico de sólidos sujeitos a elevados gradientes térmicos. Dissertação de Mestrado. Faculdade de Engenharia da Universidade do Porto. Porto – Portugal.

Reis, A. S., 2011 - Determinação de Cenários de Incêndio em Edifícios. Tese de Mestrado Universidade de Aveiro.

Silva, V. P. Estruturas de Aço em Situação de Incêndio. São Paulo: Editora Zigurate, 2001.

Silva, R.G.L., Lavall, A.C.C., Costa, R.S., Viana, H.F., “Formulation for second-order inelastic analysis of steel frames including shear deformation effect,” J. Constr. Steel Res., vol. 151, 2018, doi: 10.1016/j.jcsr.2018.09.011.

SOARES, E. M. P. “Verificação de peças usuais de concreto armado em situação de incêndio”. Dissertação de Mestrado. Curso de Pós-Graduação em Engenharia de Estruturas. Escola de Engenharia da UFMG. Belo Horizonte, 2003

Viana, H.F., Silva, R.G.L., Costa, R.S., Lavall, A.C.C., “Formulation for nonlinear dynamic analysis of steel frames considering the plastic zone method,” Eng. Struct., vol. 223, p.111197, Nov. 2020, doi: 10.1016/j.engstruct.2020.111197.

Xiong, M.-X., Wang, Y., Liew, J. R., 2016. Evaluation on thermal behavior of concrete-filled steel tubular columns based on modified finite difference method. Advances in Structural Engineering, SAGE Publications Sage UK: London, England, v. 19, n. 5, p. 746–761.

Xu, Y.-Y. & Wu, B. (2009). Fire resistance of reinforced concrete columns with I-, t-, and+- shaped cross-sections. Fire Safety Journal, Elsevier, v. 44, n. 6, p. 869–880.

Zandi, Y., Burnaz, O. & Durmuş, A. (2012). Determining the temperature distributions of fire exposed reinforced concrete cross-sections with different methods. Research Journal of Environmental and Earth Sciences, Maxwell Science Publishing, v. 4, n. 8, p. 782–788.

## APÊNDICE

Neste apêndice é apresentado o código fonte do módulo PPLANLEP/AI implementado, como já informado, em linguagem Fortran 90/95. O código fonte mostrado é referente ao exemplo da viga mista de aço e concreto abordado na seção 5.3.

### APÊNDICE A - PROGRAMA PRINCIPAL

```

program PPLANLEPAI
implicit none
write(*,*) 'running Program: PPLANLEPAI'
write(*,*) 'running input ..... '
call input
write(*,*) 'running preprocessor .....'
call preprocessor
write(*,*) 'running solver .....'
call solver
write (*,*) "End Program: PPLANLEPAI"
end program PPLANLEPAI

```

### APÊNDICE B - MÓDULOS

```

module modinput
implicit none
real*8      :: timetot, dt, alpha = 0.66666d0
integer      :: numnp,nume,nummat,nfiles,nume_partc,nume_partv
real*8 , allocatable :: x(:),y(:),temp_presc(:),flux_presc(:),flux_var(:)
real*8 , allocatable :: prop(:,:)
integer , allocatable :: incid(:,:)
integer , allocatable :: incidfluxpresc(:,:),incidfluxvar(:,:)
integer , allocatable :: mtype(:),node_presc(:)
integer , allocatable :: node_fluxvar(:,:),node_fluxpresc(:,:)
end module modinput

```

```

module modprep
implicit none
integer      :: neq, neqp, nd, ngl = 1, nnoel = 3
integer , allocatable :: id(:,:), lm(:,:)
end module modprep

```

```

module modfluxprescvar
implicit none
real*8      :: Tamb = 20.d0, hc = 25.d0, sigma = 5.67e-8
real*8      :: qp = 50.0d0, eres1 = 0.7d0, eres2 = 0.7d0, eres2p = 0.0
real*8, allocatable :: fluxpresc(:), fluxvar(:)
end module modfluxprescvar

```

```

module modmatrix
implicit none
integer (KIND=8)  :: nwk
integer (KIND=8) , allocatable :: maxa(:) , mht(:)
real*8 , allocatable :: stiff(:), sgp(:,), ske_ebe(:,)
end module modmatrix

```

```

module modloads
implicit none
real*8 , allocatable :: upresc(:), fload(:)
end module modloads

```

```

module modresults
implicit none
real*8 , allocatable :: tempnodal(:), fluxnodal(:,)
end module modresults

```

```

module modtime
implicit none
real*8 , allocatable :: temp(:)
end module modtime

```

```

module modaux
implicit none
real*8 , allocatable :: flux_x(:), flux_y(:), area(:), areanodal(:), xcount(:)
end module modaux

```

## APÊNDICE C - SUBROTINA INPUT

```

subroutine input
use modinput
use modprep , only: nnoel
implicit none
character*80 :: trash_line
character*5  :: etype, part
character*1  :: aux
integer      :: inputfile=1, outputfile=2, inode, iel, n, node, ipart
integer      :: id_part , nume_part, p
real*8       :: var_aux
open (unit=inputfile,file='model.geo',form='formatted')
read (inputfile,'(a)')trash_line
read (inputfile,*) nummat, nume, timetot, dt, nfiles

```

```

read (inputfile,'(a)')trash_line
read (inputfile,'(a)')trash_line
read (inputfile,'(a)')trash_line
read (inputfile,*) numnp
allocate (x(numnp)) ; x(:) = 0.d0
allocate (y(numnp)) ; y(:) = 0.d0
do inode = 1, numnp
read (inputfile,'(i8,2e12.5)') n, x(inode), y(inode)
enddo
allocate ( incid(ume,3) ) ; incid(:, :) = 0
allocate ( mtype(ume) ) ; mtype(:) = 0
allocate ( prop(nummat,4)) ; prop(:, :) = 0.0
do ipart = 1, nummat
read(inputfile,'(a,i8)') part, id_part
read(inputfile,*) part,prop(ipart,1),prop(ipart,2),prop(ipart,3),prop(ipart,4)
read(inputfile,*) etype
read (inputfile,*) nume_part
do iel = 1, nume_part
read (inputfile,'(4i8)') n, incid(n,1), incid(n,2), incid(n,3)
mtype (n) = ipart
enddo
enddo
read (inputfile,'(a,i8)') part, id_part
read (inputfile,*) trash_line
read (inputfile,*) etype
read (inputfile,*) nume_part
allocate ( temp_presc(numnp) ) ; temp_presc(:) = 0.0
allocate ( node_presc(numnp) ) ; node_presc(:) = 0
if(nume_part .ne. 0) then
do inode = 1, nume_part
read (inputfile,*) var_aux, node
temp_presc(node) = var_aux
node_presc(node) = node
enddo
endif
read (inputfile,'(a,i8)') part, id_part
read (inputfile,*) trash_line
read (inputfile,*) etype
read (inputfile,*) nume_partv
allocate ( flux_var(numnp) ) ; flux_var(:) = 0.0
allocate ( node_fluxvar(numnp,1) ) ; node_fluxvar(:, :) = 0
allocate ( incidfluxvar(ume,3) ) ; incidfluxvar(:, :) = 0
if(nume_partv .ne. 0) then
do inode = 1, nume_partv
read (inputfile,*) var_aux, node
flux_var(node) = var_aux
node_fluxvar(node,1) = node
enddo
do iel = 1, ume
do inode = 1, nnoel
incidfluxvar(iel,inode) = node_fluxvar(incid(iel,inode),1)

```

```

enddo
enddo
endif
read (inputfile,'(a,i8)') part, id_part
read (inputfile,*) trash_line
read (inputfile,*) etype
read (inputfile,*) nume_partc
allocate ( flux_presc(numnp) ) ; flux_presc(:) = 0.0
allocate ( node_fluxpresc(numnp,1) ) ; node_fluxpresc(:,1) = 0
allocate ( incidfluxpresc(numc,3) ); incidfluxpresc(:,1:3) = 0
if(numc_partc .ne. 0) then
do inode = 1, numc_partc
read (inputfile,*) var_aux, node
flux_presc(inode) = var_aux
node_fluxpresc(inode,1) = node
enddo
do iel = 1, numc
do inode = 1, nnoel
incidfluxpresc(iel,inode) = node_fluxpresc(iel,inode,1)
enddo
enddo
endif
end subroutine input

```

## APÊNPICE D - SUBROTINA PREPROCESSOR

```

subroutine preprocessor
use modinput , only: numnp, numc, nummat, incid, temp_presc, node_presc
use modprep
implicit none
integer :: iel, inode, igl, icount=0
nd = nnoel*ngl
allocate ( id(numnp,ngl) ) ; id(:,1) = 0
allocate ( lm(numc,nd) ) ; lm(:,1) = 0
do inode = 1, numnp
if (node_presc(inode) .ne. 0 ) then
if ( temp_presc(inode) .eq. 0.0 ) then
id(inode,1) = 0
elseif ( temp_presc(inode) .ne. 0.0 ) then
id(inode,1) = -1
endif
else
id(inode,1) = 1
endif
enddo
neq = 0
neqp = 0
do inode = 1, numnp

```



```

if (id(inode,1) .eq. 1) then
  neq = neq + 1
  id(inode,1) = neq
elseif (id(inode,1) .eq. -1) then
  neqp = neqp + 1
  id(inode,1) = -neqp
endif
enddo
do iel = 1, nume
  do inode = 1, nnoel
    lm(iel,inode) = id(incid(iel,inode),1)
  enddo
enddo
end subroutine preprocessor

```

## APÊNDICE E - SUBROTONA SOLVER

```

subroutine solver
  use modinput , only: numnp,nume,timetot,temp_presc,timetot,dt,alpha,nfiles,node_fluxvar
  use modprep
  use modfluxprescvar, only: fluxpresc,fluxvar,Tamb
  use modmatrix , only: stiff,maxa,nwk,mht,sgp,ske_ebe
  use modloads , only: fload,upresc
  use modresults, only: tempnodal,fluxnodal
  use modtime
  use modaux , only: flux_x,flux_y,area,areanodal,xcount
  implicit none
  integer :: i,kel,keq,ino,ieq,istep,nsteps,kn,ikn=0,ienergy,itime,itempnodp,itempnodt,ifluxvar,alloc_stat,inoc,f
  integer :: iter(numnp)
  real*8 :: time,energy_out
  real*8 :: tol = 0.1,erro
  integer (KIND=8) :: ls,me,mk,mm
  allocate ( fload (neq) ) ; fload = 0.d0
  allocate ( fluxpresc(numnp) ) ; fluxpresc = 0.d0
  allocate ( fluxvar(numnp) ) ; fluxvar = 0.d0
  allocate ( sgp(neq,neqp) ) ; sgp(:, :) = 0.d0
  allocate ( maxa(neq+1) ) ; maxa(:) = 0
  allocate ( mht(neq) ) ; mht(:) = 0
  allocate ( tempnodal(numnp) ) ; tempnodal(:) = 0.d0
  allocate ( fluxnodal(numnp,2) ) ; fluxnodal(:, :) = 0.d0
  allocate ( flux_x(nume) ) ; flux_x(:) = 0.d0
  allocate ( flux_y(nume) ) ; flux_y(:) = 0.d0
  allocate ( area(nume) ) ; area(:) = 0.d0
  allocate ( areanodal(numnp) ) ; areanodal(:) = 0.d0
  allocate ( xcount(numnp) ) ; xcount(:) = 0.d0
  allocate ( temp(numnp) ) ; temp(:) = 0.d0
  if (timetot .gt. 0.0) then
    allocate ( ske_ebe(6,nume) )
    ske_ebe(:, :) = 0.d0
  end if
end subroutine solver

```

```

endif
do kel = 1 , nume
ls = 1e9
do i = 1 , nd
if ( lm(kel,i) .gt. 0 ) then
if ( lm(kel,i) .lt. ls )  ls = lm(kel,i)
endif
enddo
do i = 1 , nd
if ( lm(kel,i) .gt. 0 ) then
me = lm(kel,i) - ls
if ( me .gt. mht(lm(kel,i)) )  mht(lm(kel,i)) = me
endif
enddo
enddo
maxa (1) = 1
maxa (2) = 2
mk      = 0
if ( neq .gt. 1 ) then
do keq = 2 , neq
if ( mht ( keq ) .gt. mk ) mk = mht ( keq )
maxa ( keq+1 ) = maxa ( keq ) + mht ( keq ) + 1
enddo
endif
mk  = mk + 1
nwk = maxa ( neq+1 ) - maxa ( 1 )
mm  = nwk / neq
deallocate (mht)
allocate ( stiff(nwk), stat=alloc_stat )
if (alloc_stat .eq. 0 ) then
stiff(:) = 0.d0
else
write (55,*) 'ERROR: STIFF ARRAY IS NOT ALLOCATE... STOPPING PROGRAM !'
write (*,*) 'ERROR: STIFF ARRAY IS NOT ALLOCATE... STOPPING PROGRAM !'
stop
endif
if (timetot .eq. 0.d0) then
itempnodp = 86
do ino = 1, numnp
ieq = id(ino,1)
if ( ieq .gt. 0 ) then
temp(ino) = Tamb
elseif ( ieq .lt. 0 ) then
temp(ino) = temp_presc(ino)
elseif ( ieq .eq. 0 ) then
temp(ino) = 0.d0
endif
enddo
time = 0.d0
call matrix(temp,time,sgp,stiff)
call fluxprescvar(temp,time,fluxvar,fluxpresc)

```

```

call rhs(fluxpresc,fluxvar,sgp,fload)
call colsol (stiff,fload,maxa,neq,nwk,neq+1,1,55)
call colsol (stiff,fload,maxa,neq,nwk,neq+1,2,55)
open (unit=itempnodp,file='tempnodp.txt',form='formatted')
do ino = 1, numnp
  ieq = id(ino,1)
  if ( ieq .gt. 0 ) then
    tempnodal(ino) = fload(ieq)
  elseif ( ieq .lt. 0 ) then
    tempnodal(ino) = temp_presc(ino)
  elseif ( ieq .eq. 0 ) then
    tempnodal(ino) = 0.d0
  endif
  write (itempnodp,*) ino,tempnodal(ino)
enddo
call flux_vector (tempnodal,fluxnodal)
call ensq_scalnode (tempnodal,numnp,1)
call ensq_vecnode (fluxnodal,numnp,1)
close (itempnodp)
elseif (timetot .gt. 0.d0) then
  itempnodt = 87
  ienergy = 88
  itime = 89
  ifluxvar = 90
  open (unit=itempnodt,file='tempnodt.txt',form='formatted')
  open (unit=ienergy,file='energy.txt',form='formatted')
  open (unit=itime,file='time.txt',form='formatted')
  open (unit=ifluxvar,file='fluxv.txt',form='formatted')
  nsteps = int(timetot/dt)
  kn = int(nsteps/nfiles)
  kn = max(kn,1)
  time = 0.d0
  do ino = 1, numnp
    ieq = id(ino,1)
    if ( ieq .gt. 0 ) then
      temp(ino) = Tamb
    elseif ( ieq .lt. 0 ) then
      temp(ino) = temp_presc(ino)
    elseif ( ieq .eq. 0 ) then
      temp(ino) = 0.d0
    endif
  enddo
  time = dt
  do istep = 1, nsteps
    call matrix(temp,time,sgp,stiff)
    call fluxprescvar(temp,time,fluxvar,fluxpresc)
    call rhs(fluxpresc,fluxvar,sgp,fload)
    call colsol (stiff,fload,maxa,neq,nwk,neq+1,1,55)
    call colsol (stiff,fload,maxa,neq,nwk,neq+1,2,55)
    do ino = 1, numnp
      ieq = id(ino,1)

```

```

if ( ieq .gt. 0 ) then
  tempnodal(ino) = fload(ieq) !(((1.d0/alpha)*fload(ieq))+((1.d0-(1.d0/alpha))*temp(ino)))
elseif ( ieq .lt. 0 ) then
  tempnodal(ino) = temp_presc(ino)
elseif ( ieq .eq. 0 ) then
  tempnodal(ino) = 0.d0
endif
enddo
do inoc = 1, numnp
  iter (inoc) = 0
  erro = 1.0e99
  do while (erro .gt. tol)
    iter (inoc) = iter (inoc) + 1
    erro = (abs(tempnodal(inoc) - temp(inoc)))/(abs(temp(inoc)))
    !write (*,*) erro
    if (erro .gt. tol) then
      call matrix(tempnodal,time,sgp,stiff)
      call fluxprescvar(tempnodal,time,fluxvar,fluxpresc)
      call rhs(fluxpresc,fluxvar,sgp,fload)
      call colsol (stiff,fload,maxa,neq,nwk,neq+1,1,55)
      call colsol (stiff,fload,maxa,neq,nwk,neq+1,2,55)
      do ino = 1, numnp
        ieq = id(ino,1)
        if ( ieq .gt. 0 ) then
          temp(ino) = tempnodal (ino)
        elseif ( ieq .lt. 0 ) then
          temp(ino) = temp_presc(ino)
        elseif ( ieq .eq. 0 ) then
          temp(ino) = 0.d0
        endif
      enddo
      do ino = 1, numnp
        ieq = id(ino,1)
        if ( ieq .gt. 0 ) then
          tempnodal(ino) = fload(ieq) !(((1.d0/alpha)*fload(ieq))+((1.d0-(1.d0/alpha))*temp(ino)))
        elseif ( ieq .lt. 0 ) then
          tempnodal(ino) = temp_presc(ino)
        elseif ( ieq .eq. 0 ) then
          tempnodal(ino) = 0.d0
        endif
      enddo
    end if
  enddo
  if (MOD(istep-1,kn) .eq. 0) then
    ikn = ikn+1
    call flux_vector (tempnodal,fluxnodal)
    call ensd_scalnode (tempnodal,numnp,ikn)
    call ensd_vecnode (fluxnodal,numnp,ikn)
    write (itime,*) time
  endif

```

```

call energy (fload,temp,ske_ebe,energy_out)
write (ienergy,*) time,energy_out
do ino = 1, numnp
  ieq = id(ino,1)
  if ( ieq .gt. 0 ) then
    temp(ino) = (((1.d0/alpha)*tempnodal (ino))+((1.d0-(1.d0/alpha))*temp(ino)))
  elseif ( ieq .lt. 0 ) then
    temp(ino) = temp_presc(ino)
  elseif ( ieq .eq. 0 ) then
    temp(ino) = 0.d0
  endif
enddo
write (itempnodt,*) temp(76) !, time
time = time + dt
enddo
do f = 1, numnp
  if(node_fluxvar(f,1) .ne. 0) then
    write (ifluxvar,*) f,fluxvar(f)
  endif
enddo
close (itempnodt)
close (ienergy)
close (itime)
close (ifluxvar)
endif
end subroutine solver

```

## APÊNDICE F - SUBROTINA MATRIX

```

subroutine matrix(tempsr,timesr,sgpsr,stiffr)
use modinput , only: x,y,numnp,nume,nummat,incid,incidfluxvar,mtype,prop,timetot,dt,alpha
use modprep
use modfluxprescvar , only: Tamb,hc,sigma,eres1,eres2,eres2p
use modmatrix , only: nwk,maxa,ske_ebe
implicit none
integer :: i,j,ieq,iel,kel,keq,jnd,ieq,node1,node2,node3,lmaux(3),imat
real*8 :: x1,x2,x3,y1,y2,y3,x21,y21,x31,y31,x32,x13,y23,y12,xf1,yf1,xf2,yf2,xf3,yf3,L12,L23,L13
real*8 :: area,factor,xkx,xky,cp,density,factor1,factor2,factorf,Tb,Tg,Ta,timesr,heq
real*8 :: sx(3,3),sm(3,3),ske((nd**2 + nd)/2),sf(3,3)
real*8 :: tempsr(numnp),stiffr(nwk),sgpsr(neq,neqp)
stiffr(:) = 0.d0
sgpsr(:, :) = 0.d0
if(timetot .gt. 0.d0) then
  do iel = 1, nume
    node1 = incid(iel,1)
    node2 = incid(iel,2)
    node3 = incid(iel,3)
    x1 = x ( node1 )
    x2 = x ( node2 )

```

```

x3 = x ( node3 )
y1 = y ( node1 )
y2 = y ( node2 )
y3 = y ( node3 )
x21 = x2 - x1
y21 = y2 - y1
y31 = y3 - y1
x31 = x3 - x1
area = 0.5d0 * ( x21*y31 - x31*y21 )
if ( area .le. 0.d0 ) then
write(55,*)'ERROR AREA OF ELEMENT IS NOT POSITIVE',iel
stop
endif
x32 = x3 - x2
x13 = x1 - x3
y23 = y2 - y3
y12 = y1 - y2
factor = .25d0 / area
imat = mtype(iel)
Tb = (tempsr(node1) + tempsr(node2) + tempsr(node3))/3.d0
Tg = (20.d0 + (345.d0*log10((8.d0*(timesr/60.d0))+1.d0)))
if((incidfluxvar(iel,1) .ne. 0) .and. (incidfluxvar(iel,2) .ne. 0) .and. (incidfluxvar(iel,3) .eq. 0)) then
xf1 = x(incidfluxvar(iel,1))
yf1 = y(incidfluxvar(iel,1))
xf2 = x(incidfluxvar(iel,2))
yf2 = y(incidfluxvar(iel,2))
L12 = (((((xf2 - xf1)**2) + ((yf2 - yf1)**2))*0.5)
Ta = (tempsr(node1) + tempsr(node2))/2.d0
if(imat .eq. 1) then
heq = (hc + (sigma * eres1 * ((Tg + 273.d0) + (Ta + 273.d0)) * (((Tg + 273.d0)**2) + ((Ta + 273.d0)**2))))
elseif(imat .eq. 2) then
heq = (hc + (sigma * eres2 * ((Tg + 273.d0) + (Ta + 273.d0)) * (((Tg + 273.d0)**2) + ((Ta + 273.d0)**2))))
endif
factorf = heq * L12
sf(1,1) = factorf * 0.33333d0 !(1/3)
sf(1,2) = factorf * 0.16666d0 !(1/6)
sf(1,3) = 0.d0
sf(2,2) = factorf * 0.33333d0 !(1/3)
sf(2,3) = 0.d0
sf(3,3) = 0.d0
sf(2,1) = sf(1,2)
sf(3,1) = sf(1,3)
sf(3,2) = sf(2,3)
if(imat .eq. 1) then
if((Tb .ge. 20.0) .and. (Tb .lt. 800.0)) then
xkx = 54.d0 - (3.33e-2 * Tb)
xky = 54.d0 - (3.33e-2 * Tb)
elseif((Tb .ge. 800.0) .and. (Tb .le. 1200.0)) then
xkx = 27.3d0
xky = 27.3d0
endif
endif

```

```

elseif(imat .eq. 2) then
  xkx = (1.36d0 - (0.136d0 * (Tb/100.d0)) + (0.0057d0 * ((Tb/100.d0)**2))) !0.043d0
  xky = (1.36d0 - (0.136d0 * (Tb/100.d0)) + (0.0057d0 * ((Tb/100.d0)**2))) !0.043d0
endif
sx(1,1) = factor * ( xkx*y23*y23 + xky*x32*x32 )
sx(1,2) = factor * ( xkx*y23*y31 + xky*x32*x13 )
sx(1,3) = factor * ( xkx*y23*y12 + xky*x32*x21 )
sx(2,2) = factor * ( xkx*y31*y31 + xky*x13*x13 )
sx(2,3) = factor * ( xkx*y31*y12 + xky*x13*x21 )
sx(3,3) = factor * ( xkx*y12*y12 + xky*x21*x21 )
sx(2,1) = sx(1,2)
sx(3,1) = sx(1,3)
sx(3,2) = sx(2,3)
if(imat .eq. 1) then
  density = 7850.d0
  if((Tb .ge. 20.0) .and. (Tb .lt. 600.0)) then
    cp = 425.d0 + (7.73e-1 * Tb) - (1.69e-3 * (Tb**2)) + (2.22e-6 * (Tb**3))
  elseif((Tb .ge. 600.0) .and. (Tb .lt. 735.0)) then
    cp = 666.d0 + (13002.d0/(738.d0 - Tb))
  elseif((Tb .ge. 735.0) .and. (Tb .lt. 900.0)) then
    cp = 545.d0 + (17820.d0/(Tb - 731.d0))
  elseif((Tb .ge. 900.0) .and. (Tb .le. 1200.0)) then
    cp = 650.d0
  endif
  elseif(imat .eq. 2) then
    if((Tb .ge. 20.0) .and. (Tb .le. 115.0)) then
      density = 2500.d0 !240.d0
    elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then
      density = (2500.d0 * (1.d0 - (0.02d0 * ((Tb - 115.d0)/85.d0))))
    elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
      density = (2500.d0 * (0.98d0 - (0.03d0 * ((Tb - 200.d0)/200.d0))))
    elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
      density = (2500.d0 * (0.95d0 - (0.07d0 * ((Tb - 400.d0)/800.d0))))
    endif
    if((Tb .ge. 20.0) .and. (Tb .le. 100.0)) then
      cp = 900.d0 !2093.d0
    elseif((Tb .gt. 100.0) .and. (Tb .le. 115.0)) then
      cp = 2020.d0
    elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then
      cp = (2020.d0 - (12.d0 * (Tb - 115.d0)))
    elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
      cp = (1000.d0 + ((Tb - 200.d0)/2.d0))
    elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
      cp = 1100.d0
    endif
  endif
  factor1 = cp*density*area/6.d0
  factor2 = cp*density*area/12.d0
  sm(1,1) = factor1
  sm(1,2) = factor2
  sm(1,3) = factor2

```

```

sm(2,2) = factor1
sm(2,3) = factor2
sm(3,3) = factor1
sm(2,1) = sm(1,2)
sm(3,1) = sm(1,3)
sm(3,2) = sm(2,3)
ske (1) = (sx(1,1) + sf(1,1)) + (sm(1,1)/(dt * alpha))
ske (2) = (sx(1,2) + sf(1,2)) + (sm(1,2)/(dt * alpha))
ske (3) = (sx(2,2) + sf(2,2)) + (sm(2,2)/(dt * alpha))
ske (4) = (sx(1,3) + sf(1,3)) + (sm(1,3)/(dt * alpha))
ske (5) = (sx(2,3) + sf(2,3)) + (sm(2,3)/(dt * alpha))
ske (6) = (sx(3,3) + sf(3,3)) + (sm(3,3)/(dt * alpha))
ske_ebe (1,iel) = (sx(1,1) + sf(1,1))
ske_ebe (2,iel) = (sx(1,2) + sf(1,2))
ske_ebe (3,iel) = (sx(2,2) + sf(2,2))
ske_ebe (4,iel) = (sx(1,3) + sf(1,3))
ske_ebe (5,iel) = (sx(2,3) + sf(2,3))
ske_ebe (6,iel) = (sx(3,3) + sf(3,3))
elseif((incidfluxvar(iel,1) .eq. 0) .and. (incidfluxvar(iel,2) .ne. 0) .and. (incidfluxvar(iel,3) .ne. 0)) then

xf2 = x(incidfluxvar(iel,2))
yf2 = y(incidfluxvar(iel,2))
xf3 = x(incidfluxvar(iel,3))
yf3 = y(incidfluxvar(iel,3))
L23 = (((xf3 - xf2)**2) + ((yf3 - yf2)**2))*0.5)
Ta = (tempsr(node2) + tempsr(node3))/2.d0
if(imat .eq. 1) then
heq = (hc + (sigma * eres1 * ((Tg + 273.d0) + (Ta + 273.d0)) * (((Tg + 273.d0)**2) + ((Ta + 273.d0)**2))))
elseif(imat .eq. 2) then
heq = (hc + (sigma * eres2 * ((Tg + 273.d0) + (Ta + 273.d0)) * (((Tg + 273.d0)**2) + ((Ta + 273.d0)**2))))
endif
factorf = heq * L23
sf(1,1) = 0.d0
sf(1,2) = 0.d0
sf(1,3) = 0.d0
sf(2,2) = factorf * 0.33333d0 !(1/3)
sf(2,3) = factorf * 0.16666d0 !(1/6)
sf(3,3) = factorf * 0.33333d0 !(1/3)
sf(2,1) = sf(1,2)
sf(3,1) = sf(1,3)
sf(3,2) = sf(2,3)
if(imat .eq. 1) then
if((Tb .ge. 20.0) .and. (Tb .lt. 800.0)) then
xkx = 54.d0 - (3.33e-2 * Tb)
xky = 54.d0 - (3.33e-2 * Tb)
elseif((Tb .ge. 800.0) .and. (Tb .le. 1200.0)) then
xkx = 27.3d0
xky = 27.3d0
endif
elseif(imat .eq. 2) then
xkx = (1.36d0 - (0.136d0 * (Tb/100.d0)) + (0.0057d0 * ((Tb/100.d0)**2))) !0.043d0

```



```

xky = (1.36d0 - (0.136d0 * (Tb/100.d0)) + (0.0057d0 * ((Tb/100.d0)**2))) !0.043d0
endif

sx(1,1) = factor * ( xkx*y23*y23 + xky*x32*x32 )
sx(1,2) = factor * ( xkx*y23*y31 + xky*x32*x13 )
sx(1,3) = factor * ( xkx*y23*y12 + xky*x32*x21 )
sx(2,2) = factor * ( xkx*y31*y31 + xky*x13*x13 )
sx(2,3) = factor * ( xkx*y31*y12 + xky*x13*x21 )
sx(3,3) = factor * ( xkx*y12*y12 + xky*x21*x21 )
sx(2,1) = sx(1,2)
sx(3,1) = sx(1,3)
sx(3,2) = sx(2,3)
if(imat .eq. 1) then
density = 7850.d0
if((Tb .ge. 20.0) .and. (Tb .lt. 600.0)) then
cp = 425.d0 + (7.73e-1 * Tb) - (1.69e-3 * (Tb**2)) + (2.22e-6 * (Tb**3))
elseif((Tb .ge. 600.0) .and. (Tb .lt. 735.0)) then
cp = 666.d0 + (13002.d0/(738.d0 - Tb))
elseif((Tb .ge. 735.0) .and. (Tb .lt. 900.0)) then
cp = 545.d0 + (17820.d0/(Tb - 731.d0))
elseif((Tb .ge. 900.0) .and. (Tb .le. 1200.0)) then
cp = 650.d0
endif
elseif(imat .eq. 2) then
if((Tb .ge. 20.0) .and. (Tb .le. 115.0)) then
density = 2500.d0 !240.d0
elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then
density = (2500.d0 * (1.d0 - (0.02d0 * ((Tb - 115.d0)/85.d0))))
elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
density = (2500.d0 * (0.98d0 - (0.03d0 * ((Tb - 200.d0)/200.d0))))
elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
density = (2500.d0 * (0.95d0 - (0.07d0 * ((Tb - 400.d0)/800.d0))))
endif
if((Tb .ge. 20.0) .and. (Tb .le. 100.0)) then
cp = 900.d0 !2093.d0
elseif((Tb .gt. 100.0) .and. (Tb .le. 115.0)) then
cp = 2020.d0
elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then
cp = (2020.d0 - (12.d0 * (Tb - 115.d0)))
elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
cp = (1000.d0 + ((Tb - 200.d0)/2.d0))
elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
cp = 1100.d0
endif
endif
factor1 = cp*density*area/6.d0
factor2 = cp*density*area/12.d0
sm(1,1) = factor1
sm(1,2) = factor2
sm(1,3) = factor2
sm(2,2) = factor1
sm(2,3) = factor2

```

```

sm(3,3) = factor1
sm(2,1) = sm(1,2)
sm(3,1) = sm(1,3)
sm(3,2) = sm(2,3)
ske (1) = (sx(1,1) + sf(1,1)) + (sm(1,1)/(dt * alpha))
ske (2) = (sx(1,2) + sf(1,2)) + (sm(1,2)/(dt * alpha))
ske (3) = (sx(2,2) + sf(2,2)) + (sm(2,2)/(dt * alpha))
ske (4) = (sx(1,3) + sf(1,3)) + (sm(1,3)/(dt * alpha))
ske (5) = (sx(2,3) + sf(2,3)) + (sm(2,3)/(dt * alpha))
ske (6) = (sx(3,3) + sf(3,3)) + (sm(3,3)/(dt * alpha))
ske_ebe (1,iel) = (sx(1,1) + sf(1,1))
ske_ebe (2,iel) = (sx(1,2) + sf(1,2))
ske_ebe (3,iel) = (sx(2,2) + sf(2,2))
ske_ebe (4,iel) = (sx(1,3) + sf(1,3))
ske_ebe (5,iel) = (sx(2,3) + sf(2,3))
ske_ebe (6,iel) = (sx(3,3) + sf(3,3))
elseif((incidfluxvar(iel,1) .ne. 0) .and. (incidfluxvar(iel,2) .eq. 0) .and. (incidfluxvar(iel,3) .ne. 0)) then
xf1 = x(incidfluxvar(iel,1))
yf1 = y(incidfluxvar(iel,1))
xf3 = x(incidfluxvar(iel,3))
yf3 = y(incidfluxvar(iel,3))
L13 = (((xf3 - xf1)**2) + ((yf3 - yf1)**2))*0.5)
Ta = (tempsr(node1) + tempsr(node3))/2.d0
if(imat .eq. 1) then
heq = (hc + (sigma * eres1 * ((Tg + 273.d0) + (Ta + 273.d0)) * (((Tg + 273.d0)**2) + ((Ta + 273.d0)**2))))
elseif(imat .eq. 2) then
heq = (hc + (sigma * eres2 * ((Tg + 273.d0) + (Ta + 273.d0)) * (((Tg + 273.d0)**2) + ((Ta + 273.d0)**2))))
endif
factorf = heq * L13
sf(1,1) = factorf * 0.33333d0 !(1/3)
sf(1,2) = 0.d0
sf(1,3) = factorf * 0.16666d0 !(1/6)
sf(2,2) = 0.d0
sf(2,3) = 0.d0
sf(3,3) = factorf * 0.33333d0 !(1/3)
sf(2,1) = sf(1,2)
sf(3,1) = sf(1,3)
sf(3,2) = sf(2,3)
if(imat .eq. 1) then
if((Tb .ge. 20.0) .and. (Tb .lt. 800.0)) then
xkx = 54.d0 - (3.33e-2 * Tb)
xky = 54.d0 - (3.33e-2 * Tb)
elseif((Tb .ge. 800.0) .and. (Tb .le. 1200.0)) then
xkx = 27.3d0
xky = 27.3d0
endif
elseif(imat .eq. 2) then
xkx = (1.36d0 - (0.136d0 * (Tb/100.d0)) + (0.0057d0 * ((Tb/100.d0)**2))) !0.043d0
xky = (1.36d0 - (0.136d0 * (Tb/100.d0)) + (0.0057d0 * ((Tb/100.d0)**2))) !0.043d0
endif
sx(1,1) = factor * ( xkx*y23*y23 + xky*x32*x32 )

```

```

sx(1,2) = factor * ( xkx*y23*y31 + xky*x32*x13 )
sx(1,3) = factor * ( xkx*y23*y12 + xky*x32*x21 )
sx(2,2) = factor * ( xkx*y31*y31 + xky*x13*x13 )
sx(2,3) = factor * ( xkx*y31*y12 + xky*x13*x21 )
sx(3,3) = factor * ( xkx*y12*y12 + xky*x21*x21 )
sx(2,1) = sx(1,2)
sx(3,1) = sx(1,3)
sx(3,2) = sx(2,3)
if(imat .eq. 1) then
  density = 7850.d0
  if((Tb .ge. 20.0) .and. (Tb .lt. 600.0)) then
    cp = 425.d0 + (7.73e-1 * Tb) - (1.69e-3 * (Tb**2)) + (2.22e-6 * (Tb**3))
  elseif((Tb .ge. 600.0) .and. (Tb .lt. 735.0)) then
    cp = 666.d0 + (13002.d0/(738.d0 - Tb))
  elseif((Tb .ge. 735.0) .and. (Tb .lt. 900.0)) then
    cp = 545.d0 + (17820.d0/(Tb - 731.d0))
  elseif((Tb .ge. 900.0) .and. (Tb .le. 1200.0)) then
    cp = 650.d0
  endif
  elseif(imat .eq. 2) then
    if((Tb .ge. 20.0) .and. (Tb .le. 115.0)) then
      density = 2500.d0 !240.d0
    elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then
      density = (2500.d0 * (1.d0 - (0.02d0 * ((Tb - 115.d0)/85.d0))))
    elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
      density = (2500.d0 * (0.98d0 - (0.03d0 * ((Tb - 200.d0)/200.d0))))
    elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
      density = (2500.d0 * (0.95d0 - (0.07d0 * ((Tb - 400.d0)/800.d0))))
    endif
    if((Tb .ge. 20.0) .and. (Tb .le. 100.0)) then
      cp = 900.d0 !2093.d0
    elseif((Tb .gt. 100.0) .and. (Tb .le. 115.0)) then
      cp = 2020.d0
    elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then
      cp = (2020.d0 - (12.d0 * (Tb - 115.d0)))
    elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
      cp = (1000.d0 + ((Tb - 200.d0)/2.d0))
    elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
      cp = 1100.d0
    endif
  endif
  factor1 = cp*density*area/6.d0
  factor2 = cp*density*area/12.d0
  sm(1,1) = factor1
  sm(1,2) = factor2
  sm(1,3) = factor2
  sm(2,2) = factor1
  sm(2,3) = factor2
  sm(3,3) = factor1
  sm(2,1) = sm(1,2)
  sm(3,1) = sm(1,3)

```

```

sm(3,2) = sm(2,3)
ske (1) = (sx(1,1) + sf(1,1)) + (sm(1,1)/(dt * alpha))
ske (2) = (sx(1,2) + sf(1,2)) + (sm(1,2)/(dt * alpha))
ske (3) = (sx(2,2) + sf(2,2)) + (sm(2,2)/(dt * alpha))
ske (4) = (sx(1,3) + sf(1,3)) + (sm(1,3)/(dt * alpha))
ske (5) = (sx(2,3) + sf(2,3)) + (sm(2,3)/(dt * alpha))
ske (6) = (sx(3,3) + sf(3,3)) + (sm(3,3)/(dt * alpha))
ske_ebe (1,iel) = (sx(1,1) + sf(1,1))
ske_ebe (2,iel) = (sx(1,2) + sf(1,2))
ske_ebe (3,iel) = (sx(2,2) + sf(2,2))
ske_ebe (4,iel) = (sx(1,3) + sf(1,3))
ske_ebe (5,iel) = (sx(2,3) + sf(2,3))
ske_ebe (6,iel) = (sx(3,3) + sf(3,3))
elseif(((incidfluxvar(iel,1) .ne. 0) .and. (incidfluxvar(iel,2) .eq. 0) .and. (incidfluxvar(iel,3) .eq. 0)) .or. ((incidfluxvar(iel,1) .eq. 0)
.and. (incidfluxvar(iel,2) .ne. 0) .and. (incidfluxvar(iel,3) .eq. 0)) .or. ((incidfluxvar(iel,1) .eq. 0) .and. (incidfluxvar(iel,2) .eq. 0)
.and. (incidfluxvar(iel,3) .ne. 0)) .or. ((incidfluxvar(iel,1) .eq. 0) .and. (incidfluxvar(iel,2) .eq. 0) .and. (incidfluxvar(iel,3) .eq. 0)))
then
if(imat .eq. 1) then
if((Tb .ge. 20.0) .and. (Tb .lt. 800.0)) then
xkx = 54.d0 - (3.33e-2 * Tb)
xky = 54.d0 - (3.33e-2 * Tb)
elseif((Tb .ge. 800.0) .and. (Tb .le. 1200.0)) then
xkx = 27.3d0
xky = 27.3d0
endif
elseif(imat .eq. 2) then
xkx = (1.36d0 - (0.136d0 * (Tb/100.d0)) + (0.0057d0 * ((Tb/100.d0)**2))) !0.043d0
xky = (1.36d0 - (0.136d0 * (Tb/100.d0)) + (0.0057d0 * ((Tb/100.d0)**2))) !0.043d0
endif
sx(1,1) = factor * ( xkx*y23*y23 + xky*x32*x32 )
sx(1,2) = factor * ( xkx*y23*y31 + xky*x32*x13 )
sx(1,3) = factor * ( xkx*y23*y12 + xky*x32*x21 )
sx(2,2) = factor * ( xkx*y31*y31 + xky*x13*x13 )
sx(2,3) = factor * ( xkx*y31*y12 + xky*x13*x21 )
sx(3,3) = factor * ( xkx*y12*y12 + xky*x21*x21 )
sx(2,1) = sx(1,2)
sx(3,1) = sx(1,3)
sx(3,2) = sx(2,3)
if(imat .eq. 1) then
density = 7850.d0
if((Tb .ge. 20.0) .and. (Tb .lt. 600.0)) then
cp = 425.d0 + (7.73e-1 * Tb) - (1.69e-3 * (Tb**2)) + (2.22e-6 * (Tb**3))
elseif((Tb .ge. 600.0) .and. (Tb .lt. 735.0)) then
cp = 666.d0 + (13002.d0/(738.d0 - Tb))
elseif((Tb .ge. 735.0) .and. (Tb .lt. 900.0)) then
cp = 545.d0 + (17820.d0/(Tb - 731.d0))
elseif((Tb .ge. 900.0) .and. (Tb .le. 1200.0)) then
cp = 650.d0
endif
elseif(imat .eq. 2) then
if((Tb .ge. 20.0) .and. (Tb .le. 115.0)) then

```

```

density = 2500.d0 !240.d0
elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then
density = (2500.d0 * (1.d0 - (0.02d0 * ((Tb - 115.d0)/85.d0))))
elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
density = (2500.d0 * (0.98d0 - (0.03d0 * ((Tb - 200.d0)/200.d0))))
elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
density = (2500.d0 * (0.95d0 - (0.07d0 * ((Tb - 400.d0)/800.d0))))
endif
if((Tb .ge. 20.0) .and. (Tb .le. 100.0)) then
cp = 900.d0 !2093.d0
elseif((Tb .gt. 100.0) .and. (Tb .le. 115.0)) then
cp = 2020.d0
elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then
cp = (2020.d0 - (12.d0 * (Tb - 115.d0)))
elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
cp = (1000.d0 + ((Tb - 200.d0)/2.d0))
elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
cp = 1100.d0
endif
endif
factor1 = cp*density*area/6.d0
factor2 = cp*density*area/12.d0
sm(1,1) = factor1
sm(1,2) = factor2
sm(1,3) = factor2
sm(2,2) = factor1
sm(2,3) = factor2
sm(3,3) = factor1
sm(2,1) = sm(1,2)
sm(3,1) = sm(1,3)
sm(3,2) = sm(2,3)
ske (1) = sx(1,1) + (sm(1,1)/(dt * alpha))
ske (2) = sx(1,2) + (sm(1,2)/(dt * alpha))
ske (3) = sx(2,2) + (sm(2,2)/(dt * alpha))
ske (4) = sx(1,3) + (sm(1,3)/(dt * alpha))
ske (5) = sx(2,3) + (sm(2,3)/(dt * alpha))
ske (6) = sx(3,3) + (sm(3,3)/(dt * alpha))
ske_ebe (1,iel) = sx(1,1)
ske_ebe (2,iel) = sx(1,2)
ske_ebe (3,iel) = sx(2,2)
ske_ebe (4,iel) = sx(1,3)
ske_ebe (5,iel) = sx(2,3)
ske_ebe (6,iel) = sx(3,3)
endif
lmaux(:) = 0.d0
do jnd = 1, nd
ieq = lm (iel,jnd)
if(ieq.gt.0) lmaux(jnd) = lm (iel,jnd)
enddo
call addban2 (stiffsr,maxa,ske,lmaux,nd,neq+1,nwk)
if (neqp.gt.0) then

```

```

do i = 1, nd
  ieq = lm(iel,i)
  if (ieq.gt.0) then
    do j = 1, nd
      jeq = lm(iel, j)
      if (jeq.lt.0) then
        jeq = -jeq
        sgpsr(ieq,jeq) = sgpsr(ieq,jeq) + sx(i,j)
      endif
    enddo
  endif
enddo
elseif(timetot .eq. 0.d0) then
  do iel = 1, nume
    node1 = incid(iel,1)
    node2 = incid(iel,2)
    node3 = incid(iel,3)
    x1 = x ( node1 )
    x2 = x ( node2 )
    x3 = x ( node3 )
    y1 = y ( node1 )
    y2 = y ( node2 )
    y3 = y ( node3 )
    x21 = x2 - x1
    y21 = y2 - y1
    x31 = x3 - x1
    area = 0.5d0 * ( x21*y31 - x31*y21 )
    if ( area .le. 0.d0 ) then
      WRITE(55,*)'ERROR AREA OF ELEMENT IS NOT POSITIVE',iel
      stop
    endif
    x32 = x3 - x2
    x13 = x1 - x3
    y23 = y2 - y3
    y12 = y1 - y2
    factor = .25d0 / area
    imat = mtype(iel)
    Tb = (tempsr(node1) + tempsr(node2) + tempsr(node3))/3.d0
    Tg = Tamb
    if(((incidfluxvar(iel,1) .ne. 0) .and. (incidfluxvar(iel,2) .ne. 0) .and. (incidfluxvar(iel,3) .eq. 0)) then
      xf1 = x(incidfluxvar(iel,1))
      yf1 = y(incidfluxvar(iel,1))
      xf2 = x(incidfluxvar(iel,2))
      yf2 = y(incidfluxvar(iel,2))
      L12 = (((xf2 - xf1)**2) + ((yf2 - yf1)**2))**0.5
      Ta = (tempsr(node1) + tempsr(node2))/2.d0
      if(imat .eq. 1) then
        heq = (hc + (sigma * eres1 * ((Tg + 273.d0) + (Ta + 273.d0)) * (((Tg + 273.d0)**2) + ((Ta + 273.d0)**2))))

```

```

if((Tb .ge. 20.0) .and. (Tb .lt. 800.0)) then
  xkx = 54.d0 - (3.33e-2 * Tb)
  xky = 54.d0 - (3.33e-2 * Tb)
elseif((Tb .ge. 800.0) .and. (Tb .le. 1200.0)) then
  xkx = 27.3d0
  xky = 27.3d0
endif
elseif(imat .eq. 2) then
  heq = (hc + (sigma * eres2p * ((Tg + 273.d0) + (Ta + 273.d0)) * (((Tg + 273.d0)**2) + ((Ta + 273.d0)**2))))
  xkx = 1.4d0
  xky = 1.4d0
endif
factorf = heq * L12
sf(1,1) = factorf * 0.33333d0 !(1/3)
sf(1,2) = factorf * 0.16666d0 !(1/6)
sf(1,3) = 0.d0
sf(2,2) = factorf * 0.33333d0 !(1/3)
sf(2,3) = 0.d0
sf(3,3) = 0.d0
sf(2,1) = sf(1,2)
sf(3,1) = sf(1,3)
sf(3,2) = sf(2,3)
sx(1,1) = factor * ( xkx*y23*y23 + xky*x32*x32 )
sx(1,2) = factor * ( xkx*y23*y31 + xky*x32*x13 )
sx(1,3) = factor * ( xkx*y23*y12 + xky*x32*x21 )
sx(2,2) = factor * ( xkx*y31*y31 + xky*x13*x13 )
sx(2,3) = factor * ( xkx*y31*y12 + xky*x13*x21 )
sx(3,3) = factor * ( xkx*y12*y12 + xky*x21*x21 )
sx(2,1) = sx(1,2)
sx(3,1) = sx(1,3)
sx(3,2) = sx(2,3)
ske (1) = sx(1,1) + sf(1,1)
ske (2) = sx(1,2) + sf(1,2)
ske (3) = sx(2,2) + sf(2,2)
ske (4) = sx(1,3) + sf(1,3)
ske (5) = sx(2,3) + sf(2,3)
ske (6) = sx(3,3) + sf(3,3)
elseif((incidfluxvar(iel,1) .eq. 0) .and. (incidfluxvar(iel,2) .ne. 0) .and. (incidfluxvar(iel,3) .ne. 0)) then
  xf2 = x(incidfluxvar(iel,2))
  yf2 = y(incidfluxvar(iel,2))
  xf3 = x(incidfluxvar(iel,3))
  yf3 = y(incidfluxvar(iel,3))
  L23 = (((xf3 - xf2)**2) + ((yf3 - yf2)**2))*0.5)
  Ta = (tempsr(node2) + tempsr(node3))/2.d0
  if(imat .eq. 1) then
    heq = (hc + (sigma * eres1 * ((Tg + 273.d0) + (Ta + 273.d0)) * (((Tg + 273.d0)**2) + ((Ta + 273.d0)**2))))
    if((Tb .ge. 20.0) .and. (Tb .lt. 800.0)) then
      xkx = 54.d0 - (3.33e-2 * Tb)
      xky = 54.d0 - (3.33e-2 * Tb)
    elseif((Tb .ge. 800.0) .and. (Tb .le. 1200.0)) then
      xkx = 27.3d0

```

```

xky = 27.3d0
endif
elseif(imat .eq. 2) then
heq = (hc + (sigma * eres2p * ((Tg + 273.d0) + (Ta + 273.d0)) * (((Tg + 273.d0)**2) + ((Ta + 273.d0)**2))))
xkx = 1.4d0
xky = 1.4d0
endif
factorf = heq * L23
sf(1,1) = 0.d0
sf(1,2) = 0.d0
sf(1,3) = 0.d0
sf(2,2) = factorf * 0.33333d0 !(1/3)
sf(2,3) = factorf * 0.16666d0 !(1/6)
sf(3,3) = factorf * 0.33333d0 !(1/3)
sf(2,1) = sf(1,2)
sf(3,1) = sf(1,3)
sf(3,2) = sf(2,3)
sx(1,1) = factor * ( xkx*y23*y23 + xky*x32*x32 )
sx(1,2) = factor * ( xkx*y23*y31 + xky*x32*x13 )
sx(1,3) = factor * ( xkx*y23*y12 + xky*x32*x21 )
sx(2,2) = factor * ( xkx*y31*y31 + xky*x13*x13 )
sx(2,3) = factor * ( xkx*y31*y12 + xky*x13*x21 )
sx(3,3) = factor * ( xkx*y12*y12 + xky*x21*x21 )
sx(2,1) = sx(1,2)
sx(3,1) = sx(1,3)
sx(3,2) = sx(2,3)
ske (1) = sx(1,1) + sf(1,1)
ske (2) = sx(1,2) + sf(1,2)
ske (3) = sx(2,2) + sf(2,2)
ske (4) = sx(1,3) + sf(1,3)
ske (5) = sx(2,3) + sf(2,3)
ske (6) = sx(3,3) + sf(3,3)
elseif((incidfluxvar(iel,1) .ne. 0) .and. (incidfluxvar(iel,2) .eq. 0) .and. (incidfluxvar(iel,3) .ne. 0)) then
xf1 = x(incidfluxvar(iel,1))
yf1 = y(incidfluxvar(iel,1))
xf3 = x(incidfluxvar(iel,3))
yf3 = y(incidfluxvar(iel,3))
L13 = (((xf3 - xf1)**2) + ((yf3 - yf1)**2))*0.5
Ta = (tempsr(node1) + tempsr(node3))/2.d0
if(imat .eq. 1) then
heq = (hc + (sigma * eres1 * ((Tg + 273.d0) + (Ta + 273.d0)) * (((Tg + 273.d0)**2) + ((Ta + 273.d0)**2))))
if((Tb .ge. 20.0) .and. (Tb .lt. 800.0)) then
xkx = 54.d0 - (3.33e-2 * Tb)
xky = 54.d0 - (3.33e-2 * Tb)
elseif((Tb .ge. 800.0) .and. (Tb .le. 1200.0)) then
xkx = 27.3d0
xky = 27.3d0
endif
elseif(imat .eq. 2) then
heq = (hc + (sigma * eres2p * ((Tg + 273.d0) + (Ta + 273.d0)) * (((Tg + 273.d0)**2) + ((Ta + 273.d0)**2))))
xkx = 1.4d0

```



```

xky = 1.4d0
endif
factorf = heq * L13
sf(1,1) = factorf * 0.33333d0 !(1/3)
sf(1,2) = 0.d0
sf(1,3) = factorf * 0.16666d0 !(1/6)
sf(2,2) = 0.d0
sf(2,3) = 0.d0
sf(3,3) = factorf * 0.33333d0 !(1/3)
sf(2,1) = sf(1,2)
sf(3,1) = sf(1,3)
sf(3,2) = sf(2,3)
sx(1,1) = factor * ( xkx*y23*y23 + xky*x32*x32 )
sx(1,2) = factor * ( xkx*y23*y31 + xky*x32*x13 )
sx(1,3) = factor * ( xkx*y23*y12 + xky*x32*x21 )
sx(2,2) = factor * ( xkx*y31*y31 + xky*x13*x13 )
sx(2,3) = factor * ( xkx*y31*y12 + xky*x13*x21 )
sx(3,3) = factor * ( xkx*y12*y12 + xky*x21*x21 )
sx(2,1) = sx(1,2)
sx(3,1) = sx(1,3)
sx(3,2) = sx(2,3)
ske (1) = sx(1,1) + sf(1,1)
ske (2) = sx(1,2) + sf(1,2)
ske (3) = sx(2,2) + sf(2,2)
ske (4) = sx(1,3) + sf(1,3)
ske (5) = sx(2,3) + sf(2,3)
ske (6) = sx(3,3) + sf(3,3)
elseif(((incidfluxvar(iel,1) .ne. 0) .and. (incidfluxvar(iel,2) .eq. 0) .and. (incidfluxvar(iel,3) .eq. 0)) .or. ((incidfluxvar(iel,1) .eq. 0)
.and. (incidfluxvar(iel,2) .ne. 0) .and. (incidfluxvar(iel,3) .eq. 0)) .or. ((incidfluxvar(iel,1) .eq. 0) .and. (incidfluxvar(iel,2) .eq. 0)
.and. (incidfluxvar(iel,3) .ne. 0)) .or. ((incidfluxvar(iel,1) .eq. 0) .and. (incidfluxvar(iel,2) .eq. 0) .and. (incidfluxvar(iel,3) .eq.
0))))then
if(imat .eq. 1) then
if((Tb .ge. 20.0) .and. (Tb .lt. 800.0)) then
xkx = 54.d0 - (3.33e-2 * Tb)
xky = 54.d0 - (3.33e-2 * Tb)
elseif((Tb .ge. 800.0) .and. (Tb .le. 1200.0)) then
xkx = 27.3d0
xky = 27.3d0
endif
elseif(imat .eq. 2) then
xkx = 1.4d0
xky = 1.4d0
endif
sx(1,1) = factor * ( xkx*y23*y23 + xky*x32*x32 )
sx(1,2) = factor * ( xkx*y23*y31 + xky*x32*x13 )
sx(1,3) = factor * ( xkx*y23*y12 + xky*x32*x21 )
sx(2,2) = factor * ( xkx*y31*y31 + xky*x13*x13 )
sx(2,3) = factor * ( xkx*y31*y12 + xky*x13*x21 )
sx(3,3) = factor * ( xkx*y12*y12 + xky*x21*x21 )
sx(2,1) = sx(1,2)
sx(3,1) = sx(1,3)

```

```

sx(3,2) = sx(2,3)
ske (1) = sx(1,1)
ske (2) = sx(1,2)
ske (3) = sx(2,2)
ske (4) = sx(1,3)
ske (5) = sx(2,3)
ske (6) = sx(3,3)
endif

lmaux(:) = 0.d0

do jnd = 1, nd
  ieq = lm (iel,jnd)
  if(ieq.gt.0) lmaux(jnd) = lm (iel,jnd)
enddo

call addban2 (stiffr,maxa,ske,lmaux,nd,neq+1,nwk)
if (neqp.gt.0) then
  do i = 1, nd
    ieq = lm(iel,i)
    if (ieq.gt.0) then
      do j = 1, nd
        jeq = lm(iel, j)
        if (jeq.lt.0) then
          jeq = -jeq
          sgpsr(ieq,jeq) = sgpsr(ieq,jeq) + sx(i,j)
        endif
      enddo
    endif
  enddo
endif

end subroutine matrix

```

## APÊNDICE G - SUBROTINA FLUXPRESCVAR

```

subroutine fluxprescvar(tempsr,timesr,fluxvarsr,fluxprescsr)
use modinput , only: x,y,numnp,nume,mtypc,incid,incidfluxpresc,incidfluxvar,nume_partc,nume_partv,alpha,dt,timetot
use modprep , only: id
use modfluxprescvar , only: Tamb,hc,sigma,eres1,eres2,eres2p,qp
implicit none
integer :: iel,imat,node1,node2,node3,ino,ieq
real*8 :: xfp1,yfp1,xfp2,yfp2,xfp3,yfp3,xfv1,yfv1,xfv2,yfv2,xfv3,yfv3,L12,L23,L13
real*8 :: x1,x2,x3,y1,y2,y3,x21,y21,x31,y31,x32,x13,y23,y12,xf1,yf1,xf2,yf2,xf3,yf3
real*8 :: area,cp,density,Ta,Tb,Tg,timesr,heq
real*8 :: fluxvarsr(numnp)
real*8 :: fluxprescsr(numnp)
real*8 :: tempsr(numnp)
fluxvarsr(:) = 0.0
fluxprescsr(:) = 0.0
if(timetot.gt. 0.d0) then

```

```

if(ume_partv .ne. 0) then
do iel = 1, nume
node1 = incid(iel,1)
node2 = incid(iel,2)
node3 = incid(iel,3)
x1 = x ( node1 )
x2 = x ( node2 )
x3 = x ( node3 )
y1 = y ( node1 )
y2 = y ( node2 )
y3 = y ( node3 )
x21 = x2 - x1
y21 = y2 - y1
y31 = y3 - y1
x31 = x3 - x1
area = 0.5d0 * ( x21*y31 - x31*y21 )
imat = mtype(iel)
Tb = (tempsr(node1) + tempsr(node2) + tempsr(node3))/3.d0
Tg = (20.d0 + (345.d0*log10((8.d0*(timesr/60.d0))+1.d0)))
if(incidfluxvar(iel,1) .ne. 0) then
xfv1 = x(incidfluxvar(iel,1))
yfv1 = y(incidfluxvar(iel,1))
endif
if(incidfluxvar(iel,2) .ne. 0) then
xfv2 = x(incidfluxvar(iel,2))
yfv2 = y(incidfluxvar(iel,2))
endif
if(incidfluxvar(iel,3) .ne. 0) then
xfv3 = x(incidfluxvar(iel,3))
yfv3 = y(incidfluxvar(iel,3))
endif
if((incidfluxvar(iel,1) .ne. 0) .and. (incidfluxvar(iel,2) .ne. 0) .and. (incidfluxvar(iel,3) .eq. 0)) then
Ta = (tempsr(node1) + tempsr(node2))/2.d0
L12 = (((xfv2 - xfv1)**2) + ((yfv2 - yfv1)**2))**0.5
if(imat .eq. 1) then
heq = (hc + (sigma * eres1 * ((Tg + 273.d0) + (Ta + 273.d0)) * (((Tg + 273.d0)**2) + ((Ta + 273.d0)**2))))
density = 7850.d0
if((Tb .ge. 20.0) .and. (Tb .lt. 600.0)) then
cp = 425.d0 + (7.73e-1 * Tb) - (1.69e-3 * (Tb**2)) + (2.22e-6 * (Tb**3))
elseif((Tb .ge. 600.0) .and. (Tb .lt. 735.0)) then
cp = 666.d0 + (13002.d0/(738.d0 - Tb))
elseif((Tb .ge. 735.0) .and. (Tb .lt. 900.0)) then
cp = 545.d0 + (17820.d0/(Tb - 731.d0))
elseif((Tb .ge. 900.0) .and. (Tb .le. 1200.0)) then
cp = 650.d0
endif
elseif(imat .eq. 2) then
heq = (hc + (sigma * eres2 * ((Tg + 273.d0) + (Ta + 273.d0)) * (((Tg + 273.d0)**2) + ((Ta + 273.d0)**2))))
if((Tb .ge. 20.0) .and. (Tb .le. 115.0)) then
density = 2500.d0 !240.d0
elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then

```

```

density = (2500.d0 * (1.d0 - (0.02d0 * ((Tb - 115.d0)/85.d0))))
elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
density = (2500.d0 * (0.98d0 - (0.03d0 * ((Tb - 200.d0)/200.d0))))
elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
density = (2500.d0 * (0.95d0 - (0.07d0 * ((Tb - 400.d0)/800.d0))))
endif
if((Tb .ge. 20.0) .and. (Tb .le. 100.0)) then
cp = 900.d0 !2093.d0
elseif((Tb .gt. 100.0) .and. (Tb .le. 115.0)) then
cp = 2020.d0
elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then
cp = (2020.d0 - (12.d0 * (Tb - 115.d0)))
elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
cp = (1000.d0 + ((Tb - 200.d0)/2.d0))
elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
cp = 1100.d0
endif
endif
if(id(incidfluxvar(iel,1),1) .gt. 0) then
fluxvarsr(incidfluxvar(iel,1)) = fluxvarsr(incidfluxvar(iel,1)) + (heq * Tg * (L12/2.d0)) + (((density * cp * area)/(alpha * dt)) *
((0.16666 * tempsr(node1)) + (0.08333 * tempsr(node2)) + (0.08333 * tempsr(node3))))
elseif(id(incidfluxvar(iel,1),1) .lt. 0) then
fluxvarsr(incidfluxvar(iel,1)) = fluxvarsr(incidfluxvar(iel,1)) + (((density * cp * area)/(alpha * dt)) * ((0.16666 * tempsr(node1)) +
(0.08333 * tempsr(node2)) + (0.08333 * tempsr(node3))))
endif
if(id(incidfluxvar(iel,2),1) .gt. 0) then
fluxvarsr(incidfluxvar(iel,2)) = fluxvarsr(incidfluxvar(iel,2)) + (heq * Tg * (L12/2.d0)) + (((density * cp * area)/(alpha * dt)) *
((0.08333 * tempsr(node1)) + (0.16666 * tempsr(node2)) + (0.08333 * tempsr(node3))))
elseif(id(incidfluxvar(iel,2),1) .lt. 0) then
fluxvarsr(incidfluxvar(iel,2)) = fluxvarsr(incidfluxvar(iel,2)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) +
(0.16666 * tempsr(node2)) + (0.08333 * tempsr(node3))))
endif
fluxvarsr(incid(iel,3)) = fluxvarsr(incid(iel,3)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) + (0.08333 *
tempsr(node2)) + (0.16666 * tempsr(node3))))
elseif((incidfluxvar(iel,1) .eq. 0) .and. (incidfluxvar(iel,2) .ne. 0) .and. (incidfluxvar(iel,3) .ne. 0)) then
Ta = (tempsr(node2) + tempsr(node3))/2.d0
L23 = (((xfv3 - xfv2)**2) + ((yfv3 - yfv2)**2))**0.5
if(imat .eq. 1) then
heq = (hc + (sigma * eres1 * ((Tg + 273.0) + (Ta + 273.0)) * (((Tg + 273.0)**2) + ((Ta + 273.0)**2))))
density = 7850.0
if((Tb .ge. 20.0) .and. (Tb .lt. 600.0)) then
cp = 425.0 + (7.73e-1 * Tb) - (1.69e-3 * (Tb**2)) + (2.22e-6 * (Tb**3))
elseif((Tb .ge. 600.0) .and. (Tb .lt. 735.0)) then
cp = 666.0 + (13002.0/(738.0 - Tb))
elseif((Tb .ge. 735.0) .and. (Tb .lt. 900.0)) then
cp = 545.0 + (17820.0/(Tb - 731.0))
elseif((Tb .ge. 900.0) .and. (Tb .le. 1200.0)) then
cp = 650.0
endif
elseif(imat .eq. 2) then
heq = (hc + (sigma * eres2 * ((Tg + 273.0) + (Ta + 273.0)) * (((Tg + 273.0)**2) + ((Ta + 273.0)**2))))

```

```

if((Tb .ge. 20.0) .and. (Tb .le. 115.0)) then
density = 2500.d0 !240.d0
elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then
density = (2500.d0 * (1.d0 - (0.02d0 * ((Tb - 115.d0)/85.d0))))
elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
density = (2500.d0 * (0.98d0 - (0.03d0 * ((Tb - 200.d0)/200.d0))))
elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
density = (2500.d0 * (0.95d0 - (0.07d0 * ((Tb - 400.d0)/800.d0))))
endif

if((Tb .ge. 20.0) .and. (Tb .le. 100.0)) then
cp = 900.d0 !2093.d0
elseif((Tb .gt. 100.0) .and. (Tb .le. 115.0)) then
cp = 2020.d0
elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then
cp = (2020.d0 - (12.d0 * (Tb - 115.d0)))
elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
cp = (1000.d0 + ((Tb - 200.d0)/2.d0))
elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
cp = 1100.d0
endif
endif

fluxvarsr(incid(iel,1)) = fluxvarsr(incid(iel,1)) + (((density * cp * area)/(alpha * dt)) * ((0.16666 * tempsr(node1)) + (0.08333 *
tempsr(node2)) + (0.08333 * tempsr(node3))))

if(id(incidfluxvar(iel,2),1) .gt. 0) then
fluxvarsr(incidfluxvar(iel,2)) = fluxvarsr(incidfluxvar(iel,2)) + (heq * Tg * (L23/2.d0)) + (((density * cp * area)/(alpha * dt)) *
((0.08333 * tempsr(node1)) + (0.16666 * tempsr(node2)) + (0.08333 * tempsr(node3))))
elseif(id(incidfluxvar(iel,2),1) .lt. 0) then
fluxvarsr(incidfluxvar(iel,2)) = fluxvarsr(incidfluxvar(iel,2)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) +
(0.16666 * tempsr(node2)) + (0.08333 * tempsr(node3))))
endif

if(id(incidfluxvar(iel,3),1) .gt. 0) then
fluxvarsr(incidfluxvar(iel,3)) = fluxvarsr(incidfluxvar(iel,3)) + (heq * Tg * (L23/2.d0)) + (((density * cp * area)/(alpha * dt)) *
((0.08333 * tempsr(node1)) + (0.08333 * tempsr(node2)) + (0.16666 * tempsr(node3))))
elseif(id(incidfluxvar(iel,3),1) .lt. 0) then
fluxvarsr(incidfluxvar(iel,3)) = fluxvarsr(incidfluxvar(iel,3)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) +
(0.08333 * tempsr(node2)) + (0.16666 * tempsr(node3))))
endif

elseif((incidfluxvar(iel,1) .ne. 0) .and. (incidfluxvar(iel,2) .eq. 0) .and. (incidfluxvar(iel,3) .ne. 0)) then
Ta = (tempsr(node1) + tempsr(node3))/2.d0
L13 = (((x fv3 - x fv1)**2) + ((y fv3 - y fv1)**2))**0.5
if(imat .eq. 1) then
heq = (hc + (sigma * eres1 * ((Tg + 273.0) + (Ta + 273.0)) * (((Tg + 273.0)**2) + ((Ta + 273.0)**2))))
density = 7850.0
if((Tb .ge. 20.0) .and. (Tb .lt. 600.0)) then
cp = 425.0 + (7.73e-1 * Tb) - (1.69e-3 * (Tb**2)) + (2.22e-6 * (Tb**3))
elseif((Tb .ge. 600.0) .and. (Tb .lt. 735.0)) then
cp = 666.0 + (13002.0/(738.0 - Tb))
elseif((Tb .ge. 735.0) .and. (Tb .lt. 900.0)) then
cp = 545.0 + (17820.0/(Tb - 731.0))
elseif((Tb .ge. 900.0) .and. (Tb .le. 1200.0)) then
cp = 650.0

```

```

endif
elseif(imat .eq. 2) then
heq = (hc + (sigma * eres2 * ((Tg + 273.0) + (Ta + 273.0)) * (((Tg + 273.0)**2) + ((Ta + 273.0)**2))))
if((Tb .ge. 20.0) .and. (Tb .le. 115.0)) then
density = 2500.d0 !240.d0
elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then
density = (2500.d0 * (1.d0 - (0.02d0 * ((Tb - 115.d0)/85.d0))))
elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
density = (2500.d0 * (0.98d0 - (0.03d0 * ((Tb - 200.d0)/200.d0))))
elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
density = (2500.d0 * (0.95d0 - (0.07d0 * ((Tb - 400.d0)/800.d0))))
endif
if((Tb .ge. 20.0) .and. (Tb .le. 100.0)) then
cp = 900.d0 !2093.d0
elseif((Tb .gt. 100.0) .and. (Tb .le. 115.0)) then
cp = 2020.d0
elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then
cp = (2020.d0 - (12.d0 * (Tb - 115.d0)))
elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
cp = (1000.d0 + ((Tb - 200.d0)/2.d0))
elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
cp = 1100.d0
endif
endif
if(id(incidfluxvar(iel,1),1) .gt. 0) then
fluxvarsr(incidfluxvar(iel,1)) = fluxvarsr(incidfluxvar(iel,1)) + (heq * Tg * (L13/2.d0)) + (((density * cp * area)/(alpha * dt)) *
((0.16666 * tempsr(node1)) + (0.08333 * tempsr(node2)) + (0.08333 * tempsr(node3))))
elseif(id(incidfluxvar(iel,1),1) .lt. 0) then
fluxvarsr(incidfluxvar(iel,1)) = fluxvarsr(incidfluxvar(iel,1)) + (((density * cp * area)/(alpha * dt)) * ((0.16666 * tempsr(node1)) +
(0.08333 * tempsr(node2)) + (0.08333 * tempsr(node3))))
endif
fluxvarsr(incid(iel,2)) = fluxvarsr(incid(iel,2)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) + (0.16666 *
tempsr(node2)) + (0.08333 * tempsr(node3))))
if(id(incidfluxvar(iel,3),1) .gt. 0) then
fluxvarsr(incidfluxvar(iel,3)) = fluxvarsr(incidfluxvar(iel,3)) + (heq * Tg * (L13/2.d0)) + (((density * cp * area)/(alpha * dt)) *
((0.08333 * tempsr(node1)) + (0.08333 * tempsr(node2)) + (0.16666 * tempsr(node3))))
elseif(id(incidfluxvar(iel,3),1) .lt. 0) then
fluxvarsr(incidfluxvar(iel,3)) = fluxvarsr(incidfluxvar(iel,3)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) +
(0.08333 * tempsr(node2)) + (0.16666 * tempsr(node3))))
endif
elseif((incidfluxvar(iel,1) .ne. 0) .and. (incidfluxvar(iel,2) .eq. 0) .and. (incidfluxvar(iel,3) .eq. 0)) then
if(imat .eq. 1) then
density = 7850.0
if((Tb .ge. 20.0) .and. (Tb .lt. 600.0)) then
cp = 425.0 + (7.73e-1 * Tb) - (1.69e-3 * (Tb**2)) + (2.22e-6 * (Tb**3))
elseif((Tb .ge. 600.0) .and. (Tb .lt. 735.0)) then
cp = 666.0 + (13002.0/(738.0 - Tb))
elseif((Tb .ge. 735.0) .and. (Tb .lt. 900.0)) then
cp = 545.0 + (17820.0/(Tb - 731.0))
elseif((Tb .ge. 900.0) .and. (Tb .le. 1200.0)) then
cp = 650.0

```

```

endif
elseif(imat .eq. 2) then
  if((Tb .ge. 20.0) .and. (Tb .le. 115.0)) then
    density = 2500.d0 !240.d0
    elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then
      density = (2500.d0 * (1.d0 - (0.02d0 * ((Tb - 115.d0)/85.d0))))
    elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
      density = (2500.d0 * (0.98d0 - (0.03d0 * ((Tb - 200.d0)/200.d0))))
    elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
      density = (2500.d0 * (0.95d0 - (0.07d0 * ((Tb - 400.d0)/800.d0))))
    endif
  if((Tb .ge. 20.0) .and. (Tb .le. 100.0)) then
    cp = 900.d0 !2093.d0
    elseif((Tb .gt. 100.0) .and. (Tb .le. 115.0)) then
      cp = 2020.d0
    elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then
      cp = (2020.d0 - (12.d0 * (Tb - 115.d0)))
    elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
      cp = (1000.d0 + ((Tb - 200.d0)/2.d0))
    elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
      cp = 1100.d0
    endif
  endif
  fluxvarsr(incidfluxvar(iel,1)) = fluxvarsr(incidfluxvar(iel,1)) + (((density * cp * area)/(alpha * dt)) * ((0.16666 * tempsr(node1)) +
  (0.08333 * tempsr(node2)) + (0.08333 * tempsr(node3))))
  fluxvarsr(incid(iel,2)) = fluxvarsr(incid(iel,2)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) + (0.16666 *
  tempsr(node2)) + (0.08333 * tempsr(node3))))
  fluxvarsr(incid(iel,3)) = fluxvarsr(incid(iel,3)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) + (0.08333 *
  tempsr(node2)) + (0.16666 * tempsr(node3))))
  elseif((incidfluxvar(iel,1) .eq. 0) .and. (incidfluxvar(iel,2) .ne. 0) .and. (incidfluxvar(iel,3) .eq. 0)) then
    if(imat .eq. 1) then
      density = 7850.0
      if((Tb .ge. 20.0) .and. (Tb .lt. 600.0)) then
        cp = 425.0 + (7.73e-1 * Tb) - (1.69e-3 * (Tb**2)) + (2.22e-6 * (Tb**3))
      elseif((Tb .ge. 600.0) .and. (Tb .lt. 735.0)) then
        cp = 666.0 + (13002.0/(738.0 - Tb))
      elseif((Tb .ge. 735.0) .and. (Tb .lt. 900.0)) then
        cp = 545.0 + (17820.0/(Tb - 731.0))
      elseif((Tb .ge. 900.0) .and. (Tb .le. 1200.0)) then
        cp = 650.0
      endif
    elseif(imat .eq. 2) then
      if((Tb .ge. 20.0) .and. (Tb .le. 115.0)) then
        density = 2500.d0 !240.d0
        elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then
          density = (2500.d0 * (1.d0 - (0.02d0 * ((Tb - 115.d0)/85.d0))))
        elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
          density = (2500.d0 * (0.98d0 - (0.03d0 * ((Tb - 200.d0)/200.d0))))
        elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
          density = (2500.d0 * (0.95d0 - (0.07d0 * ((Tb - 400.d0)/800.d0))))
        endif
      endif

```

```

if((Tb .ge. 20.0) .and. (Tb .le. 100.0)) then
  cp = 900.d0 !2093.d0
elseif((Tb .gt. 100.0) .and. (Tb .le. 115.0)) then
  cp = 2020.d0
elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then
  cp = (2020.d0 - (12.d0 * (Tb - 115.d0)))
elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
  cp = (1000.d0 + ((Tb - 200.d0)/2.d0))
elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
  cp = 1100.d0
endif
endif

fluxvarsr(incid(iel,1)) = fluxvarsr(incid(iel,1)) + (((density * cp * area)/(alpha * dt)) * ((0.16666 * tempsr(node1)) + (0.08333 *
tempsr(node2)) + (0.08333 * tempsr(node3))))
fluxvarsr(incidfluxvar(iel,2)) = fluxvarsr(incidfluxvar(iel,2)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) +
(0.16666 * tempsr(node2)) + (0.08333 * tempsr(node3))))
fluxvarsr(incid(iel,3)) = fluxvarsr(incid(iel,3)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) + (0.08333 *
tempsr(node2)) + (0.16666 * tempsr(node3))))
elseif((incidfluxvar(iel,1) .eq. 0) .and. (incidfluxvar(iel,2) .eq. 0) .and. (incidfluxvar(iel,3) .ne. 0)) then
  if(imat .eq. 1) then
    density = 7850.0
    if((Tb .ge. 20.0) .and. (Tb .lt. 600.0)) then
      cp = 425.0 + (7.73e-1 * Tb) - (1.69e-3 * (Tb**2)) + (2.22e-6 * (Tb**3))
    elseif((Tb .ge. 600.0) .and. (Tb .lt. 735.0)) then
      cp = 666.0 + (13002.0/(738.0 - Tb))
    elseif((Tb .ge. 735.0) .and. (Tb .lt. 900.0)) then
      cp = 545.0 + (17820.0/(Tb - 731.0))
    elseif((Tb .ge. 900.0) .and. (Tb .le. 1200.0)) then
      cp = 650.0
    endif
    elseif(imat .eq. 2) then
      if((Tb .ge. 20.0) .and. (Tb .le. 115.0)) then
        density = 2500.d0 !240.d0
        elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then
          density = (2500.d0 * (1.d0 - (0.02d0 * ((Tb - 115.d0)/85.d0))))
        elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
          density = (2500.d0 * (0.98d0 - (0.03d0 * ((Tb - 200.d0)/200.d0))))
        elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
          density = (2500.d0 * (0.95d0 - (0.07d0 * ((Tb - 400.d0)/800.d0))))
        endif
        if((Tb .ge. 20.0) .and. (Tb .le. 100.0)) then
          cp = 900.d0 !2093.d0
        elseif((Tb .gt. 100.0) .and. (Tb .le. 115.0)) then
          cp = 2020.d0
        elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then
          cp = (2020.d0 - (12.d0 * (Tb - 115.d0)))
        elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
          cp = (1000.d0 + ((Tb - 200.d0)/2.d0))
        elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
          cp = 1100.d0

```



```

endif
endif
fluxvarsr(incid(iel,1)) = fluxvarsr(incid(iel,1)) + (((density * cp * area)/(alpha * dt)) * ((0.16666 * tempsr(node1)) + (0.08333 *
tempsr(node2)) + (0.08333 * tempsr(node3))))
fluxvarsr(incid(iel,2)) = fluxvarsr(incid(iel,2)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) + (0.16666 *
tempsr(node2)) + (0.08333 * tempsr(node3))))
fluxvarsr(incidfluxvar(iel,3)) = fluxvarsr(incidfluxvar(iel,3)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) +
(0.08333 * tempsr(node2)) + (0.16666 * tempsr(node3))))
elseif((incidfluxvar(iel,1) .eq. 0) .and. (incidfluxvar(iel,2) .eq. 0) .and. (incidfluxvar(iel,3) .eq. 0)) then
if(imat .eq. 1) then
density = 7850.0
if((Tb .ge. 20.0) .and. (Tb .lt. 600.0)) then
cp = 425.0 + (7.73e-1 * Tb) - (1.69e-3 * (Tb**2)) + (2.22e-6 * (Tb**3))
elseif((Tb .ge. 600.0) .and. (Tb .lt. 735.0)) then
cp = 666.0 + (13002.0/(738.0 - Tb))
elseif((Tb .ge. 735.0) .and. (Tb .lt. 900.0)) then
cp = 545.0 + (17820.0/(Tb - 731.0))
elseif((Tb .ge. 900.0) .and. (Tb .le. 1200.0)) then
cp = 650.0
endif
elseif(imat .eq. 2) then
if((Tb .ge. 20.0) .and. (Tb .le. 115.0)) then
density = 2500.d0 !240.d0
elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then

density = (2500.d0 * (1.d0 - (0.02d0 * ((Tb - 115.d0)/85.d0))))
elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
density = (2500.d0 * (0.98d0 - (0.03d0 * ((Tb - 200.d0)/200.d0))))
elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
density = (2500.d0 * (0.95d0 - (0.07d0 * ((Tb - 400.d0)/800.d0))))
endif
if((Tb .ge. 20.0) .and. (Tb .le. 100.0)) then
cp = 900.d0 !2093.d0
elseif((Tb .gt. 100.0) .and. (Tb .le. 115.0)) then
cp = 2020.d0
elseif((Tb .gt. 115.0) .and. (Tb .le. 200.0)) then
cp = (2020.d0 - (12.d0 * (Tb - 115.d0)))
elseif((Tb .gt. 200.0) .and. (Tb .le. 400.0)) then
cp = (1000.d0 + ((Tb - 200.d0)/2.d0))
elseif((Tb .gt. 400.0) .and. (Tb .le. 1200.0)) then
cp = 1100.d0
endif
endif
fluxvarsr(incid(iel,1)) = fluxvarsr(incid(iel,1)) + (((density * cp * area)/(alpha * dt)) * ((0.16666 * tempsr(node1)) + (0.08333 *
tempsr(node2)) + (0.08333 * tempsr(node3))))
fluxvarsr(incid(iel,2)) = fluxvarsr(incid(iel,2)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) + (0.16666 *
tempsr(node2)) + (0.08333 * tempsr(node3))))
fluxvarsr(incid(iel,3)) = fluxvarsr(incid(iel,3)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) + (0.08333 *
tempsr(node2)) + (0.16666 * tempsr(node3))))
endif
enddo

```

```

endif
if(ume_partc .ne. 0) then
do iel = 1, ume
node1 = incid(iel,1)
node2 = incid(iel,2)
node3 = incid(iel,3)
x1 = x ( node1 )
x2 = x ( node2 )
x3 = x ( node3 )
y1 = y ( node1 )
y2 = y ( node2 )
y3 = y ( node3 )
x21 = x2 - x1
y21 = y2 - y1
y31 = y3 - y1
x31 = x3 - x1
area = 0.5d0 * ( x21*y31 - x31*y21 )
imat = mtype(iel)
Tb = (tempsr(node1) + tempsr(node2) + tempsr(node3))/3.d0
if(incidfluxpresc(iel,1) .ne. 0) then
xfp1 = x(incidfluxpresc(iel,1))
yfp1 = y(incidfluxpresc(iel,1))
endif
if(incidfluxpresc(iel,2) .ne. 0) then
xfp2 = x(incidfluxpresc(iel,2))
yfp2 = y(incidfluxpresc(iel,2))
endif
if(incidfluxpresc(iel,3) .ne. 0) then
xfp3 = x(incidfluxpresc(iel,3))
yfp3 = y(incidfluxpresc(iel,3))
endif
if(((incidfluxpresc(iel,1) .ne. 0) .and. (incidfluxpresc(iel,2) .ne. 0) .and. (incidfluxpresc(iel,3) .eq. 0)) then
L12 = (((x21 - xfp1)**2) + ((y21 - yfp1)**2))*0.5)
if(imat .eq. 1) then
density = 7850.0
if((Tb .ge. 20.0) .and. (Tb .lt. 600.0)) then
cp = 425.0 + (7.73e-1 * Tb) - (1.69e-3 * (Tb**2)) + (2.22e-6 * (Tb**3))
elseif((Tb .ge. 600.0) .and. (Tb .lt. 735.0)) then
cp = 666.0 + (13002.0/(738.0 - Tb))
elseif((Tb .ge. 735.0) .and. (Tb .lt. 900.0)) then
cp = 545.0 + (17820.0/(Tb - 731.0))
elseif((Tb .ge. 900.0) .and. (Tb .le. 1200.0)) then
cp = 650.0
endif
elseif(imat .eq. 2) then
density = 240.0
cp = 2093.0
endif
if(id(incidfluxpresc(iel,1),1) .gt. 0) then
fluxprescsr(incidfluxpresc(iel,1)) = fluxprescsr(incidfluxpresc(iel,1)) + (qp * (L12/2.d0)) + (((density * cp * area)/(alpha * dt)) *
((0.16666 * tempsr(node1)) + (0.08333 * tempsr(node2)) + (0.08333 * tempsr(node3))))

```

```

elseif(id(incidfluxpresc(iel,1),1) .lt. 0) then
fluxprescsr(incidfluxpresc(iel,1)) = fluxprescsr(incidfluxpresc(iel,1)) + (((density * cp * area)/(alpha * dt)) * ((0.16666 *
tempsr(node1)) + (0.08333 * tempsr(node2)) + (0.08333 * tempsr(node3))))
endif
if(id(incidfluxpresc(iel,2),1) .gt. 0) then
fluxprescsr(incidfluxpresc(iel,2)) = fluxprescsr(incidfluxpresc(iel,2)) + (qp * (L12/2.d0)) + (((density * cp * area)/(alpha * dt)) *
((0.08333 * tempsr(node1)) + (0.16666 * tempsr(node2)) + (0.08333 * tempsr(node3))))
elseif(id(incidfluxpresc(iel,2),1) .lt. 0) then
fluxprescsr(incidfluxpresc(iel,2)) = fluxprescsr(incidfluxpresc(iel,2)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 *
tempsr(node1)) + (0.16666 * tempsr(node2)) + (0.08333 * tempsr(node3))))
endif
fluxprescsr(incid(iel,3)) = fluxprescsr(incid(iel,3)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) + (0.08333 *
tempsr(node2)) + (0.16666 * tempsr(node3))))
elseif((incidfluxpresc(iel,1) .eq. 0) .and. (incidfluxpresc(iel,2) .ne. 0) .and. (incidfluxpresc(iel,3) .ne. 0)) then
L23 = (((xfp3 - xfp2)**2) + ((yfp3 - yfp2)**2))*0.5
if(imat .eq. 1) then
density = 7850.0
if((Tb .ge. 20.0) .and. (Tb .lt. 600.0)) then
cp = 425.0 + (7.73e-1 * Tb) - (1.69e-3 * (Tb**2)) + (2.22e-6 * (Tb**3))
elseif((Tb .ge. 600.0) .and. (Tb .lt. 735.0)) then
cp = 666.0 + (13002.0/(738.0 - Tb))
elseif((Tb .ge. 735.0) .and. (Tb .lt. 900.0)) then
cp = 545.0 + (17820.0/(Tb - 731.0))
elseif((Tb .ge. 900.0) .and. (Tb .le. 1200.0)) then
cp = 650.0
endif
elseif(imat .eq. 2) then
density = 240.0
cp = 2093.0
endif
fluxprescsr(incid(iel,1)) = fluxprescsr(incid(iel,1)) + (((density * cp * area)/(alpha * dt)) * ((0.16666 * tempsr(node1)) + (0.08333 *
tempsr(node2)) + (0.08333 * tempsr(node3))))
if(id(incidfluxpresc(iel,2),1) .gt. 0) then
fluxprescsr(incidfluxpresc(iel,2)) = fluxprescsr(incidfluxpresc(iel,2)) + (qp * (L23/2.d0)) + (((density * cp * area)/(alpha * dt)) *
((0.08333 * tempsr(node1)) + (0.16666 * tempsr(node2)) + (0.08333 * tempsr(node3))))
elseif(id(incidfluxpresc(iel,2),1) .lt. 0) then
fluxprescsr(incidfluxpresc(iel,2)) = fluxprescsr(incidfluxpresc(iel,2)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 *
tempsr(node1)) + (0.16666 * tempsr(node2)) + (0.08333 * tempsr(node3))))
endif
if(id(incidfluxpresc(iel,3),1) .gt. 0) then
fluxprescsr(incidfluxpresc(iel,3)) = fluxprescsr(incidfluxpresc(iel,3)) + (qp * (L23/2.d0)) + (((density * cp * area)/(alpha * dt)) *
((0.08333 * tempsr(node1)) + (0.08333 * tempsr(node2)) + (0.1666 * tempsr(node3))))
elseif(id(incidfluxpresc(iel,3),1) .lt. 0) then
fluxprescsr(incidfluxpresc(iel,3)) = fluxprescsr(incidfluxpresc(iel,3)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 *
tempsr(node1)) + (0.08333 * tempsr(node2)) + (0.1666 * tempsr(node3))))
endif
elseif((incidfluxpresc(iel,1) .ne. 0) .and. (incidfluxpresc(iel,2) .eq. 0) .and. (incidfluxpresc(iel,3) .ne. 0)) then
L13 = (((xfp3 - xfp1)**2) + ((yfp3 - yfp1)**2))*0.5
if(imat .eq. 1) then
density = 7850.0
if((Tb .ge. 20.0) .and. (Tb .lt. 600.0)) then

```

```

cp = 425.0 + (7.73e-1 * Tb) - (1.69e-3 * (Tb**2)) + (2.22e-6 * (Tb**3))
elseif((Tb .ge. 600.0) .and. (Tb .lt. 735.0)) then
cp = 666.0 + (13002.0/(738.0 - Tb))
elseif((Tb .ge. 735.0) .and. (Tb .lt. 900.0)) then
cp = 545.0 + (17820.0/(Tb - 731.0))
elseif((Tb .ge. 900.0) .and. (Tb .le. 1200.0)) then
cp = 650.0
endif
elseif(imat .eq. 2) then
density = 240.0
cp = 2093.0
endif
if(id(incidfluxpresc(iel,1),1) .gt. 0) then
fluxprescsr(incidfluxpresc(iel,1)) = fluxprescsr(incidfluxpresc(iel,1)) + (qp * (L13/2.d0)) + (((density * cp * area)/(alpha * dt)) *
((0.16666 * tempsr(node1)) + (0.08333 * tempsr(node2)) + (0.08333 * tempsr(node3))))
elseif(id(incidfluxpresc(iel,1),1) .lt. 0) then
fluxprescsr(incidfluxpresc(iel,1)) = fluxprescsr(incidfluxpresc(iel,1)) + (((density * cp * area)/(alpha * dt)) * ((0.16666 *
tempsr(node1)) + (0.08333 * tempsr(node2)) + (0.08333 * tempsr(node3))))
endif
fluxprescsr(incid(iel,2)) = fluxprescsr(incid(iel,2)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) + (0.16666 *
tempsr(node2)) + (0.08333 * tempsr(node3))))
if(id(incidfluxpresc(iel,3),1) .gt. 0) then
fluxprescsr(incidfluxpresc(iel,3)) = fluxprescsr(incidfluxpresc(iel,3)) + (qp * (L13/2.d0)) + (((density * cp * area)/(alpha * dt)) *
((0.08333 * tempsr(node1)) + (0.08333 * tempsr(node2)) + (0.16666 * tempsr(node3))))
elseif(id(incidfluxpresc(iel,3),1) .lt. 0) then
fluxprescsr(incidfluxpresc(iel,3)) = fluxprescsr(incidfluxpresc(iel,3)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 *
tempsr(node1)) + (0.08333 * tempsr(node2)) + (0.16666 * tempsr(node3))))
endif
elseif((incidfluxpresc(iel,1) .ne. 0) .and. (incidfluxpresc(iel,2) .eq. 0) .and. (incidfluxpresc(iel,3) .eq. 0)) then
if(imat .eq. 1) then
density = 7850.0
if((Tb .ge. 20.0) .and. (Tb .lt. 600.0)) then
cp = 425.0 + (7.73e-1 * Tb) - (1.69e-3 * (Tb**2)) + (2.22e-6 * (Tb**3))
elseif((Tb .ge. 600.0) .and. (Tb .lt. 735.0)) then
cp = 666.0 + (13002.0/(738.0 - Tb))
elseif((Tb .ge. 735.0) .and. (Tb .lt. 900.0)) then
cp = 545.0 + (17820.0/(Tb - 731.0))
elseif((Tb .ge. 900.0) .and. (Tb .le. 1200.0)) then
cp = 650.0
endif
elseif(imat .eq. 2) then
density = 240.0
cp = 2093.0
endif
fluxprescsr(incidfluxpresc(iel,1)) = fluxprescsr(incidfluxpresc(iel,1)) + (((density * cp * area)/(alpha * dt)) * ((0.16666 *
tempsr(node1)) + (0.08333 * tempsr(node2)) + (0.08333 * tempsr(node3))))
fluxprescsr(incid(iel,2)) = fluxprescsr(incid(iel,2)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) + (0.16666 *
tempsr(node2)) + (0.08333 * tempsr(node3))))
fluxprescsr(incid(iel,3)) = fluxprescsr(incid(iel,3)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) + (0.08333 *
tempsr(node2)) + (0.16666 * tempsr(node3))))
elseif((incidfluxpresc(iel,1) .eq. 0) .and. (incidfluxpresc(iel,2) .ne. 0) .and. (incidfluxpresc(iel,3) .eq. 0)) then

```

```

if(imat .eq. 1) then
  density = 7850.0
  if((Tb .ge. 20.0) .and. (Tb .lt. 600.0)) then
    cp = 425.0 + (7.73e-1 * Tb) - (1.69e-3 * (Tb**2)) + (2.22e-6 * (Tb**3))
  elseif((Tb .ge. 600.0) .and. (Tb .lt. 735.0)) then
    cp = 666.0 + (13002.0/(738.0 - Tb))
  elseif((Tb .ge. 735.0) .and. (Tb .lt. 900.0)) then
    cp = 545.0 + (17820.0/(Tb - 731.0))
  elseif((Tb .ge. 900.0) .and. (Tb .le. 1200.0)) then
    cp = 650.0
  endif
elseif(imat .eq. 2) then
  density = 240.0
  cp = 2093.0
endif
fluxprescsr(incid(iel,1)) = fluxprescsr(incid(iel,1)) + (((density * cp * area)/(alpha * dt)) * ((0.16666 * tempsr(node1)) + (0.08333 *
tempshr(node2)) + (0.08333 * tempsr(node3))))
fluxprescsr(incidfluxpresc(iel,2)) = fluxprescsr(incidfluxpresc(iel,2)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 *
tempshr(node1)) + (0.16666 * tempsr(node2)) + (0.08333 * tempsr(node3))))
fluxprescsr(incid(iel,3)) = fluxprescsr(incid(iel,3)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) + (0.08333 *
tempshr(node2)) + (0.16666 * tempsr(node3))))
elseif((incidfluxpresc(iel,1) .eq. 0) .and. (incidfluxpresc(iel,2) .eq. 0) .and. (incidfluxpresc(iel,3) .ne. 0)) then
if(imat .eq. 1) then
  density = 7850.0
  if((Tb .ge. 20.0) .and. (Tb .lt. 600.0)) then
    cp = 425.0 + (7.73e-1 * Tb) - (1.69e-3 * (Tb**2)) + (2.22e-6 * (Tb**3))
  elseif((Tb .ge. 600.0) .and. (Tb .lt. 735.0)) then
    cp = 666.0 + (13002.0/(738.0 - Tb))
  elseif((Tb .ge. 735.0) .and. (Tb .lt. 900.0)) then
    cp = 545.0 + (17820.0/(Tb - 731.0))
  elseif((Tb .ge. 900.0) .and. (Tb .le. 1200.0)) then
    cp = 650.0
  endif
elseif(imat .eq. 2) then
  density = 240.0
  cp = 2093.0
endif
fluxprescsr(incid(iel,1)) = fluxprescsr(incid(iel,1)) + (((density * cp * area)/(alpha * dt)) * ((0.16666 * tempsr(node1)) + (0.08333 *
tempshr(node2)) + (0.08333 * tempsr(node3))))
fluxprescsr(incid(iel,2)) = fluxprescsr(incid(iel,2)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) + (0.16666 *
tempshr(node2)) + (0.08333 * tempsr(node3))))
fluxprescsr(incidfluxpresc(iel,3)) = fluxprescsr(incidfluxpresc(iel,3)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 *
tempshr(node1)) + (0.08333 * tempsr(node2)) + (0.16666 * tempsr(node3))))
elseif((incidfluxpresc(iel,1) .eq. 0) .and. (incidfluxpresc(iel,2) .eq. 0) .and. (incidfluxpresc(iel,3) .eq. 0)) then
if(imat .eq. 1) then
  density = 7850.0
  if((Tb .ge. 20.0) .and. (Tb .lt. 600.0)) then
    cp = 425.0 + (7.73e-1 * Tb) - (1.69e-3 * (Tb**2)) + (2.22e-6 * (Tb**3))
  elseif((Tb .ge. 600.0) .and. (Tb .lt. 735.0)) then
    cp = 666.0 + (13002.0/(738.0 - Tb))
  elseif((Tb .ge. 735.0) .and. (Tb .lt. 900.0)) then

```

```

cp = 545.0 + (17820.0/(Tb - 731.0))
elseif((Tb .ge. 900.0) .and. (Tb .le. 1200.0)) then
cp = 650.0
endif
elseif(imat .eq. 2) then
density = 240.0
cp = 2093.0
endif
fluxprescsr(incid(iel,1)) = fluxprescsr(incid(iel,1)) + (((density * cp * area)/(alpha * dt)) * ((0.16666 * tempsr(node1)) + (0.08333 *
tempsr(node2)) + (0.08333 * tempsr(node3))))
fluxprescsr(incid(iel,2)) = fluxprescsr(incid(iel,2)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) + (0.16666 *
tempsr(node2)) + (0.08333 * tempsr(node3))))
fluxprescsr(incid(iel,3)) = fluxprescsr(incid(iel,3)) + (((density * cp * area)/(alpha * dt)) * ((0.08333 * tempsr(node1)) + (0.08333 *
tempsr(node2)) + (0.16666 * tempsr(node3))))
endif
enddo
endif
elseif(timetot .eq. 0.d0) then
if(ume_partv .ne. 0) then
do iel = 1, nume
node1 = incid(iel,1)
node2 = incid(iel,2)
node3 = incid(iel,3)
x1 = x ( node1 )
x2 = x ( node2 )
x3 = x ( node3 )
y1 = y ( node1 )
y2 = y ( node2 )
y3 = y ( node3 )
x21 = x2 - x1
y21 = y2 - y1
y31 = y3 - y1
x31 = x3 - x1
area = 0.5d0 * ( x21*y31 - x31*y21 )
imat = mtype(iel)
Tb = (tempsr(node1) + tempsr(node2) + tempsr(node3))/3.d0
Tg = Tamb
if(incidfluxvar(iel,1) .ne. 0) then
xfv1 = x(incidfluxvar(iel,1))
yfv1 = y(incidfluxvar(iel,1))
endif
if(incidfluxvar(iel,2) .ne. 0) then
xfv2 = x(incidfluxvar(iel,2))
yfv2 = y(incidfluxvar(iel,2))
endif
if(incidfluxvar(iel,3) .ne. 0) then
xfv3 = x(incidfluxvar(iel,3))
yfv3 = y(incidfluxvar(iel,3))
endif
if(((incidfluxvar(iel,1) .ne. 0) .and. (incidfluxvar(iel,2) .ne. 0) .and. (incidfluxvar(iel,3) .eq. 0)) then
Ta = (tempsr(node1) + tempsr(node2))/2.d0

```

```

L12 = (((xfv2 - xfv1)**2) + ((yfv2 - yfv1)**2))**0.5)
if(imat .eq. 1) then
  heq = (hc + (sigma * eres1 * ((Tg + 273.0) + (Ta + 273.0)) * (((Tg + 273.0)**2) + ((Ta + 273.0)**2))))
elseif(imat .eq. 2) then
  heq = (hc + (sigma * eres2p * ((Tg + 273.0) + (Ta + 273.0)) * (((Tg + 273.0)**2) + ((Ta + 273.0)**2))))
endif
fluxvarsr(incidfluxvar(iel,1)) = fluxvarsr(incidfluxvar(iel,1)) + (heq * Tg * (L12/2.d0))
fluxvarsr(incidfluxvar(iel,2)) = fluxvarsr(incidfluxvar(iel,2)) + (heq * Tg * (L12/2.d0))
elseif((incidfluxvar(iel,1) .eq. 0) .and. (incidfluxvar(iel,2) .ne. 0) .and. (incidfluxvar(iel,3) .ne. 0)) then
  Ta = (tempsr(node2) + tempsr(node3))/2.d0
  L23 = (((xfv3 - xfv2)**2) + ((yfv3 - yfv2)**2))**0.5)
  if(imat .eq. 1) then
    heq = (hc + (sigma * eres1 * ((Tg + 273.0) + (Ta + 273.0)) * (((Tg + 273.0)**2) + ((Ta + 273.0)**2))))
  elseif(imat .eq. 2) then
    heq = (hc + (sigma * eres2p * ((Tg + 273.0) + (Ta + 273.0)) * (((Tg + 273.0)**2) + ((Ta + 273.0)**2))))
  endif
  fluxvarsr(incidfluxvar(iel,2)) = fluxvarsr(incidfluxvar(iel,2)) + (heq * Tg * (L23/2.d0))
  fluxvarsr(incidfluxvar(iel,3)) = fluxvarsr(incidfluxvar(iel,3)) + (heq * Tg * (L23/2.d0))
  elseif((incidfluxvar(iel,1) .ne. 0) .and. (incidfluxvar(iel,2) .eq. 0) .and. (incidfluxvar(iel,3) .ne. 0)) then
    Ta = (tempsr(node1) + tempsr(node3))/2.d0
    L13 = (((xfv3 - xfv1)**2) + ((yfv3 - yfv1)**2))**0.5)
    if(imat .eq. 1) then
      heq = (hc + (sigma * eres1 * ((Tg + 273.0) + (Ta + 273.0)) * (((Tg + 273.0)**2) + ((Ta + 273.0)**2))))
    elseif(imat .eq. 2) then
      heq = (hc + (sigma * eres2p * ((Tg + 273.0) + (Ta + 273.0)) * (((Tg + 273.0)**2) + ((Ta + 273.0)**2))))
    endif
    fluxvarsr(incidfluxvar(iel,1)) = fluxvarsr(incidfluxvar(iel,1)) + (heq * Tg * (L13/2.d0))
    fluxvarsr(incidfluxvar(iel,3)) = fluxvarsr(incidfluxvar(iel,3)) + (heq * Tg * (L13/2.d0))
  endif
endif
enddo
endif
if(ume_partc .ne. 0) then
  do iel = 1, ume
    node1 = incid(iel,1)
    node2 = incid(iel,2)
    node3 = incid(iel,3)
    x1 = x ( node1 )
    x2 = x ( node2 )
    x3 = x ( node3 )
    y1 = y ( node1 )
    y2 = y ( node2 )
    y3 = y ( node3 )
    x21 = x2 - x1
    y21 = y2 - y1
    y31 = y3 - y1
    x31 = x3 - x1
    area = 0.5d0 * ( x21*y31 - x31*y21 )
    imat = mtype(iel)
    Tb = (tempsr(node1) + tempsr(node2) + tempsr(node3))/3.d0
    if(incidfluxpresc(iel,1) .ne. 0) then
      xfp1 = x(incidfluxpresc(iel,1))

```

```

yfp1 = y(incidfluxpresc(iel,1))
endif
if(incidfluxpresc(iel,2) .ne. 0) then
xfp2 = x(incidfluxpresc(iel,2))
yfp2 = y(incidfluxpresc(iel,2))
endif
if(incidfluxpresc(iel,3) .ne. 0) then
xfp3 = x(incidfluxpresc(iel,3))
yfp3 = y(incidfluxpresc(iel,3))
endif
if((incidfluxpresc(iel,1) .ne. 0) .and. (incidfluxpresc(iel,2) .ne. 0) .and. (incidfluxpresc(iel,3) .eq. 0)) then
L12 = (((xfp2 - xfp1)**2) + ((yfp2 - yfp1)**2))*0.5)
fluxprescsr(incidfluxpresc(iel,1)) = fluxprescsr(incidfluxpresc(iel,1)) + (qp * (L12/2.d0))
fluxprescsr(incidfluxpresc(iel,2)) = fluxprescsr(incidfluxpresc(iel,2)) + (qp * (L12/2.d0))
elseif((incidfluxpresc(iel,1) .eq. 0) .and. (incidfluxpresc(iel,2) .ne. 0) .and. (incidfluxpresc(iel,3) .ne. 0)) then
L23 = (((xfp3 - xfp2)**2) + ((yfp3 - yfp2)**2))*0.5)
fluxprescsr(incidfluxpresc(iel,2)) = fluxprescsr(incidfluxpresc(iel,2)) + (qp * (L23/2.d0))
fluxprescsr(incidfluxpresc(iel,3)) = fluxprescsr(incidfluxpresc(iel,3)) + (qp * (L23/2.d0))
elseif((incidfluxpresc(iel,1) .ne. 0) .and. (incidfluxpresc(iel,2) .eq. 0) .and. (incidfluxpresc(iel,3) .ne. 0)) then
L13 = (((xfp3 - xfp1)**2) + ((yfp3 - yfp1)**2))*0.5)
fluxprescsr(incidfluxpresc(iel,1)) = fluxprescsr(incidfluxpresc(iel,1)) + (qp * (L13/2.d0))
fluxprescsr(incidfluxpresc(iel,3)) = fluxprescsr(incidfluxpresc(iel,3)) + (qp * (L13/2.d0))
endif
enddo
endif
endif
end subroutine fluxprescvar

```

## APÊNDICE H - SUBROTINA ADDBAN2

```

subroutine addban2 (a,maxa,s,lm,nd,neq1,nwk)
implicit real*8 (a-h,o-z)
integer (KIND=8) :: kk, nw
integer (KIND=8) :: maxa(neq1)
real*8 :: a(nwk),s(6)
integer :: lm(nd)
do j = 1, nd
jeq = lm(j)
if (jeq.gt.0) then
do i = 1, j
ieq = lm(i)
if (ieq.gt.0) then
is = j*(j-1)/2 + i
if (jeq.ge.ieq) then
kk = maxa(jeq) + jeq-ieq
else
kk = maxa(ieq) + ieq-jeq
endif
a(kk) = a(kk) + s(is)

```



```

endif
enddo
endif
enddo
return
end subroutine addban2

```

## APÊNDICE I - SUBROTINA RHS

```

subroutine rhs(fluxprescsr,fluxvarsr,sgpsr,floadsr)
use modinput , only: numnp, temp_presc
use modprep , only: id, neq, neqp
use modloads , only: upresc
implicit none
integer :: ino,i,j,ieq
real*8 :: faux
real*8 :: fluxvarsr(numnp)
real*8 :: fluxprescsr(numnp)
real*8 :: sgpsr(neq,neqp)
real*8 :: floadsr(neq)
floadsr(:) = 0.d0
if (neqp .gt. 0) then
allocate ( upresc (neqp) ) ; upresc(:) = 0.d0
do ino = 1, numnp
if ( id(ino,1) .lt. 0 ) then
ieq = - id(ino,1)
upresc(ieq) = temp_presc(ino)
endif
enddo
do i = 1, neq
faux = 0.d0
do j = 1, neqp
faux = faux + sgpsr(i,j)*upresc(j)
enddo
floadsr (i) = -faux
enddo
endif
do ino = 1, numnp
if ( id(ino,1) .gt. 0 ) then
ieq = id(ino,1)
floadsr (ieq) = floadsr (ieq) - fluxprescsr(ino) + fluxvarsr(ino)
endif
enddo
end subroutine rhs

```

## APÊNDICE J - SUBROTINA COLSOL

```

subroutine COLSOL (A,V,MAXA,NN,NWK,NNM,KKK,IOUT)
IMPLICIT real*8 (A-H,O-Z)
integer (KIND=8) KK, KI, MND , L, KLT, IC, KN, KL, KU, K
integer (KIND=8) NWK
integer (KIND=8) MAXA(NNM)
DIMENSION A(NWK),V(1:NN)
icont = 0
iflag = 0
IF (KKK-2) 40,150,150
40 DO 140 N=1,NN
icont = icont + 1
KN=MAXA(N)
KL=KN + 1
KU=MAXA(N+1) - 1
KH=KU - KL
IF (KH) 110,90,50
50 K=N - KH
IC=0
KLT=KU
DO 80 J=1,KH
IC=IC + 1
KLT=KLT - 1
KI=MAXA(K)
MND=MAXA(K+1) - KI - 1
IF (MND) 80,80,60
60 KK=MIN0(IC,MND)
C=0.
DO 70 L=1,KK
70 C=C + A(KI+L)*A(KLT+L)
A(KLT)=A(KLT) - C
80 K=K + 1
90 K=N
B=0.
DO 100 KK=KL,KU
K=K - 1
KI=MAXA(K)
C=A(KK)/A(KI)
B=B + C*A(KK)
100 A(KK)=C
A(KN)=A(KN) - B
110 IF (A(KN)) 120,120,140
120 WRITE (IOUT,2000) N,A(KN)
GO TO 800
140 CONTINUE
GO TO 900
150 DO 180 N=1,NN
KL=MAXA(N) + 1
KU=MAXA(N+1) - 1

```

```

IF (KU-KL) 180,160,160
160 K=N
C=0.
DO 170 KK=KL,KU
K=K - 1
170 C=C + A(KK)*V(K)
V(N)=V(N) - C
180 CONTINUE
DO 200 N=1,NN
K=MAXA(N)
200 V(N)=V(N)/A(K)
IF (NN.EQ.1) GO TO 900
N=NN
DO 230 L=2,NN
KL=MAXA(N) + 1
KU=MAXA(N+1) - 1
IF (KU-KL) 230,210,210
210 K=N
DO 220 KK=KL,KU
K=K - 1
220 V(K)=V(K) - A(KK)*V(N)
230 N=N - 1
GO TO 900
800 STOP
900 RETURN
2000 FORMAT ('// STOP - STIFFNESS MATRIX NOT POSITIVE DEFINITE',//, &
' NONPOSITIVE PIVOT FOR EQUATION ',I8,//, & ' PIVOT = ',E20.12 )
end subroutine colsol

```

## APÊNDICE K - SUBROTINA FLUX\_VECTOR

```

subroutine flux_vector(tempnodalsr,fluxnodalsr)
use modinput , only: x,y,numnp,nume,incid,mtype,timetot
use modprep
use modaux , only: flux_x,flux_y,area,areanodal,xcount
implicit none
real*8 :: tempnodalsr(numnp), fluxnodalsr(numnp,2)
integer :: i,j,iel,inoel,ino,node1,node2,node3,imat
real*8 :: x1,x2,x3,y1,y2,y3,x21,y21,x31,y31,x32,x13,y23,y12
real*8 :: factor, xkx, xky
real*8 :: dn1x, dn1y, dn2x, dn2y, dn3x, dn3y, ti, tj, tk, tavg
xcount(:) = 0
do iel = 1, nume
node1 = incid(iel,1)
node2 = incid(iel,2)
node3 = incid(iel,3)
x1 = x ( node1 )
x2 = x ( node2 )
x3 = x ( node3 )

```

```

y1 = y ( node1 )
y2 = y ( node2 )
y3 = y ( node3 )
x21 = x2 - x1
y21 = y2 - y1
y31 = y3 - y1
x31 = x3 - x1
area(iel) = 0.5d0 * ( x21*y31 - x31*y21 )
x32 = x3 - x2
x13 = x1 - x3
y23 = y2 - y3
y12 = y1 - y2
factor = .5d0 / area(iel)
ti = tempnodalsr (node1)
tj = tempnodalsr (node2)
tk = tempnodalsr (node3)
tavg = ( ti + tj + tk )/3.0d0
imat = mtype(iel)
if(imat .eq. 1) then
  if((tavg .ge. 20) .and. (tavg .lt. 800)) then
    xkx = 54 - (3.33e-2 * tavg)
    xky = 54 - (3.33e-2 * tavg)
  elseif((tavg .ge. 800) .and. (tavg .le. 1200)) then
    xkx = 27.3
    xky = 27.3
  endif
elseif(imat .eq. 2) then
  if(timetot .gt. 0.d0) then
    xkx = 0.043
    xky = 0.043
  elseif(timetot .eq. 0.d0) then
    xkx = 1.4
    xky = 1.4
  endif
endif
dn1x = y23
dn1y = x32
dn2x = y31
dn2y = x13
dn3x = y12
dn3y = x21
flux_x(iel) = - factor*XKX*(dn1x*ti + dn2x*tj + dn3x*tk)
flux_y(iel) = - factor*XKY*(dn1y*ti + dn2y*tj + dn3y*tk)
enddo
areanodal(:) = 0.d0
do ino = 1, numnp
  do iel = 1, nume
    do inoel = 1, 3
      if (incid (iel, inoel) .eq. ino ) then
        areanodal(ino) = areanodal(ino) + area(iel)
        xcount(ino) = xcount(ino) + 1.d0
      endif
    enddo
  enddo
enddo

```

```

endif
enddo
enddo
enddo
fluxnodalsr(:, :) = 0.d0
do ino = 1, numnp
do iel = 1, nume
do inoel = 1, 3
if (incid(iel, inoel) .eq. ino) then
fluxnodalsr(ino,1) = fluxnodalsr(ino,1) + flux_x(iel)/xcount(ino)
fluxnodalsr(ino,2) = fluxnodalsr(ino,2) + flux_y(iel)/xcount(ino)
endif
enddo
enddo
enddo
end subroutine flux_vector

```

## APÊNDICE L - SUBROTINA ENERGY

```

subroutine energy (floads,tempsr,ske_ebesr,energy_outsr)
use modinput , only : nume,numnp,alpha,incid
use modprep , only : ngl, nnoel, lm, neq, neqp
implicit none
real*8 :: floads(neq),tempsr(numnp),ske_ebesr(6,nume)
real*8 :: energy_outsr
integer :: iel, ind, ieq, nd
real*8 :: temp_iel(3), fint_iel(3)
nd = nnoel * ngl
energy_outsr = 0.d0
do iel = 1, nume
fint_iel(:) = 0.d0
temp_iel(:) = 0.d0
do ind = 1, nd
ieq = lm(iel,ind)
if (ieq.gt.0) then
temp_iel(ind) = (((1/alpha)*floads(ieq))+(1-(1/alpha))*(tempsr(incid(iel,ind))))
elseif (ieq.le.0) then
temp_iel(ind) = 0.d0
endif
fint_iel(1) = ske_ebesr(1,iel)*temp_iel(1) + ske_ebesr(2,iel)*temp_iel(2) + ske_ebesr(4,iel)*temp_iel(3)
fint_iel(2) = ske_ebesr(2,iel)*temp_iel(1) + ske_ebesr(3,iel)*temp_iel(2) + ske_ebesr(5,iel)*temp_iel(3)
fint_iel(3) = ske_ebesr(4,iel)*temp_iel(1) + ske_ebesr(5,iel)*temp_iel(2) + ske_ebesr(6,iel)*temp_iel(3)
energy_outsr = energy_outsr + fint_iel(1)*temp_iel(1) + &
fint_iel(2)*temp_iel(2) + &
fint_iel(3)*temp_iel(3)
enddo
enddo
end subroutine energy

```

## APÊNDICE M - SUBROTINA ENSG\_SCALNODE

```

subroutine ensg_scalnode (scal,numnp,istep)
implicit real*8 (a-h,o-z)
dimension  scal (numnp)
character*70 filename
character*11 filer
character*4  tipo
integer :: iplt
iplt = 81
tipo = 'temp'
filer = 'temperature'
write (filename(1:4),'(i4.4)') istep
write(filename,'(a)') filer// './tipo//filename(1:4)
write(*,'(a)') ' Gravando '//filename(1:50)
open (unit=iplt,file=filename,form='formatted')
write (iplt,'(a,i5,1x,a)') 'Ensignt Scalar step ',istep
write (iplt,'(1p,6e12.4E3)') (scal(i), i=1,numnp)
close (unit=iplt)
return
end subroutine ensg_scalnode

```

## APÊNDICE N - SUBROTINA ENSG\_VECNODE

```

subroutine ensg_vecnode (vec,numnp,istep)
implicit real*8 (a-h,o-z)
dimension  vec (numnp,2)
character*70 filename
character*4  filer
character*3  tipo
integer :: iplt
iplt = 81
tipo = 'fxy'
filer = 'flux'
write (filename(1:4),'(i4.4)') istep
write(filename,'(a)') filer// './tipo//filename(1:4)
write(*,'(a)') ' Gravando '//filename(1:50)
open (unit=iplt,file=filename,form='formatted')
write (iplt,'(a,i5,1x,a)') 'Ensignt vector step ',istep
write (iplt,'(1p,6e12.4E3)') ( vec(i,1),vec(i,2),0.d0, i=1,numnp)
close (unit=iplt)
return
end subroutine ensg_vecnode

```