TRANSIENT NONLINEAR HEAT TRANSFER ANALYSIS USING THE FINITE ELEMENT METHOD IN THE CONTEXT OF THE REQUIREMENTS OF THERMAL ANALYSIS IN A MINE

L de F FRENCH

MSc thesis (1990)

Submitted to the University of Cape Town
in partial fulfillment for the degree of
Master of Science in Engineering

DECLARATION

I, Lance de Freyene French, declare that, apart from supervision and help received from Professor J.B. Martin, Colin Mercer and Nick Marais, this thesis is my own, unaided work.

Lance French

Lance de Freyene French

ABSTRACT

The aim of this thesis is to develop a computer program, together with a users' guide, to analyse two-dimensional, nonlinear, transient heat conduction in non-isotropic solids using the finite element method. This program is an extension of an existing program that analyses thermo-mechanical stress in solids which may have prescribed temperature and flux boundary conditions. The program has been extended using the requirements for modelling heat transfer in mines as a guide. The theory of conduction, thermal radiation, convection and heat transfer due to evaporation and condensation is presented. The transient heat conduction equation is discretised in space using the finite element method. An implicit/explicit difference scheme is used to discretise the time differential term and a substitution method is used to converge to solution. The implementation of the finite element equations in the program and its overall structure is described. Seven examples are analysed to compare program-generated and analytical solutions which test prescribed temperature, prescribed flux, convection, radiation and mass transfer boundary conditions, internal heat generation and infinite elements. It was found that the results for all examples, apart from the mass transfer boundary conditions example, compare well with analytical results. The program does not include the ability to model variable domain geometries. A users' guide for the program has been completed.

ACKNOWLEDGEMENTS

I would like to express my gratitude to Professor J.B. Martin of the Applied Mechanics Research Unit of the University of Cape Town for his supervision and guidance.

I would like to express my appreciation to the University of Cape Town and the Council for Scientific and Industrial Research for their financial support.

Lastly, thanks to Nick Marias and Colin Mercer of the Applied Mechanics Research Unit for all the help given during the course of this work.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Page

CHAPTER 1 : INTRODUCTION

The transfer of heat is a naturally occurring, ubiquitous phenomenon which influences the techniques used in engineering when attempting to control and manipulate the environment. Heat transfer is an influential factor in chemical, mechanical, nuclear and electrical processes. Thus, in order to design efficient and stable processes, it is necessary to understand the effects of, and be able to model, the heat transfer occurring in such processes.

The governing equations of heat transfer are well known and it is possible to derive analytical solutions for a few simple cases. In order to generalise the solution of these equations, it is necessary to use numerical methods together with simplifying assumptions. In the past, the implementation of numerical methods has been hindered by the amount of calculation required to generate solutions. However, as sophisticated computers and programming languages are now available, it is possible to use these methods effectively.

The development of a computer program to solve any heat transfer problem is beyond the scope of an MSc(Eng) thesis since, firstly, heat transfer is carried out by a number of different mechanisms, each sufficiently complex to merit its own detailed study. Secondly, there are so many physical processes in which heat transfer occurs that the possibility of successfully writing such a general program is remote. Therefore, this thesis aims to develop a program which models one mechanism of heat transfer and which is based to some extent on the requirements of a particular application.

These conditions were partly fulfilled by Korvink 1986 who developed a program which uses the finite element method to model transient heat conduction in solids. Korvink's program analyses problems with prescribed temperature and flux boundary conditions and may also be used for thermo-mechanical stress analysis. It was decided to extend the work of Korvink 1986 using heat transfer in mines as a guideline. This selection was made firstly because a program capable of analysing heat transfer in a mine would be useful and, secondly, because the heat transfer occurring in a mine is sufficiently complex to enable the development of a reasonably general heat transfer analysis program.

In this chapter, a description of the heat transfer which occurs in mines is presented. Next, the solution of thermal problems is discussed and, finally, the thesis objectives are presented.

1.1 HEAT TRANSFER IN A MINE.

The rock in an underground mine has a temperature significantly higher than the rock temperature at the surface - for example, ±50°C at a depth of 3000m in a South African mine. Due to the temperature of the rock at the depth at which mining often occurs and the restricted temperature and humidity conditions in which humans can function comfortably, it is necessary to provide some means of conditioning the environment in which men are required to work (Mitchell and Whillier 1971). In order to condition the air in a mine it is necessary to determine the heat flow into the mine and, to do this efficiently and economically, accuracy is required.

A mine is basically comprised of a linked network of shafts, tunnels and stopes. Shafts provide access to tunnels on different levels of the mine and the tunnels, in turn, provide access to stopes. A stope is an area where ore is extracted by blasting the rock face in a direction following the mineral bearing rock vein and then removing the blasted material from the stope area for processing elsewhere.

The heat transfer processes which occur in a mine are:

1) Conduction in the rock
2) Convection in the air in tunnels, shafts and stopes
3) Convection in water which collects on surfaces and which is sprayed on to rock
4) Radiation between rock surfaces and between air and rock
5) Condensation and evaporation of water vapour

Sources of heat within a mine are:

1) Geothermal energy

2) Energy from mechanical devices

3) Energy released by explosions while blasting.

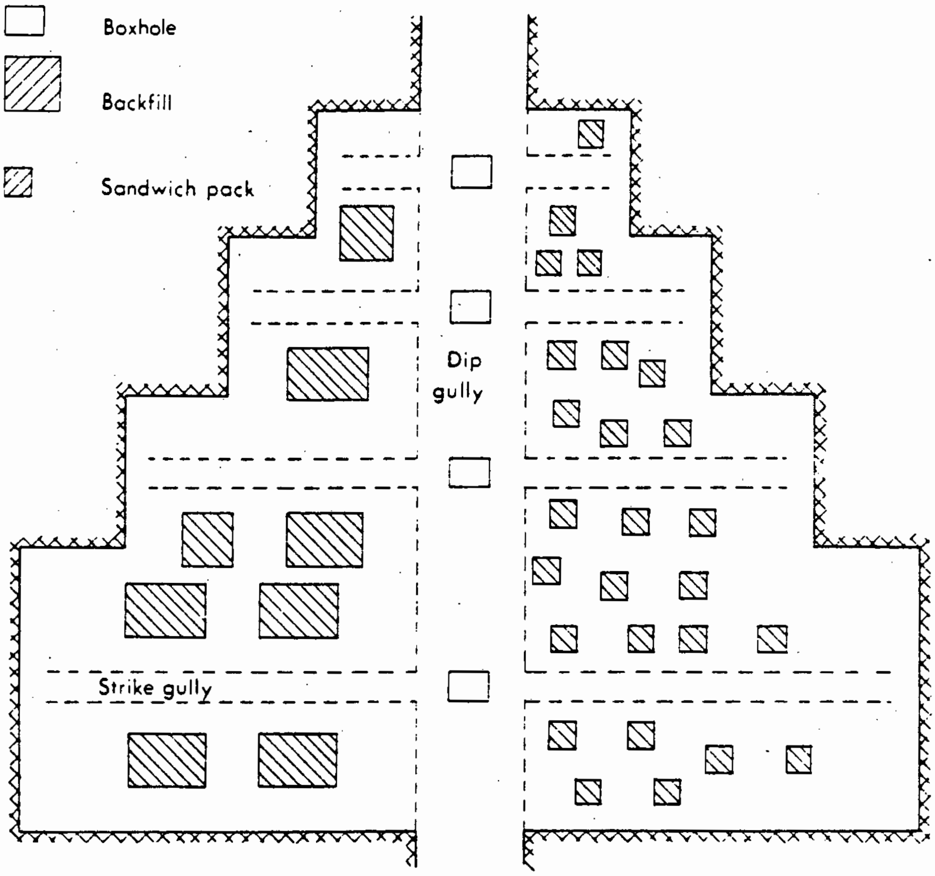


Figure 1.1 *Plan view of a typical mine stope.*

Figure 1.1 represents a plan view of a typical stope area. The stope is generally connected by tunnels at two different levels at either end of the dip gully. Perpendicular to the dip gully - which is the main access-way along the length of the stope - are the strike gullies, which lead to the stope face on either side of the dip gully.

The dimensions of a typical stope are large. The approximate figures below give some indication of the actual sizes involved.

Dip gully	± 200 m
Strike gully	± 50 m
Stope height	± 1.4-1.7 m

Modelling the heat transfer in the region of the stope face is complex for the following reasons:

1) Fourteen distinct operations at the stope face from blast to blast have been identified (von Glehn 1987). These operations are represented in figure 1.2.

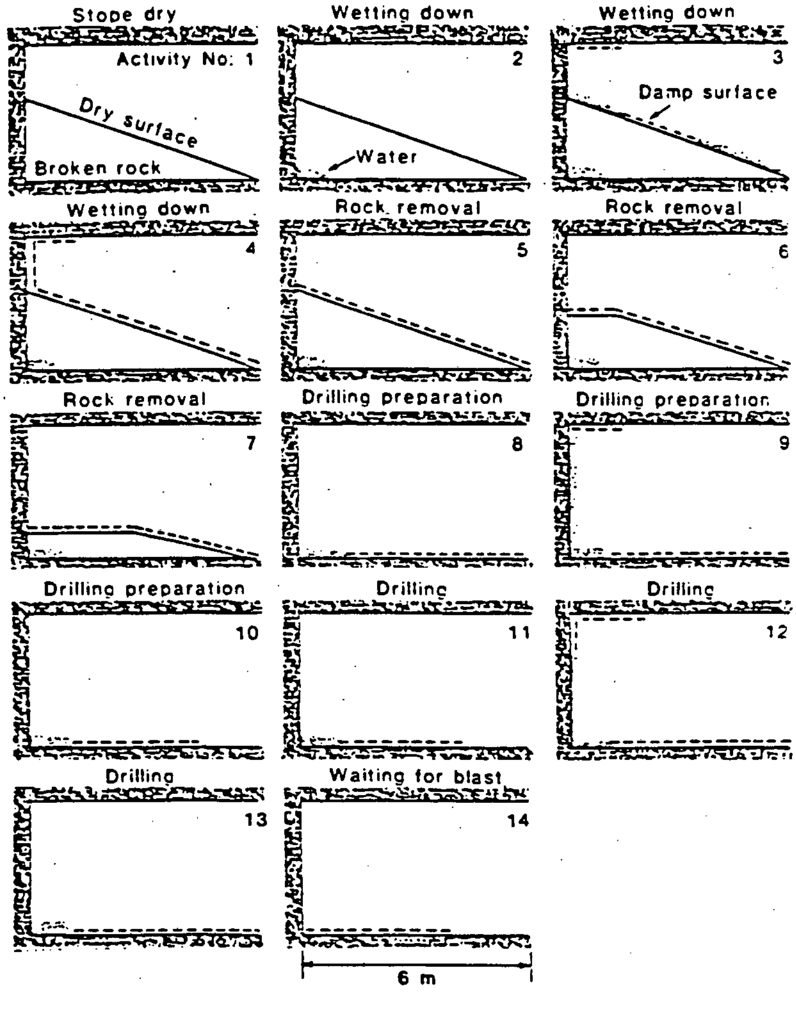


Figure 1.2 *Activities in a stope between blasts.*

2) As the stope face is mined, it moves at a certain rate per day due to blasting. Therefore, the numerical method used must be capable of modelling heat transfer in domains with dynamic geometries.

3) The blasted rock collects at the base of the stope face and is 'wetted down' to facilitate dust control. However, the distribution of water on the rock is not uniform and, in addition, the entire blasted rock mass is not 'wetted down' simultaneously.

4) The blasted rock is mechanically hauled along the strike gully to the dip gully where it is dropped down a boxhole into an appropriately positioned ore truck. Not all the blasted rock is removed at the same time. Consequently, it is necessary to be able to model domains with dynamic geometries.

Existing commercial thermal analysis programs do not allow the user to model heat transfer in a mine without regular re-definition of the problem definition parameters during the analysis. Thus it is desirable to develop a program that addresses this problem.

1.2 The solution of transient thermal problems

Heat transfer occurs via a number of different mechanisms:

- conduction
- convection
- radiation
- mass transfer (latent energy effects).

When modelling heat transfer in general systems, some, or all, of the heat transfer mechanisms listed above can occur. Ideally, it is desirable to be able to determine the temperature at any time and at any point in the system. To do this it is necessary to solve simultaneously the governing equations of each heat transfer mechanism as it applies to the system. In this study, however, only the temperature distribution in solids is considered and, therefore, only the heat conduction equation is rigorously modelled. However, the influence of the environment on the temperature distribution in the solid must be included in the analysis. This is achieved by defining conditions on the boundary of the solid which reflect the heat transfer occurring in the environment.

Carslaw and Jaeger 1959 present analytical solutions to the heat conduction equation for many different problems. However, all these problems involve uniform boundary conditions and very simple geometries. In order to solve more realistic problems, it is necessary to use numerical methods. Traditionally, numerical heat transfer conduction analysis has been done using variants of the finite difference method which results in a pointwise approximation to the heat conduction equation. It is a versatile and useful method but becomes difficult to use with irregular geometries and more complex boundary conditions. More recently, the finite element method and the boundary element method have been used. The finite element method generates a piecewise solution to the heat conduction equation. It is able to model a greater range of problems than the finite difference method. The boundary element method generates a piecewise solution to the heat conduction equation. Unlike the finite element method which involves dividing the domain into a number of sub-domains, the boundary element method involves dividing the boundary of the domain into segments. An appropriate integral form of the heat conduction equation is then solved over each boundary segment.

In this thesis the finite element method is used to solve the heat conduction equation since this method was previously used by Korvink 1986. The method models the solution region as a collection of subregions or elements. A number of discrete points are selected in the domain - nodal points - and the temperature over each element is approximated by a linear combination of simple, known functions and unknown constants. These temperature approximations are substituted into a weakened, integral form of the governing partial differential equation and the resulting equation is multiplied by a weighting function and forced to vanish in an average sense. This process is repeated for each element and the resulting element equations are grouped into a system of simultaneous equations. The time derivative term in the governing equation is discretised using a finite difference scheme. The assembled set of element equations is then solved for the unknown parameters which, by judicious choice of approximation functions, represent the nodal temperatures.

1.3 Thesis objectives

The aim of this thesis is to develop a finite element analysis program capable of modelling the temperature distribution in solids. Using the heat transfer which occurs in a mine as a guide requires that the program have the features listed below.

1) The program must model the nonlinear, transient heat conduction equation together with the boundary and internal conditions listed below

    - prescribed flux boundary condition
    - prescribed temperature boundary condition
    - point heat sources
    - radiation boundary condition
    - convection boundary condition
    - condensation and evaporation effects (mass transfer boundary condition)
    - internal heat generation effects.

    The program must also be able to model infinite and variable domains. The former requirement is modelled using infinite elements and the latter using the concepts of element death and birth.

2) All boundary conditions must be functions of time and it must be possible to specify a time dependent analysis time-step size.

3) The program must be able to model non-isotropic solids.

4) The program must be accompanied by a detailed user's guide.

Korvink 1986 developed a nonlinear, transient two-dimensional heat conduction finite element analysis program as part of NOSTRUM (Duffet et al. 1983), a nonlinear structural mechanics analysis program, which is capable of modelling problems with prescribed temperature, prescribed flux and internal heat generation boundary and internal conditions. It was therefore necessary to extract the thermal and other necessary finite element analysis routines from NOSTRUM and combine them into a separate thermal analysis program. The program was extended to include the conditions listed above.

This document is comprised of the following sections:

- Chapter 2: Pertinent heat transfer theory. This chapter presents a discussion of the different mechanisms of heat transfer considered, the derivation of the heat conduction equation and the mathematical details of convection, radiation, evaporation and condensation.

- Chapter 3: Finite element analysis in the context of thermal problems. In this chapter the development of the finite element method is described. The details of the isoparametric formulation and infinite elements are presented, followed by a concise discussion of numerical integration.

- Chapter 4: Solution of the finite element equations. This chapter deals with the time-integration of the finite element equations derived in chapter 3. It describes the details of the scheme used to deal with nonlinear problems and presents a brief discussion on the stability of finite element analyses.

- Chapter 5: Implementation of the finite element method. This chapter presents details of the overall structure of the program and the manner in which the extensions have been implemented in the program.

- Chapter 6: Validative sample problems. This chapter presents numerical and analytical solutions to a number of problems selected to verify various modelling capabilities of the program.

- Chapter 7: Conclusions and recommendations.

CHAPTER 2 : PERTINENT HEAT TRANSFER THEORY

The process of heat transfer may be clearly described in terms of general statements of the first ~~and~~ second laws of thermodynamics. One of the ways in which the first law may be formally stated is 'Although energy assumes many forms, the total quantity of energy is constant, and when energy disappears in one form it appears simultaneously in other forms' (Smith *et al*. 1975). This is a generalisation of the law of conservation of energy. The statement of the second law which is applicable is 'Any process which consists solely in the transfer of heat from one temperature to a higher one is impossible'.

Having defined the framework within which energy transport occurs, it is useful to define various mechanisms of heat transfer.

- Heat transfer by conduction
- Heat transfer by convection
- Heat transfer by mass transfer
- Heat transfer by radiation

The first three mechanisms of heat transfer can be directly attributed to interactions at the molecular level within a material, while the fourth, radiation, differs in that heat is transferred by means of the interaction of electromagnetic waves, which require no medium through which to propagate, with the molecules of a material.

These heat transfer mechanisms can occur singly or simultaneously in a system which might be comprised of a number of material types and phases.

One of the aims of studying and attaining comprehension of the fundamentals of energy transport, is to be able to define the principles governing the process in concise mathematical form for subsequent use in modelling the temperature and flux distribution in some system of interest. For instance, it might be desirable to be able to predict the temperature distribution within and the fluxes at the interfaces of a system of solid, liquid and gas phases of various materials. Although this degree of generality is desirable, in this study the following limiting assumptions have been made:

- Only the temperature distributions in two-dimensional solids are of interest
- All mechanisms of heat transfer, apart from conduction, are treated in the context of their influence through the boundaries of the conduction system.

In this chapter, the conduction equation is derived and it scope is defined. Convective heat transfer is then discussed in more detail, as is heat transfer due to both radiation and mass transfer effects.

2.1 DERIVATION AND DEFINITION OF THE HEAT TRANSFER CONDUCTION EQUATION

Heat transfer by conduction occurs due to the fact that molecules manifest higher energy levels at higher rates of rotation, vibration and translation. These higher energy molecules transfer energy to other lower energy molecules by molecular collisions (or in the case of pure metals, by means of the transfer of free electrons).

At the macroscopic level, heat transfer can be mathematically described in terms of a constitutive law - Fourier's Law of heat conduction - and a conservation law, which must be derived within the context of the limiting assumptions mentioned above.

Fourier's Law can be written as:

$$q_i = -k_i \frac{\partial T}{\partial x_i} \qquad (i = 1,2;\ \text{no sum}) \qquad [2.1]$$

where q_i = heat flux in the x_i th direction (W/m^2)
k_i = heat conductivity in the x_i th direction (W/m°C)
T = the temperature field $(T(\mathbf{x}))$ (°C)
x_i = the i^{th} principle direction (m)

If the material is isotropic, then Fourier's law can be written in more concise form:

$$q = k\nabla T \qquad [2.2]$$

The thermal conductivity is the term of greatest interest in Fourier's law and will be discussed later.

In order to derive the mathematical expression of the conservation law it is best to analyse the situation in three dimensions.

Consider a body, Ω, in real space R^3 with boundary denoted Γ, such that any point x within Ω or on Γ can be defined by:

$$x = \{x_i\} \ , \ i = 1,2,3 \qquad [2.3]$$

Next consider an arbitrary element of the body, as represented in figure 2.1.

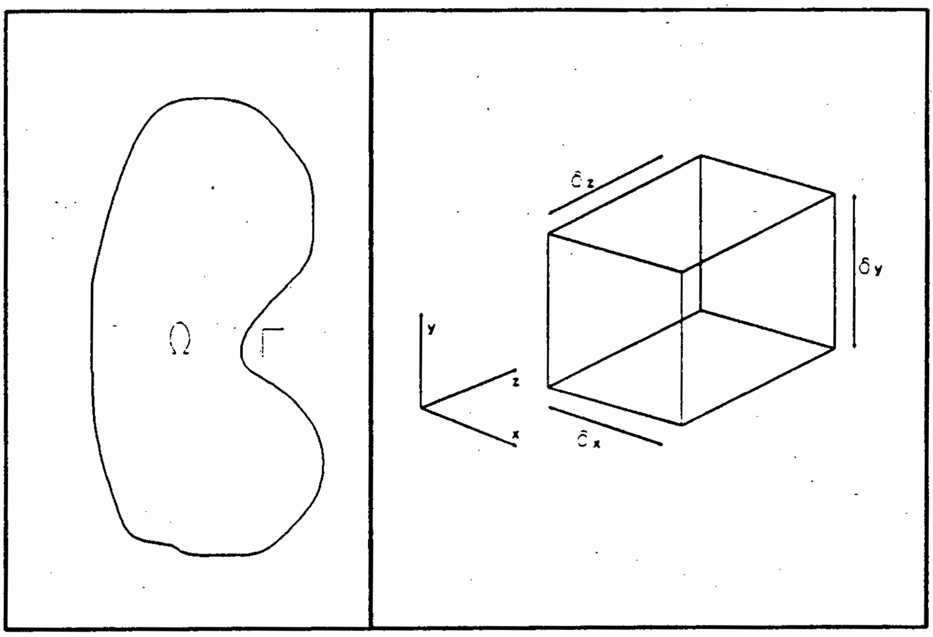


Figure *2.1 Problem domain and an arbitrary element thereof.*

The appropriate conservation law can be derived by performing an energy balance over the element shown in figure 2.1, ignoring mechanical energy effects and bearing in mind that the material is a solid. The energy balance can be expressed as:

Rate of accumulation of internal energy = rate of heat addition by conduction - rate of heat removal by conduction + rate of heat generated internally.

Each term in this expression can be written in equation form as:

<u>ENERGY IN</u>

$$Q_x = \left[-k\delta y\delta z\frac{\partial T}{\partial x}\right]_x$$

$$Q_y = \left[-k\delta x\delta z\frac{\partial T}{\partial y}\right]_y$$

$$Q_z = \left[-k\delta x\delta y\frac{\partial T}{\partial x}\right]_z$$

<u>ENERGY OUT</u>

$$Q_{x+\delta x} = \left[-k\delta y\delta x\frac{\partial T}{\partial x}\right]_{x+\delta x}$$

$$Q_{y+\delta y} = \left[-k\delta x\delta z\frac{\partial T}{\partial y}\right]_{y+\delta y}$$

$$Q_{z+\delta z} = \left[-k\delta x\delta y\frac{\partial T}{\partial z}\right]_{z+\delta z}$$

<u>INTERNAL ENERGY</u>

$$Q_i\delta x\delta y\delta z$$

<u>ACCUMULATION</u>

$$\delta x\delta y\delta z\rho C_p\frac{\partial T}{\partial t}$$

Combining all the terms, rearranging the resulting equation, dividing by $\delta x\delta y\delta z$ and taking the limit as δx, δy and δz tend to zero results in

$$\frac{\partial}{\partial x}\left(k_x\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k_z\frac{\partial T}{\partial z}\right) + Q_i = \rho C_p\frac{\partial T}{\partial t} \qquad [2.4]$$

or, in more concise form:

$$\rho C_p\frac{\partial T}{\partial t} = \nabla.\mathbf{Kg} \qquad [2.5]$$

where	ρ	=	density (kg/m^3)
	C_p	=	specific heat capacity (J/kg°C)
	T	=	temperature (°C)
	t	=	time (seconds)
	$\mathbf{K}$	=	thermal conductivity matrix (W/m°C)
	$\mathbf{g}$	=	$\left\{\frac{\partial T}{\partial x}, \frac{\partial T}{\partial y}, \frac{\partial T}{\partial z}\right\}^T$
	Q_i	=	internal heat generation (W/m^3)

The material property terms which appear in this equation are thermal conductivity, density and heat capacity, all of which may be functions of position and temperature. The degree of variation of these properties is strongly dependent on the nature of the material. Generally, the temperature dependence of these material properties is weak.

Values of these material properties can be found in most physical and thermodynamic databooks. If physical data cannot be found, methods of prediction are presented in the literature (Perry and Chilton 1974).

In order to apply equation 2.4 to a particular problem it is necessary to define boundary and initial conditions. The latter consists of defining an initial temperature distribution throughout the problem domain whereas the definition of boundary conditions is somewhat more complicated. Firstly, there are two types of admissible boundary conditions:

1) Condition on T (Derichlet)

$$T = T_{\Gamma} \text{ on } \Gamma_D$$

2) Condition on $\partial T/\partial n$ (flux condition)

$$\frac{\partial T}{\partial n} + \alpha T = f_N \text{ on } \Gamma_N$$

if $\alpha \neq 0$, then the boundary condition is known as a Cauchy condition. If $\alpha = 0$, then it is known as a Neuman condition.

It is possible that both of these types of boundary conditions occur in the same problem and for different areas of the boundary to have boundary conditions of different values, but of the same type. This is often the case in real problems and is one of the factors which makes analytical solution of the heat equation extremely difficult, if not impossible.

It is mandatory that the entire boundary has some condition defined on it, otherwise the problem is ill-conditioned.

2.2 CONVECTION HEAT TRANSFER

Heat transfer by convection occurs only in fluids and is effected by the movement of the fluid. This movement can be induced by external means such as a fan - forced convection - or as a result of temperature difference induced density gradients within the fluid - natural or free convection.

In order to determine the temperature distribution in a fluid experiencing convective heat transfer, it is necessary to analyse the fluid dynamics of the system and then carry out an energy balance on it. The appropriate equations of change for pure fluids which must be solved are listed below (Bird et al. 1960):

Continuity:
$$\frac{D\rho}{Dt} = -\rho(\nabla\cdot\mathbf{v}) \qquad [2.6a]$$

Motion: Forced:
$$\rho\frac{D\mathbf{v}}{Dt} = -\nabla p - [\nabla\cdot\tau] + \rho\mathbf{g} \qquad [2.6b]$$

Free:
$$\rho\frac{D\mathbf{v}}{Dt} = -[\nabla\cdot\tau]-\rho\beta\mathbf{g}(T-T_{ref}) \qquad [2.6c]$$

Energy:
$$\rho C_v\frac{DT}{Dt} = -(\nabla\cdot q) - T\left(\frac{\partial p}{\partial T}\right)_\rho(\nabla\cdot\mathbf{v}) - (\tau:\nabla\mathbf{v}) \qquad [2.6d]$$

where

ρ	=	density (kg/m^3)
p	=	fluid pressure (Pa)
$\mathbf{g}$	=	gravitational acceleration vector (m/s^2)
$\mathbf{v}$	=	mass average velocity vector (m/s)
τ	=	viscous stress tensor (kg/s^{2}m)
β	=	thermal coefficient of volumetric expansion (K^{-1}) $\frac{1}{V}\left(\frac{\partial V}{\partial T}\right)_p$
q	=	heat flux vector (W/m^2)
T	=	temperature (K)
V	=	volume (m^3)
t	=	time (s)
$\frac{D}{Dt}$	=	substantial derivative: $\left\{\frac{\partial}{\partial t} + (\mathbf{v}\cdot\nabla)\right\}$
$\tau:\nabla\mathbf{v}$	=	$\tau_{ij}\frac{\partial}{\partial x_j}v_i$

In order to determine the temperature distribution in a fluid, it is necessary to solve the above equations. If a system comprised of a solid medium as well as a fluid medium were being analysed, it would be necessary to solve both the heat conduction equation and the equations presented above, under the constraint that the fluid and solid temperatures are the same at the interface separating the two phases.

In this study, only the temperature distribution in solid media is of interest. Therefore, it seems that it requires a great deal of computational effort to include the effects of convection heat transfer in the fluid phase on the temperature distribution in the solid. Fortunately, considering the convection heat transfer in the fluid as a boundary condition of the conduction system, greatly simplifies the analysis at the cost of no longer being able to model the temperature distribution in the fluid. This is made possible by using "Newton's law of cooling":

$$Q = hA(T - T_f) \qquad [2.7]$$

where	Q	=	rate of energy flow (W)
	h	=	heat transfer coefficient (W/m^2°C)
	T	=	interface temperature (°C)
	T_f	=	temperature in the fluid (°C)
	A	=	interfacial surface area (m^2)

Equation 2.7 is not really a law, but rather the defining equation for h, the heat transfer coefficient. Table 2.1 presents the range of magnitude of h for various systems.

Table *2.1 Order of Magnitude of the Heat Transfer Coefficient.*

Mode	h (W/m^2°C)
Free convection	
Gases	4 - 24
Liquids	120 - 700
Boiling water	1200 - 23000
Forced convection	
Gases	10 - 120
Viscous liquids	60 - 600
Water	600 - 11600
Condensing vapour	1200 - 116000

The heat transfer coefficient can be determined experimentally or from analytically derived correlations which have been adapted to conform to experimental data. These correlations are presented in many heat transfer texts and databooks and generally have the form:

$$Nu = f(Re,Pr) \quad : \text{forced} \qquad [2.8]$$

$$Nu = f(Gr,Pr) \quad : \text{free} \qquad [2.9]$$

where Nu = Nusselt number for heat transfer
Re = Reynold's number
Pr = Prandtl number
Gr = Grashof number

$$Nu = \frac{hL}{k} \qquad Pr = \frac{C_p\mu}{k} \qquad Re = \frac{vL\rho}{\mu} \tag{2.10}$$

$$Gr = \frac{g\beta(T_w - T_\infty)L^3\rho^2}{\mu^2}$$

where h = heat transfer coefficient ($W/m^{2o}C$)
L = characteristic length (m)
k = fluid thermal conductivity (W/m^oC)
C_p = fluid heat capacity (J/kg^oC)
μ = fluid viscosity (kg/m.s)
v = fluid mass average velocity (m/s)
ρ = fluid density (kg/m^3)
g = gravitational acceleration (m/s^2)
β = thermal coefficient of volumetric expansion (K^{-1})
T_w = surface temperature (K)
T_∞ = bulk fluid temperature (K)

The fluid physical properties are all functions of temperature and prediction equations are available for all of them in the literature (Perry and Chilton 1974)

It is important to note that the heat transfer coefficient is generally defined in terms of a particular temperature difference; for instance, logarithmic mean temperature difference.

Examination of equation 2.7 shows that it may be used as a Cauchy boundary condition of the heat conduction equation.

2.3 Radiation heat transfer

Radiation heat transfer is effected by an electromagnetic wave mechanism which allows energy to be transported at the speed of light through regions of space which are devoid of any matter. This is clearly a very different mechanism from those of conduction and convection.

When a solid body is heated, the solid surface emits radiation of wavelength primarily in the range 0.1 to 10 microns. This radiation is usually referred to as thermal radiation. As this is the only radiation of interest in this study, the term 'radiation' is freely used to denote thermal radiation. In order to fully understand the radiation phenomenon, a quantum mechanics approach must be adopted. However, it is possible to qualitatively describe it as follows: when energy is supplied to a body, some of the constituent molecules or atoms are raised to "excited states". There is a tendency for these atoms or molecules to return spontaneously to lower energy levels and, when this occurs, energy is emitted in the form of electromagnetic radiation which is distributed over a range of wavelengths. The above is a description of the emission of radiant energy, when a molecular system goes from a high to a low energy state. The reverse process, absorption, occurs when the addition of radiant energy causes the molecular or atomic system to go from a low to a high energy state.

When radiant energy interacts with a medium, be it fluid or solid, part of the radiation is reflected or absorbed and part is transmitted. It is useful to define the reflectivity, ρ, the fraction reflected, the absorptivity, α, the fraction absorbed and the transmissivity, τ, the fraction transmitted. Thus, for non-fluorescing media:

$$\rho + \alpha + \tau = 1 \qquad [2.11]$$

and, generally for solids, most of which do not transmit radiation

$$\rho + \alpha = 1 \qquad [2.12]$$

It is important to note that ρ, α and τ are all functions of the wavelength of the radiation.

Reflection of radiation from a surface may be specular or diffuse. Specular reflection refers to the case when the angle of incidence is equal to the angle of reflection. Diffuse reflection refers to the case when an incident beam of radiation is distributed uniformly in all directions after it has been reflected.

Having dealt with some of the qualitative aspects of radiation, it is now appropriate to commence a more rigorous approach. For the purposes of the subsequent development, it is useful to conceive of a body which does not transmit radiation and which absorbs all incident radiation. This hypothetical body is known as a black body.

Planck's distribution law gives the radiated energy flux from a black surface for radiation of wavelength λ

$$E_{b\lambda} = \frac{2\pi c^2 h}{\lambda^5} \frac{1}{e^{ch/\lambda\kappa T} - 1} \qquad [2.13]$$

where $E_{b\lambda}$ = black body emissive power at wavelength λ (W/m^2)
c = velocity of light (2.9979•10^8 m/s)
h = Planck's constant (6.625•10^{-34} Js)
λ = wavelength of radiation (m)
T = absolute temperature (K)
κ = Boltzmann's constant (1.38049•10^{-23} J/molecule K)

When Planck's distribution law is interpreted over all wavelengths, the Stefan-Boltzmann law results, which relates the total energy emitted to the fourth power of the absolute temperature.

$$E_b = \sigma T^4 \qquad [2.14]$$

where E_b = black body emissive power (W/m^2)
σ = Stefan-Boltzmann constant (5.667•10^{-8} W/m^{2}K^4)
T = absolute temperature (K)

Although the concept of a black body is useful for the mathematical analysis of radiation fundamentals, the aim of radiation analysis is to be able to deal with real surfaces - at the very least using a more realistic model than the black body hypothesis. This is done using a combination of the concept of emissivity, Kirchoff's law and the gray body assumption.

The emissivity of a body, ϵ, is defined as the ratio of the emissive power of a real body and the emissive power of a black body at the same temperature:

$$\epsilon = \left.\frac{E}{E_b}\right|_T \qquad [2.15]$$

Rearranging equation 2.15 and substituting equation 2.14 results in

$$E = \epsilon E_b = \epsilon\sigma T^4 \qquad [2.16]$$

which equation enables the emissive power of a real body to be determined if the emissivity, ϵ, is known. Emissivity data can be found in the literature, for example Kern 1965.

Kirchoff's law states that:

$$\epsilon = \alpha \tag{2.17}$$

where ϵ and α are the total properties of the particular material. That is, they represent the integrated behaviour of the material over all wavelengths. In reality, both of these properties vary widely with the temperature and wavelength of the radiation, which complicates the radiation analysis. To avoid these complications, the gray body assumption is used which is defined such that the monochromatic emissivity, ϵ_λ, of the body is independent of wavelength. ϵ_λ is defined as:

$$\epsilon_\lambda = \left.\frac{E_\lambda}{E_{b\lambda}}\right|_T \tag{2.18}$$

It can be stated that

$$\epsilon = \epsilon_\lambda \tag{2.19}$$

As a consequence of equations 2.19 and 2.17 (Kirchoff's law), it may be stated that:

$$\epsilon_\lambda = \alpha_\lambda \tag{2.20a}$$

and $$\alpha_\lambda = \text{constant} \tag{2.20b}$$

Effectively, the above equations, which are the consequence of the gray body assumption, mean that both emissivity and absorptivity are constant over all wavelengths, which implies that they are independent of temperature.

In reality, real surfaces are not "gray" and significant errors may arise from assuming gray body behaviour. However, radiation analysis is so complicated when real body behaviour is assumed that the gray body assumption is justified, to some extent, by its simplicity and ease of use.

Most of the mathematical tools necessary for analysing radiant heat transfer between two or more surfaces have been presented. First, however, the concept of shape factors must be considered.

Whenever radiant heat transfer occurs between two surfaces, be they black or gray, a portion of the radiant energy leaving one surface does not intercept the receiving surface. In order to quantify this phenomenon, a shape factor value, F_{ij}, is determined which is dependent on the orientation of the two surfaces with respect to each other and which may be defined as the fraction of energy leaving surface i which reaches surface j. An expression for the evaluation of shape factors which must be interpreted with reference to figure 2.2, is presented below.

$$F_{12} = \frac{1}{\pi A_1}\int_{A_1}\int_{A_2}\frac{\cos\theta_1\cos\theta_2}{s^2}\,dA_1dA_2 \qquad [2.21a]$$

and

$$F_{21} = \frac{1}{\pi A_2}\int_{A_1}\int_{A_2}\frac{\cos\theta_1\cos\theta_2}{s^2}\,dA_1dA_2 \qquad [2.21b]$$

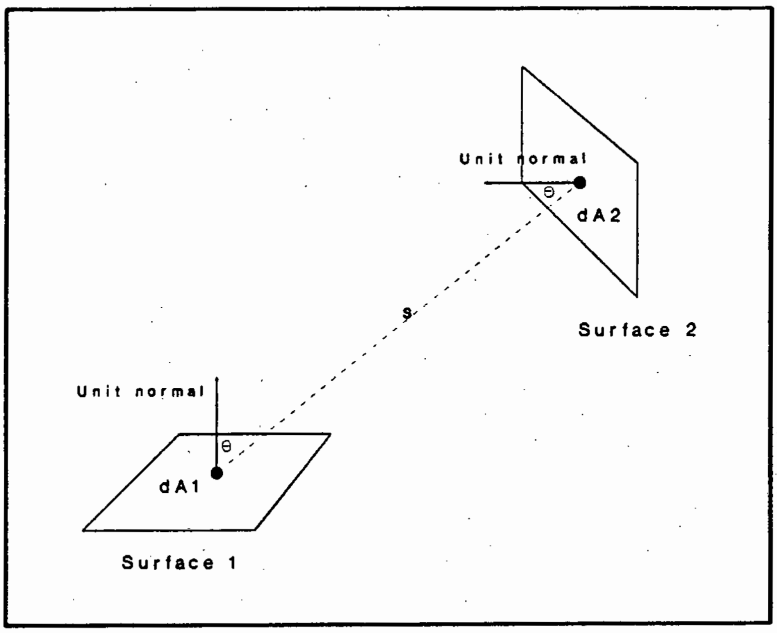


Figure *2.2 Shape factor reference configuration.*

The analytical calculation of shape factors is extremely difficult except for a few simple configurations such as two parallel planes of the same size or rectangles with a common side. Shape functions for various simple configurations are presented in graphical form in the radiation literature (Hottel 1967).

It is now possible to determine the net radiation energy transfer between two arbitrarily positioned black surfaces at different temperatures.

$$
\begin{aligned}
Q_{ij} &= A_i F_{ij} E_{bj} - A_j F_{ji} E_{bj} \\
&= A_i F_{ij} (E_{bi} - E_{bj}) \\
&= \sigma A_i F_{ij} (T_i^4 - T_j^4)
\end{aligned}
\qquad [2.22]
$$

where Q_{ij} = net energy lost by surface i to surface j (W)
A_i = surface area of surface i (m^2)
T_i = surface temperature of surface i (K)

Determining the radiant heat exchange between non-black bodies is not as simple as equation 2.22 indicates, as a fraction, ρ, of the incident radiation is reflected. To proceed, it is necessary to define

G_i = irradiation - total radiation incident upon a surface i per unit time and area (W/m^2)
G_{ij} = irradiation - total radiation incident upon surface i originating at surface j (W/m^2)
J_i = radiosity - total radiation which leaves a surface i per unit time and area (W/m^2)

In order to keep the analysis simple, it is necessary to impose the additional assumption that G_i and J_i are uniform over each surface.

It is possible to write

$$J_i = \epsilon_i \sigma T_i + (1 - \epsilon_i) G_i \qquad [2.23]$$

$$G_i = \Sigma_j G_{ij} \qquad j = 1,2,\ldots,n \qquad [2.24]$$

$$G_{ij} = [J_j A_j F_{ji}]/A_i \qquad [2.25]$$

The above three equations can be combined to yield:

$$\Sigma_j [\delta_{ij} - (1 - \epsilon_i) F_{ij}] J_j = \epsilon_i \sigma T_i^4 \qquad i,j = 1,2,\ldots,n \qquad [2.26]$$

where n = the number of surfaces - number of equations
δ_{ij} = Kronecker delta
i = surface i
j = surface j

Equation 2.26 represents a system of linear, simultaneous equations in J. Once the radiosity of each surface in the system has been determined, the net flux leaving surface i can be evaluated using the equation below:

$$q_i = J_i - G_i = J_i - \frac{(J_i - \epsilon_i \sigma T_i^4)}{(1 - \epsilon_i)}$$

$$= \frac{\epsilon_i}{1 - \epsilon_i} [\sigma T_i^4 - J_i] \qquad [2.27]$$

This development ignores the possible existence of absorbing media between the interacting surfaces.

Radiation exchange between a gas and a solid surface is considerably more complex than that dealt with so far. Gases are in many cases transparent to radiation and, when they do absorb or emit radiation, they usually do so in narrow wavelength bands. As a general rule, polar and asymmetrical gas molecules are radiatively active - for example carbon dioxide and water. Generally, gases are non-reflective.

In this treatment only gases will be considered. The reason for this is that at temperatures high enough for radiation heat transfer to be appreciable, most liquids are gases (at normal pressures). Transmitting solids are not considered as they are generally reflective to some extent and usually transmit radiation in a very narrow wavelength band.

As the calculation of gas-radiation properties is extremely complicated, an approach developed by Hottel 1967 for engineering purposes is used in this study. This simplified approach is used to calculate the emissivities of carbon dioxide and water at various temperatures and pressures.

Hottel 1967 presents the gas emissivities in graphical form - gas emissivity plotted against temperature for various values of the partial pressure of the gas multiplied by the mean beam length of the system. The mean beam length is a factor used to account for the fact that not all radiation passing through an absorbing gas travels the same distance. Tables of mean beam lengths for various system configurations can also be found in Hottel 1967. The gas emissivity-temperature graphs assume a total system pressure of one atmosphere. As this is not always the case in a real situation, these graphs are accompanied by a correction factor versus total system pressure graph.

Having determined the gas emissivity and assuming that Kirchoff's identity applies, it can be written:

$$\alpha_m + \tau_m = 1 = \epsilon_m + \tau_m \qquad [2.28]$$

where	α_m	=	absorptivity of the transmitting-absorbing medium
	τ_m	=	transmissivity of the medium
	ϵ_m	=	emissivity of the medium

It is now possible to attempt the evaluation the radiant heat exchange between a group of gray surfaces which interact with a gray transmitting-absorbing medium at uniform temperature, T_m

$$J_i = \epsilon_i \sigma T_i^4 + (1-\epsilon_i)G_i \qquad [2.29a]$$

$$G_i = \Sigma_j G_{ij} \qquad j = 1,2,\ldots,n \qquad [2.29b]$$

$$G_{ij} = [(J_j A_j F_{ji})\tau_{mji} + \epsilon_{mij}\sigma T_m^4 A_j F_{ji}]/A_i \qquad [2.29c]$$

Using view factor reciprocity and substituting equations 2.29c and 2.29b into equation 2.29a yields:

$$J_i = \epsilon_i \sigma T_i^4 + (1-\epsilon_i)\Sigma_j[(J_j A_i F_{ij})\tau_{mji} + \epsilon_{mij}\sigma T_m^4 A_i F_{ij}]/A_i$$
$$j = 1,2,\ldots,n \qquad [2.29d]$$

and $J_i = \delta_{ij} J_j$ [2.29e]

$$\Sigma_j[\delta_{ij} - (1 - \epsilon_i)F_{ij}\tau_{mji}]J_j = \epsilon_i \sigma T_i^4 + (1 - \epsilon_i)\sigma T_m^4 \Sigma_j \epsilon_{mij} F_{ij}$$
$$i,j = 1,2,\ldots,n \qquad [2.30]$$

where	n	=	the number of surfaces - number of equations
	J_i	=	radiosity from surface i (W/m^2)
	G_i	=	irradiation to surface i (W/m^2)
	G_{ij}	=	irradiation from surface j to surface i (W/m^2)
	δ_{ij}	=	Kronecker delta
	ϵ_i	=	the emissivity of surface i
	τ_{ij}	=	the transmissivity of the medium separating surfaces i and j; includes the effects of

geometry

F_{ij} = the shape factor between surfaces i and j

σ = Stefan-Boltzmann constant $5.669 \cdot 10^{-8}$ (W/m^2K^4)

T_i = Absolute temperature of surface i (K)

T_m = Absolute temperature of the medium between surfaces (K)

ϵ_{mij} = emissivity of the medium between surfaces i and j including geometrical effects

Equation 2.30 is a system of simultaneous linear equations which can be solved for the J_i's. The surface flux, q_i, can be evaluated using equation 2.27

2.4 Mass transfer heat transfer

In the context of this study, mass transfer heat transfer implies the transfer of energy in or between media due to the movement of one or more of the chemical species which comprise the media. In addition to the transfer of sensible heat in this fashion, energy is also transferred in the form of latent heat due to phase changes within or between media.

The definition of heat transfer by mass transfer above appears to include convective heat transfer. However, while convective heat transfer generally occurs together with mass transfer heat transfer, the latter mechanism of heat transfer is due to concentration gradients of a particular chemical species within a mixture. This implies that mass transfer heat transfer cannot occur in pure fluids, while convection does and can, whereas both mechanisms occur in fluid mixtures.

In this treatment of mass transfer, only binary mixtures are considered - mixtures containing only species A and species B. The rigorous theory presented below is not complete and not general. It is presented to indicate the complexity of modelling of mass transfer problems analytically.

As in the case of heat conduction problems, mass transfer in binary mixtures can be presented in terms of a conservation law - the equations of continuity (or conservation of mass) - and a constitutive law - Fick's law of binary diffusion.

$$N_A - x_A(N_A + N_B) = -c\mathcal{D}_{AB}\nabla x_A \qquad [2.31]$$

where N_A = molar flux of species A (kgmol/m^{2}s)
x_A = mole fraction of species A
c = molar density of solution (kgmol/m^3)
D_{AB} = mass diffusivity (m^2/s)

In order to determine the concentration and temperature distribution in a binary system, it is necessary to solve the equations of change of the system. These equations are:

Continuity: $$\frac{\partial \rho_i}{\partial t} = -(\nabla \cdot \mathbf{n}) + r_i \qquad i = A,B \qquad [2.32a]$$

Motion: $$\frac{\partial (\rho \mathbf{v})}{\partial t} = -[\nabla \cdot \boldsymbol{\phi}] + \rho_A \mathbf{g}_A + \rho_B \mathbf{g}_B \qquad [2.32b]$$

Energy: $$\frac{\partial \rho \{U+v^2/2\}}{\partial t} = -(\nabla . \mathbf{e}) + \mathbf{n}_A \cdot \mathbf{g}_A + \mathbf{n}_B \cdot \mathbf{g}_B \qquad [2.32c]$$

and

$$\mathbf{n}_i = \rho \omega_i \mathbf{v} + \mathbf{j}_i \qquad i = A,B \qquad [2.32d]$$

$$\boldsymbol{\phi} = \rho \mathbf{v}\mathbf{v} + \boldsymbol{\pi} \qquad [2.32e]$$

$$\boldsymbol{\pi} = \boldsymbol{\tau} + p\boldsymbol{\delta} \qquad [2.32f]$$

$$\mathbf{e} = \rho\{U+v^2/2\}\mathbf{v} + \mathbf{q} + [\boldsymbol{\pi} \cdot \mathbf{v}] \qquad [2.32g]$$

where ρ_i = mass concentration of species i (kg of i/m^3)
ρ = mass density of solution (kg/m^3)
ω_i = mass fraction of species i
$\mathbf{v}$ = mass average velocity (m/s)
r_i = rate of production of i (kg/m^{2}s)
$\mathbf{g}_i$ = external force acting on species i (N/kg)
U = internal energy (J/kg)
v = magnitude of local fluid velocity (m/s)
$\mathbf{n}_i$ = mass flux of species i relative to stationary coordinates (kg/m^{2}s)
$\mathbf{j}_i$ = mass flux of species i relative to mass average velocity (kg/m^{2}s)
$\boldsymbol{\phi}$ = momentum flux relative to stationary coordinates (kg/ms^2)
$\boldsymbol{\pi}$ = pressure tensor (kg/ms^2)
p = fluid pressure (kg/ms^2)
$\boldsymbol{\delta}$ = unit tensor
$\mathbf{e}$ = energy flux relative to stationary coordinates (kg/s^3)
$\mathbf{q}$ = energy flux relative to mass average velocity (kg/s^3)

The above equations refer to a coordinate system fixed in space.

The complete description of mass, momentum and energy transport in a binary system is contained in these equations. One also requires the thermal equation of state, $p = p(\rho,T,x_i)$, the caloric equation of state, $U = U(\rho,T,x_i)$ and information about the chemical kinetics of the system for use in the reaction term. Also required are explicit expressions for the fluxes j_i, π and q, in terms of the gradient and transport coefficients, the temperature, density and composition dependencies of which must also be known.

As has been mentioned in the convection heat transfer section, only the temperature distribution in solid media is of interest in this study. In view of this and in order to simplify the complexity of determining the effect of mass transfer heat transfer on the temperature distribution in the solid, an approximation analogous (at least initially) to that in the convection section, is used to model mass transfer heat transfer as a boundary condition.

Before the equations are presented, it is necessary to define the limitations of the mass transfer theory to be used. Firstly, only binary mixtures are considered in which only one of the components is condensable in the temperature and pressure ranges considered. Henceforth, the condensable component in the gaseous phase is called vapour and the other gaseous component, gas. The air-water vapour system is an example of this. Secondly, it is assumed that the solubility of the non-condensable gas in the liquid phase of the condensable component is negligible. Thirdly, it is assumed that the solubility of the solid is negligible. Fourthly, ideal gas and liquid behaviour is assumed.

The mass transfer equation equivalent to Newton's law of cooling for convection is:

$$q = Lh_m(p_{dew} - p_s) \qquad [2.33]$$

where q = latent heat flux (W/m^2)
p_{dew} = vapour saturation pressure at T_{dew} (Pa)
p_s = vapour saturation pressure at T_s (Pa)
T_{dew} = dew-point temperature of the gas-vapour mixture (°C)
T_s = liquid layer or solid surface temperature (°C)
L = latent heat of vaporisation of the vapour at T_{dew} (J/kg)
h_m = mass transfer coefficient (kg/m^2sPa)

Equation 2.33 predicts the energy transfer due to condensation of vapour from the gaseous phase or evaporation of vapour from the liquid phase.

The latent heat of vaporisation of a vapour is a function of temperature and can be predicted using Watson's correlation (Perry and Chilton 1974).

$$\frac{L}{L_r} = \left[\frac{T_c - T}{T_c - T_r}\right]^{0.38} \qquad [2.34]$$

where L_r = reference latent heat of vaporisation (J/kg)
T_c = vapour critical temperature (K)
T_r = reference temperature (K)
L = latent heat of vaporisation at T (J/kg)

The saturation pressure may be taken from tables in a databook such as Perry and Chilton 1974, or predicted using Antoine's equation (Himmelblau 1982)

$$\ln(p) = A - \frac{B}{C + T} \qquad [2.35]$$

where p = vapour saturation pressure (Pa)
A,B,C = species specific constants
T = temperature (°C)

The last term on the right hand side of equation 2.33, which must still be evaluated, is the mass transfer coefficient, h_m. This term is difficult to evaluate directly and the mathematical analysis which yields the various forms of the heat transfer coefficient correlations in convection analysis, is not generally possible in the case of mass transfer. There is, however, a means of predicting h_m indirectly which is presented below:

If one considers the mechanism of mass transfer, it may be appreciated that there are definite similarities to the mechanism of convective heat transfer. It is also clear that whenever condensation or evaporation occurs, so does convection heat transfer. Investigation of these apparent similarities resulted in the Chilton-Colburn j-factor analogy which relates mass and heat transfer coefficients by equating two dimensionless quantities, j_H and j_M.

$$j_H = \frac{h}{\rho v C_p} Pr^{2/3} \qquad [2.36a]$$

$$j_M = h_m \frac{RT\, p_g}{vM_v p} Sc^{2/3} \qquad [2.36b]$$

where	h	=	heat transfer coefficient ($W/m^{2o}C$)
	ρ	=	fluid density (kg/m^2)
	v	=	fluid average mass velocity (m/s)
	C_p	=	fluid heat capacity (J/kg^oC)
	μ	=	fluid viscosity (kg/ms)
	k	=	fluid thermal conductivity (W/m^oC)
	h_m	=	mass transfer coefficient (kg/m^2sPa)
	R	=	Ideal Gas constant $8.314 \cdot 10^3$ ($Pam^2/kgmolK$)
	T	=	fluid temperature (K)
	M_v	=	vapour molecular mass (kg/kgmol)
	p_g	=	partial pressure of the gas (Pa)
	p	=	total gas-vapour mixture pressure (Pa)
	D_{AB}	=	mass diffusivity (m^2/s)
	Pr	=	Prandtl number ($C_p\mu/k$)
	Sc	=	Schmidt number ($\mu/\rho D_{AB}$)

Equating Chilton-Colburn j-factors results in the following expression for the mass transfer coefficient, h_m.

$$h_m = \frac{h}{\rho C_p} \frac{M_v}{RT} \frac{p}{p_g} \left[\frac{Pr}{Sc}\right]^{2/3} \qquad [2.37]$$

The only unknown term on the right hand side of equation 2.37 is h and this can be determined from correlations or experimental data.

Apart from the mass diffusivity of a binary mixture, D_{AB}, all the physical properties are the same as those used in the analysis of convective heat transfer and the same comments apply. The mass diffusivity can generally be found in standard databooks but, as a last resort, prediction equations such as those presented in Perry and Chilton 1974 or Bird et al. 1960 may be used.

CHAPTER 3 : FINITE ELEMENT ANALYSIS IN THE CONTEXT OF THERMAL PROBLEMS

The finite element method has its origins in the analysis of systems in structural mechanics. The method was formally introduced by Turner, Clough, Martin and Topp in 1956. Other people associated with the development of the finite element method are Argyris and Kelsey, Courant, Hrennikoff and McHenry (Dhatt and Touzot 1984). The method is well documented in books by Zienkiewicz 1977, Huebner 1975, Bathe 1982, Dhatt and Touzot 1984 and many others.

During the 1960's the method was formulated as a special case of the weighted residual method. A large variety of elements together with the isoparametric concept were introduced. The method was generalised to a method of solution for partial differential equations. A mathematical basis was established for the method using the concepts of functional analysis.

The finite element method can be described as a general technique for constructing approximate solutions to boundary value problems. It involves dividing the domain of the solution into a finite number of simple subdomains - finite elements - and using some method to construct an approximation of the solution over the collection of finite elements.

There are a number of methods which may be used to derive the finite element equations, the most widely used of which is known as the variational method and is based on classical variational principles. However, the variational principles of practical problems are not always known and transient heat conduction, the governing equation of which is parabolic, is a case in point. For this reason and the fact that it is more intuitive, the method of weighted residuals - specifically Galerkin's method - is used to construct the approximation.

In this chapter a brief description of the conversion of the classical form of the transient heat transfer equation to the fully discretised system of finite element equations is presented. Initially, the reduction of the classical form to the variational form via Galerkin's method is discussed. Once Galerkin's method has been presented the finite element method is discussed using the isoparametric approach. The concept and incorporation of infinite elements is then presented, followed by a discussion of the numerical integration of the discretised finite element equations.

3.1 VARIATIONAL STATEMENT OF THE HEAT CONDUCTION EQUATION

The classical form of the transient heat conduction equation is presented in chapter 2. In order to simplify the development of the finite element method it is assumed that the domain is isotropic. It is also assumed that the domain is two-dimensional. The classical form of the transient heat conduction equation can then be written as:

$$\rho C_p \frac{\partial T}{\partial t} = k\nabla^2 T + Q \qquad [3.1]$$

with boundary and initial conditions

$$T = T_\Gamma(x,y,t) \qquad \text{on } \Gamma_D,\ t > 0$$

$$k\frac{\partial T}{\partial n} + q = 0 \qquad \text{on } \Gamma_N,\ t > 0$$

and $T = T_o(x,y) \qquad \text{in } \Omega,\ \ t = 0$

Applying the weighted residual method to equation 3.1 involves two steps, the first of which is to assume the general functional behaviour of the temperature in some way so as to approximately satisfy the heat conduction equation and the boundary conditions. Substitution of this approximation into the original equations results in some error called the residual. This residual is required to vanish in some average sense over the entire solution domain. The second step is to solve the equation resulting from the first step.

The unknown exact solution T(x,y,t) is approximated by ϕ, where:

$$T(x,y,t) \approx \phi(x,y,t) = \Sigma_i N_i(x,y)\phi_i(t) \qquad i = 1,2,..,m \qquad [3.2]$$

where N_i = assumed functions
$\phi_i(t)$ = unknown parameters
m = number of approximating functions

The m functions N_i are generally chosen to satisfy the boundary conditions.

When ϕ is substituted into equation 3.1 the residual can be expressed as:

$$r = k\nabla^2\phi + Q - \rho C_p\frac{\partial\phi}{\partial t} \qquad [3.3]$$

The method of weighted residuals aims to determine the unknown parameters, ϕ_i, in such a way that the error, r, over the solution domain is small. This is done by forming a weighted average of the error and coercing it to vanish over the solution domain. This can be expressed in equation form as:

$$\int_\Omega\left\{k\nabla^2\phi + Q - \rho C_p\frac{\partial\phi}{\partial t}\right\}W_i d\Omega = \int_\Omega rW_i d\Omega \qquad i = 1,2,..,m \qquad [3.4]$$

where W_i = one of m linearly independent weighting functions

Galerkin's method involves selecting the weighting functions to be the same as the approximating functions used to represent ϕ, that is $W_i = N_i$ for i = 1,2...m. Thus Galerkin's method requires that:

$$\int_\Omega (k\nabla^2\phi)N_i d\Omega = \int_\Omega\left\{\rho C_p\frac{\partial\phi}{\partial t} - Q\right\}N_i d\Omega \qquad i = 1,2,..,m \qquad [3.5]$$

In order to decrease the order of the derivatives in the Laplacian and to reduce equation 3.5 to a more useful form, Green's Theorem is applied to the left hand side of equation 3.5 resulting in:

$$\int_\Omega k\nabla\phi\cdot\nabla N_i d\Omega - \oint_\Gamma k\nabla\phi N_i d\Gamma = \int_\Omega\left\{Q - \rho C_p\frac{\partial\phi}{\partial t}\right\}N_i d\Omega \qquad i = 1,m \qquad [3.6]$$

Substituting Fourier's law into equation 3.6 yields:

$$\int_\Omega k\nabla\phi\cdot\nabla N_i d\Omega + \int_{\Gamma N} qN_i d\Gamma = \int_\Omega\left\{Q - \rho C_p\frac{\partial\phi}{\partial t}\right\}N_i d\Omega \qquad i = 1,m \qquad [3.7]$$

which can be rearranged into the weak form which is used to determine the finite element equations.

$$\int_\Omega k\nabla\phi\cdot\nabla N_i d\Omega + \int_\Omega \rho C_p\frac{\partial\phi}{\partial t}N_i d\Omega = \int_\Omega QN_i d\Omega - \int_{\Gamma N} qN_i d\Gamma \qquad i = 1,m \qquad [3.8]$$

3.2 Formulation of the Finite Element Equations

To generate the finite element equations it is first necessary to discretise the solution domain. This involves dividing the body, Ω, into n subdomains or elements. It is then necessary to assign l nodes to the discretised domains in such a way that the majority of the nodes are evenly distributed on the boundaries of the subdomains.

As equation 3.8 holds for any point in Ω, it also holds for any point in Ω_i. So, equation 3.8 can be rewritten for an element Ω_j

$$\int_{\Omega j} k^e \nabla\phi^e \bullet \nabla N_i^e d\Omega_j + \int_{\Omega j} \rho^e C_p^e \frac{\partial\phi^e}{\partial t} N_i^e d\Omega_j = \int_{\Omega j} Q^e N_i^e d\Omega_j - \int_{\Gamma N j} q^e N_i^e d\Gamma_j \quad i = 1,r \qquad [3.9]$$

where the superscript 'e' restricts the range to one element.

$$\phi^e = \Sigma_i N_i^e \phi_i^e \quad i = 1,2,..,r \qquad [3.10]$$

r = number of unknown parameters assigned to the element

If equation 3.9 is summed over all the elements of the domain it becomes equivalent to equation 3.8.

In the finite element method the subdomain approximation functions, N_i, have the following condition imposed on them:

$$N_i^e(x_j, y_j) = \delta_{ij} \qquad [3.11]$$

where x_j, y_j = the coordinates of the j^{th} node of the element
δ_{ij} = Kronecker delta

This means that the unknown parameters, ϕ_i^e, are, in fact, the unknown nodal temperatures, T_i^e. Equation 3.10 may now be rewritten as:

$$\phi^e = \Sigma_i N_i^e T_i^e \quad i = 1,2,..,r \qquad [3.12]$$

Substituting equation 3.12 into equation 3.10 and rewriting it in matrix notation, yields:

$$\left[\int_{\Omega i} B^{eT}k^eB^e d\Omega_i\right]T^e + \left[\int_{\Omega i} \rho^e C_p^e N^{eT}N^e d\Omega_i\right]\frac{dT^e}{dt} = \left[\int_{\Omega i} N^{eT}Q^e d\Omega_i\right] - \left[\int_{\Gamma Ni} N^{eT}q^e d\Gamma_i\right] \qquad [3.13]$$

where N^e = vector of element approximation functions
T^e = vector of element nodal temperatures
B^e = $\nabla\phi^e$

Earlier, it was assumed that the entire domain is isotropic. Inspection of equation 3.13 indicates that this assumption may be relaxed in the sense that only the subdomains need be isotropic.

One of the consequences of the integration by parts is that the Neuman and Cauchy boundary conditions are accounted for in the second term on the right hand side of equation 3.13. Two boundary conditions are considered, a prescribed flux and a convection condition:

Prescribed flux : $q^e = q_p^e(t)$ [3.14]

Convection : $q^e = h(T - T_f)$
$= h(N^e \cdot T^e - T_f)$ [3.15]

where h = heat transfer coefficient (W/m^2°C)
T_f = fluid temperature (°C)

Other boundary conditions exist - radiation and mass transfer, which are discussed in chapter 2 - but these are reduced to prescribed flux or convection boundary condition equivalents. The prescribed temperature boundary condition is taken into account during the solution of the finite element equations. Another condition which must be considered and which has been ignored so far, is point energy sources, P^e. These are similar to internal heat generation and have been lumped with that term in equation 3.13 during the preceding development. As point energy sources are associated with nodes rather than areas or volumes, they are accounted for when the finite element equations are solved.

Substitution of equations 3.14 and 3.15 into equation 3.13 yields:

$$\left[\int_{\Omega i} B^{eT}k^eB^e d\Omega_i + \int_{\Gamma Ni} N^{eT}h^eN^e d\Gamma_i\right]T^e + \left[\int_{\Omega i} \rho^e C_p^e N^{eT}N^e d\Omega_i\right]\frac{dT^e}{dt}$$

$$= \left[\int_{\Omega i} N^{eT} Q^e d\Omega_i + P^e\right] - \left[\int_{\Gamma Ni} N^{eT} q^e d\Gamma_i - \int_{\Gamma Ni} N^{eT} h^e T_f d\Gamma_i\right] \qquad [3.16]$$

Equation 3.16 may be expressed in more concise form as:

$$[K_c^e + K_h^e]T^e + [C^e]\frac{dT^e}{dt} = f_g^e + f_s^e + f_c^e + f_p^e \qquad [3.17]$$

where

K_c^e	=	$\int_{\Omega i} B^{eT} k^e B^e d\Omega_i$	element conductivity stiffness matrix
K_h^e	=	$\int_{\Gamma Ni} N^{eT} h^e N^e d\Gamma_i$	element convection stiffness matrix
C^e	=	$\int_{\Omega i} \rho^e C_p^e N^{eT} N^e d\Omega_i$	element heat capacity matrix
f_g^e	=	$\int_{\Omega i} N^{eT} Q^e d\Omega_i$	element heat generation load vector
f_s	=	P^e	element point energy source vector
f_p^e	=	$-\int_{\Gamma Ni} N^{eT} q^e d\Gamma_i$	element prescribed flux load vector
f_c^e	=	$\int_{\Gamma Ni} N^{eT} h^e T_f d\Gamma_i$	element convection load vector

Once the element stiffness matrices and load vectors have been summed and assembled, the system of equations may be concisely denoted by

$$[K]T + [C]\frac{dT}{dt} = f \qquad [3.18]$$

where

K	=	assembled thermal stiffness matrix
C	=	assembled heat capacity matrix
T	=	nodal temperature vector
f	=	assembled load vector

Equation 3.18 represents a system of first order differential equations

which is linear if none of the parameters in the above equations are functions of temperature.

3.3 Isoparametric approach

So far, little has been said about the approximation functions, N_i^e. The condition represented by equation 3.11 has been imposed and it has been specified that a finite number, r, of these functions is associated with each element. However, these specifications do little to make the choice of the functions simple or systematic. One advantage of the finite element method is that it provides a general and systematic technique for constructing the approximation functions. The basic idea is that the approximation functions N_i (that is, the domain functions) can be defined piecewise over the subdomains in such a way that they are chosen to be very simple functions, such as low order polynomials. These simple, piecewise approximation functions are known as shape functions. These functions must be sufficiently smooth to be both continuous at element interfaces and for the purposes of this study, they must be once differentiable.

The family of Lagrangian polynomials conforms to these requirements and is represented by:

$$L_i(x) = \frac{(x-x_1)..(x-x_{i-1})(x-x_{i+1})..(x-x_k)}{(x_i-x_1)(x_i-x_2)..(x_i-x_{i-1})(x_i-x_{i+1})..(x_i-x_k)} \qquad [3.19]$$

where $L_i(x)$ = Lagrangian polynomial corresponding to x_i
k = the number of points minus 1

To find the shape functions of a two-dimensional element, the one-dimensional polynomials in x and y are multiplied together. This allows the order of the polynomial to be different in each direction.

In the preceding discussion it has been stated that the finite element equations must be constructed over each element and then assembled for solution over the whole domain. This procedure involves evaluating shape functions for every element of the mesh, which is an arduous process. The amount of effort required may be substantially reduced by defining a master element, Ω_m, of simple geometry, in another coordinate system and mapping it onto each element in the domain. A further simplification can be made by using the same shape functions which approximate the temperature variation over the master element, as the mapping functions. The mapping functions may then be written as:

$$x = \Sigma_i \hat{N}_i(\xi,\eta)x_i \quad i = 1,2,..,r \qquad [3.20a]$$

$$y = \Sigma_i \hat{N}_i(\xi,\eta)y_i \quad i = 1,2,..,r \qquad [3.20b]$$

where x,y = coordinates in the domain coordinate system
ξ,η = coordinates in the master coordinate system
$\hat{N}_i(\xi,\eta)$ = shape function of node i of the master element system
x_i,y_i = coordinates of node i in the domain coordinate system
r = number of nodes in the element.

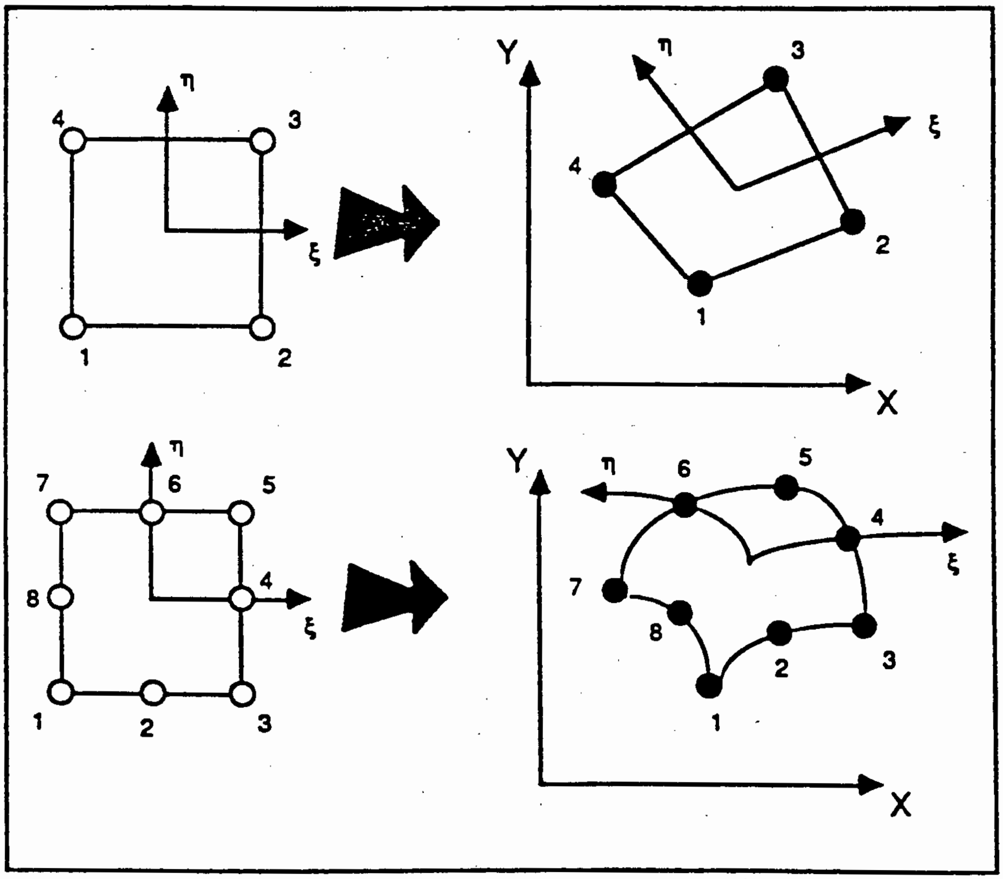


Figure 3.1 *Mapping of the master element onto the problem domain.*

The mathematical expression of this mapping involves the use of the Jacobean matrix, J, and its determinant. It is used in equation 3.16 to express B in terms of the derivatives of the master element shape function and to express the element volume or surface over which the integration has to be carried out, in terms of the master element coordinate system. It is also necessary to change the limits of integration.

Equation 3.16 can now be rewritten to reflect these changes.

$$\left[\iint_{\Omega m} B^{eT}k^eB^e|J|d\Omega_m + \int_{\Gamma m} \hat{N}^{eT}h^e\hat{N}^e|J|d\Gamma_m\right]T^e + \left[\iint_{\Omega m} \rho^e C_p^e\hat{N}^{eT}\hat{N}^e|J|d\Omega_m\right]\frac{dT^e}{dt}$$

$$= \left[\iint_{\Omega m} \hat{N}^{eT}Q^e|J|d\Omega_m + P^e\right] - \left[\iint_{\Gamma m} \hat{N}^{eT}q^e|J|d\Gamma_m - \int_{\Gamma m} \hat{N}^{eT}h^eT_f|J|d\Gamma_m\right] \qquad [3.21]$$

3.4 Infinite Elements

Many continua have boundaries which are well defined by their geometry. There are some cases, however, in which the boundary is defined so far from the regiȯn of interest in comparison with its size, that the boundary is effectively at infinity. To deal with these unbounded problems a number of approximate methods have been developed or adapted by researchers. Some of these approximations are briefly discussed below.

The simplest and most widely used method of approximating an infinite boundary is to truncate a mesh at some point judged to be far enough from the area of interest so that the truncation will not adversely affect the solution generated. The weakness of this method lies in the need to judge where to truncate the mesh. This method is apparently unreliable and inaccurate especially in the case of transient problems (Zienkiewicz et al. 1981).

Other more sophisticated methods have been developed to model far field problems. Some of these are:

1) Matching to analytical solutions - matching finite element solutions to analytical or semi-analytical solutions.

2) Mappings - attempting to map an exterior domain onto an interior domain using the principle that the exterior of any polygon can be

mapped onto the exterior of any circle and then onto the interior of the circle using conformal mapping.

3) Boundary integrals - the coupling of boundary integral models to finite element models. There has been little or no work on transient exterior problems.

4) Infinite elements - these are used to approximate a boundary at infinity by designing elements with decay functions (reciprocal or exponential type) or mapping the element extending to infinity in global coordinates onto a standard master element in local coordinates.

Of these the mapped infinite elements proposed by Zienkiewicz 1981 are regarded as the simplest and the most elegant by many researchers (Bettess and Bettess 1984, Marques and Owen 1984). The mathematical details of this technique are presented below.

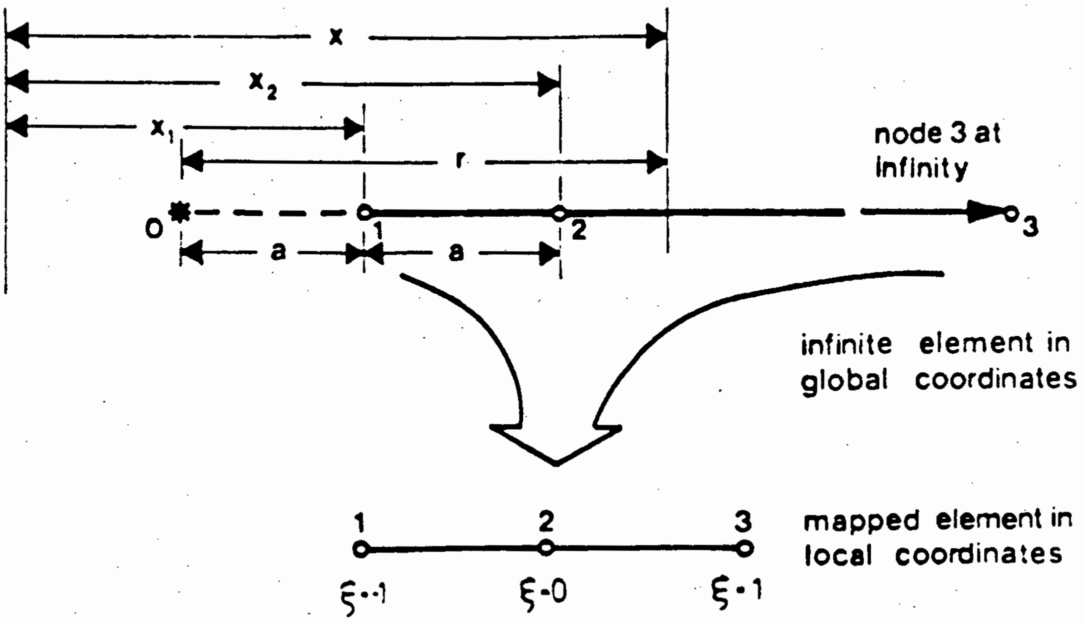

x
x_2
x_1
r
node 3 at infinity
O
1
2
3
a
a
infinite element in global coordinates
1
2
3
mapped element in local coordinates
$\xi=-1$
$\xi=0$
$\xi=1$

Figure *3.2 Mapping of an Infinite Element to a Master Element.*

In order to examine the mathematical development of this method it is best to consider, initially, a simple example - a one-dimensional element. Figure 3.2 presents an illustration of the procedure about to be described.

Consider a one-dimensional element extending from a point x_1, through x_2, to the point at infinity, x_3. This element can be mapped to the standard one-dimensional master element using the following rule:

$$x = N_0(\xi)x_0 + N_2(\xi)x_2 \qquad [3.22]$$

where $N_0(\xi) = \dfrac{\xi}{1 - \xi}$ $\qquad N_2(\xi) = 1 + \dfrac{\xi}{1 - \xi}$

Damjanic and Owen 1984 present the following mapping:

$$x = N_1(\xi)x_1 + N_2(\xi)x_2 \qquad [3.23]$$

where $N_1(\xi) = \dfrac{-2\xi}{1 - \xi}$ and $N_2(\xi) = 1 + \dfrac{2\xi}{1 - \xi}$

It can be shown that these mappings are equivalent.

From the mapping equation (3.22) it can be seen that:

$$\begin{array}{llll} \text{when} & \xi = +1 \quad , & x = x_2 - N_0(\xi)(x_2 - x_0) & = \infty \\ & \xi = \ \ 0 \quad , & x = x_2 & \\ & \xi = -1 \quad , & x = (x_2 + x_0)/2 & \end{array}$$

As it is desired that $\xi = -1$ corresponds to the point x_1, x_1 is defined to be:

$$x_1 = (x_2 + x_0)/2 \qquad [3.24]$$

which enables the mapping to be rewritten as:

$$x = (2x_1 - x_2)N_0(\xi) + x_2N_2(\xi) \qquad [3.25]$$

This expression, when expanded and rearranged, can be shown to be equivalent to the Damjanic and Owen 1984 mapping equation (3.23).

In a finite element analysis of a field problem, the field variable, ψ say, is modelled over an element using an equation formed by the summation of a number of shape functions. This equation might have the expanded form presented below:

$$\psi(\xi) = \alpha_0 + \alpha_1\xi + \alpha\xi^2 + \ldots\ldots \qquad [3.26]$$

It would be of interest to see the form of this equation in global coordinates. The inverse mapping is

$$\xi = 1 - 2(x_2 - x_1)/(x - x_0) = 1 - 2a/r \qquad [3.27]$$

This leads to the equation in global coordinates

$$\psi(x) = \gamma_0 + \gamma_1/r + \gamma_2/r^2 + \ldots\ldots \qquad [3.28]$$

where $\gamma_0 = 0$ if ψ tends to zero at infinity. It can be seen that x_0 is the 'pole' of the expansion. The significance of the pole will be discussed later. Damjanic and Owen 1984 point out that suitable decay rates, 1/rn, of the field variable can be arranged by appropriately choosing the order of the shape functions used.

The mathematical development presented above is now extended to include two-dimensional infinite element mapping functions.

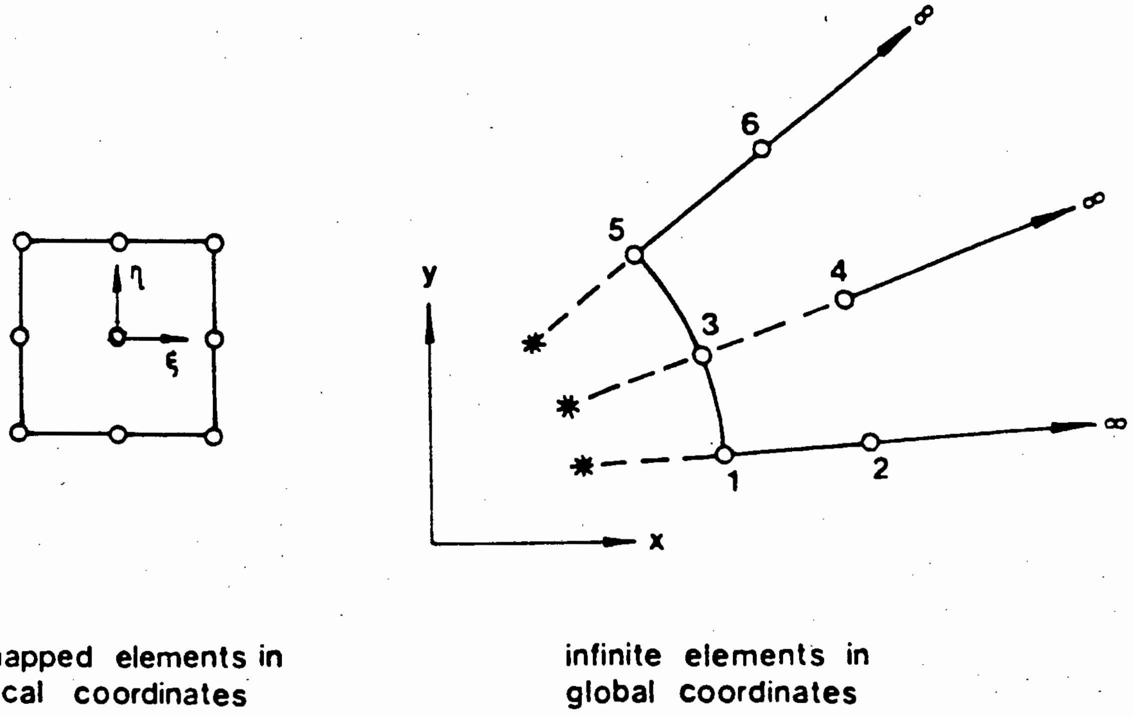


Figure *3.3 A two-dimensional mapped infinite element.*

Consider the two-dimensional element shown in figure 3.3. It can be seen that this element is finite in the H direction but infinite in the U direction. This leads to a group of mapping functions that are a product of the standard, one-dimensional shape functions in the finite direction and the modified one-dimensional mapping functions in the infinite direction. That is

$$\begin{aligned} x = {} & \hat{N}_1(\eta)[(2x_1 - x_2)N_0(\xi) + x_2N_2(\xi)] \\ & + \hat{N}_2(\eta)[(2x_3 - x_4)N_0(\xi) + x_4N_2(\xi)] \\ & + \hat{N}_3(\eta)[(2x_5 - x_6)N_0(\xi) + x_6N_2(\xi)] \end{aligned} \qquad [3.29a]$$

and,

$$\begin{aligned} y = {} & \hat{N}_1(\eta)[(2y_1 - y_2)N_0(\xi) + y_2N_2(\xi)] \\ & + \hat{N}_2(\eta)[(2y_3 - y_4)N_0(\xi) + y_4N_2(\xi)] \\ & + \hat{N}_3(\eta)[(2y_5 - y_6)N_0(\xi) + y_6N_2(\xi)] \end{aligned} \qquad [3.29b]$$

where the $\hat{N}_i(\xi)$ are standard isoparametric, one-dimensional Lagrangian shape functions.

In the case of an isoparametric finite element, the Jacobian matrix is determined from the shape functions because, by definition, they are also the mapping functions. As an infinite element is not isoparametric, equation 3.29 must be used to determine the Jacobian matrix for the element shown in figure 3.3.

To describe the unknown function over the element, standard shape functions can be used. The order of these shape functions must match the number of nodes present in the finite H direction to ensure continuity. In the infinite U direction, the order of the shape functions used is immaterial. Unless quadratic shape functions are used, the node positions will not coincide with those used in the mapping.

In order to use standard summation notation, it is simpler to use the mapping functions presented by Damjanic and Owen 1984. This leads to the following mappings:

$$x(\xi,\eta) = \Sigma_i M_i(\xi,\eta)x_i \qquad [3.30a]$$

$$y(\xi,\eta) = \Sigma_i M_i(\xi,\eta)y_i \qquad [3.30b]$$

in which the summation extends over the number of finite nodes only. The number of finite nodes depends on the number of nodes per element and the number of directions in which the element extends to infinity.

The infinite element shape functions for use in the above mappings are (with reference to figure 3.3):

$$M_1(\xi,\eta) = N_1(\xi)\hat{N}_1(\eta) \qquad M_2(\xi,\eta) = N_2(\xi)\hat{N}_1(\eta)$$
$$M_3(\xi,\eta) = N_1(\xi)\hat{N}_2(\eta) \qquad M_4(\xi,\eta) = N_2(\xi)\hat{N}_2(\eta)$$
$$M_5(\xi,\eta) = N_1(\xi)\hat{N}_3(\eta) \qquad M_6(\xi,\eta) = N_2(\xi)\hat{N}_3(\eta)$$
$$\Sigma_i M_i(\xi,\eta) = 1 \qquad [3.31]$$

Where $\hat{N}_i(\eta)$ = the shape functions associated with a one-dimensional, quadratic isoparametric element

$N_i(\xi)$ = the Damjanic and Owen 1984 infinite element mapping functions.

The infinite element shown in figure 3.3 extends to infinity in one direction only. It is possible for a two-dimensional infinite element to extend to infinity in two direction as well. The former is termed a singly infinite element and the latter a doubly infinite element. The mapping functions of a doubly infinite element are determined by multiplying the one-dimensional infinite element mapping functions in each direction.

A list of the mapping functions for various singly and doubly infinite elements, as tabulated by Marques and Owen 1984, is presented in the appendix.

3.5 Numerical Integration

As the power of the finite element method can only be exploited using a computer, some method of numerical integration is required to use equation 3.21.

The forms of the integrals which are relevant are:

$$I_1 = \int_{-1}^{+1} g(\xi,\eta)d\xi \qquad [3.32a]$$

$$I_2 = \int^{+1}\int^{+1} f(\xi,\eta)d\xi d\eta \qquad [3.32b]$$

$$\int_{-1}\int_{-1}$$

The values of the limits of the integrals and the degree of accuracy of the method make Gauss-Legendre quadrature a good choice. The method is used to approximate integrals of the form of equation 3.22 in the following way:

$$I_1 \approx \Sigma_i[w_i g(\xi_i,\eta_i] \qquad i = 1,2,..,s \qquad [3.33a]$$

$$I_2 \approx \Sigma_i\Sigma_j[w_i w_j f(\xi_i,\eta_j)] \quad i = 1,2,..,s;\ j = 1,2,..,t \qquad [3.33b]$$

where s,t = number of evaluation points
w_i = weighting factor associated with (ξ_i,η_i), the position of the evaluation point

The number of evaluation points is variable and should be selected based on the fact that an s-point Gauss-Legendre integration generates the exact solution to integrals of polynomials of order up to 2s-1. In equation 3.23 it is usual, but not necessary, to select s = t. The evaluation points and corresponding weighting values for up to 3-point Gauss-Legendre integration, are shown in table 3.1:

Table *3.1: Abscissae and weight coefficients of the Gauss-Legendre quadratic formula.*

Order	Weight w_i	Position ξ_i
1	2.00000 00000 00000	0.00000 00000 00000
2	1.00000 00000 00000	±0.57735 02691 89626
3	0.55555 55555 55555 0.88888 88888 88888	±0.77459 66692 41483 0.00000 00000 00000

Replacing all integrals in equation 3.16 by the quadrature formula, results in:

$$[K] = \Big[\Sigma_i\Sigma_j w_i w_j B^T(\xi_i,\eta_j) k B(\xi_i,\eta_j)\,|J(\xi_i,\eta_j)| \; + \; \Sigma_i\Sigma_j w_j \hat{N}^T(\xi_j,1) h \hat{N}(\xi_j,1)\,|J(\xi_j,1)|\Big] \qquad [3.34]$$

$$[C] = \Big[\Sigma_i\Sigma_j w_i w_j \rho C_p \hat{N}^T(\xi_i,\eta_j)\hat{N}(\xi_i,\eta_j)\,|J(\xi_i,\eta_j)|\Big] \qquad [3.35]$$

$$f = \Big[\Sigma_i\Sigma_j w_i w_j \hat{N}^T(\xi_i,\eta_j) Q\,|J(\xi_i,\eta_j)| \; + \; P^e \; +$$

$$\Sigma_1\Sigma_j w_j\hat{N}^T(\xi_j,1)hT_f|J(\xi_j,1)| \; -$$

$$\Sigma_1\Sigma_j w_j\hat{N}^T(\xi_j,1)q_p|J(\xi_j,1)|\Big] \qquad [3.36]$$

$$i,j = 1,2,..,s;\; 1 = 1,2,..,\text{number of sides}$$

The finite element equations represented by equation 3.24 to 3.26 constitute a system of equations in which the nodal temperatures, T, and the temperature rates, dT/dt, are unknown. Chapter 4 presents a discussion on how these equations are solved.

CHAPTER 4 : SOLUTION OF THE FINITE ELEMENT EQUATIONS.

In chapter 3 the finite element equations of the transient heat conduction equation were developed. In matrix form, they can be represented as:

$$[K]T + [C]\frac{dT}{dt} = f \qquad [4.1]$$

with initial condition $T(t_o) = T_o$

where			
	[K]	=	assembled thermal stiffness matrix
	[C]	=	assembled thermal heat capacity matrix
	f	=	assembled thermal load vector
	T	=	vector of unknown nodal temperatures

If [K], [C], or {f} are dependent on T then the system of equations is nonlinear. As material properties such as thermal conductivity and heat capacity are generally functions of temperature, it is sensible to assume that equation 4.1 is always nonlinear for the purposes of finding an appropriate solution method.

In this chapter, a brief discussion of various methods of solving systems of nonlinear, first order ordinary differential equations is presented. The selected method is then described followed by some comments about its stability and the accuracy of the generated solution.

4.1 METHODS OF SOLVING NONLINEAR FIRST ORDER DIFFERENTIAL EQUATIONS

A large number of methods exist for the solution of equation 4.1 and they can be grouped into two categories:

- Direct integration methods
- Mode superposition methods.

Direct integration methods consist of a sequential construction of values of the solution from its initial value T_o, at times $t_o+\Delta t$, $t_o+2\Delta t, \ldots,$ $t_o+n\Delta t$, where Δt is a specified time-step size. Many direct integration methods can be derived from finite difference approximations of dT/dt. Application of these methods generally results in equation 4.1 becoming fully discretised, leading to a modified equation.

$$[K^*]T = f^* , \qquad [4.2]$$

where $[K^*]$ = effective stiffness matrix
f^* = effective load vector
T = unknown nodal temperature vector.

Mode superposition methods involve decoupling the finite element equations, as represented by equation 4.1, into modal equations. Each modal equation can then be solved independently using some numerical integration method, and the final results obtained by linear combination of the various modes. The major disadvantage of mode superposition is that the technique cannot be applied to nonlinear problems. Hence, some direct integration method must be selected.

In order to take into account possible nonlinearities, a solution technique must be selected to be used in conjunction with the integration method. Most methods for solving nonlinear equations are iterative in nature and are generally derived from the following three schemes, singly or in combination.

- Substitution method
- Newton-Raphson method
- Incremental method

The substitution method uses the most recently evaluated nodal temperatures to evaluate the various constituents of the finite element equations, which are then assembled and solved, generating a new solution. The temperature used to evaluate the various parameters and properties is sometimes modified using equation 4.3:

$$T^{i*} = T^{i-1} + \omega(T^i - T^{i-1}) \qquad [4.3]$$

where T^{i*} = temperature used to evaluate [K], [C] and f
T^i = solution based on T^{i-1}
T^{i-1} = solution at previous iteration
i = iteration number
ω = over-relaxation factor.

The over-relaxation factor frequently accelerates the convergence rate, but its value is problem specific and must be determined experimentally.

The Newton-Raphson method involves applying the same technique to the assembled finite element equations as that which is applied to scalar functions of a single variable by Newton's Method. This means that both the stiffness and heat capacity matrices, as well as the load vector, must be differentiated with respect to temperature at each iteration.

The incremental method involves selecting a number of load vector multipliers in the range (0,1] which are used sequentially to reduce the magnitude of the load vector of the system of equations. The equations - which are still nonlinear - are then solved with the reduced load vector by the Newton-Raphson method. This process is repeated until the system is solved using the full value of the load vector.

For various reasons which are elucidated in his thesis, Korvink 1986 chose the following iterative time-integration scheme in which 'i' represents the iteration number and 'n' the time-step number.

Equation 4.1 can be written in the following form to reflect the iterative time-integration scheme:

$$[K(T_n^i)]T_n^{i+1} + [C(T_n^i)]\frac{dT_n^{i+1}}{dt} = f(T_n^i) \qquad [4.4]$$

Korvink 1986 approximated the time derivative using equation 4.5.

$$\frac{dT_n^{i+1}}{dt} = \frac{1}{\alpha \Delta t_n}(T_n^{i+1} - T_{n-1}) - \frac{(1-\alpha)}{\alpha}\frac{dT_{n-1}}{dt} \qquad [4.5]$$

where α = a scalar multiplier in the range (0,1].
Δt_n = time-step size at time-step n.

Substitution of equation 4.5 into 4.4 yields

$$\left[K(T_n^i) + \frac{1}{\alpha \Delta t_n} C(T_n^i)\right]T_n^{i+1} = \left[f(T_n^i) + \frac{1}{\alpha \Delta t_n}[C(T_n^i)]T_{n-1} + \frac{(1-\alpha)}{\alpha}\frac{dT_{n-1}}{dt}\right] \qquad [4.6]$$

which may be condensed to

$$[K^{*}]^{i}_{n}T^{i+1}_{n} = f^{*i}_{n} \qquad [4.7]$$

Equation 4.7 is used to iterate until the following convergence requirement is satisfied:

$$|T^{i+1}_{n} - T^{i}_{n}| < \text{convergence tolerance} \qquad [4.8]$$

So far, the assembly of the element equations and the solution of the resulting system of linear equations has been mentioned with no details of the techniques involved. Due to the often huge dimensions of the system of equations which must be solved, special techniques of solution have been developed which take advantage of the sparse and often symmetrical nature of the finite element equations.

The element equations are assembled bearing in mind that contiguous elements share boundary nodes. For instance, a corner node would receive contributions from four elements, a mid-side node from two elements and an internal node from one element - assuming that these nodes are not on the domain boundary. In order to keep track of the influence of elements on nodes, it is necessary to relate the local element nodal numbering system to the actual numbering system for each element.

Two general methods of solution have been developed:

- Direct solution in which an exact solution is sought. Examples of methods which implement this technique are the Gauss elimination procedure, the sparse matrix method and the front solution method.
- Iteration in which a successive approximation technique is used to converge on the true solution. Gauss-Siedel iteration is an example of this class of method.

4.2 Accuracy and stability of finite element solutions

The quality of the finite element solution to the transient, nonlinear heat conduction equation can be analysed in terms of two aspects of the solution procedure:

- Stability
- Accuracy

The characteristics of the finite element method which cause stability problems are the discretisation of the domain in space and time. It is possible to analyse the stability characteristics of most transient solution techniques using spectral analysis. A number of researchers, Hughes 1977, for instance, have carried out such analysis on the transient heat conduction equation. Using spectral analysis, it is possible to prove, for instance, that the semi-implicit Euler time-integration scheme is unconditionally stable when $\alpha \geq 0.5$. Introduction of nonlinearity, however, prevents the derivation of such general results. Most methods require the time-step size to be less than a certain critical value to guarantee stability and, additionally, less than a smaller critical value to avoid oscillation. The stability of a solution does not, however, imply accuracy. To achieve an accurate result, the time-step size must be chosen to reduce the truncation error to an acceptable value.

Instability due to discretisation in space is caused by the use of too coarse a mesh. The instability generally occurs in the initial stages of an analysis when the imposition of boundary conditions causes high temperature gradients at the domain boundary. If the time-step size is small the temperature is changed from its starting value only in a narrow region close to the boundary and if the elements on the boundary are wider than this region, instability is likely to occur.

There are two ways of eliminating spatial instability:

1) increasing the time-step size and
2) increasing the mesh density in the vicinity of the boundary.

Increasing the time-step size results in a decrease in the temperature gradient near the boundary and effectively widens the area in which the initial temperature is changing. However, the reduction in spatial instability could well be effected at the cost of a corresponding increase in the instability of the time-integration scheme used.

It is clear that reducing the size of the elements near the boundary would improve the quality of the solution in that region. The problem is deciding just how small to make them, bearing in mind the computational expense of arbitrarily adding a number of new nodes to the system.

It is possible to evaluate minimum element widths using penetration depth analysis. This method is based on the fact that a short time after the boundary conditions have been imposed, there is a region, a short distance from the boundary, where the initial conditions have not changed. This distance is referred to as the penetration depth, δ. If a linear or quadratic temperature gradient is assumed over the region in which the initial temperature has changed, it is possible to evaluate the penetration depth as a function of time. This procedure is shown below for one-dimensional heat conduction in a semi-infinite, isotropic body.

The appropriate governing equations and conditions are:

$$\begin{aligned}
&\alpha\frac{\partial^2 T}{\partial^2 x} = \frac{\partial T}{\partial t} \\
&T(0,t) = T_1 \\
&T(x,0) = T_o \\
&T(x,t) = T_o \qquad \forall\, x \in [\delta,\infty)
\end{aligned} \qquad [4.9]$$

Equation 4.9 can be reduced to equation 4.10 by treating it as a first order differentiation equation in x and making judicious use of the boundary and initial conditions in eliminating terms.

$$-\alpha\frac{\partial T}{\partial x}\bigg|_{x=0} = \frac{d}{dt}\left[\int_0^\delta T dx - T_o\delta\right] \qquad [4.10]$$

To proceed, it is necessary to approximate the actual temperature profile with some expression. As the finite element shape functions are polynomials, it is reasonable to approximate the temperature using a polynomial of the same order of the element shape functions. As bilinear and biquadratic elements are used in this study, one of the following two approximations is used, depending on the nature of the boundary elements.

$$T = C_0 + C_1x \qquad \forall\, x \in [0,\delta] \qquad [4.11a]$$

$$T = C_0 + C_1x + C_2x^2 \qquad \forall\, x \in [0,\delta] \qquad [4.11b]$$

where C_i = constants evaluated using the boundary conditions

In this development, the quadratic approximation is used, and, having evaluated the constants, equation 4.11 has the form:

$$T = T_1 - 2(T_1-T_o)x/\delta + (T_1-T_o)(x/\delta)^2 \qquad [4.12]$$

Using equation 4.12, the derivative and integral terms in equation 4.10 can be evaluated, yielding the following differential equation:

$$\frac{d\delta}{dt} = 6\alpha/\delta \qquad [4.13]$$

which may be solved to yield:

$$\delta(\Delta t) = (12\alpha\Delta t)^{0.5} \qquad [4.14]$$

Equation 4.14 was derived by assuming a fixed temperature at the infinite end of the body. If a fixed flux condition is imposed on the infinite boundary, equation 4.15 can be derived using the same method.

$$\delta(\Delta t) = (6\alpha\Delta t)^{0.5} \qquad [4.15]$$

Similar equations can be derived using a linear temperature profile approximation.

Equations 4.14 and 4.15 can be used to evaluate the minimum widths of elements on the boundary. Depending on the problem being analysed, it may be advisable to, firstly, set Δt equal to the sum of the first two time-steps and, secondly, place two or three elements in the penetration zone.

Equations 4.14 and 4.15 indicate that the smaller the initial time-step size, the smaller is the penetration depth. This means that a reduction in time-step size might have to be accompanied by a corresponding increase in mesh density in the vicinity of the boundary in order to maintain stability.

CHAPTER 5 : IMPLEMENTATION OF THE FINITE ELEMENT METHOD

This chapter describes some aspects of the manner in which the preceding heat transfer theory and finite element method descriptions are combined to create the finite element method heat conduction analysis computer program. Some background information relating to the history of the program is given, followed by an overview of the structure of the program. Next, some aspects of the data expected by the program are discussed, followed by a summary of all the material properties used by the program and a description of how the temperature dependence and spatial variation of the various properties are approximated by the program. The implementation of the three more complex boundary conditions - convection, radiation, and mass transfer - is then discussed, followed by a description of the manner in which infinite elements are incorporated. Finally, a summary and brief discussion of the functions of the more important program modules is presented.

For convenience, the program is called NOTHCAN which is a quasi-acronym derived from the following program description: NOnlinear, Transient Heat Conduction ANalysis.

5.1 BACKGROUND INFORMATION

The current version of NOTHCAN is based on the work of, firstly, the various researchers who have contributed to the structural analysis program NOSTRUM (Duffet et al. 1983), secondly, Pennington 1985, who added a steady state nonlinear heat transfer and thermo-mechanical stress analysis option to NOSTRUM, thirdly, Korvink 1986, who extended Pennington's 1985 work to include the analysis of transient problems and, fourthly, the writer, who has made the extensions and changes to the work of Korvink 1986 in accordance with the requirements discussed in chapter 1.

At present, the program differs substantially from Korvink's 1986 program. Firstly, it is no longer part of NOSTRUM. All the heat transfer and finite element analysis related procedures of NOSTRUM were extracted and where necessary, stripped of superfluous variables and code. Secondly, the program no longer includes thermo-mechanical analysis options. These modifications have resulted in a program - NOTHCAN - which, while significantly different from the parent NOSTRUM, retains many similarities in terms of code, variable names, procedure names and general algorithmic form.

Apart from the changes mentioned above, the following additions have been made to the program:

1) Point energy source boundary condition
2) Internal heat generation option
3) Reaction flux option
4) Convection boundary condition option
5) Radiation boundary condition option
6) Mass transfer boundary condition option
7) Infinite element option

The first three items in the above list are elementary and therefore do not require discussion. The next four do require discussion and each has a section devoted to it below. Two boundary condition options were already included in the program - prescribed temperature and distributed flux boundary conditions - and the details of these conditions can be found in Korvink's 1986 thesis.

Currently, NOTHCAN is installed on a VAX/VMS 6230 computer. Originally, NOSTRUM was developed on a Sperry Univac 1100 computer. The writer had to port the program from Sperry to VAX which involved converting all Sperry specific code to an alternative form which could work on the VAX. As far as possible the writer has tried to stay within the limits of standard Fortran 77. Thus, having undergone one machine-to-machine conversion, already, the NOTHCAN code is, in the opinion of the writer, reasonably portable.

In the past, analysing real problems using computer-oriented discrete numerical methods has been done on expensive mainframes or mini-computers for reasons of superior processing speed and memory availability. However, with the advent of recent micro-computer hardware and software technology which has both allowed the memory limitation barrier to be broken and enabled processing speeds to approach those of micro-computers, it is now possible to run large programs with high processing speed requirements on microcomputers. Floating point arithmetic on microcomputers has also reached comparable and even equivalent levels of accuracy. Thus a program such as NOTHCAN might successfully be converted to run on a top-range micro-computer, resulting in a very significant reduction in hardware costs. Additionally, in the past, most numerical method oriented programs have been written in the now outdated Fortran language. More modern computer languages such as Pascal and C (as implemented on micro-computers) are, in the writer's opinion, preferable to Fortran for finite element method requirements due to their sophisticated and powerful data structures, strongly structured code, dynamic memory allocation features and powerful graphics interfaces.

5.2 General Program Structure

The general structure of NOTHCAN is directly derived from that of NOSTRUM and can be divided into three fundamental model data manipulation operations which are sequentially initiated by a 'driver' procedure. These three operations are pre-processing, processing, and post-processing. Currently, pre-processing involves reading the data file in which the data which defines and describes the physical problem to be analysed, is stored. While this information is being read, it is checked for errors. If errors are found, the analysis is terminated with a message indicating the nature of the error. Errors which are trapped are data type mismatches (a Fortran runtime error), violation of data magnitude limitations and attempts to exceed bounds previously defined in the data file. Currently, the error trapping mechanism of NOTHCAN is neither exhaustive nor sufficiently sophisticated.

Processing involves performing the preliminary calculations necessary to set up mass transfer and radiation boundary conditions, carrying out the finite element method calculations, evaluating convergence for nonlinear problems, performing various calculations related to mass transfer, performing various calculations related to mass transfer and radiation boundary conditions, output of results and updating the analysis time.

Post-processing involves interpretation and analysis of the results generated during the processing operation. Currently NOTHCAN only writes the results of the analysis for every node in the mesh for each iteration of every time-step to a file. As a simple aid to post-processing it is possible to have converged results at certain specified nodes written to a separate file.

From the above description of the current status of the three operations in the case of NOTHCAN, it is clear that both pre- and post-processing facilities could be substantially improved. Possible improvements to pre-processing are discussed first.

A good pre-processing facility should offer the user an interactive, semi-intelligent, graphics assisted data capture facility together with a strong error trapping facility. The pre-processor should be interactive in the sense that the user should be prompted for data in such a way that the user can alter previously specified information. It should be semi-intelligent in that it would prevent the user entering nonsensical information. It should be graphics assisted in that a visual representation of the mesh is useful, together with options of zooming in on selected areas of the mesh. It would be useful for the graphics facility to be linked to some pointing device both to draw the mesh initially and to delimit regions for various subsequent operations.

A good post-processing facility should have the same attributes as the pre-processor discussed above. The user should be able to specify a data file containing results and then instruct the post-processor to display or print various types of graphical interpretations of selected results. Two examples of desirable graphical display would be firstly, an image of the mesh overlayed with temperature or flux contours and secondly, a graph of temperature or flux plotted as a function of time for selected nodes.

It seems possible to fulfill the above-mentioned requirements for pre- and post-processor utilities in a micro-computer environment. Both processors could be implemented on a micro-computer in such a way that the pre-processor would be used to create the data file containing information which describes the problem to be modelled. This data file could then be transferred to a mainframe where the processing operation would be performed. The results of the analysis could then be transferred back to the micro-computer environment in which they could be analysed and examined using the post-processor.

5.3 Data input

A problem to be analysed - be it a text-book problem statement or a real situation - is generally not immediately analysable by the program. The original problem statement must be re-posed in the context of the method of analysis which, in the case of NOTHCAN, results in a data file containing numerical information of two categories. The first category is comprised of data which describes and defines the problem to be modelled and the second category includes data which controls how the program analyses the problem.

A typical problem data file would include:

- Problem and Data file description
- Analysis and Mesh Control parameters
- Transient and nonlinear analysis control parameters
- Output control parameters
- Nodal coordinate specification
- Element topology specification
- Initial conditions specification
- Boundary condition specification
- Time function specification.

More information about the data input requirements for the use and operation of NOTHCAN can be found in the NOTHCAN user's guide.

5.4 Physical properties and parameters

The various physical properties used by NOTHCAN for the analysis of conduction in solids with various boundary conditions are listed below:

SOLIDS

Thermal conductivity
Heat capacity
Density

FLUIDS

Density
Diffusivity
Heat capacity
Volume expansivity
Thermal conductivity
Viscosity

Of the solid physical properties, only density is not a function of temperature. The reason for this is that the density variation of solids with temperature is negligible. All the fluid physical properties are functions of temperature.

The temperature dependence of the properties is defined in a piecewise linear sense. Hence, nonlinear material property dependence can be emulated by assuming linear temperature dependence over small temperature ranges in such a way that the material property values at the extremities of each linear segment coincide with those of the actual temperature dependence curve.

The spatial dependence of the material properties of solids is defined using the concept of material property groups. Each element in the mesh is associated with a particular material property group. As material properties are not functions of position in a material group, elements are isotropic with respect to thermal conductivity, heat capacity and density, whereas the entire domain may be non-isotropic in a discrete manner.

The spatial dependence of the properties of fluids is defined by associating different fluids with each element edge. Within each fluid the physical properties are functions of temperature only.

5.5 IMPLEMENTATION OF CONVECTION BOUNDARY CONDITION

The convective heat transfer effects which occur due to a fluid being in contact with the surface of a solid, are modelled by Newton's Law of Cooling:

$$q = h(T_s - T_f) \qquad [5.1]$$

where q = heat flux at surface (W/m^2)
h = heat transfer coefficient (W/m^{2}°C)
T_s = surface temperature (°C)
T_f = fluid bulk temperature (°C)

For the purposes of this study, it is assumed that the fluid temperature, T_f, does not change due to the energy exchange between solid and fluid. It is possible to endow time dependence on T_f by associating it with a time function - a dimensionless multiplier which is a piecewise linear function of time. Time functions are discussed in the NOTHCAN user's guide.

In the context in which equation 5.1 is used, the surface temperature, T_s, is unknown and is solved for in the finite element analysis. The implementation of the convection boundary condition , results in a contribution to both the stiffness matrix and the load vector in the finite element formulation. The way in which this occurs is shown in chapter 3.

The heat transfer coefficient, h, of each fluid may be defined in one of two ways. Firstly, it can be defined as a piecewise, linear function of temperature and, secondly, it can be indirectly defined by specifying a correlation from which to evaluate it. Table 5.1 below lists the currently offered correlations.

Table *5.1: Heat transfer coefficient correlations for various configurations.*

Description	Valid Range	Correlation	ID
	Natural Convection		
Horizontal surface, heated surface up or cooled surface down.	$10^5 < GrPr < 2\cdot10^7$ $2\cdot10^7 < GrPr < 3\cdot10^{10}$	$Nu = .54(GrPr)^{.25}$ $Nu = .14(GrPr)^{.33}$	1
Horizontal surface, heated surface down or cooled surface up.	$3\cdot10^5 < GrPr < 3\cdot10^{10}$	$Nu = .27(GrPr)^{.25}$	2
Vertical surface	$10^3 < GrPr < 10^9$ $10^9 < GrPr < 10^{12}$	$Nu = .59(GrPr)^{.25}$ $Nu = .14(GrPr)^{.33}$	3
	Forced Convection		
Flow parallel to a flat plate.	$Re < 5\cdot10^5$	$Nu = .664Re^{.5}Pr^{.33}$	4
	$Re >= 5\cdot10^5$	$Nu = .036Pr^{.33}(Re^{.8} - 23200)$	5
	$Re > 5\cdot10^5$	$Nu = .036Re^{.8}Pr^{.33}$	6

Information about the dimensionless groups Nu, Gr, Pr and Re can be found in chapter 2. These correlations can be found in standard heat transfer texts such as McAdams 1954. If the correlations listed above are not appropriate for a particular problem, it is necessary for the user to evaluate the heat transfer coefficients in some alternative manner and then use the first option mentioned above.

It is possible to associate a time function with the heat transfer coefficient which makes it possible to switch the convection boundary condition on or off at various times during the analysis.

The basic algorithm used in implementing the convection boundary condition is presented below:

```
• while the time is less than the total analysis time
    • while the temperature distribution has not converged
        . for each edge
            . evaluate the heat transfer coefficient
                . determine the average temperature of the edge
                . determine the current ambient fluid temperature
                . if the heat transfer coefficient is user specified
                    . use the piecewise linear curve to set the coefficient to
                      its value at the average edge temperature
                . if the heat transfer coefficient is to be determined from
                  correlations
                    . determine the film temperature
                    . evaluate the fluid thermal conductivity, density,
                      viscosity, heat capacity, and volume expansivity at the
                      film temperature
                    . evaluate the heat transfer coefficient
            . determine the convection stiffness matrix for the edge
            . determine the convection load vector for the edge
        . perform a finite element analysis
    • check the temperature distribution convergence
• increment the analysis time
```

5.6 IMPLEMENTATION OF THE RADIATION BOUNDARY CONDITION

The implementation of radiant heat transfer as a boundary condition in this study is based on the following assumptions:

1) Radiant exchange occurs between flat, rectangular surfaces.

2) All surfaces are gray and opaque.

3) Surfaces may be separated by vacuum, non-absorbing media

or gray, absorbing media.

4) All surfaces reflect radiation uniformly and diffusely

5) All surfaces have uniform temperatures.

6) All surfaces may interact with an environment which is assumed to be gray.

It is necessary to subdivide the various surfaces interacting radiatively into two categories due to the fact that not all surfaces are part of the solid body undergoing analysis. The first category is 'active surfaces' which are those surfaces which are part of the body under examination. As all analyses are two-dimensional, an active surface is created by projecting the line which defines the element edge into a plane. The second category is 'passive surfaces' which are surfaces which interact with, but are not part of, the body being analysed. Groups of interacting active and passive surfaces are termed 'clusters' and the surfaces belonging to one cluster do not 'see' any of the surfaces which belong to another cluster.

Passive surfaces have uniform temperatures which can be linked to a time function. These temperatures are unaffected by any energy exchange which occurs. The temperatures of active surfaces, however, have to be solved for and a brief description of this procedure follows. Firstly, based on the current nodal temperatures of the relevant element edges, the averaged active surface temperatures are calculated and used, together with the passive surfaces temperatures, to determine the flux entering the body through the active surfaces. Then, using this flux as a prescribed flux boundary condition, a new temperature distribution is calculated by carrying out a finite element analysis from which new active surface temperatures are evaluated. If the new and previous temperature distributions differ significantly, the procedure is repeated. When, or if, convergence is achieved, the analysis proceeds to the next time-step.

Firstly, the flux entering each active surfaces must be determined for given and assumed temperature conditions. This entails solving the system of linear equations represented by equation 5.2 for the radiosity of each surface. These radiosities are then used to evaluate the fluxes.

$$\Sigma_j[\delta_{ij} - (1 - \epsilon_i)F_{ij}\tau_{mji}]J_j = \epsilon_i\sigma T_i^4 + (1 - \epsilon_i)\sigma T_m^4\Sigma_j\epsilon_{mij}F_{ij}$$

$$i,j = 1,2,...,n \qquad [5.2]$$

$$q_i = \frac{\epsilon_i}{1-\epsilon_i}[\sigma T_i^4 - J_i] \qquad [5.3]$$

Of the various factors in the above equations, three require further discussion. The first two concern non-opaque media - the transmissivity and the emissivity, τ_{ij} and ϵ_{gij}. The values are 1 and 0, respectively, if the non-opaque medium is non-absorbing. However, if the medium does absorb radiation, then the calculation of these values is dependent on the mean beam length between the surface pair of interest and, if the absorbing medium is a gas - which it normally is at the temperatures at which these properties are significant - then they are also dependent on the partial pressure of the absorbing component. Currently the user must determine these values manually as they are difficult to evaluate using an algorithm. Hottel 1967 explains how this may be done.

The next factor which must be discussed and which is of central importance whether or not absorbing media are present, is the shape factor. This is a factor which is configuration dependent and represents the fraction of the radiation leaving surface i which is received by surface j. The shape factor between two surfaces can be determined analytically for some simple geometries. As simple geometries (parallel planes, perpendicular planes with common edges) are not necessarily a part of the problem to be analysed, one has to resort to numerical methods. Some commonly used numerical methods are:

- The crossed-string method
- Specialised methods relating to limited range of geometries
- The Monte-Carlo method
- The method of ray casting

The crossed-string method is a very approximate method of evaluating shape factors and is best applied to bundles of tubes such as are found in nuclear reactors. The details of the method can be found in many radiation heat transfer texts - for example Siegel 1972. The Monte-Carlo method has been applied generally to the determination of shape factors (Djuric and Novakovic 1979). It is based on the determination of the number of times radiant energy emitted from a random position on the surface of interest in a random direction 'hits' the other surface of interest. A statistical analysis of both the number of 'hits' and the total number of energy emissions, yields the shape factor.

Both the crossed-string method and the specialised methods were not seriously considered as they are not sufficiently general. The Monte-Carlo method was not considered due to the fact that a large number of random numbers - of the order of 500 000 energy emissions to attain an accuracy of about 1% - is required to attain reasonable accuracy and, even then, the method does not guarantee convergence to the true value.

The method of ray casting (Maxwell <u>et</u> <u>al</u>. 1986) was chosen to determine shape factors because it is general, does not involve the random emission of energy and always converges to the true value. The method is described below.

Consider two arbitrarily oriented flat surfaces, surfaces 1 and 2. Assume that F_{12} is to be determined. Select a point on surface 1, dA_1, and construct a hemisphere which completely encloses surface 2, centred on dA_1. Project a large number of lines (rays) from dA_1 which intercept the surface of the hemisphere and which touch the edge of surface 2 at a number of different points along its perimeter. Effectively, this process demarcates the approximate shadow of surface 2 on the surface of the hemisphere with respect to radiation from dA_1. Next, map the shadow area on the hemispherical surface onto the plane in which surface 1 lies. The ratio of this projected area and the area of the circular plane formed by the intersection of the hemisphere and the plane containing surface 1, is the approximate shape factor between the point, dA_1, and surface 2. One needs to repeat this procedure over the entire area of surface 1 in order to obtain F_{12}.

This method is implemented in NOTHCAN by subdividing each surface into a large number of identical subareas and casting a finite number of rays from the centre of each subarea. The final shape factor is the arithmetic average of the sum of the subarea shape factors. If the group of n surfaces under consideration forms an enclosure then equation 5.4 is valid.

$$\Sigma_j F_{ij} = 1 \quad i,j = 1,2,..,n \qquad [5.4]$$

If this condition is violated then the shape factors are normalised. If the group of surfaces does not form an enclosure and 'sees' an environment, then the above condition is used to evaluate the shape factor between each surface and the environment.

The accuracy of the shape factors generated using the method of ray casting is improved by using a larger number of subareas and by casting more rays. However, this can become very expensive in terms of computer time and a compromise between time and accuracy requirements must be made.

Once the shape factors have been determined the problem is relatively easily solved. The algorithm for radiation boundary conditions is present below.

```
• Determine the shape factors between all interacting surfaces
    • Convert element edges to active surfaces - planes
    • Determine the equations of the surfaces
    • For each surface
        . Divide the current surface into sub-areas
        . For each sub-area
            . Determine the radius and equation of the hemisphere, centered on the current sub-area, completely enclosing the receiving surface
            . Cast a number of rays from the centre of the sub-area and record where they intersect the hemisphere
            . Map the surface's shadow onto the plane of the emitting body
            . Determine the area of the mapped shadow
            . Calculate the shape factor $F_{dAi-j}$
    • Calculate the shape factor $F_{i-j}$
• While the analysis time is less than the final time
    • While the difference between current and previous temperature distributions within the body is not sufficiently small
        . Set the previous temperature distribution to be the current distribution
            . Determine the fluxes leaving the active surfaces
                . Determine the temperatures of the active and passive surfaces
                . Assemble the system of radiation equations
                . Solve the system of radiation equations
                . Calculate the fluxes
            . Using the fluxes as if they were prescribed flux boundary conditions, perform a finite element analysis to determine the new temperature distribution within the body
            . Determine the difference between the current and previous temperature distributions
• Increment the time-step
```

5.7 Implementation of the Mass Transfer Boundary Condition

As discussed in the relevant section in chapter 2, the mass transfer boundary condition implies the effects of evaporation from or condensation on the surface of a body, the temperature distribution within which is being modelled. As the mass and energy transfer processes are not analysed rigorously, a model has been developed to approximate the mechanism, and is presented below. Before the model is discussed, the major assumptions are listed. Additional assumptions are mentioned in the course of the model description to preserve the structure of the discussion. The main assumptions are:

1) The gas which is involved in the mass transfer process is an ideal, binary mixture, one component of which is not condensable in the range of operating conditions anticipated by the model.
2) The solubility of the non-condensable component in the liquid is negligible as is the solubility of the solid in the liquid.

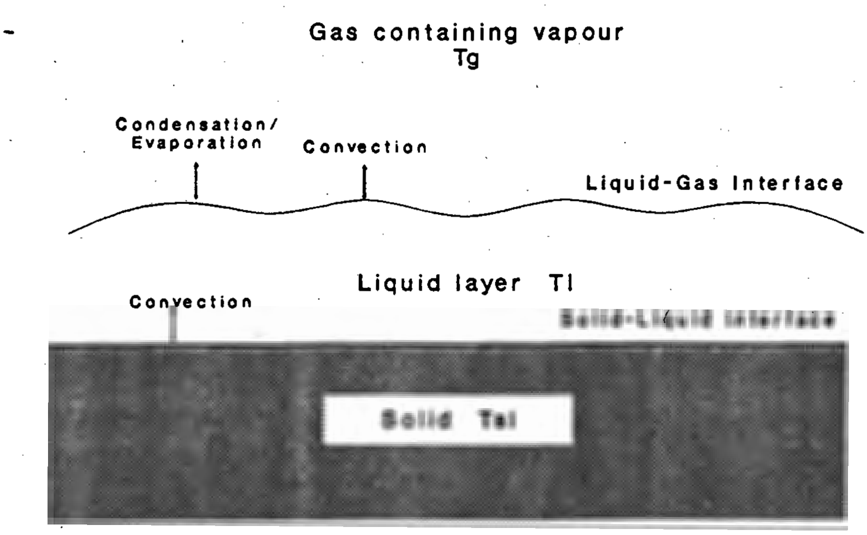


Figure 5.1 *A solid surface in contact with a liquid layer and a moist gas.*

3) The mass transfer rate is not sufficiently high that it interferes with the convective heat transfer which occurs simultaneously.

With reference to the first assumption, the term 'binary mixture' refers to both mixtures consisting of two pure components and mixtures consisting of a pure condensable component together with a number of non-condensable components. In the case of a typical problem with a mass transfer boundary condition, there are three phases interacting - solid, liquid and a gas mixture - which results in two phase interfaces, one between the solid and the liquid and the other between the liquid and the gas mixture. This situation is depicted in figure 5.1.

Figure 5.1 also depicts the various heat transfer processes which occur in conjunction with mass transfer. Firstly, there is heat conduction in the solid; secondly, there is convective heat transfer between the solid and the liquid layer; thirdly, there is convective heat transfer between the liquid layer and the gas and, fourthly, energy is transferred between the liquid layer and the gas due to the latent heat effects of condensation or evaporation. Heat conduction is already modelled directly by the finite element method which leaves the other three processes to be dealt with in the mass transfer boundary condition model.

It is assumed that the following properties of the gas are known:

- Temperature and its time dependency
- The absolute pressure and its time dependency
- The humidity and its time dependency
- The gas physical properties and their temperature dependency. The relevant properties are thermal conductivity, density, viscosity, heat capacity and volume expansivity.

It is further assumed that the gas temperature and humidity are not affected by any energy or mass transfer between the gas and the liquid.

It is assumed that the following properties of the liquid layer are known:

- The initial temperature and its time dependency

- The initial thickness and its time dependency
- The liquid physical properties and their temperature dependency. The relevant properties are thermal conductivity, density, viscosity, heat capacity, volume expansivity, gas-liquid mass transfer diffusivity.

In order to determine the effect of the mass transfer boundary condition on the temperature distribution in the solid, it is necessary to reduce it to a pseudo-prescribed flux or temperature boundary condition. In the former case, convection into the liquid is assumed if the edge is wet and convection into the gas is assumed if it is dry. In the latter case, which may be applied only to wet edges, the prescribed surface temperature is taken as a weighted average according to equation 5.5 below:

$$T_{sl} = \omega T_l + (1-\omega) T_s \qquad [5.5]$$

T_{sl} = solid-liquid interface temperature (°C)
T_l = liquid layer bulk temperature (°C)
T_s = averaged nodal edge temperatures from the previous iteration or time-step (°C)
ω = user-defined weighting factor

Irrespective of the method used, the liquid bulk temperature, T_l, is required and, as it is affected by the energy transfer through both interfaces, it is also necessary to approximate the liquid-gas interface temperature. This temperature is used in approximating the convective energy exchange between the liquid and the gas according to equation 5.6.

$$q = h(T_{lg} - T_g) \qquad [5.6]$$

where q = flux exchanged between liquid and gas (W/m^2)
h = heat transfer coefficient (W/m$^{2\circ}$C)
T_{lg} = liquid-gas interface temperature (°C)
T_g = gas bulk temperature .(°C)

The liquid-gas interface temperature may be approximated in one of two ways. Firstly, it may be evaluated as a weighted average according to equation 5.7.

$$T_{lg} = \omega T_l + (1-\omega) T_g \qquad [5.7]$$

where T_{lg} = liquid-gas interface temperature (°C)
T_l = liquid layer bulk temperature (°C)
T_g = gas bulk temperature (°C)

ω = user-defined weighting factor

Secondly, the liquid-gas interface temperature may be approximated by means of modified lumped-parameter analysis. Lumped-parameter analysis is explained in Holman 1976 (in which it is termed Lumped-heat-capacity analysis). It is a method of predicting the time dependence of the temperature of a solid which is exchanging energy convectively with a fluid at constant temperature, based on the assumption that the solid temperature is uniform at all times. This assumption implies that the resistance to conductive heat transfer in the solid is small in comparison with the convection resistance at the surface. The ratio of these resistances, called the Biot number, Bi, can be used to establish the validity of the uniform temperature assumption.

The modification to the lumped-parameter analysis explained above, entails assuming that the solid can be replaced by a liquid and the fluid by a gas. Although this is a questionable assumption it is felt that it leads to a better approximation to the liquid-gas interface temperature in many circumstances. The modified lumped-parameter analysis results in the following equation:

$$T_{gl} = (T_o - T_g)e^{-(hA/\rho VC)t} + T_g \qquad \text{if } Bi = \frac{hL}{k} < 0.1 \qquad [5.8]$$

where T_{gl} = liquid-gas interface temperature (°C)
T_o = initial liquid layer temperature (°C)
T_g = gas bulk temperature (°C)
h = gas heat transfer coefficient ($W/m^{2}{}^{o}C$)
A = surface area of contact between liquid and gas (m^2)
ρ = liquid density (kg/m^3)
C = heat capacity of the liquid ($J/kg^{o}C$)
V = volume of the liquid (m^3)
t = time (s)
L = liquid layer thickness (m)
k = liquid thermal conductivity ($W/m^{o}C$)

It is possible that, even if the Biot number condition is relaxed to some extent, the temperature approximation will be better than that determined using the weighted average approach.

The heat transferred due to mass transfer is now discussed. The energy added to or removed from the liquid layer due to condensation or evaporation is dependent on the rate at which either of these two processes occurs and the latent heat of vaporisation. The mass transfer rate can be determined using equation 5.9.

$$m = h_m(p_{dew} - p_s) \qquad [5.9]$$

where m = mass transfer rate (kg/m^2s)
h_m = mass transfer coefficient ($kg/m^2atm.s$)
p_{dew} = vapour saturation pressure at the dew point temperature T_{dew}, or, equivalently, the partial pressure of the vapour in the gas (atm)
p_s = vapour saturation pressure at T_{lg} (atm)

The mass transfer coefficient is determined using the Chilton-Colburn j-factor analogy explained in the mass transfer section of chapter 2. The partial pressure of the vapour is indirectly specified by the humidity and is evaluated using the following equation:

$$P_{dew} = HP_{dry} \qquad [5.10]$$

where p_{dew} = vapour pressure of vapour (atm)
H = relative humidity of the gas
p_{dry} = vapour saturation pressure at T_g (atm)

The saturation pressures are calculated using Antoine's equation which is presented in the mass transfer section of Chapter 2.

The latent heat of vaporisation is evaluated at the liquid-gas interface temperature using Watson's correlation which is presented in chapter 2.

The energy entering or leaving the liquid layer due to mass transfer is determined using equation 5.11.

$$q = mL \qquad [5.11]$$

where q = mass transfer energy flux (W/m^2)
m = mass transfer rate (kg/m^2s)
L = latent heat of vaporisation (J/kg)

Having evaluated all the energy transferred by the various processes which occur, it is now possible to determine the resultant change in the liquid layer bulk temperature. This is done by carrying out an energy balance over the liquid layer.

$$C_{pl}(T_{ln} - T_l)A\rho l = At(q_m + q_{cs} + q_{cg}) \qquad [5.12]$$

where C_{pl} = the liquid heat capacity (J/kg°C)
T_{lm} = the new liquid layer bulk temperature (°C)
T_l = the previous liquid layer bulk temperature (°C)
A = liquid layer surface area (m^2)
l = liquid layer thickness (m)
ρ = liquid density (kg/m^3)
t = length of time during which the energy transfer processes occur (s)
q_m = flux due to evaporation of condensation (W/m^2)
q_c = flux entering the liquid from the solid (W/m^2)
q_{cg} = flux entering the liquid due to convection within the gas (W/m^2)

In equation 5.12 the flux entering the liquid from the solid is either that resulting from convective heat transfer or the reaction flux, depending on which of the pseudo-convection or prescribed temperature boundary conditions is used.

As mass transfer is occurring, it is necessary to calculate a new liquid layer thickness. This is done using equation 5.13.

$$l_n = l + tm/\rho \qquad [5.13]$$

where l_n = new liquid layer thickness (m)
l = previous liquid layer thickness (m)
t = length of time during which energy transfer processes occur (s)
m = mass transfer rate (kg/m^2s)
ρ = liquid density (kg/m^3)

In the preceding discussion it has been assumed that the surface is wet. However, this is not always the case and it is possible to identify three surface conditions:

1) **A dry surface.** In this case the mass transfer boundary condition is treated as a convection boundary condition and a check is made at every iteration or time-step to determine if condensation will occur. This check involves comparing the solid surface temperature with the dew point temperature of the gas. If the surface temperature is less than the dew point temperature condensation occurs. The gas dew point temperature is calculated at the start of the analysis and whenever the

conditions of the gas change.

2) **A wet surface.** This is the case which has been discussed in the preceding section. At every iteration or time-step a check is made to determine if the surface has dried.

3) **A continually wetted surface.** In this case the surface is being constantly wetted by a process other than condensation - for example, the surface is being hosed down. This case is treated as a convection boundary condition into the liquid.

An algorithm describing the implementation of the mass transfer boundary condition is presented below:

```
• while the analysis time is less than the total time
    • for each edge
        . determine which type of mass transfer boundary condition is active
        . initialise some variables and control parameters
        . if the condition of the gas has changed and the edge is currently dry, then test whether or not condensation will occur.  If it will, set the liquid layer surface temperature to the gas dew point temperature
        . if the edge is newly wet, set the temperature gradient, the liquid layer thickness and the fluid temperature
        . if condensation is to occur, determine the wet bulb temperature of the gas
    • while the temperature distribution has not converged
        . for each edge
            . set parameters indicating whether the edge is currently
                          dry
                          wet
                          continuously wetted
                          dry but being condensed upon
                          wet and being condensed upon
                          not experiencing condensation
            . if a dry edge has just been condensed on
                . determine the average temperature of the side
                . determine the heat transfer coefficient between liquid and solid
                . evaluate the convection stiffness matrix and load vector
            . if the edge is dry
                . determine the gas temperature and average edge temperature
                . determine the heat transfer coefficient between solid and gas
                . evaluate the convection stiffness matrix and load vector
            . if the edge is wet
                . if the analysis is transient
                    . if the weighted average approximation is used
                        . determine the average edge temperature
                        . evaluate the weighted average temperature
        . force the nodes of the edge to be prescribed
          temperatures of value the weighted average temperature
      . if the convection approximation is used
        . determine the average edge temperature
        . determine the heat transfer coefficient between solid
          and liquid
        . determine the convection stiffness matrix and load
          vector
    . if the analysis is steady state
      . determine the current gas wet bulb temperature
      . determine the average edge temperature
      . set the liquid bulk temperature as an average of the edge
        temperature and the wet-bulb temperature
      . determine the convection stiffness matrix and load vector
  . if the edge is continuously wetted
    . set the nodes of the edge to have prescribed temperatures of
      value the current fluid temperature
. perform a finite element analysis using the pseudo-boundary
  conditions

. determine to how many elements each wetted node is connected
. for each edge
  . determine which type of mass transfer boundary condition is
    active
  . if the edge is wet
    . if the analysis is transient
      . determine the liquid gas interface temperature using either
        the weighted average or the lumped parameter approximations
      . determine the convection flux between liquid and gas
      . determine the latent heat of vaporisation
      . determine the partial and vapour pressures
      . determine the mass transfer coefficient
      . determine the latent heat change of the liquid layer
      . if the convection approximation to the solid-liquid
        temperature is used
        . determine the convection flux at each node
      . if the weighted average approximation to the solid-liquid
        temperature is used
        . determine the reaction fluxes
      . determine the total energy entering the liquid from the
        solid
      . determine the new liquid layer temperature using an energy
        balance
      . evaluate the new liquid layer thickness
      . check that the surface has not dried
      . check that the liquid layer temperature is not 'impossible'
    . if the analysis is steady state
      . set the new liquid layer temperature to the average of the
        surface and the gas wet-bulb temperatures
```

```
          . if the edge is dry but is being condensed upon
              . evaluate the average edge temperature
              . set the liquid surface temperature to the gas wet-bulb
                temperature
              . evaluate the gas-liquid heat transfer coefficient
              . evaluate the convection flux between liquid and gas
              . determine the latent heat of vaporisation
              . evaluate partial and vapour pressures
              . determine the mass transfer coefficient
              . determine the latent heat transfer rate
              . determine the new liquid layer thickness
              . if the convection approximation is used
                  . determine the convection flux at each node
              . if the weighted average approximation is used
                  . determine the reaction flux at each node
              . determine the net energy entering the liquid and determine the
                new temperature by energy balance.
              . check whether the temperature is 'impossible' and set
                appropriate flags
      . check the convergence of the temperature distribution
  • increment the analysis time
```

5.8 Infinite elements

The inclusion of infinite elements, as described in chapter 3, into an existing finite element program which is based on isoparametric elements, is not difficult. The reason for this is that infinite elements are mapped onto the same family of master elements that are used by standard isoparametric finite elements. This means that the same shape functions are used for the field variable in both cases but that the mapping functions are different - infinite elements are not isoparametric. All this means in terms of implementation is that the Jacobean must be evaluated using the infinite element mapping functions rather than the standard shape functions. The Jacobean matrices for finite and infinite elements, respectively, are shown below. The nomenclature is consistent with that established in chapter 3.

$$J = \begin{bmatrix} \frac{\partial \hat{N}_i}{\partial \xi} x_i & \frac{\partial \hat{N}_i}{\partial \xi} y_i \\ \frac{\partial \hat{N}_i}{\partial \eta} x_i & \frac{\partial \hat{N}_i}{\partial \eta} y_i \end{bmatrix} \qquad J = \begin{bmatrix} \frac{\partial M_i}{\partial \xi} x_i & \frac{\partial M_i}{\partial \xi} y_i \\ \frac{\partial M_i}{\partial \eta} x_i & \frac{\partial M_i}{\partial \eta} y_i \end{bmatrix} \tag{5.14}$$

The only boundary condition which may be used on the boundary at infinity is a prescribed temperature boundary condition. There appear to be two approaches to dealing with a non-zero far field boundary condition. Damjanic and Owen 1984 always assume zero temperatures at infinity and hence assemble infinite element matrices of smaller size. The element matrices are then assembled and solved. The non-zero prescribed temperature is then introduced with some mathematical rationalisation. The main disadvantage of their method is that the matrix solver must be capable of dealing with different element matrix sizes. Zienkiewicz et al. 1981, on the other hand, always assume non-zero far field prescribed temperature. This allows infinite elements to be treated in exactly the same way as finite elements in terms of dealing with prescribed temperature conditions.

Figure 5.2 depicts the master element equivalents of the infinite elements selected for inclusion into the program. It can be seen that there are three singly and one doubly infinite element, the latter for use on mesh corners.

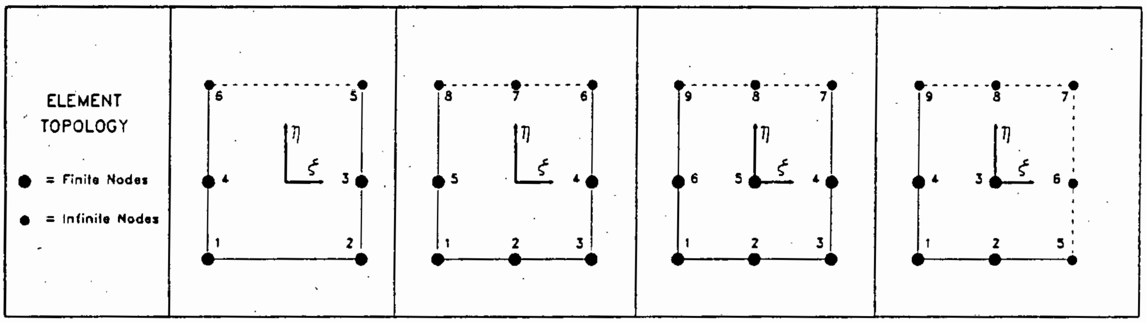
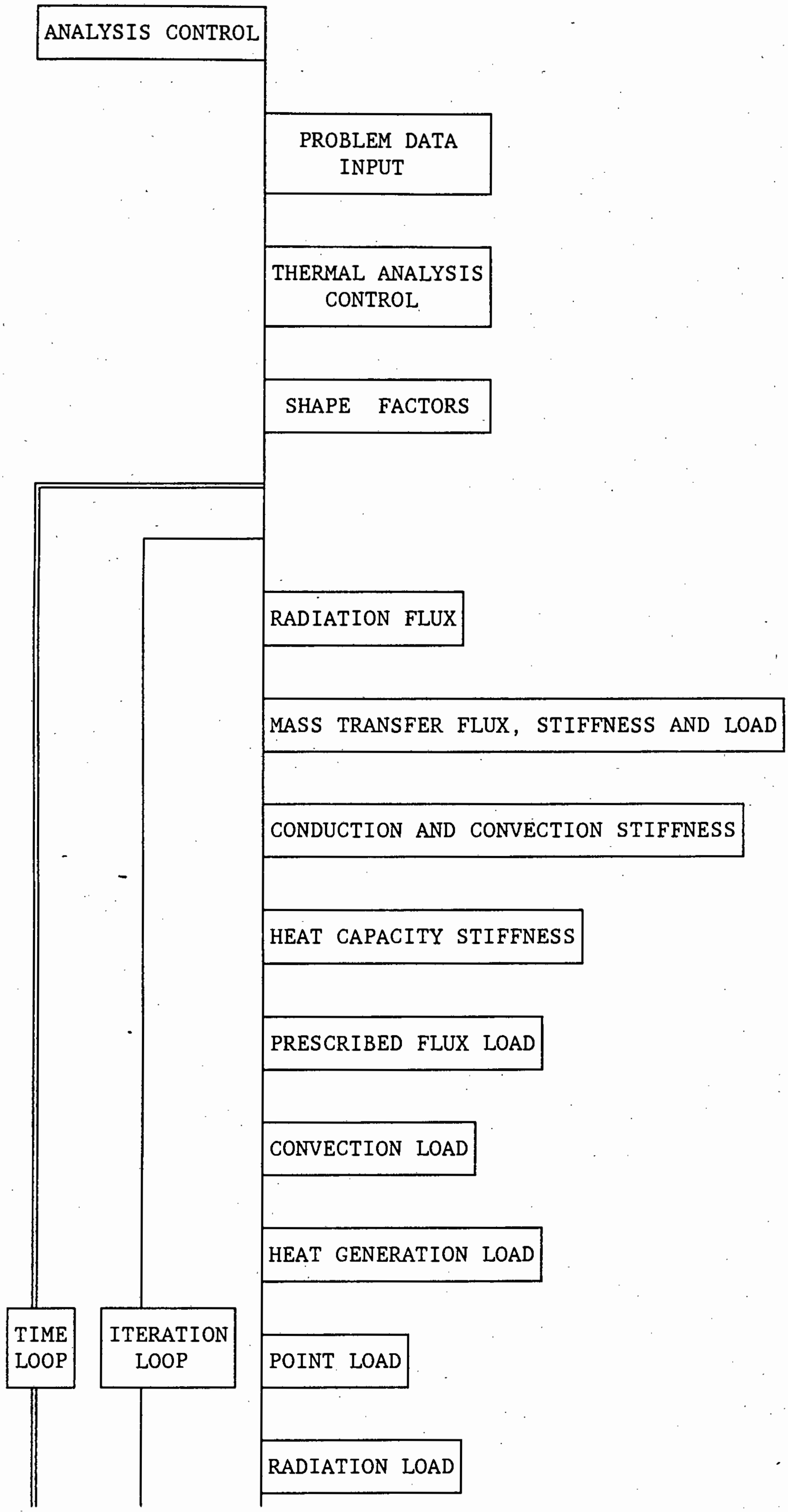


Figure *5.2 Mapped infinite elements.*

The mapping functions of each element can be found in the appendix.

5.9 Program Module Functions

Figure 5.3 depicts a flow-chart of the various functional modules of NOTHCAN. The details of each module are discussed briefly below.


ANALYSIS CONTROL
PROBLEM DATA INPUT
THERMAL ANALYSIS CONTROL
SHAPE FACTORS
RADIATION FLUX
MASS TRANSFER FLUX, STIFFNESS AND LOAD
CONDUCTION AND CONVECTION STIFFNESS
HEAT CAPACITY STIFFNESS
PRESCRIBED FLUX LOAD
CONVECTION LOAD
HEAT GENERATION LOAD
POINT LOAD
RADIATION LOAD
TIME LOOP
ITERATION LOOP

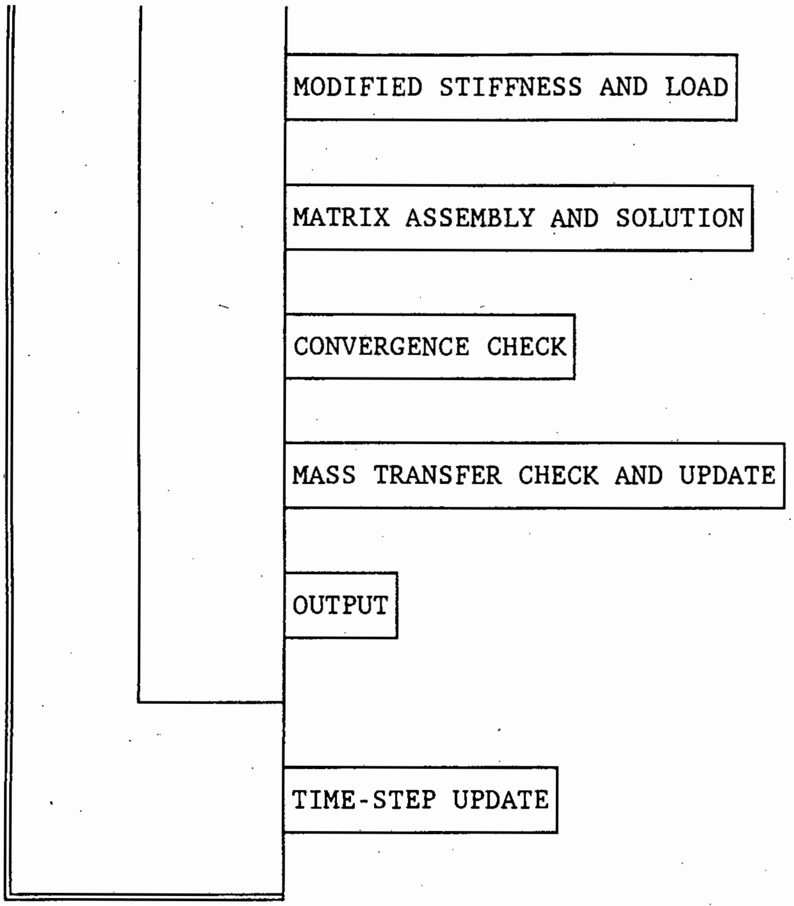

MODIFIED STIFFNESS AND LOAD
MATRIX ASSEMBLY AND SOLUTION
CONVERGENCE CHECK
MASS TRANSFER CHECK AND UPDATE
OUTPUT
TIME-STEP UPDATE

Figure *5.3 Module function flowchart of NOTHCAN.*

Analysis Control. This is the routine which controls the program. Initially, the input routines are called and, depending on the data read, control is passed to the thermal analysis control module. If an error occurred in the input routines or the data-check only operation was specified, the program terminates. The existence of this high level control structure allows the addition of modules which might be used to analyse non-thermal problems or even modules which use a different method - the boundary element method, for instance - to analyse thermal problems, depending on the analysis control data specified.

Problem Data Input. This module reads and echoes (if required) the data which describes both the problem to be analysed and the variant of the available analysis methods to use. The input data is also checked in this module.

Thermal Analysis Control. This is the module which controls all aspects of thermal analysis. For instance, if the analysis is nonlinear it controls the iteration procedure. It calls all routines which contribute to the construction of the stiffness matrices and load vectors as well as those routines which deal with the assembly and solution of the stiffness and load vectors

Shape Factors. This module determines the shape factors required for the analysis of a radiation boundary condition. This is done using the method of ray casting described earlier in this chapter. This routine is only invoked if there is a radiation boundary condition.

Radiation Flux. This module determines the flux entering each element edge with a radiation boundary condition using the method described earlier in this chapter in the radiation section. It is only invoked if there is a radiation boundary condition.

Mass Transfer Flux, Stiffness and Load. This module constructs the mass transfer boundary condition stiffness matrix and load vector based on the degree of wetness of each element edge and the solid-liquid interface temperature model selected in the problem description data file. It is invoked only if there is a mass transfer boundary condition.

Conduction and Convection Stiffness. This module constructs the element conduction stiffness matrices and, if appropriate, evaluates the convection boundary condition contribution to the overall element stiffness matrices.

Heat Capacity Stiffness. This module evaluates the heat capacity stiffness matrix. If a steady state analysis is being performed this module is not invoked.

Prescribed flux Load. This module evaluates the prescribed flux boundary condition load vector. It is only called if there is such a boundary condition.

Convection Load. This module constructs the convection boundary condition load vector if there is such a boundary condition.

Heat Generation Load. This module evaluates the contribution to the overall load vector of each element arising from internal heat generation. It is only called if there is internal heat generation.

Point Load. This routine evaluates the value of the point loads at the current time-step. If there are no point loads, this routine is not called.

Radiation Load. This module evaluates the contribution of the radiation boundary condition to the overall load vector of each element based on the fluxes evaluated previously. This load vector is constructed by treating the radiation flux a a pseudo-prescribed flux. The module is only invoked if there is a radiation boundary condition.

Modified Stiffness and Load. This module assembles the overall element stiffness matrices and load vectors and combines the assembled element stiffness matrices and load vectors into modified forms according to the time-integration scheme used.

Matrix Assembly and Solution. This module assembles the element stiffness matrices and load vectors. If there are any point loads they are added to the load vector after it has been assembled. The system of linear equations is then solved for the temperature at each node as well as the reaction flux at each node which has a prescribed temperature boundary condition.

Convergence Check. This module compares the temperature distribution in the body at the current iteration with that at the previous iteration. If any of the nodal temperatures differ by more than a user-specified value, the analysis is repeated without updating the analysis time. This module is only called if the analysis is nonlinear.

Mass Transfer Check and Update. This module performs checks to ensure that the current state of wetness of the various element edges with mass transfer boundary conditions is valid. For example, a surface from which liquid is evaporating may have dried. This routine performs the energy balance used to evaluate the new liquid layer bulk temperature and determines the new liquid layer thickness. It also carries out checks to ensure that 'impossible' temperature values do not occur. This module is only invoked if there is a mass transfer boundary condition.

Output. This module selectively writes results to a file at the end of each iteration and/or time-step. Selected nodal temperatures may be written to a second file at the end of every time-step. Information in the problem data file controls which and when results are written to file.

Time-step Update. This module sets the temperature distribution in the body for the next time-step in accordance with the time-integration scheme used.

CHAPTER 6 : VALIDATIVE SAMPLE PROBLEMS

This chapter presents some of the problems used to verify the solutions generated by NOTHCAN. The solutions to seven simple problems are presented to demonstrate the accuracy of solution to problems with the following conditions or characteristics:

1) prescribed temperature boundary condition
2) prescribed flux boundary condition
3) convection boundary condition
4) internal heat generation
5) mass transfer boundary condition
6) radiation boundary condition
7) infinite elements.

In the cases of problems of type 1 to 4 and 7, the solutions generated by NOTHCAN are compared with analytical solutions generated from the appropriate equations in Carslaw and Jaeger 1959.

Unless otherwise stated, all the sample problems are solved using the finite element model depicted in figure 6.1:

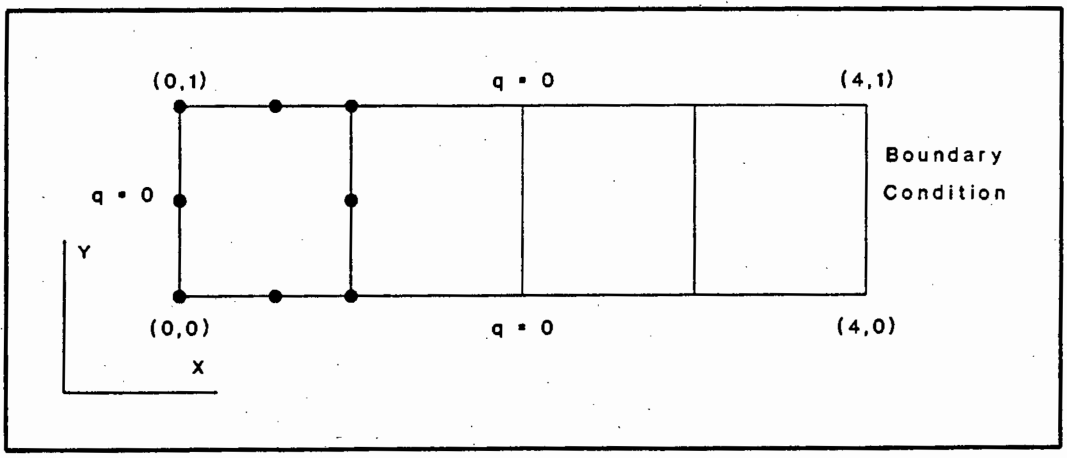


Figure *6.1 Finite element mesh for sample problems.*

Unless otherwise stated, the parameters and analytical properties listed in table 6.1 characterise the problems.

Table *6.1: Sample problem parameters.*

k	=	1.0 W/m°C
C_p	=	1.0 J/kg°C
ρ	=	1.0 kg/m^3
T_o	=	0.0 °C
α	=	1.0
Δt	=	1 s

The problem represented by figure 6.1 and table 6.1 is that of determining the transient temperature distribution in a solid bar, three sides of which are insulated while the fourth has some boundary condition imposed on it. The body has a uniform initial temperature distribution, T_o.

6.1 Prescribed temperature boundary condition

The problem being modelled is the standard problem with a prescribed temperature, $T_p = 1$, at $x=4$. The analytical solution to this problem is presented in Carslaw and Jaeger 1959 (page 101).

Figures 6.2 and 6.3 depict the analytical and finite element solutions of this problem at two different positions in the body using 4- and 8-noded elements.

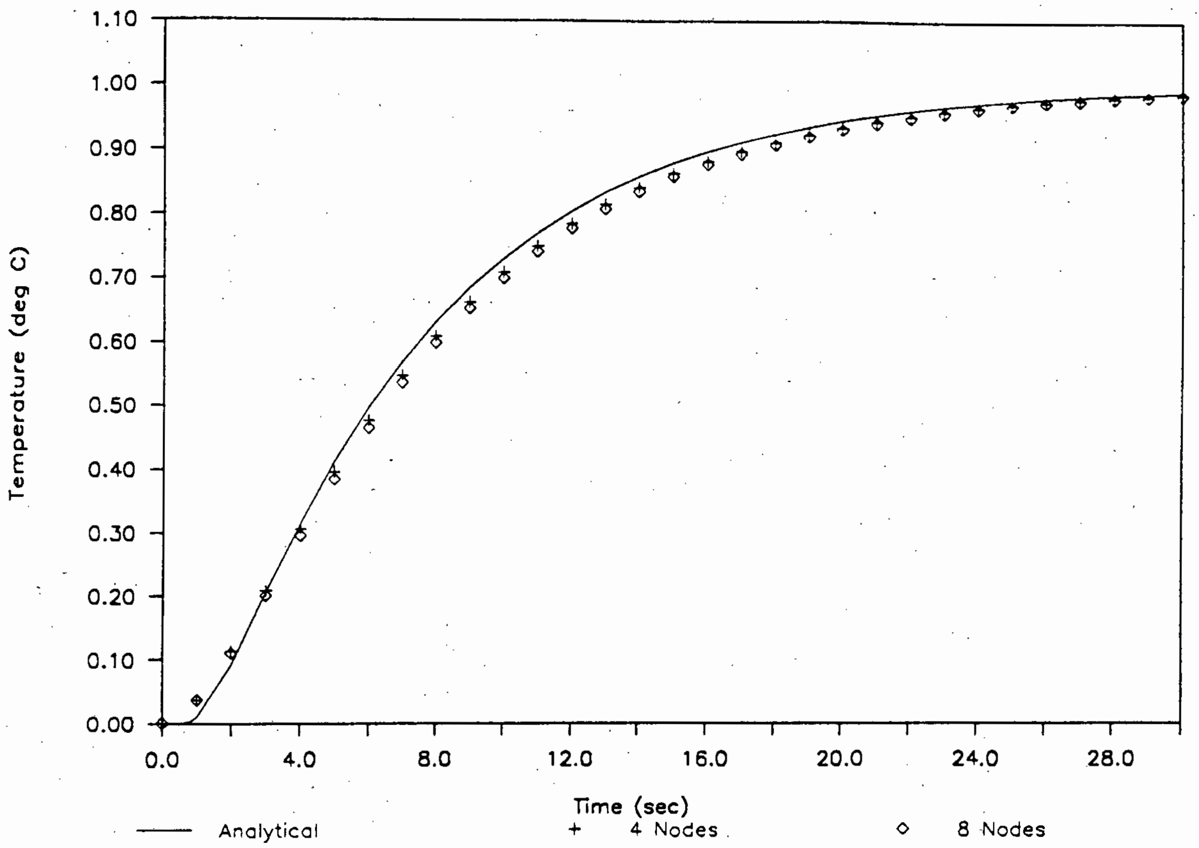


Figure *6.2 Prescribed temperature: analytical and finite element solutions at x=0.*

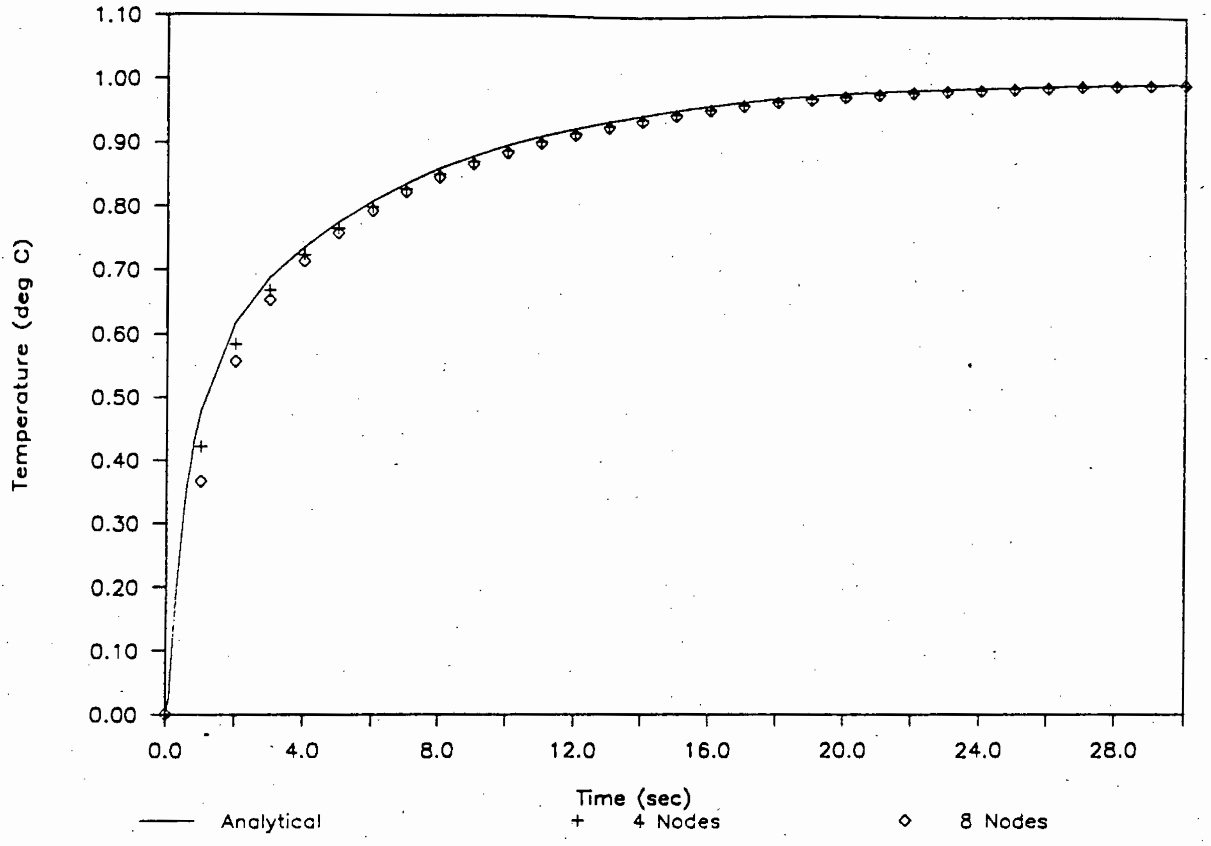


Figure *6.3 Prescribed temperature: analytical and finite element solutions at x=3.*

In both graphs, the finite element solution closely approximates the analytical solution, the worst deviation occurring during the initial time-steps. It is noticeable that, in both cases, the use of 4-noded elements results in a closer approximation of the actual solution. Damjanic and Owen 1982 have explained this observation in terms of the results of spectral analysis and a result of note is that for the same unconditionally stable time-stepping scheme, the oscillatoric critical time-step size of 4-noded elements is larger than that of 8-noded elements.

6.2 PRESCRIBED FLUX BOUNDARY CONDITION

The problem under consideration is the standard problem with a prescribed flux, q_p=1, at x=4. The analytical solution to this problem is presented on page 112 of Carslaw and Jaeger 1959.

Figures 6.4 and 6.5 show the analytical and finite element solutions at two different positions in the body using 8-noded elements.

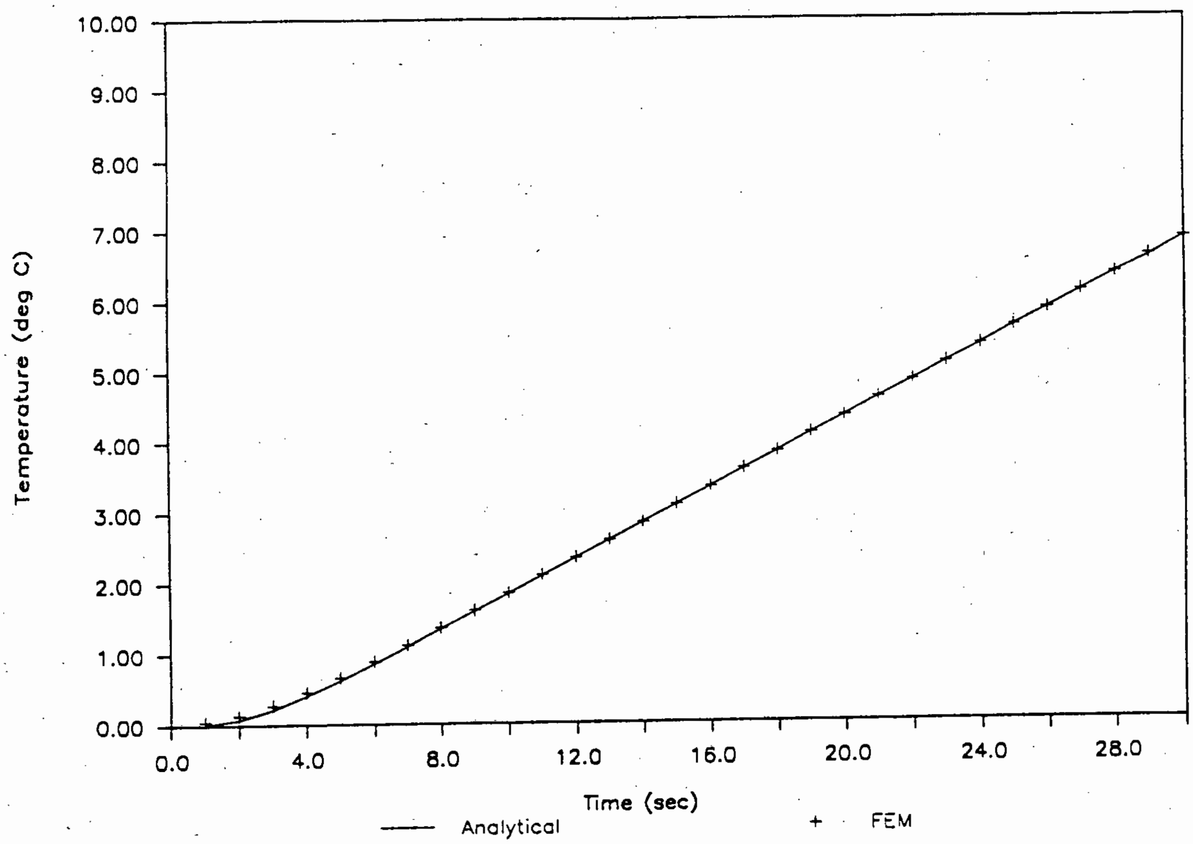


Figure 6.4 *Prescribed flux: analytical and finite element solutions at x=0.*

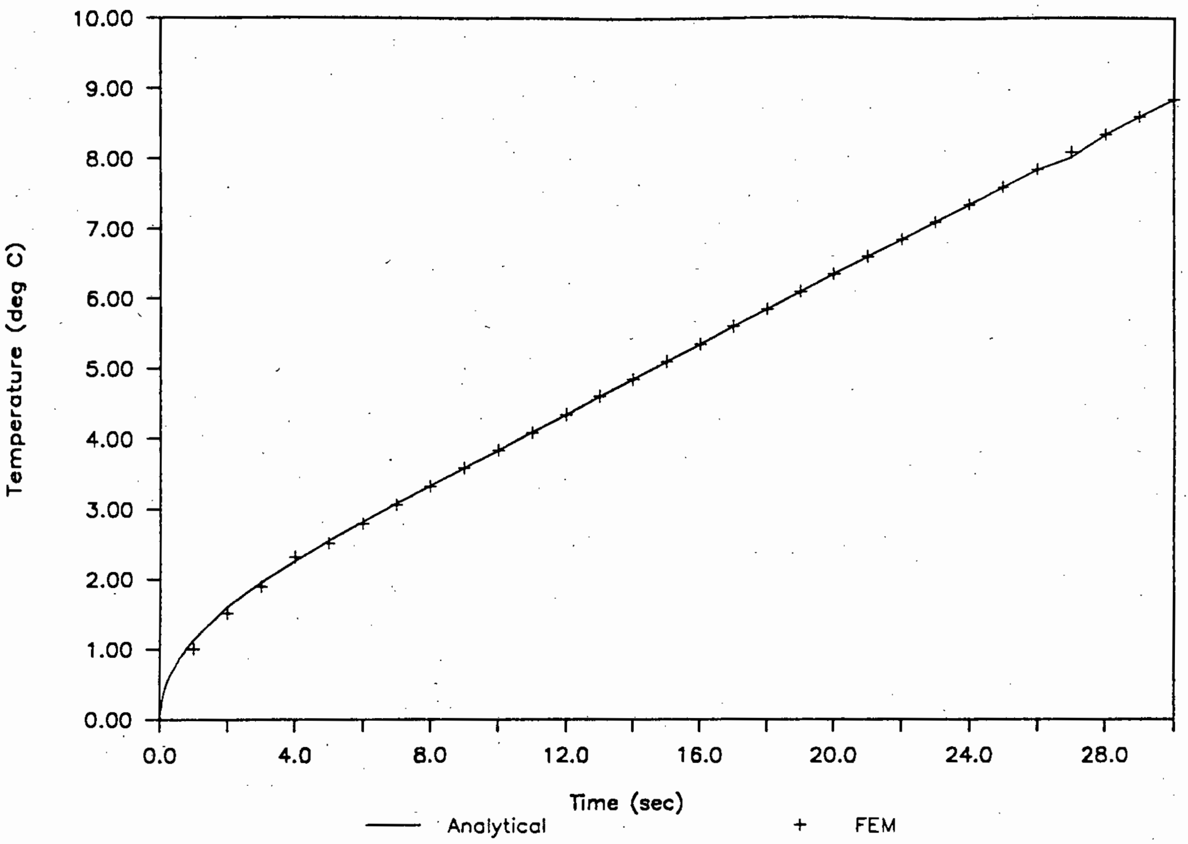


Figure *6.5 Prescribed flux: analytical and finite element solutions at x=4.*

In both cases the finite element solution closely matches the analytical solution except during the first few time-steps of the analysis.

6.3 Convection boundary condition

The problem analysed in this case is the standard problem with a convection boundary condition at x=4 and an initial temperature of 1.0°C instead of 0.0°C. The fluid temperature and heat transfer coefficients are:

$$T_f = 0.0\ ^\circ C$$
$$h = 1.0\ W/m^{2\circ}C$$

The analytical solution to this problem is presented on page 120 of Carslaw and Jaeger 1959.

Figures 6.6 and 6.7 show the analytical and finite element solutions at two positions in the body using 8-noded elements.

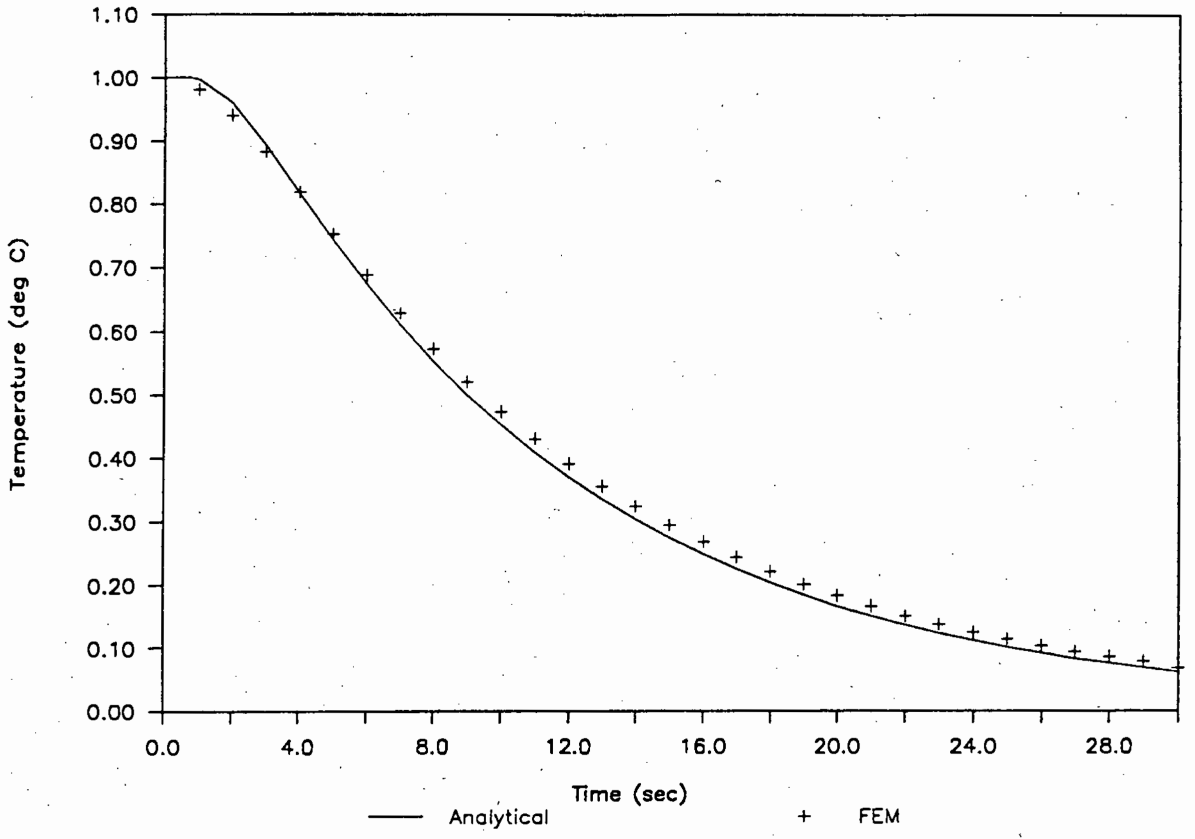


Figure *6.6 Convection: analytical and finite element solutions at x=0.*

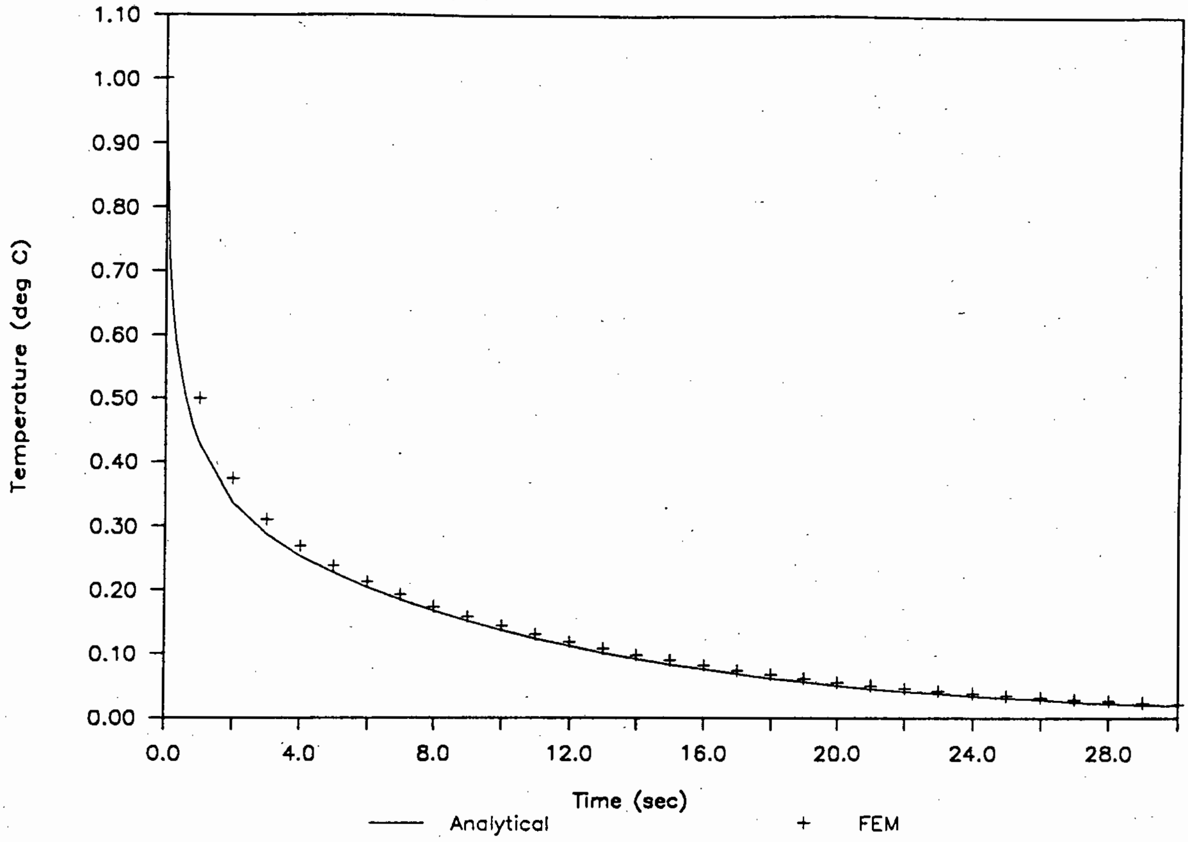


Figure *6.7 Convection: analytical and finite element solutions at x=4.*

Figures 6.6 and 6.7 both indicate that the finite element solution closely approximates the analytical solution. The finite element temperature response shown in figure 6.6 is initially lower than the analytical solution. It then becomes greater than the analytical solution and, subsequently, tends to it. The finite element temperature response depicted in figure 6.7 is always greater than the analytical solution and approaches it more rapidly.

The solutions to a different convection problem are presented in figures 6.8 and 6.9. Although the mesh is the same as that of the standard problem, the following properties and parameters are used:

ρ	=	1.5 kg/m^3
C_p	=	1.3 J/kg°C
k	=	0.75 W/m°C
h	=	2.0 W/m^2°C

T_f	=	2.0 °C	$t < 10$ s
T_f	=	4.0 °C	$t \geq 10$ s
Δt	=	0.1 s	$t \leq 1$ s
Δt	=	1 s	$1 < t < 9.5$ s
Δt	=	0.1 s	$9.5 \leq t \leq 10.5$ s
Δt	=	1 s	$t > 10.5$ s

The analytical solution to this problem can be found on page 127 of Carslaw and Jaeger 1959.

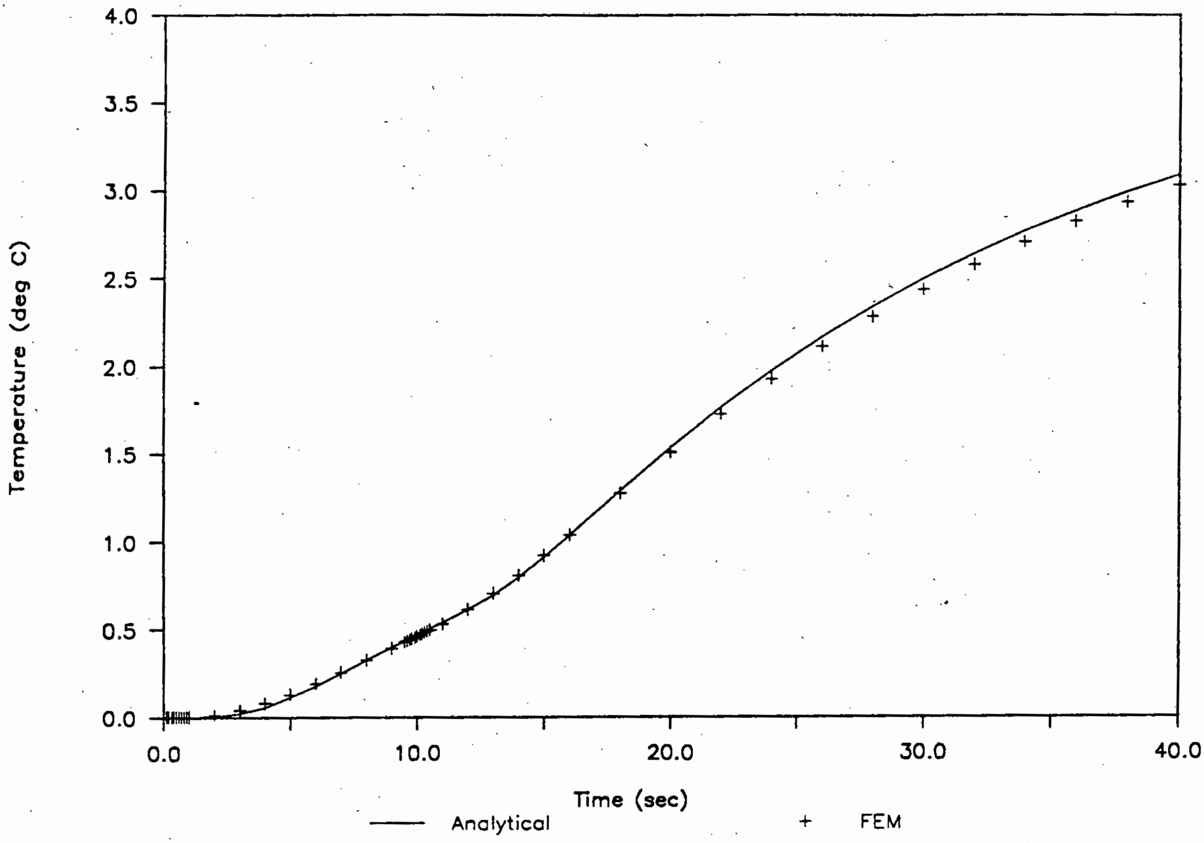


Figure *6.8 Convection: Analytical and finite element solutions at x=0.*

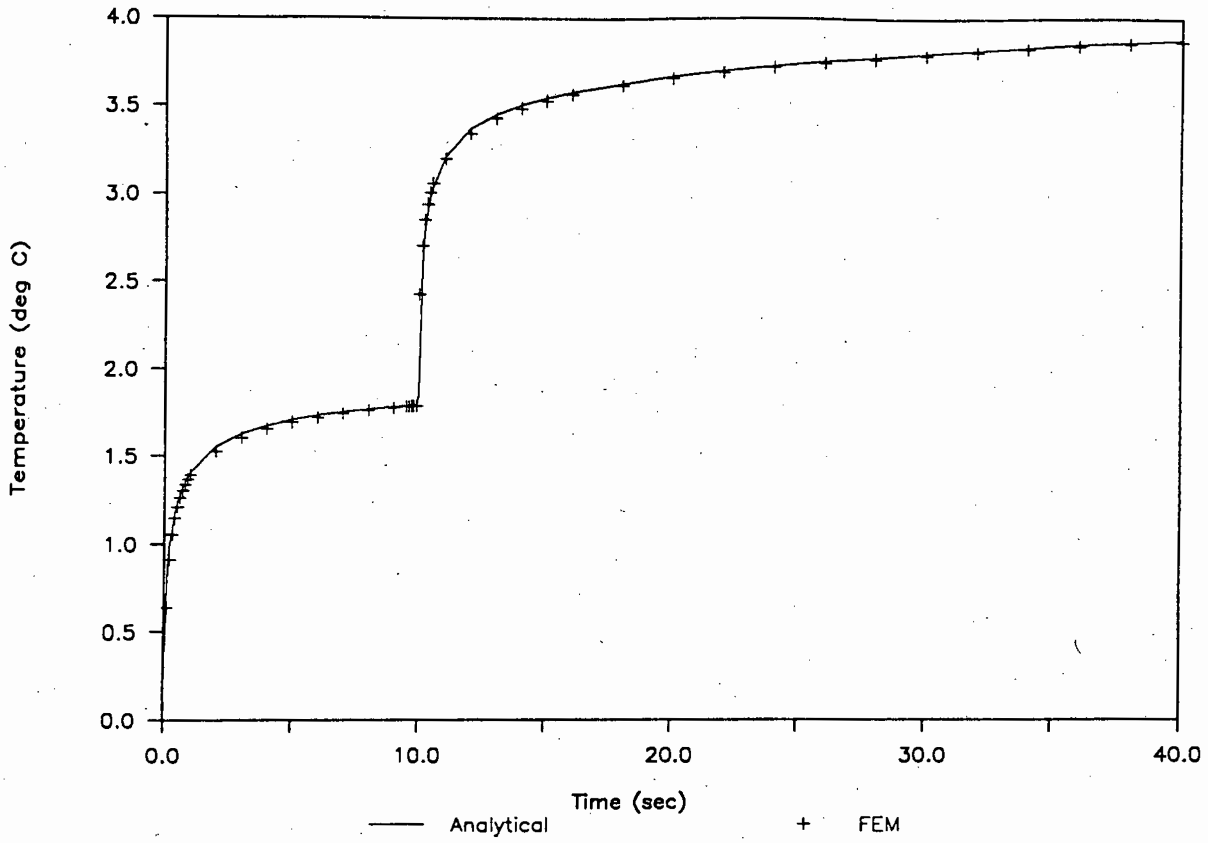


Figure *6.9* *Convection: Analytical and finite element solutions at x=4.*

It can be seen from figures 6.8 and 6.9 that the finite element solution closely approximates the analytical solution, although the approximation is better on the boundary. It is noticeable that the use of smaller time-step sizes both at the beginning of the analysis and when the ambient temperature is increased, significantly improves the finite element solution.

6.4 Internal heat generation

The problem analysed in this case is the same as the standard problem except that there is internal heat generation, Q=1.5 W/m^3, in the bar and the boundary at x=4 is maintained at a fixed temperature, T_p=0.0°C. The time-step size is reduced to 0.1s for the first second of the analysis.

The analytical solution of this problem can be found on page 130 of Carslaw and Jaeger 1959.

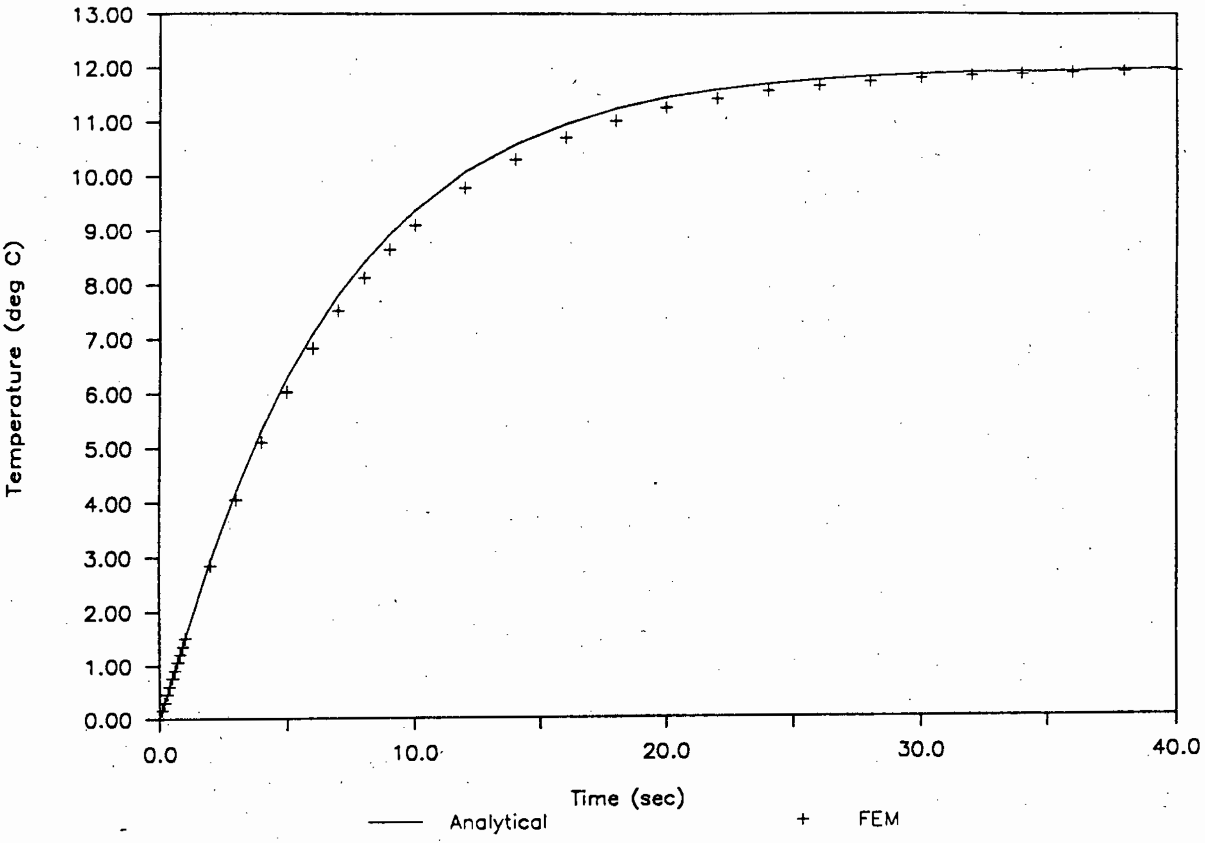


Figure *6.10 Internal heat generation: analytical and finite element solutions at x=0.*

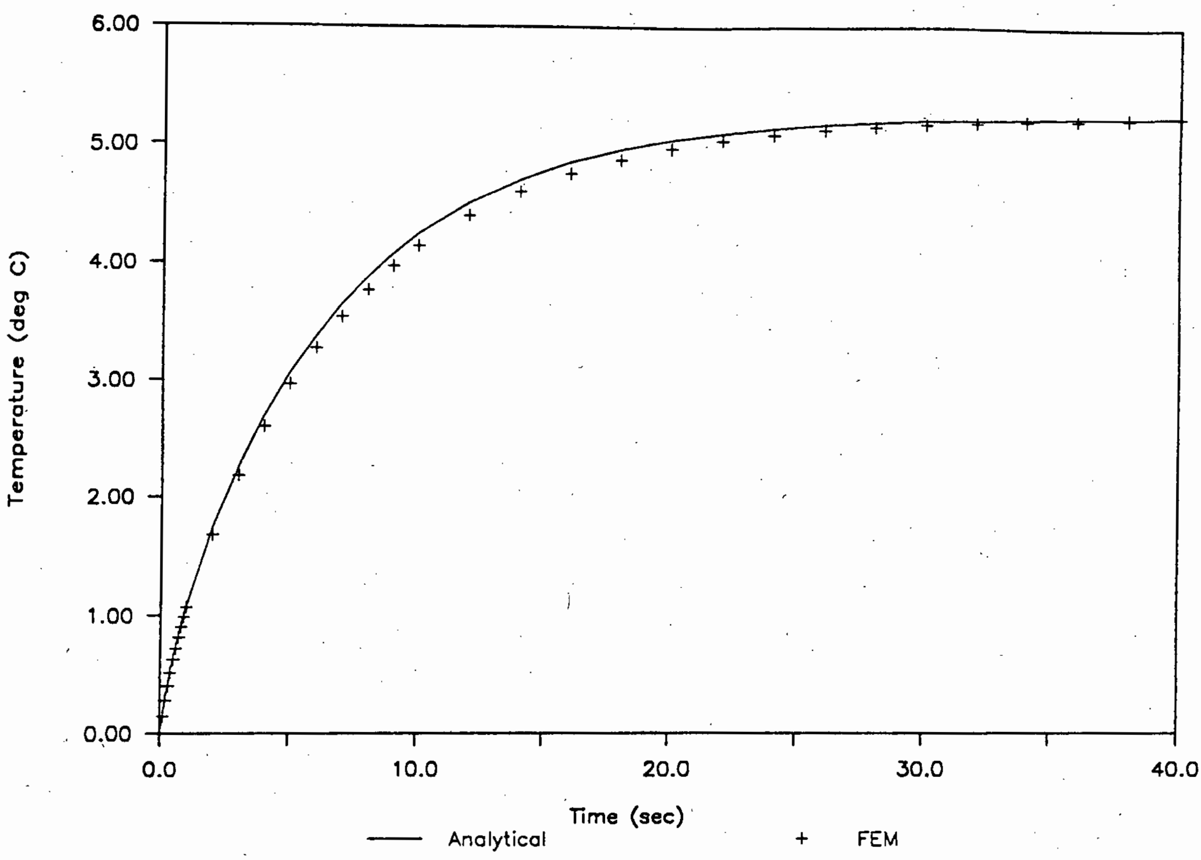


Figure *6.11 Internal heat generation: analytical and finite element solutions at x=3.*

The finite element solutions presented in figures 6.10 and 6.11 both converge to the analytical solution. The small initial time-step size ensures a good initial approximation to the analytical solution.

The solutions to a different internal heat generation problem are presented in figures 6.12 and 6.13. The problem differs from the standard problem in that it has a convection boundary condition on the edge at x=4 and the following different parameters and properties:

ρ	=	1.5	kg/m^3	
C_p	=	1.3	J/kg°C	
k	=	0.75	W/m°C	
Q	=	1.5	W/m^3	
h	=	2.0	W/m^2°C	
T_f	=	0.0	°C	
Δt	=	0.1	s	$t \leq 1$ s
Δt	=	1	s	$t > 1$ s

The analytical solution to this problem is presented on page 132 of Carslaw and Jaeger 1959.

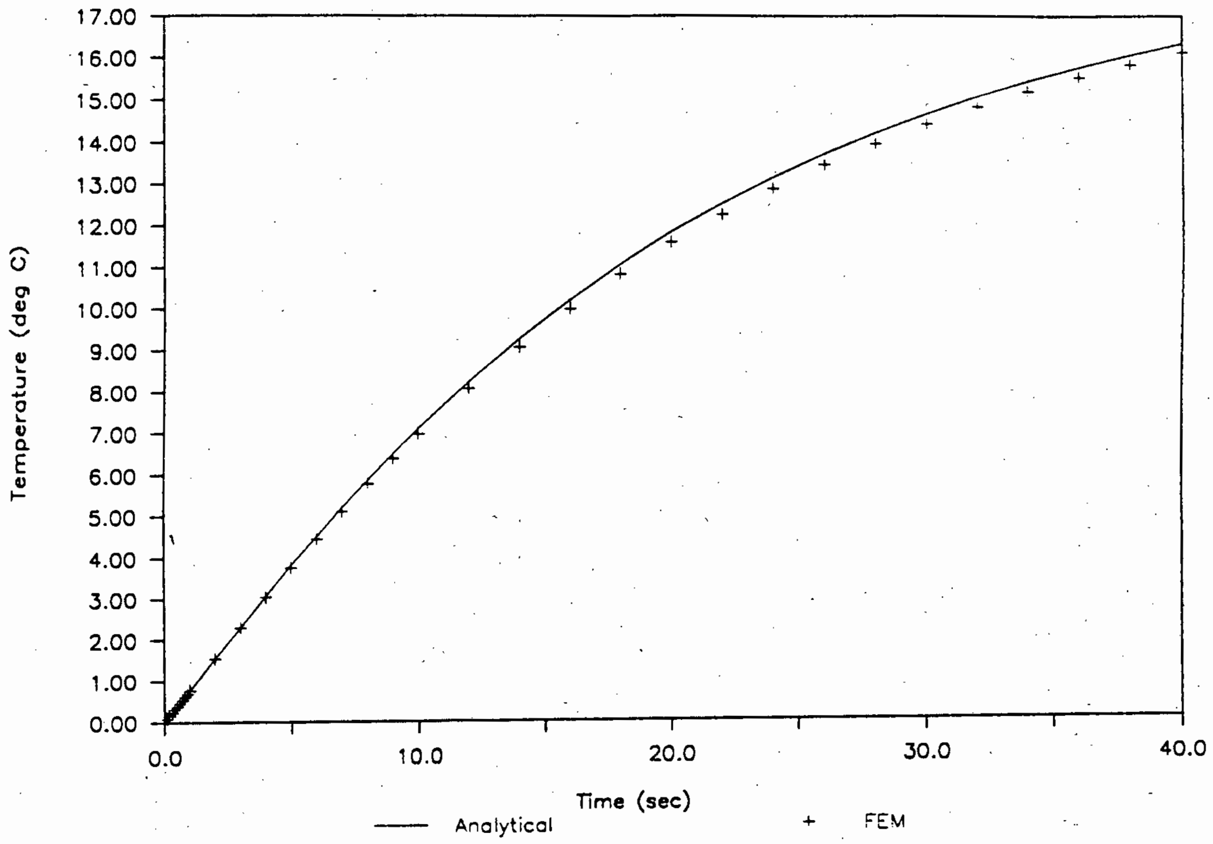


Figure *6.12 Internal heat generation: analytical and finite element solution at x=0.*

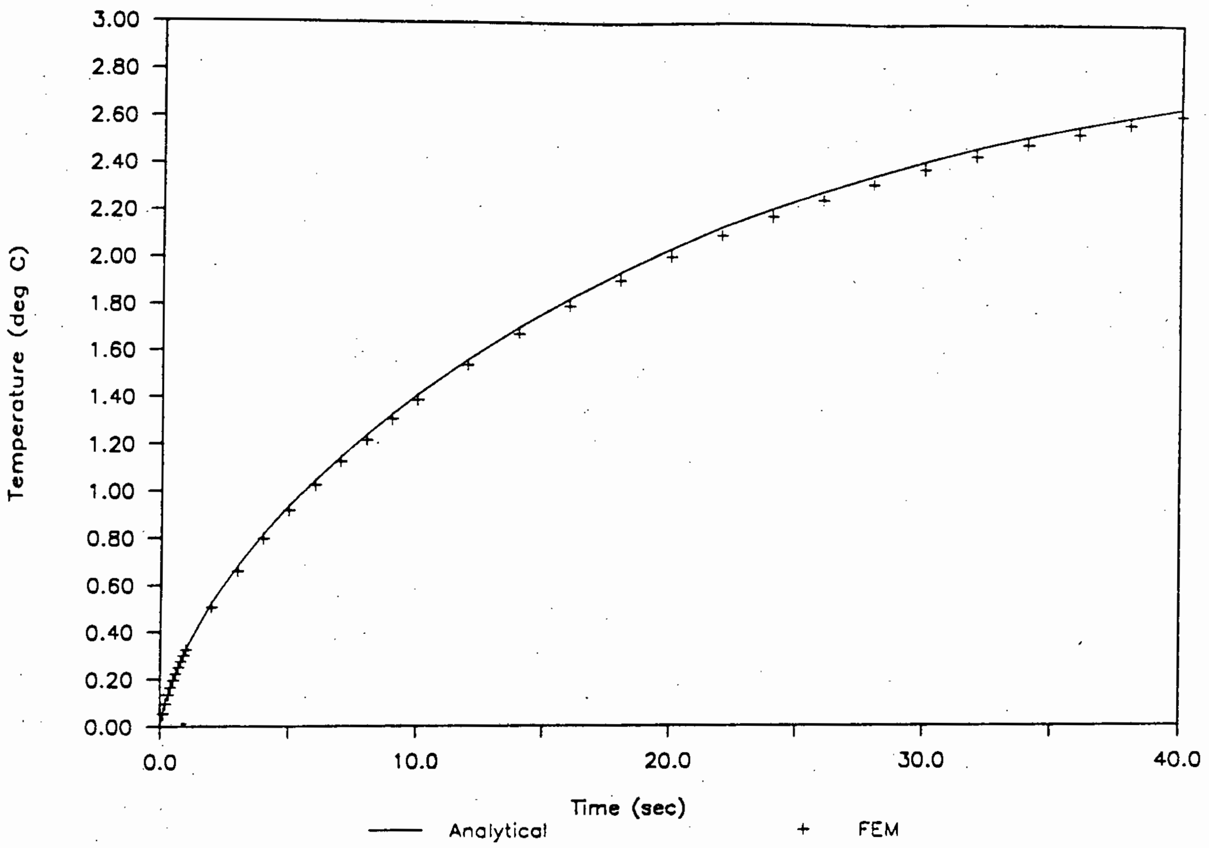


Figure *6.13 Internal heat generation: analytical and finite element solution at x=4.*

Inspection of figures 6.12 and 6.13 indicates that the finite element solution is a good approximation to the analytical solution. It is noteworthy that the solution at the boundary is a better approximation than that at x=0.

6.5 Mass transfer boundary condition

The mass transfer boundary condition problem is completely different from the standard problem described above. Additionally, no analytical or equivalent numerical solution could be found for problems of the type described below.

Figure 6.14 shows the mesh used in the analysis:

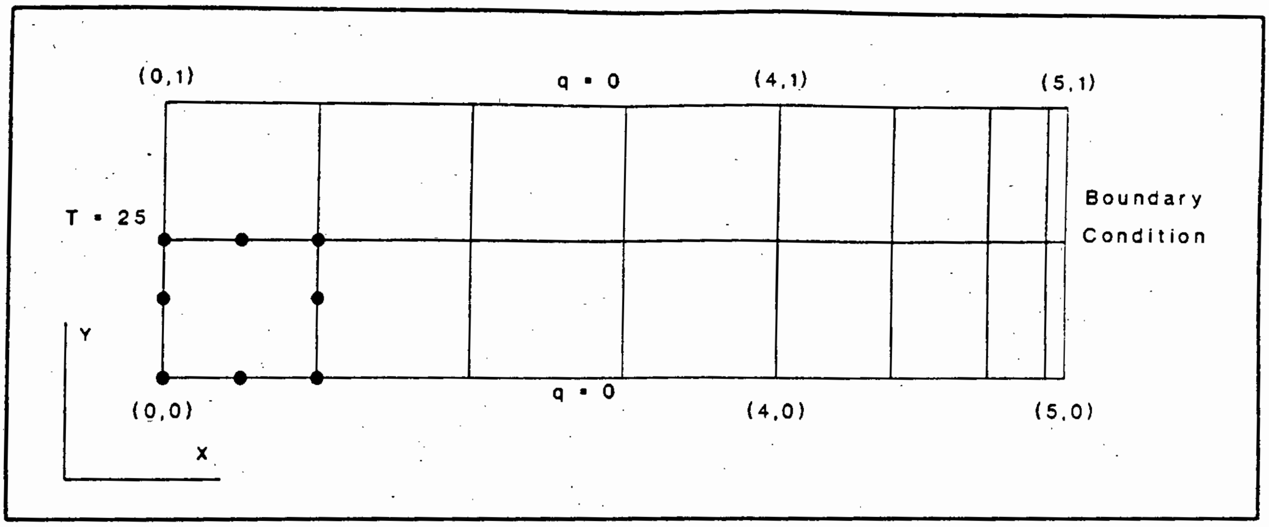


Figure *6.14 Finite element mesh used for analysing a problem with a mass transfer boundary condition.*

The properties and parameters which characterise the problem under consideration are listed below:

<u>Solid properties</u>

ρ	=	1600	kg/m^3
C_p	=	800	J/kg°C
k	=	4	W/m°C
T_o	=	20	°C
Q	=	0.0	W/m^3

<u>Liquid properties</u>

ρ	=	1000	kg/m^3
C_p	=	4184	J/kg°C
k	=	15	W/m°C
μ	=	1.55	cp
β	=	$2 \cdot 10^{-4}$	K^{-1}
D	=	$2.2 \cdot 10^{-5}$	m^2/s
T_{lo}	=	15	°C
l	=	3	mm

<u>Gas properties</u>

ρ	=	2.3	kg/m^3
C_p	=	995	J/kg°C
k	=	$2.6 \cdot 10^{-2}$	W/m°C
μ	=	$1.8 \cdot 10^{-2}$	cp
β	=	0.003	K^{-1}
T_f	=	30	°C
H	=	30	°C
p	=	2	atm

<u>Boundary condition</u>

Prescribed temperature: $T_p = 25$ °C

Prescribed flux: $q_p = 0$ W/m^2 at y=0 and y=1

Mass transfer:
liquid layer at x=5
surface initially wet
convection into liquid
convection between liquid and gas
lumped parameter analysis.

<u>Time-integration</u>

$\alpha = 1$

total analysis time = 10 000s

$0.1\text{s} \le \Delta t \le 2000\text{s}$

As neither analytical nor alternative numerical solutions with which to compare the finite element results were available, the solution to this problem was verified in a qualitative sense by determining limiting case solutions, with the expectation that they would bracket the mass transfer boundary condition solution. The following limiting case problems were considered using the same properties as the problem of interest.

1) convection into liquid at 15 °C
2) convection into gas at 15 °C
3) convection into gas at 30 °C

Considering the temperature of the surface at x=5, one would expect that:

- it would always be less than the temperature resulting from problem 3
- it would always be greater than the temperature resulting from problem 1

- it would initially be less than the temperature resulting from problem 2 but after some time would exceed it.

An examination of the results presented in figures 6.15 and 6.16 shows that the above expectations are confirmed.

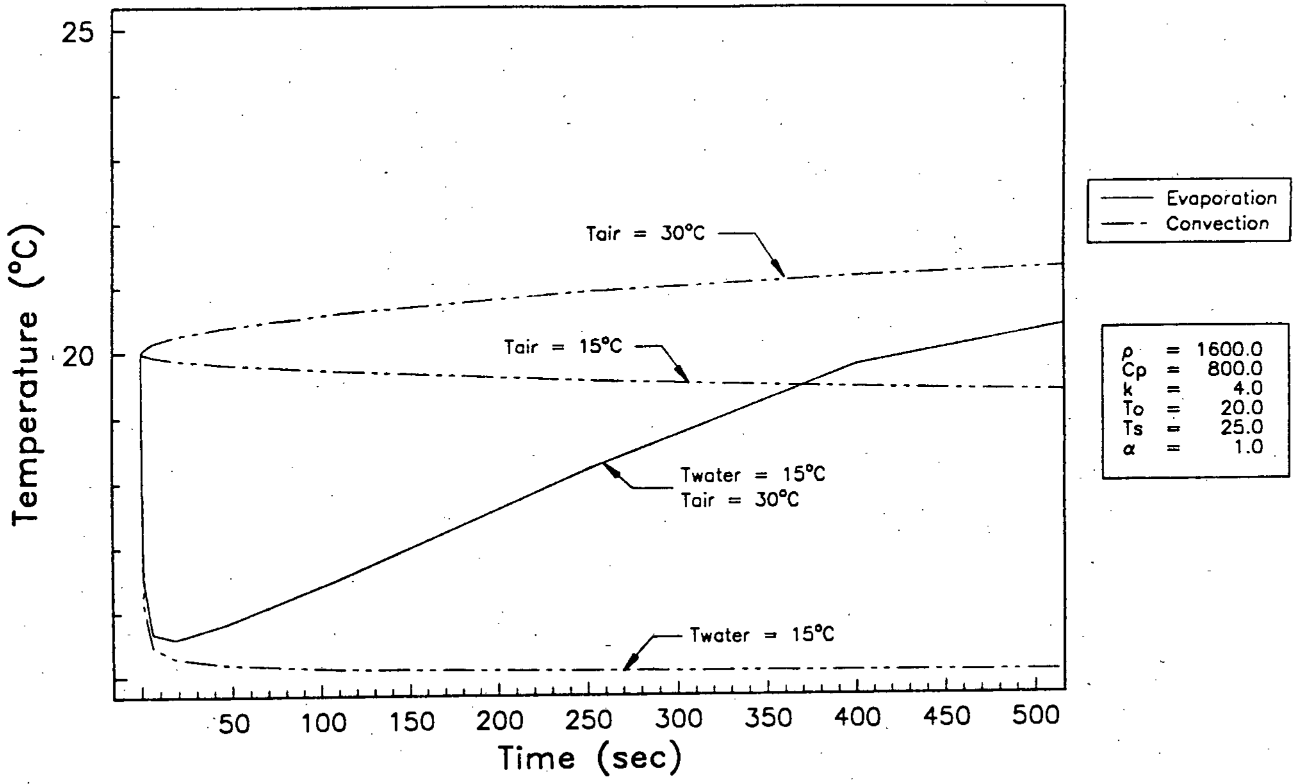


Figure *6.15 Mass transfer boundary condition: actual and limiting case solutions at x=5 over a short time range.*

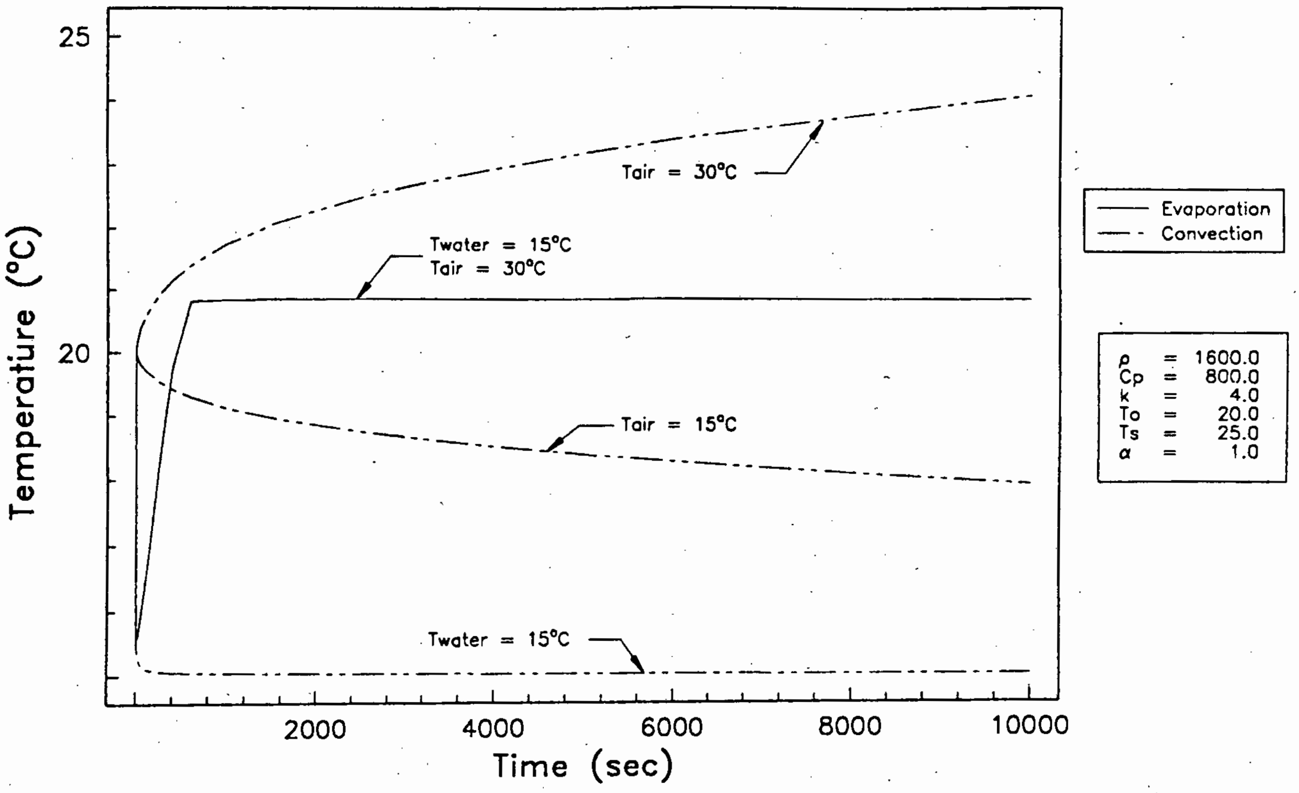


Figure *6.16 Mass transfer boundary condition: actual and limiting case solutions at x=5 over the entire analysis time.*

Although the solutions presented in figures 6.15 and 6.16 lie inside the expected range, there is no confirmation of the accuracy of the results.

6.6 Radiation boundary condition

It has been stated previously that radiation boundary conditions are reduced to pseudo-prescribed flux boundary conditions in this study. In order to show that the radiation boundary condition algorithm is accurate, it therefore suffices to demonstrate that the correct flux entering an element edge is calculated.

As the numerical calculation of shape factors forms a major role in the evaluation of the effects of radiation boundary conditions, results determined using NOTHCAN are compared with analytical results for the simple configurations shown in figure 6.17.

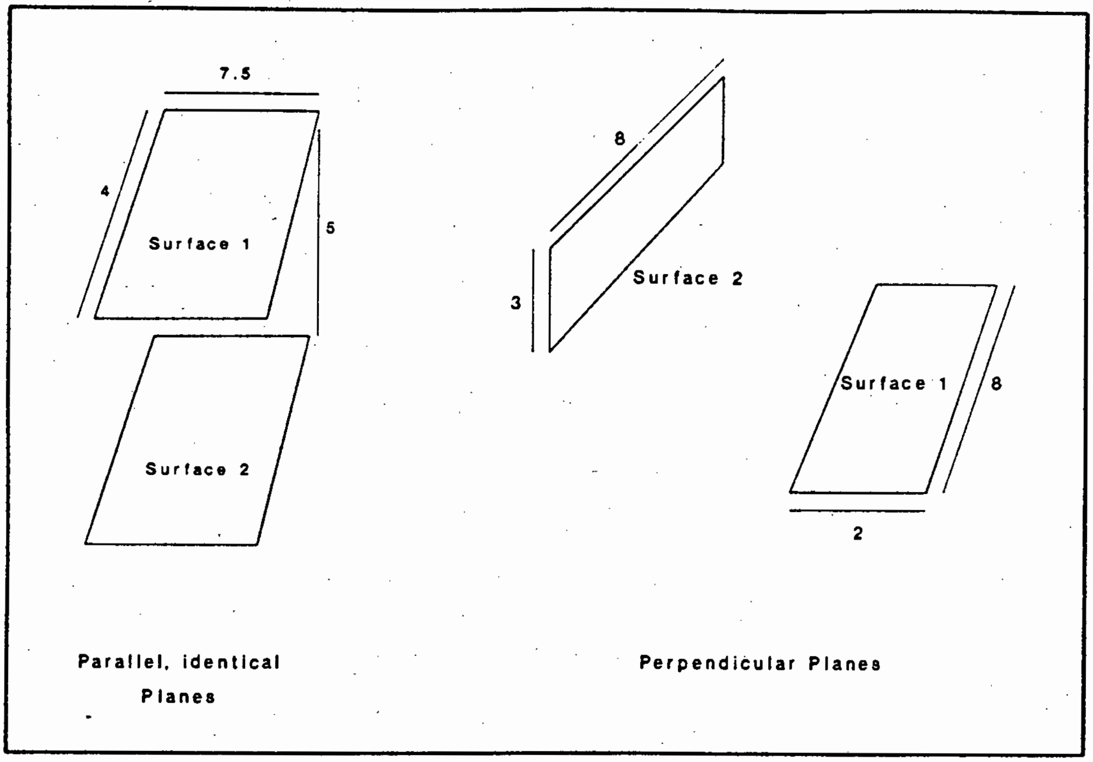


Figure *6.17 Two surface configurations used for shape factor calculation.*

The calculated and analytical shape factor values are presented in table 6.2. The parallel plane configuration is denoted by A while the perpendicular plane configuration is denoted by B.

Table: *6.2 Shape factors (F_{12}) and errors for configurations A and B.*

		Surface Configuration			
		A: F_{12}=0.215269		B: F_{12}=0.0798786	
Subareas	Rays	Value	Error (%)	Value	Error (%)
9	100	0.219377	1.9	0.081129	1.6
256	100	0.21536	0.042	0.079848	-0.04
961	100	0.215256	-0.006	0.079815	-0.08
4900	100	0.215226	-0.02	0.079805	-0.09
4900	12	0.211797	-1.6	0.074991	-6.2
4900	52	0.215088	-0.08	0.079601	-0.35
10000	1000	0.215272	0.0004	0.079879	-0.0006

From table 6.2 it can be seen that the shape factor approximation is satisfactory. Overall, as the number of subareas and rays is increased, the approximate shape factor tends to the analytical value. However, it appears that the solution oscillates as it converges.

In order to test the radiation flux evaluation algorithm, numerical solutions of problems 8.3 and 8.8 in Holman 1976 were compared with the analytical solutions. The problem statement of 8.3 is:

> "Two parallel plates 0.5m by 1.0m are spaced 0.5m apart. Plate 1 is maintained at 1000°C and plate 2 at 500°C. The plates are located in a very large room, the walls of which are maintained at 27°C. The plates exchange heat with each other and with the room, but only the plate surfaces facing each other are to be considered in the analysis. Find the net heat transfer to each plate."

The solutions to this problem are presented in table 6.3:

Table *6.3: Comparison of analytical and numerical solutions of problem 8_3 (Holman 1976).*

	Holman	Numerical	Units
F_{12}	0.285	0.2858	
F_{13}	0.715	0.7142	
q_1	28.85	28.848	kW/m^2
q_2	5.188	5.171	kW/m^2

Problem 8.8 in Holman 1976 was also analysed. It is posed as follows:

> "Two 1m^2 square surfaces are separated by a distance of 1m with T_1=1000K, T_2=400K, ϵ_1=0.8 and ϵ_2=0.5. Evaluate the net heat transfer to the plates when (a) the plates are surrounded by a large room at 300K and (b) the surfaces are connected by a re-radiating wall perfectly insulated on its outer surface."

The solutions to problem 8.8 are presented in table 6.4.

Table *6.4: Comparison of analytical and numerical solutions of problem 8_8 (Holman 1976).*

	Holman		Numerical		Units
	a	b	a	b	
F_{12}	0.2	0.2	0.1998	0.1998	
F_{13}	0.8	0.8	0.8002	0.8002	
q_1	44.184	18.936	44.184	18.920	kW/m^2
q_2	-4.023	-18.936	-4.019	-18.930	kW/m^2

The numerical results presented in tables 6.3 and 6.4 which are generated using the radiation heat transfer techniques and algorithm presented in chapters 2 and 5, are a good approximation to the analytical solution.

6.7 Infinite elements

The problem analysed to verify the implementation of infinite elements is different from the standard problem described earlier. The mesh depicted in figure 6.18 was used in the analysis.

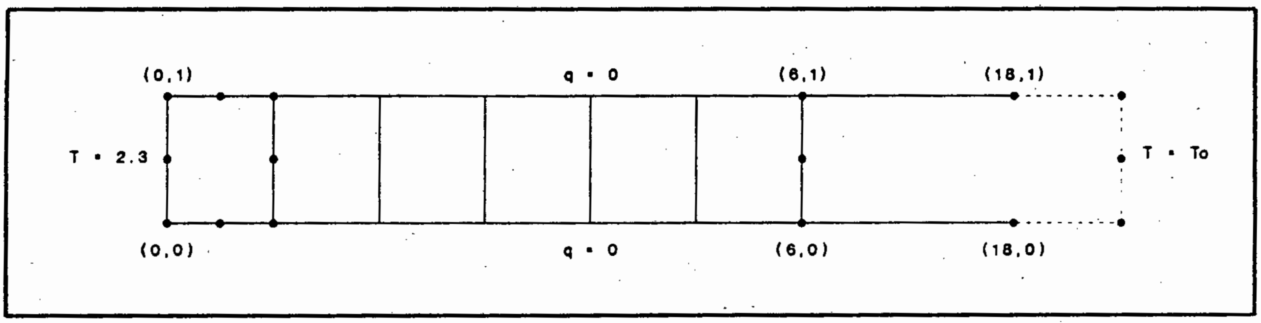


Figure *6.18 Infinite elements: mesh used for the analysis.*

The problem is characterised by the following properties, conditions and parameters:

Solid properties

ρ	=	1600	kg/m^3
C_p	=	1.3	J/kg°C
k	=	0.75	W/m°C
T_o	=	0.0	°C
Q	=	0.0	W/m^3

Boundary conditions

Prescribed temperature: T_p = 2.3 at x=0
T_p = T_o at x=∞
Prescribed flux : q_p = 0 at y=0 and y=1

Time-integration

$\alpha = 1$

Analysis time 100s

$0.1 \leq \Delta t \leq 2$s

The analytical solution to this problem can be found in Carslaw and Jaeger 1959.

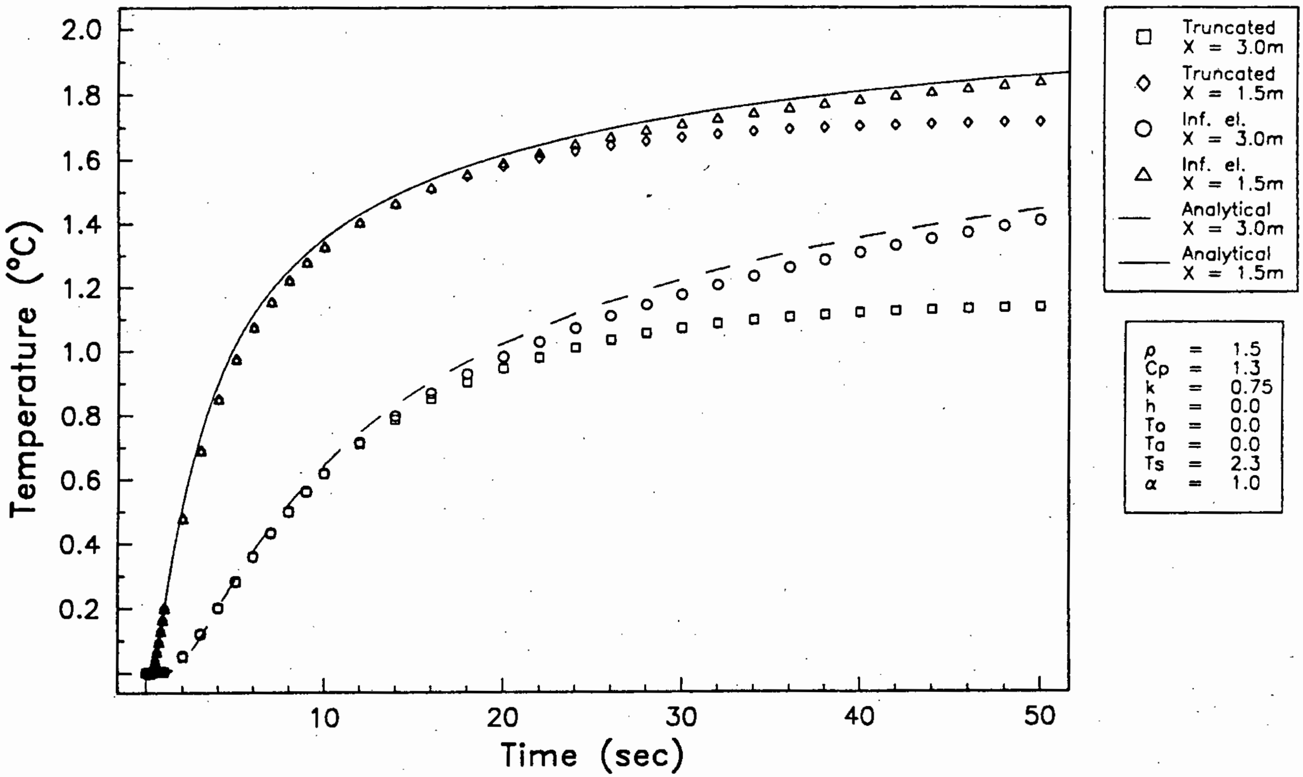


Figure *6.19 Infinite elements: analytical and finite element analysis solutions at x=1.5 and x=3.0.*

Figure 6.19 presents the analytical and finite element analysis results at two positions in the body. In addition, it presents the solution to a problem which is the same as that described above except that the mesh is truncated at x=6.

The results shown in figure 6.19 indicate that the use of infinite elements can generate very good approximations to the analytical solution, approximations which are significantly better than those generated using a truncated mesh (which uses a similar number of elements). Naturally, if the problem is analysed using many more elements in a longer domain, a good approximation can be generated using truncation. However, the use of infinite elements offers an economical and generally satisfactory alternative.

CHAPTER 7 : CONCLUSIONS AND RECOMMENDATIONS

This thesis has involved the development of a computer program which uses the finite element method to analyse transient, nonlinear heat conduction in solids which may have one or more of the following boundary or internal conditions:

1) prescribed temperature
2) prescribed flux
3) internal heat generation
4) point heat source
5) radiation
6) convection
7) evaporation or condensation

Korvink 1986 developed a nonlinear transient finite element analysis program which was incorporated into a structural mechanics analysis program, NOSTRUM, which included the first three conditions listed above. This program was extracted from NOSTRUM and all code and variables not associated with thermal analysis were removed from the program. The resulting thermal analysis program was then expanded to include the rest of the conditions listed above. The program was transferred from the Sperry Univac 1100 on which it was originally written, to a VAX/VMS 6230 computer. The program was then further expanded to include the option of infinite elements for the modelling of unbounded domains.

In chapter 6, examples of problems with these boundary and internal conditions are presented. Except for the mass transfer boundary condition these problems are sufficiently simple to be solved analytically and in all these cases the solutions generated using the finite element method closely match the corresponding analytical solution. As no analytical or other numerical solutions could be found for the mass transfer boundary condition as implemented in the program, a realistic example was presented and qualitatively verified by comparing it with the results of problems which were expected to bracket the mass transfer boundary condition results. The analytical and finite element solutions to a problem with a semi-infinite domain were found to compare favourably. It could be seen that the use of infinite elements generates a superior solution to that generated when a similar but truncated mesh is used.

The examples presented are only a few of those used to test the program during its development. In some of these other problems it was found that instabilities occurred which were not related to time-step size or nonlinearity, but rather to the use of too coarse a mesh in the region of the boundaries. It was found that the application of the penetration depth analysis technique in sizing elements on the boundary, eliminated these instabilities.

Of the requirements for modelling the heat transfer in a typical mine stope, listed in chapter 1, only those of being able to model variable domains and blasting have not been addressed. Variable domains would be modelled using element death and birth techniques and blasting, which involves a very high energy input in a short space of time, probably requires careful numerical implementation to avoid the introduction of instabilities into the analysis.

A user's guide has been compiled for the program which includes some tips for thermal analysis, details of the data which must be specified to analyse a problem, details of the installation of the program on the VAX/VMS 6230 computer and instructions for compiling and running the program. The user's guide is intended to be used in conjunction with this document, a heat transfer text and a finite element analysis text.

It is recommended that the following additions or changes be made to the program:

1) The option of element death or birth should be included to satisfy the requirements for analysing heat conduction in mines. The consequences of modelling blasting as a large point energy source on the boundary should be investigated.

2) In order to reduce the computational effort required to solve problems it would be useful to include an automatic, dynamic time-stepping facility. The time-step length would be governed by such factors as the number of iterations needed to reach the solution in the previous time-step and the current value of the

temperature rate of change.

3) It would be useful to have access to a micro-computer based, graphics-orientated, interactive pre-processor which could be used to capture the data which describes the problem as well as perform data checks on it. The thoroughly checked data file would then be transferred to the mainframe computer which would be used to perform the finite element analysis. Similarly, it would be useful to have a micro-computer based post-processor for analysing results transferred from the mainframe. Features might include temperature, flux and temperature rate versus time plots, temperature contour plots and analysis performance information. All results would be selectively directed to screen, file, printer or plotter.

4) It is essential that some means of quantitatively verifying the program's treatment of problems with mass transfer boundary conditions be derived.

5) It would extend the versatility of the program if three-dimensional problem analysis capabilities were implemented.

6) It would be useful to offer the choice of analysing nonlinear problems using either the substitution method (currently implemented) or the Newton-Raphson scheme, especially for the radiation boundary condition.

REFERENCES

ALEXANDER NA, BLUHM SJ, MARCH TW, BOTTOMLEY P and VON GLEHN F, 'The measurement of heat loads in a deep level stope in the Klerksdorp goldfield', Proceedings of the Mine Ventilation Conference, South African Institute of Mining and Metallurgy, Johannesburg, 1986.

BATHE KJ, Finite element procedures in engineering analysis, Prentice-Hall, New Jersey, 1982.

BETTESS P and BETTESS JA, 'Infinite elements for static problems', Eng. Comput., 1, 1984.

BIRD RB, STEWART WE and LIGHTFOOT EW, Transport phenomena, John Wiley & Sons, 1960.

CARSLAW HS and JAEGER JC, Conduction of heat in solids, 2nd edition, Oxford University Press, 1959.

DAMJANIC F and OWEN DRJ, 'Practical considerations for thermal transient finite element analysis using isoparametric elements', Nucl. Engng. Design, 69, 1982.

DAMJANIC F and OWEN DRJ, 'Mapped infinite elements in transient thermal analysis', Computers and Structures, 19, 673-687, 1984.

DHATT G and TOUZOT G, The finite element method displayed, John Wiley & Sons, 1984.

DJURIC M and NOVAKOVIC M, 'Monte-Carlo application for shape factor determination', Proceedings of the 1st International Conference, (edited by Lewis RW and Morgan K), University College, Swansea, 1979.

DUFFET GA, GRIFFIN TB, MARTIN JB, MERCER CD, REDDY BD and RESENDE L, 'NOSTRUM - A finite element program for nonlinear structural mechanics', Applied Mechanics Research Unit, University of Cape Town, November, 1983.

FRENCH L de F, 'NOTHCAN users' guide', MSc(Eng) Thesis, University of Cape Town, 1990.

HIMMELBLAU DM, Basic principles and calculations in chemical engineering, 4th edition, Prentice-Hall, New Jersey, 1982.

HOLMAN JP, Heat transfer, 4th edition, McGraw-Hill, New York, 1976.

HOTTEL HC and SAROFIM AF, Radiative heat transfer, McGraw-Hill, New York, 1967.

HUEBNER KH, The finite element method for engineers, John Wiley & Sons, 1975.

HUGHES TJR, 'Unconditionally stable algorithms for nonlinear heat conduction', Comp. Meth. Appl. Mech. Engng., 10, 1977.

KERN DQ, Process heat transfer, McGraw-Hill, New York, 1965.

KORVINK JG, 'Transient nonlinear heat transfer using finite elements', MSc(Eng) Thesis, University of Cape Town, 1986.

MARQUES JMMC and OWEN DRJ, 'Infinite elements in quasi-static materially nonlinear problems', Computers and Structures, 18, 739-751, 1984.

MAXWELL GM, BAILEY MJ and GOLDSCHMIDT VW, 'Calculations of the radiation configuration factor using ray casting', Computer-aided Design, 18, 1986.

McADAMS WH, Heat transmission, 3rd edition, McGraw-Hill, New York, 1954.

MITCHELL D and WHILLIER A, 'Cooling power of underground environments', J. S. Afr. Inst. Mining Metall., October, 1971.

OWEN DRJ and DAMJANIC F, 'Reduced numerical integration in thermal transient finite element analysis', J. Comp. Struct., 17, 1983.

PENNINGTON DS, 'Nonlinear heat transfer and thermo-mechanical stress analysis using finite elements', MSc(Eng) Thesis, University of Cape Town, 1985.

PERRY RH and CHILTON CH, Chemical engineers handbook, 5th edition, McGraw-Hill, New York, 1974.

SIEGEL R and HOWELL JR, Thermal Radiation Heat Transfer, McGraw-Hill, New York, 1972.

SMITH JM and VAN NESS HC, Introduction to Chemical Engineering Thermodynamics, McGraw-Hill, Kogakusha, 1982.

VON GLEHN FH and BLUHM SJ, 'The flow of heat from rock in an advancing stope', Internal report, Environmental Engineering Laboratory, Chamber of Mines, Johannesburg, 1987.

ZIENKIEWICZ OC, 'Finite element methods in thermal problems', in Numerical Methods in Heat Transfer, (edited by Lewis RW, Morgan K and Zienkiewicz OC), John Wiley & Sons, 1981.

ZIENKIEWICZ OC, The finite element method, 3rd edition, McGraw-Hill, New York, 1977.

ZIENKIEWICZ OC, BETTESS P, CHAIM TC and EMSON C, 'Numerical methods for unbounded field problems and a new infinite element formulation'. Presented at the Winter Annual Meeting of the American Society of Mechanical Engineers, Washington D.C., November, 1981.

APPENDICES

DETAILS OF COURSES COMPLETED

In order to fulfill the requirements for a MSc(Eng) half dissertation degree, the courses listed below were completed during 1986.

AMA362S	Applied Functional Analysis	3
CIV540F	Finite Element Analysis	3
AMA367F	Continuum Mechanics	1
CIV507B	Theory of Elasticity	2+
AMA364S	Dynamical Systems	2+
AMA368S	Numerical Solution of D.E's	2+

INFINITE ELEMENT MAPPING FUNCTIONS

Marques and Owen (1984) have tabulated the special mapping functions for a variety of infinite elements, both singly and doubly infinite.

Singly infinite elements

This term is chosen to describe all elements which are infinite in a single direction. ξ is chosen to be the infinite and η the finite direction. The shape and mapping functions for the elements described below are explicitly presented in table A1.

6 - Node Lagrangian isoparametric element. This is a 9-node Lagrangian element of which the three nodes corresponding to $\xi = +1$ are at infinity.

5 - Node serendipity isoparametric element. This is an 8-node serendipity element with the same three nodes at infinity as the element described above.

4 - Node isoparametric element. This element is a 6-node Lagrangian element (quadratic in ξ, linear in η) with the two nodes associated with $\xi = +1$ at infinity.

2 - **Node superparametric element.** This is originally a 4-node isoparametric element. When forming an infinite element, with the two nodes corresponding to $\xi = +1$ at infinity, it appears difficult, according to Marques and Owen, to enforce, simultaneously, the singularity at $\xi = +1$ and the invariance requirement under change of coordinate origin. This inconsistency can be overcome by including two extra nodes for mapping purposes only. This results in 4 nodes for mapping and two for field variable modelling, yielding a superparametric element. The mapping functions are the same as those for the element described above and the two shape functions are taken from the 4-node linear finite element.

Table *A1: Singly infinite element mapping and shape functions.*

ELEMENT	MAPPING FUNCTIONS	SHAPE FUNCTIONS
2 NODE SUPER PARAM.	$M_1 = (1-\eta)(\ -\xi)/(1-\xi)$ $M_2 = (1-\eta)(1+\xi)/[2(1-\xi)]$ $M_3 = (1+\eta)(1+\xi)/[2(1-\xi)]$ $M_4 = (1+\eta)(\ -\xi)/(1-\xi)$	$N_1 = (1-\xi)(1-\eta)/4$ $N_4 = (1-\xi)(1+\eta)/4$
4 NODE LAGRAN. ISOPARA.	AS FOR THE 2 NODE ELEMENT	$N_1 = (1-\eta)(\xi^2-\xi)/4$ $N_2 = (1-\eta)(1-\xi^2)/2$ $N_3 = (1+\eta)(1-\xi^2)/2$ $N_4 = (1+\eta)(\xi^2-\xi)/4$
5 NODE SEREND. ISOPARA.	$M_1 = (1-\eta)(-1-\eta-\xi)/(1-\xi)$ $M_2 = (1-\eta)(1+\xi)\quad /[2(1-\xi)]$ $M_3 = (1+\eta)(1+\xi)\quad /[2(1-\xi)]$ $M_4 = (1+\eta)(-1+\eta-\xi)/(1-\xi)$ $M_5 = (1-\eta^2)\quad 2\quad /(1-\xi)$	$N_1 = (1-\eta)(1-\xi)(-1-\eta-\xi)/4$ $N_2 = (1-\eta)(1-\xi^2)\quad /2$ $N_3 = (1+\eta)(1-\xi^2)\quad /2$ $N_4 = (1+\eta)(1-\xi)(-1+\eta-\xi)/4$ $N_5 = (1-\eta^2)(1-\xi)\quad /2$
6 NODE LAGRAN. ISOPARA.	$M_1 = (\eta^2-\eta)(\ -\xi)/(1-\xi)$ $M_2 = (\eta^2-\eta)(1+\xi)/[2(1-\xi)]$ $M_3 = (1-\eta^2)(1+\xi)/(1-\xi)$ $M_4 = (\eta^2+\eta)(1+\xi)/[2(1-\xi)]$ $M_5 = (\eta^2+\eta)(\ -\xi)/(1-\xi)$ $M_6 = (1-\eta^2)(-2\xi)/(1-\xi)$	$N_1 = (\eta^2-\eta)(\xi^2-\xi)/4$ $N_2 = (\eta^2-\eta)(1-\xi^2)/2$ $N_3 = (1-\eta^2)(1-\xi^2)$ $N_4 = (\eta^2+\eta)(1-\xi^2)/2$ $N_5 = (\eta^2+\eta)(\xi^2-\xi)/4$ $N_6 = (1-\eta^2)(\xi^2-\xi)/2$

Doubly infinite elements

These elements tend to infinity in two directions. Table A2 shows the mapping and shape functions for the doubly infinite elements described below.

4 - Node Lagrangian isoparametric element. This is a 9-node Lagrangian element which extends to infinity in both directions, with the five nodes corresponding to $\xi = +1$ and $\eta = +1$ at infinity.

3 - Node serendipity isoparametric element. This is an 8-node element which extends to infinity in both directions.

1 - Node superparametric element. This is a 4-node Lagrangian element which extends to infinity in two directions. Three additional nodes are required to define the mapping functions for the same reasons presented for the singly infinite case.

Table *A2: Doubly infinite element mapping and shape functions.*

ELEMENT	MAPPING FUNCTIONS	SHAPE FUNCTIONS
1 NODE	$M_1 = 4\eta\xi/[(1-\eta)(1-\xi)]$	$N_1 = (1-\xi)(1-\eta)/4$
SUPER	$M_2 = (1+\eta)(-2\xi)/[(1-\eta)(1-\xi)]$	
PARAM.	$M_3 = (1+\eta)(1+\xi)/[(1-\eta)(1-\xi)]$	
	$M_4 = (-2\eta)(1+\xi)/[(1-\eta)(1-\xi)]$	
3 NODE	$M_1 = [\eta\xi+3(-1-\eta-\xi)]/[(1-\eta)(1-\xi)]$	$N_1 = (1-\xi)(1-\eta)(-1-\xi-\eta)/4$
SEREND.	$M_2 = 2(1+\eta)\ /[(1-\eta)(1-\xi)]$	$N_2 = (1-\xi^2)(1-\eta)\ /2$
ISOPARA.	$M_3 = 2(1+\xi)\ /[(1-\eta)(1-\xi)]$	$N_3 = (1-\xi)(1-\eta^2)\ /2$
6 NODE		$N_1 = (\xi^2-\xi)(\eta^2-\eta)/4$
LAGRAN.	AS FOR THE 1	$N_2 = (1-\xi^2)(\eta^2-\eta)/2$
ISOPARA.	NODE ELEMENT	$N_3 = (1-\xi^2)(1-\eta^2)$
		$N_4 = (\xi^2-\xi)(1-\eta^2)/2$

NOTHCAN

USER'S GUIDE

A Finite Element Program for the Nonlinear,

Transient Anaysis of Heat Conduction

in Two-dimensional Continua

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1. INTRODUCTION

This document provides information which explains how to use a program for the analysis of transient, nonlinear, two dimensional heat conduction in solids. The program is called NOTHCAN for convenience. This name is a quasi-acronym derived from the following program description: NOnlinear Transient Heat Conduction ANalysis. The program uses the finite element method of analysis to solve the heat transfer equation. As the name indicates, NOTHCAN can analyse problems which are nonlinear, but the nonlinearity must be in temperature not in space. For example, any physical properties specified by the user may be functions of temperature and/or of time but not of position. NOTHCAN can also successfully analyse both transient and steady state problems. It is not capable of modelling domains which change phase and it treats all other modes of heat transfer such as convection and radiation as boundary conditions. The program is designed to model two dimensional geometries only, but is capable of modelling one dimensional domains as well by using a symmetrical element mesh combined with appropriate insulation boundary conditions.

The current version of the program is based on the work of two UCT graduate students, D. Pennington and J. Korvink. The former wrote a steady state heat conduction analysis finite element method program which formed a subset of a larger structural analysis code called NOSTRUM. This was to be used in thermal stress analysis. The latter re-structured and extended this program so that it could perform transient analysis for the determination of both thermal stresses in structures and stand-alone heat conduction analysis. Both of these codes followed the structure of NOSTRUM closely and used NOSTRUM subroutines some of which had to be modified. At present the code of NOTHCAN is completely independent of the parent NOSTRUM while retaining many remnants of it in its structure and variable names. However it cannot now form a subset of NOSTRUM without significant changes to both programs.

This users' manual is in turn largely based on a condensed version of the NOSTRUM users' manual. This explains the sometimes unusual order in which the variables must be input. The manual is divided into a number of sections.

- Introduction
- Thermal analysis considerations
- Description and overview of data categories
- Data input
    - Problem and Data File Description
    - Analysis and Mesh Control Parameters
    - Transient and Nonlinear Analysis Control Parameters
    - Output Control Parameters
    - Nodal Coordinate Specification
    - Element Specification
    - Initial Condition Specification
    - Boundary Condition Specification
    - Time Functions Definition
- Using NOTHCAN
- Appendix

A user who is not familiar with this manual is strongly advised to read it thoroughly before attempting an analysis. All users are advised to use this manual together with a book on the finite element method and its application to heat transfer as well as a heat transfer text.

2. THERMAL ANALYSIS CONSIDERATIONS

In order to put the use of this program into some perspective it is useful to consider some of the stages in the analysis of a typical problem using NOTHCAN.

In its original form, be it a text-book problem statement or a real-life situation, a problem to be analysed is generally not immediately analysable by the program. To remedy this situation a problem statement must be restructured in the context of the method of analysis and its requisite data input. In the case of NOTHCAN this involves creating a data file containing information which describes the problem under consideration in accordance with the 'syntax' presented in the 'Data Input' section of this manual. Once this data file has been correctly created, the program can read the information and proceed with the analysis. The results generated during the analysis are stored in an output file which can subsequently be examined and manipulated at leisure. So, the analysis of a problem can be divided into three stages:

- expression of the original problem in the context of the method of analysis
- -analysis of the problem
- interpretation of the results

This users' guide is concerned with the first stage and so only this is discussed in detail. The expression of the original problem in the context of the finite element method involves creating a problem data file the structure of which is described in the 'Data Input' section of this users' manual. The data file is created by starting at the first data set of the 'Data Input' section, filling in appropriate values for the variables of that data set and then proceeding to the next data set. This process must be repeated until the last data set has been completed. To avoid unnecessary error and confusion when analysing a problem, the user is advised to follow the steps listed below before attempting to create the data file.

- establish what boundary conditions are present

- establish if any boundaries occur at infinity
- establish the nature of the temperature dependency of the physical properties
- locate areas which are likely to have high temperature gradients and rates

Having done the above and having read the relevant sections in the 'Data Input' section of this manual, the user should be able to define the linear or nonlinear nature of the problem and plan the time integration strategy and scheme to use.

Perhaps the most crucial aspect of expressing the original problem in terms of the finite element method is the discretisation process or the division of the domain into sub-domains. If the user is not familiar with the elements offered by the program, he should refer to the relevant section now. There are no fixed rules for dividing a domain into elements but the following three generalised guidelines should be used:

- elements should have sides which are as close to straight lines as possible
- areas of high temperature gradients should be the densest regions of the mesh
- there should not be fewer than four elements in a mesh

One of the main difficulties of creating the finite element mesh is the selection of element size and number. This process is largely a matter of experience. There is fortunately some aid available for use in the sizing of elements on boundaries which can be used to avoid the introduction of instabilities due to the use of elements which are too big. The means of doing this is based on penetration depth analysis and an explanation follows.

Consider transient, one dimensional heat conduction in a body. Assume that either a prescribed temperature or a prescribed flux boundary condition is applied at one end. After a short time, Δt, it is reasonable to assume that the bulk of the body will still be at the initial temperature and that the temperature conditions of a small portion of the body will have changed. The width of this portion - the penetration depth - can be approximated by assuming a quadratic temperature profile in it - one end at the boundary temperature and the other end at the initial temperature. Using this assumption, the heat equation can be solved to yield an expression for the penetration depth.

The following expressions can be derived:

$$\delta(\Delta t) = (12\alpha\Delta t)^{0.5} \quad \text{- prescribed temperature}$$
$$\delta(\Delta t) = (6\alpha\Delta t)^{0.5} \quad \text{- prescribed flux}$$

where δ = penetration depth (m)

Δt = initial time-step size (s)

α = thermal diffusivity : $k/\rho C_p$ (m^2/s)

It is clear that the penetration depth is dependent on the initial time-step size.

In order to obtain a reasonable approximation to the new temperature distribution in the body during the first few time-steps it is essential that the width of at least the first element is less than or equal to the penetration depth.

This method can only be used to size an element in one dimension at a time. However, it is a simple matter to size an element in both dimensions if it is required.

As all the boundary conditions which the program is capable of modelling - excepting prescribed temperature boundary conditions - are flux type boundary conditions, the prescribed flux boundary condition penetration depth equation can be generally used.

Appropriate use of the concept of penetration depth when sizing the boundary elements results in a significant improvement in the quality of results generated for the earlier time-steps in any analysis.

Once the mesh has been defined the user should begin to create the data file. Once this has been completed the problem is ready to be analysed by executing the program which will generate results and store them in an output file. As the results in the output file are in the form of long lists of numbers - temperatures and node numbers for various iteration numbers in each time-step - it simplifies the interpretation, analysis and presentation, of the results to select those nodal points which are regarded as being significant and examining their temperature-time response graphically. Once this has been done it is possible to check for anomalies and inconsistencies in the results.

It is often advisable to repeat an analysis a number of times using different analysis control parameters to verify that the calculated solutions have in fact converged.

3. DESCRIPTION AND OVERVIEW OF DATA CATEGORIES

The various values entered in a data file to enable an analysis of a heat transfer problem can be divided into two categories. The first category is made up of data which describes and defines the problem to be modelled and the second is made up of data which controls the way in which the program analyses the problem.

The first-mentioned data category includes the number of nodes in the finite element mesh used to represent the real domain, the number of element groups into which this mesh is broken up, the number and type of elements in each group, the coordinates of the nodes of the finite element mesh, the duration of the analysis, the specification of which types of heat transfer occur and which elements and groups of nodes have one or more of the various boundary conditions available applied to them. Initial conditions and details of the various internal and boundary conditions such as internal heat generation, convection and radiation are also specified. These might include fluxes, temperatures and fluid flow rates among others.

The last-mentioned data sub-category includes information specifying whether or not data checks only should be carried out or both data checks and finite element analysis. It includes information indicating whether the analysis is transient or steady state and linear or nonlinear, the time integration scheme to be used, the time-step size (which is variable), parameters to control nonlinear analysis - stiffness reformation (currently unused), convergence tolerances and over-relaxation factors. It includes parameters which control the output of results such as which results are to be printed - temperatures, fluxes or temperature rates, ranges of time-steps in which results are to be printed, how many iterations must occur between successive output of results, for which nodes results are to be printed, whether specified results are to be directed to a post-processing file and whether a time log of the various steps in the analysis is to be maintained.

As a loosely defined rule of thumb it can be stated that analysis control variables are generally integers while data which define the problem being analysed are more often real numbers. It should be clear from the format of this users' guide that the two categories of data are freely intermixed. This is done to preserve both logical grouping of related information and to maintain a form of data hierarchy.

It must be emphasised that it is possible for the program's error checking routines to miss even obvious errors and this necessitates extreme care in data specification. The program does, however, trap type mismatches in the input data and the data is echoed until the point where the type mismatch occurred. This simplifies the location of errors due to incorrect inclusion or omission of data. Errors can often be avoided if the following conventions are closely followed.

3.1 CONVENTIONS

The following conventions are used by NOTHCAN in terms of both input format and system of units.

Generally the format of values specified in the data file must be type sensitive and be separated by some delimiter - a space or a comma. Data input can be in explicit or exponential format. For example

reals	:	2.0 3.0 4.0 3600.0
		2.0,3.0,4.0,3.6E3
integers	:	2 5 19 12 5000
		2,5,19,12,5E3

are all acceptable forms of input. The type of the data to be specified is determined according to the FORTRAN variable naming convention - all variable names beginning with the letters I through N inclusive are integers while those starting with any other letter are reals.

Any strings which are to be entered must be upper case and must start in the first column of a line.

All physical properties used in an analysis must be in SI units. There is no facility included in the program for converting between SI and cgs systems, for example. The following is a list of the properties used by the program together with their units.

Physical properties	Units
Density	kg/m^3
Diffusivity	m^2/s
Distributed fluxes	W/m^2
Heat capacity	J/kg.°C
Heat transfer coefficient	W/m^2.°C
Internal heat generation	W/m^3
Latent heat	J/kg
Point heat sources	W
Pressure	atm
Relative humidity	%
Temperature	°C
Thermal coefficient of volume expansivity	K^{-1}
Thermal conductivity	W/m.°C
Time	user specified
Velocity	m/s
Viscosity	kg/m.s

Many of the data values which must be specified are switches which indicate whether or not certain options are to be included. The conventions used in this regard are as follows:

0 : (zero) the option is not required or is 'off'

1 : (one) the option is required or is 'on'

This convention applies throughout unless otherwise specified.

Energy entering a body being analysed is defined as positive and energy leaving is defined as negative.

When creating a data file or a data file template, it is often desirable to label the variables with names corresponding to those in the user guide. This can be done by prefixing any line which is not to be regarded as input data with "C**" in the first column. The program regards any line beginning with this string as a comment.

Whenever the user enters a value in response to a question which has the meaning 'How many of this particular entity are there?' the user must be aware of the existence of bounds on the magnitude of certain parameters. For example, a mesh can currently have a maximum of 100 nodes. If the user specifies more than 100 nodes then the analysis will abort. A list of the current limits is included in the appendix. These bounds are necessary due to the lack of dynamic dimensioning in standard Fortran 77. It is possible, however, for the user to specify new values of these limits and then to recompile and relink the program. Refer to the 'Using NOTHCAN' section for the details of this procedure.

4. DATA INPUT

4.1 Problem and data file description

The first entry in the data file is the problem title and it must start in the first column of the first line and may be an alphanumeric string with a maximum length of 72 characters. This string is mandatory. The user can of course leave the first line of the data file blank if he does not want a title to be printed.

There is an option to echo the input data file as it exists in storage. This option is on by default and can be switched off by including the string 'ECHO' starting in the first column of the second line of the data file. If the string 'ECHO' is omitted, then the data file will be reproduced in the output.

It is useful for the enhancement of the readability of the data file to be able to include comments. This is done by adding the prefix 'C**', starting in the first column of a line, to any desired comment string. This prefix may be used for the first time after the 'ECHO', if it is present. If the 'ECHO' is not present, the comment prefix can only be used for the first time after the problem title string

PROBLEM TITLE

Data set 1

Problem title string

ITITLE

ITITLE an alphanumeric string which describes the problem to be analysed. It may occupy columns 1 to 72 inclusive and is obligatory.

DATA FILE OUTPUT CONTROL

Data set 2

Data echo indicator

ECHO

ECHO the program echoes the input data at the beginning of an analysis unless this string is included immediately after the title string.

DATA FILE COMMENT PREFIX

Data set 3

Comment indicator

```
C**
```

C** this character string must be entered starting in column 1 wherever a comment statement is included in the data file. It cannot be used before the title or the 'ECHO' string, however.

4.2 Analysis and mesh control parameters

The program has a number of modes of execution. It can ,in the first mode, read the data file and carry out some rudimentary checks on the values read. It can, in the second mode, do this as well as perform a finite element analysis. The third mode although not currently active is included with a subsequent version of the program in mind, and entails performing a data check followed by a finite element analysis with an option of restart analysis.

The problem to be analysed can be specified to be steady state or transient and linear or nonlinear.

Care must be taken in the specification of these parameters as the program does not check them. It is recommended for the first run of any analysis that the data be verified by the program before a finite element analysis is commenced.

A particular problem is definitely nonlinear in the following cases:

- if any of the physical properties of the body are functions of temperature
- if any of the physical properties used in the evaluation of boundary conditions are functions of temperature
- if the values of the boundary conditions imposed are dependent on temperature - for instance a prescribed distributed flux or internal heat generation value
- if there is a radiation boundary condition
- if there is a mass transfer boundary condition

If the user is unsure of the degree of linearity of an analysis, it is best to assume that it is nonlinear.

ANALYSIS CONTROL

Data set 4

Analysis control parameters

```
MODEX  INDEX
```

MODEX mode of execution

- 0 the data file is read and checked; no analysis is carried out.
- 1 analysis without restart
- 2 restart analysis

INDEX nature of the analysis for this run

- 1 nonlinear steady state analysis
- 2 nonlinear transient analysis
- 3 linear steady state analysis
- 4 linear transient analysis

In the following data set the total number of nodes in the finite element mesh used to model the domain must be entered. The user must be aware of the existence of bounds on the magnitude of some variables. The number of element groups present must also be specified. The user can refer to the 'Element Specification' section for a detailed description of element groups.

The number of nodes with either a prescribed temperature or temperature rate must also be specified. In many cases there are no nodes with prescribed temperature rates. In addition, the total number of time functions used must be specified. Time functions are explained in the 'Time Function Definition' section.

MESH DEFINITION

Data set 5

Structural model parameters

NPOIN NEGRP NVTEM NVTDT NTFUN

NPOIN total number of nodes in the mesh

NEGRP total number of element groups

NVTEM total number of nodes with prescribed temperatures

NVTDT total number of nodes with prescribed temperature rates

NTFUN total number of time functions

The following data set is used to specify which types of internal and boundary conditions occur in the problem under consideration. Except for the mass transfer boundary condition each condition is often used in standard heat transfer analysis and is discussed in greater detail in the subsequent sections dealing with boundary and internal conditions. At this stage in its development the program can only analyse conduction in a solid which means that the conduction option must always be 'on' - that is 1 instead of 0 in accordance with the specified convention.

HEAT TRANSFER OPTIONS

Data set 6

Heat transfer options specification

NCDUC NCVEC NCRAD NCGEN NCPNT NCFLX NCMSTR

NCDUC include conduction effects

NCVEC include convection boundary condition effects

NCRAD include radiation boundary condition effects

NCGEN include internal heat generation effects

NCPNT include point heat source boundary condition effects

NCFLX include prescribed flux boundary condition effects

NCMSTR include the effects of a liquid layer with sensible and latent heat exchange

4.3 Transient and nonlinear analysis control parameters

The data sets in this section are required whether or not a steady state and/or linear analysis is being carried out.

The method of time integration implemented in the program is a single step, direct integration method and has the condensed form shown below.

$$[K^*]\{T_n^{i+1}\} = \{F^*\}$$

$[K^*]$, the effective stiffness matrix has the form

$$[K^*] = [K(T_n^i)] + [C(T_n^i)]/\alpha\Delta t_n$$

and $[F^*]$, the effective load vector, has the form

$$[F^*] = \{F(T_n^i)\} + [C(T_n^i)]\{T_{n-1}\}/\alpha\Delta t_n + (1 - \alpha)\{\dot{T}_{n-1}\}/\alpha$$

where i = iteration number

n = time-step number

$\{T\}$ = vector of nodal temperatures

$\{F\}$ = load vector

$[K]$ = conductivity (stiffness) matrix

$[C]$ = heat capacity matrix

α = scalar multiplier

Δt = time-step size

The scalar multiplier, α, can have a value in the range $0 < \alpha < 1$. The value of α controls the explicit - implicit nature of the algorithm. A value close to zero results in an algorithm which is essentially explicit whereas a value close to unity results in an implicit algorithm. The closer the value of the multiplier is to 1, the lower the degree of oscillation and the higher the degree of stability of the solution. For most purposes a value between 0.5 and 0.8 should be used.

The first parameter,in this data set is used to indicate to the program which of six possible units of time is being used for the time values being specified. Clearly the user should specify a set of units which is consistent with the time scale of the analysis. The time at the start of the analysis, the time-step increment and the time at the end of the analysis must then be specified (in the predefined system of units). The program adjusts all units of time to seconds at the beginning of the analysis. The parameter which controls the type of time integration scheme to be used must then be specified. Finally the identification number of the time function associated with the time-step size must be specified. This enables a variable time-step to be used - small time-steps should be used when large, suddenly imposed temperature gradients occur such as at the start of an analysis.

TRANSIENT ANALYSIS CONTROL

Data set 7

Time integration control data

NUNITS	START	DTIME	TTIME	ALPHA	DTYMFN

NUNITS indicator of the system of units for time measurement

1 seconds

2 minutes

3 hours

4 days

5 weeks

6 months (30 days)

START time at start of analysis

DTIME time increment

TTIME time at end of analysis

ALPHA time integration algorithm parameter (0 < ALPHA < 1)

DTYMFN time function identification number associated with DTIME

The rest of the data sets in this section deal with nonlinear solution strategy control

In nonlinear analysis, a problem can only be solved by initially assuming values for the unknown temperatures. The solution of the problem generates a new set of temperatures which are not necessarily the same as the assumed values. If they are not sufficiently close to the assumed values, the new temperatures are used as the next assumed temperatures. This procedure can be called iterating towards a converged solution or an equilibrium. In most cases the greater the degree of nonlinearity, the more iterations are required to achieve convergence. If the user knows that the degree of nonlinearity is small or that the temperature distribution in the body is not a strong function of time during certain time intervals, he can use this knowledge to reduce forcibly the computational effort and number of iterations required to converge to a solution. This can be done in two ways. Firstly, by controlling how often the effective stiffness matrix is reformed - a realistic option if both the thermal conductivity and the heat capacity are weak functions of temperature. Secondly, by controlling how often equilibrium iterations are to be performed - a realistic option if it is known that the temperatures are weak functions of time at a certain stage in the analysis

The next data set is used to specify in how many time-step blocks the effective stiffness matrix is to be reformed and equilibrium iterations are to be performed. Currently neither of these options is active, but have been included for the purposes of a subsequent release of the program. In this version both stiffness reformation and equilibrium iterations are always carried out. The maximum number of equilibrium iterations in any time-step is also specified in this data set. Generally this value need not exceed 25 but in the case of a radiation boundary condition 500 iterations may be required. If this limit is reached during the course of an analysis the program notifies the user in the output file that convergence was not achieved and then continues with the analysis.

The next value which must be specified in this data set is the convergence tolerance. This is the factor which the program uses in the expression below to judge whether or not convergence has been achieved.

$$\frac{| T^{i} - T^{i+1} |}{| T^{i+1} |} \leq \delta$$

where T^{i} = previous iteration temperature (°C)
T^{i+1} = current temperature (°C)
δ = convergence tolerance

When this relationship is true for all the nodal temperatures then convergence is judged to have been achieved.

Considering that double precision arithmetic is used at all times, it does not make sense to specify a tolerance of value less than $1\mathrm{x}10^{-15}$. A value of $1\mathrm{x}10^{-6}$ is more than adequate for most purposes.

The method used to determine the next estimate of the nodal temperatures is a modified direct substitution method. The next temperature is predicted using the following equation

$$T^{i+1} = \omega T^{i} + (1 - \omega)[(T^{i} + T^{i-1})/2] \qquad \omega \in [0,1]$$

where T^{i} = current temperature (°C)
T^{i-1} = previous temperature (°C)
T^{i+1} = new temperature (°C)
ω = modification factor

Setting $\omega = 1$ is equivalent to using the direct substitution method while setting $\omega = 0$ is equivalent to approximating the new temperature by averaging the current and previous temperatures. If there is no oscillation in the approach to the converged solution (such as there is in the case of a radiation boundary condition) it is best to set ω equal to 1. If there is oscillation then the user must choose a value between 0 and 1 according to the nature of the problem. Aitken's acceleration method may also be implemented.

$$T_j = \frac{T_{i-2}T_i - T_{i-1}^2}{T_i - T_{i-1} + T_{i-2}}$$

where T_i = temperature at iteration i (°C)
T_j = adjusted temperature at iteration i (j=i) (°C)

In the case of nonlinear analysis a number of analyses may have to be performed before a combination of nonlinear analysis control parameters is found which results in a converged solution.. The specification of the last parameter in this data set, the over-relaxation factor, allows the way in which the analysis converges to a solution to be controlled more closely. The value of the over-relaxation factor used and whether or not Aitken's acceleration is included generally influences the nature of the convergence or divergence of a nonlinear analysis.

NONLINEAR ANALYSIS CONTROL

Data set 8

Nonlinear solution strategy control data

NBSTR	**NBEQI**	**MITER**	**TOLER**	**OVRFAC**	**NAITKN**

NBSTR number of blocks of time-steps in which the effective stiffness matrix is to be reformed

NBEQI number of blocks of time-steps in which equilibrium iterations are to be performed

MITER maximum number of equilibrium iterations to be performed in any time-step within any one of the NBEQI time-step blocks.(MITER > 0)

TOLER convergence tolerance for equilibrium iterations

OVRFAC over-relaxation factor used in the prediction of the nodal temperatures for use in the next iteration of an analysis. It must lie in the range 1 >= OVRFAC >= 0. It is used in the following equation.

$$T^{i+1} = \alpha T^{i} + (1-\alpha)(T^{i} + T^{i-1})/2$$

where α = OVRFAC

This variable is intended for use with radiation boundary condition problems to prevent oscillation of the nodal temperatures. When not dealing with a radiation boundary condition it is best to set this variable to 1.0.

NAITKN control variable to activate or deactivate the use of Aitken's acceleration method.

The next two data sets are used to specify blocks of time-steps in which effective stiffness reformation and equilibrium iterations are suppressed. As these options are both self-explanatory and currently not used, they are not discussed.

NONLINEAR ANALYSIS CONTROL

Data set 8.1r

Stiffness reformation blocks

NBSTR of this data set are required

IFIRST LAST INC

IFIRST time-step number of the first time-step in this block

LAST time-step number of the last time-step in this block

INC time-step increment

NONLINEAR ANALYSIS CONTROL

Data set 8.2r

Equilibrium iteration blocks

NBEQI of this data set are required

IFIRST	LAST	INC

IFIRST time-step number of the first time-step in this block

LAST time-step number of the last time-step in this block

INC time-step increment

4.4 Output control parameters

The results generated by the program during the course of an analysis include:

- nodal temperatures
- nodal temperature rates
- nodal fluxes - if there are prescribed temperatures at the nodes
- an analysis and breakdown of the time spent on various computational operations during the analysis

There user can control the frequency and format of the presentation of the first three as well as specify which of the three are to be printed. The time-logging option can only be activated or deactivated (1/0).

The user has three means of controlling the output of heat transfer analysis results at his disposal. Firstly, he can specify that results be printed only at certain times. Secondly, he can specify how often results are printed during the process of iterating to equilibrium. For instance, he could instruct the program to print the intermediate results every fifth iteration or only when the temperatures have converged. Thirdly, he can specify that only the results for certain node numbers be printed. The user should be aware that he has the above-mentioned control over the time of the output of the thermal results as a group. In other words, he cannot specify that temperatures be printed at every time-step and fluxes at every tenth time-step. Of course, if he has specified that reaction fluxes and temperature rates are not to be printed at all then nothing he does in the rest of the output specification section can override that. He may, however, specify the output of results for different node numbers for each type of thermal result. It should be noted that reaction fluxes are only printed for node numbers which have a prescribed temperature boundary condition.

The way in which the data sets below are used to control output is best explained by means of an example. Consider the transient, nonlinear analysis of a domain of 20 nodes over a time period of 10 seconds with a time-step size of 0.1 seconds. Assume that the results are to be printed for nodes 1, 2, 3, 4, 5, 10, 12, 14 and 20 every second iteration. Assume also that the results are to be printed at the following times: 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 1.0, 1.5, 2.0, 4.0, 6.0 and 10.0 seconds. It can be seen that the node numbers and times can each be divided up into a number of arithmetic series with different increments. In the terminology of this manual, these distinct series are called 'blocks' of node numbers or times. The various blocks are written below in the format: starting point, end point, increment and (block number).

Node numbers

1	5	1	(1)
10	14	2	(2)
20	20	0	(3)

Times

0.0	0.4	0.1	(1)
0.5	1.5	0.5	(2)
2.0	6.0	2.0	(3)
10.0	10.0	0.0	(4)

For the specification of the time-step blocks the actual times must be converted to their corresponding time-step numbers bearing in mind the possibility of a time dependent time-step size. So, the time blocks would be:

Time-step numbers

0	4	1	(1)
5	15	5	(2)
20	60	20	(3)
100	100	0	(4)

Considering the above and with reference to the following data set:

NBTSP = 4, IPRIN = 2, NBTEM = 3, NBFLX = 0, and NBTDT = 0

It is possible to specify whether or not converged temperature results are to be also directed to a post-processing file. Only one block of node numbers can be output but they need not form an arithmetic series.

OUTPUT CONTROL

Data set 9

Selection of variables for output

NBTSP	IPRIN	NBTEM	NBFLX	NBTDT	NPOST	NLOG

NBTSP number of blocks of time-steps for which results are to be printed. This value should always be greater than or equal to 1; otherwise no results are printed

IPRIN iteration number at which results are to be printed at each time-step, in addition to the converged results; if IPRIN = 0 only the converged results are printed

NBTEM number of blocks of node numbers for which temperature results are to be printed

NBFLX number of blocks of node numbers for which reaction fluxes are to be printed

NBTDT number of blocks of node numbers for which temperature rates are to be printed

NPOST indicator of whether or not to write specified results to a post-processing file.

NLOG control variable to activate or deactivate time logging during an analysis

The next data set is used to specify the number of columns in which the various results are to be printed. This option is useful for customising the output for viewing on either screen or hardcopy.

OUTPUT CONTROL

Data set 9.1

Output format control

NBTSP of this data set are required

```
NPCOLS
```

NPCOLS the number of columns of output required. The variable can have a value in the range 1 to 6 inclusive. For a standard 80 character per line printer or screen, three columns of temperatures, reaction fluxes or temperature rates are recommended. In the case of 132 characters per line, the maximum of 6 columns can be used.

The following four data sets are required for the specification of the time-step and node number blocks referred to above.

OUTPUT CONTROL

Data set 9.2r

Time-step blocks

NBTSP of this data set are required

```
IFIRST     LAST     INC
```

IFIRST time-step number of the first time-step in this block

LAST time-step number of the last time-step in this block

INC time-step increment

OUTPUT CONTROL

Data set 9.3r

Temperature blocks

NBTEM of this data set are required

IFIRST	LAST	INC

IFIRST	node number of the first node in this block
LAST	node number of the last node in this block
INC	node number increment

OUTPUT CONTROL

Data set 9.4r

Reaction flux blocks

NBFLX of this data set are required

```
IFIRST    LAST    INC
```

IFIRST node number of the first node in this block

LAST node number of the last node in this block

INC node number increment

OUTPUT CONTROL

Data set 9.5r

Temperature rate blocks

NBTDT of this data set are required

IFIRST	**LAST**	**INC**

IFIRST	node number of the first node in this block
LAST	node number of the last node in this block
INC	node number increment

The next three data sets are required for the specification of the details of the post-processing file and are fully explained in the accompanying text. They are only required if results are in fact to be written to a post-processing file.

OUTPUT CONTROL

Data set 9.6

Post-processing file name

This data set is required if NPOST > 0

FILENAME

FILENAME the name of the logical variable with which the post-processing data file is associated. The name of the post-processing file is assigned to this logical variable with the DCL ASSIGN statement.

OUTPUT CONTROL

Data set 9.7

Post-processing file status

This data set is required if NPOST > 0

STATUS

STATUS this is a three character string which can be either 'OLD' or 'NEW'. This string indicates whether the post-processing file exists prior to the start of the analysis - OLD - or whether this data file must be created - NEW.

OUTPUT CONTROL

Data set 9.8

Post-processing control data

This data set is required if NPOST > 0

```
NNPOST     N1   N2   N3 .....
```

NNPOST number of nodes for which output to the post-processing file is required. This number may not exceed 6.

Ni node numbers of the nodes of which the temperatures are to be written to file. NNPOST of these node numbers must be specified

The next data set deals with restart analysis. It is not discussed here as it is currently not an active option.

OUTPUT CONTROL

Data set 9.9

Restart analysis control data

This data set is required if MODEX = 2

IREST

IREST time-step interval at which the state of the model is written to file

4.5 Nodal coordinate specification

Nodes define the boundaries and the positions of the elements within the finite element mesh. As this program can only analyse two dimensional domains, only two coordinates for each node need be entered. As it is very easy to make typing errors in this section which cannot be trapped by the program's data checking facility, great care must be exercised when entering the coordinates.

Although the ordering of the node numbers in the mesh is irrelevant with regard to solution efficiency, there are some rules and guidelines which must be adhered to.

- a node number can never be greater than the total number of nodes in the model
- a node number can never be less than 1
- a node number cannot be repeated.

Nodal coordinates may either be entered explicitly for each node or a primitive nodal coordinate generation facility may be used. Use of the latter option reduces the tedium of entering every node's coordinates. It should be considered whenever the mesh contains contiguous elements of the same size in one or both dimensions. To enable use of the coordinate generation facility it is a prerequisite that the pertinent node numbers form an arithmetic series. This enables the coordinates of each node between the two specified nodes to be calculated. For example,

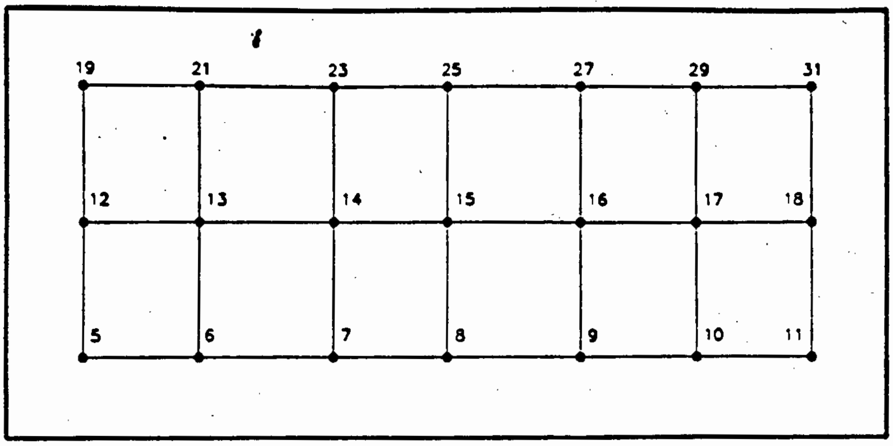


Figure 1 Numbered nodes in a mesh of finite elements.

to enable automatic coordinate generation the following data sets would be entered:

5	0.0	0.0	1
11	10.0	0.0	1
12	0.0	1.0	1
18	10.0	1.0	1
19	0.0	2.0	2
31	10.0	2.0	2

If the user had entered a data set such as

5	0.0	0.0	1
18	10.0	1.0	1

hoping to generate the lower half of the mesh in figure 1, an error would have been incurred. The program would interpret the above data sets to mean:

> Position node numbers 5 and 18 at (0.0;0.0) and (10.0;1.0) respectively and draw a straight line between them. Position twelve equally spaced nodes numbered consecutively 6 to 17 between these two nodes.

This is definitely not the same as the mesh in figure 1.

If a mesh contains nodes which are effectively at infinity these nodes must be assigned node numbers and be included in the coordinate specification list. However, the user should assign large values to the coordinates of these nodes, (999.99;999.99) for example.

It is essential that the coordinates of the nodes be entered in order of ascending node number.

NODAL COORDINATE SPECIFICATION

Data set 10r

Nodal coordinates

N X Y INC

N node number

X X coordinate of node N

Y Y coordinate of node N

INC node number increment used for nodal coordinate generation (INC >= 0). If the node immediately following node N in the list, say node N^*, is not node N+1 then the nodal coordinates of all nodes in the series N + INC, N + 2*INC, ..., N^* - INC are automatically generated by linearly interpolating between the coordinates of nodes N and N*. Note that setting INC = 0 prevents nodal coordinate generation if node N* is not equal to node N + 1

4.6 Element specification

Whenever a heat transfer analysis is carried out on some body, there are generally areas of the body where the temperature gradients are high and other areas where they are low. The former case requires more rigorous modelling than the latter and one of the ways in which this is managed is in the allocation of different element types to different areas during the domain discretisation process. To deal with the complexities - from the program's point of view - of mixed element types, the concept of element groups must be introduced. Element groups do not only differ from each other in terms of their constituent element type, but also due to the nature of their material property temperature dependency. There may be more than one type of material within an element group but their property temperature dependencies must all be of the same type.

To summarise, a collection of elements is regarded as an 'element group' if the conditions listed below are all satisfied:

- The elements within the collection are all of the same type - NETYP is a constant as are NNODE and NGAUS
- The temperature dependency model of all thermal properties is the same within the collection of elements - NTHID is constant

It is permissible to have element groups with similar or identical characteristics if desired.

It is important from a point of view of solution efficiency that contiguous element groups are numbered consecutively. For example

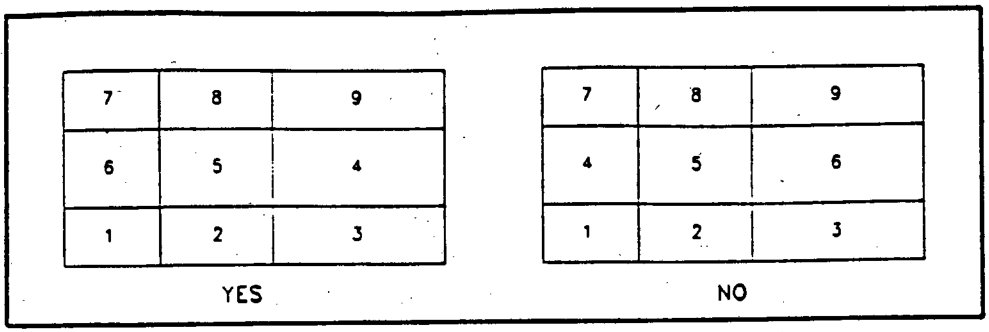


Figure 2 Correct and incorrect element numbering schemes.

Similarly, contiguous elements within an element group must also be numbered consecutively.

The data required to specify each element group can be sub-divided into the following categories:

- element group control parameters
- element group material properties
- element group topology

The data within each element group must be complete in the sense that no cross-referencing of data between groups is allowed.

NOTHCAN currently supports two types of element

- Two dimensional continuum finite elements
- Two dimensional continuum infinite elements

Although these two element types merely result in different element groups, the input data they require is sufficiently different for them to be dealt with in two sub-sections. Finite elements are used to model domains which have well defined, finite boundaries. Some domains, however, have one or more boundaries which are defined at such a distance from the region of interest that they can be regarded as being at infinity. Domains of this nature can be modelled by truncating a mesh consisting of a large number of finite elements at some distance away from the region of interest. This distance should be sufficiently large that the truncation does not adversely affect the quality of the solution generated. A far more satisfactory and elegant way of modelling such domains involves the use of both finite and infinite elements - the finite elements to model the regions of interest and the finite boundaries and the infinite elements to model the boundaries at infinity.

The following data set is used to specify both the element type and the thermal properties temperature dependency model.

ELEMENT GROUP CONTROL

Data set 11r

Primary element group and thermal model specification data

This is the first data set in a group of element group data sets. There must be NEGRP repetitions of these data sets. The final data set in this group is data set 11.1.4 if NETYP = 1 and data set 11.2.3 if NETYP = 2.

NETYP NTHID

NETYP element type

1 2-D finite elements

2 2-D infinite elements

NTHID heat conduction model identification number

1 isotropic constant conductivity, heat generation and heat capacity

2 isotropic linearly varying conductivity, heat generation and heat capacity

3 isotropic piecewise linearly varying conductivity, heat generation and heat capacity

If NETYP = 2 then go to data set 11.2.1.

4.6.1 2D continuum finite elements

Within this category of element types there are three different options available - 4, 8 or 9 noded elements. They are all isoparametric, quadrilateral, two dimensional continuum elements and the 4 noded element is bilinear while the other two are biquadratic. This means that whenever an element has curved sides or is in an area where there is likely to be a significant temperature gradient, one of the biquadratic elements should be used. Figure 3 illustrates the various elements available.

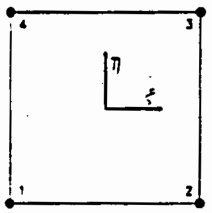
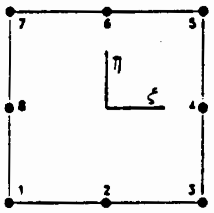
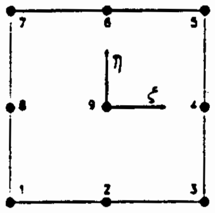
NUMBER OF NODES PER ELEMENT (NNODE)	4	8	9
ELEMENT TOPOLOGY			 7 6 5 η 8 9 ξ 4 1 2 3

Figure 3 Finite element topology.

The following data sets require the specification of the number of elements within the current element group, the number of nodes per element - referred to above - and the order of Gaussian Integration to be used. The options for this parameter are listed as 2 or 3, but the following rule must be followed - third order integration must be used with quadratic elements. The number of different materials occurring in this element group is also specified in this data set. These properties include material density and thickness as well as the base values of internal heat generation, thermal conductivity and heat capacity.

FINITE ELEMENTS

Data set 11.1.1

Secondary element group specification data

This data set is required if NETYP = 1

NELEM NNODE NGAUS NMAT

NELEM number of elements in this element group

NNODE number of nodes per element (4, 8 or 9)

NGAUS order of Gaussian Quadrature (2 or 3)

NMATS number of material property sets within this element group

The next group of data sets is used to specify which elements have one or more sides with a boundary condition. The first data set is used to specify how many elements are subjected to each of prescribed flux, convection, mass transfer, radiation and point load boundary conditions while the data sets which follow are used to specify the actual element numbers. This group of data sets is always required unless only prescribed temperature boundary conditions occur, when it must be omitted.

FINITE ELEMENTS

Data set 11.1.2

Specification of elements with boundary conditions

This data set is required if NCFLX > 0, NCVEC > 0, NCRAD > 0, NCPNT > 0 or NMSTRL > 0.

NFLXEL NVECEL NRADEL NPNTEL NMSTRL

NFLXEL number of elements with prescribed flux boundary conditions in this element group

NVECEL number of elements with convection boundary conditions in this element group

NRADEL number of elements with radiation boundary conditions in this element group

NPNTEL number of elements with point heat sources in this element group

NMSTRL number of elements with mass transfer boundary conditions in this element group

FINITE ELEMENTS
SPECIFICATION OF ELEMENTS WITH BOUNDARY CONDITIONS

Data set 11.1.2.1r

Elements with flux boundary conditions

NFLXEL of this data set are required

NUMELF

NUMELF number of an element with one or more flux boundary conditions imposed on it. The element numbers must be entered in ascending order

FINITE ELEMENTS
SPECIFICATION OF ELEMENTS WITH BOUNDARY CONDITIONS

Data set 11.1.2.2r

Elements with convection boundary conditions

NVECEL of this data set are required

NUMELC

NUMELC number of an element with one or more convection boundary conditions imposed on it. The element numbers must be entered in ascending order

FINITE ELEMENTS
SPECIFICATION OF ELEMENTS WITH BOUNDARY CONDITIONS

Data set 11.1.2.3r

Elements with radiation boundary conditions

NRADEL of this data set are required

NUMELR

NUMELR number of an element with one or more radiation boundary conditions imposed on it. The element numbers must be entered in ascending order

FINITE ELEMENTS
SPECIFICATION OF ELEMENTS WITH BOUNDARY CONDITIONS

Data set 11.1.2.4r

Elements with point heat sources

NPNTEL of this data set are required

NUMELP

NUMELP number of an element with one or more point heat sources imposed on it. The element numbers must be entered in ascending order

FINITE ELEMENTS
SPECIFICATION OF ELEMENTS WITH BOUNDARY CONDITIONS

Data set 11.1.2.5r

Elements with mass transfer boundary conditions

NMSTRL of this data set are required

NUMELM

NUMELM number of an element with one or more edges which have a mass transfer boundary condition. The element numbers must be entered in ascending order

The next group of data sets is used to specify material property information. The first data set in this group requires the specification of the density and thickness of the material. As the program is only able to analyse two dimensional continua, the thickness of the material must be the same in element groups. It is only included in this section to allow for the inclusion at some later date of three dimensional continuum elements.

FINITE ELEMENTS
MATERIAL PROPERTIES

Data set 11.1.3.1r

Physical properties

This is the first data set of a group of material properties data sets. NMATS repetitions of this group are required. The final data set in this group is data set 11.1.3.4.4

```
MATERIAL    DENSITY    THICKNESS
```

MATERIAL material set identification number, (MATERIAL cannot exceed NMATS and must be greater than zero)

DENSITY material mass density

THICKNESS element thickness

The next sub-group of data sets within the properties group deals with the thermal physical properties of internal heat generation, conductivity and heat capacity. (the units for all physical properties can be found at the beginning of this manual). There are a number of models of thermal property temperature dependence. These include:

- constant properties
- linearly varying properties
- piecewise linearly varying properties

The first case is self-explanatory. The second merely involves choosing a sensible maximum and minimum temperature in the context of the problem, evaluating the properties at these extreme temperatures and determining the equation of the straight line joining the two data pairs. The piecewise linear approximation is used to approximate a curve. It entails specifying the number of property-temperature data pairs and then entering them in order of increasing temperature. The initial and final temperatures must bracket the range of temperatures likely to occur in the course of the analysis.

FINITE ELEMENTS
MATERIAL PROPERTIES

Data set 11.1.3.2

Thermal physical properties

This data set is required if NTHID = 1

Q	K	C

Q	internal heat generation
K	thermal conductivity
C	specific heat capacity

FINITE ELEMENTS
MATERIAL PROPERTIES

Data set 11.1.3.3

Thermal physical properties

This data set is required if NTHID = 2

a	b	c	d	e	f

$$Q = a + bT$$

$$K = c + dT$$

$$C = e + fT$$

FINITE ELEMENTS
MATERIAL PROPERTIES

Data set 11.1.3.4

Thermal physical properties

This data set is required if NTHID = 3.

```
NQPTS   NKPTS   NCPTS
```

NQPTS number of points at which internal heat generation is to be specified

NKPTS number of points at which thermal conductivity is to be specified

NCPTS number of points at which specific heat capacity is to be specified

FINITE ELEMENTS
MATERIAL PROPERTIES

Data set 11.1.3.4.2

Piecewise linear thermal conductivity

This data set is required if NTHID = 3

```
K1  T1    K2  T2 ..........
```

Ki	thermal conductivity value
Ti	reference temperature for Ki

There must be NKPTS data pairs in this data set

FINITE ELEMENTS
MATERIAL PROPERTIES

Data set 11.1.3.4.3

Piecewise linear heat capacity

This data set is required if NTHID = 3

```
C1  T1    C2  T2 ..........
```

Ci	specific heat capacity value
Ti	reference temperature for Ci

There must be NCPTS data pairs in this data set

The final data set in the material properties group is used to identify the time function associated with internal heat generation. It enables the user to define the time dependency of internal heat generation. Time functions are explained in a later section.

FINITE ELEMENTS
MATERIAL PROPERTIES

Data set 11.1.3.4.4

Internal heat generation time function number

This is the last data set in the group of material properties data sets; return to data set 11.1.3.1 if there have been less than NMATS repetitions of this group.

```
NQDAT
```

NQDAT identification number of the time function associated with the value of the internal heat generation, Q. If Q = 0 then set this value to zero

The final data set in the element group specification group of data sets is used to specify what material and which nodes are associated with individual elements within the element group. The order in which the node numbers may be entered is constrained by the following rules:

- node numbers must be specified in sequential order in an anti-clockwise direction around the perimeter of an element
- the first node specified must be a corner node
- if there is a centre node - in the case of a 9 noded element - it must be specified last

Entering the node numbers associated with each element is an error prone procedure. An error in a node number entered in this section generally results in a run-time error with an associated 'Invalid Jacobean' error message.

It should be noted that elements must be entered in order of ascending identification number and that the element numbering must start at 1 and finish at NELEM. There is no global element numbering scheme as there is in the case of node numbers and the element identification numbers in each element group must begin at 1. It is also important to note that the second variable in the data set, the material identification number must fall in a meaningful range. The last data set, then, requires an element identification number, a material identification number and the node numbers of the nodes of the current element.

FINITE ELEMENTS
ELEMENT GROUP TOPOLOGY

Data set 11.1.4r

Specification of element material and node numbers

NELEM of this data set are required. This is the last data set in the group of element group data sets. If there have been less than NEGRP repetitions of this group then return to data set 11.

```
IELEM    MATNO    N1    N2    N3 ........
```

IELEM element number (1 <= IELEM <= NELEM). These numbers must be entered in ascending order

The following data set requires the specification of the number of elements within the current element group and the identification number of the infinite element type to be used. There are four possibilities. The first element type is for use with 4 noded quadrilateral elements and is singly infinite. The following two are also singly infinite but may only be used with 8 or 9 noded quadrilateral elements. The fourth type is doubly infinite and may only be used with 8 or 9 noded infinite elements. It should be noted that singly infinite elements are generally used on edges while doubly infinite elements are used on corners. This data set also requires the specification of the order of integration and the number of material property sets.

INFINITE ELEMENTS

Data Set 11.2.1

Secondary element group specification

This data set is required if NETYP = 2

NELEM	**NCODE**	**NGAUS**	**NMAT**

NELEM number of elements in this element group

NCODE code number which identifies the specific infinite element type

1 singly infinite element - 4 finite nodes, based on a 6 noded isoparametric element

2 singly infinite element - 5 finite nodes, based on an 8 noded isoparametric element

3 singly infinite element - 6 finite nodes, based on a 9 noded isoparametric element

4 doubly infinite element - 4 finite nodes, based on a 9 noded isoparametric element

NGAUS order of Gaussian Quadrature (2 or 3)

NMATS number of material property sets within this element group

---

The material properties data sets are the same as for finite elements and information regarding these can be found in the preceding section.

INFINITE ELEMENTS
MATERIAL PROPERTIES

Data Set 11.2.2.1r

Physical properties

This is the first data set of a group of material properties data sets. NMATS repetitions of this group are required. The final data set in this group is data set 11.2.2.5.

MATERIAL DENSITY THICKNESS

MATERIAL	material set identification number, (MATERIAL cannot exceed NMATS and must be greater than zero)
DENSITY	material mass density
THICKNESS	element thickness

INFINITE ELEMENTS MATERIAL PROPERTIES

Data Set 11.2.2.2

Thermal physical properties

This data set is required if NTHID = 1

Q	K	C

Q	internal heat generation
K	thermal conductivity
C	specific heat capacity

INFINITE ELEMENTS
MATERIAL PROPERTIES

Data Set 11.2.2.3

Thermal physical properties

This data set is required if NTHID = 2

a	b	c	d	e	f

$$Q = a + bT$$

$$K = c + dT$$

$$C = e + fT$$

INFINITE ELEMENTS
MATERIAL PROPERTIES

Data Set 11.2.2.4

Thermal physical properties

This data set is required if NTHID = 3

```
NQPTS   NKPTS   NCPTS
```

NQPTS number of points at which internal heat generation is to be specified

NKPTS number of points at which thermal conductivity is to be specified

NCPTS number of points at which specific heat capacity is to be specified

INFINITE ELEMENTS
MATERIAL PROPERTIES

Data Set 11.2.2.4.2

Piecewise linear thermal conductivity

This data set is required if NTHID = 3

```
K1  T1    K2  T2 ..........
```

Ki thermal conductivity value

Ti reference temperature for Ki

There must be NKPTS data pairs in this data set

INFINITE ELEMENTS
MATERIAL PROPERTIES

Data Set 11.2.2.4.3

Piecewise linear heat capacity

This data set is required if NTHID = 3

```
C1  T1   C2  T2 ..........
```

Ci	specific heat capacity value
Ti	reference temperature for Ci

There must be NCPTS data pairs in this data set

INFINITE ELEMENTS
MATERIAL PROPERTIES

Data Set 11.2.2.5

Internal heat generation time function number

This is the last data set in the group of material properties data sets; return to data set 11.2.2.1 if the number of repetitions of this group is less than NMATS.

NQDAT

NQDAT the identification number of the time function associated with the value of the internal heat generation, Q. If Q = 0 then set this card to zero

The first two items of data required in the element topology data set are the same as for finite elements. However, more care must be taken when entering the node numbers. At this stage the user should refer to figure 4 which shows the various infinite elements available.

INFINITE ELEMENTS
ELEMENT GROUP TOPOLOGY

Data set 11.2.3r

Specification of element material and node numbers

NELEM of this data set are required. This is the last data set in the group of element group data sets. If the number of repetitions of this group is less than NEGRP then return to data set 11.

```
IELEM    MATNO    N1    N2    N3 ........
```

IELEM element number (1 <= IELEM <= NELEM). These numbers must be entered in ascending order

MATNO material properties set identification number

Ni the i^{th} node number in the element topology list. The order of entry of these nodes is important. Both the finite and infinite node numbers must be specified in this data set in the order explained above

4.7 Initial conditions specification

Any transient thermal analysis requires the specification of the initial temperature distribution throughout the body. The finite element analogue of this is the specification of the initial temperature of each node in the mesh. The temperature specification can be done in one of two ways - specifying the temperature of each node explicitly or generating the temperatures for various blocks of node numbers. If a steady state analysis is being carried out, the initial temperatures must still be specified - but with the generation option this is easy. It is essential that all nodes are assigned an initial temperature.

The following data set requires the specification of two data values - a value specifying whether or not temperatures are to be either explicitly entered or generated and, if generated, the number of blocks of node numbers for which this is to be done.

INITIAL CONDITIONS
Data Set 12

Initial conditions specification control

```
IPTYP    NIPGEN
```

IPTYP indicator of the method by which initial temperatures are to be assigned to nodes

1 initial nodal temperatures to be entered individually

node numbers, the last node number and the node number increment. If the user wishes to specify the initial temperature of a single node while using the generation facility, he must include it in the block count mentioned earlier and enter a data set such as

```
100.0   21    21    1
```

where '100.0' is the initial temperature, and '21' is the node number. Notice that the increment must have a value of at least 1.

INITIAL CONDITIONS

Data Set 12.2r

Generated initial temperatures

This data set is required if IPTYP = 2. NIPGEN of this data set are required

TEMPI KFNOD KLNOD INCRMT

TEMPI	initial temperature of the nodes in this block
KFNOD	first node number of this block of nodes
KLNOD	final node number of this block of nodes
INCRMT	node number increment

4.8 Boundary conditions

The boundary conditions which can be used in problems of heat transfer analysis in the context of the program can be broadly sub-divided into two categories:

- prescribed temperatures
- directly and indirectly prescribed fluxes

The first category is self-explanatory. The second category consists of the following types of boundary conditions

- point heat sources or sinks
- distributed edge fluxes
- fluxes due to convection
- fluxes due to mass transfer effects
- fluxes due to radiation

As the boundary conditions (together with the initial conditions and domain geometry and properties) are what distinguish one heat analysis problem from another, each type is described in some detail in its own section.

4.8.1 Prescribed temperatures

This is the simplest type of boundary condition which is possible. It merely involves specifying the temperatures of various edge nodes.

The following data set requires the specification of an edge node number followed by the nodal temperature and the identification number of the time function associated with this temperature. The time function enables the user to specify a time dependent nodal temperature. It should be noted that it also enables the user to model the application of a temperature change to a surface more realistically. For example if the initial temperature of a boundary were 20 °C and it were suddenly brought into contact with a body at 2000 °C it would not be reasonable to assume that the surface temperature of the body of interest would increase to the new temperature instantaneously, nor would it be reasonable to expect the numerical method being used to be sufficiently robust to model accurately such a high temperature gradient. Therefore the boundary temperature would be 'eased' up to its final value of 2000 °C using a combination of time functions and small time-step sizes initially.

The node numbers must be entered in ascending order.

PRESCRIBED TEMPERATURES

Data Set 13.1r

Prescribed temperatures

NVTEM of this data set are required

```
N    T    NTIMF
```

N boundary node number. These numbers must be entered in ascending order

T temperature value

NTIMF identification number of the time function associated with the temperature at node N. (if T = 0 set NTIMF = 0)

4.8.2 Flux type boundary conditions

As has been mentioned previously, there are a number of different flux boundary conditions. The number of sides with one or more of these various boundary conditions is specified in the following data set

FLUX BOUNDARY CONDITIONS

Data Set 13.2

Flux boundary condition options

```
NTEMP   NEGTP   NEGVC   NRDSD   NEGMTR
```

NTEMP number of nodes with point heat sources

NEGTP number of element sides with prescribed flux boundary conditions

NEGVC number of element sides with convection boundary conditions

NRDSD number of element sides with radiation boundary conditions

NEGMTR number of element sides with mass transfer boundary conditions

It is important to note that by convention the energy entering a body is defined to be positive while energy leaving a body is defined to be negative.

4.8.3 Point heat source boundary conditions

A point heat source (or sink) is an energy input which acts on a very small area of a surface. For example a Bunsen burner heating some arbitrary point on a large block of steel. This would be modelled by placing a node at the position of the Bunsen burner. It is very similar to a prescribed flux.

The following data set requires the node number at which there is a point heat source or sink, the value of the energy input (not a flux) and a time function identification number for it. It is important to note that the units of the energy value are Watts not Watts per square metre.

FLUX BOUNDARY CONDITIONS
POINT HEAT SOURCES

Data Set 13.2.1r

Point heat source boundary conditions.

NTEMP of this data set are required

N P NTIMF

N node number at which a point energy source occurs. These numbers must be entered in ascending order

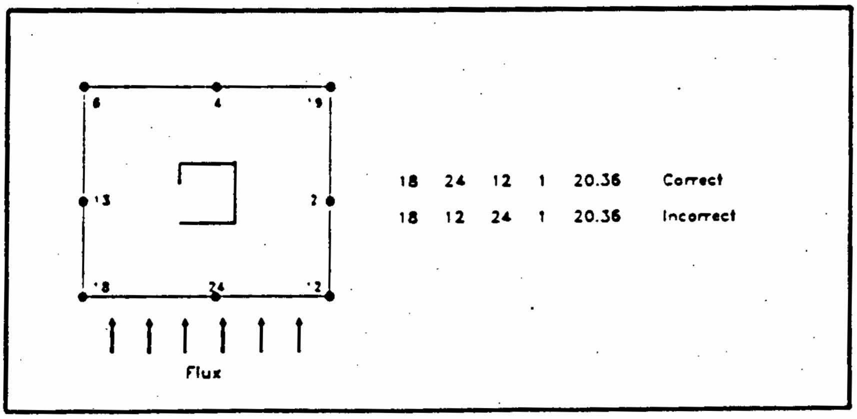
P - magnitude of the point source (W)

NTIMF time function identification number

4.8.4 Prescribed flux boundary condition

This condition is very similar to the point source boundary condition except that it is an energy input distributed over a finite area. This boundary condition is limited to non-spatially varying fluxes over a particular element side. It can, however, be time dependent. It is of interest that the finite element method requires the resolution of this distributed, continuous flux into discrete, point energy sources at the nodes of the edge. This is accomplished by the program.

The prescribed flux data set requires the specification of the current side's node numbers, a time function identification number for this prescribed flux and the value of the flux. The order in which the node numbers are specified is important. They must be entered in an anti-clockwise, sequential order, starting at a corner node. If the element is four noded then the third node number must be set to zero.



18 24 12 1 20.36 Correct
18 12 24 1 20.36 Incorrect

Figure 5 Node number specification for an element side with a flux boundary condition.

If the value of the flux entering a side is prescribed to be zero (that is, the side is insulated) then it need not be specified.

FLUX BOUNDARY CONDITIONS
PRESCRIBED FLUX

Data Set 13.2.2r

Prescribed flux boundary conditions

NEGTP of this data set are required

N1	N2	N3	NTIMF	P

Ni node numbers defining the loaded edge. The node numbers must be entered in anti-clockwise order. For 4-noded elements N3 = 0

NTIMF identification number of the time function associated with the flux

P magnitude of the prescribed flux over the element edge. At present only constant flux can be dealt with. Energy entering the edge is considered positive (W/m^2)

4.8.5 Convection boundary condition

Convection is a mode of heat transfer which can only occur in fluid media. It involves the transfer of energy from one portion of a fluid medium to another by means of induced fluid flow rather than by energy transfer on the molecular level only. This fluid flow is effected either by temperature induced density gradients - natural convection - or by some exterior agency such as a fan or pump - forced convection.

The program can only analyse the conduction of heat in solids and cannot model the temperature distribution within fluids due to convective heat transfer. However, in many instances a solid is in contact with some fluid and energy is transferred between the two. Although it is not possible to model the temperature distribution in both the solid and the fluid, it is possible to determine the temperature distribution in the solid if certain of the fluid properties are assumed to be constant or to change in some predictable manner. The convective flux entering an edge is determined using the following equation.

$$Q = h_c A(T_a - T_w)$$

where Q = energy transfer between solid and fluid (W)
h_c = heat transfer coefficient (W/m^2°C)
A = contact surface area (m^2)
T_w = surface temperature (°C)
T_a = ambient fluid temperature (°C)

The critical factor in the above equation is the heat transfer coefficient. This is the value which, together with the temperature difference expression, controls the rate of energy transfer. This value can be determined in a number of different ways, the most common of which is by use of correlations which take into account the physical properties of solid and fluid at some average temperature, the nature of the convection occurring and the state of flow of the fluid. It is clear that the heat transfer coefficient is a function of the temperature of both the solid and the fluid.

The program allows the user to specify the value of the heat transfer coefficient in three different ways. Firstly it can be entered as a time dependent constant. Secondly, it can be entered as a time dependent piecewise linearly temperature dependent value and, thirdly, it can be determined from the specification of a heat transfer coefficient correlation identification number together with temperature dependent fluid physical properties. The heat transfer coefficient derived from the correlation can also be a function of time. In all cases the ambient temperature of the fluid is also time dependent. This time dependency enables the user to exercise control over the convection boundary condition. For instance the user could turn the convection condition off over a certain time period.

FLUX BOUNDARY CONDITIONS
CONVECTION

Data Set 13.2.3.1

Convection boundary condition control data

This data set is required if NEGVC > 0

NVCTYP

NVCTYP identifies the convection heat transfer option desired:

1 the heat transfer coefficients are explicitly specified by the user and are only functions of time

2 the same as option 1 except that the heat transfer coefficients are piecewise linear functions of temperature

3 the heat transfer coefficients are determined using one of six different correlations. The requisite fluid physical properties are entered as piecewise linear functions of temperature

The next data set is for use in case 1. It requires the specification of the heat transfer coefficient, the fluid temperature and the node numbers of the current side. The same node numbering rules for specifying a side with a prescribed flux apply. It also requires the specification of the time function identification numbers for both the heat transfer coefficient and the ambient fluid temperature.

FLUX BOUNDARY CONDITIONS
CONVECTION
TYPE 1

Data Set 13.2.3.2r

Edge definition

This data set is required if NVCTYP = 1. NEGVC of this data set are required. Go to data set 13.2.4.1.

OHTC	TAMBC	N1	N2	N3	NTIMF1	NTIMF2

OHTC overall film heat transfer coefficient

TAMBC ambient temperature near the edge of interest

Ni numbers of the nodes defining the loaded edge in anti-clockwise order. For 4-noded elements N3 = 0

NTIMF1 identification number of the time function associated with OHTC

NTIMF2 identification number of the time function associated with TAMBC

The rest of the data sets in the convection boundary condition section are required only if NVCTYP is greater then 1

The following data set requires the specification of the number of different fluids interacting convectively with the body being analysed. A fluid is regarded as being different from other fluids if, when NVCTYP is equal to 2, its heat transfer coefficient value or temperature dependency or the fluid ambient temperature is different. If NVCTYP is equal to 3, a fluid is regarded as being different if physical property values or temperature dependencies are different or if its ambient temperature, characteristic length, flow rate or correlation identification numbers is different.

FLUX BOUNDARY CONDITIONS
CONVECTION
TYPE 2 OR 3

Data Set 13.2.3.3

Number of fluids

This data set is required if NVCTYP > 1

```
NFLDS
```

NFLDS the number of different fluids with which the continuum under consideration is in contact.

The following data sets refer to case 2. If dealing with case 3 proceed directly to data set 13.2.3.5.1.

The next data set requires the specification of a fluid identification number, the node numbers of the current side - according to convention - and the time function identification numbers for both the heat transfer coefficient and the fluid ambient temperature. This data must be specified for each side which has a convection boundary condition.

FLUX BOUNDARY CONDITIONS
CONVECTION
TYPE 2

Data Set 13.2.3.4.1r

Edge definition

This data set is required if NVCTYP = 2. NEGVC of this data set are required. If NVCTYP = 3 go to data set 13.2.3.5.1

NFLUNO	N1	N2	N3	NTIMF1	NTIMF2

NFLUNO identification number of the fluid with which this side is in contact. These numbers must be entered in ascending order

Ni numbers of the nodes defining the loaded edge in anti-clockwise order. For 4-noded elements N3 = 0

NTIMF1 identification number of the time function associated with OHTC

NTIMF2 identification number of the time function associated with TAMBC

The following two data sets are used for the specification of the piecewise linearly temperature dependent heat transfer coefficient. Firstly the number of data pairs must be specified and then that number of data pairs must be entered.

FLUX BOUNDARY CONDITIONS
CONVECTION
TYPE 2

Data Set 13.2.3.4.2

User defined heat transfer coefficient data

This data set is required if NVCTYP = 2. It is the first data set in a group of data sets. NFLDS repetitions of this group of data sets are required. The final data set in this group is data set 13.2.3.4.3

NHPTS

NHPTS - the number of data pairs required to define the piecewise linearly temperature dependent heat transfer coefficient

FLUX BOUNDARY CONDITIONS
CONVECTION
TYPE 2

Data Set 13.2.3.4.2.1r

Piecewise linear heat transfer coefficient

NHPTS of this data set are required

```
HTCi    TEMPi
```

HTCi the value of the heat transfer coefficient at temperature TEMPi. These values must be entered in order of increasing temperature

TEMPi the temperature associated with HTCi

The next data set is used to specify the ambient temperature of the current fluid.

FLUX BOUNDARY CONDITIONS
CONVECTION
TYPE 2

Data Set 13.2.3.4.3

Ambient fluid temperature

If the number of repetitions of this group of data sets is less than NFLDS return to data set 13.2.3.4.2. Else go to data set 13.2.4.1.

TAMBC

TAMBC the bulk temperature of the current fluid

Once all the fluids of case 2 have been dealt with, convection boundary condition data specification is finished.

The rest of the data sets in this section are used for the specification of data required in case 3.

The following data set is the same as data set 13.2.3.4.1r

NKFPTS the number of points required for the piecewise linear definition of the fluid thermal conductivity as a function of temperature

NRFPTS the number of points required for the piecewise linear definition of the fluid density as a function of temperature

NVFPTS the number of points required for the piecewise linear definition of the fluid viscosity as a function of temperature

NHFPTS the number of points required for the piecewise linear definition of the fluid specific heat capacity as a function of temperature

NEFPTS the number of points required for the piecewise linear definition of the fluid coefficient of volume expansivity as a function of temperature

---

FLUX BOUNDARY CONDITIONS
CONVECTION

TYPE 3

FLUID PHYSICAL PROPERTIES

Data Set 13.2.3.5.2.1r

Piecewise linear fluid thermal conductivity

NKFPTS of this data set are required

```
Ki    TEMPi
```

Ki the value of the fluid thermal conductivity at temperature TEMPi. These values must be entered in order of increasing temperature

TEMPi the temperature associated with Ki

FLUX BOUNDARY CONDITIONS
CONVECTION
TYPE 3
FLUID PHYSICAL PROPERTIES

Data Set 13.2.3.5.2.2r

Piecewise linear fluid density

NRFPTS of this data set are required

```
RHOi    TEMPi
```

RHOi the value of the fluid density at temperature TEMPi. These values must be entered in order of increasing temperature

TEMPi the temperature associated with RHOi

FLUX BOUNDARY CONDITIONS

CONVECTION

TYPE 3

FLUID PHYSICAL PROPERTIES

Data Set 13.2.3.5.2.3r

Piecewise linear fluid viscosity

NVFPTS of this data set are required

```
VISCi    TEMPi
```

VISCi the value of the fluid viscosity at temperature TEMPi. These values must be entered in order of increasing temperature

TEMPi the temperature associated with VISCi

FLUX BOUNDARY CONDITIONS
CONVECTION
TYPE 3
FLUID PHYSICAL PROPERTIES

Data Set 13.2.3.5.2.4r

Piecewise linear fluid heat capacity

NHFPTS of this data set are required

```
Cpi    TEMPi
```

Cpi the value of the fluid specific heat capacity at temperature TEMPi. These values must be entered in order of increasing temperature

TEMPi the temperature associated with Cpi

FLUX BOUNDARY CONDITIONS
CONVECTION
TYPE 3
FLUID PHYSICAL PROPERTIES

Data Set 13.2.3.5.2.5r

Piecewise linear fluid coefficient of volume expansivity

NEFPTS of this data set are required

```
EXPi    TEMPi
```

EXPi the value of the thermal coefficient of volume expansivity of the fluid at temperature TEMPi. These values must be entered in order of increasing temperature

TEMPi the temperature associated with EXPi

The final data set in the convection boundary condition data specification section for case 3 requires the specification of some additional properties of the current fluid, namely, its ambient temperature, the characteristic length of the surface with which it is in contact - it should not be the length of a single element side, the fluid velocity - which becomes meaningless in the case of natural convection - and a correlation identification number.

FLUX BOUNDARY CONDITIONS
CONVECTION
TYPE 3

Data Set 13.2.3.5.3

Heat transfer coefficient correlation control data

If the number of repetitions of this group of data sets is less than NFLDS then return to data set 13.2.3.5.2.

```
TAMBC    CHRLEN    FLDVLC    NCORID
```

TAMB the bulk temperature of the fluid under consideration

CHRLEN the characteristic length of the surface with which the fluid under consideration is in contact. This is not the length or width of a single element.

FLDVLC the bulk velocity of the fluid under consideration

NCORID the identification number of the heat transfer coefficient correlation to be used in this analysis

Description	Valid Range	Correlation	ID
Natural Convection			
Horizontal surface, heated surface up or cooled surface down.	$10^5< GrPr <2.10^7$	$Nu = .54(GrPr)^{.25}$	1
	$2.10^7<GrPr<3.10^{10}$	$Nu = .14(GrPr)^{.33}$	
Horizontal surface, heated surface down or cooled surface up.	$3.10^5<GrPr<3.10^{10}$	$Nu = .27(GrPr)^{.25}$	2
Vertical surface	$10^3< GrPr <10^9$	$Nu = .59(GrPr)^{.25}$	3
	$10^9< GrPr <10^{12}$	$Nu = .14(GrPr)^{.33}$	
Forced Convection			
Flow parallel to a flat plate.	$Re < 5.10^5$	$Nu = .664Re^{.5}Pr^{.33}$	4
	$Re >= 5.10^5$	$Nu = .036Pr^{.33} (Re^{.8} -23200)$	5
	$Re > 5.10^5$	$Nu = .036Re^{.8}Pr^{.33}$	6

4.8.6 Mass transfer boundary condition

For the purposes of this users' guide the phrase 'heat transfer by mass transfer' means the energy effects of evaporation and condensation. It is very important to note that the program can only analyse problems which deal with single component, pure fluids in gases which do not change phase - for example water vapour in air. The water vapour is the only component of the air mixture which can undergo a phase change in the temperature and pressure range which is considered realistic by the program.

As in the case of convection, mass transfer is not analysed rigorously by means of the solution of the equations of momentum, mass and energy transport but rather by means of a model of the mass transfer process as a boundary condition. A description of the model follows.

In the case of a typical mass transfer boundary condition there are three phases interacting - solid, liquid and gas - resulting in two interfaces - one between the solid and the liquid and the other between the liquid and the gas. This situation is represented in figure 6.

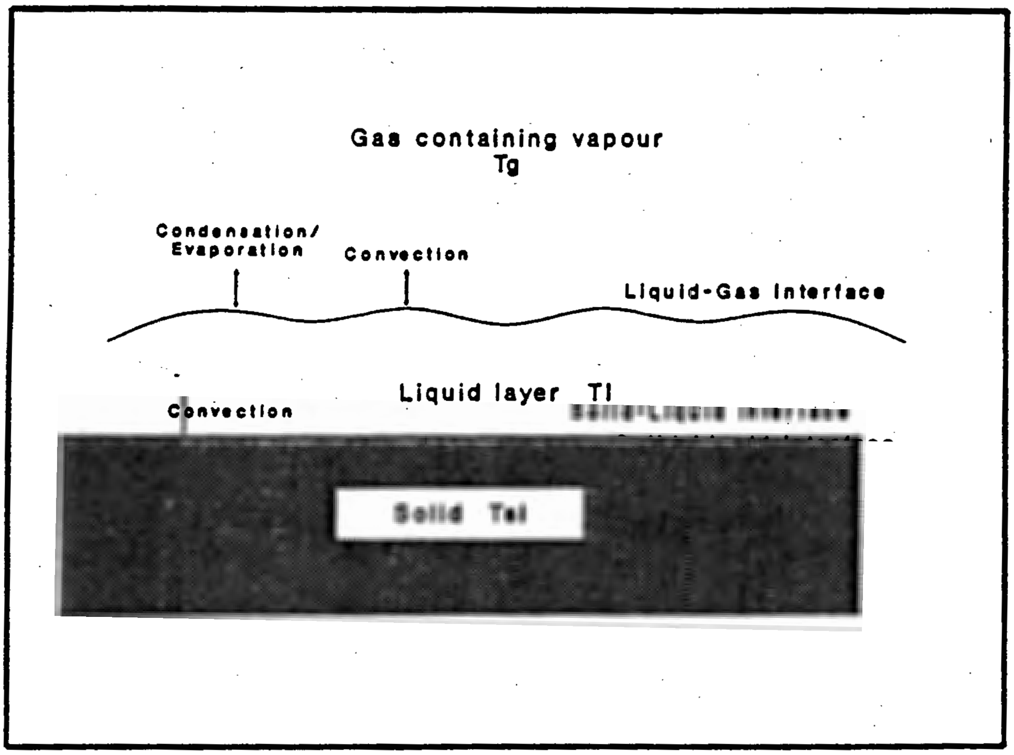


Figure 6 A solid surface in contact with a liquid layer and a moist gas.

From figure 6 it can be seen that there are a number of heat transfer processes occurring as well as mass transfer. Firstly, there is heat conduction in the solid. Secondly, there is heat transfer between the liquid and the solid - convection. Thirdly, there is heat transfer between the liquid and the gas - convection. Fourthly, there is evaporation or condensation occurring between the liquid and the gas with an associated latent energy change.

Given that the only known conditions are the initial temperature distribution in the solid, the initial liquid temperature and the temperature, humidity and pressure of the gas, it can be seen that the interfacial temperatures between the liquid and the solid and the liquid and the gas, which are required for the calculation of the convection flux, are both unknown and dependent on one another. The way in which these interfacial temperatures are approximated is explained below. First the solid-liquid interface temperature approximation is explained and then the liquid-gas interface temperature approximation.

The model allows the solid-liquid interface temperature to be approximated in two ways. Firstly, it can be set as a weighted average of the current solid surface temperature (ie from the previous time-step or iteration) and the liquid layer bulk temperature. It is up to the user to choose an appropriate weighting value - between 0 and 1. Secondly, the interfacial temperature can be calculated by assuming convection into the liquid layer and treating it as a convection boundary condition on the current edge. This is the superior method but it is not always possible to determine a heat transfer coefficient especially when the surface of interest is vertical and the fluid is flowing.

The liquid-gas interface temperature can be approximated in one of the following two ways. Firstly, the weighted average of the liquid bulk temperature and the gas bulk temperature can be used. Secondly, a value determined from a modified lumped parameter analysis of the liquid- gas system can be used. It is based on the relative values of the resistance to heat transfer of the gas phase and the liquid phase. Strictly, lumped parameter analysis should only be applied to solid-fluid systems, but as an approximation it is deemed adequate in this instance. In certain instances the lumped parameter analysis approach should be significantly more accurate than the weighted average approximation, especially in cases when the liquid layer is thin and the Biot number is less than 0.1. In other instances the weighted average approximation is as good if not better.

Whichever method is used, the resultant gas-liquid interface temperature is used as a surface temperature for determining the heat transfer between the liquid and the gas by convection. In this case the user can either explicitly specify the heat transfer coefficient or specify the identification number of an appropriate heat transfer coefficient correlation.

The next step is to determine the energy transferred between the liquid layer and the gas. This is used together with the energy exchanged between the liquid and the solid to carry out an energy balance over the liquid layer and to determine its new bulk temperature which is used in the next iteration. The energy transfer between the liquid and the gas is twofold - convection which has already been dealt with and the energy exchange due to evaporation or condensation. The amount of energy transferred is a function of the rate of mass transfer, a measure of which is provided by a factor called the mass transfer coefficient multiplied by the difference in the partial pressure of the vapour present in the gas and the vapour pressure of the liquid at its surface temperature. The mass transfer coefficient can easily be determined from the heat transfer coefficient by use of the Colburn j-factor analogy. The partial pressure of the vapour in the gas is indirectly specified by the humidity while the vapour pressure of the liquid layer can be determined using Antoine's equation.

Once the rate of mass transfer has been calculated, the value of the latent heat of vaporisation of the liquid is needed to convert the mass transfer rate into an energy transfer rate. Ideally, if a mass of some vapour were condensing, the following energy changes would have to be considered.

- Sensible heat change as the mass of vapour dropped in temperature from its ambient value to the wet-bulb temperature of the gas
- Latent heat of vaporisation as the mass changed phase
- Sensible heat change as the condensed vapour changed to the new liquid temperature

If a mass of liquid were evaporating, the following energy changes would have to be considered.

- Change from ambient liquid temperature to the liquid's bubble point temperature - sensible heat
- Change of phase - latent heat

- Change from bubble point to new ambient gas temperature - sensible heat

In order to simplify the calculation of these energy changes, only the latent heat value is used but not at the wet-bulb or bubble point temperature. Instead it is evaluated at the temperature of the liquid surface. This approximation is considered adequate for the accuracy requirements of the model. The latent heat of vaporisation of the liquid at various temperatures is determined using Watson's correlation

At this stage all the energy changes have been determined. It is still necessary to determine the new liquid layer thickness. This is simple enough as both the mass transfer rate and the time-step size are known.

The model recognises three different surface conditions:

- a dry surface
- a wet surface
- a continuously wetted surface

In the case of a dry surface the model reverts to a normal convection boundary condition but at the same time periodically checks that liquid has not condensed onto the edge. The model description presented above assumes the case of a wet edge and so nothing more needs to be written about this condition. The third case, a continually wetted edge, is modelled by setting the edge temperature to the liquid layer temperature and treating it as a prescribed temperature boundary condition.

Having presented an overview of the model used for the mass transfer boundary condition, it is now appropriate to describe the input data required.

The first data set in this section requires the specification of which method of approximation to use in the calculation of the liquid-solid interface temperature as well as the specification of whether liquid-gas heat transfer coefficients are to be entered explicitly or to be determined from correlations. The liquid-solid heat transfer coefficient can only be determined using correlations.

FLUX BOUNDARY CONDITION
MASS TRANSFER

Data Set 13.2.4.1

Mass transfer condition control data

This data set is required if NCMSTR > 0

```
MTRMDL    MVCTYP
```

MTRMDL the number of the mass transfer model to be used

1 approximate the liquid-solid interfacial temperature using a weighted average

2 approximate the liquid-solid interfacial temperature by assuming convection into the fluid

MVCTYP identifies the convection heat transfer option desired in conjunction with a mass transfer boundary condition:

1 the heat transfer coefficients are to be explicitly specified by the user and are a function of both temperature and time.

2 the heat transfer coefficients are determined using one of six different correlations. The requisite gas

physical properties are entered as piecewise linear functions of temperature. It can also have a user specified dependence on time.

The following two data sets require the specification of the number of liquid-gas systems interacting and the identification of the node numbers of each side with a mass transfer boundary condition and the time function identification numbers for both the heat transfer coefficient and the gas ambient temperature. The same rules and conventions apply to these data sets as apply in the case of a normal convection boundary condition.

FLUX BOUNDARY CONDITION MASS TRANSFER

Data Set 13.2.4.2

Number of liquid-gas systems

This data set is required if NCMSTR > 0

NFLDM

NFLDM number of liquid-gas systems which are involved in mass transfer. If NEGVC is greater than 0 then this number must be less than or equal to NFLDS, the number of fluids involved in convection only. If, however, NVCTYP is not equal to 3, then this variable is independent of the value of NFLDS.

FLUX BOUNDARY CONDITION
MASS TRANSFER

Data Set 13.2.4.3r

Edge definition

NEGMTR of this data set are required.

MFLUNO	**N1**	**N2**	**N3**	**NTIMF1**	**NTIMF2**

MFLUNO identification number of the liquid-gas system with which this side is in contact.

Ni numbers of the nodes defining the loaded edge in anti-clockwise order. For 4-noded elements N3 = 0

NTIMF1 identification number of the time function associated with the convection heat transfer coefficient

NTIMF2 identification number of the time function associated with the ambient gas temperature, TAMBC.

The next data set requires the specification of the relative humidity of the current gas, its associated time function number and the absolute pressure of the gas in atmospheres.

FLUX BOUNDARY CONDITION
MASS TRANSFER

Data Set 13.2.4.4.1r

Gas condition data

This is the first data set in a group of data sets. NFLDM repetitions of this group of data sets are required. The final data set in this group is data set 13.2.4.4.10.

```
AHUMID   NTIMF   PTOT
```

AHUMID the relative humidity of the gas with respect to the liquid which forms part of the system. This must be entered as a percentage.

NTIMF the identification number of the time function associated with humidity

PTOT the ambient absolute pressure of the gas. This must be entered in units of atms.

The following data set requires the specification of the initial liquid layer temperature and thickness. If the surface is initially dry, the user can specify arbitrary values. Each of these values has associated with it a time function. This is used to indicate changes in these values due to external influences. The final value in this data set is the identification number of the time function which controls the condition of the surface. This can have one of three values at any time. A value of 0 if the surface is dry, a value of 1 if the surface is wet and a value of 2 if the surface is wet and the liquid layer is continuously added to. The manner in which these three different cases are analysed is described in the overview presented above.

FLUX BOUNDARY CONDITION
MASS TRANSFER

Data Set 13.2.4.4.2

Gas-liquid layer characterisation

FTINIT FLTHK NFTTF NFLTF NMTTR

FTINIT the initial temperature of the liquid layer. It has a time function associated with it to enable the user to control the conditions on the side

FLTHK the initial thickness of the liquid layer. It has a time function associated with it for the same reason as the initial temperature.

NFTTF the identification number of the time function associated with the initial liquid layer temperature

NFLTF the identification number of the time function associated with the liquid layer thickness

NMTTF the identification number of the time function which controls the condition of the surface. This time function can only have three values

0 if the surface is dry

1 if the surface is wet

2 if the surface is wet and the liquid layer is continually replenished..

The next data set requires the specification of the molecular masses of both the non-condensing gas and the liquid.

FLUX BOUNDARY CONDITION
MASS TRANSFER

Data Set 13.2.4.4.3

Liquid and gas molecular masses

```
WTMOLL   WTMOLG
```

WTMOLL the molecular mass of the liquid.

WTMOLG the molecular mass of the gas.

The next data set requires the specification of the constants used in Antoine's equation for the calculation of the vapour pressure of the liquid. These values can be found in almost any book containing lists of physical constants. For instance Himmelblau or Perry. As the pressure calculated using this equation is often in non-metric units, a conversion factor to alter the units to atmospheres must be specified.

FLUX BOUNDARY CONDITION
MASS TRANSFER

Data Set 13.2.4.4.4

Saturated vapour pressure prediction data

A B C D

A, B, C constants in the Antoine equation for saturated vapour pressure. The equation has the form $\ln(p) = A - B/(C + T)$ where T is the temperature and p the saturated vapour pressure. The units of temperature must be degrees Celsius.

D conversion factor by which the pressure calculated using the Antoine equation must be multiplied to ensure that the units of pressure are atmospheres.

unit conversion			multiply by
bar	to	atm	0.9869
Pa	to	atm	$9.8692\text{x}10^{-6}$
psia	to	atm	0.068046
mm Hg	to	atm	$1.31579\text{x}10^{-3}$

The next data set requires the specification of various parameters used in Watson's correlation for the determination of the latent heat of vaporisation at various temperatures. This data includes the critical temperature of the liquid, a reference temperature and the latent heat of vaporisation at this reference temperature.

FLUX BOUNDARY CONDITION
MASS TRANSFER

Data Set 13.2.4.4.5

Latent heat of vaporisation prediction data

```
TCRIT   TREF   HTREF
```

TCRIT the critical temperature of the liquid-gas system (K)

TREF a reference temperature (K)

HTREF a reference latent heat of vaporisation of the liquid-gas system (kJ/kgK). These values are for use in the Watson correlation which adjusts the latent heat according to temperature

The following group of data sets require the specification of various liquid physical properties which are piecewise linearly dependent on temperature. These physical properties include:

- liquid thermal conductivity
- liquid density
- liquid bulk viscosity
- liquid heat capacity
- liquid volume expansivity
- liquid-gas mass transfer diffusivity

The only physical property which has not been described in the convection boundary condition section is the mass transfer diffusivity. This can be defined as an measure of the rate of diffusion of a vapour through a gas. It can be found in standard texts such as Perry. These data sets are the same as their analogue in the convection boundary condition section and are not described.

FLUX BOUNDARY CONDITION
MASS TRANSFER
LIQUID PHYSICAL PROPERTIES

Data Set 13.2.4.4.6

Liquid physical properties control data

```
NKLPTS  NRLPTS  NVLPTS  NCLPTS  NELPTS  NDLPTS
```

NKLPTS the number of data pairs required for the piecewise linear definition of the temperature dependence of the liquid thermal conductivity

NRLPTS the number of data pairs required for the piecewise linear definition of the temperature dependence of the liquid density.

NVLPTS the number of data pairs required for the piecewise linear definition of the temperature dependence of the liquid bulk viscosity

NCLPTS the number of data pairs required for the piecewise linear definition of the temperature dependence of the liquid specific heat capacity.

NELPTS the number of data pairs required for the piecewise linear definition of the temperature dependence of the liquid thermal coefficient of volume expansivity

NDLPTS the number of data pairs required for the piecewise linear definition of the temperature dependence of the gas-liquid mass transfer diffusivity.

FLUX BOUNDARY CONDITION
MASS TRANSFER
LIQUID PHYSICAL PROPERTIES

Data Set 13.2.4.4.6.lr

Piecewise linear liquid thermal conductivity

NKLPTS of this data set are required

```
Ki    TEMPi
```

Ki the value of the liquid thermal conductivity at temperature TEMPi. These values must be entered in order of increasing temperature

TEMPi the temperature associated with Ki

FLUX BOUNDARY CONDITION
MASS TRANSFER
LIQUID PHYSICAL PROPERTIES

Data Set 13.2.4.4.6.2r

Piecewise linear liquid density

NRLPTS of this data set are required

```
RHOi    TEMPi
```

RHOi the value of the liquid density at temperature TEMPi. These values must be entered in order of increasing temperature

TEMPi the temperature associated with RHOi

FLUX BOUNDARY CONDITION
MASS TRANSFER
LIQUID PHYSICAL PROPERTIES

Data Set 13.2.4.4.6.3r

Piecewise linear liquid bulk viscosity

NVLPTS of this data set are required

```
VISCi    TEMPi
```

VISCi the value of the liquid bulk viscosity at temperature TEMPi. These values must be entered in order of increasing temperature

TEMPi the temperature associated with VISCi

FLUX BOUNDARY CONDITION
MASS TRANSFER
LIQUID PHYSICAL PROPERTIES

Data Set 13.2.4.4.6.4r

Piecewise linear liquid heat capacity

NCLPTS of this data set are required

```
Cpi    TEMPi
```

Cpi	the value of the liquid specific heat capacity at temperature TEMPi. These values must be entered in order of increasing temperature
TEMPi	the temperature associated with Cpi

FLUX BOUNDARY CONDITION
MASS TRANSFER
LIQUID PHYSICAL PROPERTIES

Data Set 13.2.4.4.6.5r

Piecewise linear liquid coefficient of volume expansivity

NELPTS of this data set are required

```
EXPi    TEMPi
```

EXPi the value of the liquid thermal coefficient of volume expansivity at temperature TEMPi. These values must be entered in order of increasing temperature

TEMPi the temperature associated with EXPi

FLUX BOUNDARY CONDITION
MASS TRANSFER
LIQUID PHYSICAL PROPERTIES

Data Set 13.2.4.4.6.6

Piecewise linear fluid diffusivity

NDLPTS of this data set are required

```
DIFFi    TEMPi
```

DIFFi the value of the liquid vapour diffusivity in the gas at temperature TEMPi. These values must be entered in order of increasing temperature

TEMPi the temperature associated with RHOi

Which of the following data sets are relevant is determined by the value of MVCTYP. If it is equal to 1 then the heat transfer coefficient between the liquid and the gas is entered explicitly as a piecewise linear function of temperature.

FLUX BOUNDARY CONDITION
MASS TRANSFER
TYPE 1

Data Set 13.2.4.4.7.1

Liquid-gas heat transfer coefficient control data

This data set together with the next two is required if MVCTYP = 1. If MVCTYP <> 1 go to data set 13.2.4.4.7.2

NHMPTS

NHMPTS the number of data pairs required for the piecewise linear definition of the heat transfer coefficient

FLUX BOUNDARY CONDITION
MASS TRANSFER
TYPE 1

Data Set 13.2.4.4.7.1.1r

Piecewise linear heat transfer coefficient

NHMPTS of this data set are required

```
HTCi    TEMPi
```

HTCi the value of the heat transfer coefficient at temperature TEMPi. It refers to the surface temperature of the water, not the film temperature. These values must be entered in order of increasing temperature

TEMPi the temperature associated with HTCi

FLUX BOUNDARY CONDITION
MASS TRANSFER
TYPE 1

Data Set 13.2.4.4.7.1.2

Ambient temperature of the gas

This is the final data set in the data set group associated with MVCTYP = 1. Go to data set 13.2.4.4.8

TAMBM

TAMBM the bulk temperature of the gas under consideration.

If MVCTYP is equal to 2 then the heat transfer coefficient must be determined by correlation. This requires the specification of physical property data in the same way as for the convection boundary condition. For more information, see the relevant part of the convection boundary condition section.

FLUX BOUNDARY CONDITION
MASS TRANSFER
TYPE 2
GAS PHYSICAL PROPERTIES

Data Set 13.2.4.4.7.2

Gas physical properties control data

This data set together with the next six is required if MVCTYP = 2.

NKGPTS	NRGPTS	NVGPTS	NHGPTS	NEGPTS

NKGPTS - the number of data pairs required for the piecewise linear definition of the gas thermal conductivity

NRGPTS the number of data pairs required for the piecewise linear definition of the gas density

NVGPTS the number of data pairs required for the piecewise linear definition of the gas viscosity

NRGPTS the number of data pairs required for the piecewise linear definition of the gas specific heat capacity

NEGPTS the number of data pairs required for the piecewise linear definition of the gas thermal coefficient of volume expansivity

FLUX BOUNDARY CONDITION
MASS TRANSFER
TYPE 2
GAS PHYSICAL PROPERTIES

Data Set 13.2.4.4.7.2.1r

Piecewise linear gas thermal conductivity

NKGPTS of this data set are required

```
Ki    TEMPi
```

Ki the value of the gas thermal conductivity at temperature TEMPi. These values must be entered in order of increasing temperature

TEMPi the temperature associated with Ki

FLUX BOUNDARY CONDITION
MASS TRANSFER
TYPE 2
GAS PHYSICAL PROPERTIES

Data Set 13.2.4.4.7.2.2r

Piecewise linear gas density

NRGPTS of this data set are required

```
RHOi    TEMPi
```

RHOi the value of the gas density at temperature TEMPi. These values must be entered in order of increasing temperature

TEMPi the temperature associated with RHOi

FLUX BOUNDARY CONDITION
MASS TRANSFER

TYPE 2

GAS PHYSICAL PROPERTIES

Data Set 13.2.4.4.7.2.3r

Piecewise linear gas viscosity

NVGPTS of this data set are required

VISCi	TEMPi

VISCi the value of the gas viscosity at temperature TEMPi. These values must be entered in order of increasing temperature

TEMPi the temperature associated with VISCi

FLUX BOUNDARY CONDITION
MASS TRANSFER
TYPE 2
GAS PHYSICAL PROPERTIES

Data Set 13.2.4.4.7.2.4r

Piecewise linear gas heat capacity

NHGPTS of this data set are required

```
Cpi    TEMPi
```

Cpi the value of the gas specific heat capacity at temperature TEMPi. These values must be entered in order of increasing temperature

TEMPi the temperature associated with Cpi

FLUX BOUNDARY CONDITION
MASS TRANSFER
TYPE 2
GAS PHYSICAL PROPERTIES

Data Set 13.2.4.4.7.2.5r

Piecewise linear gas coefficient of volume expansivity

NEGPTS of this data set are required

```
EXPi    TEMPi
```

EXPi the value of the gas specific heat capacity at temperature TEMPi. These values must be entered in order of increasing temperature

TEMPi the temperature associated with EXPi

FLUX BOUNDARY CONDITION
MASS TRANSFER
TYPE 2

Data Set 13.2.4.4.7.2.6

Gas-liquid heat transfer coefficient correlation control data

This is the final data set in the data set group associated with MVCTYP = 2.

TAMBM	CHRLNM	GSVLCM	MCORID

TAMBM the bulk temperature of the gas under consideration

CHRLNM the characteristic length of the surface with which the fluid under consideration is in contact. This is **not** the length or width of a single element.

GSVLCM the bulk velocity of the gas under consideration

MCORID the identification number of the heat transfer coefficient correlation to be used in this analysis. A choice of possible correlations can be found in the convection section

The data in the next data set is used to specify how to determine the gas-liquid interface temperature. As discussed previously, it can be a weighted average value or be determined from a modified lumped parameter analysis.

FLUX BOUNDARY CONDITION
MASS TRANSFER

Data Set 13.2.4.4.8

Interfacial temperature approximation control

LAFLG

LAFLG a flag to control which method is to be used to approximate the temperature of the gas-liquid interface.

1 a weighted average value between the liquid layer and the gas bulk temperatures

2 a value determined from a modified lumped parameter analysis

The next two data sets require the specification of the weighting value for the averaged interface temperatures. These values need only be entered if the appropriate models were selected. Both should have values between 0 and 1.

FLUX BOUNDARY CONDITION
MASS TRANSFER

Data Set 13.2.4.4.8.1

Liquid-gas interface weight factor

This data set is required if LAFLG = 1.

FACT2

FACT2 the weight factor used to approximate the gas-liquid interface temperature.

FLUX BOUNDARY CONDITION
MASS TRANSFER

Data Set 13.2.4.4.9

Liquid-solid interface weight factor

This data set is required if MTRMDL = 1. This is the last data set required in the group of data sets of which there must be NFLDM repetitions. If the number of repetitions is less than NFLDS then return to data set 13.2.4.4.1.

FACT1

FACT1 the weight factor used to approximate the solid-liquid interface temperature

The following data set is the last in the mass transfer boundary condition section and group of data sets used to specify properties and parameters for each gas-liquid system. It is only required if the solid-liquid interface temperature is to be determined using the convection assumption MTRMDL = 2. It requires the specification of the correlation identification number for the heat transfer coefficient as well as a characteristic length for use in that correlation.

FLUX BOUNDARY CONDITION
MASS TRANSFER

Data Set 13.2.4.4.10

Liquid layer convection condition data

This data set is required if MTRMDL = 2. This is the last data set required in the group of data sets of which there must be NFLDM repetitions. If the number of repetitions is less than NFLDS then return to data set 13.2.4.4.1.

LCORID	CLLEN

LCORID the identification number of the heat transfer coefficient correlation for the liquid layer. This number must be between 1 and 3 as it is assumed that only natural convection occurs in the liquid layer.

CLLEN the characteristic length for use in the selected heat transfer coefficient correlation. It is not the same as the characteristic length used for forced gas convection and can be as small as a single element length.

It should be emphasised that the mass transfer boundary condition model is very much an approximation and is designed to take into account the effects of condensation or evaporation rather than to model these processes accurately

4.8.7 Radiation boundary condition

Heat transfer by radiation is unlike the previously discussed modes of heat transfer in that it is due to the interaction of electromagnetic waves with the molecules of some medium which can be either a fluid or a solid. In this interaction the molecules of the medium absorb energy from the electromagnetic waves and their increased energy level manifests itself as an increase in the temperature of the medium. A medium not only absorbs but also emits radiant energy in quantities proportional to its absolute temperature.

The way in which solid and fluid media interact with radiant energy differs in that only a thin surface layer of molecules in a solid absorbs the electromagnetic waves while the more mobile molecules of a fluid all interact with the electromagnetic waves passing through them.

Apart from the radiation it emits, a medium can transmit, absorb or reflect impinging radiation and the extent to which it does one or more of these is a function of the nature and temperature of the medium interacting as well as the wavelength of the radiation. Generally, solids absorb and reflect but do not transmit while fluids do all three.

Consider initially the interaction of radiation with a solid.

Radiation falling on a surface can be reflected in two ways - specularly or diffusely. Diffuse reflection implies that impinging radiation is reflected uniformly in all possible directions which means that the intensity of the reflected radiation is uniform. This is assumed to be the case in the radiation analysis carried out by the program.

A surface can exhibit black body, white body or gray body behaviour. The first implies that all impinging radiation is absorbed, the second that all impinging radiation is reflected.

As has been mentioned before, the radiation emitted by a medium is a function of its temperature. In the case of a black body, the energy leaving the surface, the emissive power, can be expressed by the Stefan-Boltzmann Law

$$E_b = \sigma T^4$$

Where T = temperature of the surface in Kelvin (K)
σ = Stefan-Boltzmann constant; $5.699 . 10^{-8}$ (W/m^2K^4)
E_b = black body emissive power (W/m^2)

As most bodies are not black, an expression is required to determine their emissive powers. This can be written as:

$$E = \int_0^\infty \epsilon_\lambda E_b d\lambda$$

where ϵ_λ = the emissivity and is a function of the wavelength of the radiation
λ = the wavelength of the radiation (m)
E = gray body emissive power (W/m^2)

The gray body assumption, which greatly simplifies radiation analysis, entails assuming that emissivity, absorbtivity, reflectivity and transmissivity of media are not functions of wavelength. A consequence of the gray body assumption is that Kirchoff's Law leads to the equations:

$$\epsilon = \alpha$$
$$E = \epsilon\sigma T^4$$

where ϵ = emissivity
α = absorbtivity
E = emissive power (W/m^2)
σ = Stefan-Boltzmann constant ; $5.699 . 10^{-8}$ (W/m^2K^4)
T = temperature (K)

In the radiation analysis performed by the program the gray body assumption is used.

In addition to surface temperature and emissivity, the relative position of surfaces exchanging radiant energy influences the net energy transfer. To illustrate this the user should visualise two flat plates, surfaces 1 and 2, in some random orientation with respect to each other. Select a point on surface 1 and, using it as the centre, construct an imaginary hemisphere on the surface. The radiant energy passing through the hemisphere has the same value at every point on its surface. Now expand the hemisphere until it completely encloses surface 2. It can be seen that not all of the radiation leaving the point on surface 1 is intercepted by surface 2. Indeed not all of the radiation leaving surface 1 is 'seen' by surface 2. Clearly, to determine the influence of surface 1 on surface 2, some method must be found to predict how much of the radiant energy leaving surface 1 is received by surface 2. This leads to the concept of 'shape factors'. The shape factor between surface 1 and surface 2, denoted F_{1-2}, is the ratio of the energy received by surface 2 from surface 1 and the total radiant energy emitted by surface 1. The determination of these shape factors for use in determining the net radiant energy exchange between multiple surfaces is the most complicated aspect of radiation analysis. It is possible to determine analytically shape factors between surface pairs which are parallel or which have a common side, and equations for this are available in many texts on radiation. It is also possible to determine the shape factors for a slightly larger set of configurations using the various axioms of shape factor algebra. However, this is not easy to implement in a program.

There are a number of numerical methods for determining shape factors such as the Monte Carlo method. The method implemented in the program is known as the method of ray casting and is presented in a paper by Maxwell et al. A description of the method follows.

Consider two arbitrarily oriented flat surfaces, surfaces 1 and 2. Assume that F_{1-2} is to be determined. Select a point on surface 1, dA_1, and use it as the centre of a hemisphere which completely encloses surface 2. Project a large number of lines from dA_1 which intercept the surface of the hemisphere and which touch different points on the perimeter of surface 2. Effectively, this process marks out the shadow of surface 2 on the surface of the hemisphere. Next, map the shadow area on the hemispherical surface onto the plane containing surface 1.

The ratio of this projected area and the area of the circle formed by the intersection of the plane containing surface 1 and the hemisphere, is the shape factor between the point dA_1 and surface 2. One needs to repeat this procedure over the entire area of surface 1 in order to obtain F_{1-2}.

This method is implemented in the program by sub-dividing each surface into a large number of sub-areas and casting a finite number of rays from the centre of each sub-area. The final shape factor is the arithmetic average of the sum of the equally sized sub-area shape factors. The accuracy of the shape factors generated using this method is improved by using a larger number of sub-areas and by casting more rays. However, this can become computationally very expensive and a compromise between analysis time and accuracy demands must obviously be made.

As has been intimated before, radiant heat exchange is highly dependent on the geometrical layout of the interacting surfaces. The importance of the positions of surfaces has been dealt with from the point of view of surface pairs when calculating shape factors. However, for the purposes of finite element analysis, additional attention must be given to the layout of surfaces. Firstly, in some problems it is possible that some of the surfaces are part of the mesh whereas others are separate, merely acting as energy sources or sinks. It is necessary to distinguish between these different types of surfaces and to this end surfaces are divided up into those which are active - the former - and those which are passive - the latter. It is also possible that many surfaces interact together but that some do not 'see' other surfaces. To deal with this possibility, the concept of clusters of surfaces has been adopted. It is also possible that within a surface cluster a number of the passive surfaces do not see each other. As a result the user is able to specify to the program those surfaces which do not see each other within a cluster.

As has been mentioned before, active surfaces are those which form the various edges of the body in which the temperature distribution is being modelled. In any radiation analysis it is possible for all the surfaces to be active. On the other hand there must always be at least one active surface in a cluster. As the user is aware, the program analyses two dimensional continua, ignoring possible temperature gradients in the third dimension. However, the body is assigned a thickness in the third direction for the purpose of determining the areas of the edges. An edge is converted into a rectangular surface for the purposes of radiation analysis in the following way. Assume that the coordinates of the edge nodes lie in the x-y plane. Assign a common z-axis coordinate to each edge node and project lines in the positive and negative z-axis directions through the extreme nodes. The projected lines must be of the same length as the thickness of the domain and an extreme node must be at the mid-point of each projected line. The four ends of the projected lines constitute the corners of the rectangular surface. It is important that a global coordinate system be used for both active and passive surfaces. This process is illustrated in the diagram below.

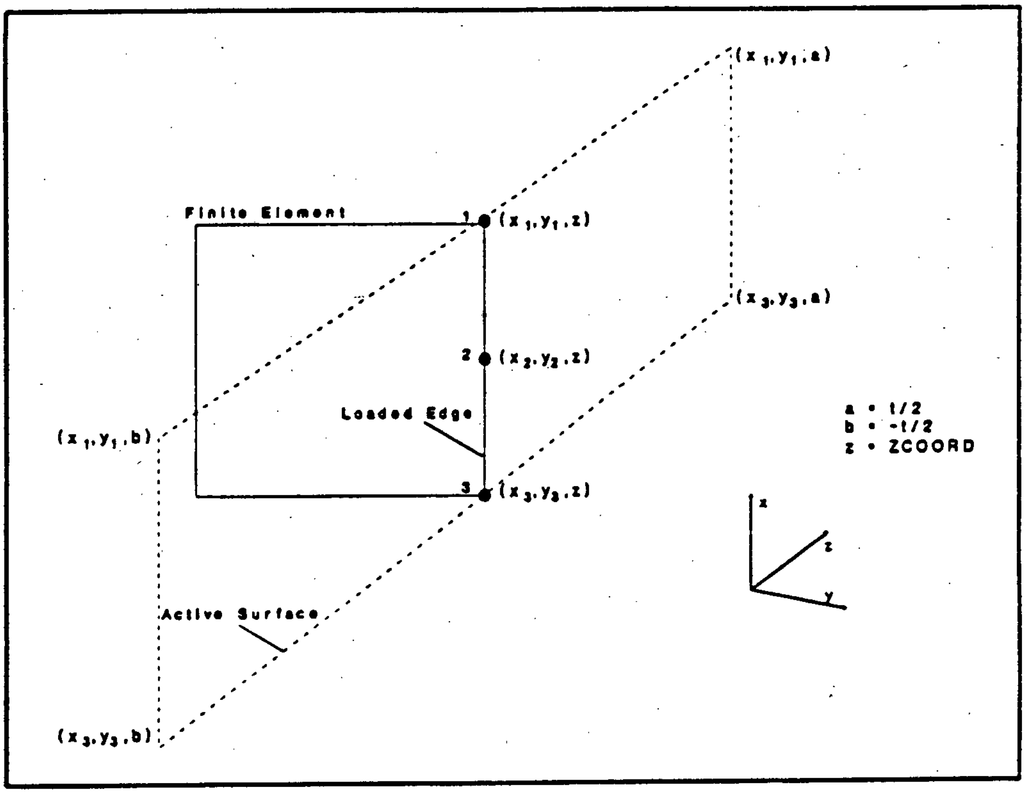


Figure 7 The relationship between an active surface and its finite element edge.

As passive surfaces are not part of the finite element mesh, they do not have nodes associated with them. For this reason the coordinates of the corners of all passive surfaces must be explicitly specified. Passive surfaces must be rectangular planes.

Great care must be taken by the user when defining the passive surfaces. It is altogether possible that some of these surfaces might overlap or occlude each other. As the program does not check for these eventualities itself, the user must sub-divide the relevant surfaces into 'seen' and partly 'unseen' portions and then specify which surfaces - both active and passive - in the cluster do not see the 'unseen' surfaces.

In any physical situation it is unrealistic to assume that the groups of interacting surfaces are unaffected by their surroundings. For this reason a final type of surface is included - an environment surface. A surface of this type can be treated like a passive surface.

In addition to radiant heat transfer between solid surfaces separated by a vacuum it is also possible to include the effects of a non-opaque, absorbing medium. As a rule, the temperatures at which radiant heat transfer is significant are sufficiently high that the fluids of interest are in the gaseous phase. It is a general rule that non-polar gases are effectively transparent to thermal radiation. It is not always easy to find emissivity and transmissivity data for gases and the most commonly available data is for carbon dioxide and water vapour. This information is generally presented graphically and a good source of data can be found in Hottel. As the prediction of gas emissivity data is difficult to include in a program, the onus is on the user to determine these values himself.

Having presented an overview, it is now appropriate to describe the various data sets required for radiation analysis.

The first data set requires the specification of the number of surface clusters, the number of passive surfaces, the number of non-opaque media between surfaces and the number of environments. It also requires the specification of how the shape factors and surface areas are to be obtained. The options are:

- shape factor and area calculation without storage for subsequent analyses
- shape function and area calculation with storage for subsequent analyses
- retrieval from storage of the shape factors and areas which were evaluated during a previous analysis

When using the third option the problem under consideration must be the same as that for which the stored shape factors and areas were calculated. It should be noted that only the shape factors and areas for real active and passive surfaces are stored. It is, therefore, possible to include or exclude environments without effectively changing the problem from a shape factor point of view. The final data value in the data set is used to combat possible analysis instabilities. These instabilities can arise from temperature differences between active and passive surfaces which are too large. Specifying a value of FLXFAC enables the user to control the interim value of the passive surface temperatures. It is generally required for steady state analyses and seldom for transient analyses.

FLUX BOUNDARY CONDITIONS RADIATION

Data Set 13.2.5.1

Radiation condition control data

NCLUST	**NSURFP**	**NTRANP**	**NENV**	**NFILE**	**FLXFAC**

NCLUST number of distinct groups of surfaces interacting in radiative heat transfer. These groups are known as clusters.

NSURFP total number of passive surfaces. It is possible for the model to be made up entirely of active surfaces.

NTRANP total number of non-opaque media interacting in radiative heat transfer. This variable can have a value of zero only when all surfaces are separated by vacuum. If only one cluster has some transparent medium separating its surfaces, then the transmissivities and emissivities of all surfaces must be specified.

NENV total number of environments seen by the clusters

NFILE control parameter for shape factor and surface area input, output and/or generation

0 shape factors and areas are calculated. No storage of the calculated values

1 shape factors and areas are calculated and written to a specified file.

2 shape factors and areas are retrieved from a specified file. No calculations are done.

FLXFAC a parameter governing the shape of the growth/decay curves used to model the temperatures of the environments and passive surfaces. These temperatures vary according to the following formula:

$$T = Ts(1.0 - e(-IITER/FLXFAC))$$

when the scaling factor enclosed in brackets has a value greater or equal to 0.99, then the temperature is set to its actual value.

It is not always necessary to use this method of scaling. Setting FLXFAC to a value less than 0.0 prevents any scaling. Obviously FLXFAC must be non-zero. A value of 15 appears to work satisfactorily but it will vary from problem to problem.

The following data set is used to specify the name of the file to which shape factor data is to be written or from which it is to be retrieved.

FLUX BOUNDARY CONDITIONS RADIATION

Data Set 13.2.5.2

Shape factor storage information

This data set is required if NFILE > 0.

```
FILNAM
```

FILNAM the name of the data file which is assigned to the logical variable SHAPEFILE using the DCL ASSIGN command. This data file is used to store shape factor and surface area information. If data is to be retrieved from a data file, the file must obviously exist. Only data pertaining to real surfaces - ie active and passive surfaces - is stored in this file. So it is possible to include or exclude an environment in a subsequent analysis without having to regenerate the shape factors

The next three data sets are used to specify data pertaining to active surfaces. The first data set requires the specification of the node numbers of the nodes on an edge according to previously discussed conventions. The second data set requires the specification of the number of sub-areas into which the current active surface is to be subdivided. The number of rays to be projected from each sub-area must be entered next, followed by the third dimension coordinate of the nodes of the current edge. The next data set requires the specification of the emissivity of the current active surface. The user should note that this value must always be less then 1.0. These three data sets must be entered in sequence for each active edge. It should be noted that the program assigns an identification number to each active surface in the order in which they are specified.

FLUX BOUNDARY CONDITIONS
RADIATION
ACTIVE SURFACES

Data Set 13.2.5.3.1

Definition of an edge

This data set is the first in a group of data sets which is used for characterising active surfaces. NEGRAD repetitions of this group are required. The number of the last data set in this group is 13.2.5.3.3.

Ni

Ni the node numbers of the nodes which define the current active surface. The normal conventions with respect to node number specification apply.

FLUX BOUNDARY CONDITIONS
RADIATION
ACTIVE SURFACES

Data Set 13.2.5.3.2

Active surface plane characterisation

```
NSUBA    NRAYS    ZCOORD
```

NSUBA the number of sub-areas into which the plane is to be divided for ray-casting. It should be a number with an integer square root.

NRAYS the number of rays to be cast from the centre of each sub-area. It must be an integer multiple of 4 and greater than or equal to 8.

ZCOORD the third dimension coordinate of the nodes of the edge. The resulting plane extends a distance THICK/2 on either side of ZCOORD in the direction of the third dimension

FLUX BOUNDARY CONDITIONS
RADIATION
ACTIVE SURFACES

Data Set 13.2.5.3.3

Active surface emissivity

This is the last data set in the active surface definition group of data sets. If the number of repetitions of this group is less than NEGRAD, return to data set 13.2.5.3.1.

```
EMISSIV
```

EMISSIV the emissivity of the active surface.

The next three data sets are used for the specification of data pertaining to passive surfaces. The first data set requires the specification of the number of sub-areas and projected rays. The comments made above are applicable in this case as well. The next data set is used to specify the coordinates of the corners of the current passive surface. The final set in the group of three is used to specify the passive surface emissivity, temperature and temperature time function identification number.

Passive surfaces are also assigned identification numbers in the order in which they are entered and these numbers continue from where the active surface numbering sequence ended.

FLUX BOUNDARY CONDITIONS
RADIATION
PASSIVE SURFACES

Data Set 13.2.5.4.1

Passive surface ray-casting information

This is the first data set in a group of data sets used to characterise passive surfaces. There must be NSURFP repetitions of this group. The number of the last data set in this group is 13.2.5.4.3.

NSUBA NRAYS

NSUBA - the number of sub-areas into which the plane is to be divided for ray-casting. It should be a number with an integer square root.

NRAYS the number of rays to be cast from the centre of each sub-area. It must be an integer multiple of 4 and greater than or equal to 8.

FLUX BOUNDARY CONDITIONS
RADIATION
PASSIVE SURFACES

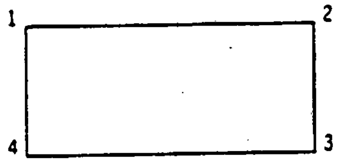
Data Set 13.2.5.4.2

Passive surface corner coordinates

SCOORDi

SCOORDi the three dimensional coordinates of each corner of the current passive surface. The coordinates of the corners should be input in one of the two possible orders resulting from following a clockwise or anti-clockwise path around the perimeter of the surface. Typically

Surface



Corner	Coordinates		
1	x_1	y_1	z_1
2	x_2	y_2	z_2
3	x_3	y_3	z_3
4	x_4	y_4	z_4

FLUX BOUNDARY CONDITIONS
RADIATION
PASSIVE SURFACES

Data Set 13.2.5.4.3

Passive surface radiation properties

This is the final data set in a group of data sets used to characterise passive surfaces. If the number of repetitions of this group is less than NSURFP, return to data set 13.2.5.4.1.

```
SEMIS   STEMP   NTIMF
```

SEMIS the emissivity of the current passive surface.

STEMP the temperature of the current surface.

NTIMF the time function number identification number associated with the surface temperature.

The next four data sets are used for the specification of surface cluster information. The first data set requires the specification of the number of surfaces - both active and passive - interacting within the current cluster. The second parameter to be specified defines the nature of the cluster. There are three possible configurations:

- an enclosure
- a group of surfaces which does not form an enclosure and which is affected by an environment

- a group of surfaces which does not form an enclosure which is not affected by an environment

The third parameter to be specified in this data set is the identification number of the environment seen by the current cluster. If no environment is seen, then set this value to zero.

FLUX BOUNDARY CONDITIONS
RADIATION
SURFACE CLUSTERS

Data Set 13.2.5.5.1

Cluster characterisation information

This is the first data set in a group of data sets used to characterise clusters. There must be NCLUST repetitions of this group. The number of the last data set in this group is 13.2.5.5.3

```
NCLSTS  NCONFG  NENVID
```

NCLSTS the number of surfaces interacting in this cluster.

NCONFG the identification number of the configuration type of this cluster:

0 an enclosure - a cluster of this sort cannot see an environment.

1 a group of surfaces which can see and be affected by an environment - not an enclosure

2 a group of surfaces which is not affected by its environment - not an enclosure

NENVID the identification number of the environment seen by this cluster. If the cluster is not affected by an environment then set this variable to zero.

The next two data sets within the group of four are used to identify which surfaces are members of the current cluster. The first data set requires specification of the identification number of the current surface and the number of surfaces with which this surface does not interact. The second data set requires the specification of the identification numbers of the surfaces which are not seen by the current surface.

FLUX BOUNDARY CONDITIONS
RADIATION
SURFACE CLUSTERS

Data Set 13.2.5.5.2.1

Cluster topology information

This is the first data set in a group of data sets used to characterise cluster topology. There must be NCLSTS repetitions of this group. The number of the last data set in this group is 13.2.5.5.2.2.

```
NCLSTI  NCLSEE
```

NCLSTI the identification number of the current surface in the global numbering scheme.

NCLSEE the number of surfaces in the cluster which the current surface does not see. It can take the value of zero.

FLUX BOUNDARY CONDITIONS
RADIATION
SURFACE CLUSTERS

Data Set 13.2.5.5.2.2r

Surfaces not seen by a surface

NCLSEE of this data set are required. This is the final data set in a group of data sets used to characterise cluster topology. If the number of repetitions of this group is less than NCLSTS return to data set 13.2.5.5.2.1

NCLCIDi

NCLCIDi - the global identification number of a surface which is not seen by the current surface.

The final data set in the cluster specification group is the identification number of the non-opaque, absorbing medium separating the surfaces in the current cluster.

FLUX BOUNDARY CONDITIONS
RADIATION
SURFACE CLUSTERS

Data Set 13.2.5.5.3

Non-opaque medium identification

This is the final data set in a group of data sets used to characterise clusters. If the number of repetitions of this group is less than NCLUST then return to data set 13.2.5.5.1

IDNOGS

IDNOGS the identification number of the non-opaque medium with which the surfaces in this cluster interact. If the surfaces in this cluster are not in contact with a non-opaque medium, this variable must have a minimum value of NTRANP + 1.

The next two data sets are used to define parameters and properties for the non-opaque media interacting in the radiant heat transfer. The first set requires the specification of the emissivity and transmissivity of the current medium from the point of view of radiation travelling from surface i to surface j. These values must be evaluated from information available in the literature. The next data set is used to specify the temperature of the medium and the time function identification numbers of the temperature,emissivity and transmissivity of the medium.

FLUX BOUNDARY CONDITIONS
RADIATION
NON-OPAQUE MEDIA

Data Set 13.2.5.6.1r

Non-opaque medium physical properties

This data set is required if NTRANP > 0. There can be at most (NEGRAD + NSURFP + NENV)^2 repetitions of this data set

```
ISURF   JSURF    GEMIS(i,j)   GTRANS(i,j)
```

ISURF identification number of the surface from which radiation is originating.

JSURF identification number of the surface which is receiving radiation.

GEMIS the emissivity of the medium between surfaces i and j.

GTRANS the transmissivity of the medium between surfaces i and j.

These values must be entered for every surface pair which interacts with a non-opaque medium. The values of 'i' and 'j' can have minimum and maximum values of 1 and (NEGRAD + NSURFP + NENV) respectively. If a surface pair is omitted from the list, GEMIS and GTRANS are assigned the values 0.0 and 1.0 by default. The order in which the surface pairs are entered is immaterial as long as the list of values terminates with SURF i = SURF j = (NEGRAD + NSURFP + NENV). For example:

1	1	GEMIS(1,1)	GTRANS(1,1)
1	2	GEMIS(1,2)	GTRANS(1,2)
2	1	GEMIS(2,1)	GTRANS(2,1)
1	3	GEMIS(1,3)	GTRANS(1,3)
3	1	GEMIS(3,1)	GTRANS(3,1)
	.	.	
	.	.	
8	2	GEMIS(8,2)	GTRANS(8,2)
	.	.	
	.	.	
1	6	GEMIS(1,6)	GTRANS(1,6)
	.	.	
	.	.	
N	N	GEMIS(N,N)	GTRANS(N,N)

where N = NEGRAD + NSURFP + NENV

FLUX BOUNDARY CONDITIONS
RADIATION
NON-OPAQUE MEDIA

Data Set 13.2.5.6.2r

Non-opaque medium information

NTRANP of this data set are required.

GTEMP	NGTMPTF	NGEMSTF	NGTRTF

GTEMP	the temperature of the medium (K)
NGTMPTF	identification number of the time function associated with the temperature of the medium
NGEMSTF	the emissivity of the medium time function identification number
NGTRFT	the transmissivity of the medium time function identification number.

The next data set, the last in the radiation section , is used to define the properties and characteristics of the environments. This set requires the specification of the environment's temperature, emissivity, surface area, non-opaque medium identification number and temperature time function identification number. An identification number is associated with each environment according to the order in which it is entered.

FLUX BOUNDARY CONDITIONS
RADIATION
ENVIRONMENTS

Data Set 13.2.5.7r

Environment information

NENV of this data set are required.

```
ETEMP   EEMIS   EAREA   NGID   NENVTF
```

ETEMP the temperature of the environment

EEMIS the emissivity of the environment.

EAREA the area of the environment. Basically, if it is not known, it must reflect the order of magnitude of the size difference between the largest surface area and the environment area.

NGID the identification number of the non-opaque medium associated with the environment. This value must correspond with that specified in the cluster information data group. If there is no medium associated with the current environment, this value must have a minimum value of NTRANP + 1.

NENVTF the identification number of the time function associated with the temperature of the environment.

4.9 Time function definition

Time functions enable the user to define the time dependency of many the parameters used in a heat transfer analysis. The use of time functions can greatly enhance both the users' control over and the complexity of any heat transfer analysis.

A time function is a piecewise linear curve relating a dimensionless factor to a time scale. The factor can have any reasonable, real value while the time scale must at least begin at the analysis starting time and terminate at the analysis finishing time. An example of a time function is presented in figure 8 below. It can be seen that each point on the curve - which represents the value of the factor at a particular time - is joined to its predecessor or successor by a straight line. There must be at least two defined points.

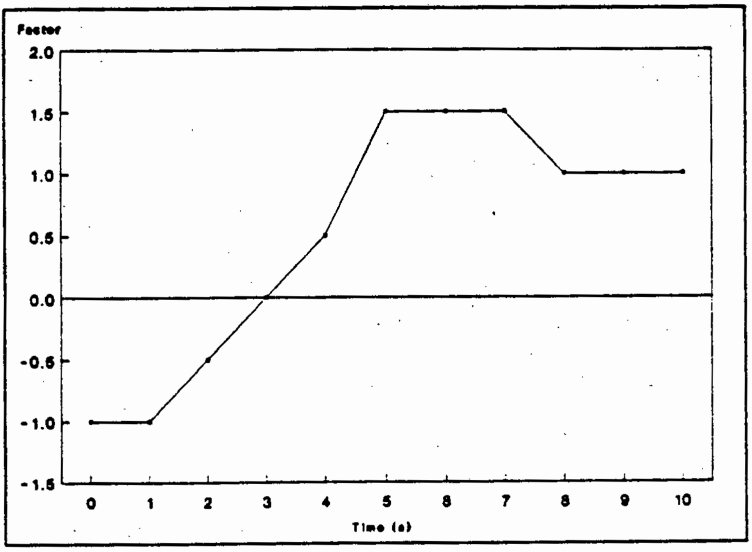


Figure 8 A piecewise linear time function curve.

Each time function has a unique identification number and any number of parameters can be associated with a particular time function by means of its identification number. Whenever a parameter which is linked to a time function is used in the program, the time function is evaluated at the current analysis time. The factor and the parameter are then multiplied together to yield the current value of the parameter. It is important to understand that the parameter's original value is conserved through the multiplication operation.

Each time function requires the specification of a number of data sets to define it completely. The first data set is used to specify the identification number of the time function and the number of data pairs which defines the shape of the time function curve. The second, repeated data set is used to specify these data pairs. Firstly, the time and then the value of the function at that time is entered. The data pairs must be entered in order of increasing time and the first data set should correspond to the starting time while the last should correspond to the finishing time of the analysis.

TIME FUNCTION CONTROL

Data Set 14.1

Time function control data

This is the first data set in group of time function definition data sets. NTFUN repetitions of this group are required

IDTF NPAIRS

IDTF time function identification number

NPAIRS the number of data pairs used to define this time function

TIME FUNCTION CONTROL

Data Set 14.2

Time function data pairs

NPAIRS of this data set are required. This is the final data set in the group of time function definition data sets. If the number of repetitions of this group is less than NTFUN then return to data set 14.1.

```
TIMEi    VALUEi
```

TIMEi	time at the i^{th} definition point
VALUEi	value of the time function at time ti

4.10 Data file terminator

In order to indicate, to the program that the correct end of file mark of the data file has been reached, the user must enter the terminator starting in the first column of the last line of the data file. It cannot be preceded by a blank line.

INPUT DATA TERMINATOR

Data Set 15

Input data termination string

```
END
```

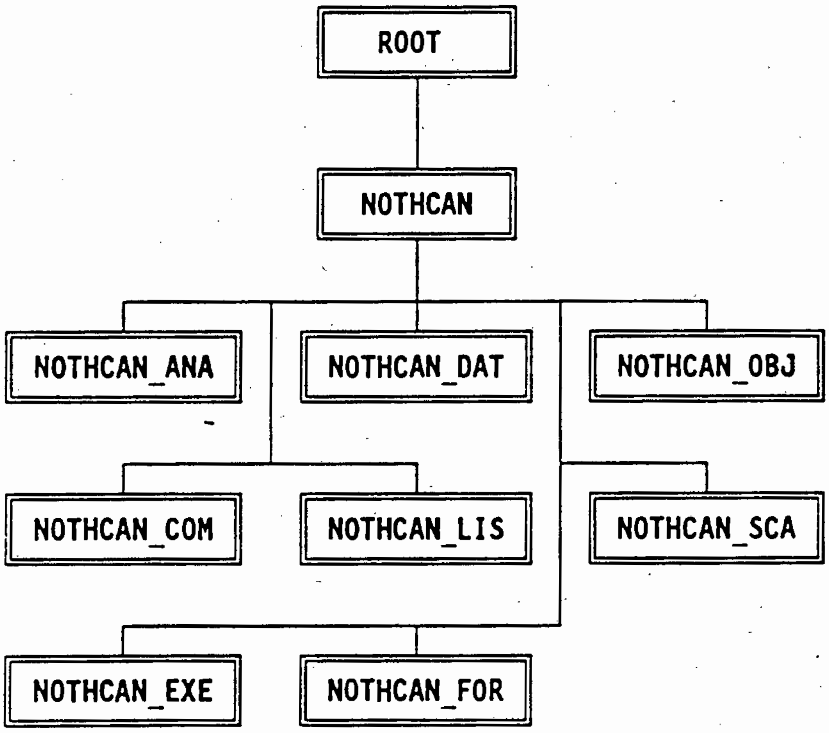
END this string indicates the end of the input data. It is essential for the correct execution of the program and must start in the first column and must be in upper case.

5. USING NOTHCAN

This section of the users' guide describes how to use the program. It is assumed that a data file has already been created. It describes the way in which the various files used by the program are stored, how to compile and link the program and how to execute it.

5.1 NOTHCAN's directory structure

The diagram below represents the relationship between the various directories in which the files associated with NOTHCAN are stored.



The function of each directory is explained below.

5.1.1 NOTHCAN_ANA

This directory is used to store the .ANA files which are generated by the Fortran compiler (with the /ANALYTICAL flag) for use by the Source Code Analyzer (SCA). Files are stored temporarily in this directory and so if the temporary storage facilities are used this directory is redundant.

5.1.2 NOTHCAN_DAT

This directory is used to store NOTHCAN data files (.DAT files), NOTHCAN result files (.OUT files) and NOTHCAN post-processing files (.PST files). It is probably advisable to create separate directories for each of these groups of files.

5.1 3 NOTHCAN_OBJ

This directory is used to store the compiled NOTHCAN source code files. These compiled files have the extension .OBJ. These relocatable object files are retained to enable 'intelligent' compilation which is explained in the 'Compiling and Linking NOTHCAN' section.

5.1.4 NOTHCAN_COM

This directory contains all the command procedures necessary for the running, compiling and linking of NOTHCAN. All the command procedures have the extension .COM. In addition to command procedures, this directory also contains a MMS description file which has the extension .MMS. Details of the MMS description file are presented in the 'Compiling and Linking NOTHCAN' section. A brief description of each file in this directory is given below.

IO_SETUP.COM : this file is used before executing NOTHCAN interactively. When NOTHCAN is being executed in batch mode the contents of this file must be duplicated in the command procedure which is to be SUBMITted.

In this command procedure directory paths are assigned to logical variables. For example,

```
$DEFINE  PDAT  CHE:[FRNLAN01.NOTHCAN.NOTHCAN_DAT].
```

These logical variables are then used to assign file names to **reserved** logical variables. For example,

```
$ASSIGN   PDAT:FILENAME.EXT   INDATA
```

The following table lists the reserved logical variables and their purpose.

<u>Logical Name</u>	<u>Function</u>
INDATA	data file name
OUTDATA	output file name
RESTART	restart file name
POSTFILE	post-processing file name
SHAPEFILE	shape factor file name

All of these logical variable must be defined whenever the program is executed.

NOTHCAN.COM : this command procedure is used either to print the source code of NOTHCAN or to compile and link the program. This command procedure can only be executed interactively. It accepts one of the following three command-line parameters: MAKE, BUILD_ALL or PRINT. When the command procedure is executing it prompts the user whether compilation for debugging or optimisation is desired. It invokes MMS description file NOTHCAN.MMS. More details of the use of this file are presented below.

NOTHCAN_BATCH.COM : this command procedure is similar to that described above but operates in batch mode. It does not accept any command-line parameters, nor does it prompt the user for any information. The user must edit this file in order to specify the options required.

NOTHCAN.MMS : this is the MMS description file which is used by both NOTHCAN.COM and NOTHCAN_BATCH.COM. It is described in more detail in the 'Compiling and Linking NOTHCAN' section.

5.1.5 NOTHCAN_EXE

This directory is used to store the executable file NOTHCAN.EXE as well as its associated map file NOTHCAN.MAP.

5.1.6 NOTHCAN_FOR

This directory is used to store the source code of NOTHCAN. It includes the Fortran source code - .FOR files - Fortran common block statements - .TXT files - and a text library file which contains all the .TXT files and is used when compiling NOTHCAN - .TLB file. One other file type is stored in this directory - a .LSE file. This file is a list of the subroutines used by NOTHCAN in the order in which they are called and is created by the Language Sensitive Editor (LSE) in conjunction with the Source Code Analyzer (SCA) library files.

5.1.7 NOTHCAN_SCA

This directory is used to store the library files created by the Source Code Analyzer (SCA) for use with the Language Sensitive Editor (LSE). This combination of the Vax Set is extremely helpful if the source code is to be modified.

5.1.8 NOTHCAN_LIS

This directory is used to store the source code listings which are generated when the source code is compiled. This directory will not contain any files unless a compile-time error has occurred.

5.2 Compiling and Linking NOTHCAN

NOTHCAN consists of over one hundred subroutines, each of which is stored as a file in the NOTHCAN_FOR directory. When one wishes to make a change to the program it is often necessary to edit several of the NOTHCAN subroutines. Then, in order to execute the program, one must compile the altered files and link them to form the final executable file NOTHCAN.EXE. To minimise the effort required to keep track of which subroutines need to be recompiled, use is made of the Module Management System package (MMS). Use of this system requires the creation of a description file (NOTHCAN.MMS in directory NOTHCAN_COM) in which the dependencies of the .EXE file on .OBJ files and .OBJ files on .FOR and .TLB files are explicitly stated. This description file is used by the MMS processor to compile all source code files which have a last-saved date more recent than their associated relocatable object files and then to link both old and new object files into an executable file.

MMS is not used directly but rather through a command procedure. There are in fact two command procedures to do the same job, one for interactive mode - NOTHCAN.COM - and the other for batch mode - NOTHCAN_BATCH.COM. Both of these command procedures are stored in the NOTHCAN_COM directory. Interactive compilation and linking is discussed first.

The command procedure NOTHCAN.COM accepts one of three command-line parameters - MAKE, BUILD_ALL or PRINT. MAKE instructs the command procedure to invoke MMS in such a way that only those source code files which have been changed since the last compilation are compiled. BUILD_ALL instructs the command procedure to invoke MMS in such a way that all the source code files are compiled whether or not they have been changed since the last compilation. The PRINT command-line parameter causes the source code to be printed. Whenever compilation occurs, the command procedure prompts the user whether he wants compilation with optimisation or debugging. It is important that all source code be compiled with the same option.

Currently it is assumed that the user always wants to use the Language Sensitive Editor (LSE) together with the Source Code Analyzer (SCA) which means that SCA library files must be updated and maintained during compilation. As this is a time consuming process, the user is advised to edit NOTHCAN.COM and NOTHCAN.MMS if he does not intend using the LSE.

The MAKE concept works extremely well for normal .EXE, .OBJ and .FOR files. However, in the case of .TXT and .TLB files it breaks down. This is unfortunate as the user is most likely to change the .TXT and .TLB files as it is in these files that the program's variable bounds are defined as well as the various common blocks used by the program. If the user makes any changes to any of the .TXT files, the following steps must be followed.

- Complete all changes.
- Replace the changed .TXT files in the text library file PROCLIB.TLB. If only one .TXT file has been changed, then that file can be replaced explicitly but if more than one .TXT file has been altered, the following command could be used, assuming that the current directory is NOTHCAN_FOR:

  ```
  $LIBRARY/REPLACE   PROCLIB.TLB   *.TXT
  ```
- Rename the updated text library file to PROCLIB.TLB;1.
- Compile the program using the BUILD_ALL option.

It is essential that whenever the .TXT files - and hence the .TLB file - are changed, the program be compiled using the BUILD_ALL option.

When MMS is invoked in the MAKE mode, it is generally reasonable from a time point of view to do it interactively. However, when the BUILD_ALL option is used it is virtually imperative to invoke MMS in batch mode. Hence the existence of the command procedure NOTHCAN_BATCH.COM.

This command procedure is basically the same as its interactive mode counterpart except that all compilation modes and options must be predefined by the user in the command procedure prior to executing it. For example, to BUILD the program with compilation for optimisation, the following statements must be included in the command procedure:

```
$ P1  =  "BUILD_ALL"
$ COM_OPT  =  "O"
$ CALL SUB_OPTIM
```

If, on the other hand, the user wishes to MAKE the program with compilation for debugging, the following statements must be included in the command procedure:

```
$ P1  =  "MAKE"
$ COM_OPT  =  "D"
$ CALL SUB_DEBUG
```

Once these changes had been made to the command procedure, it could be directed to batch execution as follows:

```
$SUBMIT/NOTIFY/NOPRINT/LOG=BUILD   NOTHCAN_BATCH.COM
```

All the directories used during the compiling and linking process are assigned to logical variables in both of these command procedures. If the user wishes to use different directories all he has to do is edit the logical variable definitions. These logical directory variables are:

Logical Variable	Purpose
OBJ$	relocatable object files
LIS$	source code listing files
FOR$	source code files
SCA$	SCA library files
EXE$	executable and map files
COM$	command procedures and MMS description files
DAT$	input data files
ANA$	.ANA files for use with SCA

It is sensible use temporary storage facilities for the ANA$ and LIS$ directories as they both occupy a significant amount of storage space.

Having compiled and linked NOTHCAN, the program can be used to analyse problems.

5.3 Executing NOTHCAN

NOTHCAN can be executed in either batch or interactive mode although interactive mode should be used only for small problems or if the debugger is being used.

If the program is being executed interactively then it is essential that the command procedure IO_SETUP.COM in directory NOTHCAN_COM be altered to correspond to the current problem and be executed initially. If the program is to be executed in batch mode then the contents of IO_SETUP.COM must be duplicated in the command procedure which is to be submitted to batch processing. A possible command procedure might be:

```
$ !---------------------------------------------------------
$ !:                                                       :
$ !:  Command procedure to execute NOTHCAN.EXE in          :
$ !:                                                       :
$ !:                  batch mode                           :
$ !:                                                       :
$ !---------------------------------------------------------
$ !
$ !    Define the logical directory names
$ !
$ DEFINE   PDAT   CHE:[FRNLAN01.NOTHCAN.NOTHCAN_DAT]
$ DEFINE   PEXE   CHE:[FRNLAN01.NOTHCAN.NOTHCAN_EXE]
$ !
$ !    Assign the file names to the reserved logical variables
$ !
$ ASSIGN   PDAT:LFBM28D.DAT    INDATA
$ ASSIGN   PDAT:LFBM28D.OUT    OUTDATA
$ ASSIGN   PDAT:LFBM28D.RST    RESTART
$ ASSIGN   PDAT:LFBM28D.PST    POSTFILE
$ ASSIGN   PDAT:LFBM28D.SHP    SHAPEFILE
$ !
$ !    Execute the program
$ !
$ RUN   PEXE:NOTHCAN.EXE
$ !
$ EXIT
```

This command procedure, NOTHRUN.COM say, could be executed using the following command:

```
$SUBMIT/NOTIFY/NOPRINT/LOG=NOTHRUN     NOTHRUN.COM
```

It could also be executed interactively using the following command:

```
$@NOTHRUN.COM
```

6. REFERENCES

GERALD, C.F. and WHEATLEY, P.O., 'Applied numerical analysis.', 3rd Ed., Addison-Wesley, (1984).

HIMMELBLAU, D.M., 'Basic priciples and calculations in chemical engineering.', 4th Ed., Prentice-Hall, New Jersey, (1982).

HOTTEL H.C. and SAROFIM A.F., 'Radiative heat transfer', McGraw-Hill, (1967).

MAXWELL, G.M., BAILEY, M.J. and GOLDSCHMIDT, V.W., 'Calculations of the radiation configuration factor using ray casting.', Computer Aided Design, v. 18, n. 7, (1986).

PERRY, R.H. and CHILTON, C.H., 'Chemical engineers' handbook.', 5th Ed, McGraw-Hill, (1983).

7. APPENDIX

Volume expansivity calculation

The coefficient of volume expansivity is a physical property used when attempting to estimate the heat transfer coefficient in the case of natural convection. It can be approximated by using the equation below which can be found in Perry.

$$\beta = \frac{\rho_1^2 - \rho_2^2}{2(t_2 - t_1)\rho_1\rho_2}$$

where β = coefficient of volume expansivity (K^{-1})
ρ = density (kg/m^3)
t = temperature (K)

Current NOTHCAN variable bounds

The various maximum values listed below can be found in the file PROCPA.TXT. It is possible for the user to alter these values and the procedure to follow is described in the 'Using NOTHCAN' section. It is essential that this procedure be strictly obeyed.

MPOIN	100	maximum number of nodes in the finite element mesh
MELEM	50	maximum number of elements within a single element group
MELET	50	maximum number of elements within all element groups
MVTEM	100	maximum number of nodes with prescribed temperature boundary conditions
MEGRP	2	maximum number of element groups
MEVAB	9	maximum number of nodes per element
MMATS	8	maximum number of different materials

MPROP	40	maximum number of different properties
MFRON	80	maximum frontwidth in the matrix solver
MTIMF	15	maximum number of time functions
MTIFV	50	maximum number of values per time function
MDOFN	3	maximum number of degrees of freedom
MFLDS	10	maximum number of heat exchange fluids
MBLKP	20	maximum number of I/O blocks
MCLUST	10	maximum number of surface clusters
MSURF	20	maximum number of active and passive surfaces
MRAYS	1001	maximum number of rays which may be cast
MSUBA	10000	maximum number of subareas into which a surface may be divided
MEDGE	25	maximum number of sides with boundary conditions