DIGITAL SIMULATION OF THERMAL REGENERATORS–A NEW APPROACH

By

IQTEDAR ASKARI ABDI
B.A. (HONOURS)

A Thesis submitted for an award of

Master of Applied Science

to

Department of Chemical Engineering

This thesis embodies the results
of supervised project work making up
all of the work for the degree.

THE UNIVERSITY OF ADELAIDE
ADELAIDE, SOUTH AUSTRALIA-5001
JULY—1980

CONTENTS

.. contd.

CONTENTS (contd.)

.. contd.

CONTENTS (contd.)

..

ABSTRACT

Thermal regenerators have been widely used in metal reduction and glass furnace industries for well over past 150 years. With the renewed interest in gas turbine and Stirling engines for land transport, it has become more desirable and necessary to study regenerative heat exchangers (i.e., thermal regenerators).

In this thesis on Digital Simulation of Thermal Regenerators, at first the basic principles and industrial significance of thermal regenerators are reviewed. A comparison between a recuperator and a regenerator is presented and then an outline of the scope and purpose of this work is presented. Here it is noted that early efforts are directed towards determination of thermal efficiency which is a useful parameter towards the convergence of method of solution.

The second chapter presents the physical assumptions, mathematical model and review of previous theories in regenerators. In this connection at first the "Open" methods of solutions are looked at. Then a number of "Closed" methods arc reviewed and it is noted that Nahavandi and Weinstein's (NW) closed method stands out to be the most reliable of the closed methods. It is also noted that open methods are in general very time consuming and

may not give as accurate results as closed methods. Hence a closed method based on NW's work was required to be developed.

Therefore, the third chapter is devoted to development of a new method of solving the regenerator problem. This new method although based on NW's method however avoids the use of quadrature formulae (used by NW to evaluate complicated forms of Bessel functions) and uses numerical inversion of Laplace transforms. The method developed considers a general, non-symmetric, unbalanced regenerator whereas **it is noted that NW's method was developed** for a balanced, symmetric thermal regenerator. This proposed method is developed in terms of two normalized parameters q and z-scales separately. This was required as in the first case of q-scale the matrices to be inverted had terms which could become very large for large values of q. So z-scale where $z \varepsilon [0,1]$ was employed thus avoiding lack of precision and other problems as reported. It should be noted that full formulae for the various temperature profiles and other required parameters such as thermal efficiency are derived in this chapter for the two scales considered.

The fourth and final chapter describes the computer programs written and compares the results obtained. It is found that the results obtained compare very favourably

with those obtained by earlier workers such as Willmott and some of the results concerning the limiting steady-state profile are also found to be in agreement. After an analysis of ill-conditioning it is observed that ill-conditioning experienced by Willmott through Illiffe's method does not occur with the proposed method for the same parameters. Results of computer programs for single and double precisions using q and z-scale are compared and it is concluded that the proposed method being relatively superior than the existing methods reviewed will be useful in calculation of temperature profiles, thermal efficiency, etc. This work also includes a finite stage method which was developed earlier and is presented in Appendix A4.

...

STATEMENT

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma in any university and that, to the best of my knowledge and belief the thesis contains no material previously published or written by another person, except when due reference is made in the text.

Bangalore, India
July 20, 1980.

(IQTEDAR ASKARI ABDI)

ACKNOWLEDGEMENTS

I am deeply indebted to my supervisor Dr. C.P. Jeffreson for suggesting the problem and providing invaluable guidance during the course of this work. I am also grateful to him for his constant encouragement and immense patience with me.

I am thankful to Mr. Ee-Pan Chow of Chemical Engineering Department, Adelaide University, for his help on many aspects of this work.

The members of my family have provided me with moral support for which I remain ever indebted.

I thank the staff of School of Mathematics and Computer Studies, South Australian Institute of Technology for their immense consideration in adjusting my teaching timings so as to enable me to carry out part-time work on this project. I am particularly grateful to Prof. D.H. Lee in this regard.

Dr. A.A. Shamim, the Chairman, Computer Centre, Indian Institute of Science, Bangalore, India, and the staff at the Centre have extended full co-operation and encouragement to me in completion of this thesis. I am grateful to them for their kindness.

I am grateful to my colleagues at the Computer Centre, I.I.Sc., Mr. A.S. Balasubramanyam, Mr. A.K. Pal and

Mr. Mir Shaeq Ali in particular for help with the various computer print-outs required for this thesis.

My friends at I.I.Sc., Dr. Joy Mukhopadhyay, Dr. P.R. Sarode and Dr. M.K. Uppal helped with the corrections and proof reading, I offer them my thanks.

Last but not least, I am thankful to Mr. B.R. Seshadri for the excellent typing of the thesis which he did in spite of an extremely busy schedule.

Bangalore IQTEDAR ASKARI ABDI

NOMENCLATURE

A	regenerator heating surface area (m^2)
C	specific heat of heat storing solid matrix (J/Kg, deg C)
h	heat transfer coefficient (W/m^2, deg C)
$\bar{h}$	bulk heat transfer coefficient (W/m^2, deg C)
L	length of regenerator (m)
M	mass of heat storing solid matrix (Kg)
m	mass of fluid resident in regenerator (Kg)
P	length of period of operation (sec)
S	specific heat of fluid (J/Kg, deg C)
T_1	temperature of solid matrix (deg C)
t_1	fluid temperature (deg C)
T	normalized solid temperature (dimensionless)
t	normalized fluid temperature (dimensionless)
$\bar{t}$	steady state fluid temperature (deg C)
$\bar{T}$	steady state solid temperature (deg C)
W	flowrate of fluid through regenerator (Kg/sec)
x	distance from regenerator entrance in direction parallel to fluid flow (m)
y	time (sec)
z	normalized distance; $z = x/L$
θ	normalized time; $\theta = y(W/m)$
λ	the number of transfer units or reduced length parameter; $\lambda = hA/(WS)$.

π	the period of a cycle
α	the ratio of thermal capacitance of packing to that of fluid held in regenerator; $\alpha = MC/(mS)$.
q	$= \lambda z$
r	$= (\lambda/\alpha)\theta$
E	thermal efficiency
R	thermal ratio

Superscripts

$'$	refers to hot period
$''$	refers to cold period

Subscripts

in	refers to fluid entrance
out	refers to fluid exit

...

CHAPTER 1: INTRODUCTION

The Regenerative Heat Exchanger was first proposed by Stirling in 1816 in connection with his regenerative hot-air engine. Since then, the thermal regenerator has been widely used for the pre-heating of the blast for metal reduction furnaces and for the pre-heating of the combustion air for glass furnaces. With the possibility of gas turbine engines and Stirling engines now being explored as alternatives to the internal combustion engine for land transport, there is a renewed interest in the regenerative heat exchanger due to promise it shows in the effecting of fuel economy.

In this chapter, we first review the basic principles of Thermal Regenerators (Regenerative Heat Exchangers) and their industrial significance. Then a comparison between a regenerator and a recuperator is made. This is followed by the scope and purpose of present work.

1.1 Basic principles and industrial significance of thermal regenerators:

A thermal regenerator is a device which increases the efficiency of an industrial operation by storing 'waste' heat produced at one stage of the process and returning this heat to the system when required at a later stage. It effects the transfer of heat between two fluids, generally gases.

The simplest form of a thermal regenerator consists of a heat retaining solid called a 'matrix' or 'chequerwork'. The method of operation of regenerators is cyclic. A hot gas is passed through the heat absorbing (and storing) matrix for a period of time. The hot gas is then turned off at the end of this 'hot period' with the matrix now holding the heat transferred from the hot gas. This heat is then used to raise the temperature of a cold gas which is passed through the matrix in the opposite direction to the flow of the hot gas earlier. When the cold gas is turned off the end of the 'cold period' is reached. The total time thus involved in the hot period and cold period is known as a 'Complete Cycle' of the regenerator. These regenerators are termed 'Counterflow' since hot and cold gas flow in opposite directions.

It must be noted here that it is possible to find regenerators in which the two gases flow in the same direction, called 'Parallel Flow' regenerators. Where the gases flow at right angles to each other the regenerators are said to be 'Cross Flow'.

A complete cycle of operations consists of a pair of successive hot and cold periods, and after a sufficiently large number of such cycles, the temperature behaviour of the solid matrix becomes periodic, the period being the duration of the cycle. At this stage, the regenerator is said to have reached 'Cyclic equilibrium' which is independent of the initial temperature conditions within the regenerator.

Regenerators have remained significant in the steel industry for sometime. They are used to pre-heat the air for both open hearth and blast furnaces to improve the efficiency of the steel and iron making processes. The regenerators connected to a blast furnace are called 'Blast Heaters'. Two or more of these blast heaters working in conjunction can heat hourly 200,000 cubic metres of air, to approximately 1100°C cold gas (1500°C hot gas) inlet temperature. These regenerators are associated with high temperatures. For medium and low temperature regenerators, the packings i.e., solid matrix are very often metallic. They are designed to maximise the area of surface available for heat transfer whilst keeping the volume of the regenerator down to reasonable proportions. Sheet metal strips and beds of spheres arranged parallel to each other are two simple examples of such packings. These metallic regenerators are very efficient. In low temperature regenerators with a height of only 4 metres, a heat exchange efficiency of 98 to 99 percent is attained whilst in blast furnace stoves the efficiency is only about 80 to 85 percent. A Staggered Parallel or a By-Pass-Main system is normally used in blast furnace stoves in order to ensure a constant blast temperature. Regenerators are also being used in industries making use of large boilers, gas turbine engines etc.

1.2. Regenerator vs. Recuperator:

In a recuperator i.e., an ordinary heat exchanger two fluids of different temperatures flow continuously in spaces separated by a wall, and they exchange heat by convection at and conduction through the wall. On the other hand, a regenerator is built up of solids, which alternately store internal energy taken from the warmer fluid and release it to the colder fluid while the fluids pass, one at a time, each being in contact with the solids during a certain period of time. A diagrammatic representation of both a regenerator and a recuperator is presented in Figure (1.1).

The recuperator is operated continuously i.e., two gases are passed continuously along their appropriate channels and when the heat exchanger has been running for a sufficient length of time the temperatures of the separate output gases are constant. In contrast by the very nature of its construction it is not easy to provide even a continuous supply of heated gas using a single regenerator. At the very least two regenerators would be required to provide continuous heated gas. Even then it is impossible to operate the regenerators in order that the heated gas be constant in temperature.

In the recuperator, after operating for sufficient time, the temperatures within the heat exchanger become independent of time and are functions only of position down the bed (or body) of the recuperator, measured from the gas entrance. On the other hand, after the regenerator has been operating for a sufficient length of time

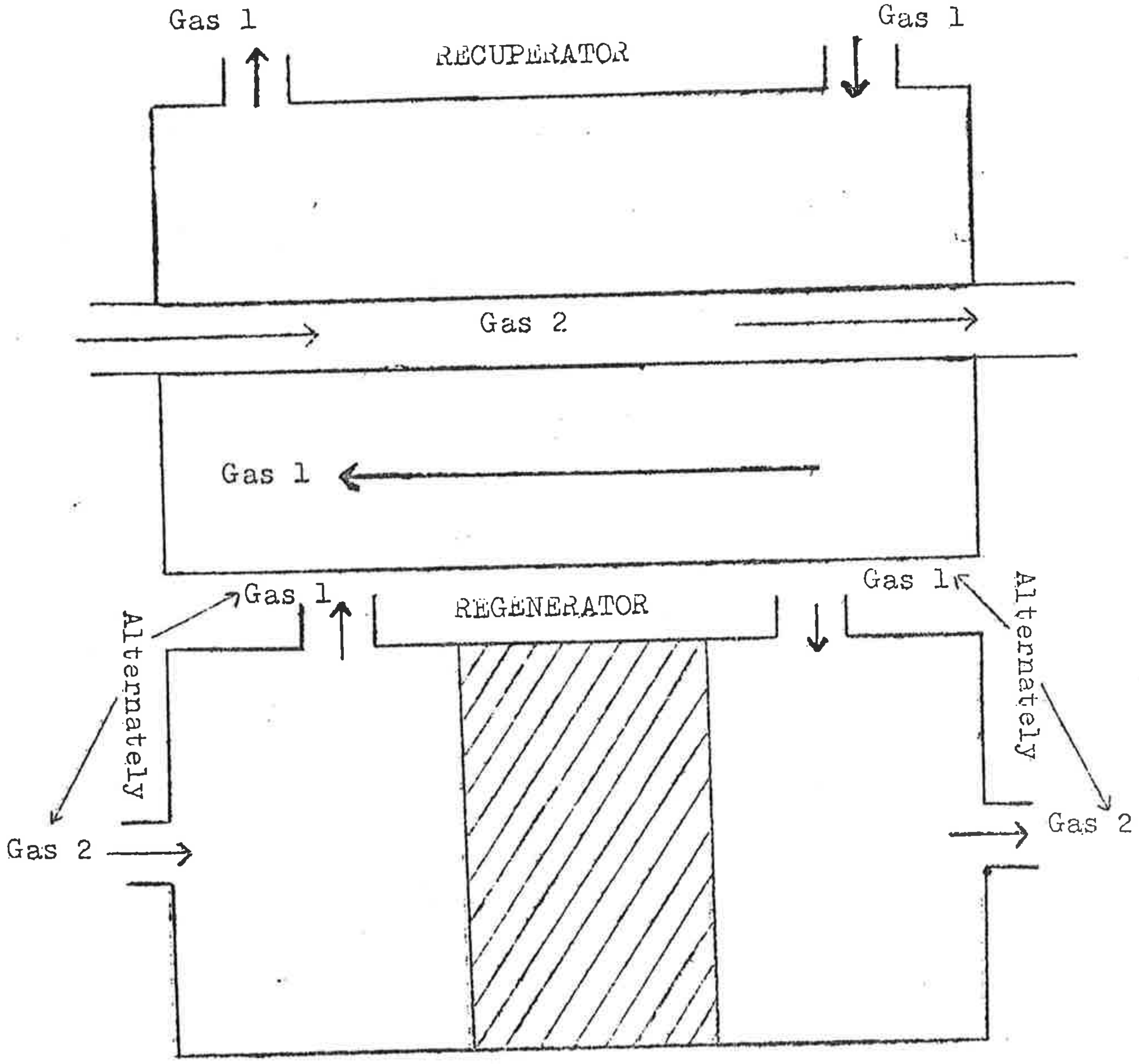


Figure (1.1) Recuperator and Regenerator both in counterflow under constant conditions, the reversing flows of the hot and cold gases through the same channels cause the temperatures within the regenerator to become periodic. The temperature of the gas or solid at any point in the regenerator is thus a function not only of position but also of time.

Thus, in the regenerator the heat is transferred under unsteady state conditions, namely, in a cyclic process, whereas the recuperator generally operates under steady-state conditions.

In the design and control of regenerators in blast furnace industry most work to date has been done assuming constant flowrate of gases, in order to compute the temperature distributions etc. when cyclic equilibrium has been reached. However, since heat transfer coefficients are strongly dependent on flowrate, the parameters of the system change during a blow if the flowrate is variable. In blast furnace stoves either for Staggered Parallel or By-Pass-Main systems a variable cold flowrate is necessary in order to ensure a constant blast temperature. Hence it is of interest to look into variable mass flow situations in regenerators.

1.3 Scope and purpose of present work:

Here we give a brief outline of present work, a detailed treatment follows in later chapters.

a) Early efforts are concentrated towards the determination of a real number the 'Thermal Efficiency'. Thermal efficiency is useful to indicate whether or not convergence has occurred.

'Open' methods such as those of Willmott [23,24] require the equations to be solved repeatedly for a zero initial temperature distribution until the cyclic equilibrium has been achieved or at least until the computed thermal efficiency is constant for successive cycles. Although the true criterion for cyclic equilibrium is that temperature profiles at identical stages of successive cold (or hot) blows should be the same to a specified accuracy, the achievement of constant thermal efficiency in successive cycles is also a good indication that equilibrium has been achieved. The open methods are in general very time consuming since many cycles may be required for convergence to take place for which computational time may be excessive. This is where thermal efficiency calculation is useful as after solving the equations only thermal efficiency is computed and this is done for successive cycles without obtaining the actual temperature profile and when thermal efficiency remains constant for a number of cycles, only then the temperature profiles are obtained. This thermal efficiency work has been employed mainly in obtaining solid (chequerwork) temperature distributions and fluid exit temperature time histories for use in control of blast furnace stoves by Jeffreson [8]. We also note that Jeffreson [8] method has been developed to allow constant mass flow solutions to be transformed to variable mass flow i.e., constant mass flow solutions have been used to obtain the solutions for variable mass flow conditions.

b) More detailed review follows in next chapters, but here we note that the main object of the present work has been the investigation of and extensions to 'closed' method of Nahavandi and Weinstein [17] which requires simultaneous solution of (analytical) integral equations.

As noted above 'open' methods in general are very time consuming and may not even converge sometimes and although closed methods are 'difficult' analytically, once a solution is obtainable they may be considered to be more efficient, hence the interest in the closed method of NW (Nahavandi and Weinstein) [17].

The method developed here based upon NW method has following main features:

(i) It extends NW method to general unbalanced non-symmetric case, which is a more appropriate consideration since NW method is only applied to a 'Balanced Symmetric' regenerator (to be defined in next chapter). The need for this extension arises from the fact that most industrial regenerators are unbalanced and non-symmetric.

(ii) NW method relies upon computation of special functions like Bessel's function whereas this extended method avoids these calculations by employing Numerical Laplace inversion. We note, however that this method is also not totally 'trouble free'.

(iii) The proposed method extends NW method to permit the generation of exit fluid temperatures at cyclic equilibrium.

In connection with NW method some computational problems reported by Willmott [26] are also investigated.

c) Finally, we mention that an earlier attempt was made to obtain closed form solution using a finite stage model; this is outlined in appendix.

...

CHAPTER 2: PHYSICAL ASSUMPTIONS, MATHEMATICAL MODEL AND REVIEW OF PREVIOUS WORK

2.1. Physical assumptions and mathematical formulation of model:

2.1.1. Assumptions:

In the mathematical treatment of the regenerator problem, a number of simplifying assumptions have been made and used successfully e.g., Nusselt [18], Willmott and Thomas [26], Heggs and Carpenter [4] etc.

Before going on to mathematical treatment of the regenerator model we list the physical assumptions used:

1) Fluid thermal capacitance is zero at the end of each period.
2) The thermal conductivity of the solid is zero in the direction parallel to the fluid flow.
3) The thermal conductivity of the solid is infinitely large in the direction perpendicular to the fluid flow.
4) The thermal conductivity in the fluid is small in the direction of fluid flow.

The first assumption is quite justified because for most practical regenerator work the thermal capacitance of the solid is so large relative to that of the fluid (the ratio being about 10^4 usually) that the rate of accumulation of heat in the fluid may be ignored.

The last three assumptions are of great practical significance as the regenerators employed these days are such that there is no appreciable temperature difference in the cross section. So the thermal conductivity of the solid is infinitely large in direction perpendicular to the fluid which is our assumption 3). Jeffreson [32] points out that Hausen [1] and others report that internal packing resistance plays no part in the dynamics of the system since heat has no time to penetrate into the packing especially when period is small. This means that assumptions 2) and 4) are well justified.

2.1.2. Mathematical formulation:

We now derive the regenerator differential equations from first principles based on Jeffreson [14] using symbols as defined in Nomenclature. Additional symbols are defined as they occur.

For a thermal balance over fluid phase between lengths x and $x+\Delta x$ along the regenerator:

Rate of heat input at distance x and time y due to convection is $W.S.t_1(x,y)$ J/sec or Watts

where $t_1(x,y)$ is temperature of fluid at distance x obtained at time y.

If fluid thermal conductivity is defined as:

k_f J/(sec-m^2-^{o}C/m) or k_f W/(m-^{o}C),

the rate of heat transfer at distance x due to fluid conductivity will be:

$$A_c \; k_f \frac{\partial t_1}{\partial x} \quad \text{Watts}$$

where A_c is the area available for fluid flow normal to flow direction. [$A_c = m/(r_f.L)$, where r_f is the density of fluid in the regenerator at any time].

Supposing the temperature gradient $\partial t_1/\partial x$ is positive, then heat transfer due to conduction will be against the fluid flow i.e., Rate of heat in to volume of gas between x and $x + \Delta x$ due to conduction and convection will be:

$$WSt_1(x,y) - A_c k_f \frac{\partial t_1}{\partial x}(x,y) \quad \text{Watts} \qquad .. (2.1.2.(1))$$

Similarly rate of heat flow out at $x + \Delta x$ will be:

$$WSt_1(x + \Delta x,y) - A_c k_f \frac{\partial t_1}{\partial x}(x + \Delta x,y) \quad \text{Watts} \qquad .. (2.1.2.(2))$$

or

$$\text{if} \quad t_1(x + \Delta x,y) \simeq t_1(x,y) + \left.\frac{\partial t_1}{\partial x}(x,y)\right]_x \Delta x$$

$$\text{and} \quad \frac{\partial t_1}{\partial x}(x + \Delta x,y) \simeq \frac{\partial t_1}{\partial x}(x,y) + \left.\frac{\partial^2 t_1}{\partial x^2}(x,y)\right]_x . \Delta x$$

$$.. (2.1.2.(3))$$

the net rate of heat INPUT due to convection and conduction is then approximately:

$$WSt_1 - A_c k_f \frac{\partial t_1}{\partial x} - [WSt_1 + WS \frac{\partial t_1}{\partial x}\Delta x - A_c k_f \frac{\partial t_1}{\partial x} - A_c k_f \frac{\partial^2 t_1}{\partial x^2}\Delta x]$$

$$= -WS \frac{\partial t_1}{\partial x}\Delta x + A_c k_f \frac{\partial^2 t_1}{\partial x^2}\Delta x \qquad .. (2.1.2.(4))$$

where $\frac{\partial^2 t_1}{\partial x^2}$ and $\frac{\partial t_1}{\partial x}$ are evaluated at distance x.

Besides this, heat will be transferred from the solid to the fluid at the rate of:

$$h \frac{A}{L} \Delta x \,(T_1 - t_1) \quad \text{Watts} \qquad .. \; (2.1.2.(5))$$

The difference between the rate of heat input and the rate of heat output must be the rate of accumulation of heat in the fluid stored between x and x $+\Delta x$, a mass of $m\Delta x/L$ Kg i.e.,

$$-WS \frac{\partial t_1}{\partial x}\Delta x + A_c k_f \frac{\partial^2 t_1}{\partial x^2}\Delta x + h \frac{A}{L} \Delta x(T_1 - t_1)$$

$$= \frac{m}{L}\Delta x \; S \frac{\partial t_1}{\partial y}$$

multiplying this equation by $L/\Delta x$, we obtain:

$$A_c k_f L \frac{\partial^2 t_1}{\partial x^2} - WSL \frac{\partial t_1}{\partial x} + hA(T_1 - t_1) = mS \frac{\partial t_1}{\partial y} \qquad .. \; (2.1.2.(6))$$

For solid phase a similar treatment of heat balance over the solid between x and x $+\Delta x$ gives:

$$A_s k_s L \frac{\partial^2 T_1}{\partial x^2} - hA(t_1 - T_1) = MC \frac{\partial T_1}{\partial y} \qquad .. \; (2.1.2.(7))$$

as there is no convection term present.

2.2. Normalization of equations:

Define normalized distance $z = x/L$, $x = L/z$;

normalized time $\theta = y(W/m)$; $y = (m\theta/W)$;

denoting steady state temperatures with a - over them; using ' for hot blow and " for cold blow temperatures; and by defining normalized fluid temperature as:

$$t(z,\theta) = [t_1(x,y) - \bar{t}''_{1\ in}] / [\bar{t}'_{1\ in} - \bar{t}''_{1\ in}]$$

$$[\text{i.e., } 0 \leqslant t(z,\theta) \leqslant 1.0]$$

where $\bar{t}'_{1\ in}$ represents the steady state normal (or design) inlet temperature for hot blow, and $\bar{t}''_{1\ in}$ represents the steady state normal (or design) inlet temperature for cold blow; alongwith similar normalization for solid temperature as:

$$T[z,\theta) = [T_1(x,y) - \bar{t}''_{1\ in}] / [\bar{t}'_{1\ in} - \bar{t}''_{1\ in}]$$

and substituting into (2.1.2.(6)) we obtain:

$$\frac{A_c k_f}{L} \frac{\partial^2 t}{\partial z^2} - WS \frac{\partial t}{\partial z} + hA(T - t) = WS \frac{\partial t}{\partial \theta}$$

or

$$\frac{1}{P_f} \frac{\partial^2 t}{\partial z^2} - \frac{\partial t}{\partial z} + \lambda (T - t) = \frac{\partial t}{\partial \theta} \qquad .. (2.2.(1))$$

where

$$\frac{1}{P_f} = \frac{A_c k_f}{LWS}, \quad \lambda = \frac{hA}{WS} .$$

P_f is called the 'fluid Peclet number' and λ represents twice the 'number of transfer units' (or 'reduced length parameter'). Similarly substituting into (2.1.2.(7)) we obtain:

$$\frac{1}{P_s}\frac{\partial^2 T}{\partial z^2} + \lambda(t - T) = \alpha \frac{\partial T}{\partial \theta} \qquad \text{.. (2.2.(2))}$$

where P_s is the 'solid Peclet number' and $\alpha = MC/(mS)$. Note that $\lambda/\alpha = (hA/MC)(m/W)$ is independent of fluid specific heat. So although $S \longrightarrow 0$, $\alpha \longrightarrow \infty$, λ/α can be kept finite.

Now for zero fluid conduction $1/P_f = 0$, so (2.2.(1)) becomes:

$$\lambda(T - t) = \frac{\partial t}{\partial \theta} + \frac{\partial t}{\partial z} \qquad \text{.. (2.2.(3))}$$

Hence using normalization based upon Hausen's normalization as follows:

$$q = \lambda z, \quad r = \frac{\lambda}{\alpha}(\theta - z) \simeq \frac{\lambda}{\alpha}\theta$$

because for time values of interest $\theta \gg z$ and $\frac{\partial t}{\partial \theta}$ being very small is ignored.

Hence, we have for (2.2.(3)):

$$\lambda(T - t) = \frac{\partial t}{\partial z} \quad \text{i.e.,} \quad \lambda(T(q,r)-t(q,r)) = \frac{\partial t}{\partial z}(q,r)$$

as z and θ are arbitrary.

So,

$$\frac{1}{\lambda}\frac{\partial t}{\partial z}(q,r) = T(q,r) - t(q,r)$$

but,

$$\frac{1}{\lambda}\frac{\partial t}{\partial z}(q,r) = \frac{1}{\lambda}\frac{\partial t}{\partial q}(q,r).\frac{\partial q}{\partial z}$$

$$= \frac{1}{\lambda}\frac{\partial t}{\partial q}(q,r).\lambda = \frac{\partial t}{\partial q}(q,r).$$

Hence,

$$\frac{\partial t}{\partial q}(q,r) = T(q,r) - t(q,r) \qquad .. (2.2.(4))$$

Similarly for the solid: Zero conduction implies that

$\frac{1}{P_s} = 0$ and (2.2.(2)) becomes:

$$\lambda(t - T) = \alpha\frac{\partial T}{\partial \theta}$$

or

$$\frac{\alpha}{\lambda}\frac{\partial T}{\partial \theta} = t - T \qquad .. (2.2.(5))$$

i.e.,

$$\frac{\alpha}{\lambda}\frac{\partial T}{\partial \theta}(q,r) = t(q,r) - T(q,r)$$

since $q = \lambda z$, $r = (\lambda/\alpha)\theta$ and z,θ are arbitrary as before, but

$$\frac{\alpha}{\lambda}\frac{\partial T}{\partial \theta}(q,r) = \frac{\alpha}{\lambda}\frac{\partial T}{\partial r}(q,r)\frac{\partial r}{\partial \theta}$$

$$= \frac{\alpha}{\lambda}\frac{\partial T}{\partial r}(q,r)\frac{\lambda}{\alpha} = \frac{\partial T}{\partial r}(q,r)$$

Hence

$$\frac{\partial T}{\partial r}(q,r) = t(q,r) - T(q,r) \qquad .. (2.2.(6))$$

We notice that with normalized distance z, we have $0 \leq z \leq 1$ since $z = x/L$. Whereas Hausen's normalization uses $q = \lambda z$ for the distance parameter i.e., $0 \leq q \leq \lambda$. The advantage in using q rather than z variable is that one obtains rather simpler set of differential equations (2.2.(4)) and (2.2.(6)) to deal with as compared to the equations (2.2.(1)) and (2.2.(2)) where the second order terms in(2.2.(1)) and (2.2.(2)) are not involved because of conduction being zero. Similarly normalized time Θ has been utilised to obtain Hausen's r parameter in order to obtain simpler equations (2.2.(4)) and (2.2.(6)).

2.3. Thermal efficiency and thermal ratio:

Define efficiency by

$$E = \frac{\text{Heat out in cold blow above datum } (t''_{in})}{\text{Heat in during hot blow above same datum}} .$$

Since the exit fluid temperature during cold blow t''_{out} varies with time, the mean exit temperature:

$$t''_{out\ m} = \frac{1}{\pi''} \int_0^{\pi''} t''_{out}\, dr$$ must be used in

calculating E, i.e.,

$$E = \frac{W''S''(t''_{out\ m} - t''_{in})P''}{W'S'(t'_{in} - t''_{in})P'}$$

where P'' and P' are periods in seconds for cold and hot blow respectively. So, in terms of normalized parameters:

$$E = \frac{h'A}{W'S'} \frac{W''S''}{h''A} \frac{h''P''}{h'P'} \frac{t''_{out\ m} - t''_{in}}{t'_{in} - t''_{in}}$$

$$= \frac{\lambda'}{\lambda''} \frac{\pi''}{\pi'} \frac{t''_{out\ m} - t''_{in}}{t'_{in} - t''_{in}} \qquad \text{.. (2.3.(1))}$$

Willmott [25] defines a 'hot side thermal ratio' as:

$$R'_{REG} = \frac{t'_{in} - t'_{out\ m}}{t'_{in} - t''_{in}} \qquad \text{.. (2.3.(2))}$$

and a 'cold side thermal ratio' as:

$$R''_{REG} = \frac{t''_{out\ m} - t''_{in}}{t'_{in} - t''_{in}} \qquad \text{.. (2.3.(3))}$$

Hence for $\lambda' = \lambda''$, $\pi' = \pi''$ thermal efficiency can be compared with Willmott's cold side thermal ratio.

Efficiency in terms of solid temperature distributions:

We know that the heat removed from the packing during the hot blow is equal to the difference between the total heat stored at the beginning of the cold blow i.e.,

$$MC \int_0^1 T''(z,0)dz \quad \text{Joules}$$

and that left at the beginning of the hot blow. So,

$$E = \frac{MC[\int_0^1 T''(z,0)dz - \int_0^1 T'(z,0)dz]}{W'S'P'(t'_{in} - t''_{in})} \quad .. (2.3.(4))$$

2.4. Single blow solutions of Anzelius and Nusselt:

The stage of infancy of the theory of regenerators can be traced back to middle 1920's when consideration was initially devoted to the 'Single Blow' problem. Here the fluid is assumed to flow through a heat storing solid matrix in one direction, the problem being to determine how the solid and gas temperatures vary with distance down the bed for the duration of the "Blow". Anzelius [1A] in 1926 and then Nusselt [18] were the pioneers who tackled this problem.

Anzelius derived the 2-D equations governing the heat transfer in the following form:

$$\frac{\partial t}{\partial q} = T - t \quad .. (2.4.(1))$$

$$\frac{\partial T}{\partial r} = t - T \quad .. (2.4.(2))$$

To solve these, a transformation was proposed:

$$\Delta(q,r) = e^{q+r}(t - T)$$

and this resulted in the equation:

$$\frac{\partial^2 \Delta}{\partial q \partial r} = \Delta$$

The notation in use here is not that of Anzelius, but q and r are dimensionless parameters comparable with those described earlier.

Assuming an initial solid temperature distribution $T(q,0) = T_o$ and the entrance fluid temperature $t(0,r) = t_{in}$, the closed solution was derived as:

$$t(q,r) = t_{in} - (t_{in} - T_o)e^{-r} \int_0^q e^{-s} J_o(2i\sqrt{sr})ds \quad .. (2.4.(3))$$

and

$$T(q,r) = T_o - (t_{in} - T_o)e^{-q} \int_0^r e^{-s} J_o(2i\sqrt{sq})ds \quad .. (2.4.(4))$$

where $J_o(x)$ is the Bessel Function of order zero and is real for a pure imaginary argument.

Nusselt extended Anzelius' work by first establishing the simultaneous equations:

$$\frac{\partial T}{\partial r} = \frac{\bar{h}A}{MC}(t - T) \quad .. (2.4.(5))$$

$$\frac{\partial t}{\partial q} = \frac{\bar{h}A}{WSL}(T - t) \qquad .. (2.4.(6))$$

with the boundary conditions $t(0,r) = t_o$ and $T(q,0) = f(q)$.

In deriving above equations Nusselt did not take into account the heat content of the fluid resident in the regenerator packing i.e., solid. This is reasonable when the period of operation of the regenerator is measured in hours (i.e., large periods), however for shorter periods (measured in minutes) a term must be included in the equations to account for this resident heat. The term to be included is $mS/(WSL)(\partial t/\partial r)$ which becomes significant as P is such that m approaches WP.

However, in solving (2.4.(5)) and (2.4.(6)) Nusselt set $\bar{h}A/(MC) = p$, $\bar{h}A/(WSL) = n$ and eliminated T for the equations (2.4.(5)) and (2.4.(6)) yielding:

$$\frac{\partial^2 t}{\partial q \partial r} + p\frac{\partial t}{\partial q} + n\frac{\partial t}{\partial r} = 0 \qquad .. (2.4.(7))$$

Using Riemann's method (2.4.(7)) was integrated to obtain the solution:

$$t(q,r) = e^{-nq-pr}\,[t_o J_o(2i\sqrt{pqr}\,) + n\int_0^q e^{ns}f(s).$$

$$J_o(2i\sqrt{pnr(q-s)})ds + pt_o\int_0^r e^{ps}J_o(2i\sqrt{pnq(r-s})ds]$$

$$.. (2.4.(8))$$

Differentiation of (2.4.(8)) with respect to y and substitution into (2.4.(6)) yields the solid temperature profile as follows:

$$T(q,r) = f(q)e^{-pr} - e^{-nq-pr}[t_o\sqrt{pr/(nq)}\ iJ_1(2i\sqrt{pnqr}) + $$
$$+ \int_0^q e^{ns}f(s)\sqrt{pnr/(q-s)}\ iJ_1(2i\sqrt{pnr(q-s)})ds +$$
$$+ \int_0^r e^{ps}\ \sqrt{pn(q-s)}/q\ iJ_1(2i\sqrt{pnq(r-s)})ds] \quad .. (2.4.(9))$$

The solutions $t(q,r)$, $T(q,r)$ are heavily dependent upon evaluation of Bessel functions $J_o(x)$ and $J_1(x)$. They involve lengthy calculations which were not practicable in an era prior to that of the digital computer.

Once the digital computer was available, the speed and length of calculations were no longer deterrents for practical calculations. However, the methods of solution of the regenerator problem incorporating the 'reversal condition' were developed before the advent of digital computers.

2.5. Reversal conditions:

Basically the reversal condition is defined as 'the condition which specifies that the temperature distribution at the beginning of a period is identical to that at the conclusion of the previous period.' More explicitly they can be defined as follows:

For the 'hot' blow, the $\theta' = 0$ temperature distributions in fluid and solid are the same as the corresponding distributions at the end of the cold blow. Similarly, for the 'cold' blow, the $\theta'' = 0$ temperature distributions are the same as those at the end of the hot blow.

It is advantageous to reverse distance coordinates in describing hot and cold blows so that z' (or z'') increases in direction of the hot (or cold) blow. This preserves the form of equations (2.2.(1)) and (2.2.(2)).

Now if P' and P'' are assumed to be the period of hot and cold blows respectively then coordinate reversal implies that:

$$z'' = 1 - z' \qquad .. (2.5.(1))$$

and time starts again at the beginning of each blow so that:

$$\theta'' = \theta' - \frac{P'W'}{m'} \qquad .. (2.5.(2))$$

assuming θ' to be zero at the beginning of an arbitrary (repeating) hot blow.

Hence using Hausen's normalization (i.e., for dimensionless time parameter $r = \lambda\,\theta/\alpha$), we have π as the normalized duration of a blow which is related to P as follows:

$$\pi = \frac{\lambda}{\alpha}\left[\frac{PW}{m} - 1\right] \qquad .. (2.5.(3))$$

So for time scale θ of (2.2.(1)) and (2.2.(2)) we multiply π by α/λ and add 1.0 before substituting for the reversal conditions.

The reversal conditions then become:

For the Solid:

$$T'(z',\theta' = 0) = T''(1 - z'', \pi'' \alpha'' / \lambda'') \quad .. (2.5.(4))$$

and

$$T''(z'',\theta'' = 0) = T'(1 - z', \pi' \alpha'/ \lambda') \quad .. (2.5.(5))$$

and similarly for the Fluid:

$$t'(z',\theta' = 0) = t''(1 - z'', \pi'' \alpha''/ \lambda'') \quad .. (2.5.(6))$$

and

$$t''(z'',\theta'' = 0) = t'(1 - z', \pi' \alpha'/ \lambda') \quad .. (2.5.(7))$$

where T', T'' are solid temperatures for hot and cold blow respectively, and t', t'' are fluid temperatures for hot and cold blow respectively.

We note that the fluid temperatures at the end of the heating period do not have to equal those established at the beginning of the next because of assumption 1) in Sec. 2.1.1. Therefore future behaviour of the bed depends at any time only on the solid packing temperatures.

The methods of solution of the regenerator problem incorporating the reversal boudary conditions fall into two distinct classes:

a) The "open" method and b) the "closed" method.

We now review "open" methods.

2.6. Review of "Open" methods:

The "open" methods are those in which some arbitrary, but if possible well chosen distribution of temperature is imposed upon the solid matrix. Subsequently, the temperatures at the end of successive periods are evaluated repeatedly until the mathematical model achieves cyclic equilibrium.

The solving of equations (2.2.(4)) and (2.2.(6)) is regarded as an initial value problem and typical of these methods are those of Lambertson [16] and Willmott [23].

With the advent of digital computers, the calculations involved in continuous cycling of the model to equilibrium no longer presented major practical difficulties. It is interesting to note however, that immediately before digital computing machines became generally available, open methods of solution of the differential equations were attempted using Analogue computers. A review of the Analogue methods will not be attempted here but Heindlhofer and Larsen [5], Tipler [22] are cited as references.

2.6.1. Lambertson's method:

Just after digital computers became more readily available, Lambertson [16] described his method for representing the regenerator. Although he analysed the case for a 'Rotary' regenerator, the method of obtaining the temperature profiles is of substance here and this is why we review his method which is considered to be one of the earliest schemes involving finite differences and digital computing machines.

Lambertson did not solve the differential equations in an explicit manner, he considered the temperature behaviour of successive equally spaced sections of regenerator packing. A step-by-step procedure suitable for digital calculations was adopted instead of obtaining the temperature variations in a continuous manner.

Lambertson [16] obtained a finite difference scheme for the temperature distribution in a rotary regenerator by expressing the heat balance equation directly in differential form. This had the advantage of avoiding the derivation of the controlling differential equations and then subsequently discretising these to obtain a numerical solution. He obtained the outlet temperatures in terms of the inlet temperatures in the following form:

$$t_{m\ out} = t_{m\ in} - K_1(t_{m\ in} - T_{in}) \qquad .. (2.6.1(1))$$

$$T_{out} = T_{in} + K_2(t_{m\ in} - T_{in}) \qquad .. (2.6.1(2))$$

where m refers to the maximum heating fluid flow and on the cold side we get:

$$t_{n\ out} = t_{n\ in} + K_3(T_{in} - t_{n\ in}) \qquad .. (2.6.1(3))$$

$$T_{out} = T_{in} - K_4(T_{in} - t_{n\ in}) \qquad .. (2.6.1(4))$$

where the subscript n refers to the cold side of the regenerator. The constants K_i, $i = 1,\ldots,4$ are functions of maximum and minimum fluid capacity rates, solid capacity rate, the number of subdivisions of these capacity rates, regenerator heating surface area and bulk heat transfer coefficient.

The fluid and solid temperatures can be obtained in the order indicated by repetitive use of (2.6.1(1)) to (2.6.1.(4)), given the starting values T_{in}, $t_{m\ in}$, $t_{n\ in}$.

In Lambertson's scheme the regenerator is represented by a rectangle which is obtained if the 'rotary' regenerator cylinder is opened out. The fluid and matrix streams have been divided into three equal substreams to form differential elements. The left edge is then physically the same as the right edge. So the matrix inlet temperature for a substream on the left must be identical to the outlet temperature of that substream on the right. This of course is the reversing condition.

If the temperature distribution obtained on the right is the same as at the left, then equilibrium has been reached. If this is not the case then the resulting temperature distribution is now used on the left and the process repeated until the reversal condition is met.

Lambertson [16] also calculated the efficiency of the regenerator and tabulated his results for a variety of ranges of parameters. These calculations were used to determine the convergence of the iterative scheme. He pointed out that the convergence depended on both the physical conditions of the problem and the second law of thermodynamics. It was observed by him that a greater number of subdivisions will improve the accuracy and enhance the convergence at the same time. But an increase in number of subdivisions causes an increase in computing time so some compromise must be reached. Lambertson computed the efficiencies of several values of the subdivision and then extrapolated the results to zero element area.

Whereas Lambertson considered the 2-D model of a rotary regenerator, Willmott [23] used a similar technique in modelling regenerators in 'conventional' counterflow operation.

2.6.2. Willmott's [23] trapezoidal method:

In 1964, Willmott implemented a finite difference method for Ferranti Pegasus digital computer. In his method equations (2.2.(4)) and (2.2.(6)) are represented in a difference form using the trapezoidal rule.

Equation (2.2.(4)) is integrated using:

$$t_{j+1,k} = t_{j,k} + \frac{\Delta q}{2}\left[\left(\frac{\partial t}{\partial q}\right)_{j+1,k} + \left(\frac{\partial t}{\partial q}\right)_{j,k}\right] \quad .. (2.6.2(1))$$

and equation (2.2.(6)) is integrated as:

$$T_{j,k+1} = T_{j,k} + \frac{\Delta r}{2} [(\frac{\partial T}{\partial r})_{j,k+1} + (\frac{\partial T}{\partial r})_{j,k}] \quad .. (2.6.2.(2))$$

where the subscripts j and k refer to distance and time positions on a finite difference grid, Δq is the distance step length and Δr the time step length. But from (2.2.(4)) and (2.2.(6)) we have:

$$(\frac{\partial t}{\partial q})_{j,k} = (T_{j,k} - t_{j,k}) = (T - t)_{j,k} \quad .. (2.6.2.(3))$$

and

$$(\frac{\partial T}{\partial r})_{j,k} = (t_{j,k} - T_{j,k}) = (t - T)_{j,k} \quad .. (2.6.2.(4))$$

So (2.6.2.(1)) and (2.6.2.(2)) become:

$$t_{j+1,k} = t_{j,k} + \frac{\Delta q}{2} [(T-t)_{j+1,k} + (T-t)_{j,k}] \quad .. (2.6.2.(5))$$

$$T_{j,k+1} = T_{j,k} + \frac{\Delta r}{2} [(t-T)_{j,k+1} + (t-T)_{j,k}] \quad .. (2.6.2.(6))$$

Now setting $a = \Delta q/2$, $b = \Delta r/2$, Willmott arrived at the two equations:

$$t_{j+1,k} = (\frac{1-a}{1+a})t_{j,k} + (T_{j+1,k} + T_{j,k}) \quad .. (2.6.2.(7))$$

$$T_{j,k+1} = (\frac{1-b}{1+b})T_{j,k} + \frac{b}{1+b} (t_{j,k+1} + t_{j,k}) \quad .. (2.6.2.(8))$$

So the fluid and solid temperature distributions throughout the regenerator can be obtained via repetitive use of (2.6.2.(7)) and (2.6.2.(8)).

As in the Lambertson's method the whole process is repeated until the state of equilibrium has been reached. Willmott found the results of his method to be in excellent agreement with those obtained by the 'closed' method of Illiffe [6]. He gave the truncation errors associated with the difference equations (2.6.2.(5)) and (2.6.2.(6)) as:

$$e_q = -\frac{(\Delta q)^3}{12}\left(\frac{\partial^3 t}{\partial q^3}\right)_{j,k} + \ldots \qquad \ldots (2.6.2.(9))$$

$$e_r = -\frac{(\Delta r)^3}{12}\left(\frac{\partial^3 T}{\partial r^3}\right)_{j,k} + \ldots \qquad \ldots (2.6.2.(10))$$

These errors decrease for decreasing values of Δq and Δr so the errors depend upon the size of distance and time steps. On the other hand the size of the derivatives $\partial^3 t/(\partial q^3)$ and $\partial^3 T/(\partial r^3)$ also matters. If the variations of temperature with respect to q and r are approximately linear then resulting truncation errors will be low. This suggests that for these cases larger time and distance steps can be taken.

Willmott's method forms the basis of a more recently developed open method by Jeffreson [8]. This is reviewed next.

2.6.3. Jeffreson's [8] method:

This method simulates the dynamic behaviour of any number of blast furnace stoves simultaneously. It involves trapezoidal integration in the distance coordinate and subsequent integration of the resultant lumped system forward in time. For integration a fourth order Runge-Kutta time step is used. The method has been used in developing a simulation package which has a teletype plot facility and other features like forcing of "hold"* states from the simulation subroutine, ability to reset "integrators" during execution, provision for changing of parameters while the system is in a hold state, etc. Based on equation (2.6.2.(7)) obtained on application of trapezoidal integration, the 'hot' blow equation for fluid temperature at each time node was rewritten as:

$$t_{j+1} = Bt_j + C(T_{j+1} + T_j) \qquad \text{.. (2.6.3.(1))}$$

where $B = (1-a)/(1+a)$, $C = a/(1+a)$ and $t_o = t'_{in}$.

Here for a bed of unit normalized length divided into M intervals, $j = 0,1,\ldots,M-1$; $a = \bar{\lambda} H/(2M\bar{W})$, $H = h/\bar{h}$, $\bar{W}$ is the reference fluid flowrate through regenerator (Kg/sec) and $\bar{\lambda} = \bar{h}A/(\bar{W}S)$.

(2.6.3.(1)) has a truncation error of order $(1/M)^3$ as defined in Section 2.6.2.

* "hold" state refers to the state during simulation on a computer in which software itself asks for input of any new (changed) parameters before executing further.

For a 'cold' blow, distance integration was performed in the reverse direction yielding:

$$t_{j-1} = Bt_j + C(T_{j-1} + T_j) \qquad \text{.. (2.6.3.(2))}$$

where B and C are as defined earlier (but for cold blow) and $t_M = 0$; $j = M+1, M, \ldots, 1$.

Time derivatives of solid temperature for each blow are also evaluated at every nodal point in the same order as fluid temperatures in (2.6.3.(1)) and (2.6.3.(2)) from:

$$\frac{\partial T_k}{\partial r} = H.H^*(t_k - T_k) \qquad \text{.. (2.6.3.(3))}$$

where H is as defined above; H* is defined as the ratio $\bar{h}/\bar{h}''$ where of course H* = 1 for a cold blow.

These time derivatives are then used by a Runge-Kutta integration subroutine in forward integration to yield the next array of solid temperatures.

For the cold blow integration, cold blow flowrate is adjusted depending upon the blow exit temperature, appropriate heat transfer coefficients are applied in order to obtain a, B and C as a function of the final exit temperature t_{M+1}, which is obtained only when integration is completed. So, each distance integration should be an iterative process.

It was considered by Jeffreson that since fourth order R-K integration in effect involves successive estimations of the derivatives and hence of t_{M+1}, the need for iteration would be eliminated by reducing the mesh spacing Δr. In general, a maximum of thirty distance steps (M = 29), with relatively low values of λ', λ'' were used. The time step increment Δr was selected to be no more than smallest value of 1/(10.H.H*) where 1/(H.H*) is the "time constant" of equation (2.6.3(3)). He found that the period converged towards zero as cold blow fluid flowrate W" approached its analytical limit. The question of computational efficiency of iteration was not investigated by Jeffreson in his paper [8].

2.7. Analytical solutions U and V:

Starting with:

$$\frac{\partial t}{\partial q} = T - t \qquad \text{.. (2.7.(1))}$$

$$\frac{\partial T}{\partial r} = t - T \qquad \text{.. (2.7.(2))}$$

we first Laplace transform these equations with respect to the distance parameter to get:

$$p\hat{t} - t_{in}(r) = \hat{T} - \hat{t}$$

or

$$(p+1)\hat{t} = t_{in}(r) + \hat{T}$$

or

$$\hat{t} = \frac{1}{p+1}[t_{in}(r) + \hat{T}] \qquad \text{.. (2.7.(3))}$$

where q is the Laplace transformed variable resulting in p. The Laplaced functions are denoted by ^ ; $t_{in}(r) = t(q = 0,r)$; and for (2.7.(2)) we have:

$$\frac{\partial \hat{T}}{\partial r} = \hat{t} - \hat{T} \qquad .. (2.7.(4))$$

Using (2.7.(3)) in (2.7.(4)) we obtain:

$$\frac{\partial \hat{T}}{\partial r} = [\frac{1}{p+1} - 1]\hat{T} + \frac{1}{p+1} t_{in}(r)$$

or

$$\frac{\partial \hat{T}}{\partial r} = \frac{1}{p+1} [t_{in}(r) - p\hat{T}]$$

or

$$\frac{d\hat{T}}{dr} = b\ t_{in}(r) + a\ \hat{T} \qquad .. (2.7.(5))$$

as only the variable r is involved in differentiation, so can use full derivative instead of partials; $b = 1/(p+1)$ and $a = -pb$.

(2.7.(5)) has a complete solution in the form of a homogeneous equation (depending only on the initial solid temperature distribution) and a particular integral or forced solution depending on the inlet temperature function, $t_{in}(r)$:

$$\hat{T}(p,r) = e^{ar}.\hat{T}(p,0) + \int_0^r e^{(r-w)}\ b\ t_{in}(w)dw \qquad .. (2.7.(6))$$

Now defining $\hat{V}(p,r) \;=\; e^{ar}$ .. (2.7.(7))

where $a \;=\; -p/(p+1)$ and $L[V(q,r)] = \hat{V}(p,r)$, we have:

$$\hat{V}(p,r-w) \;=\; e^{a(r-w)} \;=\; e^{ar}.e^{-aw}$$

$$= \hat{V}(p,r).\hat{V}(p,-w) \qquad .. \; (2.7.(8))$$

Thus transforming (2.7.(5)) again with respect to normalized time variable r (resulting Laplace transform variable s) and writing $L[\hat{T}(p,r)] = \hat{T}(p,s) = \hat{\underline{T}}$, we have:

$$s\hat{\underline{T}} - \hat{T}(p,0) = b\underline{t}_{in}(s) + a\hat{\underline{T}}$$

i.e., $\hat{\underline{T}}(s-a) = b\underline{t}_{in}(s) + \hat{T}(p,0)$

So, $\hat{\underline{T}} \;=\; \dfrac{1}{s-a}\,(\hat{T}(p,0) + b\underline{t}_{in}(s))$

Therefore,

$$\hat{V}(p,r) \;=\; L^{-1}(\frac{1}{s-a}) \;=\; e^{ar} \;=\; e^{-pr/(p+1)}.$$

Hence (2.7.(6)) implies:

$$\hat{T}(p,r) = \hat{V}(p,r).\hat{T}(p,0) + \int_0^r \hat{V}(p,r-w)\,\frac{1}{p+1}\,t_{in}(w)dw \qquad .. \; (2.7.(9))$$

For zero initial conditions throughout the bed and for $t_{in}(w)$ a unit step function the 'step response' or 'breakthrough' or 'single blow response' results:

$$\hat{U}(p,r) \;=\; \int_0^r \hat{V}(p,r-w)\,\frac{1}{p+1}\,dw \qquad .. \; (2.7.(10))$$

$$= \int_0^r \frac{1}{p+1}\; e^{-p/(p+1).(r-w)}\; dw$$

where $L[U(q,r)] = \hat{U}(p,r)$; which may be inverted to yield the distribution of temperature along the regenerator as a function of time. For $t_{in}(r)$ a Dirac or impulse function $\delta(r)$, the 'impulse response function' (or 'Heat Pole' function of Hausen) results:

$$g(p,r) = \hat{V}(p,r) \frac{1}{p+1} = (e^{-pr/(p+1)})/(p+1) \qquad .. \text{ (2.7.(11))}$$

Substituting (2.7.(10)) into (2.7.(9)) and inverting (2.7.(9)) one can obtain solid temperature distributions as:

$$T(q,r) = U(q,r) + \int_0^q V(v,r).T(q-v,0)dv \qquad .. \text{ (2.7.(12))}$$

by making use of Convolution theorem.

From (2.7.(3)) we obtain the fluid temperatures as:

$$t(q,r) = L^{-1}\left[\frac{1}{p+1}[\hat{T} + t_{in}(r)]\right]$$

$$= \int_0^q e^{-(q-v)}.T(v,r)dv + \bar{t}_{in}\int_0^q e^{-v}dv \qquad .. \text{ (2.7.(13))}$$

where $\bar{t}_{in}$ is either constant (step input) or zero during hot or cold blows.

So, for hot blow (2.7.(12)) becomes:

$$T'(q',r') = U'(q',r') + \int_0^{q'} V'(v,r').T'(q'-v,0)dv \qquad .. \text{ (2.7.(14))}$$

and for cold blow since $t''_{in}(r) = 0$, (2.7.(12)) yields:

$$T''(q'',r'') = \int_0^{q''} V''(v,r'').T''(q''-v,0)dv \quad .. (2.7.(15))$$

using Convolution theorem and noting the assumption that $V(q,r) = 0$ for $q < 0$, (2.7.(14)) and (2.7.(15)) can be written as:

$$T'(q',r') = U'(q',r') + \int_0^{q'} V'(q'-v,r').T'(v,0)dv \quad .. (2.7.(16))$$

and

$$T''(q'',r'') = \int_0^{q''} V''(q''-v,r'').T''(v,0)dv \quad .. (2.7.(17))$$

Similarly for hot blow (2.7.(13)) becomes:

$$t'(q',r') = \int_0^{q'} e^{-(q'-v)}.T'(v,r')dv + \bar{t}'_{in} \int_0^{q'} e^{-v}dv \quad .. (2.7.(18))$$

and for cold blow assuming $\bar{t}''_{in} = 0$, we have:

$$t''(q'',r'') = \int_0^{q''} e^{-(q'-v)}.T''(v,r'')dv \quad .. (2.7.(19))$$

This analytical solution subject to reversal conditions could be used to generate open solution by repeated cycling, but it is envisaged that the closed form solution will be potentially more efficient.

2.8. Review of "closed" methods:

In the "closed" methods, the reversal condition, which specifies that the solid temperature distribution at the beginning of a period is identical to that at the conclusion of the previous period, is incorporated directly. By embodying this condition for both hot and cold periods simultaneously, within the mathematical method for the solving of the differential equations, one specifies implicitly the cyclic equilibrium condition that the solid temperature distribution at the beginning of a complete cycle of operation, a cycle consisting of a hot/cold period followed by a cold/hot period, is identical to that at the beginning of the previous cycle.

In chronological order the methods of Hausen [1], Illiffe [6] and Nahavandi and Weinstein [17] stand out as milestones on the path taken to obtain closed form solutions. We review these now.

2.8.1. Hausen's [1] method:

Hausen proposed his "Heat Pole" method in 1931 to determine the temperature distribution within the regenerator. This method, although not the most efficient of its type, still deserves emphasis as it was the first method of its type to be used on this problem and it should be rated as a 'significant' contribution.

Hausen's method was based on the linearity of the differential equations enabling him to add particular solutions of the equations together. At first the initial temperature distribution of the solid matrix is decomposed into several parts, then the final temperature for each of the parts is determined and finally the total temperature distribution is obtained as the sum of constituent parts.

The regenerator length is divided up into N equal strips each of width Δn, the mean values of the initial excess temperatures of each strip above the constant inlet temperature t_{in} are denoted by $f_1, f_2, \ldots, f_N$.

A "Heat Pole" is defined to be a single strip of height 1 and of width Δn between the points $q = n$ and $q = n + \Delta n$. The "Heat Pole function" denoted by V is the temperature distribution established in the chequerwork after a time r of passing a fluid which enters at a temperature t_{in} flowing through the regenerator with a particular temperature distribution. As the fluid passes from left to right the point with the heat pole is gradually cooled while elements to the right are warmed up by the passing fluid.

V depends on the width of the pole Δn, the position in the regenerator q and on the time r. $V_1, V_2, \ldots, V_N$ are the mean values of this function in the strips 1 to N.

The influence of each of the strips on the final temperature distribution is obtained as follows:

A heat pole of height f_1 placed at strip 1 would produce the strip temperatures: $f_1V_1, f_1V_2, \ldots, f_1V_N$ because the mean initial temperature excess in strip 1 is f_1.

Similarly the effect of the second strip on the final temperature in the N-strips is: $0, f_2V_1, f_2V_2, \ldots, f_2V_{N-1}$. The total final distribution is the sum of individual contributions:

$$\begin{aligned}
&\text{Strip 1:} \quad T_1 - t_{in} = f_1V_1 \\
&\text{Strip 2:} \quad T_2 - t_{in} = f_1V_2 + f_2V_1 \\
&\text{Strip 3:} \quad T_3 - t_{in} = f_1V_3 + f_2V_2 + f_3V_1 \\
&\quad\vdots \\
&\text{Strip N:} \quad T_N - t_{in} = f_1V_N + f_2V_{N-1} + \ldots + f_NV_1.
\end{aligned} \qquad \ldots (2.8.1(1))$$

So, once the heat pole function V is known it is straightforward to obtain solid temperature distribution T(q,r) where q is normalized distance (position 1 to N here for N strips) at any given time r for any arbitrary initial temperature distribution. The fluid temperatures can also be obtained using a similar approach.

So the problem was reduced to determining V. Hausen applied two approaches. The first one was a graphical method which used the curves obtained as a solution to the single blow problem.

The second approach which is much more interesting was an analytical one in which the solution of Anzelius was adopted. For calculation of V, evaluation of the integral:

$$\int_0^{\bar{\pi}} e^{-s} J_0(2i\sqrt{qs})\,ds \qquad \text{.. (2.8.1(2))}$$

is necessary, where $\bar{\pi}$ is a normalized period associated with bulk heat and since Hausen proposed numerical methods to evaluate this integral it is noticeable that a direct application of numerical quadrature to the governing integral equations will be more suitable thus undermining the computational value of the Heat Pole method.

The integral equation:

$$f(r) + f(\lambda - q) + \int_0^q f(w)K(q-w)dw = t'_{in} - t''_{in} \qquad \text{.. (2.8.1(3))}$$

where $K(q-w) = e^{-(q-w+r)}\sqrt{r/(q-w)}\ .\ iJ_1(2i\sqrt{(q-w)r})$ was obtained by Hausen as the governing equation for the case of <u>infinitely narrow heat poles</u>. Hence the Heat Pole method can be considered to be an approximation method for solving the governing integral equation (2.8.1.(3)).

Hausen refined his Heat Pole method in 1950. The technique used was almost identical to that of Illiffe [6].

Hausen acknowledged Illiffe's contribution which we review next.

2.8.2. Illiffe's [6] method:

In 1948 Illiffe [6] published his method for solving the differential equations (2.7.(1)) and (2.7.(2)). This method of steady-state calculation was based on the solution of these equations by Nusselt [18] (as described earlier for a single blow problem).

The simplifying assumption used was that the cold fluid entry temperature should be zero and the hot fluid entry temperature unity. By introducing the concept of reduced temperature he simplified Nusselt's solution from an equation containing four terms, two of which contained indefinite integrals, to two terms of which just one contained an indefinite integral.

On the reduced temperature scale with f(q) in Nusselt's method becoming F(q), the solution is given as:

For the cold blow:

$$T''(q'',r'') = e^{-r''}F''(q'') - \int_0^{q''} i \, \frac{J_1(2i\sqrt{(q''-s)r''})}{\sqrt{(q''-s)r''}} \, . \, (r''e^{-[(q''-s)+r'']})F''(s)ds \quad .. \; (2.8.2(1))$$

and for the hot blow:

$$T'(q',r') = 1 - e^{-r'}[1-F'(q')] + $$

$$+ \int_0^{q'} i \frac{J_1[2i\sqrt{(q'-s)r'}]}{\sqrt{(q'-s)r'}} r' e^{-[(q'-s)+r']} .$$

$$. [1-F'(s)]ds \qquad .. (2.8.2.(2))$$

When the reversal conditions were applied, following simultaneous equations were obtained for F' and F'':

$$F'(\lambda'(1 - \frac{q''}{\lambda''})) = e^{-\pi''}F''(q'') + \int_0^{q''} K''(q''-s)F''(s)ds \qquad .. (2.8.2.(3))$$

and

$$1-F''(\lambda''(1 - \frac{q'}{\lambda'})) = e^{-\pi'}[1-F'(q')] + \int_0^{q'} K'(q'-s)[1-F'(s)]ds \qquad .. (2.8.2.(4))$$

where

$$K(q-s) = - i \frac{J_1[2i\sqrt{(q-s)\pi}]}{\sqrt{(q-s)\pi}} \pi e^{-[(q-s)+\pi]}$$

This pair of equations was solved by making use of Simpson's rule to approximate the value of each of the integrals. This produced a set of 2(n+1) simultaneous equations of the form:

$$\underline{A}\underline{x} = \underline{b} \qquad .. (2.8.2.(5))$$

where

$$\underline{x}^T = (F'_o, F'_1, \ldots, F'_n, F''_o, F''_1, \ldots, F''_n).$$

The solution of (2.8.2.(5)) is:

$$\underline{x} = A^{-1}\,\underline{b} \qquad .. (2.8.2.(6))$$

which exists if the matrix A is non-singular. The method introduces significant errors in the components of $\underline{x}$ if the determinant $|A|$ of A becomes very small i.e., approaches zero. The set of simultaneous linear equations (obtained as a result of discretisation of the integrals) becomes increasingly ill-conditioned the larger the ratio of the reduced length λ to the reduced period π. In such cases it is advisable to adopt an alternative approach, perhaps the method of Nahavandi and Weinstein [17].

2.8.3. Nahavandi and Weinstein's [17] method:

A detailed review follows in the next section; here, we review this method very briefly by virtue of its being a close method.

Nahavandi and Weinstein (in short NW) in 1961 solved equations (2.7.(1)) and (2.7.(2)) by a Laplace transform technique. The equations thus evolved were similar to those used by Illiffe, but NW approached the solution to these latter equations in a different manner, by introducing infinite series representations for the initial matrix temperature distribution as follows:

For hot blow:

$$f'(q') = \sum_{n=0}^{\infty} a'_n \, q'^n \qquad .. \ (2.8.3.(1))$$

and for cold blow:

$$f''(q'') = \sum_{n=0}^{\infty} a''_n \, q''^n \qquad .. \ (2.8.3.(2))$$

If an n-power series is assumed the 2(n+1) coefficients of q'^n and q''^n are found by solving the set of simultaneous integral equations arising from substitution of (2.8.3.(1)) and (2.8.3.(2)). Then in a similar manner to that of Illiffe [6], Newton-Coates formulae are used to find approximate values of integrals involved such as in (2.8.2.(1)) and (2.8.2.(2)).

2.9. Detailed review of Nahavandi and Weinstein's method:

Nahavandi and Weinstein (NW) [17] followed Anzelius and Nusselt (see section 2.4) and derived the equation:

$$\frac{\partial^2 \Delta}{\partial q \partial r} = \Delta \qquad .. \ (2.9.(1))$$

using the customary transformation:

$$\Delta = (e^{q+r}) t_{in}$$

Equation (2.9.(1)) was then solved using Laplace transforms to obtain the following equations:

For hot blow:

$$T'(q',r') = 1-e^{-r'}[1-f'(q')] + e^{-q'-r'}\int_0^{q'} [1-f'(s)]e^s i\sqrt{\frac{r'}{q'-s}}\, J_1(2i\sqrt{r'(q'-s)})ds \qquad .. (2.9.(2))$$

$$t'(q',r') = 1-e^{-q'-r'}\int_0^{q'}[1-f'(s)]e^s J_o(2i\sqrt{r'(q'-s)})ds \qquad .. (2.9.(3))$$

and for cold blow:

$$T''(q'',r'') = e^{-r''}f''(q'')-e^{-q''-r''}\int_0^{q''} f''(s)e^s\, i\sqrt{\frac{r''}{q''-s}}\,\cdot$$

$$\cdot\, J_1(2i\sqrt{r''(q''-s)})ds \qquad .. (2.9.(4))$$

$$t''(q'',r'') = e^{-q''-r''}\int_0^{q''} f''(s)e^s J_o(2i\sqrt{r''(q''-s)})ds \qquad .. (2.9.(5))$$

where as usual J_o and J_1 are Bessel functions, $f(q)$ represents the initial matrix temperature distribution and the temperature scale is chosen such that $t'_{in} = 1$, $t''_{in} = 0$ (which are identical conditions to those of Illiffe).

The unknown functions $f'(q')$ and $f''(q'')$ were determined by applying the reversal boundary conditions from:

$$f'(q') = e^{-r''_o} f''(q'') + \int_0^{q''} f''(s)K''(q''-s)ds \qquad .. (2.9.(6))$$

$$f''(q'') =1-e^{-r'_o}[1-f'(q')] + \int_0^{q'}[1-f'(s)]K'(q'-s)ds \qquad .. (2.9.(7))$$

where r'_o and r''_o are the known values of normalized time variables r' and r'' when the matrix leaves the hot or cold medium respectively,

$$K'(q'-s) \equiv -e^{-r'_o-(q'-s)} . i\sqrt{\frac{r'_o}{q' - s}}\, J_1(2i\sqrt{r'_o(q'-s)}) \quad .. (2.9.(8))$$

and

$$K''(q''-s) \equiv -e^{-r''_o-(q''-s)}\; i\sqrt{\frac{r''_o}{q''-s}}\, J_1(2i\sqrt{r''_o(q''-s)}) \quad .. (2.9.(9))$$

It was assumed that $f'(q')$ and $f''(q'')$ are represented by their power series expansions:

$$f'(q') = \sum_{n=0}^{\infty} a'_n q'^n , \quad f''(q'') = \sum_{n=0}^{\infty} a''_n\, q''^n \quad .. (2.9.(10))$$

where a'_n and a''_n are coefficients of the power series. These coefficients can be evaluated by writing each of the equations (2.9.(6)) and (2.9.(7)) for a number of n points chosen arbitrarily along the matrix. The result will be 2n equations in 2n unknowns from which the coefficients a'_n and a''_n and consequently $f'(q')$ and $f''(q'')$ could be computed. This then allowed the evaluation of hot and cold period temperatures for solid and fluid through equations (2.9.(2)) to (2.9.(5)).

The procedure adopted by NW was to initially fit a quadratic

(N = 2) for f' and f" and to increase the order of the polynomial till convergence occurred i.e., till the temperatures calculated with a polynomial of degree N were sufficiently close to those with the polynomial of degree N+1. For a range of parameters comparable with those used by Lambertson the results obtained by NW were in good agreement with the ones reported by Lambertson.

It is noted that NW considered the rather simplified case of a balanced, symmetric regenerator and their method used evaluation of integrals through quadrature which can be classed as a problem within itself.

Willmott [26] defines a 'balanced' regenerator as the one where $\lambda'/\lambda'' = \pi'/\pi'' = k$; when $k = 1$ it is said to be 'symmetric' and $\lambda = \lambda' = \lambda''$, $\pi = \pi' = \pi''$ (where $0 \leqslant q \leqslant \lambda$, $0 \leqslant r \leqslant \pi$); also the temperature behaviour of the solid (chequerwork) in the hot period is exactly symmetrical with that in the cold period at cyclic equilibrium. In these particular circumstances, the reversal condition can be rewritten as: $f'(q) = 1-f''(q)$, thus reducing the problem to a 'single period' boundary value problem.

2.10. An overview of closed methods using consistent notations:

So far in the review of closed methods in order to explain the methods more completely, the notations used by the authors were 'tailored' and equations presented in forms similar to the ones given by the authors. However, a clearer picture of the closed methods and their inter-relations can only be obtained if consistent notations are used to explain the various features. This is now done using the notations of section 2.7.

Taking the hot blow conditions and considering the solid temperatures (dropping the superscripts ') we obtain for (2.7.(16)):

$$T(q,r) = U(q,r) + \int_0^q V(q-v,r).T(v,0)dv \quad .. (2.10.(1))$$

and for cold blow the solid temperatures (again dropping the superscripts ") (2.7.(17)) becomes:

$$T(q,r) = \int_0^q V(q-v,r).T(v,0)dv \quad .. (2.10.(2))$$

The closed methods basically involve the evaluation of a function $T(q,0)$ which represents the solid temperature distribution at the beginning of a period and q is reduced distance down the regenerator.

Hausen and Illiffe replace the integrals in (2.10.(1)) and (2.10.(2)) by quadrature formulae so that the integral equations are reduced to a set of simultaneous linear equations which are solved for the values of $T_0, T_1, T_2, ...$

where, $T_j = T(j\Delta q,0) \quad ; \quad j = 0,1,2,...$ .. (2.10.(3))

and Δq is the distance between the equally spaced positions where $T(q,0)$ is evaluated.

The approach of Nahavandi and Weinstein (NW) has been to represent the initial temperature distribution $T(q,0)$ by a power series:

$$T(q,0) = a_o + a_1q + \ldots + a_mq^m \qquad .. \ (2.10.(4))$$

Here also, the integrals are replaced by numerical quadrature formulae. A set of simultaneous linear equations is generated which is solved for the coefficients $a_o, a_1, \ldots$

In both approaches, it is required to solve a set of linear algebraic equations of the general form:

$$A\underline{x} = \underline{b} \qquad .. \ (2.10.(5))$$

This method breaks down if the determinant $|A|$ of A becomes very small, i.e., if the matrix becomes almost or exactly singular.

In fact for the Illiffe's method the set of simultaneous equations represented by (2.10.(5)) become ill-conditioned if the determinant $|A|$ becomes very small. In such circumstances, small perturbations in the elements of A cause large perturbations in the solution $\underline{x}$. In the Illiffe's method, as the ratio of the reduced length to the reduced period, λ/π increases, Willmott and Thomas [26] report that the determinant $|A|$ decreases in size.

The matrix A is perturbed by the truncation errors associated with the quadrature representation of the integrals which give rise to errors in the computed values of the elements of $\underline{x}$. This ill-conditioning is relieved by increasing the number of levels in the regenerator at which the temperature T(q,0) is computed. The larger the number of levels, shorter the step length between levels resulting in smaller truncation errors. But with an increase in number of levels, the number of simultaneous linear equations also increases thus creating problems of a different nature which however can be controlled on a digital computer.

The NW method does not suffer from this ill-conditioning, in fact the reverse effect occurs. As λ/π increases in value, the determinant $|A|$ becomes large and if stretched to the limits may cause computational problems.

Summarising the three methods we note that Hausen in his heat pole method adopted a form of finite difference representation of the differential equations, while Illiffe and Nahavandi and Weinstein considered the problem in its integral equation form. All three methods divided the length of the regenerator into strips but where Hausen used the mean temperature of the strips in his calculations, Illiffe and NW used the temperatures at the end points of the strips. NW method which does not seem to suffer from ill-conditioning can be classed as the most reliable of the three closed methods, however, in their paper NW cover a symmetric balanced regenerator and do not give explicit forms for general non-symmetric, unbalanced . case.

2.11. Outline of a 'Finite Stage' method:

A 'finite stage' model is an approximation to the continuous model obtained by dividing a parameter range into say N stages of length 1/N, where the distributions are treated as uniform throughout each stage by the use of finite difference approximations to the derivatives with respect to the parameter being divided into 'stages'.

Razelos and Paschkis (RP) [19] utilised a finite stage approach to solve for dimensionless temperatures for each blow for solid as well as fluid. As they were using a Blast Furnace Stove Regenerator, terms such as 'gas' for hot fluid and 'air' for cold fluid, along with 'wall' for solid 'brick' matrix were used. The method itself is well documented in terms of 14 steps which can be attempted manually with the help of plots and interpolations. However, once the values go outside the ranges of plots or otherwise the number of calculations increases, the need for a digital computer becomes absolutely essential.

RP attempted to solve equations similar to (2.7.(1)) and (2.7.(2)) on the heat and mass flow analyzer but found that the results were unsatisfactory due to equipment limitations. So the equations were then expanded into a set of ordinary differential equations wherein the space derivative was discretized but time kept continuous i.e., bed length was divided into finite stages. The resulting set of ordinary differential equations was solved analytically, the constants of integration being obtained from cyclic equilibrium conditions. This resulted in a number of algebraic equations which was twice the number of nodes used along the length (or the bed) of the regenerator. This set of equations was then solved on a digital computer. As many as 400 nodes were used in this analysis. Then assuming an initial 'blast' temperature (dimensionless) along with an 'air exit' temperature t''_{out}, a new value of t''_{out} was computed from a heat balance equation. In most cases the resulting value was not found to be the assumed value of t''_{out}. Only after several trials was an acceptable match achieved. The process was required to be repeated if the 'blast' temperature did not 'converge' either.

A similar method was developed, based on Jeffeson [9] by the author of this thesis whereby a program FISZEFC has been written to obtain profiles. It was observed that the truncation error limits accuracy. ✓ However, the derivations, program and some results are presented as an appendix.

2.12. Chapter Summary:

This chapter has been called 'Physical assumptions, Mathematical model and Review of previous work' and so as the title suggests at first in section 2.1 the physical assumptions and mathematical formulation of regenerator model are presented. The four basic assumptions made are well justified as supported in the work done by authors such as Nusselt [18], Willmott and Thomas [26], Heggs and Carpenter [4], Jeffreson [8] and others.

The mathematical formulation is presented next and here equations are derived from first principles. Note that here subscript 1 refers to real (i.e., non-normalized) parameters. So in this section most general form of heat balance equations (2.1.2.(6)) and (2.1.2(7)) representing a thermal regenerator are obtained. In section 2.2 at first, the real parameters x and y are normalized to obtain normalized parameters z and θ. Then the fluid and solid temperatures themselves are normalized and equations (2.2.(1)), (2.2.(2)) are obtained. This is followed by application of physical assumptions and usage of Hausen's [1] normalization to give us equations (2.2.(4)) and (2.2.(6)) which are simpler than before. In section 2.3, a description of Thermal efficiency and Thermal ratio is given. This is followed by a treatise of single blow solutions of Anzelius and Nusselt in section 2.4 in which solutions are obtained in the form of equations (2.4.(8)) and (2.4.(9)) which involve evaluation of Bessel functions $J_o(x)$ and $J_1(x)$. Section 2.5 describes the reversal conditions. Then in section 2.6 a review of "open" methods of solution of regenerator problem is done. The methods reviewed are due to Lambertson [5], Willmott's trapezoidal method [23] and Jeffreson's method [8]. Section 2.7 deals with obtaining analytical solutions U and V where equations (2.7.(1)) and (2.7.(2)) are Laplace transformed with respect to the distance parameter q and through further analysis functions U and V are obtained which are then used in deriving T' and T" through equations (2.7.(16)) and (2.7.(17)) followed by t' and t" through equations (2.7.(18)) and (2.7.(19)) respectively. It is noted that this analytical solution subject to reversal conditions could be used to generate "open" solutions by repeated cycling but "closed" form solutions may be found to be more efficient.

Therefore, section 2.8 reviews "closed" methods. In particular, Hausen [1], Illiffe [6] and Nahavandi and Weinstein [17] methods are looked at. In section 2.9, a detailed reviewing of Nahavandi and Weinstein (NW) method is done. It is noted here that NW considered the rather simplified case of a balanced symmetric regenerator and their method used evaluation of integrals through quadrature. This is followed by an overview of "closed" methods using consistent notations and it is concluded that NW method can be classed as the most reliable of the closed methods reviewed.

Finally in section 2.11, a finite stage method due to Razelos and Paschkis [19] is outlined and it is noted that a similar method is presented as an appendix. (See Appendix A4).

...

CHAPTER 3: A NEW METHOD FOR SOLVING THE COUNTERFLOW REGENERATOR PROBLEM

It was noted in section 2.10 of the previous chapter that Nahavandi and Weinstein (NW) method [17] can be classed as one of the most reliable methods for solving the counterflow regenerator problem. However, NW only considered a symmetric balanced regenerator and their method depended on evaluation of Bessel functions which involve evaluation of integrals such as:

$$iJ_1(2i\pi z) = \frac{1}{\pi}\int_0^{\pi} e^{-2z\pi \cos(s)} \cos(s)ds \qquad .. (3.(1))$$

obtained by using a quadrature formula.

In this chapter we develop a method based on the approach used by NW; however it avoids the use of quadrature formulae by making use of numerical inversion of Laplace transforms. A general non-symmetric, unbalanced regenerator is considered here as opposed to NW's balanced symmetric treatment. In the later part of this chapter we describe the particular numerical Laplace inversion technique developed by Zakian [27]. However, for present the method assumes the existence of a reliable numerical Laplace inversion scheme. We shall describe the advantages of using a numerical Laplace inversion technique as the analysis of the method continues. The proposed method has been developed for the normalized distance parameters z and q separately as in each case different matrix equations are solved, nevertheless resulting in comparable final values. We first deal with the analysis using variable q.

3.1. Proposed analytical closed method using parameter q:

In this section we first derive equilibrium solid temperature distribution in polynomial form, this is followed by derivation of equilibrium fluid temperatures. Based on a personal communication with Jeffreson [10] we solve the following:

$$\frac{\partial t}{\partial q} = T - t \qquad .. (3.1.(1))$$

$$\frac{\partial T}{\partial r} = t - T \qquad .. (3.1.(2))$$

See sections 2.2 and 2.7.

where $q = \lambda z$, i.e., $0 \leqslant q \leqslant \lambda$ since $0 \leqslant z \leqslant 1$.

Now Laplace transforming the above equations with respect to q (transform variable p) implies:

$$p.\hat{t}(p,r) = t(q=0,r) + \hat{T} - \hat{t} \qquad .. (3.1.(3))$$

and

$$\frac{d\hat{T}}{dr} = \hat{t} - \hat{T} \qquad .. (3.1.(4))$$

where ^ refers to the transformed function, $t(q=0,r) = t_{in}(r)$ is temperature of fluid at bed inlet.

Eliminating $\hat{t}$ between equations (3.1.(3)) and (3.1.(4)) we have:

$$\frac{d\hat{T}}{dr} + \frac{p}{p+1}\hat{T} = \frac{t_{in}(r)}{p+1} \qquad .. (3.1.(5))$$

Solution of (3.1.(5)) is given by:

$$\hat{T}(p,r) = \hat{V}(p,r).\hat{T}(p,0) + \int_0^r \hat{V}(p,r-w)\,\frac{t_{in}(w)}{p+1}\,dw \qquad .. (3.1.(6))$$

where,

$$\hat{V}(p,r) = \exp(-\frac{pr}{p+1}).$$

For hot blow:

$$t(q'=0,r) = t_{in}(r) = \begin{cases} 0, & r < 0 \\ 1, & r \geqslant 0 \end{cases} = \mu(r), \text{ the unit step function}$$

and for cold blow:

$$t(q''=0,r) = 0.$$

So, for hot blow (3.1.(6)) implies:

$$\begin{aligned}\hat{T}'(p,r) &= \hat{V}(p,r).\hat{T}'(p,0) + \int_0^r \hat{V}(p,r-w)\,\frac{\mu(w)}{p+1}\,dw \\ &= \hat{V}(p,r).\hat{T}'(p,0) + \hat{U}'(p,r) \qquad .. (3.1.(7))\end{aligned}$$

where

$$\hat{U}'(p,r) = \int_0^r \frac{\hat{V}(p,w)}{p+1}\,dw$$

and for cold blow (3.1.(6)) implies:

$$\hat{T}''(p,r) = \hat{V}(p,r).\hat{T}''(p,0) \qquad .. (3.1.(8))$$

Inverse transforming (3.1.(7)) and using convolution theorem gives us:

$$T'(q',r) = U'(q',r) + \int_0^{q'} V'(q'-v,r).T'(v,0)dv \quad .. (3.1.(9))$$

during hot blow and inverse transforming (3.1.(8)) yields:

$$T''(q'',r) = \int_0^{q''} V''(q''-v,r).T''(v,0)dv \quad .. (3.1.(10))$$

during cold blow, where T(q,0) is the initial solid temperature distribution. Application of reversal conditions implies:

$$z' = 1-z'' \quad , \qquad 0 \leq z' \leq 1$$
$$q' = \lambda' z' \quad , \qquad 0 \leq q' \leq \lambda'$$
$$q'' = \lambda'' z'' \quad , \qquad 0 \leq q'' \leq \lambda''$$

So,

$$q'/\lambda' = 1 - q''/\lambda'' \text{ i.e., } q' = \lambda'(1-q''/\lambda'')$$
$$\text{and} \quad q'' = \lambda''(1-q'/\lambda')$$

Based on equations (2.5.(4)) and (2.5.(5)) the reversal conditions for solid can be written as:

$$T'(q',0) = T''(q'',\pi'') \quad .. (3.1.(11))$$

$$T''(q'',0) = T'(q',\pi') \quad .. (3.1.(12))$$

So (3.1.(9)) and (3.1.(10)) imply:

$$T'(q',r'=0) = T'(\lambda'(1-q''/\lambda''),r'=0) = T''(q'',\pi'')$$
$$= \int_0^{q''} V''(q''-v,\pi'').T''(v,0)dv \quad .. (3.1.(13))$$

and

$$T''(q'',r''=0) = T''(\lambda'(1-q'/\lambda'),r''=0) = T'(q',\pi')$$

$$= U'(q',\pi') + \int_0^{q'} V'(q'-v,\pi').T'(v,0)dv \qquad \text{.. (3.1.(14))}$$

So, it can be seen that solid temperature profiles may be obtained from arbitrary initial temperature distributions via equations (3.1.(13)) and (3.1.(14)).

We now employ the technique used by NW and assume:

$$T'(q',0) = \sum_{i=1}^{N} a_i'(q')^{i-1}, \quad T''(q'',0) = \sum_{i=1}^{N} a_i''(q'')^{i-1}$$

So, Laplace transforming (3.1.(13)) with respect to the distance parameter we have:

$$L\left(\int_0^{q''} V''(q''-v,\pi'').T''(v,0)dv\right)$$

$$= \sum_{i=1}^{N} a_i'' \, L\left(\int_0^{q''} V''(q''-v,\pi'').(v)^{i-1}dv\right)$$

$$= \sum_{i=1}^{N} a_i'' \, \hat{V}''(p,\pi''). \frac{(i-1)!}{p^i}$$

using convolution theorem:

$$(L(f*g) = L(f)L(g)) \text{ where } f*g = \int_0^{t} f(w)g(t-w)dw)$$

and so,

$$\sum_{i=1}^{N} a_i'(q')^{i-1} = L^{-1}\left(\sum_{i=1}^{N} a_i'' \, \hat{V}''(p,\pi'') \frac{(i-1)!}{p^i}\right) \qquad \text{.. (3.1.(15))}$$

Similarly from (3.1.(14)) we obtain:

$$\sum_{i-1}^{N} a''_i (q'')^{i-1} = \sum_{i=1}^{N} a''_i [\lambda''(1-q'/\lambda')]^{i-1}$$

$$= L^{-1}(\hat{U}'(p,\pi') + \sum_{i=1}^{N} a'_i \hat{V}'(p,\pi') \frac{(i-1)!}{p^i}) \qquad \text{.. (3.1.(16))}$$

So we now see that the problem of evaluating the integrals (as in the case of NW) has been reduced to obtaining the coefficients a'_i, a''_i. Using N terms in each series we require N values of q: q'_i twice to obtain coefficients a'_i, a''_i ; $i = 1,2,\ldots,N$.

So, (3.1.(15)) and (3.1.(16)) imply:

$$\sum_{i=1}^{N} a'_i (q'_j)^{i-1} = L^{-1}(\sum_{i=1}^{N} a''_i \hat{V}''(p,\pi'') \frac{(i-1)!}{p^i})]_{q=\lambda''(1-q'_j/\lambda')} \qquad \text{.. (3.1.(17))}$$

and

$$\sum_{i=1}^{N} a''_i (\lambda''(1-q'_j/\lambda'))^{i-1}$$

$$= L^{-1}(\hat{U}'(p,\pi') + \sum_{i=1}^{N} a'_i \hat{V}'(p,\pi') \frac{(i-1)!}{p^i})]_{q=q'_j} \qquad \text{.. (3.1.(18))}$$

$; \; j = 1,2,\ldots,N$

Writing (3.1.(17)) and (3.1.(18)) in matrix form we have:

$$[B']\underline{a}' = [C'']\underline{a}'' \qquad \text{.. (3.1.(19))}$$

and

$$[B'']\underline{a}'' = \underline{U} + [C']\underline{a}' \qquad .. (3.1.(20))$$

where,

$$\underline{a}' = \begin{bmatrix} a'_1 \\ a'_2 \\ \cdot \\ \cdot \\ a'_N \end{bmatrix}, \ \underline{a}'' = \begin{bmatrix} a''_1 \\ a''_2 \\ \cdot \\ \cdot \\ a''_N \end{bmatrix}, \ [B'] = \begin{bmatrix} 1 & q'_1 & (q'_1)^2 & \dots & (q'_1)^{N-1} \\ 1 & q'_2 & (q'_2)^2 & \dots & (q'_2)^{N-1} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 1 & q'_N & (q'_N)^2 & \dots & (q'_N)^{N-1} \end{bmatrix}$$

$$[B''] = \begin{bmatrix} 1 & \lambda''(1-q'_1/\lambda')\dots(\lambda''(1-q'_1/\lambda'))^{N-1} \\ 1 & \lambda''(1-q'_2/\lambda')\dots(\lambda''(1-q'_2/\lambda'))^{N-1} \\ \cdot & \cdot \qquad\qquad \cdot \\ \cdot & \cdot \qquad\qquad \cdot \\ 1 & \lambda''(1-q'_N/\lambda')\dots(\lambda''(1-q'_N/\lambda'))^{N-1} \end{bmatrix}, \ [C''] = (C''_{ij}) \ ,$$

$$[C'] = (C'_{ij}) \ \text{ and } \ \underline{U} = (U_i)$$

where

$$C''_{ij} = L^{-1} \left(\frac{\hat{V}''(p,\pi'')}{p^j} \right)]_{q = \lambda''(1-q'_i/\lambda')} \cdot (j-1)!$$

$$C'_{ij} = L^{-1} \left(\frac{\hat{V}'(p,\pi')}{p^j} \right)]_{q = q'_i} \cdot (j-1)!$$

and

$$U_i = L^{-1}\left(\frac{1}{p} [1 - \exp(\frac{-p}{p+1} \pi')]\right)]_{q = q'_i} .$$

since

$$\hat{U}(p,r) = \int_{0}^{r} \frac{\hat{V}(p,w)}{p+1}\, dw = \frac{1}{p+1} \int_{0}^{r} \exp(-pw/(p+1))dw$$

$$= - \frac{1}{p} [\exp(-pw/(p+1))]_{w=0}^{r}$$

$$= \frac{1}{p} [1 - \exp(-pr/(p+1))].$$

We note that, Jeffreson [9] has already shown:

$$\lim_{r \to 0} U'(q',r) = e^{-\lambda' q'}$$

Now, initial value theorem for Laplace transforms says:

$$\lim_{t \to 0} f(t) = \lim_{s \to \infty} sF(s)$$

therefore,

$$L^{-1}(\frac{1}{p} [1 - \exp(-p\pi'/(p+1))])]_{q=0}$$

$$= \lim_{p \to \infty} [1 - \exp(-p\pi'/(p+1))] = 1 - \exp(-\pi') \qquad .. (3.1.(21))$$

and

$$L^{-1}(\frac{\hat{V}(p,\pi)}{p^{j}})]_{q=0} = \lim_{p \to \infty} \frac{\hat{V}(p,\pi)}{p^{j-1}} \quad ; \quad j = 1,\ldots,N$$

$$= \lim_{p \to \infty} \frac{\exp(-p\pi/(p+1))}{p^{j-1}} = \begin{cases} e^{-\pi} & ; j = 1 \\ 0 & ; j=2,3,..,N \end{cases}$$

Now equations (3.1.(19)) and (3.1.(20)) imply:

$$\underline{a}'' = [B'']^{-1} [\underline{U} + [C']\underline{a}']$$

$$= [B'']^{-1} \underline{U} + [B'']^{-1}[C']\underline{a}'$$

Therefore,

$$[B']\underline{a}' = [C'']([B'']^{-1} \underline{U} + [B'']^{-1}[C']\underline{a}')$$

or

$$([B'] - [C''][B'']^{-1}[C'])\underline{a}' = [C''][B'']^{-1} \underline{U}$$

Hence

$$\underline{a}' = [[B'] - [C''][B'']^{-1}[C']]^{-1} [C''][B'']^{-1} \underline{U} \qquad .. (3.1.(22))$$

and

$$\underline{a}'' = [B'']^{-1} \underline{U} + [B'']^{-1}[C']\underline{a}' \qquad .. (3.1.(23))$$

So having solved for coefficients a' and a", one can easily obtain the solid temperatures as follows:

$$T'(q',0) = T''(q'',\pi'') = \sum_{i=1}^{N} a_i'(q')^{i-1} \qquad .. (3.1.(24))$$

and

$$T''(q'',0) = T'(q',\pi') = \sum_{i=1}^{N} a_i''(q'')^{i-1} \qquad .. (3.1.(25))$$

So, now the fluid temperatures can also be determined using equation (3.1.(3)) as follows:

$$\hat{t}(p,r) = \frac{1}{p} [t(q=0,r) + \hat{T} - \hat{t}]$$

or

$$\hat{t}(p,r)[1 + \frac{1}{p}] = \frac{1}{p} [t(q=0,r) + \hat{T}]$$

Therefore,

$$\hat{t}(p,r) = \frac{1}{p+1} [t(q=0,r) + \hat{T}(p,r)]$$

So for hot blow:

$$\hat{t}'(p,r') = \frac{1}{p+1} [1 + \hat{T}(p,r')] \quad .. (3.1.(26))$$

and for cold blow:

$$\hat{t}''(p,r'') = \frac{1}{p+1} \hat{T}(p,r'') \quad .. (3.1.(27))$$

Hence

$$t'(q',0) = L^{-1}[\frac{1}{p+1} (1 + \sum_{i=1}^{N} a_i' \frac{(i-1)!}{p^i})]_{q=q'} \quad .. (3.1.(28))$$

and

$$t''(q'',0) = L^{-1}[\frac{1}{p+1} \sum_{i=1}^{N} a_i'' \frac{(i-1)!}{p^i}]_{q=q''} \quad .. (3.1.(29))$$

since

$$L(q^{i-1}) = \frac{(i-1)!}{p^i} \quad ; \quad i = 1,2,\ldots,N.$$

But,

$$L^{-1}(\frac{1}{p+1} \cdot \frac{1}{p^i})(q) = \int_0^q e^{-v} \frac{(q-v)^{i-1}}{(i-1)!} dv$$

using convolution theorem, as

$$L(e^{-q}) = \frac{1}{p+1} , \quad L(\frac{q^{i-1}}{(i-1)!}) = \frac{1}{p^i} .$$

Hence,

$$t'(q',0) = e^{-q} + \sum_{i=1}^{N} a_i' \int_0^{q'} e^{-v}(q'-v)^{i-1}dv \quad .. (3.1.(30))$$

and

$$t''(q'',0) = \sum_{i=1}^{N} a_i'' \int_0^{q''} e^{-v}(q''-v)^{i-1}dv \quad .. (3.1.(31))$$

Consider

$$I_n = \int_0^{q} e^{-v}(q-v)^n dv \ , \quad n \geqslant 0; \ n \ \varepsilon \text{ set of Natural numbers}$$

So, using integration by parts, we have:

$$I_n = -(q-v)^n \ e^{-v}\Big]_{v=0}^{q} - \int_0^{q} n(q-v)^{n-1} \ e^{-v} \ dv$$

$$= q^n - nI_{n-1} \quad ; \quad n = 1,2,...,N$$

where,

$$I_o = \int_0^{q} e^{-v} \ dv = -e^{-v} \ \Big]_{v=0}^{q} = 1 - e^{-q} \ .$$

Hence,

$$t'(q',0) = e^{-q} + \sum_{i=1}^{N} a_i' \ I_{i-1}' \quad .. (3.1.(32))$$

where,

$$I_o' = 1 - e^{-q'} \ , \text{ and } \ I_i' = (q')^i - iI_{i-1}' \ ; \ i = 1,2,..,N-1$$

and

$$t''(q'',0) = \sum_{i=1}^{N} a_i''I_{i-1}'' \quad .. (3.1.(33))$$

where,

$$I_o'' = 1 - e^{-q''}, \text{ and } I_i'' = (q'')^i - iI_{i-1}'' \ ; \ i = 1,2,..,N-1.$$

Using reversal conditions we have

$$t'(q',0) = t''(q'',\pi'')$$

and

$$t''(q'',0) = t'(q',\pi').$$

So fluid temperatures have been derived in terms of the coefficients a_i' and a_i''; $i = 1,2,\ldots,N$.

3.2. Thermal efficiency:

From equation (2.3.(4)) we have thermal efficiency E given by:

$$E = \frac{MC\left\{\int_0^1 T''(z,0)dz - \int_0^1 T'(z,0)dz\right\}}{W'S'P'(t_{in}' - t_{in}'')}$$

So for $t_{in}'' = 0$, $t_{in}' = 1$ and using the q normalisation $(0 \leqslant q \leqslant \lambda)$, we have:

$$E = \frac{MC}{h'AP'}\,\frac{h'A}{W'S'}\left\{\frac{1}{\lambda''}\int_0^{\lambda''} T''(q'',0)dq'' - \right.$$

$$\left. - \frac{1}{\lambda'}\int_0^{\lambda'} T'(q',0)dq'\right\}$$

$$= \frac{\lambda'}{\pi'}\left\{\frac{1}{\lambda''}\int_0^{\lambda''} T''(q'',0)dq'' - \frac{1}{\lambda'}\int_0^{\lambda'} T'(q',0)dq'\right\}$$

.. (3.2.(1))

or

$$E = \frac{\lambda'}{\pi'}\left\{\frac{1}{\lambda''}\int_0^{\lambda''}\sum_{i=1}^{N} a_i''(q'')^{i-1}dq'' - \frac{1}{\lambda'}\int_0^{\lambda'}\sum_{i=1}^{N} a_i'(q')^{i-1}dq'\right\}$$

$$= \frac{\lambda'}{\pi'}\left\{\frac{1}{\lambda''}\sum_{i=1}^{N} a_i''\left[\frac{(q'')^i}{i}\right]_{q''=0}^{\lambda''} - \frac{1}{\lambda'}\sum_{i=1}^{N} a_i'\left[\frac{(q')^i}{i}\right]_{q'=0}^{\lambda'}\right\}$$

Therefore,

$$E = \frac{\lambda'}{\pi'}\left\{\frac{1}{\lambda''}\sum_{i=1}^{N} a_i''\frac{(\lambda'')^i}{i} - \frac{1}{\lambda'}\sum_{i=1}^{N} a_i'\frac{(\lambda')^i}{i}\right\}$$

.. (3.2.(2))

Here also it is noticeable that the integration has been done analytically and there is no dependence on quadratures as such for calculation of integrals.

3.3. Steady state limiting temperature distributions:

For the purposes of comparisons with the transient states it is advantageous to obtain the steady state temperature distributions for limiting recuperator. We obtain these now.

Differential equations prior to complete normalization:

For 'cold' fluid (from(2.1.2.(6)) with z=x/L and zero conduction):

$$-W''S''\frac{\partial t''}{\partial z} + h''A(T - t'') = m''S''\frac{\partial t''}{\partial y} \qquad .. (3.3.(1))$$

For 'hot' fluid (from (2.1.2.(6)) with z=x/L and zero conduction):

$$W'S' \frac{\partial t'}{\partial z} + h'A(T - t') = m'S' \frac{\partial t'}{\partial y} \qquad .. (3.3.(2))$$

Solid temperatures(from(2.1.2.(7)) with z=x/L, zero conduction):

$$MC \frac{\partial T}{\partial y} = h' A (t' - T) + h'' A (t'' - T)$$

$$.. (3.3.(3))$$

where z is normalized distance relative to bed length ($0 \leq z \leq 1$), y is real time, T and t are non-normalized solid and fluid temperatures.

For 'steady state', variables will be denoted by a bar over them.

We know that for 'steady state' derivative with respect to time is zero. Hence (3.3.(1)) to (3.3.(3)) become:

$$-\bar{W}''\bar{S}'' \frac{d\bar{t}''}{dz} + \bar{h}''A(\bar{T} - \bar{t}'') = 0 \qquad .. (3.3.(4))$$

$$\bar{W}'\bar{S}' \frac{d\bar{t}'}{dz} + \bar{h}'A(\bar{T} - \bar{t}') = 0 \qquad .. (3.3.(5))$$

and

$$\bar{h}'A(\bar{t}' - \bar{T}) + \bar{h}''A(\bar{t}'' - \bar{T}) = 0 \qquad .. (3.3.(6))$$

Defining

$$q = q'' = \lambda''z, \ \gamma_1 = \frac{\bar{h}'A}{\bar{h}''A}, \ \gamma_1 \lambda'' = \frac{\bar{h}'A}{\bar{W}''\bar{S}''} \text{ since } \lambda'' = \frac{\bar{h}''A}{\bar{W}''\bar{S}''},$$

$$\gamma_2 = \lambda'/\lambda'' = \gamma_1 \frac{\bar{W}''\bar{S}''}{\bar{W}'\bar{S}'} = \frac{\bar{h}'A}{\bar{h}''A} \cdot \frac{\bar{W}''\bar{S}''}{\bar{W}'\bar{S}'} .$$

Therefore,

$$\frac{\gamma_2}{\gamma_1} = \frac{\bar{W}''\ \bar{S}''}{\bar{W}'\ \bar{S}'} .$$ So (3.3.(4)) implies on division by $\bar{W}''\bar{S}''$:

$$-\frac{d\bar{t}''}{dz} + \frac{\bar{h}''A}{\bar{W}''\ \bar{S}''}(\bar{T} - \bar{t}'') = 0 \qquad .. (3.3.(7))$$

and (3.3.(5)) gives on division by $\bar{W}'\bar{S}'$:

$$\frac{d\bar{t}'}{dz} + \frac{\bar{h}'A}{\bar{W}'\ \bar{S}'}(\bar{T} - \bar{t}') = 0 \qquad .. (3.3.(8))$$

while (3.3.(6)) becomes after division by $\bar{h}''A$:

$$\frac{\bar{h}'A}{\bar{h}''A}(\bar{t}' - \bar{T}) + (\bar{t}'' - \bar{T}) = 0$$

Therefore,

$$\gamma_1(\bar{T} - \bar{t}') - (\bar{t}'' - \bar{T}) = 0 \qquad .. (3.3.(9))$$

Now $dq = \lambda'' dz$

Therefore, $dz = dq / \lambda''$

So (3.3.(7)) implies

$$-\frac{d\bar{t}''}{dq} + \bar{T} - \bar{t}'' = 0 \qquad .. (3.3.(10))$$

(3.3.(8)) gives

$$\frac{d\bar{t}'}{dq} + \gamma_2(\bar{T} - \bar{t}') = 0 \qquad .. (3.3.(11))$$

Hence the steady state equations to be solved are (3.3.(9)), (3.3.(10)) and (3.3.(11)).

From (3.3.(9)) we have:

$$\bar{T} = \frac{\gamma_1 \bar{t}'}{\gamma_1 + 1} + \frac{1}{\gamma_1 + 1}\bar{t}'' \qquad .. (3.3.(12))$$

Substituting (3.3.(12)) in (3.3.(10)) we have:

$$\frac{d\bar{t}''}{dq} = \bar{t}''\left(\frac{1}{\gamma_1 + 1} - 1\right) + \frac{\gamma_1\ \bar{t}'}{\gamma_1 + 1} = -\frac{\gamma_1}{\gamma_1 + 1}(\bar{t}'' - \bar{t}')$$

and (3.3.(12)) in (3.3.(11)) gives:

$$\frac{d\bar{t}'}{dq} = \bar{t}'\left(\gamma_2 - \frac{\gamma_1\ \gamma_2}{\gamma_1 + 1}\right) - \frac{\gamma_2}{\gamma_1 + 1}\bar{t}'' = -\frac{\gamma_2}{\gamma_1 + 1}(\bar{t}'' - \bar{t}')$$

So,

$$\frac{d(\bar{t}'' - \bar{t}')}{dq} = \frac{\gamma_2 - \gamma_1}{\gamma_1 + 1}(\bar{t}'' - \bar{t}') \qquad .. (3.3.(13))$$

Integrating yields

$$\ln(\bar{t}'' - \bar{t}')\Big]_{o}^{q} = \frac{\gamma_2 - \gamma_1}{\gamma_1 + 1}q$$

i.e., $\ln(\bar{t}''(q) - \bar{t}'(q)) - \ln(\bar{t}''_{in} - \bar{t}_{out}) = vq$

where $v = \dfrac{\gamma_2 - \gamma_1}{\gamma_1 + 1}$

or $\ln \dfrac{\bar{t}''(q) - \bar{t}'(q)}{\bar{t}''_{in} - \bar{t}'_{out}} = vq$

Therefore, $\bar{t}''(q) - \bar{t}'(q) = (\bar{t}''_{in} - \bar{t}'_{out})\, e^{vq}$ .. (3.3.(14))

But in steady state:

$$\bar{W}''\ \bar{S}''(\bar{t}''(q) - \bar{t}''_{out}) = \bar{W}'\ \bar{S}'(\bar{t}'(q) - \bar{t}'_{in})$$

So $\bar{t}''(q) - \dfrac{\gamma_1}{\gamma_2}\bar{t}'(q) = \bar{t}''_{out} - \dfrac{\gamma_1}{\gamma_2}\bar{t}'_{in}$ .. (3.3.(15))

(3.3.(14)) implies, at $q = \lambda''$, we have:

$$\bar{t}''_{out} = \bar{t}'_{in} + (\bar{t}''_{in} - \bar{t}'_{out})\, e^{vq}$$

i.e., $\bar{t}''_{out} = \bar{t}'_{in} + w\,\bar{t}''_{in} - w\,\bar{t}'_{out}$ .. (3.3.(16))

where $w = \exp\left\{\dfrac{\gamma_2 - \gamma_1}{\gamma_1 + 1}\lambda''\right\}$

(3.3.(15)) means that at $q = 0$, we have:

$$\bar{t}''_{in} - \bar{t}''_{out} = \frac{\gamma_1}{\gamma_2}(\bar{t}'_{out} - \bar{t}'_{in})$$

i.e., $\bar{t}'_{out} = [\frac{\gamma_2}{\gamma_1} (\bar{t}''_{in} - \bar{t}''_{out}) + \bar{t}'_{in}]$ .. (3.3.(17))

Substituting (3.3.(17)) into (3.3.(16)) we have:

$$\bar{t}''_{out} = \bar{t}'_{in} + w\,\bar{t}''_{in} - w[\frac{\gamma_2}{\gamma_1} \bar{t}''_{in} - \frac{\gamma_2}{\gamma_1} \bar{t}''_{out} + \bar{t}'_{in}]$$

Therefore,

$$\bar{t}''_{out} = \frac{1 - w}{1 - w\;\gamma_2/\gamma_1} \bar{t}'_{in} + w \frac{1 - \gamma_2/\gamma_1}{1 - w\;\gamma_2/\gamma_1} \bar{t}''_{in} \qquad .. (3.3.(18))$$

and so (3.3.(16)) gives:

$$\bar{t}'_{out} = \frac{\gamma_2}{\gamma_1} [\bar{t}''_{in}(1 - w \frac{(1 - \gamma_2/\gamma_1)}{1 - w\;\gamma_2/\gamma_1}) + \frac{w - 1}{1 - w\;\gamma_2/\gamma_1} \bar{t}'_{in}] + \bar{t}'_{in}$$

Therefore,

$$\bar{t}'_{out} = \frac{\gamma_2}{\gamma_1} \frac{(1 - w)}{1 - w\;\gamma_2/\gamma_1} \bar{t}''_{in} + \frac{1 - \gamma_2/\gamma_1}{1 - w\;\gamma_2/\gamma_1} \bar{t}'_{in} \qquad .. (3.3.(19))$$

Hence we can solve for $\bar{t}''(q)$ and $\bar{t}'(q)$ from (3.3.(14)) and (3.3.(15)) as follows:

Right hand side (RHS) of (3.3.(14))

$$= [\bar{t}''_{in} -(\frac{\gamma_2}{\gamma_1} \frac{(1 - w)}{1 - w\;\gamma_2/\gamma_1} \bar{t}''_{in} + \frac{1 - \gamma_2/\gamma_1}{1 - w\;\gamma_2/\gamma_1} \bar{t}'_{in})]\; e^{vq}$$

$$= \frac{1 - \gamma_2/\gamma_1}{1 - w\,\gamma_2/\gamma_1} (\bar{t}''_{in} - \bar{t}'_{in})\, e^{vq}$$

RHS (3.3.(15))

$$= \left(- \frac{\gamma_1}{\gamma_2} + \frac{1 - w}{1 - w\,\gamma_2/\gamma_1} \right) \bar{t}'_{in} + w\, \frac{1 - \gamma_2/\gamma_1}{1 - w\,\gamma_2/\gamma_1}\, \bar{t}''_{in}$$

$$= \frac{1}{1 - w\,\gamma_2/\gamma_1} \left[(1 - \gamma_1/\gamma_2)\bar{t}'_{in} + w(1 - \gamma_2/\gamma_1)\bar{t}''_{in} \right]$$

Let $\underline{B}$ $$= \frac{1}{1 - w\,\gamma_2/\gamma_1} \begin{bmatrix} ((1 - \gamma_2/\gamma_1)\bar{t}''_{in} - \bar{t}'_{in})\, e^{vq} \\ \\ (1 - \gamma_1/\gamma_2)\bar{t}'_{in} + w(1 - \gamma_2/\gamma_1)\bar{t}''_{in} \end{bmatrix}$$

Hence $A\underline{t} = \underline{B}$ where $\underline{t} = [\ \bar{t}''(q), \bar{\theta}(q)]^T$

and

$$A = \begin{bmatrix} 1 & -1 \\ 1 & -\gamma_1/\gamma_2 \end{bmatrix}$$ from (3.3.(14)) and (3.3.(15))

Now $$A^{-1} = \frac{1}{1 - \gamma_1/\gamma_2} \begin{bmatrix} -\gamma_1/\gamma_2 & 1 \\ -1 & 1 \end{bmatrix}$$

Hence $\underline{t} = A^{-1}\,\underline{B}$

$$\underline{t} = \frac{1}{1 - \gamma_1/\gamma_2} \begin{bmatrix} -\gamma_1/\gamma_2 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} (1-\gamma_2/\gamma_1)(\bar{t}''_{in}-\bar{t}'_{in})e^{vq} \\ (1-\gamma_1/\gamma_2)\bar{t}'_{in} + w(1-\gamma_2/\gamma_1)\bar{t}''_{in} \end{bmatrix} \cdot \frac{1}{1 - w\,\gamma_2/\gamma_1}$$

So $$t''(q) = \frac{1}{1 - w\,\gamma_2/\gamma_1}\left[(e^{vq} - w\frac{\gamma_2}{\gamma_1})\bar{t}''_{in} + (1 - e^{vq})\bar{t}'_{in}\right] \qquad ..\ (3.3.(20))$$

and

$$t'(q) = \frac{1}{1 - w\,\gamma_2/\gamma_1}\left[\ (1 - \frac{\gamma_2}{\gamma_1}e^{vq})\bar{t}'_{in} + (\frac{\gamma_2}{\gamma_1}e^{vq} - w\frac{\gamma_2}{\gamma_1})\bar{t}''_{in}\right] \qquad ..\ (3.3.(21))$$

since $$\frac{1 - \gamma_2/\gamma_1}{1 - \gamma_1/\gamma_2} = -\frac{\gamma_2}{\gamma_1}$$

Hence (3.3.(12)) implies:

$$\bar{T}(q) = \frac{1}{\gamma_1 + 1}\left[\ \gamma_1\ \bar{t}'(q) + \bar{t}''(q)\ \right]$$

$$= \frac{1}{(\gamma_1+1)(1-w\gamma_2/\gamma_1)}\ [((\gamma_1+1)(1-e^{vq}) + (\gamma_1-\gamma_2)e^{vq})\bar{t}'_{in}$$
$$+ ((\gamma_1+1)(e^{vq}-w\gamma_2/\gamma_1)-(\gamma_1-\gamma_2)e^{vq})\bar{t}''_{in}\ ] \qquad ..\ (3.3.(22))$$

So equations (3.3.(20)), (3.3.(21)) and (3.3.(22)) give the steady state limiting temperatures in terms of arbitrary steady state inlet fluid temperatures $\bar{t}'_{in}$ and $\bar{t}''_{in}$.

For our regenerator, we choose to normalise the fluid temperatures so that $\bar{t}'_{in} = 1$, $\bar{t}''_{in} = 0$.

$$\text{So,}\quad \bar{T}(q) = \frac{1 - e^{vq}}{1 - w\,\gamma_2/\gamma_1} + \frac{(\gamma_1 - \gamma_2)e^{vq}}{(\gamma_1 + 1)(1 - w\,\gamma_2/\gamma_1)}$$

$$= \frac{1 - e^{vq}}{1 - w\,\gamma_2/\gamma_1} - \frac{v\,e^{vq}}{(1 - w\,\gamma_2/\gamma_1)} \quad \text{as} \quad v = \frac{\gamma_2 - \gamma_1}{\gamma_1 + 1}$$

$$= \frac{1}{1 - w\,\gamma_2/\gamma_1}\left[\, 1 - e^{vq} - ve^{vq}\,\right]$$

For limiting recuperator we must find $\lim\limits_{\gamma_1,\gamma_2 \to 1} \bar{T}(q)$

Now because $v = \dfrac{\gamma_2 - \gamma_1}{\gamma_1 + 1}$ and $w = e^{v\lambda''}$

as $\gamma_2 \to \gamma_1 \to 1$ we have $v \to 0$ and so $w \to 1$.

Therefore, $\lim\limits_{\gamma_1,\gamma_2 \to 1} \bar{T}(q) \longrightarrow 0/0$

$$\text{So}\quad \lim_{\gamma_1,\gamma_2 \to 1} \bar{T}(q) = \lim_{\substack{\gamma_1 \to 1\\ \gamma_2 = 1}} \bar{T}(q)$$

$$= \lim_{\gamma_1 \to 1} \frac{1}{1 - \exp\left(\frac{1-\gamma_1}{\gamma_1+1}\lambda''\right)/\gamma_1}\left[1 - \exp\left(\frac{1-\gamma_1}{\gamma_1+1}\,q\right)\left(1 + \frac{1-\gamma_1}{\gamma_1+1}\right)\right]$$

Now applying L'Ôpital's rule we have

$$\lim_{\gamma_1, \gamma_2 \to 1} \bar{T}(q) = \lim_{\gamma_1 \to 1} \frac{\frac{d}{d\gamma_1}\left[-\exp\left(\frac{1-\gamma_1}{\gamma_{+}+1} q\right)\left(\frac{2}{\gamma_1+1}\right)\right]}{\frac{d}{d\gamma_1}\left[-\exp\left(\frac{1-\gamma_1}{\gamma_1+1} \lambda''\right)/\gamma_1\right]}$$

$$= \lim_{\gamma_1 \to 1} \frac{-\exp\left(\frac{1-\gamma_1}{\gamma_1+1} q\right)\left(\frac{-(\gamma_1+1)-(1-\gamma_1)}{(\gamma_1+1)^2} q\right) \frac{2}{\gamma_1} + \; + \exp\left(\frac{1-\gamma_1}{\gamma_1+1} q\right) \frac{2}{(\gamma_1+1)^2}}{\left[-\exp\left(\frac{1-\gamma_1}{\gamma_1+1} \lambda''\right)\left(\frac{-2\lambda''}{(\gamma_1+1)^2}\right)\gamma_1 + \exp\left(\frac{1-\gamma_1}{\gamma_1+1} \lambda''\right)\right]/\gamma_1^2}$$

$$= \frac{-1.(-2q/4)(2/2) + 2/4}{-1[-2\lambda''/4] + 1} = \frac{1+q}{\lambda'' + 2}$$

Hence the limiting solid temperature profile for steady state is given by:

$$\bar{T}(q) = \frac{1+q}{\lambda'' + 2} \qquad .. (3.3.(23))$$

where $q \; \varepsilon \; [0, \lambda'']$.

3.4 Proposed method using parameter z:

In section 3.1 the dimensionless distance parameter q was used where $0 \leqslant q \leqslant \lambda$ since $q = \lambda z$ and we have $0 \leqslant z \leqslant 1$.

It is noticeable that for the q scale with large values of $\lambda\ (\geqslant 40)$ and large $N(\geqslant 20)$, terms such as $(q'')^{N-1}$ in $\sum_{i=1}^{N} a_i''(q'')^{N-1}$ become very large. Also the last rows of matrices [B'] and [B''] in section 3.1, may overflow thus resulting in ill-conditioning and/or error prone values for the coefficients a_i' and a_i''. To avoid this, parameter z can be used. As $0 \leqslant z \leqslant 1$ and $\sum_{i=1}^{N} a_i(z)^{i-1}$ for initial solid temperature distribution is ~~used~~, lack of precision due to near overflow is not likely to occur.

Hence, we now develop the proposed method for the length scale 0 to 1 using the parameter z:

Required to solve:

$$\frac{\partial t}{\partial z} = \lambda (T - t)$$

$$\frac{\partial T}{\partial r} = t - T \qquad .. (3.4.(1))$$

where $0 \leqslant z \leqslant 1$, $0 \leqslant r \leqslant \pi$

and $t(z = 0,\ r) = t_{in}(r)$.

Laplace transforming (3.4.(1)) along the distance coordinate, with resulting variable p, and transformed functions denoted by ^ , we have:

$$p\,\hat{t}\,(p,r) \;=\; t\,(z = 0,\; r) + \lambda(\hat{T} - \hat{t})$$

and $\dfrac{d\,\hat{T}}{d\,r} = \hat{t} - \hat{T}$

i.e., $\hat{t}\,(p,r) = \dfrac{\lambda}{p+\lambda}\,\hat{T} + \dfrac{1}{p+\lambda}\,t_{in}(r)$ .. (3.4.(2))

and so $\dfrac{d\,\hat{T}}{d\,r} + \dfrac{p}{p+\lambda}\,\hat{T} = \dfrac{1}{p+\lambda}\,t_{in}(r)$ .. (3.4.(3))

Solution of (3.4.(3)) is given by:

$$\hat{T}(p,r) = \hat{V}\,(p,r).\hat{T}\,(p,0) + \int_0^r \hat{V}\,(p,r-w)\,\frac{t_{in}(w)}{p+\lambda}\,dw$$

.. (3.4.(4))

where $\hat{V}\,(p,r) = \exp\left[\dfrac{-p}{p+\lambda}\,r\right]$ .. (3.4.(5))

Now for hot blow:

$$t(z' = 0,\; r) = t_{in}(r) = \begin{cases} 0\,, & r < 0 \\ 1\,, & r \geqslant 0 \end{cases} = \mu(r)$$

and for cold blow, $t(z'' = 0,r) = t_{in}(r) = 0.$

So for 'hot blow' (3.4.(4)) implies:

$$\hat{T}'(p,r) = \hat{V}'(p,r).\hat{T}'(p,0) + \int_0^r \frac{\hat{V}'(p,r-w)}{p+\lambda'} \mu(w)\,dw$$

Therefore,

$$\hat{T}\,(p,r) = \hat{V}'(p,r).\hat{T}'(p,0) + \hat{U}'(p,r) \qquad .. (3.4.(6))$$

where $\hat{U}'(p,r)$ is the transformed unit step response given by:

$$\hat{U}'(p,r) = \int_0^r \frac{\hat{V}'(p,w)}{p+\lambda'}\,dw = \int_0^r \frac{e^{-aw}}{p+\lambda'}\,dw;$$

where $a = \dfrac{p}{p+\lambda'}$;

$$\hat{U}'(p,r) = \frac{-1}{p+\lambda'} \left[\frac{p+\lambda'}{p} (e^{-aw})\right]_{w=0}^{r} = \frac{1}{p}[1 - e^{-ar}]$$

$$= \frac{1}{p}[1 - \exp\left(\frac{-p}{p+\lambda'} r\right)] \qquad .. (3.4.(7))$$

and for "cold blow" (3.4.(4)) gives:

$$\hat{T}''(p,r) = \hat{V}''(p,r).\hat{T}''(p,0) \qquad .. (3.4.(8))$$

Now L^{-1} (3.4.(6)) using convolution theorem and interchanging the order of terms yields for hot blow:

$$T'(z',r) = U'(z',r) + \int_0^{z'} V'(z' - v,r).T'(v,0)dv \qquad .. (3.4.(9))$$

Similarly, L^{-1} (3.4.(8)) using convolution theorem gives us for cold blow:

$$T''(z'',r) = \int_0^{z''} V''(z'' - v,r).T''(v,0)dv \qquad .. (3.4.(10))$$

Applying the reversal conditions, we obtain:

$$T'(z',r=0) = \int_0^{z''} V''(z''-v,\pi'').T''(v,0)dv \qquad .. (3.4.(11))$$

and

$$T''(z'',r=0) = U'(z',\pi') + \int_0^{z'} V'(z'-v,\pi')T'(v,0)dv \qquad .. (3.4.(12))$$

Employing

$$T'(z',0) = \sum_{i=1}^{N} a_i'(z')^{i-1}, \quad T''(z'',0) = \sum_{i=1}^{N} a_i''(z'')^{i-1},$$

we have after Laplace transforming RHS (3.4.(11)) w.r.t. distance:

$$L\left[\int_0^{z''} V''(z''-v,\pi'').T''(v,0)dv\right]$$

$$= \sum_{i=1}^{N} a_i''\ L\left[\int_0^{z''} V''(z''-v,\pi'').(v)^{i-1}\ dv\right]$$

$$= \sum_{i=1}^{N} a_i''\ \hat{V}''(p,\pi'').\ \frac{(i-1)!}{p^i}$$ using convolution theorem;

and so

$$\sum_{i=1}^{N} a_i'(z')^{i-1} = L^{-1}\left[\sum_{i=1}^{N} a_i''\ \hat{V}''(p,\pi'')\ \frac{(i-1)!}{p^i}\right] \quad ..\ (3.4.(13))$$

Similarly,

$$\sum_{i=1}^{N} a_i''(z'')^{i-1} = \sum_{i-1}^{N} a_i''(1 - z')^{i-1}$$

$$= L^{-1}\left[\hat{U}'(p,\pi') + \sum_{i=1}^{N} a_i'\ \hat{V}'(p,\pi')\ \frac{(i-1)!}{p^i}\right] \quad ..\ (3.4.(14))$$

Using N terms in each series, we require N values of z: z_j'; $j = 1,\ldots,N$ twice to obtain coefficients a_i', a_i'' ; $i = 1,\ldots,N$.

So (3.4.(13)) and (3.4.(14)) imply:

$$\sum_{i=1}^{N} a_i'(z_j')^{i-1} = L^{-1}\left[\sum_{i=1}^{N} a_i''\,\hat{V}''(p,\pi'')\,\frac{(i-1)!}{p^i}\right]_{z=1-z_j'}$$

and

$$\sum_{i=1}^{N} a_i''(1-z_j')^{i-1} =$$

$$= L^{-1}\left[\hat{U}'(p,\pi') + \sum_{i=1}^{N} a_i'\hat{V}'(p,\pi')\,\frac{(i-1)!}{p^i}\right]_{z=z_j'}$$

Writing in matrix notation, we have:

$$[B']\,\underline{a}' = [C'']\,\underline{a}'' \qquad \text{.. (3.4.(15))}$$

and

$$[B'']\,\underline{a}'' = \underline{U} + [C']\,\underline{a}' \qquad \text{.. (3.4.(16))}$$

where

$$\underline{a}' = \begin{bmatrix} a_1' \\ a_2' \\ \cdot \\ \cdot \\ a_N' \end{bmatrix}, \quad \underline{a}'' = \begin{bmatrix} a_1'' \\ a_2'' \\ \cdot \\ \cdot \\ a_N'' \end{bmatrix}, \quad [B'] = \begin{bmatrix} 1 & z_1' & (z_1')^2 & \ldots & (z_1')^{N-1} \\ 1 & z_2' & (z_2')^2 & \ldots & (z_2')^{N-1} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 1 & z_N' & (z_N')^2 & \ldots & (z_N')^{N-1} \end{bmatrix},$$

$$[B''] = \begin{bmatrix} 1 & 1-z_1' & (1-z_1')^2 & .. & (1-z_1')^{N-1} \\ 1 & 1-z_2' & (1-z_2')^2 & .. & (1-z_2')^{N-1} \\ . & . & . & & . \\ . & . & . & & . \\ 1 & 1-z_N' & (1-z_N')^2 & .. & (1-z_N')^{N-1} \end{bmatrix},$$

$[C''] = (c''_{ij})$, $[C'] = (c'_{ij})$ and $\underline{U} = (U_i)$

where

$$c''_{ij} = L^{-1} [\frac{\hat{V}''(p,\pi'')}{p^j}]]_{z=1-z_i'} (j-1)! \quad ,$$

$$c'_{ij} = L^{-1} [\frac{\hat{V}'(p,\pi')}{p^j}]]_{z=z_i'} (j-1)!$$

and $$U_i = L^{-1} [\frac{1}{p} [1 - \exp(\frac{-p}{p+\lambda'} \pi')]]]_{z=z_i'}$$

From initial value theorem of Laplace transforms:

$$L^{-1} [\frac{1}{p} [1 - \exp(\frac{-p\pi'}{p+\lambda'})]]]_{z=0} = \lim_{p \to \infty} [1 - \exp(\frac{-p\pi'}{p+\lambda'})]$$

$$= 1 - \exp(-\pi') \quad , \quad \text{and}$$

$$L^{-1} [\frac{\hat{V}(p,\pi)}{p^j}]]_{z=0} = \lim_{p \to \infty} \frac{\hat{V}(p,\pi)}{p^{j-1}} \quad ; \quad j = 1,\ldots,N$$

$$= \lim_{p \to \infty} [\exp (\frac{-p\pi}{p+})]/p^{j-1} = \begin{cases} e^{-\pi} & ; \quad j = 1 \\ 0 & ; \quad j = 2,\ldots,N \end{cases}$$

As in section 3.1, the solution of (3.4.(15)) and (3.4.(16)) is obtained as:

$$\underline{a}' = [[B'] - [C''][B'']^{-1}[C']]^{-1}[C''][B'']^{-1}\underline{U} \quad .. (3.4.(17))$$

$$\underline{a}'' = [B'']^{-1}\underline{U} + [B'']^{-1}[C']\underline{a}' \quad .. (3.4.(18))$$

Having found coefficients a_i' and a_i'' it is an easy step to obtain solid temperatures at the beginning of the hot and cold blow periods respectively from:

$$T'(z',0) = \sum_{i=1}^{N} a_i'(z')^{i-1} \quad .. (3.4.(19))$$

and

$$T''(z'',0) = \sum_{i=1}^{N} a_i''(z'')^{i-1} \quad .. (3.4.(20))$$

To calculate the thermal efficiency we make use of

$$E = \frac{\lambda'}{\pi'} [\frac{1}{\lambda''} \int_0^{\lambda''} T''(q'',0)dq'' - \frac{1}{\lambda'} \int_0^{\lambda'} T'(q',0)dq']$$

which is equation (3.2.(1)) in a general case.

Now as $dz'' = dq''/\lambda''$, $dz' = dq'/\lambda'$, we have:

$$E = \frac{\lambda'}{\pi'} [\int_0^1 T''(z'',0)dz'' - \int_0^1 T'(z',0)dz']$$

$$E = \frac{\lambda'}{\pi'}\left[\sum_{i=1}^{N} a''_i \int_0^1 (z'')^{i-1}\, dz'' - \sum_{i=1}^{N} a'_i \int_0^1 (z')^{i-1} dz'\right]$$

$$= \frac{\lambda'}{\pi'}\left[\sum_{i=1}^{N} \frac{a''_i}{i} - \sum_{i=1}^{N} \frac{a'_i}{i}\right] \qquad \text{.. (3.4.(21))}$$

3.5 Derivation of exit fluid temperatures using z parameter:

It is evident from the section 3.4 that the solid temperature distributions yield the thermal efficiency of the the system at cyclic equilibrium. The exit fluid temperatures are also required to be evaluated as they are useful for two main purposes:

a) To allow prediction of the real period of cyclic equilibrium operation for variable flow from the constant flow periods π' and π''. This is dealt with in the next chapter.

b) To allow the maximum exit temperature during the hot blow and the minimum exit temperature during the cold blow. The first allows a designer to determine whether maximum allowable fluid temperature will be exceeded and the second determines the blast temperature attainable for a given set λ', λ'', π' and π'' of constant flow parameters.

(3.5.(6)) has a solution:

$$\hat{t}\,(z,s) = \hat{V}_f(z,s).\hat{t}\,(z=0,s) + \int_0^z \hat{V}_f(z-w,s)\,\frac{\lambda}{s+1}T(w,0)dw \qquad ..\ (3.5.(7))$$

where $\hat{V}_f(z,s) \;=\; \exp\left(\frac{-\lambda s}{s+1}\,z\right)$ .. (3.5.(8))

Hence, we need to invert the following time Laplace transformed equations:

For hot blow:-

$$\hat{t}'(z',s) = \hat{V}'_f(z',s).\hat{t}'(z'=0,s) + \frac{\lambda'}{s+1}\int_0^{z'} \hat{V}'_f(z'-w,s).\;.\;T'(w,0)dw \qquad ..\ (3.5.(9))$$

where

$$\hat{t}'(z'=0,s) = L\,[t'_{in}(r)] \;=\; \frac{1}{s}\ ,$$

$$T'(w,0) \;=\; \sum_{i=1}^{N} a'_i\, w^{i-1} \quad \text{and} \quad \hat{V}'_f(z',s) = \exp\left(\frac{-\lambda' s}{s+1}\,z'\right)\ ;$$

and for cold blow:-

$$\hat{t}''(z''\,,s) = \frac{\lambda''}{s+1}\int_0^{z''} \hat{V}''_f(z''-w,s).T''(w,0)dw \qquad ..\ (3.5.(10))$$

where

$$T''(w,0) = \sum_{i=1}^{N} a''_i\,(w)^{i-1},\quad \hat{V}''_f(z'',s) = \exp\left(\frac{-\lambda'' s}{s+1}\,z''\right)$$

since $t''_{in}\,(r) \;=\; 0.$

So, (3.5.(9)) implies:

$$t'(z',r') = L^{-1}\left[\exp(-\frac{\lambda' s}{s+1} z') / s \right]_{r=r'} +$$

$$+ L^{-1}\left[\frac{\lambda'}{s+1} \int_0^{z'} \exp(\frac{\lambda' s}{s+1}(w-z')) \sum_{i=1}^{N} a_i' w^{i-1} dw \right]_{r=r'}$$

.. (3.5.(11))

Consider

$$L^{-1}\left[\frac{\lambda'}{s+1} \int_0^{z'} \exp(\frac{\lambda' s}{s+1}(w-z')) \sum_{i=1}^{N} a_i' w^{i-1} dw \right]_{r=r'}$$

$$= L^{-1}\left[\sum_{i=1}^{N} a_i' \left(\frac{\lambda'}{s+1} \right) \int_0^{z'} \exp(\frac{\lambda' s}{s+1}(w-z')) w^{i-1} dw \right]_{r=r'}$$

.. (3.5.(12))

Now let

$$I = \int_0^{z'} \exp(\frac{\lambda' s}{s+1}(w - z')) w^{i-1} dw$$

$$= e^{-az'} \int_0^{z'} e^{aw} w^{i-1} dw \quad \text{where} \quad a = \frac{\lambda' s}{s+1}$$

$$= e^{-az'} I_{i-1} \quad \text{(say)}$$

we have

$$I_n = \int_0^{z'} e^{aw} w^n dw = \frac{1}{a}[e^{az'} (z')^n - nI_{n-1}] \quad ; n = 1,2,\ldots,N-1$$

.. (3.5.(13))

from using integration by parts.

and

$$I_o = \int_0^{z'} e^{aw}\,dw = \frac{1}{a}[e^{az'} - 1]$$

From (3.5.(13)), therefore,

$$I_1 = \frac{1}{a}[e^{az'}z' - I_o] = \frac{1}{a}[e^{az'}z' - \frac{1}{a}e^{az'} + \frac{1}{a}]$$

$$I_2 = \frac{1}{a}[e^{az'}(z')^2 - \frac{2}{a}[e^{az'}z' - \frac{1}{a}e^{az'} + \frac{1}{a}]]$$

$$= \frac{1}{a}[e^{az'}(z')^2 - \frac{2}{a}e^{az'}z' + \frac{2!}{a^2}e^{az'} - \frac{2!}{a^2}]$$

$$I_3 = \frac{1}{a}[e^{az'}(z')^3 - \frac{3}{a}e^{az'}(z')^2 + \frac{3!}{a^2}e^{az'}z' - \frac{3!}{a^3}e^{az'} + \frac{3!}{a^3}]$$

$$\vdots$$

$$I_k = \frac{1}{a}[e^{az'}(z')^k - \frac{k}{a}e^{az'}(z')^{k-1} + \frac{k!\,e^{az'}(z')^{k-2}}{(k-2)!\,a^2} - \frac{k!}{(k-3)!\,a^3}e^{az'}(z')^{k-3} + \ldots + (-)^{k+1}\frac{k!}{a^k}]$$

$$= \frac{1}{a}[e^{az'}(z')^k\left(1 + k!\sum_{j=1}^{k}\frac{(-)^j}{(k-j)!(z'a)^j}\right) + (-)^{k+1}\frac{k!}{a^k}] \quad .. \ (3.5.(14))$$

$$; \ k = 1,2,\ldots,N-1$$

So from (3.5.(12)):

$$L^{-1}\Big[\sum_{i=1}^{N} a_i'\left(\frac{\lambda'}{s+1}\right)\int_{o}^{z'}\exp\left(\frac{\lambda' s}{s+1}(w-z')\right)w^{i-1}\,dw\Big]_{r=r'}$$

$$= L^{-1}\Big[\sum_{i=1}^{N} a_i'\left(\frac{\lambda'}{s+1}\right)e^{-az'}\,I_{i-1}\Big]_{r=r'},\quad a = \frac{\lambda' s}{s+1}$$

$$= L^{-1}\Big[\sum_{i=2}^{N} a_i'\left(\frac{\lambda'}{s+1}\right)e^{-az'}\,\frac{1}{a}\Big\{e^{az'}(z')^{i-1}\,.$$

$$[1+(i-1)!\sum_{j=1}^{i-1}\frac{(-)^j}{(i-1-j)!(z'a)^j}]\;+$$

$$(-)^i\,\frac{(i-1)!}{a^{i-1}}\Big] + a_1'\left(\frac{\lambda'}{s+1}\right)e^{-az'}\,\frac{1}{a}\,(e^{az'}-1)\Big]_{r=r'}$$

$$= L^{-1}\Big[\frac{a_1'}{s}\left(1-\exp\left(-\frac{\lambda' s}{s+1}z'\right)\right)+$$

$$\sum_{i=2}^{N}\Big\{\frac{a_i'}{s}\;(z')^{i-1}\,[1+(i-1)!\sum_{j=1}^{i-1}\frac{(-)^j}{(i-1-j)!}\left(\frac{s+1}{\lambda' s z'}\right)^j +$$

$$(-)^i(i-1)!\left(\frac{s+1}{\lambda' s}\right)^{i-1}\exp\left(-\frac{\lambda' s z'}{s+1}\right)]\Big\}\Big]_{r=r'}$$

So from (3.5.(11)) we obtain:

$$t'(1,r') = L^{-1}\left\{ \frac{e^{-\frac{\lambda' s}{s+1}}}{s} + \frac{a_1'}{s}\left(1 - e^{-\frac{\lambda' s}{s+1}}\right) + \sum_{i=2}^{N} \frac{a_i'}{s}\left[1 + (i-1)! \sum_{j=1}^{i-1} \frac{(-)^j}{(i-1-j)!} \left(\frac{s+1}{\lambda' s} \right)^j + (-)^i (i-1)! \left(\frac{s+1}{\lambda' s} \right)^{i-1} e^{-\frac{\lambda' s}{s+1}} \right] \right\}_{r=r'} \quad .. (3.5.(15))$$

Using initial value theorem:

$$t'(1,0) = \lim_{s \to \infty} \left[e^{-\frac{\lambda' s}{s+1}} + a_1'\left(1 - e^{-\frac{\lambda' s}{s+1}} \right) + \sum_{i=2}^{N} a_i' \left[1 + (i-1)! \sum_{j=1}^{i-1} \frac{(-)^j}{(i-1-j)!} \left(\frac{s+1}{\lambda' s} \right)^j + (-)^i (i-1)! \left(\frac{s+1}{\lambda' s} \right)^{i-1} e^{-\frac{\lambda' s}{s+1}} \right] \right]$$

$$t'(1,0) = e^{-\lambda'} + a_1' (1 - e^{-\lambda'}) + \sum_{i=2}^{N} a_i' [1 +$$

$$(i-1)! \sum_{j=1}^{i-1} \frac{(-1/\lambda')^j}{(i-1-j)!} - (-1/\lambda')^{i-1} (i-1)!\ e^{-\lambda'}]$$

.. (3.5.(16))

Equation (3.5.(15)) reduces to:

$$t'(1,r') = a_1' + (1 - a_1')\ L^{-1} \left[\frac{e}{s}^{-\frac{\lambda' s}{s+1}} \right]_{r=r'} +$$

$$\sum_{i=2}^{N} a_i' [1 + (i-1)! \sum_{j=1}^{i-1} \frac{(-)^j}{(i-1-j)!} L^{-1} \left[\frac{1}{s} \left(\frac{s+1}{\lambda' s}\right)^j \right]_{r=r'}$$

$$+ (-)^i (i-1)!\ L^{-1} \left[\left(\frac{s+1}{\lambda' s}\right)^{i-1} \frac{e}{s}^{-\frac{\lambda' s}{s+1}} \right]_{r=r'}]$$

Now

$$\frac{1}{s} \left(\frac{s+1}{\lambda' s} \right)^j = \left(\frac{1}{\lambda'} \right)^j \frac{1}{s} \left(1 + \frac{1}{s}\right)^j$$

$$= \left(\frac{1}{\lambda'} \right)^j \frac{1}{s} \sum_{k=0}^{j} \binom{j}{k} \left(\frac{1}{s} \right)^k$$

using Binomial Theorem

Therefore,

$$L^{-1}\left[\frac{1}{s}\left(\frac{s+1}{\lambda' s}\right)^{j}\right]_{r=r'} = \left(\frac{1}{\lambda'}\right)^{j} L^{-1}\left[\frac{1}{s}\sum_{k=0}^{j}\binom{j}{k}\left(\frac{1}{s}\right)^{k}\right]_{r=r'}$$

$$= \left(\frac{1}{\lambda'}\right)^{j} L^{-1}\left[\sum_{k=0}^{j}\binom{j}{k}\left(\frac{1}{s}\right)^{k+1}\right]_{r=r'}$$

$$= \left(\frac{1}{\lambda'}\right)^{j}\sum_{k=0}^{j}\binom{j}{k} L^{-1}\left[\left(\frac{1}{s}\right)^{k+1}\right]_{r=r'}$$

$$= \left(\frac{1}{\lambda'}\right)^{j}\sum_{k=0}^{j}\binom{j}{k}\frac{(r')^{k}}{k!}$$

Hence,

$$t'(1,r') = a_1' + (1-a_1')\, L^{-1}\left[\frac{e^{-\frac{\lambda' s}{s+1}}}{s}\right]_{r=r'} +$$

$$\sum_{i=2}^{N} a_i' \left[1 + (i-1)! \sum_{j=1}^{i-1} \frac{(-1/\lambda')^{j}}{(i-1-j)!} \;.\right.$$

$$\sum_{k=0}^{j}\binom{j}{k}\frac{(r')^{k}}{k!} - (-1/\lambda')^{i-1}(i-1)! \;.$$

$$\left. L^{-1}\left[\frac{e^{-\frac{\lambda' s}{s+1}}}{s}\left(\frac{s+1}{s}\right)^{i-1}\right]_{r=r'}\right] \quad .. \ (3.5.(17))$$

Similarly for cold blow (3.5.(10)) yields:

$$t''(z'',r'') = L^{-1}\left[\sum_{i=1}^{N} a_i'' \left(\frac{\lambda''}{s+1} \right) \int_0^{z''} \exp\left(\frac{\lambda'' s}{s+1} (w - z'')\right) . w^{i-1}\, dw \right]_{r=r'} \quad .. (3.5.(18))$$

From which with similar analysis as in the case of hot blow, we obtain:

$$t''(1,0) = a_1'' - a_1'' e^{-\lambda''} + \sum_{i=2}^{N} a_i'' \left[1 + (i-1)! \sum_{j=1}^{i-1} \frac{(-1/\lambda'')^j}{(i-1-j)!} - (-1/\lambda'')^{i-1}(i-1)!\, e^{-\lambda''} \right] \quad .. (3.5.(19))$$

and

$$t''(1,r'') = a_1'' - a_1'' L^{-1}\left[\frac{e^{-\frac{\lambda'' s}{s+1}}}{s} \right]_{r=r'} + \sum_{i=2}^{N} a_i'' \left[1 + (i-1)! \sum_{j=1}^{i-1} \frac{(-1/\lambda'')^j}{(i-1-j)!} . \sum_{k=0}^{j} \binom{j}{k} \frac{(r'')^k}{k!} - (-1/\lambda'')^{i-1}(i-1)! . L^{-1}\left[\frac{e^{-\frac{\lambda'' s}{s+1}}}{s} \left(\frac{s+1}{s} \right)^{i-1} \right]_{r=r'} \right] \quad .. (3.5.(20))$$

3.6 Proposed method for a Symmetric Balanced Regenerator:

As noted earlier Nahavandi and Weinstein [17] consider the regenerator to be symmetric and balanced. Whereas the proposed method outlined in previous sections treats the general problem thus increasing the complexity of the algorithm. We now develop the case for a symmetric, balanced regenerator and thus show a comparatively simpler algorithm as opposed to the general case which has already been discussed.

We start off with equations (3.1.(13)) and (3.1.(14)), as upto this point the analysis is identical; so:

$$T'(q',r'=0) = \int_0^{q''} V''(q''-v,\pi'').T''(v,0)\,dv \quad .. (3.6.(1))$$

and

$$T''(q'',r''=0) = U'(q',\pi') + \int_0^{q'} V'(q'-v,\pi').T'(v,0)\,dv \quad .. (3.6.(2))$$

Now consider the symmetric, balanced case where

$$\lambda = \lambda' = \lambda'', \quad \pi = \pi' = \pi'' \ :-$$

Let $T''(q'',0) = F''(q'')$; $T'(q',0) = F'(q')$.

So (3.6.(1)) and Reversal Conditions **imply:**

$$F'(\lambda - q'') = 1 - F''(\lambda - q'') = \int_0^{q''} V''(q''-v,\pi).F''(v)\,dv \quad .. (3.6.(3))$$

and (3.6.(2)) with R.C. yield:

$$1 - F''(\lambda - q') = F'(\lambda - q')$$

$$= U'(q',\pi) + \int_0^{q'} V'(q' - v,\pi)\, F'(v)\, dv \quad .. \ (3.6.(4))$$

Therefore, from (3.6.(3)) we have:

$$1 - \sum_{j=1}^{N} a_j'' \, (\lambda - q'')^{j-1} = \sum_{j=1}^{N} a_j'' \int_0^{q''} V''(q''-v,\pi) v^{j-1}\, dv \quad .. \ (3.6.(5))$$

and (3.6.(4)) implies:

$$\sum_{j=1}^{N} a_j'(\lambda - q')^{j-1} = U'(q',\pi) + \sum_{j=1}^{N} a_j' \int_0^{q'} V'(q'-v,\pi) v^{j-1}\, dv \quad .. \ (3.6.(6))$$

Hence

$$1 = \sum_{j=1}^{N} a_j'' \left[\int_0^{q''} V''(q''-v,\pi) v^{j-1}\, dv + (\lambda - q'')^{j-1} \right]$$

and

$$U'(q',\pi) = \sum_{j=1}^{N} a_j' \left[(\lambda - q')^{j-1} - \int_0^{q'} V'(q'-v,\pi) v^{j-1}\, dv \right]$$

These can be rewritten in matrix form as:

$$\underline{1} = [P''] \underline{A}'' \qquad .. (3.6.(7))$$

and $$\underline{U}' = [P'] \underline{A}' \qquad .. (3.6.(8))$$

where

$$\underline{1} = [1,1,\ldots,1]^T, \ \underline{A}' = [a_1', a_2', \ldots, a_N']^T, \ \underline{A}'' = [a_1'', \ldots, a_N'']^T,$$

$$[P''] = (p_{ij}''), \ [P'] = (p_{ij}') \text{ and } \underline{U}' = (U_i') \ ;$$

$$p_{ij}'' = L^{-1} \left[\frac{\exp(-p\pi/(p+1))}{p^j} \right]_{q=q_i''} (j-1)! + (\lambda - q_i'')^{j-1} ,$$

$$p_{ij}' = -L^{-1} \left[\frac{\exp(-p\pi/(p+1))}{p^j} \right]_{q=q_i'} (j-1)! + (\lambda - q_i')^{j-1} ,$$

$$U_i' = L^{-1} \left[\frac{1}{p} (1 - \exp(-p\pi/(p+1))) \right]_{q=q_i'}$$

$$q_i' = (i-1)\Delta q; \quad q_i'' = (N-i)\Delta q \quad \text{for} \quad i = 1,\ldots,N$$

$$\text{and } \Delta q = \lambda/(N-1).$$

Note, as before use of Initial Value Theorem yields:

$$L^{-1} \left[\frac{\exp(-p\pi/(p+1))}{p^j} \right]_{q=0} = \begin{cases} e^{-\pi} & ; \ j = 1 \\ 0 & ; \ j = 2,\ldots,N \end{cases}$$

and

$$L^{-1}\left[\frac{1}{p}[1-\exp(-p\pi/(p+1))]\right]_{q=0} = 1-\exp(-\pi)\ .$$

Hence, (3.6.(7)) can be solved for coefficients a_i', a_i'' from:

$$\underline{A}'' = [P'']^{-1}\,\underline{1} \qquad ..\ (3.6.(9))$$

and $$\underline{A}' = [P']^{-1}\,\underline{U}' \qquad ..\ (3.6.(10))$$

From these, temperature profiles can easily be obtained. It is noticeable that equations (3.6.(9)) and (3.6.(10)) are fairly simple and straight forward to implement on a digital computer when compared with other closed methods reviewed earlier.

3.7. Review of Zakian's inverse Laplace transform method:

From sections 3.1, 3.4, 3.5 and 3.6 it is quite evident that an accurate method for inversion of Laplace transforms is required. The following equations describe the relationships between a function and its Laplace and Inverse Laplace Transforms:-

Laplace transform:

$$F(s) = L(f(\tau)) = \int_0^{\infty} f(\tau)\,e^{-s\tau}\,d\tau \qquad ..\ (3.7.(1))$$

Inverse Laplace transform (Bromwich integral):

$$f(\tau) = L^{-1}(F(s)) = \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} F(s)\, e^{s\tau}\, d\tau \quad ..\ (3.7.(2))$$

where c is such that $Re(s) \geqslant c$.

Jeffreson and Chow [11,12] report that although there exist a large number of different methods and approximation formulae to (3.7.(2)), the numerical approximation of the integration in (3.7.(2)) using Gaussian's quadrature coefficients as used by Zakian [27] seems quite suitable.

Zakian [27] method requires that displaced impulse or delta function $\delta(\tau-1)$ be approximated by a finite sum of exponentials either by least squares approximation or by a Pade' approximation. For a delta function, we have:

$$f(\tau) = \int_{0}^{\infty} f(\theta\tau).\delta(\theta - 1)\, d\theta \qquad ..\ (3.7.(3))$$

Approximating $\delta(\tau-1)$ by a weighted sum of exponential functions, we can write:

$$\delta(\tau-1) \cong \delta_N(\tau-1) = \sum_{i=1}^{N} K_i \exp(-\alpha_i\tau) \qquad ..\ (3.7.(4))$$

Substituting (3.7.(4)) into (3.7.(3)) and interchanging the order of summation and integration gives:

$$f(\tau) \cong f_N(\tau) = \sum_{i=1}^{N} K_i \int_0^{\infty} f(\theta\tau) \exp(-\alpha_i\theta)\, d\theta$$

or

$$f_N(\tau) = \sum_{i=1}^{N} \frac{K_i}{\tau} F\left(\frac{\alpha_i}{\tau}\right) \qquad .. (3.7.(5))$$

where as defined in (3.7.(1)), $F(s)$ is the Laplace transform of $f(\tau)$; $0 < \tau < \infty$.

Equation (3.7.(5)) is straightforward and simple to use, besides coefficients K_i, α_i, once determined for a particular value of N, are independent of the function $F(s)$ being inverted. So for a given N, once an optimal set of K_i, α_i; $i = 1,2,\ldots,N$ are found, a large range of functions can be inverted by using equation (3.7.(5)).

Jeffreson and Chow [11,12] have compared a number of sets of the coefficients α_i, K_i and concluded that, for a given N, the Gaussian coefficients in general give good approximations in (3.7.(5)).

Zakian and Edwards [28,30] have obtained sets of "Pade'" coefficients by approximating the rational fraction:

$$\emptyset_N(z) = L(\delta_N(\tau-1)) = \sum_{i=1}^{N} \frac{K_i}{z+\alpha_i} \qquad .. (3.7.(6))$$

to the Taylor series expansion of the exponential function $\exp(-z)$.

Zakian and Gannon [29] earlier derived a set of quasi-least-squares coefficients (K_i) for a chosen set of (α_i) which were used by Jeffreson and Chow (JC) [12] in a comparison with Pade' coefficients. JC [12] report that these quasi-least-square coefficients [29] are generally unsuitable for (3.7.(5)) and can produce poor approximations.

JC [12] have produced a set of what they term "true" least-squares (LS) coefficients (α_i, K_i). These along with Pade' coefficients were tested to invert five test functions for $N = 15$ in order to compare the two sets of coefficients by them. They report that although the delta function can be more accurately approximated using least-squares approximants than Pade' approximants, their corresponding coefficients, when used in Zakian's Laplace inversion formula (3.7.(5)) do not necessarily produce more accurate approximations to $f(\tau)$. In fact, fairly accurate results can be obtained using the Pade' coefficients as Integral Square Errors (ISE) defined by:

$$\text{ISE} = \sum_{\theta=0}^{\theta=\tau} \Delta\theta \, [f_N(\theta) - f(\theta)]^2 \qquad .. \; (3.7.(7))$$

for $N = 15$ is found to be in the range of 10^{-3} to 10^{-24} by JC [12] for the test functions used. They also found that ISE using least-squares coefficients was greater for each function tested when compared with Pade' coefficients.

Hence Pade' coefficients for N=15 as listed by JC [12] were used in conjunction with (3.7.(5)) in order to obtain the inverse of Laplace transform functions occurring in sections 3.1 and 3.4. They can be found in subroutines ZACOFS in program CLOZAK listed in the appendices.

Zakian's method using Pade' coefficients was particularly chosen because of its accuracy, reliability, simplicity and ease in programming. The method is computationally fast, although it requires at least [N/2] (= integer part of N/2) evaluations of F(s) at every value of τ as reported by JC [12].

Drawbacks of Zakian's method seem to be:

a) It does not provide a ready estimate of the error of an approximation of an unknown function.

b) It requires a number of evaluations of F(s) at each value of τ, so may take up a lot of computer time especially if F(s) happens to be a complicated function.

c) The formula (3.7.(5)) cannot be used to compute $f_N(\tau)$ at $\tau = 0$. This is why one has to resort to the initial value theorem in order to evaluate $f_N(\tau = 0)$.

d) The need for a long computer word length restricts implementation on all machines.

3.8 Chapter Summary:

This chapter describes a new method for solving the counterflow regenerator problem. At first reasons for developing this new method are outlined and it is noted that although this method is based on Nahavandi and Weinstein method [17], it avoids the use of quadrature formulae (used to obtain complicated forms of Bessel functions) and utilizes numerical inversion of Laplace transform. We also note that the proposed method considers a general non-symmetric, unbalanced regenerator as compared to NW method [17] which only took the balanced, symmetric regenerator into account.

The proposed analytical closed method is developed in section 3.1 and we note that equations (3.1.(24)) and (3.1.(25)) can be utilized to obtain solid temperatures in terms of coefficients $a_i^{'}$ and $a_i^{''}$ respectively.

Analysis in this section beyond equation (3.1.(25)) derives equations (3.1.(32)) and (3.1.(33)) which determine fluid temperatures in terms of coefficients $a_i^{'}$ and $a_i^{''}$. We note that these equations are very easy to program on a digital computer as they are simple sums once coefficients a are known.

In section 3.2, thermal efficiency has been derived in terms of coefficients a and other parameters and once again we note that this is very easy to program.

Section 3.3 derives steady state limiting temperature profile, this is required as the limiting case of period becoming zero cannot be handled through the Laplace inversion method used. Thus a separate derivation for the limiting case solid temperature profiles in steady state determines equation (3.3.(23)).

Having developed the proposed method for q-scale it was noticed that for large values of $\lambda (\geqslant 40)$ and large values of $N (\geqslant 20)$, terms in the initial solid temperature profile such as $(q'')^{N-1}$ in $\sum_{i=1}^{N} a_i'' (q'')^{N-1}$ become very large which may cause overflow in rows of $[B']$ and $[B'']$ matrices in section 3.1, thus resulting in ill-conditioning and/or error prone results. Therefore, normalized parameter z, $z\epsilon[0,1]$ was used in $\sum_{i=1}^{N} a_i(z)^{i-1}$ for initial solid temperature distribution in order to avoid overflow and lack of precision. This meant that the proposed method was to be rederived using the z-parameter and this is what is done in section 3.4, and solid temperatures are obtained through equations (3.4.(19)) and (3.4.(20)). The thermal efficiency is also derived once again (see equation (3.4.(21)). It is once again noticed that final equations obtained in these sections are very easy to program on a digital computer.

Section 3.5 develops exit fluid temperature equations (3.5.(19)) to (3.5.(20)) using the z-parameter.

In section 3.6 the case of a symmetric, balanced regenerator is considered and equations (3.6.(9)), (3.6.(10)) for the coefficients of proposed method are derived. It is noted that these equations are extremely simple when compared with NW method [17] which was developed only for a symmetric, balanced case.

This is followed by section 3.7 which is a review of the inverse Laplace transform method used, the reasons why this method was used along with its advantages and disadvantages are also presented.

...

CHAPTER 4: DETAILS ABOUT COMPUTER PROGRAMS, RESULTS AND COMPARISONS

4.1. Description of Computer Programs:

Based on sections 3.1, 3.2, 3.4 and 3.5 computer programs have been written in FORTRAN for the CDC 6000 Computer System housed at the University of Adelaide. A listing of these programs can be found in the appendices.

Appendix A1 lists the FORTRAN program which obtains coefficients a' and a" as expressed in equations (3.1.(22)) and (3.1.(23)) respectively for a specific set of parameters N (number of terms in the finite series approximation of initial solid temperature profile), π', π'' (hot and cold periods in normalized dimensionless time units) and λ', λ'' (hot and cold blow reduced dimensionless bed lengths). The coefficients a' and a" are output. Note that each are N in number . Following this, on demand, temperature profiles for solid are obtained through equations (3.1.(24)) and (3.1.(25)) for hot and cold blow. This is followed by calculation and printing of thermal efficiency E as defined in equation (3.2.(1)), if required. The program is flexible enough to allow a consecutive number of runs with different parameters just by entering the new set each time as prompted by the program. Alternatively results can be obtained from the program for different orders of polynomial approximation of initial solid temperature profiles (i.e., for different values of N) while keeping the other parameters unchanged. The program uses Zakian's coefficients for Pade' approximation to obtain inverse Laplace transforms as required in section 3.1 to determine matrices [C'], [C''] and vector $\underline{U}$. All the arithmetic is in single precision and inversion of matrices is done by using the Control Data Library MATRIX routine. This program uses (as do sections 3.1 and 3.2) the dimensionless distance parameter $q \varepsilon [0, \lambda]$.

The FORTRAN program listed in appendix A2 evaluates coefficients a' and a'', but this time double precision arithmetic is used and distance parameter $z \varepsilon [0,1]$ scale is utilized instead of q scale which was used in appendix A1. In this program, after initial set of parameters: N, λ', λ'', π' and π'' have been entered, evaluation of a' and a'' is undertaken and these are obtained via equations (3.4.(17)) and (3.4.(18)). In these calculations the determinants of the matrices inverted via Library MATRIX routine are output. After calculation of each set of coefficients a' or a'' the valuesfor a' and/or a'' are printed on demand. After this, if required, when number of bed mesh points has been input solid temperatures are calculated and output on demand via equations (3.4.(19)) and (3.4.(20)). Then, if exit fluid temperature profiles are required they are obtained by subroutine FLUTE which makes use of equations (3.5.(16)), (3.5.(17)), (3.5.(19)) and (3.5.(20)). Within the subroutine FLUTE there is provision to obtain variable time integral of cold blow exit fluid temperature. If required, then this value is obtained within FLUTE by evaluating a function ETASTAR which makes use of the derivations which follow in the next section 4.2. As in the case of appendix A1, here also flexibility has been built into the program so that a number of consecutive runs can be made with changed parameter(s) without having to execute the program separately each time.

Appendix A3 comprises of the FORTRAN program which forms the single precision counterpart of the program in A2. In fact this program was written before the one in A2. However, it was felt that for greater accuracy double precision must be used and so changes were made into A3 to incorporate this feature which resulted in A2. The sequence of computations described for A2 therefore holds good for the program in Appendix A3 as well.

4.2. Derivation of time scale transformation for variable flow:

In chapter 1, the need for variable time integral of cold blow exit fluid temperature was stressed. We derive this now. Starting with equation (3.5.(18)), we let:

$$I'' = \int_0^{z''} \exp\left(\frac{\lambda'' s}{s+1}(w-z'')\right) w^{i-1}\, dw$$

$$= e^{-bz''} \int_0^{z''} e^{bw}\, w^{i-1}\, dw \quad \text{where} \quad b = \lambda'' s/(s+1)$$

$$= e^{-bz''}\, I''_{i-1} \quad \text{say.}$$

So, from similar analysis as for hot blow in section 3.5, we obtain:

$$I_o'' = \frac{1}{b}[e^{bz''} - 1] \qquad \text{.. (4.2.(1))}$$

$$\text{and} \quad I''_k = \frac{1}{b}[e^{bz''}(z'')^k \left(1+k! \sum_{j=1}^{k} \frac{(-)^j}{(k-j)!(z''b)^j}\right) + (-)^{k+1} \frac{k!}{b^k}] \;;\quad k = 1,2,\ldots,N-1 \qquad \text{.. (4.2.(2))}$$

Therefore, (3.5.(18)) implies:

$$t''(z'',r'') = L^{-1}\left[\sum_{i=1}^{N} a''_i \left(\frac{\lambda''}{s+1}\right) e^{-bz''}\, I''_{i-1}\right]_{r=r''}$$

So from (4.2.(1)) and (4.2.(2)) and substituting the value of b we have:

$$t''(z'',r'') = L^{-1}\,[\,\frac{a_1''}{s}(1 - e^{-\frac{\lambda'' s z''}{s+1}}) +$$

$$\sum_{i=2}^{N} (a_i''/s)[\,(z'')^{i-1}\,(1 + (i-1)!\sum_{j=1}^{i-1}\frac{(-)^j}{(i-1-j)!}\,(\frac{s+1}{\lambda'' s z''})^j) +$$

$$(-)^i(i-1)!\,(\frac{s+1}{\lambda'' s})^{i-1}\,e^{-\lambda'' s z''/(s+1)}\,]]_{r=r''} \qquad ..\,(4.2.(3))$$

But variable time integral of cold blow exit fluid temperature η^* is defined as

$$\eta^* = \int_0^{r''} t''\,(z'' = 1,r)\,dr \qquad ..\,(4.2.(4))$$

obviously $\eta^* = 0$ when $r'' = 0$.

So, for $r'' \varepsilon\ (0,\pi'']$, from (4.2.(3)) and (4.2.(4)) we have:

$$\eta^* = \int_0^{r''} [a_1'' - a_1''\,L^{-1}\,(\frac{e}{s}^{-\lambda'' s/(s+1)})_{r=r''} +$$

$$+ \sum_{i=2}^{N} (-)^{i} a_i'' (i-1)! \, L^{-1} \Big[\Big(\frac{s+1}{\lambda'' s} \Big)^{i-1} \frac{e^{-\lambda'' s/(s+1)}}{s} \Big]_{r=r''} +$$

$$\sum_{i=2}^{N} a_i'' \Big[1 + (i-1)! \sum_{j=1}^{i-1} \frac{(-)^{j}}{(i-1-j)!} L^{-1} \Big[\frac{1}{s} \Big(\frac{s+1}{\lambda'' s} \Big)^{j} \Big]_{r=r''} \Big]\Big] \, dr$$

$$\text{i.e.,}\ \eta^* = a_1'' r'' - a_1'' L^{-1} \Big(\frac{e^{-\lambda'' s/(s+1)}}{s^2} \Big)_{r=r''} +$$

$$\sum_{i=2}^{N} (-)^{i} a_i'' (i-1)! \, L^{-1} \Big[\Big(\frac{s+1}{\lambda'' s} \Big)^{i-1} \frac{e^{-\lambda'' s/(s+1)}}{s^2} \Big]_{r=r''} +$$

$$\sum_{i=2}^{N} a_i'' \Big[r'' + (i-1)! \sum_{j=1}^{i=1} \frac{(-)^{j}}{(i-1-j)!} \Big(\frac{1}{\lambda''} \Big)^{j} L^{-1} \Big(\frac{1}{s^2} \Big(\frac{s+1}{s} \Big)^{j} \Big) \Big]_{r=r''}$$

.. (4.2.(5))

Now

$$\frac{1}{s^2} \Big(\frac{s+1}{s} \Big)^{j} = \frac{1}{s^2} \sum_{k=0}^{j} \binom{j}{k} \Big(\frac{1}{s} \Big)^{k} = \sum_{k=0}^{j} \binom{j}{k} \Big(\frac{1}{s} \Big)^{k+2}$$

So,

$$L^{-1} \Big[\frac{1}{s^2} \Big(\frac{s+1}{s} \Big)^{j} \Big]_{r=r''} = \sum_{k=0}^{j} \binom{j}{k} \frac{(r'')^{k+1}}{(k+1)!}$$

Hence, (4.2.(5)) implies:

$$\eta^* = a_1'' r'' - a_1'' L^{-1} (\exp(-\lambda' s/(s+1))/s^2)_{r=r''} +$$

$$\sum_{i=2}^{N} (-)^i a_i'' (i-1)! \, L^{-1} [(\frac{s+1}{\lambda'' s})^{i-1} \exp(-\lambda' s/(s+1))/s^2]_{r=r''} +$$

$$\sum_{i=2}^{N} a_i'' [r'' + (i-1)! \sum_{j=1}^{i-1} \frac{(-)^j}{(i-1-j)!} (\frac{1}{\lambda''})^j \, .$$

$$. \sum_{k=0}^{j} \binom{j}{k} \frac{(r'')^{k+1}}{(k+1)!}] \qquad .. (4.2.(6))$$

This expression (4.2.(6)) can be utilized to obtain time scale transformation for variable flow.

4.3.Comparison of results: Because Willmott [25] has expanded on work done by NW [17] and he reports results which compare favourably with earlier authors, it seems fit to compare results with those obtained by him.

In table 4-1, values for thermal efficiency obtained using q and z-scale programs are displayed along with the values obtained by Willmott [25] in table 4 of that paper.

As noted in section 2.3, Willmott [23,25] uses a thermal ratio η_{REG} (denoted as R_{REG} in section 2.3). For $\lambda' = \lambda''$, $\pi' = \pi''$ thermal efficiency is comparable with Willmott's cold side thermal ratio.

TABLE 4-1

A)WILLMOTT'S VALUE AS GIVEN IN TABLE 4 OF[25]
B)Q-SCALE SINGLE PRECISION VALUE
C)Z-SCALE DOUBLE PRECISION VALUE

ORDER	REDUCED	PERIOD			
OF	LENGTH	1		2	
POLY-	1 A)	.3221		.2930	
NOMIAL	10 A)	.8322		.8289	
		B)	C)	B)	C)
2	1	.320999496	.320999495	.290368305	.290368318
	10	.822213717	.822213630	.793063351	.793063290
3	1	.322055246	.322055268	.292958043	.292958046
	10	.824858113	.824858297	.807307587	.807307352
4	1	.322076646	.322076609	.292982484	.292982491
	10	.829520753	.829520666	.823277187	.823277176
5	1	.322078199	.322078210	.292984323	.292984318
	10	.830919392	.830919395	.826764878	.826764914
6	1	.322078227	.322078243	.292984320	.292984332
	10	.831882038	.831882131	.828548070	.828547929
7	1	.322078230	.322078245		.292984332
	10	.832059785	.832059779	.828808904	.828808683
8	1	.322078202	.322078246		.292984332
	10	.832165958	.832166003	.828907435	.828907567
9	1	.322078294	.322078246		.292984332
	10	.832181838	.832181847	.828913909	.828913987
10	1	.322078228	.322078244		
	10	.832189707	.832189842	.828914368	.828914628
11	1	.322078234	.322078247		
	10	.832190758	.832190813	.828913966	.828913937
12	1	.322078301	.322078263		
	10	.832191538	.832191128	.828913159	.828913732
13	1	.322078398	.322078110		
	10	.832191346	.832192222	.828915576	.828913897
14	1	.322078582	.322100280		
	10		.831994410		
15	1		.323120357		
	10	.807895893	.817866873	.832636686	.815460536

Efficiency E as defined in (2.3.(4)) has been utilised to derive (3.2.(2)) from which values for b) i.e., efficiencies for q-scale single precision have been calculated in the program. Similarly efficiencies c) **for z-scale double-precision** have been obtained through the use of equation (3.4.(21)). The values a) have been obtained straight from the table 4 of Willmott [25].

In table 4-1 it should be noted that only the smallest and largest bed length used by Willmott, namely, 1 and 10 are used in comparison. Although other values can be easily obtained and a number of tables produced,it is hoped that the table 4-1 alone will suffice.

From table 4-1 we can conclude that the proposed method (both for q and z scales) is quite compatible with the results reported by Willmott.

For bed length $\lambda = 1$ and period $\pi = 3$ agreement with Willmott's values is very close, i.e., upto 4 decimal digits for 3rd to 8th order of polynomial. We notice that b) values oscillate starting from 5th order and so values beyond 6th order were not obtained. Also c) values show a similar trend and become steady at 7th and 8th order. It is thus safe enough to conclude that for these parameters 4th to 7th order polynomial can be used.

For bed length $\lambda = 1$ and period $\pi = 2$ also very close agreement is found with Willmott's value. Here b) values increase slightly as the order of the polynomial increases but there is a very slight drop (in the 9th decimal place) when the thermal efficiency for the 6th order polynomial is compared with that of the 5th order polynomial. So it is assumed that for b) values this is where the stable value of thermal efficiency has occurred and no further b) values are calculated for these parameters. Whereas, for c) values a similar trend is noticeable and values remain constant for polynomial orders 6, 7, 8 and even 9. It is thus concluded that for $\lambda = 1$, $\pi = 2$ the suitable order of polynomial lies in the range 5 to 8.

Similar conclusions can be drawn about the thermal efficiencies for $\lambda = 1$ and $\pi = 1$. Although it is clear that b) values oscillate after 6th order and c) values after remaining steady at 8th and 9th, start oscillating and eventually achieve 4 decimal place accuracy at 14th order, the behaviour of c) values beyond 9th order does not represent the general trend, this is due to ill-conditioning setting in at 10th order of polynomial. So, we conclude in general that for short bed lengths and periods of 1, 2 and 3 very close agreement (at least upto 3 decimal places) with Willmott's values takes place for small order of polynomial and any of 4, 5 or 6th order polynomial will be suitable for these short bed lengths.

However, for larger bed length $\lambda = 10$ and period $\pi = 3$, agreement with Willmott's value to 4th decimal place occurs when the order of polynomial is 7 or 8. Both b) and c) values show the same trend of reaching upto a stable value and then degenerating.

We note that for bed length $\lambda = 10$ and period $\pi = 2$ similar (upto 4th decimal place) agreement requires order of polynomial to be 8, 9, 10, 11, 12 or 13 but thermal efficiency for both b) and c) oscillates (in the 6th decimal place) after 10th order of polynomial.

For bed length $\lambda = 10$ and period $\pi = 1$, in order to obtain very close agreement with Willmott's value the order of polynomial should be 11 or more but not greater than 13 as b) values decrease after 12th order. However, c) values increase upto 13th order and then come down for 14th and 15th order of polynomial. So, any of 11, 12 or 13th order of polynomial is quite appropriate for these parameters. Hence, it is concluded that for larger bed lengths where periods are 1, 2 and 3, the order of polynomial should be chosen from 8, 9, 10, 11, 12 or 13. In general, higher orders must be chosen for shorter period and smaller orders for larger period.

Comparing b) values against c) values, if one does not consider the 15th order polynomial (for which unsatisfactory results occur), then a fairly close agreement at least upto 5th decimal place is found. On average, most of the values agree at least upto 6th decimal place and a fair few even upto 8th decimal place.

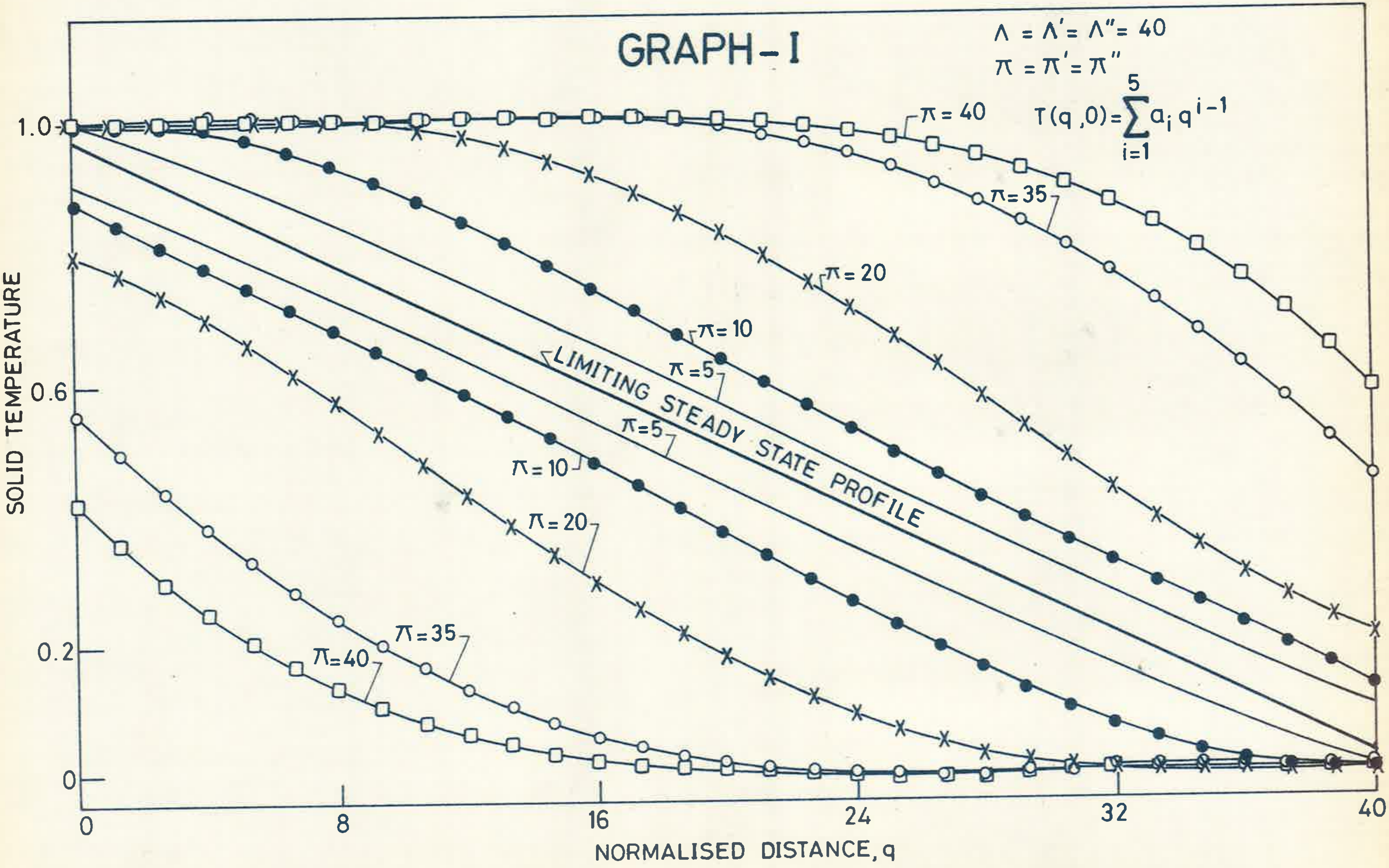
This indicates that there is not much difference in precision of the two methods despite the fact that c) values have been obtained by using double precision calculations. But, we notice that in general c) values show a more "smooth and stable" trend than b) values which appear to be more oscillatory e.g., 7, 8, 9 and 10th order values for $\pi = 1$ in table 4-1. Therefore, the method involving the z-scale double precision could be given a preference over the single precision method although the results may not be appreciably different. It is however important to note that the preference may have to be exercised at the cost of run time.

4.4 Comparison of profiles:

Graphs I and II indicate the behaviour of Solid Temperatures at the start of hot and cold blows respectively along the whole length of the bed. The limiting steady state profile is a linear graph represented by:

$$T(q) = \frac{1 + q}{\lambda'' + 2} \qquad .. \ (3.3.(23)) \text{ as derived in section 3.3.}$$

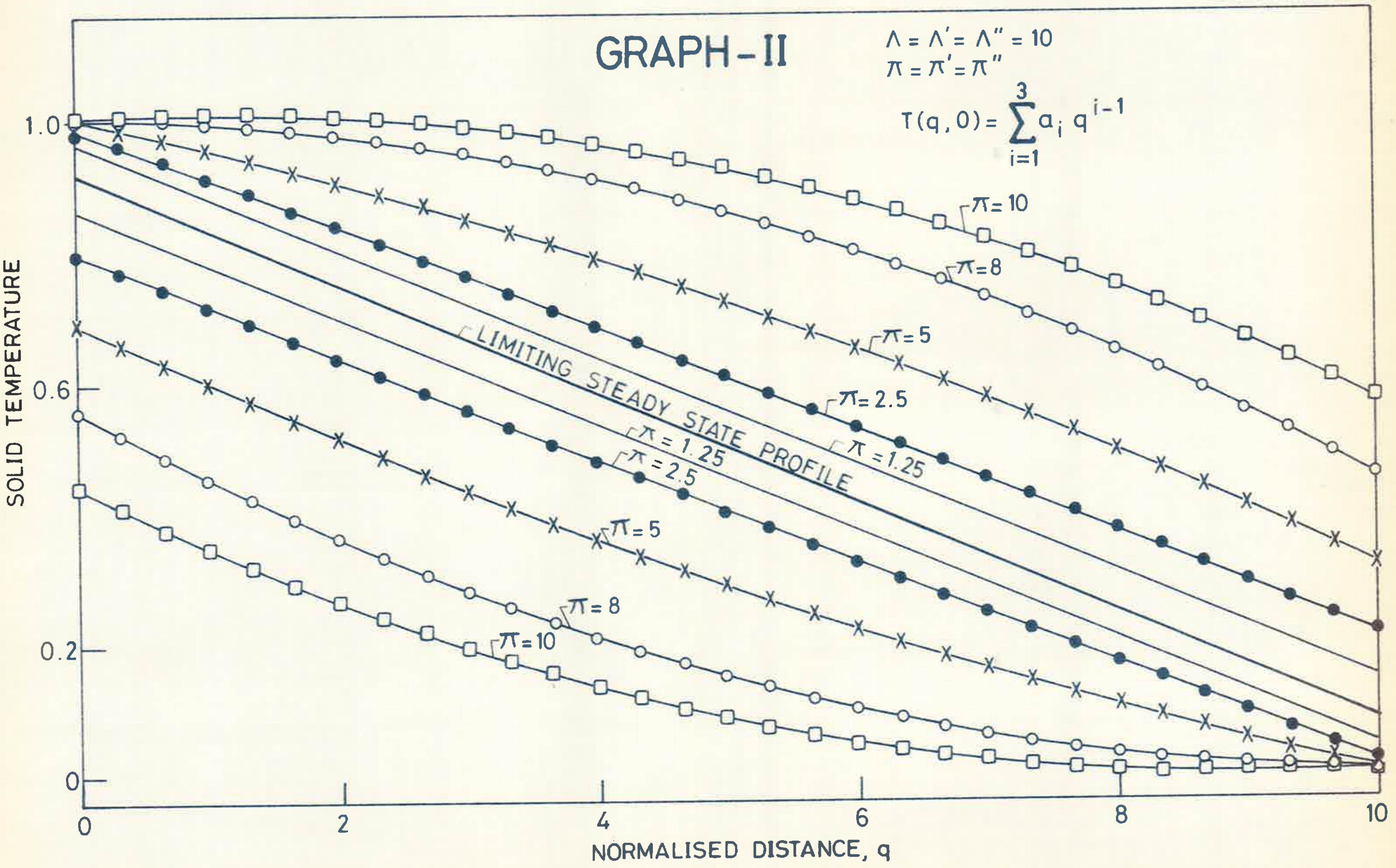
where $q \in [0, \lambda'']$.


GRAPH - I
$\Lambda = \Lambda' = \Lambda'' = 40$
$\pi = \pi' = \pi''$
$T(q,0) = \sum_{i=1}^{5} a_i q^{i-1}$
$\pi = 40$
$\pi = 35$
$\pi = 20$
$\pi = 10$
$\pi = 5$
LIMITING STEADY STATE PROFILE
$\pi = 5$
$\pi = 10$
$\pi = 20$
$\pi = 35$
$\pi = 40$
SOLID TEMPERATURE
1.0
0.6
0.2
0
0
8
16
24
32
40
NORMALISED DISTANCE, q

In Graph I the reduced length λ is 40. The curves below the limiting steady state profile represent the temperatures of solid matrix along the whole bed length at the start of hot blow for periods $\pi' = 5$ to $\pi' = 40$, as indicated. The curves above the limiting profile represent temperatures at the start of cold blow. As here a large reduced length $\lambda = 40$ was being used with periods varying from $\pi = 5$ to $\pi = 40$, a fairly "average" value for the order of the polynomial was to be chosen, hence initial solid temperature was chosen to be of 5th order.

An off-line CALCOMP plotter was used to obtain all the curves except the limiting profile which was drawn manually via equation (3.3.(23)). The plotter routines (not attached here) were written so that X-Y data from the program could straightaway be taken and plotted. Provision for drawing many curves on the same plot was deliberately included so that many curves could be plotted once the data had been taken up by the plotting routine from the driving program and interactive terminal. Graphs I and II are a draughtsman's true copy of the original CALCOMP plotter output.

It is quite evident from Graph-I that for a fixed reduced length the solid temperature profiles approach the straight line limiting profile as the period is decreased from $\pi = 40$ down to $\pi = 5$.


GRAPH-II
$\Lambda = \Lambda' = \Lambda'' = 10$
$\pi = \pi' = \pi''$
$T(q,0) = \sum_{i=1}^{3} a_i q^{i-1}$
$\pi = 10$
$\pi = 8$
$\pi = 5$
$\pi = 2.5$
$\pi = 1.25$
LIMITING STEADY STATE PROFILE
$\pi = 1.25$
$\pi = 2.5$
$\pi = 5$
$\pi = 8$
$\pi = 10$
SOLID TEMPERATURE
1.0
0.6
0.2
0
0
2
4
6
8
10
NORMALISED DISTANCE, q

A similar behaviour is noticeable in Graph-II where the reduced length $\lambda = 10$ is smaller and hence only a 3rd order polynomial was used. Here also curves below the limiting profile are those at start of hot blow and the ones above represent temperatures at the start of cold blow for the parameters indicated.

4.5 Discussion on ill-conditioning and other limitations:

The program CLOZAK (see appendices A1, A2 and A3) uses CDC Library 'MATRIX' routines to invert matrices and perform other matrix operations (see appendix M for full details).

We shall discuss the inversion of matrices with respect to q-scale. Discussion with respect to z-scale will follow on parallel arguments so the arguments will not be repeated for z-scale.

Now, with respect to q-scale the matrices to be inverted are:

$[B'']$ and $[B'] - [C''] [B'']^{-1} [C']$, see section 3.1 for definitions.

As $q_1' = 0$, $q_2' = \lambda'/(N-1)$ and $q_i' = (i-1)q_2'$ for $i = 3,4,\ldots,N$ are evaluated in the program, it is impossible to have two rows or two columns equal in $[B'']$. As we

OUTPUT- I

```
..RUN,F,F=C3

ENTER VALUE OF N--
   45000B   CM STORAGE USED
      2.449 CP SECONDS COMPILATION TIME
   CM LWA+1 = 33465B, LOADER USED  47300B6

ENTER HOT AND THEN COLD LAMBDA--1.,1.

ENTER HOT AND THEN COLD PERIOD--1.,1.

DET= -.11325E-05
DET=  .37847E-06
ARE HOT BLOW COEFFTS.REQUIRED?YES OR NO--N

ARE COLD BLOW COEFFTS.REQUIRED?YES OR NO--N

ARE TEMPERATURE PROFILES FOR SOLID REQUIRED?ENTER YES OR NO--N

ARE EXIT FLUID TEMPS.REQUIRED?YES OR NO--N

INTERESTED IN THERMAL EFFICIENCY? YES OR NO: Y

THERMAL EFFICIENCY=  .322078227
ANOTHER RUN WITH DIFFERENT PARAMETERS?
ENTER YES OR NO --N

ANOTHER RUN WITH SAME PARAMS. BUT DIFFERENT ORDER?
ENTER YES OR NO--Y

ENTER VALUE OF N--7

DET= -.11343E-08
DET=  .32734E-09
ARE HOT BLOW COEFFTS.REQUIRED?YES OR NO--N

ARE COLD BLOW COEFFTS.REQUIRED?YES OR NO--N

ARE TEMPERATURE PROFILES FOR SOLID REQUIRED?ENTER YES OR NO--N

ARE EXIT FLUID TEMPS.REQUIRED?YES OR NO--N

INTERESTED IN THERMAL EFFICIENCY? YES OR NO: Y

THERMAL EFFICIENCY=  .322078230
ANOTHER RUN WITH DIFFERENT PARAMETERS?
ENTER YES OR NO --N

ANOTHER RUN WITH SAME PARAMS. BUT DIFFERENT ORDER?
ENTER YES OR NO--N

     STOP
```

consider the case $N \geqslant 2$ only, it is expected that in general linear dependence between rows or columns will not occur as we are taking powers of each element of the second column to obtain the rest of entries in a row. However, ill-conditioning may occur if elements of the second column of $[B'']$ are small because then the rest of the elements become smaller in columns 3 onwards.

We note (see output-I attached) for small $\lambda = \lambda' = \lambda'' = 1$ and period $\pi = \pi' = \pi'' = 1$, the determinants of $[B'']$ and $[B'] - [C''] [B'']^{-1} [C']$ respectively are of the order 10^{-5} and 10^{-6} for a 6th order polynomial, and 10^{-8} and 10^{-9} for a 7th order polynomial, yet coefficients are calculated and the value for thermal efficiency differs from Willmott's [25] value in the 4th decimal place.

For greater values of λ the values of these determinants were found to have increased, so much so that for $\lambda = \lambda' = \lambda'' = 10$ and polynomial of 13th order the determinants were of the order 10^{38} and 10^{36} respectively; yet thermal efficiency was obtained and its value differed in the 4th decimal place from the value reported by Willmott [25], see table 4-1. At the same time it was observed that for $N = 15$, determinants were extremely large (10^{50} and 10^{47}) and although the program itself did not stop executing, the thermal efficiency obtained was unsatisfactory.

OUTPUT - II

```
   CM LWA+1 = 33617B, LOADER USED 47500B

ENTER HOT AND THEN COLD LAMBDA--20. 20.

ENTER HOT AND THEN COLD PERIOD--1. 1.

ENTER N   5
DET=  .27466E-03
DET=  .17438E-06
ETAREG=  .905039055
ANOTHER RUN? YES=1  1

ENTER N   6
DET= -.11325E-05
DET=  .30154E-09
ETAREG=  .906859011
ANOTHER RUN? YES=1  1

ENTER  N   8
DET=  .27264E-12
DET=  .23525E-16
ETAREG=  .908264406
ANOTHER RUN? YES=1  1

ENTER N   10
DET= -.21024E-21
DET=  .86702E-26
ETAREG=  .908654424
ANOTHER RUN? YES=1  1

ENTER N   12
DET=  .49274E-33
DET=  .11571E-37
ETAREG=  .908737527
ANOTHER RUN? YES=1  1

ENTER N   14
DET= -.33909E-47
DET=  .49621E-52
ETAREG=  .908964755
ANOTHER RUN? YES=1  1

ENTER N   15
DET= -.31346E-55
DET=  .36973E-60
ETAREG=  .908635536
ANOTHER RUN? YES=1  1

ENTER N   17
DET=  .32552E-73
DET=  .25933E-78
ETAREG=************
ANOTHER RUN? YES=1  1

ENTER N   16
DET=  .66972E-64
DET=  .64167E-69
ETAREG=************
ANOTHER RUN? YES=1  0

     STOP
     14.432 CP SECONDS EXECUTION TIME
```

Hence it is concluded that as very large order polynomial (beyond N = 14) will be used, ill-conditioning is expected to creep in.

The above mentioned fact about ill-conditioning is also borne out by the z-scale program, where, as expected, the determinants become exceedingly smaller as the order of polynomial is increased.

Referring to output-II (result of program written for z-scale just for testing the convergence of thermal efficiency), we notice that for $\lambda' = \lambda'' = 20$ and $\pi' = \pi'' = 1$ the ETAREG (i.e., thermal efficiency) converges smoothly upto N = 14 (having started from N = 5). But, as N = 15 is used ill-conditioning takes over and beyond this for N = 16 and N = 17 ETAREG values become so small that they are not output (asterisks are printed). We also note that determinants to the order of 10^{-73} and 10^{-78} are used in the inversion of matrices, yet the program does not stop by itself and further calculations (if required) are carried out.

As the inversion of $[A] = [B'] - [C''] [B'']^{-1} [C']$ is mainly dependent upon $[B'']^{-1}$, judging from the definitions of $[B']$, $[C'']$ and $[C']$ (in chapter 3), it is 'safe' to say that if $[B'']$ can be inverted so can $[A]$. This is also noticeable from the proximity of the absolute value of the determinants of these matrices (see Output-II).

So, we conclude that although ill-conditioning is a particularly disadvantageous and limiting but unavoidable aspect of these programs, the programs themselves are fairly "flexible" and can be stretched to limits such as 10^{-75} and 10^{75} in calculation of determinants for inversion of matrices.

We now turn to other limitations of these programs (see appendices A1, A2, A3 for listings etc.).

The programs have been written to accommodate an N value of upto 20, if a higher order elementary distribution is required, the dimension statements will have to be changed. It is worth a mention here that on CYBER 6000 with dimension statements of 50, the core space was used up and the program would not work.

As the library routine MATRIX (see appendix M) is being used to invert and to do other operations, if a matrix is singular; the determinant is set to zero and no further calculation is done. One should obtain "DET = 0" (or words to this effect) in the output, and further execution will cause an 02 mode arithmetic error on the FORTRAN compiler.

As noted earlier, N value entered should be greater than 1 otherwise division by zero is likely to be attempted and this would cause an 02 mode response to be output.

Number of hot and cold bed mesh points should be at least 2, otherwise division by zero would be attempted.

As such there is no restriction on the number of intervals of hot and cold bed length, but obviously greater the number of intervals, greater the execution time. So CP (Central Processor) time limit could be exceeded if very fine intervals are used.

4.6 Summary and concluding remarks:

In this chapter at first a description of the computer programs presented in appendices A1, A2 and A3 is given. This is followed by derivation of time scale transformation for variable flow which allows constant mass flow solutions to be transformed to variable mass flow.

When comparing results with those obtained by Willmott [25] it was found that for short bed lengths ($\lambda = 1$) very close agreement (to 4 decimal digits) was obtained between Willmott's "thermal ratios" and the proposed closed form method with only 3 to 5 terms in the polynomial expansion for reduced period values of 1, 2 and 3.

However, a longer bed length $\lambda = 10$ required a greater number of terms (upto $N = 15$) for the same degree of convergence. This result is generally as predicted by Willmott and Thomas [26].

A comparison of solid temperature profiles with those predicted by limiting "zero period" model by Jeffreson [8] was made. The solid profiles are found to be converging to the limiting profile. This comparison explains and supports some of the results reported by Willmott and Thomas [26], which were found in agreement in the preceding paragraph.

An analysis of ill-conditioning is then reported along-with the limitations of the proposed method. It is found that ill-conditioning experienced by Willmott through Illiffe's method does not occur with the proposed method for the same parameters. However, it is reported that ill-conditioning is likely to take over if the order of polynomial N becomes 15 or more.

In comparing Willmott's open trapezoidal method [25] with the proposed method, it was found that for the same parameters the proposed method is relatively more efficient e.g., a typical run for a 6th order polynomial of CLOZAK used 1.642 CP seconds compilation and .906 CP seconds execution time for $\lambda = 20$, $\pi = 5$, whereas the trapezoidal program (much simpler to write) for the same parameters converged after 33 cycles and took .797 CP seconds compilation and 4.602 CP seconds execution time. It is obvious that for such normal parameters there is a 'good' (to the order of 3 CP seconds) saving of RUN time. It is expected that for larger parameters the open

Trapezoidal method will require more cycles to converge and so will need more execution time, whereas the increase in CLOZAK's execution will be comparatively smaller.

So, in conclusion we state that the proposed method, being relatively superior than the existing methods, will be useful in calculation of temperature profiles, thermal efficiency and other results connected with a thermal regenerator.

It is recommended that the z-scale double precision program be utilized although single precision q-scale program is not very far in precision.

APPENDIX M

ABOUT CDC LIBRARY ROUTINE NAMED 'MATRIX'

The main external routines used in program CLOZAK are used for finding inverses and multiplication of matrices. We will not discuss the multiplication routine as it is quite straightforward and self explanatory in [31]. Here we describe the subroutine used for inversion as explained on page 13 of [31]. The calling sequence is as follows:

```
CALL MATRIX (10,m,n,kop,a,ka,b)
```

where 10 calls the inversion subroutine.

m is number of rows of A (the matrix to be inverted).

n is number of columns of A; $n \geq m$.

kop determines how the search for pivot is made.

If kop = 0, the entire matrix is searched each time.

If kop = 1, the first row is searched the first, the second the second time, and so on.

If kop = 2, no search. The diagonal elements, from upper left to lower right are used in turn as pivot elements.

a is Matrix A.

The first m columns contain the matrix of coefficients. If $n > m$, the remaining n-m columns contain the right-hand sides of the n-m sets of linear equations to be solved. (Note: for inversion only, we need have n = m).

ka is column size of A; ka $\geqslant$ m (as reserved in the dimension statement for A).

b is determinant (obtainable on output).

The subroutine uses Gauss-Jordan elimination method employing the pivoting option as requested by value of kop. For the program CLOZAK kop is chosen to be zero because this is envisaged to be the most efficient way of reducing a matrix. If at any time the pivot is such that a division by zero is likely to occur and/or determinant is zero, exit from subroutine is made with b = 0.

APPENDIX A1

This appendix has been basically written to incorporate the FORTRAN program CLOZAK attached herewith. For a description of computer program section 4.1 of the last chapter **should be consulted.** Here we outline the purpose of this program. This program obtains coefficients a' and a'' as given in equations (3.1.(22)) and (3.1.(23)) using single precision arithmetic and q-scale where $q \varepsilon [0, \lambda]$. The parameters to be input to the program are: N, π', π'', λ', λ''. The coefficients a' and a'' are output. Then if required,temperature profiles for solid are obtained and printed for hot and cold blow. This is followed by calculation and printing of thermal efficiency E on demand. The program is written in such a way as to allow a consecutive number of runs with different parameters in the same execution.

A typical output is also attached in this **appendix** where the following sequence of events take place:

When the program is 'RUN' at first the question 'IS THIS YOUR SECOND RUN? ENTER YES OR NO -- ' is prompted (along with compilation time and other system details). An answer to this was given as NO (the last characters on fifth line of output). Then as a reply to 'ENTER VALUE OF N --' 5 is entered thereby defining

$N = 5$. Then the values $\lambda' = 19.7254$ and $\lambda'' = 17.3553$ alongwith $\pi' = \pi'' = 8.5$ are entered as required. This is followed by printing of the Determinant values of matrices to be inverted, here we note that dets are fairly large(being of the order of 10^9) yet the program does not report any ill-conditioning. Then the coefficients a' and a" are printed. Note that these are $N = 5$ in number. When the question whether solid profiles are required is answered in affirmative, the number of bed mesh points i.e., number of points on the q-scale are required to be input. This value is given here as 31, on which solid temperature profiles are output for hot and cold blow. Then the thermal efficiency is output as it was asked for and here it is noted that for the parametersit is a fairly efficient thermal regenerator. After this program comes to a STOP as no more runs with different parameters or different order were required.

It should be noted here that outputs obtained from CDC 6000 Computer System have been printed using DEC-1090 system housed at Indian Institute of Science, Bangalore, India, with their kind permission.

..

Iqtedar Askari Abdi.

<ins>PLEASE NOTE</ins>:

The computer printout pages A1-1 to A4-4 have a line missing at the bottom of each page.

An unsuccessful attempt was made to obtain completed pages through the office of the Assistant Registrar in September 1982.

A1-1

```
      PROGRAM CLOZAK(INPUT,OUTPUT,XYDATA,TAPE1=INPUT,TAPE2=OUTPUT,
     $TAPE3=XYDATA)
      COMPLEX ALFA,VK
      DIMENSION X1(20),X2(20),US(20),AH(20),AC(20),CC(20,20),
     $CH(20,20),BC(20,20),BH(20,20),C1(20,20),C2(20,20),CIN(20,20),
     $ALFA(15),VK(15)
C   THIS PROGRAM SOLVES FOR VECTORS AH AND AC WHICH ARE COEFICIENTS OF
C   THE FINITE SERIES APPROXIMATION FOR INITIAL HOT AND COLD BLOWS.
      WRITE(2,720)
 720 FORMAT(1X,'IS THIS YOUR SECOND RUN ?  ENTER YES OR NO-- ')
      READ(1,110)AND
 600  WRITE(2,1)
   1  FORMAT(1X,'ENTER VALUE OF N-- ')
      READ(1,*)N
      WRITE(2,2)
   2  FORMAT(1X,'ENTER HOT AND THEN COLD LAMBDA--')
      READ(1,*)HLAM,CLAM
      WRITE(2,3)
   3  FORMAT(1X,'ENTER HOT AND THEN COLD PERIOD--')
      READ(1,*)PIH,PIC
 500  N1=N-1
      X1(1)=0.  $  X2(1)=CLAM  $  X1(N)=HLAM
      X1(2)=HLAM/N1  $  X2(2)=CLAM*(1.-X1(2)/HLAM)  $  X2(N)=0.
      DO 4 I=3,N1
      X1(I)=(I-1)*X1(2)
 4    X2(I)=CLAM*(1.-X1(I)/HLAM)
      DO 5 I=1,N
      BH(I,2)=X1(I)
      BC(I,2)=X2(I)
 5    AH(I)=IFAC(I-1)
      DO 40 J=3,N
      J1=J-1
      DO 40 I=1,N
      BH(I,1)=BC(I,1)=1.
      BH(I,J)=BH(I,2)**J1
      BC(I,J)=BC(I,2)**J1
 40   CONTINUE
      CALL ZACOFS(ALFA,VK)
      US(1)=1.-EXP(-PIH)
      DO 6 J=2,N
 6    US(J)=ZAK(0,X1(J),PIH,ALFA,VK)
      DO 7 I=1,N1
      I1=I+1
      DO 7 J=1,N
      CH(I1,J)=ZAK(J,X1(I1),PIH,ALFA,VK)*AH(J)
      CC(I,J)=ZAK(J,X2(I),PIC,ALFA,VK)*AH(J)
 7    CONTINUE
      DO 20 J=2,N
 20   CH(1,J)=CC(N,J)=0.
      CH(1,1)=EXP(-PIH)$CC(N,1)=EXP(-PIC)
      CALL MATRIX(10,N,N,0,BC,20,DET)
      WRITE(2,30)DET
 30   FORMAT(1X,'DET=',E12.5)
      CALL MATRIX(20,N,N,N,CC,20,BC,20,C1,20)
      CALL MATRIX(20,N,N,N,C1,20,CH,20,C2,20)
      DO 10 I=1,N
      DO 10 J=1,N
 10   CIN(I,J)=BH(I,J)-C2(I,J)
      CALL MATRIX(10,N,N,0,CIN,20,DET)
      WRITE(2,30)DET
      DO 11 I=1,N
      C2(I,1)=C1(I,1)*US(1)
      DO 11 J=2,N
      C2(I,1)=C2(I,1)+C1(I,J)*US(J)
 11   CONTINUE
```

```
      AH(I)=CIN(I,1)*C2(1,1)
      DO 12 J=2,N
      AH(I)=AH(I)+CIN(I,J)*C2(J,1)
12    CONTINUE
      WRITE(2,13)
13    FORMAT(1X,'COEFFICIENTS OF HOT BLOW--'/1X,28(1H=))
      WRITE(2,14)(AH(I),I=1,N)
14    FORMAT(3(2(1X,E12.5)))
      DO 15 I=1,N
      CIN(I,1)=BC(I,1)*US(1)
      DO 15 J=2,N
      CIN(I,1)=CIN(I,1)+BC(I,J)*US(J)
15    CONTINUE
      CALL MATRIX(20,N,N,N,BC,20,CH,20,C2,20)
      DO 16 I=1,N
      CIN(I,2)=C2(I,1)*AH(1)
      DO 16 J=2,N
      CIN(I,2)=CIN(I,2)+C2(I,J)*AH(J)
16    CONTINUE
      DO 17 I=1,N
17    AC(I)=CIN(I,1)+CIN(I,2)
      WRITE(2,18)
18    FORMAT(1X,'COEFFICIENTS OF COLD BLOW--'/1X,28(1H=))
      WRITE(2,14)(AC(I),I=1,N)
      WRITE(2,98)
98    FORMAT(1X,'ARE TEMPERATURE PROFILES FOR SOLID REQUIRED?',
     $'ENTER YES OR NO--')
      READ(1,110)ANS
110   FORMAT(A1)
      IF(ANS.EQ.1HN)GOTO 200
      IF(AND.EQ.1HY)GOTO 700
      WRITE(2,45)
45    FORMAT(1X,'ENTER NUMBER OF BED MESH POINTS--')
      READ(1,*)MB
700   HM=HLAM/(MB-1.)$CM=CLAM/(MB-1.)$XH=0.$XC=CLAM
      TH=AH(1)$TC=AC(1)
      DO 47 I=2,N
47    TC=TC+AC(I)*XC**(I-1)
      WRITE(2,46)
46    FORMAT(1X,' SOLID TEMPERATURE PROFILE'/1X,26(1H=)/2X,
     $'NORMALISED DISTANCE',10X,'HOT BLOW',10X,'COLD BLOW')
      IF(AND.EQ.1HY)GOTO 610
      WRITE(3,*)XH,TH,TC
      GOTO 620
610   WRITE(3,*)TH,TC
620   WRITE(2,48)XH,TH,TC
48    FORMAT(5X,F12.9,10X,F12.9,8X,F12.9)
      DO 49 J=2,MB
      XH=XH+HM$XC=XC-CM$TH=AH(1)$TC=AC(1)
      DO 50 I=2,N
      TH=TH+AH(I)*XH**(I-1)
50    TC=TC+AC(I)*XC**(I-1)
      IF(AND.EQ.1HY)GOTO 630
      WRITE(3,*)XH,TH,TC
      GO TO 640
630   WRITE(3,*)TH,TC
640   WRITE(2,48)XH,TH,TC
49    CONTINUE
200   WRITE(2,201)
201   FORMAT(1X, 'INTERESTED IN THERMAL EFFICIENCY? YES OR NO:')
      READ(1,110)ANS
      IF(ANS.EQ.1HN)GOTO 999
      ETAREG=AC(1)*CLAM$SUM=AH(1)*HLAM
      DO 90 I=2,N
      SUM=SUM+AH(I)*(HLAM**I)/I
90    ETAREG=ETAREG+AC(I)*(CLAM**I)/I
```

```
      ETAREG=HLAM*ETAREG/(PIH*CLAM)-SUM/PIH
      WRITE(2,91)ETAREG
 91   FORMAT(1X,'THERMAL EFFICIENCY=',F12.9)
 999  WRITE(2,300)
 300  FORMAT(1X,'ANOTHER RUN WITH DIFFERENT PARAMETERS?'/1X,
     $'ENTER YES OR NO --')
      READ(1,110)AND
      IF(AND.EQ.1HY)GOTO 600
      WRITE(2,301)
 301  FORMAT(1X,'ANOTHER RUN WITH SAME PARAMS. BUT DIFFERENT ORDER?'/
     $1X,'ENTER YES OR NO--')
      READ(1,110)ANS
      IF(ANS.EQ.1HN)STOP
      WRITE(2,1)
      READ(1,*)N
      GOTO 500
      END
      SUBROUTINE ZACOFS(ALFA,VK)
C  THIS SUB EVALUATES THE ZAKIAN'S COEFFICIENTS FOR PADE" APPROXIMATION.
      COMPLEX ALFA,VK
      DIMENSION ALFA(15),VK(15)
      ALFA(1)=CMPLX(1.953965103383233E+01,0.)
      A=1.933570620860414E+01 $ B=3.379399088597234E+00
      ALFA(2)=CMPLX(A,B)$ALFA(3)=CMPLX(A,-B)
      A=1.871433205030498E+01 $ B=6.772981645000657E+00
      ALFA(4)=CMPLX(A,B)$ALFA(5)=CMPLX(A,-B)
      A=1.764452177737171E+01 $ B=1.019774391691488E+01
      ALFA(6)=CMPLX(A,B)$ALFA(7)=CMPLX(A,-B)
      A=1.606503145933782E+01 $ B=1.367780303746658E+01
      ALFA(8)=CMPLX(A,B)$ALFA(9)=CMPLX(A,-B)
      A=1.386207821875056E+01 $ B=1.725343258836830E+01
      ALFA(10)=CMPLX(A,B)$ALFA(11)=CMPLX(A,-B)
      A=1.080652491390860E+01 $ B=2.100620730400388E+01
      ALFA(12)=CMPLX(A,B)$ALFA(13)=CMPLX(A,-B)
      A=6.301979854806708E+00 $ B=2.516447268568806E+01
      ALFA(14)=CMPLX(A,B)$ALFA(15)=CMPLX(A,-B)
      VK(1)=CMPLX(1.645619599469101E+08,0.)
      A=-1.268572985368156E+08 $ B=-4.749121856114229E+07
      VK(2)=CMPLX(A,B)$VK(3)=CMPLX(A,-B)
      A=5.574098488453668E+07 $ B=4.999812394225393E+07
      VK(4)=CMPLX(A,B)$VK(5)=CMPLX(A,-B)
      A=-1.136893282997024E+07 $ B=-2.450428931522289E+07
      VK(6)=CMPLX(A,B)$VK(7)=CMPLX(A,-B)
      A=-1.694097331423655E+05 $ B=6.009306354368669E+06
      VK(8)=CMPLX(A,B)$VK(9)=CMPLX(A,-B)
      A=4.138883037657413E+05 $ B=-6.184004276684025E+05
      VK(10)=CMPLX(A,B)$VK(11)=CMPLX(A,-B)
      A=-4.058457858252957E+04 $ B=9.752029126666363E+03
      VK(12)=CMPLX(A,B)$VK(13)=CMPLX(A,-B)
      A=3.800167535061704E+02 $ B=5.088313306242982E+02
      VK(14)=CMPLX(A,B)$VK(15)=CMPLX(A,-B)
      RETURN
      END
      INTEGER FUNCTION IFAC(I)
C  THIS FUNCTION EVALUATES THE FACTORIAL VALUE OF I.
      IFAC=1
      IF(I.LE.1)RETURN
      DO 1 J=2,I
 1    IFAC=IFAC*J
      RETURN
      END
       COMPLEX FUNCTION ZAK(I,T,PI,ALFA,VK)
      COMPLEX ALFA,VK,F1,F2,ZAC
      DIMENSION ALFA(15),VK(15)
C  THIS FUNCTION EVALUATES THE INVERSE OF LAPLACED F AT TIME T USING
C  COMPLEX VECTORS ALFA AND VK WHICH ARE THE COEFFTS. OF ZAKIAN'S FORMU
```

A1-4

```
      IF(I.GT.0)GOTO 10
      ZAC=CMPLX(0.,0.)
      DO 1 J=1,15
  1      ZAC=ZAC+VK(J)*F1(ALFA(J)/T,PI)
      ZAK=CMPLX(REAL(ZAC),AIMAG(ZAC))
      ZAK=ZAK/T
      RETURN
 10   ZAK=VK(1)*F2(ALFA(1)/T,PI,I)
      DO 2 J=2,15
 2    ZAK=ZAK+VK(J)*F2(ALFA(J)/T,PI,I)
      ZAK=ZAK/T
      RETURN
      END
      COMPLEX FUNCTION  F1(XI,PI)
      COMPLEX XI
C  LAPLACE TRANSFORM FUNCTION OF US.
      F1=1.-CEXP(-XI*PI/(XI+1.))
      F1=F1/XI
      RETURN
      END
      COMPLEX FUNCTION F2(XI,PI,I)
      COMPLEX XI
C   LAPLACE TRANSFORM FUNCTION WHICH GIVES ELEMENTS OF C MATRICES, I
C   REPRESENTS THE POWER TO WHICH THE DENOMINATOR IS RAISED.
      F2=CEXP(-XI*PI/(XI+1.))
      F2=F2/(XI**I)
      RETURN
      END
      COMPLEX FUNCTION ZAP(T,PI,ALFA,VK)
      COMPLEX ALFA,VK
      DIMENSION  ALFA(15),VK(15)
C   THIS FUNCTION EVALUATES THE HEAT POLE FUNCTION BY USE OF ZAKIAN'S
C    FORMULA OF LAPLACE INVERSION.
      ZAP=VK(1)*CEXP(-ALFA(1)*PI/(ALFA(1)+T))
      DO 1 J=2,15
 1    ZAP=ZAP+VK(J)*CEXP(-ALFA(J)*PI/(ALFA(J)+T))
      ZAP=ZAP/T
      RETURN
      END
```

A1-5

```
..RUN,F,F=CD
IS THIS YOUR SECOND RUN?ENTER YES OR NO--
   45000B   CM STORAGE USED
      1,606 CP SECONDS COMPILATION TIME
   CM LWA+1 = 32334B, LOADER USED   46200BNO
ENTER VALUE OF N--5

ENTER HOT AND THEN COLD LAMBDA--19.7254,17.3553

ENTER HOT AND THEN COLD PERIOD--8.5,8.5

DET=  ,68095E+09
DET=  .84490E+09
COEFFICIENTS OF  HOT BLOW--
============================
  ,64601E+00  -.51787E-01  -.79439E-03   .13369E-03  -.22553E-05
COEFFICIENTS OF COLD BLOW--
=============================
  ,15090E+00   .35453E-01   .21431E-02   ,12339E-03  -,11649E-04
ARE TEMPERATURE PROFILES FOR SOLID REQUIRED?ENTER YES OR NO--Y

ENTER NUMBER OF BED MESH POINTS--31

 SOLID TEMPERATURE PROFILE
==========================
 NORMALISED DISTANCE               HOT BLOW          COLD BLOW
     0,000000000                 .646014972         .999927975
      ,657513333                 .611658727        1.008742666
     1,315026667                 ,576837723        1.009978911
     1,972540000                 .541764797        1.004385796
     2,630053333                 .506642672         .992681095
     3,287566667                 .471663952         .975551266
     3,945080000                 .437011126         .953651457
     4,602593333                 .402856563         .927605501
     5,260106667                 .369362519         .898005917
     5,917620000                 .336681132         .865413913
     6,575133333                 .304954422         .830359383
     7,232646667                 .274314294         .793340907
     7,890160000                 .244882535         .754825752
     8,547673333                 .216770815         .715249872
     9,205186667                 .190080689         .675017909
     9,862700000                 .164903593         .634503189
    10,520213333                 .141320848         .594047728
    11,177726667                 .119403657         .553962227
    11,835240000                 .099213108         .514526073
    12,492753333                 .080800169         .475987341
    13,150266667                 .064205695         .438562794
    13.807780000                 .049460421         .402437880
    14,465293333                 .036584968         .367766733
    15,122806667                 .025589838         .334672176
    15,780320000                 .016475417         .303245717
    16,437833333                 .009231975         .273547553
    17,095346667                 .003839665         .245606565
    17,752860000                 .000268522         .219420323
    18.410373333                -.001521535         .194955083
    19,067886667                -.001580703         .172145787
    19,725400000                 .000030703         .150896066
INTERESTED IN THERMAL EFFICIENCY? YES OR NO: Y

THERMAL EFFICIENCY=  .935581809
ANOTHER RUN WITH DIFFERENT PARAMETERS?
ENTER YES OR NO --N

ANOTHER RUN WITH SAME PARAMS. BUT DIFFERENT ORDER?
ENTER YES OR NO--N
```

```
.435 CP SECONDS EXECUTION TIME
```

A1-6

APPENDIX A2

The FORTRAN program CLOZAK attached to this appendix evaluates coefficients a' and a'', but this time using z parameter in double precision where $z \in [0,1]$ instead of the q-scale. This program besides obtaining the results output by the program in A1 also works out the following:

Fluid exit temperature profiles for hot and cold blows and variable time integral of cold exit fluid temperature,which are printed on demand. Here also provisions exist for consecutive runs with changed parameters in the same execution.

A typical output is also attached here with sequence of events as follows:

After the run command, value of N entered is 5, this is followed by input of $\lambda' = 19.7354$, $\lambda'' = 17.3553$, $\pi' = \pi'' = 8.5$. Then the first result output is TZ which represents $\sum_{i=1}^{15} K_i/\alpha_i$ which should be equal to 1 for the given Zakian's coefficients obtainable from ZACOFS; it should be noted that here TZ = 1.0 exactly. Values of dets of the inverted matrices are output, this time they are found to be much smaller (to the order of 10^{-3}) instead of 10^9 as was the case for q-scale. Coefficients of hot and cold blow are then output but only on demand.

As temperature profiles for solid were required, the number of bed mesh points were to be input; this is given as 31, upon which solid temperature profiles for hot and cold blow are obtained. It is noticeable that normalised distance varies from 0 to 1 this time as opposed to 0 to λ' which was the case with q-scale. Following this on demand exit fluid temperatures are output for hot and cold situations respectively. Then, as required, variable time integral of cold exit fluid temperature is obtained and printed. Thermal efficiency is also printed on demand. Here also facility for consecutive runs within the same execution are provided. It should be noted here that as λ' was "inadvertently" entered to be 19.7354 instead of 19.7254 the results although comparable with values obtained in A1 are not exactly the same and there is a slight difference which is an indication towards stability of the system (i.e., to say that slight disturbances in input only produce slight changes in output). It is also noticeable that because of double precision arithmetic the total execution time is 14.268 CP seconds.

...

A2-1

```
      PROGRAM CLOZAK(INPUT,OUTPUT,XYDATA,TAPE1=INPUT,TAPE2=OUTPUT,
     1TAPE3=XYDATA)
      DOUBLE ALFR,VKR,ALFI,VKI,FR,FI
      DIMENSION X1(20),X2(20),US(20),AH(20),AC(20),CC(20,20),
     1CH(20,20),BC(20,20),BH(20,20),C1(20,20),C2(20,20),CIN(20,20),
     2ALFR(8),VKR(8),ALFI(7),VKI(7)
C  THIS PROGRAM SOLVES FOR VECTORS AH AND AC.WHICH ARE COEFFICIENTS
C   OF THE FINITE SERIES APPROXIMATION FOR INITIAL HOT AND COLD
C   BLOWS. GENERAL UNBALANCED CASE USING 0 TO 1 SCALE.
600   WRITE(2,1)
1     FORMAT(1X,'ENTER VALUE OF N--')
      READ(1,*)N
      WRITE(2,2)
2     FORMAT(1X,'ENTER HOT AND THEN COLD LAMBDA--')
      READ(1,*)HLAM,CLAM
      WRITE(2,3)
3     FORMAT(1X,'ENTER HOT AND THEN COLD PERIOD--')
      READ(1,*)PIH,PIC
500   N1=N-1
C  SET UP VECTORS OF DISTANCE COORDINATES AND SECOND COL.OF B MATS.
      X1(1)=0.  ;  X2(1)=1.  ;  X1(N)=1.
      X1(2)=1./N1  ;  X2(2)=1.-X1(2)  ;  X2(N)=0.
      DO 4 I=3,N1
      X1(I)=(I-1)*X1(2)
4     X2(I)=1.-X1(I)
      DO 5 I=1,N
      BH(I,2)=X1(I)
      BC(I,2)=X2(I)
C  FACTORIAL VALUES IN A VECTOR
5     AH(I)=IFAC(I-1)
      DO 40 J=3,N
      J1=J-1
      DO 40 I=1,N
C  FILL REMAINDER OF B MATRICES
      BH(I,1)=BC(I,1)=1.
      BH(I,J)=BH(I,J1)*X1(I)
      BC(I,J)=BC(I,J1)*X2(I)
40     CONTINUE
C  OBTAIN ZAKIAN'S COEFFICIENTS
      CALL ZACOFS(ALFR,VKR,ALFI,VKI)
      TZ=VKR(1)/ALFR(1)
      DO 5000 J=1,7
      J1=J+1
      CALL ARITH(4,1.D0,0.D0,ALFR(J1),ALFI(J),FR,FI)
      TZ=TZ+2.*(VKR(J1)*FR-VKI(J)*FI)
5000  CONTINUE
      WRITE (2,5001)TZ
5001  FORMAT(1X,'TZ=',E12.5)
C  OBTAIN STEP RESPONSE VECTOR
      US(1)=1.-EXP(-PIH)
      DO 6 J=2,N
6     US(J)=ZAK(0,X1(J),PIH,ALFR,VKR,ALFI,VKI,HLAM)
      DO 7 I=1,N1
      I1=I+1
      DO 7 J=1,N
C  OBTAIN BOTH HEAT POLE MATRICES, BC AND BH
      CH(I1,J)=ZAK(J,X1(I1),PIH,ALFR,VKR,ALFI,VKI,HLAM)*AH(J)
      CC(I,J)=ZAK(J,X2(I),PIC,ALFR,VKR,ALFI,VKI,CLAM)*AH(J)
7     CONTINUE
      DO 20 J=2,N
20    CH(1,J)=CC(N,J)=0.
      CH(1,1)=EXP(-PIH) ; CC(N,1)=EXP(-PIC)
C     INVERT BC AND PRINT DETERMINANT
      CALL MATRIX(10,N,N,0,BC,20,DET)
      WRITE(2,30)DET
```

A2-2

```
C   FIND CC*INV(BC) AND STORE IN C1 MATRIX
          CALL MATRIX(20,N,N,N,CC,20,BC,20,C1,20)
C   FIND C2=CC*INV(BC)*CH
          CALL MATRIX(20,N,N,N,C1,20,CH,20,C2,20)
C   FIND CIN=BH-C2
          DO 10 I=1,N
          DO 10 J=1,N
10        CIN(I,J)=BH(I,J)-C2(I,J)
C   INVERT CIN AND PRINT DETERMINANT
          CALL MATRIX(10,N,N,0,CIN,20,DET)
          WRITE(2,30)DET
          DO 11 I=1,N
          C2(I,1)=C1(I,1)*US(1)
          DO 11 J=2,N
          C2(I,1)=C2(I,1)+C1(I,J)*US(J)
11        CONTINUE
          DO 12 I=1,N
          AH(I)=CIN(I,1)*C2(1,1)
          DO 12 J=2,N
          AH(I)=AH(I)+CIN(I,J)*C2(J,1)
12        CONTINUE
C   SUPPRESS PRINTING OF COEFFICIENTS IF NOT REQUIRED
          WRITE(2,1050)
          READ(1,110)CAN
1050      FORMAT(1X,'ARE HOT BLOW COEFFTS. REQUIRED? YES OR NO--')
          IF(CAN.EQ.1HN) GO TO 1055
          WRITE(2,13)
13        FORMAT(1X,'COEFFICIENTS OF HOT BLOW--'/1X,28(1H=))
          WRITE(2,14)(AH(I),I=1,N)
14        FORMAT(3(2(1X,E12.5)))
1055      DO 15 I=1,N
          CIN(I,1)=BC(I,1)*US(1)
          DO 15 J=2,N
          CIN(I,1)=CIN(I,1)+BC(I,J)*US(J)
15        CONTINUE
          CALL MATRIX(20,N,N,N,BC,20,CH,20,C2,20)
          DO 16 I=1,N
          CIN(I,2)=C2(I,1)*AH(1)
          DO 16 J=2,N
          CIN(I,2)=CIN(I,2)+C2(I,J)*AH(J)
16        CONTINUE
          DO 17 I=1,N
17        AC(I)=CIN(I,1)+CIN(I,2)
          WRITE(2,1058)
1058      FORMAT(1X,'ARE COLD BLOW COEFFTS. REQUIRED? YES OR NO--')
          READ(1,110)CAN
          IF(CAN.EQ.1HN)GO TO 1060
          WRITE(2,18)
18        FORMAT(1X,'COEFFICIENTS OF COLD BLOW--'/1X,28(1H=))
          WRITE(2,14)(AC(I),I=1,N)
1060      WRITE(2,98)
98        FORMAT(1X,'ARE TEMPERATURE PROFILES FOR SOLID REQUIRED?',
          1'ENTER YES OR NO--')
          READ(1,110)ANS
110       FORMAT(A1)
          IF(ANS.EQ.1HN) GO TO 200
          WRITE(2,45)
45        FORMAT(1X,'ENTER NUMBER OF BED MESH POINTS--')
          READ(1,*)MB
          HM=1./(MB-1.)
          CM=1./(MB-1.)   ;   XH=0.   ;   XC=1.   ;   TH=AH(1)  ;   TC=AC(1)
          DO 47 I=2,N
47        TC=TC+AC(I)*XC**(I-1)
          WRITE(2,46)
46        FORMAT(1X,'SOLID TEMPERATURE PROFILE'/1X,26(1H=)/2X
          1'NORMALISED DISTANCE',10X,'HOT BLOW',10X,'COLD BLOW')
```

A2-3

```
        WRITE(2,48)XH,TH,TC
48      FORMAT(5X,F12.9,10X,F12.9,8X,F12.9)
        DO 49 J=2,MB
        XH=XH+HM
        XC=XC-CM;TH=AH(1);TC=AC(1)
        DO 50 I=2,N
        TH=TH+AH(I)*XH**(I-1)
50      TC=TC+AC(I)*XC**(I-1)
        WRITE(2,48)XH,TH,TC
49      CONTINUE
C  EXIT FLUID TEMPERATURES:
200     WRITE(2,1070)
        READ(1,110)CAN
1070    FORMAT(1X,'ARE EXIT FLUID TEMPS. REQUIRED? YES OR NO--')
        IF(CAN.EQ.1HN) GO TO 210
        CALL FLUTE(ALFR,VKR,ALFI,VKI,AH,AC,HLAM,CLAM,PIH,PIC,N)
210     WRITE(2,201)
201     FORMAT(1X,'INTERESTED IN THERMAL EFFICIENCY? YES OR NO:')
        READ(1,110)ANS
        IF(ANS.EQ.1HN) GO TO 999
        ETAREG=AC(1);SUM=AH(1)
        DO 90 I=2,N
        SUM=SUM+AH(I)/I
90      ETAREG=ETAREG+AC(I)/I
        ETAREG=(ETAREG-SUM)*HLAM/PIH
        WRITE(2,91)ETAREG
91      FORMAT(1X,'THERMAL EFFICIENCY=',F12.9)
999     WRITE(2,300)
300     FORMAT(1X,'ANOTHER RUN WITH DIFFERENT PARAMETERS?'/1X,
        1'ENTER YES OR NO--')
        READ(1,110)AND
        IF(AND.EQ.1HY)GO TO 600
        WRITE(2,301)
301     FORMAT(1X,'ANOTHER RUN WITH SAME PARAMS. BUT DIFFERENT ORDER?'/
        11X,'ENTER YES OR NO--')
        READ(1,110)ANS
        IF(ANS.EQ.1HN)STOP
        WRITE(2,1)
        READ(1,*)N
        GO TO 500
        END
C*****************************************************************************
        SUBROUTINE ZACOFS(ALFR,VKR,ALFI,VKI)
C  THIS SUB EVALUATES THE ZAKIAN'S COEFFICIENTS FOR PADE" APPROXIMATION.
        DOUBLE ALFR,VKR,ALFI,VKI,A,B
        DIMENSION ALFR(8),VKR(8),ALFI(7),VK(7)
        ALFR(1)=1.953965103383233D+01
        A=1.933570620860414D+01;B=3.379399888597234D+00
        ALFR(2)=A;ALFI(1)=B
        A=1.871433205030498D+01;B=6.772981645000657D+00
        ALFR(3)=A;ALFI(2)=B
        A=1.764452177737171D+01;B=1.019774391691488D+01
        ALFR(4)=A;ALFI(3)=B
        A=1.606503145933782D+01;B=1.367780303746658D+01
        ALFR(5)=A;ALFI(4)=B
        A=1.386207821875056D+01;B=1.725343258836830D+01
        ALFR(6)=A;ALFI(5)=B
        A=1.080652491390860D+01;B=2.100620730400388D+01
        ALFR(7)=A;ALFI(6)=B
        A=6.301979854806708D+00;B=2.516447268568806D+01
        ALFR(8)=A;ALFI(7)=B
        VKR(1)=1.645619599469101D+08
        A=-1.268572985368156D+08;B=-4.749121856114229D+07
        VKR(2)=A;VKI(1)=B
        A=5.574098488453668D+07;B=4.999812394225393D+07
        VKR(3)=A;VKI(2)=B
```

A2-4

```
        A=-1.136893282997024D+07;B=-2.450428931522289D+07
        VKR(4)=A;VKI(3)=B
        A=-1.694097331423655D+05;B=6.009306354368669D+06
        VKR(5)=A;VKI(4)=B
        A=4.138883037657413D+05;B=-6.184004276684025D+05
        VKR(6)=A;VKI(5)=B
        A=-4.058457858252957D+04;B=9.752029126666363D+03
        VKR(7)=A;VKI(6)=B
        A=3.800167535061704D+02;B=5.088313306242982D+02
        VKR(8)=A;VKI(7)=B
        RETURN
        END
C*****************************************************************
        INTEGER FUNCTION IFAC(I)
C  THIS FUNCTION EVALUATES THE FACTORIAL VALUE OF I.
        IFAC=1
        IF(I.LE.1)RETURN
        DO 1 J=2,I
1       IFAC=IFAC*J
        RETURN
        END
C**********************************************************
        DOUBLE FUNCTION ZAK(I,T,PI,ALFR,VKR,ALFI,VKI,ALAM)
        DOUBLE ALFR,VKR,ALFI,VKI,F1,F2,FR,FI,W1,W2,WR,WI
        DIMENSION ALFR(8),VKR(8),ALFI(7),VKI(7)
C  THIS FUNCTION EVALUATES THE INVERSE OF LAPLACED F AT TIME T USING
C   DOUBLE VECTORS ALFA AND VK WHICH ARE THE COEFFTS. OF ZAKIAN'S
C   FORMULA.
        IF(I.GT.0)GO TO 10
        ZAK=DBLE(T)*VKR(1)*(1.D0-DEXP(-ALFR(1)*DBLE(PI)/(ALFR(1)+DBLE(T*
        1ALAM)))/ALFR(1)
        DO 1 J=1,7
        J1=J+1
C  OBTAIN ARGUMENT OF EXPONENTIAL:
        F1=ALFR(J1)/DBLE(T);F2=ALFI(J)/DBLE(T)
        CALL ARITH(4,DBLE(PI)*F1,DBLE(PI)*F2,DBLE(ALAM)+F1,F2,FR,FI)
C  GET EXPONENTIAL TERM:
        CALL ARITH(6,-FR,-FI,F1,F2,WR,WI)
C  GET FINAL FUNCTION VALUES:
        CALL ARITH(4,1.D0-WR,-WI,F1,F2,FR,FI)
C  OBTAIN SUM IN ZAK FOR INVERSION:
        ZAK=ZAK+2.D0*(VKR(J1)*FR-VKI(J)*FI)
1       CONTINUE
        ZAK=ZAK/DBLE(T);RETURN
10      ZAK=(DBLE(T)**I)*VKR(1)*DEXP(-ALFR(1)*DBLE(PI)/ALFR(1)+DBLE(T*
        1ALAM)))/(ALFR(1)**I)
        DO 2 J=1,7
        J1=J+1
C  ARGUMENT OF EXPONENTIAL:
        F1=ALFR(J1)/DBLE(T);F2=ALFI(J)/DBLE(T)
        CALL ARITH(4,DBLE(PI)*F1,DBLE(PI)*F2,DBLE(ALAM)+F1,F2,FR,FI)
C  GET EXPONENTIAL
        CALL ARITH(6,-FR,-FI,F1,F2,WR,WI)
C  GET FUNCTION VALUES:
        CALL ARITH(5,F1,F2,DBLE(FLOAT(I)),0.D0,W1,W2)
        CALL ARITH(4,WR,WI,W1,W2,FR,FI)
C  OBTAIN SUM IN ZAK:
        ZAK=ZAK+2.D0*(VKR(J1)*FR-VKI(J)*FI)
2       CONTINUE
        ZAK=ZAK/DBLE(T)
        RETURN
        END
C ********************************************************************
        DOUBLE FUNCTION ZAP(T,PI,ALFR,VKR,ALFI,VKI,ALAM)
        DOUBLE ALFR,VKR,ALFI,VKI,F1,F2,FR,FI
        DIMENSION ALFR(8),VKR(8),ALFI(7),VKI(7)
```

A2-5

```
C  THIS FUNCTION EVALUATES THE HEAT POLE FUNCTION BY USE OF ZAKIAN'S
C   FORMULA OF LAPLACE INVERSION.
        ZAP=VKR(1)*DEXP(-ALFR(1)*DBLE(PI)/(ALFR(1)+DBLE(T*ALAM)))
        DO 1 J=1,7
        J1=J+1
        F1=ALFR(J1)/DBLE(T);F2=ALFI(J)/DBLE(T)
        CALL ARITH(4,DBLE(PI)*F1,DBLE(PI)*F2,DBLE(ALAM)+F1,F2,FR,FI)
        CALL ARITH(6,-FR,-FI,F1,F2,F1,F2)
        ZAP=ZAP+2.D0*(VKR(J1)*F1-VKI(J)*F2)
1       CONTINUE
        ZAP=ZAP/DBLE(T)
        RETURN
        END
C*******************************************************************
        FUNCTION TEMPO(AV,VLAM,N)
        DIMENSION AV(20)
C  THIS FUNCTION CALCULATES THE COMMON PART OF THE EXPRESSION FOR
C   EXIT FLUID TEMPERATURES FOR HOT OR COLD BLOW AT TIME  ZERO.
C   AV IS VECTOR OF COEFFTS.,VLAM IS THE APPROPRIATE LAMBDA.
C   FOR HOT BLOW NEED TO ADD EXPONENTIAL TERM IN THE REF.
C   BLOCK TO THE VALUE OF TEMPO.
C   EL=EXP(-LAM),VLI=1/LAM
        EL=EXP(-VLAM);VLI=1./VLAM
C  OBTAIN THE TERM OUTSIDE THE SUMMATION FROM 2 TO N FIRST,
C   STORING IN TEMPO.
        TEMPO=AV(1)*(1.-EL)
C  NOW OBTAIN THE LOOP SUM ADDING TO TEMPO,FINAL ANSWER IN
C   TEMPO. TEMPORARY VALUE OF SUMMATION ARGUMENT FOR CURRENT I
C   STORED IN TI*AV(I),I1F IS (I-1)FAC,SVLIM1=(-1/LAM)**(I-1)
        I1F=1;SVLIM1=1.
        DO 10 I=2,N
        I1=I-1;I1F=I1F*I1;SVLIM1=-SVLIM1*VLI
C  FIRST ADD TERMS OUTSIDE J LOOP IN TI:
        TI=1.-SVLIM1*I1F*EL
C  FOR THE J LOOP I1DI1JF=(I-1)FAC/(I-1-J)FAC,SVLIJ=(-1/LAM)**J
        SVLIJ=1.;I1DI1JF=1
        DO 20 J=1,I1
        I1DI1JF=(I-J)*I1DI1JF;SVLIJ=-SVLIJ*VLI
20      TI=TI+I1DI1JF*SVLIJ
C  MULTIPLY TI BY COEFFT.A(I) AND TOTAL UP IN TEMPO:
10      TEMPO=TEMPO+AV(I)*TI
        RETURN
        END
C*******************************************************************
        DOUBLE FUNCTION ZAFT(ALFR,VKR,ALFI,VKI,VLAM,ETA,I1,FL)
        DOUBLE ALFR,VKR,ALFI,VKI,F1,F2,FR,FI,W1,W2,WR,WI
        DIMENSION ALFR(8),VKR(8),ALFI(7),VKI(7)
C  THIS FN. EVALUATES THE INVERSE OF AN APPROPRIATE LAPLACED FN IN
C   CALCULATION OF FT1ETA(I.E.,EXIT FLUID TEMP. AT TIME ETA).
C   VLAM=LAMBDA(HOT OR COLD),I1 IS AN INDEX REPRESENTING THE POWER I-1
C   IN EXPRESSION ES=((S+1)/S)**(I-1).IF I1 IS ZERO,ES IS SET TO 1,IF
C   NOT I1>=1 IS USED IN ES.FL IS A FLAG INDICATING THE FORMULA TO BE US
C   ,IF FL NOT=1, WE USE FORMULA:ES*EXP(-LAM*S/(S+1))/S AND IF FL=1,
C   THE FORMULA FOR USE IN VARIABLE TIME INTEGRAL ETA* IS USED:
C   ES*EXP(-LAM*S/(S+1))/(S*S).
        IF(I1.GT.0)GO TO 50
C  CALCULATE FIRST TERM IN THE SUMMATION FOR ZAK FORMULA:
        ZAFT=VKR(1)*DEXP(-VLAM*ALFR(1)/(ALFR(1)+DBLE(ETA)))*DBLE(ETA)/
       1ALFR(1)
C  CALCULATING DIFFERENTLY FOR ETA*, SO GO TO 100 FOR THAT:
        IF(FL.EQ.1)GO TO 100
        DO 1 J=1,7
        J1=J+1
        F1=ALFR(J1)/DBLE(ETA);F2=ALFI(J)/DBLE(ETA)
C  ARGUMENT OF EXPONENTIAL:
        CALL ARITH(4,DBLE(VLAM)*ALFR(J1),DBLE(VLAM)*ALFI(J),ALFR(J1)+
```

A2-6

```
             1DBLE(ETA),ALFI(J),WR,WI)
C  GET EXPONENTIAL:
             CALL ARITH(6,-WR,-WI,F1,F2,W1,W2)
C  DIVIDE EXPL BY S:
             CALL ARITH(4,W1,W2,F1,F2,FR,FI)
C  SUM UP IN ZAFT:
             ZAFT=ZAFT+2.D0*(VKR(J1)*FR-VKI(J)*FI)
1            CONTINUE
             ZAFT=ZAFT/DBLE(ETA)
             RETURN
100          ZAFT=ZAFT*DBLE(ETA)/ALFR(1)
             DO 2 J=1,7
             J1=J+1
             F1=ALFR(J1)/DBLE(ETA);F2=ALFI(J)/DBLE(ETA)
C  ARGUMENT OF EXPONENTIAL:
             CALL ARITH(4,DBLE(VLAM)*ALFR(J1),DBLE(VLAM)*ALFI(J),ALFR(J1)
             1+DBLE(ETA),ALFI(J),WR,WI)
C  GET EXPONENTIAL:
             CALL ARITH(6,-WR,-WI,F1,F2,W1,W2)
C  GET THE DENOMINATOR:
             CALL ARITH(5,F1,F2,2.D0,0.D0,WR,WI)
C  DIVIDE OUT:
             CALL ARITH(4,W1,W2,WR,WI,FR,FI)
C  SUM UP:
             ZAFT=ZAFT+2.D0*(VKR(J1)*FR-VKI(J)*FI)
2            CONTINUE
             ZAFT=ZAFT/DBLE(ETA)
             RETURN
50           W1=ALFR(1)/(ALFR(1)+DBLE(ETA))
             ZAFT=DBLE(ETA)*VKR(1)*DEXP(-DBLE(VLAM)*W1)*(1.D0/W1)**I1,ALFR(1)
             IF(FL.EQ.1)GO TO 150
             DO 3 J=1,7
             J1=J+1
             F1=ALFR(J1)/DBLE(ETA);F2=ALFI(J)/DBLE(ETA)
C  EXPONENTIAL:
             CALL ARITH(4,ALFR(J1)*DBLE(VLAM),ALFI(J)*DBLE(VLAM),DBLE(ETA)+
             1ALFR(J1),ALFI(J),WR,WI)
             CALL ARITH(6,-WR,-WI,F1,F2,W1,W2)
C  DIVIDE BY S:
             CALL ARITH(4,W1,W2,F1,F2,WR,WI)
C  GET ((S+1)/S)**I1:
             CALL ARITH(4,F1+1.D0,F2,F1,F2,W1,W2)
             CALL ARITH(5,W1,W2,DBLE(FLOAT(I1)),0.D0,F1,F2)
C  MULTIPLY TO GET FUNCTION VALUES:
             CALL ARITH(3,WR,WI,F1,F2,FR,FI)
C SUM UP:
             ZAFT=ZAFT+2.D0*(VKR(J1)*FR-VKI(J)*FI)
3            CONTINUE
             ZAFT=ZAFT/DBLE(ETA)
             RETURN
150          ZAFT=ZAFT*DBLE(ETA)/ALFR(1)
             DO 4 J=1,7
             J1=J+1
             F1=ALFR(J1)/DBLE(ETA);F2=ALFI(J)/DBLE(ETA)
C  EXPONENTIAL:
             CALL ARITH(4,DBLE(VLAM)*ALFR(J1),DBLE(VLAM)*ALFI(J),ALFR(J1)+
             1DBLE(ETA),ALFI(J),FR,FI)
             CALL ARITH(6,-FR,-FI,F1,F2,W1,W2)
C  DENOMINATOR:
             CALL ARITH(5,F1,F2,2.D0,0.D0,FR,FI)
C  DIVIDE:
             CALL ARITH(4,W1,W2,FR,FI,WR,WI)
C  GET ((S+1)/S)**I1:
             CALL ARITH(4,F1+1.D0,F2,F1,F2,W1,W2)
             CALL ARITH(5,W1,W2,DBLE(FLOAT(I1)),0.D0,F1,F2)
```

A2-7

```
      CALL ARITH(3,WR,WI,F1,F2,FR,FI)
C SUM UP:
      ZAFT=ZAFT+2.D0*(VKR(J1)*FR-VKI(J)*FI)
4     CONTINUE
      ZAFT=ZAFT/DBLE(ETA)
      RETURN
      END
C**********************************************************************
      SUBROUTINE ARITH(KODE,AR,AI,BR,BI,CR,CI)
C THIS IS A COMPLEX ARITHMETIC PACKAGE:
C  KODE--1:A+B, 2:A-B, 3:A*B, 4:A/B, 5:A**B(REAL), 6:EXP(A)
C   RESULT STORED IN CR(REAL PART) AND CI(IMAGINARY).
      DOUBLE AR,AI,BR,BI,CR,CI,R,THETA
      GOTO(10,20,30,40,50,60)KODE
10    CR=AR+BR;CI=AI+BI
      RETURN
20    CR=AR-BR;CI=AI-BI
      RETURN
30    CR=AR*BR-AI*BI;CI=AR*BI+AI*BR
      RETURN
40    CR=(AR*BR+AI*BI)/(BR*BR+BI*BI)
      CI=(AI*BR-AR*BI)/(BR*BR+BI*BI)
      RETURN
50    R=(AR*AR+AI*AI)**(BR/2.D0);THETA=BR*DATAN2(AI,AR)
      CR=R*DCOS(THETA);CI=R*DSIN(THETA)
      RETURN
60    CR=DEXP(AR)*DCOS(AI);CI=DEXP(AR)*DSIN(AI)
      RETURN;END
C**********************************************************************
      SUBROUTINE GT(VLI,ETA,I,TES)
C THIS SUBROUTINE CALCULATES THAT PART OF THE SUMMATION ARGUMEN
C  FOR EXIT FLUID TEMP. WHICH DOES NOT INCLUDE THE USE OF LAP.INVN.
C  VLI=1/VLAM BROUGHT OVER FROM CALLING BLOCK,I IS THE INDEX NO.
C  I USED IN SUMMATION FROM I=2 TO N IN BLOCK,I1=I-1.
C  THE VALUE FOR THIS CALCULATION IS SAVED IN TES.
C  I1DI1JF=(I-1)FAC/(I-1-J)FAC,SVLIJ=(-1/LAM)**J IN J LOOP.
      I1=I-1;TES=1.;I1DI1JF=1;SVLIJ=1.
      DO 10 J=1,I1
      I1DI1JF=I1DI1JF*(I-J);SVLIJ=-SVLIJ*VLI
C SUM OF K LOOP IS STORED IN TK,HERE JMKM1=J(J-1)...(J-K+1)
C ,KF=(K)FAC,JKBC=BINOMIAL COEFFICIENT OF J AND K,AND ETAK=ETA**K.
      TK=1.;JMKM1=1;KF=1;ETAK=1.
      DO 20 K=1,J
      JMKM1=JMKM1*(J-K+1);KF=KF*K;ETAK=ETAK*ETA;JKBC=JMKM1/KF
20    TK=TK+JKBC*ETAK/KF
C NOW TOTAL UP IN TES:
      TES=TES+I1DI1JF*SVLIJ*TK
10    CONTINUE
      RETURN;END
C**********************************************************************
      FUNCTION FT1ETA(AV,ETA,N,VLAM,FLAG,ALFR,VKR,ALFI,VKI)
      DOUBLE ALFR,VKR,ALFI,VKI,ZAFT
      DIMENSION AV(20),ALFR(8),VKR(8),ALFI(7),VKI(7)
C THIS FN EVALUATES THE FLUID EXIT TEMP FOR HOT OR COLD BLOW AT TIME
C  ETA.AV IS VECTOR OF COEFFTS OBTAINED EARLIER,VLAM IS LAMBDA,FLAG
C  IS AN INDICATOR TOWARDS THE TYPE OF FLUID TEMP TO BE CALCULATED.
C  IF FLAG=1. WE ARE DEALING WITH A HOT BLOW AND FOR COLD BLOW FLAG
C  SHOULD BE SET TO SOMETHING NOT EQUAL TO 1.(SAY 0.).
C
C LAPLACE INVERSION USING ZAKIAN'S FORMULA FOR THE FIRST TERM:
      FL1=ZAFT(ALFR,VKR,ALFI,VKI,VLAM,ETA,0,0.)
C IF FLAG=1.USE (1-AH(1))*FL1 OTHERWISE OBTAIN -AC(1)*FL1,SAVE AS FT1ET
      FT1ETA=AV(1)*(1.-FL1)
      IF(FLAG.NE.1.)GO TO 10
      FT1ETA=FL1+FT1ETA
```

A2-8

```
C  A(I)*(1+(I-1)FAC*SUM OVER J=1 TO (I-1) OF (-1/LAM)**J/(I-1-J)FAC
C  *SUM OVER K=0 TO J OF JKBC*ETA**K/KFAC-(-1/LAM)**(I-1)FAC
C  *LAP.INV.(EXP(-LA*S/(S+1))/S*((S+1)/S)**(I-1)).
C  VLI=1/VLAM,SVLIM1=(-1/LAM)**(I-1),I1F=(I-1)FAC IN I LOOP.
10      VLI=1./VLAM;SVLIM1=1.;I1F=1
        DO 20 I=2,N
        I1=I-1;I1F=I1F*I1;SVLIM1=-SVLIM1*VLI
C  OBTAIN NON LAP. TERM STORED IN TES:
        CALL GT(VLI,ETA,I,TES)
C  INVERT L.T. EVALUATING AT ETA:
        FL1=ZAFT(ALFR,VKR,ALFI,VKI,VLAM,ETA,I1,0.)
C    OBTAIN THE ARGUMENT FOR SUMMATION, MULTIPLY BY A(I) AND
C  TOTAL UP IN FT1ETA:
        FT1ETA=FT1ETA+AV(I)*(TES-SVLIM1*I1F*FL1)
20      CONTINUE
        RETURN;END
C***********************************************************************
        FUNCTION ETASTAR(ALFR,VKR,ALFI,VKI,ETA,CLAM,N,AC)
        DOUBLE ALFR,VKR,ALFI,VKI,ZAFT
        DIMENSION AC(N),ALFR(8),VKR(8),ALFI(7),VKI(7)
C  THIS FN CALCULATES ETASTAR=INTEGRAL OVER ZERO TO ETA OF COLD BLOW
C   EXIT FLUID TEMP FOR ANY ETA FROM 0. TO PIC.ETASTAR=0 IF ETA=0.
        IF(ETA.NE.0.)GO TO 10
        ETASTAR=0.;RETURN
CALCULATE THE FIRST TWO TERMS I.E., THE ONES OUTSIDE THE SERIES AND
C  STORE IN ETASTAR:
10      ETASTAR=AC(1)*(ETA-ZAFT(ALFR,VKR,ALFI,VKI,CLAM,ETA,0,1.))
C  NOW OBTAIN THE REST AND ADD TO ETASTAR TO GET THE FINAL ANSWER USING
C  THE I LOOP IN WHICH TIS STORES THE NON LAPLACE TERM FROM THE
C  ARGUMENT OF SUMMATION (WHICH WILL HAVE TO BE MULTIPLIED BY AC(I) AT I
C  I1F=(I-1)FAC,SVLIM1=(-1/CLAM)**(I-1)
        SVLIM1=1.;I1F=1;CLI=1./CLAM
        DO 20 I=2,N
        TIS=ETA;I1=I-1;I1F=I1F*I1;SVLIM1=-SVLIM1*CLI
C  FOR J LOOP I1DI1JF=(I-1)FAC/(I-1-J)FAC,SVLIJ=(-1/CLAM)**J
        I1DI1JF=1;SVLIJ=1.
        DO 30 J=1,I1
        I1DI1JF=I1DI1JF*(I-J);SVLIJ=-SVLIJ*CLI
C  SUM OF K LOOP IS STORED IN TKS, HERE JMKM1=J(J-1)...(J-K+1),
C  KF=KFAC,K1F=(K+1)FAC,JKBC=BIN.COEFFT. OF J AND K, ETAK1=ETA**K1.
        TKS=ETA;JMKM1=1;KF=1;ETAK1=ETA
        DO 40 K=1,J
        JMKM1=JMKM1*(J-K+1);KF=KF*K;ETAK1=ETAK1*ETA
        JKBC=JMKM1/KF;K1F=(K+1)*KF
40      TKS=TKS+JKBC*ETAK1/K1F
C  TOTAL UP IN TIS THE NON LAP. TERM:
30      TIS=TIS+TKS*I1DI1JF*SVLIJ
C  NOW OBTAIN L.I. TERM AND SUBTRACT FROM TIS, AND GET THE FINAL
C  ANSWER AFTER MULTIPLYING BY A(I) AND SUMMING:
        FL1=ZAFT(ALFR,VKR,ALFI,VKI,CLAM,ETA,I1,1.)
        ETASTAR=ETASTAR+AC(I)*(TIS-SVLIM1*I1F*FL1)
20      CONTINUE
        RETURN;END
C***********************************************************************
        SUBROUTINE FLUTE(ALFR,VKR,ALFI,VKI,AH,AC,HLAM,CLAM,PIH,PIC,N)
        DOUBLE ALFR,VKR,ALFI,VKI
        DIMENSION ALFR(8),VKR(8),ALFI(7),VKI(7),AH(N),AC(N)
C  THIS SUB FINDS THE EXIT FLUID TEMPERATURES AT SPECIFIED POINTS OF
C  THE HOT OR COLD PERIOD.THESE POINTS ARE EQUALLY SPACED ACCORDING TO
C  THE MESH SPACING DESIRED.
        WRITE(2,10)
        READ(1,*)NPH,NPC
10      FORMAT(1X,'ENTER TOTAL  NUMBERS OF POINTS(I.E.,INTERIOR+2 END
        1POINTS)OF HOT AND THEN COLD PERIOD AT WHICH PROFILE REQUIRED-')
C  HOT BLOW FLUID EXIT PROFILE, FIRST OBTAIN TEMP. AT TIME ZERO THEN IN
```

A2-9

```
        WRITE(2,11)
11      FORMAT(1X,'HOT FLUID EXIT TEMPERATURE PROFILE:'/1X,35(1H=)/6X,
        1'TIME',10X,'TEMPERATURE')
        EH=0.;EC=0.
C  HOT TEMP AT TIME ZERO:
        TH10=EXP(-HLAM)+TEMPO(AH,HLAM,N)
        WRITE(2,12)EH,TH10
12      FORMT(5X,F12.9,2X,F12.9)
C  HOT TEMPS AT OTHER TIMES:
        HN=PIH/(NPH-1)
        DO 20 K=2,NPH
        EH=EH+HN
        TH10=FT1ETA(AH,EH,N,HLAM,1.,ALFR,VKR,ALFI,VKI)
        WRITE(2,12)EH,TH10
20      CONTINUE
C  COLD BLOW FLUID PROFILE:
        WRITE(2,21)
21      FORMAT(1X,'COLD FLUID EXIT TEMPERATURE PROFILE:'/1X,36(1H=)/6X,
        1'TIME',10X,'TEMPERATURE')
COLD TEMPERATURE AT TIME ZERO:
        TC10=TEMPO(AC,CLAM,N)
        WRITE(2,12)EC,TC10
COLD TEMPS AT OTHER TIMES:
        CN=PIC/(NPC-1)
        DO 30 K=2,NPC
        EC=EC+CN
        TC10=FT1ETA(AC,EC,N,CLAM,0.,ALFR,VKR,ALFI,VKI)
        WRITE(2,12)EC,TC10
30      CONTINUE
C  VARIABLE TIME INTEGRAL (FOR COLD BLOW ONLY) ETASTAR:
        WRITE(2,22)
        READ(1,110)CAN
22      FORMAT(1X,'IS VARIABLE TIME INTEGRAL OF COLD EXIT FLUID TEMP.
        1REQUIRED?YES OR NO--')
110     FORMAT(A1)
        IF(CAN.EQ.1HN)RETURN
        WRITE(2,25)
25      FORMAT(1X,'VARIABLE TIME COLD FLUID PROFILE:'/1X,33(1H=)/6X,
        1'TIME',10X,'ETASTAR INTEGRAL')
        EC=0.;ETS=0.
        WRITE(2,40)EC,ETS
40      FORMAT(5X,F12.9,2X,F15.9)
        DO 50 K=2,NPC
        EC=EC+CN
        ETS=ETASTAR(ALFR,VKR,ALFI,VKI,EC,CLAM,N,AC)
        WRITE(2,40)EC,ETS
50      CONTINUE
        RETURN;END
C*******************************************************************************
```

A2-10

```
.,RUN,F,F=TZ1
ENTER VALUE OF N--
   45000B   CM STORAGE USED
      3.872 CP SECONDS COMPILATION TIME
   CM LWA+1 = 37171B, LOADER USED   53100B5

ENTER HOT AND THEN COLD LAMBDA--19.7354,17.3553

ENTER HOT AND THEN COLD PERIOD--8.5,8.5

TZ=  .10000E+01
DET=  .27466E-03
DET=  .94706E-04
ARE HOT BLOW COEFFTS.REQUIRED?YES OR NO--Y

COEFFICIENTS OF  HOT BLOW--
============================
  .64561E+00  -.10225E+01  -.30581E+00   .10243E+01  -.34162E+00
ARE COLD BLOW COEFFTS.REQUIRED?YES OR NO--Y

COEFFICIENTS OF COLD BLOW--
============================
  .15052E+00   .61473E+00   .64215E+00   .65593E+00  -.10634E+01
ARE TEMPERATURE PROFILES FOR SOLID REQUIRED?ENTER YES OR NO--Y

ENTER NUMBER OF BED MESH POINTS--31

 SOLID TEMPERATURE PROFILE
==========================
 NORMALISED DISTANCE            HOT BLOW          COLD BLOW
     0.000000000               .645607319         .999927892
      .033333333               .611223009        1.008762998
      .066666667               .576380847        1.010000403
      .100000000               .541293275        1.004392328
      .133333333               .506162613         .992659486
      .166666667               .471181059         .975491083
      .200000000               .436530688         .953544815
      .233333333               .402383455         .927446871
      .266666667               .368901189         .897791932
      .300000000               .336235602         .865143169
      .333333333               .304528279         .830032247
      .366666667               .273910687         .792959321
      .400000000               .244504167         .754393038
      .433333333               .216419942         .714770539
      .466666667               .189759110         .674497453
      .500000000               .164612648         .633947905
      .533333333               .141061410         .593464508
      .566666667               .119176128         .553358368
      .600000000               .099017414         .513909084
      .633333333               .080635756         .475364746
      .666666667               .064071520         .437941936
      .700000000               .049354949         .401825726
      .733333333               .036506167         .367169683
      .766666667               .025535172         .334095862
      .800000000               .016441844         .302694813
      .833333333               .009215936         .273025577
      .866666667               .003837084         .245115685
      .900000000               .000274799         .218961161
      .933333333              -.001511531         .194526523
      .966666667              -.001572637         .171744776
     1.000000000               .000030626         .150517421
ARE EXIT FLUID TEMPS.REQUIRED?YES OR NO--Y

ENTER TOTAL NUMBERS OF POINTS(I.E.,INTERIOR + 2 END    POINTS)OF
HOT AND THEN COLD PERIOD AT WHICH PROFILE REQUIRED-31,31
```

A2-11

```
HOT FLUID EXIT TEMPERATURE PROFILE:
====================================
     TIME              TEMPERATURE
     0.000000000        .000260060
      .283333333        .001641441
      .566666667        .003357839
      .850000000        .005409943
     1.133333333        .007798136
     1.416666667        .010522510
     1.700000000        .013582892
     1.983333333        .016978871
     2.266666667        .020709819
     2.550000000        .024774933
     2.833333333        .029173261
     3.116666667        .033903740
     3.400000000        .038965226
     3.683333333        .044356531
     3.966666667        .050076447
     4.250000000        .056123776
     4.533333333        .062497345
     4.816666667        .069196023
     5.100000000        .076218724
     5.383333333        .083564406
     5.666666667        .091232056
     5.950000000        .099220674
     6.233333333        .107529240
     6.516666667        .116156674
     6.800000000        .125101792
     7.083333333        .134363246
     7.366666667        .143939463
     7.650000000        .153828577
     7.933333333        .164028357
     8.216666667        .174536128
     8.500000000        .185348697
COLD FLUID EXIT TEMPERATURE PROFILE:
=====================================
     TIME              TEMPERATURE
     0.000000000       1.001025980
      .283333333        .995780329
      .566666667        .989451013
      .850000000        .982103531
     1.133333333        .973801504
     1.416666667        .964606658
     1.700000000        .954578800
     1.983333333        .943775807
     2.266666667        .932253605
     2.550000000        .920066154
     2.833333333        .907265443
     3.116666667        .893901484
     3.400000000        .880022311
     3.683333333        .865673995
     3.966666667        .850900656
     4.250000000        .835744490
     4.533333333        .820245801
     4.816666667        .804443042
     5.100000000        .788372864
     5.383333333        .772070172
     5.666666667        .755568188
     5.950000000        .738898517
     6.233333333        .722091222
     6.516666667        .705174896
     6.800000000        .688176745
     7.083333333        .671122661
     7.366666667        .654037305
     7.650000000        .636944182
```

A2-12

```
     8.216666667     .602823306
     8.500000000     .585837429
IS VARIABLE TIME INTEGRAL OF COLD EXIT FLUID TEMP.         REQUIRED?YES OR NO--Y

VARIABLE TIME COLD FLUID PROFILE:
==================================
     TIME              ETASTAR INTEGRAL
     0.000000000          0.000000000
      .283333333           .282907267
      .566666667           .564173179
      .850000000           .843500020
     1.133333333          1.120608368
     1.416666667          1.395236557
     1.700000000          1.667140140
     1.983333333          1.936091343
     2.266666667          2.201878512
     2.550000000          2.464305563
     2.833333333          2.723191420
     3.116666667          2.978369458
     3.400000000          3.229686941
     3.683333333          3.477004465
     3.966666667          3.720195403
     4.250000000          3.959145354
     4.533333333          4.193751604
     4.816666667          4.423922596
     5.100000000          4.649577410
     5.383333333          4.870645264
     5.666666667          5.087065024
     5.950000000          5.298784738
     6.233333333          5.505761191
     6.516666667          5.707959471
     6.800000000          5.905352575
     7.083333333          6.097921015
     7.366666667          6.285652466
     7.650000000          6.468541424
     7.933333333          6.646588889
     8.216666667          6.819802070
     8.500000000          6.988194105
INTERESTED IN THERMAL EFFICIENCY? YES OR NO: Y

THERMAL EFFICIENCY=  .935782041
ANOTHER RUN WITH DIFFERENT PARAMETERS?
ENTER YES OR NO --N

ANOTHER RUN WITH SAME PARAMS. BUT DIFFERENT ORDER?
ENTER YES OR NO--N

     STOP
     14.268 CP SECONDS EXECUTION TIME
```

APPENDIX A3

Attached to this appendix is a FORTRAN program again named CLOZAK which is the single precision form of the program in A2. Here $z \varepsilon [0,1]$ scale is used but arithmetic is all single precision. A typical output is also attached. The sequence of events is exactly the same here as that in Appendix A2, however TZ value is not calculated here. So the parameters input here are exactly the same as in A2 but $\lambda' = 19.7254$ instead of 19.7354. The results obtained are quite compatible with those obtained in A1 (agreement in general at least up to sixth decimal place). Obviously here normalised distance $z \varepsilon [0,1]$ is utilised in printing solid temperature profile which is certainly a more representative presentation than the case in A1 where the normalised distance $q \varepsilon [0, \lambda']$ represents the distance values for both hot and cold blows especially when cold blow $\lambda'' \neq \lambda'$. The results obtained in A3 are only slightly different from those in A2 mainly because of slight difference in λ' value between the two cases as noted earlier. However here execution time is only 3.705 CP seconds which makes the program in this Appendix A3 much faster than that in A2 (as there the value is 14.268 CP seconds for 'virtually' the same parameters). This can only be explained in terms of additional time taken in double precision arithmetic and calls made to subroutine ARITH in Appendix A2 in order to achieve double precision complex results.

We conclude that if fast runs are required the program in this appendix is adequate enough but if more precision is the aim,program in Appendix A2 should be used.

...

A3-1

```
        PROGRAM CLOZAK(INPUT,OUTPUT,XYDATA,TAPE1=INPUT,TAPE2=OUTPUT,
        1TAPE3=XYDATA)
        COMPLEX ALFA,VK
        DIMENSION X1(20),X2(20),US(20),AH(20),AC(20),CC(20,20),
        1CH(20,20),BC(20,20),BH(20,20),C1(20,20),C2(20,20),CIN(20,20),
        2ALFA(15),VK(15)
C   THIS PROGRAM SOLVES FOR VECTORS AH AND AC WHICH ARE COEFFICIENTS
C    OF THE FINITE SERIES APPROXIMATION FOR INITIAL HOT AND COLD
C    BLOWS. GENERAL UNBALANCED CASE USING 0 TO 1 SCALE.
600     WRITE(2,1)
1       FORMAT(1X,'ENTER VALUE OF N--')
        READ(1,*)N
        WRITE(2,2)
2       FORMAT(1X,'ENTER HOT AND THEN COLD LAMBDA--')
        READ(1,*)HLAM,CLAM
        WRITE(2,3)
3       FORMAT(1X,'ENTER HOT AND THEN COLD PERIOD--')
        READ(1,*)PIH,PIC
500     N1=N-1
C   SET UP VECTORS OF DISTANCE COORDINATES AND SECOND COL.OF B MATS.
        X1(1)=0.  ;  X2(1)=1.  ;  X1(N)=1.
        X1(2)=1./N1  ;  X2(2)=1.-X1(2)  ;  X2(N)=0.
        DO 4 I=3,N1
        X1(I)=(I-1)*X1(2)
4       X2(I)=1.-X1(I)
        DO 5 I=1,N
        BH(I,2)=X1(I)
        BC(I,2)=X2(I)
C   FACTORIAL VALUES IN A VECTOR
5       AH(I)=IFAC(I-1)
        DO 40 J=3,N
        J1=J-1
        DO 40 I=1,N
C   FILL REMAINDER OF B MATRICES
        BH(I,1)=BC(I,1)=1.
        BH(I,J)=BH(I,J1)*X1(I)
        BC(I,J)=BC(I,J1)*X2(I)
40       CONTINUE
C   OBTAIN ZAKIAN'S COEFFICIENTS
        CALL ZACOFS(ALFA,VK)
C   OBTAIN STEP RESPONSE VECTOR
        US(1)=1.-EXP(-PIH)
        DO 6 J=2,N
6       US(J)=ZAK(0,X1(J),PIH,ALFA,VK,HLAM)
        DO 7 I=1,N1
        I1=I+1
        DO 7 J=1,N
C   OBTAIN BOTH HEAT POLE MATRICES, BC AND BH
        CH(I1,J)=ZAK(J,X1(I1),PIH,ALFA,VK,HLAM)*AH(J)
        CC(I,J)=ZAK(J,X2(I),PIC,ALFA,VK,CLAM)*AH(J)
7       CONTINUE
        DO 20 J=2,N
20      CH(1,J)=CC(N,J)=0.
        CH(1,1)=EXP(-PIH) ; CC(N,1)=EXP(-PIC)
C       INVERT BC AND PRINT DETERMINANT
        CALL MATRIX(10,N,N,0,BC,20,DET)
        WRITE(2,30)DET
30      FORMAT(1X,'DET=',E12.5)
C   FIND CC*INV(BC) AND STORE IN C1 MATRIX
        CALL MATRIX(20,N,N,N,CC,20,BC,20,C1,20)
C   FIND C2=CC*INV(BC)*CH
        CALL MATRIX(20,N,N,N,C1,20,CH,20,C2,20)
C   FIND CIN=BH-C2
        DO 10 I=1,N
        DO 10 J=1,N
```

A3-2

```
C  INVERT CIN AND PRINT DETERMINANT
         CALL MATRIX(10,N,N,0,CIN,20,DET)
         WRITE(2,30)DET
         DO 11 I=1,N
         C2(I,1)=C1(I,1)*US(1)
         DO 11 J=2,N
         C2(I,1)=C2(I,1)+C1(I,J)*US(J)
11       CONTINUE
         DO 12 I=1,N
         AH(I)=CIN(I,1)*C2(1,1)
         DO 12 J=2,N
         AH(I)=AH(I)+CIN(I,J)*C2(J,1)
12       CONTINUE
C  SUPPRESS PRINTING OF COEFFICIENTS IF NOT REQUIRED
         WRITE(2,1050)
         READ(1,110)CAN
1050     FORMAT(1X,'ARE HOT BLOW COEFFTS. REQUIRED? YES OR NO--')
         IF(CAN.EQ.1HN) GO TO 1055
         WRITE(2,13)
13       FORMAT(1X,'COEFFICIENTS OF HOT BLOW--'/1X,28(1H=))
         WRITE(2,14)(AH(I),I=1,N)
14       FORMAT(3(2(1X,E12.5)))
1055     DO 15 I=1,N
         CIN(I,1)=BC(I,1)*US(1)
         DO 15 J=2,N
         CIN(I,1)=CIN(I,1)+BC(I,J)*US(J)
15       CONTINUE
         CALL MATRIX(20,N,N,N,BC,20,CH,20,C2,20)
         DO 16 I=1,N
         CIN(I,2)=C2(I,1)*AH(1)
         DO 16 J=2,N
         CIN(I,2)=CIN(I,2)+C2(I,J)*AH(J)
16       CONTINUE
         DO 17 I=1,N
17       AC(I)=CIN(I,1)+CIN(I,2)
         WRITE(2,1058)
1058     FORMAT(1X,'ARE COLD BLOW COEFFTS. REQUIRED? YES OR NO--')
         READ(1,110)CAN
         IF(CAN.EQ.1HN)GO TO 1060
         WRITE(2,18)
18       FORMAT(1X,'COEFFICIENTS OF COLD BLOW--'/1X,28(1H=))
         WRITE(2,14)(AC(I),I=1,N)
1060     WRITE(2,98)
98       FORMAT(1X,'ARE TEMPERATURE PROFILES FOR SOLID REQUIRED?',
         1'ENTER YES OR NO--')
         READ(1,110)ANS
110      FORMAT(A1)
         IF(ANS.EQ.1HN) GO TO 200
         WRITE(2,45)
45       FORMAT(1X,'ENTER NUMBER OF BED MESH POINTS--')
         READ(1,*)MB
         HM=1./(MB-1.)
         CM=1./(MB-1.)  ;  XH=0.  ;  XC=1.  ;  TH=AH(1) ;  TC=AC(1)
         DO 47 I=2,N
47       TC=TC+AC(I)*XC**(I-1)
         WRITE(2,46)
46       FORMAT(1X,'SOLID TEMPERATURE PROFILE'/1X,26(1H=)/2X
         1,'NORMALISED DISTANCE',10X,'HOT BLOW',10X,'COLD BLOW')
         WRITE(2,48)XH,TH,TC
48       FORMAT(5X,F12.9,10X,F12.9,8X,F12.9)
         DO 49 J=2,MB
         XH=XH+HM
         XC=XC-CM;TH=AH(1);TC=AC(1)
         DO 50 I=2,N
         TH=TH+AH(I)*XH**(I-1)
```

A3-3

```
            WRITE(2,48)XH,TH,TC
49          CONTINUE
C   EXIT FLUID TEMPERATURES:
200         WRITE(2,1070)
            READ(1,110)CAN
1070        FORMAT(1X,'ARE EXIT FLUID TEMPS. REQUIRED? YES OR NO--')
            IF(CAN.EQ.1HN) GO TO 210
            CALL FLUTE(ALFA,VK,AH,AC,HLAM,CLAM,PIH,PIC,N)
210         WRITE(2,201)
201         FORMAT(1X,'INTERESTED IN THERMAL EFFICIENCY? YES OR NO:')
            READ(1,110)ANS
            IF(ANS.EQ.1HN) GO TO 999
            ETAREG=AC(1);SUM=AH(1)
            DO 90 I=2,N
            SUM=SUM+AH(I)/I
90          ETAREG=ETAREG+AC(I)/I
            ETAREG=(ETAREG-SUM)*HLAM/PIH
            WRITE(2,91)ETAREG
91          FORMAT(1X,'THERMAL EFFICIENCY=',F12.9)
999         WRITE(2,300)
300         FORMAT(1X,'ANOTHER RUN WITH DIFFERENT PARAMETERS?'/1X,
            1'ENTER YES OR NO--')
            READ(1,110)AND
            IF(AND.EQ.1HY)GO TO 600
            WRITE(2,301)
301         FORMAT(1X,'ANOTHER RUN WITH SAME PARAMS. BUT DIFFERENT ORDER?'/
            11X,'ENTER YES OR NO--')
            READ(1,110)ANS
            IF(ANS.EQ.1HN)STOP
            WRITE(2,1)
            READ(1,*)N
            GO TO 500
            END
C*****************************************************************************
            SUBROUTINE ZACOFS(ALFA,VK)
C   THIS SUB EVALUATES THE ZAKIAN'S COEFFICIENTS FOR PADE" APPROXIMATION.
         COMPLEX ALFA,VK
         DIMENSION ALFA(15),VK(15)
         ALFA(1)=CMPLX(1.953965103383233E+01,0.)
         A=1.933570620860414E+01 $ B=3.379399888597234E+00
         ALFA(2)=CMPLX(A,B)$ALFA(3)=CMPLX(A,-B)
         A=1.871433205030498E+01 $ B=6.772981645000657E+00
         ALFA(4)=CMPLX(A,B)$ALFA(5)=CMPLX(A,-B)
         A=1.764452177737171E+01 $ B=1.019774391691488E+01
         ALFA(6)=CMPLX(A,B)$ALFA(7)=CMPLX(A,-B)
         A=1.606503145933782E+01 $ B=1.367780303746658E+01
         ALFA(8)=CMPLX(A,B)$ALFA(9)=CMPLX(A,-B)
         A=1.386207821875056E+01 $ B=1.725343258836830E+01
         ALFA(10)=CMPLX(A,B)$ALFA(11)=CMPLX(A,-B)
         A=1.080652491390860E+01 $ B=2.100620730400388E+01
         ALFA(12)=CMPLX(A,B)$ALFA(13)=CMPLX(A,-B)
         A=6.301979854806708E+00 $ B=2.516447268568806E+01
         ALFA(14)=CMPLX(A,B)$ALFA(15)=CMPLX(A,-B)
         VK(1)=CMPLX(1.645619599469101E+08,0.)
         A=-1.268572985368156E+08 $ B=-4.749121856114229E+07
         VK(2)=CMPLX(A,B)$VK(3)=CMPLX(A,-B)
         A=5.574098488453668E+07 $ B=4.999812394225393E+07
         VK(4)=CMPLX(A,B)$VK(5)=CMPLX(A,-B)
         A=-1.136893282997024E+07 $ B=-2.450428931522289E+07
         VK(6)=CMPLX(A,B)$VK(7)=CMPLX(A,-B)
         A=-1.694097331423655E+05 $ B=6.009306354368669E+06
         VK(8)=CMPLX(A,B)$VK(9)=CMPLX(A,-B)
         A=4.138883037657413E+05 $ B=-6.184004276684025E+05
         VK(10)=CMPLX(A,B)$VK(11)=CMPLX(A,-B)
         A=-4.058457858252957E+04 $ B=9.752029126666363E+03
```

```
        A=3.800167535061704E+02 $ B=5.088313306242982E+02
        VK(14)=CMPLX(A,B)$VK(15)=CMPLX(A,-B)
          RETURN
          END
C*****************************************************************
          INTEGER FUNCTION IFAC(I)
C  THIS FUNCTION EVALUATES THE FACTORIAL VALUE OF I.
          IFAC=1
          IF(I.LE.1)RETURN
          DO 1 J=2,I
1         IFAC=IFAC*J
          RETURN
          END
C**********************************************************
          COMPLEX FUNCTION ZAK(I,T,PI,ALFA,VK,ALAM)
          COMPLEX ALFA,VK,F1,F2,ZAC
          DIMENSION ALFA(15),VK(15)
C  THIS FUNCTION EVALUATES THE INVERSE OF LAPLACED F AT TIME T USING
C   COMPLEX VECTORS ALFA AND VK WHICH ARE THE COEFFTS. OF ZAKIAN'S
C   FORMULA.
          IF(I.GT.0)GO TO 10
        ZAC=CMPLX(0.,0.)
        DO 1 J=1,15
  1       ZAC=ZAC+VK(J)*F1(ALFA(J)/T,PI,ALAM)
        ZAK=CMPLX(REAL(ZAC),AIMAG(ZAC))
        ZAK=ZAK/T
        RETURN
 10     ZAK=VK(1)*F2(ALFA(1)/T,PI,I,ALAM)
        DO 2 J=2,15
 2      ZAK=ZAK+VK(J)*F2(ALFA(J)/T,PI,I,ALAM)
        ZAK=ZAK/T
        RETURN
        END
C **********************************************************
        COMPLEX FUNCTION  F1(XI,PI,ALAM)
        COMPLEX XI
C  LAPLACE TRANSFORM FUNCTION OF US.
        F1=1.-CEXP(-XI*PI/(XI+ALAM))
        F1=F1/XI
        RETURN
        END
C   *************************************************************
        COMPLEX FUNCTION F2(XI,PI,I,ALAM)
        COMPLEX XI
C   LAPLACE TRANSFORM FUNCTION WHICH GIVES ELEMENTS OF C MATRICES, I
C   REPRESENTS THE POWER TO WHICH THE DENOMINATOR IS RAISED.
        F2=CEXP(-XI*PI/(XI+ALAM))
        F2=F2/(XI**I)
        RETURN
        END
C   ***********************************************************
        COMPLEX FUNCTION ZAP(T,PI,ALFA,VK,ALAM)
        COMPLEX ALFA,VK
        DIMENSION  ALFA(15),VK(15)
C    THIS FUNCTION EVALUATES THE HEAT POLE FUNCTION BY USE OF ZAKIAN'S
C     FORMULA OF LAPLACE INVERSION.
        ZAP=VK(1)*CEXP(-ALFA(1)*PI/(ALFA(1)+T*ALAM))
        DO 1 J=2,15
 1      ZAP=ZAP+VK(J)*CEXP(-ALFA(J)*PI/(ALFA(J)+T*ALAM))
        ZAP=ZAP/T
          RETURN
          END
C*****************************************************************
          FUNCTION TEMPO(AV,VLAM,N)
          DIMENSION AV(20)
```

A3-5

```
C   EXIT FLUID TEMPERATURES FOR HOT OR COLD BLOW AT TIME  ZE\O.
C   AV IS VECTOR OF COEFFTS.,VLAM IS THE APPROPRIATE LAMBDA.
C   FOR HOT BLOW NEED TO ADD EXPONENTIAL TERM IN THE REF.
C   BLOCK TO THE VALUE OF TEMPO.
C   EL=EXP(-LAM),VLI=1/LAM
        EL=EXP(-VLAM);VLI=1./VLAM
C  OBTAIN THE TERM OUTSIDE THE SUMMATION FROM 2 TO N FIRST,
C   STORING IN TEMPO.
        TEMPO=AV(1)*(1.-EL)
C  NOW OBTAIN THE LOOP SUM ADDING TO TEMPO,FINAL ANSWER IN
C   TEMPO. TEMPORARY VALUE OF SUMMATION ARGUMENT FOR CURRENT I
C   STORED IN TI*AV(I),I1F IS (I-1)FAC,SVLIM1=(-1/LAM)**(I-1)
        I1F=1;SVLIM1=1.
        DO 10 I=2,N
        I1=I-1;I1F=I1F*I1;SVLIM1=-SVLIM1*VLI
C  FIRST ADD TERMS OUTSIDE J LOOP IN TI:
        TI=1.-SVLIM1*I1F*EL
C  FOR THE J LOOP I1DI1JF=(I-1)FAC/(I-1-J)FAC,SVLIJ=(-1/LAM)**J
        SVLIJ=1.;I1DI1JF=1
        DO 20 J=1,I1
        I1DI1JF=(I-J)*I1DI1JF;SVLIJ=-SVLIJ*VLI
20      TI=TI+I1DI1JF*SVLIJ
C  MULTIPLY TI BY COEFFT.A(I) AND TOTAL UP IN TEMPO:
10      TEMPO=TEMPO+AV(I)*TI
        RETURN
        END
C*******************************************************************
        COMPLEX FUNCTION FGT(VLAM,I1,FL,S)
        COMPLEX S,SL1
C  THIS FN EVALUATES THE TRANSFORMED FUNCTION(S) REQUIRED TO BE INVERTED
C  IN OREDER TO OBTAIN FLUID TEMPERATURE PROFILE IN S DOMAIN.
C  VLAM=LAMBDA,I1 IS AN INDEX REPRESENTING THE POWER I-1 IN EXPRESSION
C  ES=((S+1)/S)**(I-1).IF I1 IS ZERO ES IS SET TO 1,OTHERWISE I1>=
C  1 IS USED IN ES.FL IS A FLAG INDICATING WHICH FORMULA IS TO BE USED.
C  IF FL NOT=1,WE USE THE FORMULA:ES*EXP(-LAM*S/(S+1))/S AND IF FL=1.,
C  THE FOLLOWING FORMULA FOR VARIABLE TIME INTEGRAL ETA* IS EVALUATED :
C  ES*EXP(-LAM*S/(S+1))/(S*S).
C  EXPONENTIAL TERM OBTAINED FIRST:
        SL1=VLAM*S/(S+1);FGT=CEXP(-SL1)/S
C  SHIFT TO STATEMENT 10 IF FL=1.,I.E.,ETA* IS BEING OBTAINED:
        IF(FL.EQ.1.)GOTO 10
C  MULTIPLY EXPRESSION BY((S+1)/S)**(I-1),RETURN IF I1=0:
        IF(I1.EQ.0)RETURN
C       FGT=FGT*((S+1)/S)**I1;RETURN
C  FOR EXPRESSION IN ETA* MULTIPLY BY((S+1)/S)**(I-1),RETURN IF
C  I1=0 AFTER NECESSARY ADJUSTMENT.
10      IF(I1.EQ.0)GOTO 11
        FGT=FGT*((S+1)/S)**I1
11      FGT=FGT/S
        RETURN;END
C  *****************************************************************
        COMPLEX FUNCTION ZAFT(ALFA,VK,VLAM,ETA,I1,FL)
        COMPLEX ALFA,VK,FGT
        DIMENSION ALFA(15),VK(15)
C THIS FN EVALUATES THE INVERSE OF LAPLACED F AT TIME ETA USING COMPLEX
C VECTORS ALFA AND VK OF ZAKIAN'S FORMULA.
        ZAFT=VK(1)*FGT(VLAM,I1,FL,ALFA(1)/ETA)
        DO 10  L=2,15
10      ZAFT=ZAFT+VK(L)*FGT(VLAM,I1,FL,ALFA(L)/ETA)
        RETURN;END
C*****************************************************************
        SUBROUTINE GT(VLI,ETA,I,TES)
C  THIS SUBROUTINE CALCULATES THAT PART OF THE SUMMATION ARGUMENT
C   FOR EXIT FLUID TEMP. WHICH DOES NOT INCLUDE THE USE OF LAP.INVN.
C   VLI=1/VLAM BROUGHT OVER FROM CALLING BLOCK,I IS THE INDEX NO.
```

A3-6

```
C   THE VALUE FOR THIS CALCULATION IS SAVED IN TES.
C   I1DI1JF=(I-1)FAC/(I-1-J)FAC,SVLIJ=(-1/LAM)**J IN J LOOP.
        I1=I-1;TES=1.;I1DI1JF=1;SVLIJ=1.
        DO 10 J=1,I1
        I1DI1JF=I1DI1JF*(I-J);SVLIJ=-SVLIJ*VLI
C  SUM OF K LOOP IS STORED IN TK,HERE JMKM1=J(J-1)...(J-K+1)
C  ,KF=(K)FAC,JKBC=BINOMIAL COEFFICIENT OF J AND K,AND ETAK=ETA**K.
        TK=1.;JMKM1=1;KF=1;ETAK=1.
        DO 20 K=1,J
        JMKM1=JMKM1*(J-K+1);KF=KF*K;ETAK=ETAK*ETA;JKBC=JMKM1/KF
20      TK=TK+JKBC*ETAK/KF
C  NOW TOTAL UP IN TES:
        TES=TES+I1DI1JF*SVLIJ*TK
10      CONTINUE
        RETURN;END
C***************************************************************/*********
        FUNCTION FT1ETA(AV,ETA,N,VLAM,FLAG,ALFA,VK)
        COMPLEX ALFA,VK
        DIMENSION AV(20),ALFA(15),VK(15)
C  THIS FN EVALUATES THE FLUID EXIT TEMP FOR HOT OR COLD FLOW AT TIME
C   ETA.AV IS VECTOR OF COEFFTS OBTAINED EARLIER,VLAM IS LAMBDA,FLAG
C   IS AN INDICATOR TOWARDS THE TYPE OF FLUID TEMP TO BE CALCULATED.
C   IF FLAG=1. WE ARE DEALING WITH A HOT BLOW AND FOR COLD FLOW FLAG
C   SHOULD BE SET TO SOMETHING NOT EQUAL TO 1.(SAY 0.).
C
C  LAPLACE INVERSION USING ZAKIAN'S FORMULA FOR THE FIRST TERM:
        FL1=ZAFT(ALFA,VK,VLAM,ETA,0,0.)
C  IF FLAG=1.USE (1-AH(1))*FL1 OTHERWISE OBTAIN -AC(1)*FL1,SAVE AS FT1ET
        FT1ETA=AV(1)*(1.-FL1)
        IF(FLAG.NE.1.)GO TO 10
        FT1ETA=FL1+FT1ETA
C  NOW OBTAIN THE REST I.E.,THE SUM OVER I=2 TO N OF
C  A(I)*(1+(I-1)FAC*SUM OVER J=1 TO (I-1) OF (-1/LAM)**J/(I-1-J)FAC
C  *SUM OVER K=0 TO J OF JKBC*ETA**K/KFAC-(-1/LAM)**(I-1)FAC
C  *LAP.INV.(EXP(-LA*S/(S+1))/S*((S+1)/S)**(I-1)).
C  VLI=1/VLAM,SVLIM1=(-1/LAM)**(I-1),I1F=(I-1)FAC IN I LOOP.
10      VLI=1./VLAM;SVLIM1=1.;I1F=1
        DO 20 I=2,N
        I1=I-1;I1F=I1F*I1;SVLIM1=-SVLIM1*VLI
C  OBTAIN NON LAP. TERM STORED IN TES:
        CALL GT(VLI,ETA,I,TES)
C  INVERT L.T. EVALUATING AT ETA:
        FL1=ZAFT(ALFA,VK,VLAM,ETA,I1,0.)
C    OBTAIN THE ARGUMENT FOR SUMMATION, MULTIPLY BY A(I) AND
C  TOTAL UP IN FT1ETA:
        FT1ETA=FT1ETA+AV(I)*(TES-SVLIM1*I1F*FL1)
20      CONTINUE
        RETURN;END
C*********************************************************************
        FUNCTION ETASTAR(ALFA,VK,ETA,CLAM,N,AC)
        COMPLEX ALFA,VK
        DIMENSION AC(N),ALFA(15),VK(15)
C  THIS FN CALCULATES ETASTAR=INTEGRAL OVER ZERO TO ETA OF COLD BLOW
C   EXIT FLUID TEMP FOR ANY ETA FROM 0. TO PIC.ETASTAR=0 IF ETA=0.
        IF(ETA.NE.0.)GO TO 10
        ETASTAR=0.;RETURN
CALCULATE THE FIRST TWO TERMS I.E., THE ONES OUTSIDE THE SERIES AND
C  STORE IN ETASTAR:
10      ETASTAR=AC(1)*(ETA-ZAFT(ALFA,VK,CLAM,ETA,0,1.))
C  NOW OBTAIN THE REST AND ADD TO ETASTAR TO GET THE FINAL ANSWER USING
C  THE I LOOP IN WHICH TIS STORES THE NON LAPLACE TERM FROM THE
C  ARGUMENT OF SUMMATION (WHICH WILL HAVE TO BE MULTIPLIED BY AC(I) AT
C  I1F=(I-1)FAC,SVLIM1=(-1/CLAM)**(I-1)
        SVLIM1=1.;I1F=1;CLI=1./CLAM
        DO 20 I=2,N
```

```
C  FOR J LOOP I1DI1JF=(I-1)FAC/(I-1-J)FAC,SVLIJ=(-1/CLAM)**J
      I1DI1JF=1;SVLIJ=1.
      DO 30 J=1,I1
      I1DI1JF=I1DI1JF*(I-J);SVLIJ=-SVLIJ*CL1
C  SUM OF K LOOP IS STORED IN TKS, HERE JMKM1=J(J-1)...(J-K+1),
C  KF=KFAC,K1F=(K+1)FAC,JKBC=BIN.COEFFT. OF J AND K, ETAK1=ETA**K1.
      TKS=ETA;JMKM1=1;KF=1;ETAK1=ETA
      DO 40 K=1,J
      JMKM1=JMKM1*(J-K+1);KF=KF*K;ETAK1=ETAK1*ETA
      JKBC=JMKM1/KF;K1F=(K+1)*KF
40    TKS=TKS+JKBC*ETAK1/K1F
C  TOTAL UP IN TIS THE NON LAP. TERM:
30    TIS=TIS+TKS*I1DI1JF*SVLIJ
C  NOW OBTAIN L.I. TERM AND SUBTRACT FROM TIS, AND GET THE FINAL
C  ANSWER AFTER MULTIPLYING BY A(I) AND SUMMING:
      FL1=ZAFT(ALFA,VK,CLAM,ETA,I1,1.)
      ETASTAR=ETASTAR+AC(I)*(TIS-SVLIM1*I1F*FL1)
20    CONTINUE
      RETURN;END
C***********************************************************************
      SUBROUTINE FLUTE(ALFA,VK,AH,AC,HLAM,CLAM,PIH,PIC,N)
      COMPLEX ALFA,VK
      DIMENSION ALFA(15),VK(15),AH(N),AC(N)
C  THIS SUB FINDS THE EXIT FLUID TEMPERATURES AT SPECIFIED POINTS OF
C  THE HOT OR COLD PERIOD.THESE POINTS ARE EQUALLY SPACED ACCORDING TO
C  THE MESH SPACING DESIRED.
      WRITE(2,10)
      READ(1,*)NPH,NPC
10    FORMAT(1X,'ENTER TOTAL  NUMBERS OF POINTS(I.E.,INTERIOR+2 END
     1POINTS)OF HOT AND THEN COLD PERIOD AT WHICH PROFILE REQUIRED-')
C  HOT BLOW FLUID EXIT PROFILE, FIRST OBTAIN TEMP. AT TIME ZERO THEN IN
C  LOOP AT POINTS FROM GREATER THAN ZERO TO PIH:
      WRITE(2,11)
11    FORMAT(1X,'HOT FLUID EXIT TEMPERATURE PROFILE:'/1X,35(1H=)/6X,
     1'TIME',10X,'TEMPERATURE')
      EH=0.;EC=0.
C  HOT TEMP AT TIME ZERO:
      TH10=EXP(-HLAM)+TEMPO(AH,HLAM,N)
      WRITE(2,12)EH,TH10
12    FORMAT(5X,F12.9,2X,F12.9)
C  HOT TEMPS AT OTHER TIMES:
      HN=PIH/(NPH-1)
      DO 20 K=2,NPH
      EH=EH+HN
      TH10=FT1ETA(AH,EH,N,HLAM,1.,ALFA,VK)
      WRITE(2,12)EH,TH10
20    CONTINUE
C  COLD BLOW FLUID PROFILE:
      WRITE(2,21)
21    FORMAT(1X,'COLD FLUID EXIT TEMPERATURE PROFILE:'/1X,36(1H=)/6X,
     1'TIME',10X,'TEMPERATURE')
COLD TEMPERATURE AT TIME ZERO:
      TC10=TEMPO(AC,CLAM,N)
      WRITE(2,12)EC,TC10
COLD TEMPS AT OTHER TIMES:
      CN=PIC/(NPC-1)
      DO 30 K=2,NPC
      EC=EC+CN
      TC10=FT1ETA(AC,EC,N,CLAM,0.,ALFA,VK)
      WRITE(2,12)EC,TC10
30    CONTINUE
C  VARIABLE TIME INTEGRAL (FOR COLD BLOW ONLY) ETASTAR:
      WRITE(2,22)
      READ(1,110)CAN
22    FORMAT(1X,'IS VARIABLE TIME INTEGRAL OF COLD EXIT FLUID TEMP.
```

A3-8

```
110      FORMAT(A1)
         IF(CAN.EQ.1HN)RETURN
         WRITE(2,25)
25       FORMAT(1X,'VARIABLE TIME COLD FLUID PROFILE:'/1X,33(1H=)/6X,
         1'TIME',10X,'ETASTAR INTEGRAL')
         EC=0.;ETS=0.
         WRITE(2,40)EC,ETS
40       FORMAT(5X,F12.9,2X,F15.9)
         DO 50 K=2,NPC
         EC=EC+CN
         ETS=ETASTAR(ALFA,VK,EC,CLAM,N,AC)
         WRITE(2,40)EC,ETS
50       CONTINUE
         RETURN;END
C**********************************************************************
```

A3-9

```
..RUN,F,F=C3
   45000B   CM STORAGE USED
      2.491 CP SECONDS COMPILATION TIME
ENTER VALUE OF N--
   CM LWA+1 = 334658, LOADER USED  47300B5

ENTER HOT AND THEN COLD LAMBDA--19.7254,17.3553

ENTER HOT AND THEN COLD PERIOD--8.5,8.5

DET=  .27466E-03
DET=  .94743E-04
ARE HOT BLOW COEFFTS.REQUIRED?YES OR NO--Y

COEFFICIENTS OF  HOT BLOW--
============================
  .64601E+00  -.10215E+01  -.30909E+00   .10261E+01  -.34144E+00
ARE COLD BLOW COEFFTS.REQUIRED?YES OR NO--Y

COEFFICIENTS OF COLD BLOW--
============================
  .15090E+00   .61529E+00   .64553E+00   .64502E+00    -.10568E+01
ARE TEMPERATURE PROFILES FOR SOLID REQUIRED?ENTER YES OR NO--Y

ENTER NUMBER OF BED MESH POINTS--31

 SOLID TEMPERATURE PROFILE
==========================
 NORMALISED DISTANCE          HOT BLOW         COLD BLOW
     0.000000000             .646014825        .999927975
      .033333333             .611658569       1.008742654
      .066666667             .576837554       1.009978890
      .100000000             .541764619       1.004385768
      .133333333             .506642486        .992681060
      .166666667             .471663760        .975551226
      .200000000             .437010928        .953651411
      .233333333             .402856362        .927605448
      .266666667             .369362317        .898005857
      .300000000             .336680931        .865413844
      .333333333             .304954224        .830359303
      .366666667             .274314100        .793340814
      .400000000             .244882348        .754825645
      .433333333             .216770637        .715249749
      .466666667             .190080521        .675017769
      .500000000             .164903437        .634503030
      .533333333             .141320706        .594047549
      .566666667             .119403530        .553962027
      .600000000             .099212996        .514525852
      .633333333             .080800073        .475987100
      .666666667             .064205616        .438562534
      .700000000             .049460358        .402437602
      .733333333             .036584921        .367766440
      .766666667             .025589806        .334671872
      .800000000             .016475399        .303245407
      .833333333             .009231969        .273547242
      .866666667             .003839667        .245606261
      .900000000             .000268530        .219420034
      .933333333            -.001521525        .194954819
      .966666667            -.001580696        .172145559
     1.000000000             .000030703        .150895887
ARE EXIT FLUID TEMPS.REQUIRED?YES OR NO--Y

ENTER TOTAL NUMBERS OF POINTS(I.E.,INTERIOR + 2 END    POINTS)OF
HOT AND THEN COLD PERIOD AT WHICH PROFILE REQUIRED-31,31
```

A3-10

```
#===================================
      TIME             TEMPERATURE
      0.000000000        .000258650
       .283333333        .001644714
       .566666667        .003366893
       .850000000        .005425856
      1.133333333        .007822036
      1.416666667        .010555708
      1.700000000        .013627117
      1.983333333        .017036642
      2.266666667        .020785016
      2.550000000        .024873603
      2.833333333        .029304732
      3.116666667        .034082088
      3.400000000        .039211165
      3.683333333        .044699763
      3.966666667        .050558537
      4.250000000        .056801571
      4.533333333        .063446969
      4.816666667        .070517453
      5.100000000        .078040945
      5.383333333        .086051103
      5.666666667        .094587816
      5.950000000        .103697612
      6.233333333        .113433984
      6.516666667        .123857591
      6.800000000        .135036351
      7.083333333        .147045379
      7.366666667        .159966793
      7.650000000        .173889360
      7.933333333        .188907999
      8.216666667        .205123122
      8.500000000        .222639854
COLD FLUID EXIT TEMPERATURE PROFILE:
#===================================
      TIME             TEMPERATURE
      0.000000000       1.001023698
       .283333333        .995788919
       .566666667        .989472241
       .850000000        .982138871
      1.133333333        .973852006
      1.416666667        .964672728
      1.700000000        .954659843
      1.983333333        .943869705
      2.266666667        .932356004
      2.550000000        .920169535
      2.833333333        .907357956
      3.116666667        .893965546
      3.400000000        .880032975
      3.683333333        .865597109
      3.966666667        .850690850
      4.250000000        .835343051
      4.533333333        .819578495
      4.816666667        .803417983
      5.100000000        .786878523
      5.383333333        .769973636
      5.666666667        .752713805
      5.950000000        .735107056
      6.233333333        .717159674
      6.516666667        .698877083
      6.800000000        .680264849
      7.083333333        .661329830
      7.366666667        .642081460
      7.650000000        .622533144
      7.933333333        .602703770
```

A3-11

```
     8,500000000     .562314509
IS VARIABLE TIME INTEGRAL OF COLD EXIT FLUID TEMP.      REQUIRED?YES OR NO--Y

VARIABLE TIME COLD FLUID PROFILE:
=================================
     TIME           ETASTAR INTEGRAL
     0,000000000       0.000000000
      ,283333333        .282908116
      ,566666667        .564178209
      ,850000000        .843513012
     1,133333333       1.120633433
     1,416666667       1.395277961
     1,700000000       1.667202015
     1,983333333       1.936177255
     2,266666667       2.201990813
     2,550000000       2.464444458
     2,833333333       2.723353678
     3,116666667       2.978546684
     3,400000000       3.229863328
     3,683333333       3.477153953
     3,966666667       3.720278175
     4,250000000       3.959103624
     4,533333333       4.193504654
     4,816666667       4.423361063
     5,100000000       4.648556844
     5,383333333       4.868979006
     5,666666667       5.084516509
     5,950000000       5.295059356
     6,233333333       5.500497883
     6,516666667       5.700722294
     6,800000000       5.895622492
     7,083333333       6.085088253
     7,366666667       6.269009779
     7,650000000       6.447278676
     7,933333333       6.619789405
     8,216666667       6.786441225
     8,500000000       6.947140666
INTERESTED IN THERMAL EFFICIENCY? YES OR NO: Y

THERMAL EFFICIENCY=  .935581708
ANOTHER RUN WITH DIFFERENT PARAMETERS?
ENTER YES OR NO --N

ANOTHER RUN WITH SAME PARAMS. BUT DIFFERENT ORDER?
ENTER YES OR NO--N

     STOP
      3.705 CP SECONDS EXECUTION TIME
```

APPENDIX A4

A finite stage method for regenerator problem:

Based on a finite stage model developed by Jeffreson [i] in 1975, here we develop a finite stage method for the thermal regenerator problem. This was the first attempt by the author towards tackling the regenerator problem. However, later on a much better method, namely that of Laplace inversion was developed which has been described in Chapter 3.

We start off with following equations which are equally applicable for hot or cold blow situations. As obtained by Jeffreson [i], for zero fluid capacitance, we have:

$$\frac{\partial t}{\partial z} + \lambda (t - T) = 0 \qquad \text{.. (A4.(1))}$$

$$\frac{\partial T}{\partial y} = \alpha(t - T) \qquad \text{.. (A4.(2))}$$

where t is fluid temperature, T is solid temperature, λ is number of transfer units, z is the distance parameter along bed length, y is the time (in seconds), and $\alpha = \lambda/V_H$ where V_H is ratio of thermal capacitance of solid (i.e., chequerwork matrix) to that of fluid in regenerator at any time. Now employing backward difference formula on $\partial t/\partial z$ and discretizing (with mesh spacing $= 1/N$), (A4.(1)) and (A4.(2)) become:

$$t_j = \frac{N}{N+\lambda} t_{j-1} + \frac{\lambda}{N+\lambda} T_j \qquad \text{.. (A4.(3))}$$

$$\frac{\partial T}{\partial y} = \alpha(t_j - T_j) \quad ; \quad j = 1,\ldots,N \qquad \text{.. (A4.(4))}$$

(A4.(3)) can be directly written into matrix form as:

$$\underline{t} = [C]\, \underline{T} + \underline{D}\, T_o \qquad \text{.. (A4.(5))}$$

where

$$[C] = \begin{bmatrix} k_2 & & & & & \\ k_2k_1 & k_2 & & & 0 & \\ k_2k_1^2 & k_2k_1 & k_2 & & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & & & \\ k_2k_1^{N-1} & \cdot & \cdot & \cdot\cdot & k_2k_1 & k_2 \end{bmatrix},$$

$$\underline{D} = [k_1, k_1^2, k_1^3, \ldots, k_1^N]^T \quad , \quad T_o(y) = \begin{cases} 1 & \text{for hot blow} \\ 0 & \text{for cold blow} \end{cases}$$

$$k_1 = \frac{N}{N+\lambda} \; , \; k_2 = \frac{\lambda}{N+\lambda} \quad \text{also} \quad 1-k_1 = k_2$$

and $k_1 = 1-k_2$.

Now substituting (A4.(3)) into (A4.(4)):

$$\frac{\partial T_j}{\partial y} = \alpha(k_1 t_{j-1} + k_2 T_j - T_j)$$

$$= \alpha(k_2 - 1)T_j + \alpha k_1 t_{j-1}$$

which becomes in matrix form:

$$\frac{d\,\underline{T}}{d\,y} = \alpha([C] - [I])\,\underline{T} + \alpha\,\underline{D}\,T_o$$

$$= [A]\,\underline{T} + \underline{B}\,T_o \qquad \text{.. (A4.(6))}$$

where $[A] = \alpha([C] - [I])$;

$[I]$ being the $N \times N$ identity matrix,

and $\underline{B} = \alpha\,\underline{D}$.

So the problem reduces to solving (A4.(6)) for the solid temperature distribution $\underline{T}$. From this, the fluid temperature profile can be obtained via (A4.(5)).

As noted earlier, $T_o(y) = \begin{cases} 0 & \text{for cold blow} \\ 1 & \text{for hot blow} \end{cases}$,

hence to solve for $\underline{T}$ in the two blows respectively (using ' for hot and " for cold blow) we have:

$$\frac{d\,\underline{T}'}{d\,y} = [A']\,\underline{T}' + \underline{B}'\,T_o \qquad \text{.. (A4.(7))}$$

$$\frac{d\ \underline{T}''}{d\ y} = [A'']\ \underline{T}'' \qquad .. \ (A4.(8))$$

Pipes and Hovanessian [ii] give the solution of:

$$\frac{d\ \underline{x}}{d\ y} = [A]\ \underline{x} + \underline{R}(y)$$

as $\underline{x}(y) = e^{[A]y}\ [\underline{x}_o + \int_0^y e^{-[A]u}\ \underline{R}(u)du]$

where $\underline{x}_o = \underline{x}(0)$.

Hence solution of (A4.(7)) is

$$\underline{T}'(y) = e^{[A']y}\ [\underline{T}'(0) + \int_0^y e^{-[A']u}\ \underline{B}'\ du] \qquad .. \ (A4.(9))$$

as $T_o = 1$

where $y \ \varepsilon \ (0,\pi')$, π' being period of a hot blow.

$$\underline{T}'' = e^{[A''](y-\pi')}\ \underline{T}''(y = \pi') \qquad .. \ (A4.(10))$$

where $y \ \varepsilon \ (\pi', \pi'+\pi'')$, π'' being period of a cold blow.

At cyclic equilibrium, $\underline{T}'(y=0) = \underline{T}''(y=\pi'+\pi'')$.

Substituting $[\varphi'(y)] = e^{[A']y}$ and $[\varphi''(y)] = e^{[A'']y}$

and defining $\underline{\eta}'(\pi') = [\varphi'(\pi')] \int_0^{\pi'} [\varphi'(-u)]\underline{B}'\ du$;

we have for (A4.(9)) and (A4.(10)):

$$\underline{T}'(\pi') = [\varphi'(\pi')]\ \underline{T}'(0) + \underline{\eta}'(\pi') \qquad ..\ (A4.(11))$$

$$\underline{T}''(\pi'' + \pi') = [\varphi''(\pi'')]\ \underline{T}''(y = \pi') \qquad ..\ (A4.(12))$$

Using the reversal conditions, we note:

$$\underline{T}''(\pi') = [I_R]\ \underline{T}'(\pi') \quad \text{and} \quad \underline{T}'(0) = [I_R]\ \underline{T}''(\pi'' + \pi')$$

where

$$[I_R] = \begin{bmatrix} & & 1 \\ 0 & \cdot & \\ & 1 & \\ \cdot & 0 & \\ 1 & & \end{bmatrix}$$ i.e., the reverse diagonal identity.

So we have for (A4.(11)), (A4.(12)):

$$\underline{T}'(\pi') = [\varphi'(\pi')]\ [I_R]\ \underline{T}''(\pi'' + \pi') + \underline{\eta}'(\pi') \qquad ..\ (A4.(13))$$

$$\underline{T}''(\pi'' + \pi') = [\varphi''(\pi'')]\ [I_R]\ \underline{T}'(\pi') \qquad ..\ (A4.(14))$$

Substituting (A4.(14)) into (A4.(13)) and separating $\underline{T}'(\pi')$ out we obtain:

$$\underline{T}'(\pi') = [\ [I] - [\varphi'(\pi')]\ [I_R]\ [\varphi''(\pi'')]\ [I_R]\ ]^{-1}\ \underline{\eta}'(\pi') \qquad ..\ (A4.(15))$$

and then resubstituting (A4.(15)) into (A4.(14)) we get

$$\underline{T}''(\pi'' + \pi') = [\varphi''(\pi'')] \left[[I]-[\varphi'(\pi')] [I_R] [\varphi''(\pi'')][I_R]\right]^{-1} \cdot \underline{\eta}'(\pi') \qquad .. \text{(A4.(16))}$$

Jeffreson [i] has termed $[\varphi]$ matrices as 'transition' matrices and in order to obtain these matrices he obtains elements P_i, $i = 1,\ldots,N$ as follows:

$$k_1 = N/(N+\lambda), \quad k_2 = \lambda/(N+\lambda) .$$

Also define $\hat{k}_1 = \dfrac{\lambda}{V_H} k_1$, $\hat{k}_2 = \dfrac{\lambda}{V_H} k_2$, $P_1 = \dfrac{1}{s+\hat{k}_1}$,

$$P_2 = \frac{k_1\hat{k}_2}{(s+\hat{k}_1)^2}, \quad P_j = k_1^{j-2} P_2[1 + \hat{k}_2P_1]^{j-2} ; \quad j = 3,\ldots,N$$

Using Binomial Theorem we have:

$$P_j = \frac{k_1^{j-1}\hat{k}_2}{(s+\hat{k}_1)^2}\left[1 + \sum_{k=1}^{j-2} \frac{(j-2)!}{k!(j-2-k)!}\left(\frac{\hat{k}_2}{s+\hat{k}_1}\right)^k\right] ; \quad j = 3,\ldots,N$$

$$= \frac{k_1^{j-1}\hat{k}_2}{(s+\hat{k}_1)^2} + \sum_{k=1}^{j-2}\left[\frac{(j-2)!}{k!(j-2-k)!}\,\frac{k_1^{j-1}\hat{k}_2^{(k+1)}}{(s+\hat{k}_1)^{k+2}}\right] ; \quad j = 3,\ldots,N$$

Hence

$$L^{-1}(P_j) = k_1^{j-1}\,\hat{k}_2\,L^{-1}[(\frac{1}{s+\hat{k}_1})^2] +$$

$$\sum_{k=1}^{j-2}[\frac{(j-2)!}{k!(j-2-k)!}\,k_1^{j-1}\,\hat{k}_2^{(k+1)}\,L^{-1}((\frac{1}{s+\hat{k}_1})^{k+2})]\;;$$

$$j = 3,\ldots,N$$

The use of formula

$$L^{-1}[(\frac{1}{s+\hat{k}_1})^j] = \frac{1}{(j-1)!}\,y^{j-1}\,e^{-\hat{k}_1 y}$$

is made to calculate the inverse Laplace transform.

Once $[\varphi]$ is available Jeffreson [i] obtains $\underline{\eta}(y)$ simply from:

$$\underline{\eta}(y) = [1 - \sum_{i=1}^{N}\varphi_{1i}(y),\; 1 - \sum_{i=1}^{N}\varphi_{2i}(y),\;\ldots,$$

$$,\ldots,\; 1 - \sum_{i=1}^{N}\varphi_{Ni}(y)]^T$$

where φ_{ij} are elements of transition matrix $[\varphi]$.

Then $\underline{T}'(\pi')$ and $\underline{T}''(\pi'' + \pi')$ can be obtained from (A4.(15)) and (A4.(16)) respectively.

Attached with this appendix is a FORTRAN program named FISZEFC which has the following main steps:

1. Calculate $[\varphi'(\pi')]$ and store in $[\varphi1]$ (using PHIMAT)

2. Calculate $[\varphi''(\pi'+\pi'')]$ and store in $[\varphi2]$ (using PHIMAT)

3. Calculate $[I]-[\varphi1][\varphi2]$ and store in $[\varphi]$ (using MATMUL)

4. Calculate $[\varphi]^{-1}$ and store in $[\varphi2]$ (using LINV2F)

5. Calculate $[\varphi2]$.DELTA(π') and store in $\underline{T}$ (using MATVEC)

This gives $\underline{T}'(\pi')$.

6. To calculate $\underline{T}''(\pi'')$ replace π' by π'' above except in step 2.

Note: As IMSL subroutine LINV2F is used for Matrix inversion, the ALTLIB8, CY=30 library module is required to be attached before execution of the program.

A sample output of the program is also attached, where a mesh spacing $N = 10$, $\lambda' = \lambda'' = 10.$, $\alpha' = \alpha'' = 5.0$ and $\pi' = \pi'' = 3.0$ have been used.

It is noted that matrix inversion is carried out successfully as the Error Parameter = 0, and number of digits after improvement remain unchanged at 6. Although here the transition matrices have been suppressed for printing, they could have been printed just by answering 'Y' if required.

Although the values obtained for temperature profiles for solid and fluid are certainly within the proper range of expected values, this method has been considered to be inadequate because of the following main reasons:

1. Reversal conditions were not properly incorporated.
2. Calculation of $\underline{\eta}(y)$ is based more on physical situation than mathematical derivation as it is "difficult" to obtain the integral $\int_0^\pi [\varphi(-u)]\underline{B}du$.
3. Backward difference formula has been used to discretise the system of equations originally considered. No comparisons with other difference formulae were done and no particular reason given why backward difference was used. A thorough analysis was required involving various difference formulae and the best scheme should have been adopted.
4. With the above three disadvantages it is quite natural that truncation error for the profiles even if obtained will not be representative of its true value. It is indicated by the results obtained that the truncation error itself would be large. Hence the results are not truly representative and therefore subject to suspicion.

It is concluded that although in the present form this finite stage method is not very "reliable" but with further planned modifications it can be developed into a useful simple scheme which works. Overall, the exercise in itself was worthwhile.

...

References:

[i] Jeffreson, C.P.; "A finite stage approach to a problem in regenerator theory - a progress report", Department of Chemical Engineering, Adelaide University, March (1975).

[ii] Pipes, L.A. and Hovanessian, S.A.; "Matrix-Computer Methods in Engineering", [Chapter 4], Wiley (1969).

...

```
      PROGRAM FISZEFC(INPUT,OUTPUT,TAPE1=INPUT,TAPE2=OUTPUT)
      DIMENSION PHI(20,20),PHI1(20,20),PHI2(20,20),DELTA
     1(20),TH(20),TC(20),TSH(20),TSC(20)
C     THIS PROGRAM CALCULATES THE TEMPERATURE DISTRI-
C     BUTION OF FINITE STAGE MODEL OF A THERMAL REGENERATOR
C     WITH ZERO FLUID CAPACITANCE.
      PRINT (2,30)
30    FORMAT(2X,'ENTER MESH SPACING N:-')
C     N IS NUMBER OF MESH POINTS.
      READ(1,*)N
      CALL SUBMAIN(PHI,PHI1,PHI2,DELTA,TH,TC,TSH,TSC,N)
      STOP
      END
      SUBROUTINE SUBMAIN(PHI,PHI1,PHI2,DELTA,TH,TC,TSH,TSC,N)
      DIMENSION PHI(N,N),PHI1(N,N),PHI2(N,N),DELTA(N),TH(N),
     1TC(N),TSH(N),TSC(N),WKAREA(460)
      PRINT (2,30)
30    FORMAT(2X,'ENTER VALUES:HLAM,CLAM,HLV,CLV,PIH,PIC:-')
C     VH IS THE RATIO OF THE THERMAL CAPACITANCE OF THE
C SOLID MATRIX TO THAT OF FLUID CONTAINED IN THE REGENERATOR AT
C ANY TIME. HERE IT IS ASSUMED TO BE INFINITY FOR ZERO FLUID
C CAPACITANCE.
C     HLAM IS THE NUMBER OF TRANSFER UNITS OF HOT FLUID.
C     CLAM IS THE NUMBER OF TRANSFER UNITS OF COLD FLUID.
C     HLV IS FINITE VALUE OF HLAM/VH.
C     CLV IS FINITE VALUE OF CLAM/VH.
C     PIH IS THE 'HOT' PERIOD.
C     PIC IS THE 'COLD' PERIOD.
      READ (1,*)HLAM,CLAM,HLV,CLV,PIH,PIC
      HK1=N/(N+HLAM)
      HK2=1-HK1
      CK1=N/(N+CLAM)
      CK2=1-CK1
      HK1D=HK1*HLV
      HK2D=HK2*HLV
      CK1D=CK1*CLV ; CK2D=CK2*CLV
C  CALCULATION OF HOT TRANSITION MATRIX:
      CALL PHIMAT(PHI1,PIH,N,HK1,HK1D,HK2D)
      WRITE (2,100)
      READ (1,102)ANS
100   FORMAT(1X,'IS TRANSITION MATRIX REQUIRED@'/1X,
     1'ENTER YES OR NO-')
102   FORMAT(A1)
      IF (ANS .EQ. 1HN)GOTO 104
      PRINT(2,10)
10    FORMAT(2X,'HOT TRANSITION MATRIX:')
      PRINT (2,2)((PHI1(I,J),J=1,N),I=1,N)
2     FORMAT(10(2X,E11.5))
C  CALCULATION OF DELTA VECTOR:
C  FIRST CALCULATE MATRIX C FOR HOT BLOW (CH) STORED IN PHI2:
104   CALL CMAT(PHI2,HK1,HK2,N)
C  CALCULATION OF CH-I(I.E.,VH/HLAM . MATRIX A) AND ITS INVERSE:
      DO 60 I=1,N
60    PHI2(I,I)=PHI2(I,I)-1.
C  INVERSION OF CH-I, STORED IN PHI:
      IDGT=6
      CALL LINV2F(PHI2,N,N,PHI,IDGT,WKAREA,IER)
      PRINT (2,61)
61    FORMAT(2X,'INVERSION OF MATRIX A FOR HOT BLOW:-')
      PRINT (2,9)IDGT,IER
C CALCULATION OF D VECTOR (STORED IN TH):
      CALL DVEC (TH,HK1,N)
C  PRODUCT OF AH INVERSE AND VECTOR BH (STORED IN TSH)(HLAM/VH CANCELS):
      CALL MATVEC(PHI,TH,TSH,N)
      DO 62 I=1,N
      PHI(I,J)=PHI1(I,J)
 62   PHI(I,I)=PHI(I,I)-1.
      CALL MATVEC (PHI,TSH,DELTA,N)
C CALCULATION OF COLD TRANSITION MATRIX:
      CALL PRIMAT(PHI2,PIC,N,CK1,CK1D,CK2D)
      WRITE(2,100)
      READ(1,102)ANS
      IF(ANS.EQ.1HN)GOTO 106
      PRINT(2,11)
 11   FORMAT(2X,'COLD TRANSITION MATRIX:')
      PRINT(2,2)((PHI2(I,J),J=1,N),I=1,N)
C  APPLICATION OF REVERSAL CONDITIONS:
106   DO 3 I=1,N
      DO 3 J=1,N
 3    PHI(I.J)=PHI1(I,J)
      DO 4 I=1,N
      DO 4 J=1,N
      NJ1=N-J+1
 4    PHI1(I,J)=PHI(I,NJ1)
      DO 5 I=1,N
      DO 5 J=1,N
 5    PHI(I,J)=PHI2(I,J)
      DO 6 I=1,N
      DO 6 J=1,N
      NJ1=N-J+1
 6    PHI2(I,J)=PHI(I,NJ1)
C  CALCULATION OF THE MATRIX EXPRESSION WHICH IS  TO BE INVERTED:
      CALL MATMUL(PHI1,PHI2,PHI,N)
      DO 7 I=1,N
      DO 8 J=1,N
      PHI(I,J)=-PHI(I,J)
 8    CONTINUE
      PHI(I,I)=1.+PHI(I,I)
 7    CONTINUE
C  CALCULATION OF THE INVERSE:
      IDGT=6
      CALL LINV2F(PHI,N,N,PHI1,IDGT,WKAREA,IER)
      PRINT(2,9)IDGT,IER
 9    FORMAT(2X,'IN CALCULATION OF INVERSE:',/2X,'NO. OF DIGITS
      1 UNCHANGED AFTER IMPROVEMENT=',I3,5X,'ERROR PARAMETER=',I3)
C  CALCULATION OF HOT SOLID TEMP. DISTN.:
      CALL MATVEC(PHI1,DELTA,TSH,N)
      PRINT (2,21)
 21   FORMAT(2X,'SOLID TEMPERATURE DISTRIBUTION FOR HOT PERIOD:'
      1,/1X,47(1H-)//)
      PRINT (2,2)(TSH(I),I=1,N)
C  CALCULATION OF COLD SOLID TEMPERATORE DISTRIBUTION:
      CALL MATVEC(PHI2,TSH,TSC,N)
      PRINT (2,22)
 22   FORMAT(2X,'SOLID TEMPERATURE DISTRIBUTION FOR COLD PERIOD:'
      1,/1X,48(1H-)//)
      PRINT(2,2)(TSC(I),I=1,N)
C  CALCULATION OF VECTOR D TO BE STORED IN DELTA:
      CALL DVEC(DELTA,HK1,N)
      CALL CMAT(PHI1,HK1,HK2,N)
      CALL CMAT(PHI2,CK1,CK2,N)
C  CALCULATION OF MATRICES C FOR HOT (STORED IN PHI1) AND COLD(IN PHI2)
C PERIODS:
C  CALCULATION OF FLUID TEMPERATURE DISTRIBUTIONS:
      CALL MATVEC(PHI1,TSH,TH,N)
      DO 35 I=1,N
 35   TH(I)=TH(I)+DELTA(I)
      PRINT(2,40)
 40   FORMAT(2X,'FLUID TEMPERATURE DISTRIBUTION FOR HOT PERIOD:'
      1/1X,47(1H-)//)
      CALL MATVEC(PHI2,TSC,TC,N)
      PRINT(2,50)
 50   FORMAT(2X,'FLUID TEMPERATURE DISTRIBUTION FOR COLD PERIOD:'/
     111X,48(1H-)//)
      PRINT(2,2)(TC(I),I=1,N)
      RETURN
      END
      INTEGER FUNCTION J1FAC(J)
C  THIS FUNCTION  CALCULATES THE VALUE OF (J-1) FACTORIAL WHERE J>=1.
C  J1FAC(J)=1 FOR J<=2.
      J1FAC=1
      IF(J.LE.2)RETURN
      J1=J-1
      DO 1 I=2,J1
 1    J1FAC=J1FAC*I
      RETURN
      END
      SUBROUTINE ILTSKJ(J,TIME,STORE,EDT)
      EXTERNAL J1FAC
C  THIS SUBROUTINE CALCULATES INVERSE LAPLACE TRANSFORM OF 1/(S+K1')**J
C   (EVALUATED  AT T=TIME), AND STORES THIS VALUE IN VARIABLE STORE
C   EDT=EXP(-DASHK1*TIME) IS WORKED OUT IN PHIMAT BEFORE CALL.
      STORE=TIME**(J-1)
      STORE=STORE*EDT
      STORE=STORE/J1FAC(J)
      RETURN
      END
      SUBROUTINE PHIMAT(PHI,TIME;N,RK1,DASHK1,DASHK2)
      EXTERNAL J1FAC
      DIMENSION PHI(N,N),ST(20)
C  THIS SUBROUTINE CALCULATES THE LOWER TRAINGULAR TRANSITION MATRIX
C   PHI USING ILTSKJ AND BINOMIAL EXPANSION.ST(I) STORES THE INVERSE
C   LAPLACE TRANSFORM OF 1/(S+K1')**(I+1),ST(N) IS USED AS A TEMP. VAR
      EDT=EXP(-DASHK1*TIME)
      PHI(1,1)=EDT
      N1=N-1
C  LOOP TO CALCULATE THE UPPER TRIANGLE OF PHI:
      DO 1 I=1,N1
      I1=I+1
      DO 11 J=I1,N
 11   PHI(I,J)=0.0
 1    PHI(I1,I1)=PHI(1,1)
C  LOOP TO CALCULATE THE LOWER OFF-DIAGONAL OF PHI:
      CALL ILTSKJ(2,TIME,ST(1),EDT)
      PHI(2,1)=ST(1)*DASHK2*RK1
      DO 2 I=3,N
 2    PHI(I,I-1)=PHI(2,1)
C  THIS LOOP CALCULATES INVERSE LAPLACE TRANSFORMS OF 1/(S+K1')**(I+1)
C   AND STORES THEM IN ST(I):
      DO 3 I=2,N1
      CALL ILTSKJ(I+1,TIME,ST(I),EDT)
 3    CONTINUE
C  LOOP TO CALCULATE THE LEFT HAND LOWER TRIANGLE OF PHI:
      DO 4 I=3,N
      I1=I-1
      I2=I-2
      NI=N-I+1
      RK1I1=RK1**I1
      ST(N)=RK1I1*DASHK2*ST(1)
      DO 5 K=1,I2
      K1=K+1
      ST(N)=ST(N)+RK1I1*DASHK2**K1*J1FAC(I1)*ST(K1)/(J1FAC(K1)*
     1J1FAC(I1-K))
 5    CONTINUE
      PHI(I,1)=ST(N)
```

```
          IF(J.EQ.N)GO TO 4
          DO 6 K=2,NI
          J=J+1
 6        PHI(J,K)=PHI(I,1)
 4        CONTINUE
          RETURN
          END
          SUBROUTINE CMAT(PHI,RK1,RK2,N)
          DIMENSION PHI(N,N)
 C  THIS SUBROUTINE CALCULATES THE ELEMENTS OF A LOWER TRIANGULAR MATRIX
 C   C(FOR HOT OR COLD BLOW) DEPENDING ON THE INPUT PARAMETERS RK1 AND
 C   RK2, AND STORES THEM IN PHI(OR INPUT MATRIX NAME).
 C  THE LOOP TO CALCULATE THE UPPER TRIANGLE:
          N1=N-1
          PHI(1,1)=RK2
          DO 1 I=1,N1
          I1=I+1
          DO 2 J=I1,N
 2        PHI(I,J)=0.0
 1        PHI(I1,I1)=PHI(1,1)
 C  THE LOOP TO CALCULATE THE LOWER LEFT HAND TRIANGLE:
          DO 3 I=2,N
 3        PHI(I,1)=RK1*PHI(I-1,1)
          DO 4 I=2,N
          NI=N-I+1
          J=I
          IF(J.EQ.N)GO TO 4
          DO 5 K=2,NI
          J=J+1
 5        PHI(J,K)=PHI(I,1)
 4        CONTINUE
          RETURN
          END
          SUBROUTINE DVEC(DELTA,RK1,N)
          DIMENSION DELTA(N)
 C  THIS SUBROUTINE CALCULATES THE D VECTOR AND STORES IN DELTA.
          DELTA(1)=RK1
          DO 1 I=2,N
 1        DELTA(I)=RK1*DELTA(I-1)
          RETURN
          END
          SUBROUTINE MATMUL(PHI1,PHI2,PHI,N)
          DIMENSION PHI1(N,N),PHI2(N,N),PHI(N,N)
 C        THIS SUBROUTINE CALCULATES THE PRODUCT OF MATRICES
 C         PHI1.PHI2 AND STORES RHE RESULT IN PHI.
          DO 1 I=1,N
          DO 1 J=1,N
          SUM=0.0
          DO 2 K=1,N
          SUM=SUM+PHI1(I,K)*PHI2(K,J)
 2        CONTINUE
          PHI(I,J)=SUM
 1        CONTINUE
          RETURN
          END
          SUBROUTINE MATVEC(PHI,DELTA,T,N)
          DIMENSION PHI(N,N),DELTA(N),T(N)
 C  THIS SUBROUTINE CALCULATES THE PREMULTIPLICATION OF A VECTOR BY
 C   A MATRIX,I.E., T=PHI.DELTA.
          DO 1 I=1,N
          T(I)=0.0
          DO 2 J=1,N
 2        T(I)=T(I)+PHI(I,J)*DELTA(J)
 1        CONTINUE
          RETURN
```

```
.RUN,F,F=F8
ENTER MESH SPACING N:-
  45000B   CM STORAGE USED
     1.221 CP SECONDS COMPILATION TIME
  CM LWA+1 = 24204B, LOADER USED  40000B10

ENTER VALUES:HLAM,CLAM,HLV,CLV,PIH,PIC:-10.,10.,5.,5.,3.,3.

S TRANSITION MATRIX REQUIRED?
NTER YES OR NO-N

INVERSION OF MATRIX A FOR HOT BLOW:
IN CALCULATION OF INVERSE:
NO. OF DIGITS        UNCHANGED AFTER IMPROVEMENT=  6     ERROR PARAMETER=  0
S TRANSITION MATRIX REQUIRED?
NTER YES OR NO-N

IN CALCULATION OF INVERSE:
NO. OF DIGITS        UNCHANGED AFTER IMPROVEMENT=  6     ERROR PARAMETER=  0
SOLID TEMPERATURE DISTRIBUTION FOR HOT PERIOD:
----------------------------------------------

 .99955E+00   .99785E+00   .99375E+00   .98593E+00   .97307E+00   .95406E+00
92817E+00   .89508E+00   .85498E+00   .80850E+00
SOLID TEMPERATURE DISTRIBUTION FOR COLD PERIOD:
-----------------------------------------------

 .44717E-03   .21497E-02   .62509E-02   .14075E-01   .26934E-01   .45936E-01
71831E-01   .10492E+00   .14502E+00   .19150E+00
FLUID TEMPERATURE DISTRIBUTION FOR HOT PERIOD:
----------------------------------------------

 .99978E+00   .99881E+00   .99628E+00   .99110E+00   .98208E+00   .96807E+00
94812E+00   .92160E+00   .88829E+00   .84839E+00
FLUID TEMPERATURE DISTRIBUTION FOR COLD PERIOD:
-----------------------------------------------

 .22358E-03   .11867E-02   .37188E-02   .88968E-02   .17915E-01   .31925E-01
51878E-01   .78399E-01   .11171E+00   .15161E+00
    STOP
      .245 CP SECONDS EXECUTION TIME
```

REFERENCES

[1A] Anzelius, A.; "On heating by media flowing through", Z.Angew. Math. Mech., 6, 291 (1926).

[1] Hausen, H.; "Naherungsverfahren zur Berechnung des Wärmeaustausches in Regeneratoren", Z. angew. Math. Mech., 2, 105-114 (1931).

[2] Hausen, H.; "Survey of heat transfer theories in regenerators", [Chapter 9 of "Heat exchangers; design and theory source book", Afgan and Schlünder, editors, McGraw-Hill] (1974).

[3] Haynes, A.B.; "The simulation and control by digital computer of a thermal regenerator", Ph.D. Thesis, The University of York (1973).

[4] Heggs, P.J. and Carpenter, K.J.; "The effect of fluid hold-up on the effectiveness of contraflow regenerators", Trans. Inst. Chem. Engrs., 232-238 (1976).

[5] Heindlhofer, K. and Larsen, B.M.; "An electrical analogue of the flow of heat in a regenerator system", AIME Metals Technology (August 1945).

[6] Illiffe, C.E.; "Thermal analysis of the counterflow regenerative heat exchanger", P. Inst. Mech. Engrs. 159, 363-372, (War emergency issue 44) (1948).

[7] Jakob, M.; "Heat transfer", Volume II, Chapters 34 and 35, Wiley (1957).

[8] Jeffreson, C.P.; "Feedforward control of blast furnace stoves", Automatica, 15, 149-159 (1979).

[9] Jeffreson, C.P.; "A finite stage approach to a problem in regenerator theory - a progress report", Department of Chemical Engineering, Adelaide University (March 1975).

[10] Jeffreson, C.P.; "Application of Zakian's Inverse Transform method to counter-flow Regenerator problem", personal communication (September 1977).

[11] Jeffreson, C.P. and Ee-Pan Chow; "Application of numerical Laplace Transform inversion techniques to the analysis of the dynamics of short parallel plate packed beds", Chem. Engg. in Aust., Vol. Ch.E. 2, 11-15 (1977).

[12] Jeffreson, C.P. and Ee-Pan Chow; "Least square coefficients for a Laplace Transform inversion formula", Journal of Computational and App. Maths. (March 1978).

[13] Jeffreson, C.P.; Project report on "Control and thermal design of Blast furnace stoves" to Australian Research Grants Committee (1978).

[14] Jeffreson, C.P.; "Regenerator theory - review", Dept. of Chem. Engg., Ad. Uni. (July 1977).

[15] Jeffreson, C.P., Burns, A. and Willmott, A.J.; "Time constant analysis for modelling heat exchangers", ASME Journal of Dynamic Systems, Measurement and Control (1978).

[16] Lambertson, T.J.; "Performance factors of a periodic-flow heat exchanger", Trans. Amer. Soc. Mech. Engrs., (A) 80, 586-592 (1958).

[17] Nahavandi, A.N. and Weinstein, A.S.; "A solution to the periodic-flow regenerative heat exchanger problem", Appl. Sci. Res., (A) 10, 335-348 (1961).

[18] Nusselt, W.; "Der Beharrungszustand in Winderhitzen", Z. Ver Dtch. Ing., 71, 85-91 (1927).

[19] Razelos, P. and Paschkis, V.; "The thermal design of blast furnace stove regenerators", Iron and Steel Engineer, 81-116 (August 1968).

[20] Smyth, M.P.; "Linear Engineering Systems", Section 5-3, Pergamon (1972).

[21] Thomas, R.J.; "Improved computer simulation of a thermal regenerator", Ph.D. Thesis, The University of York (1972).

[22] Tipler, W.; "An electrical analogue to the heat exchanger", Proc. VII Int. Congr. App. Mech., 3, 196-210 (1948).

[23] Willmott, A.J.; "Digital computer simulation of a Thermal Regenerator", Int. J. Heat and Mass Transfer, 7, 1291-1302 (1964).

[24] Willmott, A.J.; "Simulation of a Thermal Regenerator under conditions of variable mass flow", Int. J. Heat and Mass Transfer, 11, 1105-1116 (1968).

[25] Willmott, A.J.; "The regenerative heat exchanger computer representation", Int. J. Heat and Mass Transfer, 12, 997-1014 (1969).

[26] Willmott, A.J. and Thomas, R.J.; "Analysis of the Long Contra-flow Regenerative Heat Exchanger", J. Inst. Maths. Applics., 14, 267-280 (1974).

[27] Zakian, V.; "Numerical inversion of Laplace transform", Elec. Lett., 5, No.6, 120-121 (1969).

[28] Zakian, V.; "Optimisation of numerical inversion of Laplace transform", Elec. Lett., 6, No.21, 677-679 (1970).

[29] Zakian, V. and Gannon, D.R.; "Least-squares optimisation of numerical inversion of Laplace transforms", Elec. Lett., 7, No.3, 70-71 (1971).

[30] Zakian, V. and Edwards, M.J.; "Tabulation of constants for full grade I_{MN} approximants", Control System Centre Report No.312, Univ. of Manchester Institute of Science and Technology (June 1976).

[31] Control Data Matrix Algebra Subroutine Reference Manual (1969).

[32] Jeffreson, C.P.; "Dynamic simulation of Thermal Regenerator Systems under variable flow conditions", submitted to Chem. Eng. Journal (November 1979).

••