Delft University of Technology
Faculty of Aerospace Engineering
Delft

Prins Maurits Laboratory
Organization for Applied
Scientific Research TNO
Rijswijk

Memorandum M-564
Report PML 1987 - C20
SFCC Publication no. 41

NUMERICAL SOLUTIONS OF HEAT TRANSFER PROBLEMS IN CYLINDRICAL GEOMETRIES

C.W.M. van der Geld
J.A.M.A. Mies
R. Ramaprabhu

Delft/Rijswijk

March 1987

TABLE OF CONTENTS

NOMENCLATURE

Roman letters

C_p	heat capacity (J/kg K)
dt	increment of time (s)
F	view factor
k	number of grid points
L	length (m)
N	number of iterations
q	heat source density W/m^3
r	radial coordinate (m)
R_i	distance of the inner surface of a cylindrical shell to the axis (m)
R_f	radial coordinate of outer surface (m)
t	time (s)
T	temperature (K)
T_m	temperature of the surrounding medium (K)

Greek letters

α	heat diffusivity ($= \lambda/\rho\ C_p$)
β	convective heat transfer coefficient (W/m^2 K)
δ	radiative heat transfer coefficient ($= F.\varepsilon_{gas}.\varepsilon_{wall}.\sigma$)
ε	emission coefficient
λ	heat conduction coefficient (W/m.K)
μ_i	"eigenvalue" (see section 3.4)
ρ	mass density (kg/m^3)
σ	Boltzmann's constant

Acronyms

DUT	Delft University of Technology
Fo	Fourier number, $= \alpha t/L^2$
PMMA	Polymethylmethacrylate
PE	Polyethylene
PS	Polystyrene

RATE Regression rate (m/s)

TEMPROFIL Name of computer source, subject of this paper

TIMESTEP timestep, = N.dt (see section 3.1)

1 INTRODUCTION

1.1 Importance of transient temperature calculations for solid fuel combustors

Figure 1 is a schematic of a solid fuel combustion chamber. Air enters through a diaphragm, and directly downstream of the inlet a recirculation zone is established. The hollow cylindrical fuel pyrolyses and fuel gases mix and react with the oxidizer. Combustion products with air pass through an aft mixing chamber and are exhausted through a nozzle.

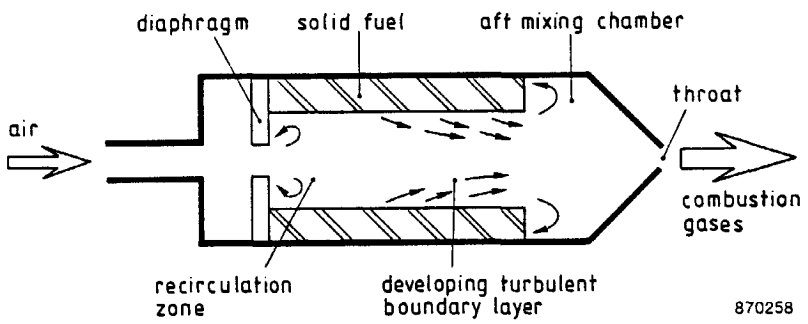

Figure 1
Schematic of a solid fuel combustion chamber.

In this report, a programme is presented to compute temperatures in a hollow, solid cylinder under a wide variety of boundary conditions. The inner boundary is not fixed in place in order to be able to simulate regression due to the pyrolising.

It is well known, that temperature has a rather severe influence on the speed of sound in a material. In the case of polymers it has been found that a strong coupling exists between the speed of sound and temperature (see North et al., ref. 6, and ref. 5).
Temperature profiles inside a fuel grain of a combustion chamber are characterized by a high surface temperature and steep gradients near the inner surface. If the travel time of a sound pulse in such a material is calculated, the temperature profile therefore has to be accurately known. Ultrasonic pulse echo techniques have been used for assessing the instantaneous regression rate

of fuel grains (ref. 4). One reason to entamate this analysis was to enhance accuracy of such pulse echo techniques.

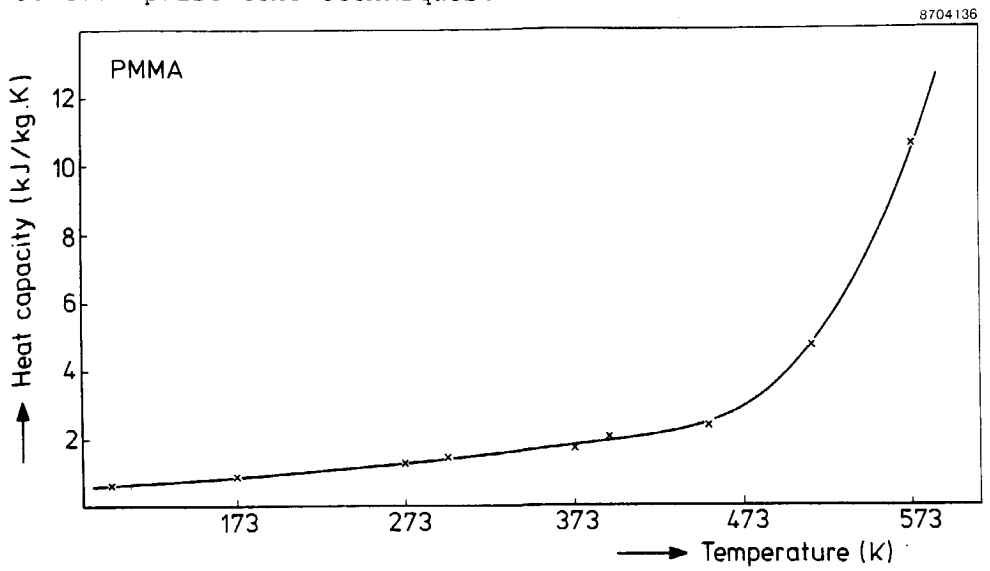

Figure 2

dependancy of the heat capacity of PMMA on temperature.

Heat diffusion inside a solid mainly controlled by the heat diffusivity, $\alpha = \lambda/\rho\ C_p$. For many polymers, the heat conduction coefficient, λ, and the

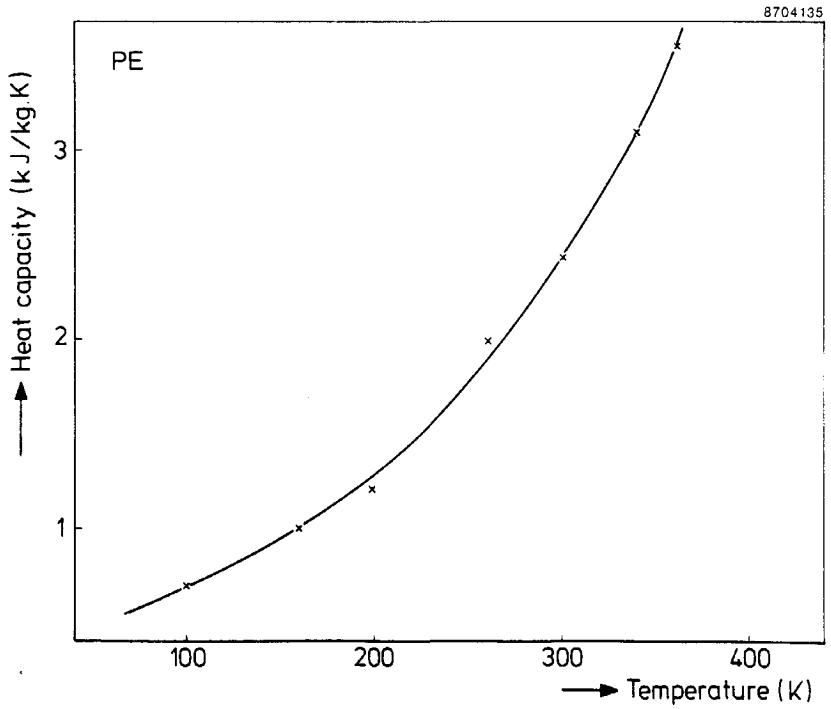

Figure 3

dependancy of the heat capacity of PE on temperature.

mass density, ρ, are only weakly dependant on temperature between 273 and 600 K. The heat capacity, C_p, on the other hand, increases dramatically above a critical value (see figure 2 through to 4).
This value is about 500 K for polymethylmethacrylate (PMMA), and about 400 K for polyethylene (PE) and polystyrene (PS). All three substances have been used for combustion experiments (ref. 4).

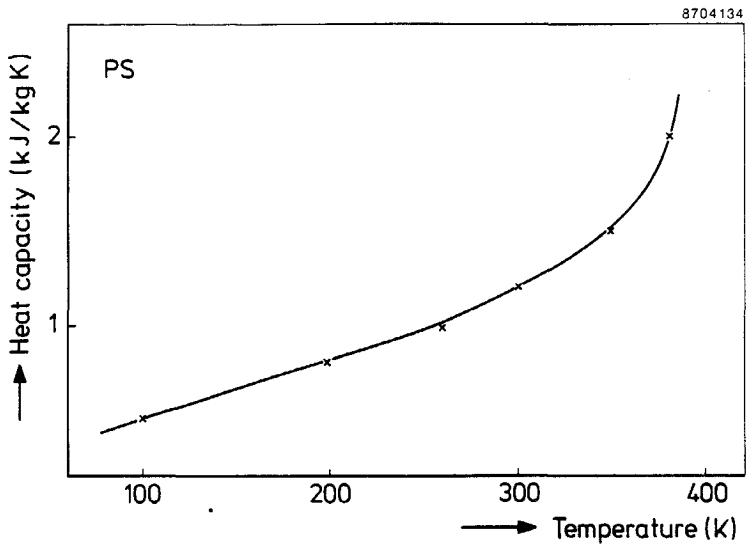

Figure 4
dependancy of the heat capacity of PS on temperature.

Figures 2 through to 4 make it clear that for accurate temperature prediction and accurate regression rate measurements, the temperature dependancy of the heat capacity has to be accounted for in the computations.

This analysis also enables effective calculation of heat transfer into the solid from the flame and the gases in a combustion chamber. Heat release from combustion gases is partly used for evaporation of the solid fuel, and partly for heating up the solid. A calculation method of this latter part is presently discussed, and can be coupled to computational models that calculate flow and combustion inside a hollow fuel grain.

1.2 Cooperation and support of the SFCC project

The overall programme "Investigation of a Solid Fuel Combustion Chamber" is financed by the Technology Foundation (Stichting voor de Technische Wetenschappen) and the Management Office for Energy Research (Stichting Projectbeheerbureau Energie Onderzoek). In addition, money and manpower are made available by a special funding of Delft University of Technology (Beleidsruimte), while also manpower, funding and computerfacilities are provided by the Faculty of Aerospace Engineering (DUT), and the Prins Maurits Laboratory (TNO).

The present investigation was carried out at the Faculty of Aerospace Engineering during the stay of R. Ramaprabhu, on sabbatical leave from Anna University, Madras, India. J. Mies is a graduate student of the Faculty of Aerospace Engineering, where C. van der Geld is senior faculty member.

1.3 Aims and scope of the present investigation

Transient heat diffusion inside a cylindrical shell is computed with the assumptions of rotatoric symmetry and axial symmetry. Boundary conditions are:

A. Inner shell surface:

-1- either the temperature, T, has to be known at any instant, or

-2- the temperature gradient, $\frac{\partial T}{\partial r}$, has to be known at any time, or

-3- the convective heat transfer coefficient, β, and the ambient temperature have to be known, or

-4- the effective radiative heat transfer coefficient, δ, and the radiation temperature have to be known, or

-5- a combination of the last two options prevails.

In addition the inner shell surface may regress, implying that the radial location of it gets larger in the course of time. The instantaneous regression rate has to be known.

B. Outer shell surface:

Either -1-, -2- or -3- (see A); no regression.

C. Inside the shell:

The dependancies on temperature of both heat capacity and heat conduction coefficient have to be known.

2 MODELLING EQUATIONS AND NUMERICAL APPROACH

2.1 Governing equation in cylindrical coordinates

The radial coordinate is chosen in a coordinate system fixed to the fuel grain (see figure 5). Note that no moving coordinate system satisfies the condition of unchanging boundaries. Hence the transformation method that Carlslaw and Jaeger (ref. 1) used to analyse an evaporating two-dimensional slab is inapplicable.

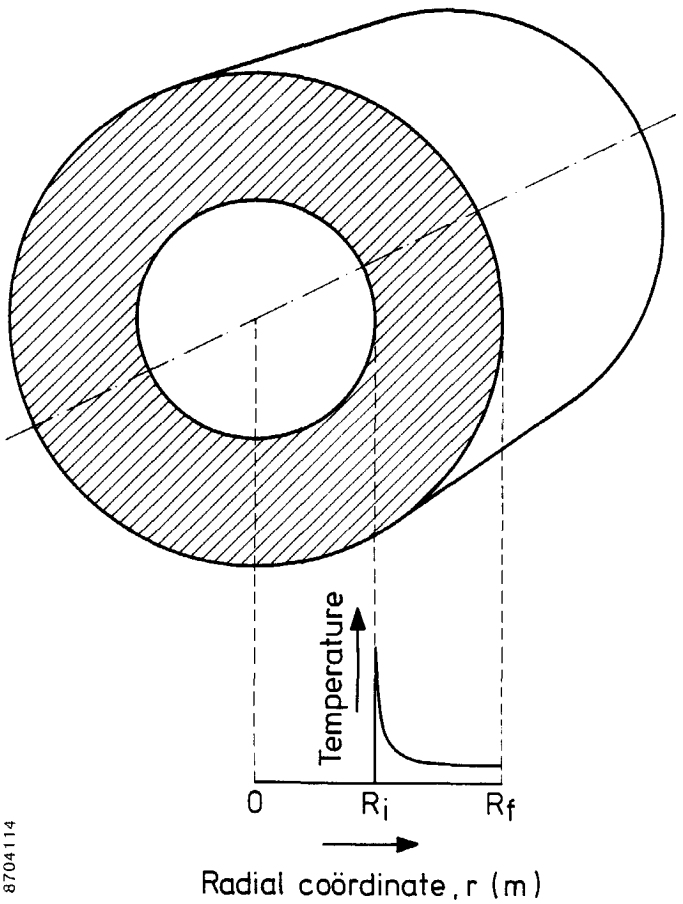

Figure 5
Cylindrical grain with coordinate definition.

The general conduction equation:

$$(2.1) \qquad \bar{\nabla} \lambda \bar{\nabla} T + q = \frac{\partial \rho C_p T}{\partial t} ,$$

with q denoting heat source density per m^3 per second, t time and λ the heat conduction coefficient (W/mK), reduces to

$$(2.2) \qquad \frac{\partial}{\partial r} \lambda \frac{\partial}{\partial r} T + \frac{\lambda}{r} \frac{\partial}{\partial r} T + q = \frac{\partial \rho C_p T}{\partial t}$$

if rotatoric symmetry is assumed.
If physical properties are independant of temperature, heat sources are absent and the heat diffusivity, α, is defined by

$$(2.3) \qquad \alpha = \lambda / \rho C_p$$

then (2.2) yields

$$(2.4) \qquad \frac{\partial T}{\partial t} = \alpha \left(\frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} \right) .$$

The above equations are valid in the solid, i.e. for $R_i < r < R_f$.

2.2 Boundary conditions

Usually four types of thermal boundary conditions are employed:

-A- Convective heat transfer, given by Newton's law of cooling:

$$(2.5) \qquad q_c = \beta \, (T_{medium} - T_{surface})$$

in which β denotes the heat transfer coefficient (W/m^2K) and q_r the heat flux (W/m^2);

-B- Fixed surface temperature;

-C- Fixed temperature gradient at the surface;

-D- Radiant heat transfer, given by

$$(2.6) \qquad q_r = \varepsilon_{surf} \, \varepsilon_{medium} \, \sigma \, (T^4_{medium} - T^4_{surface})$$

in which σ denotes the Boltzmann constant and the ε's the emissivities corrected by view factors.

Options A and D can be combined. Note that for very small or very large differences $(T_{medium} - T_{surface})$, q_r can be written as

$$(2.7) \qquad q_r = \gamma(T) \, (T_{medium} - T_{surface})$$

Options A, B and D were implemented in the programme "TEMPROFIL".

A different boundary condition is the location of the inner surface. Since the cylindrical grain pyrolyzes at the inside, the inner surface regresses and its location is a function of time. It can be calculated from instantaneous values of the regression rate, RATE.
In this report, RATE is assumed to be known as a function of time, t. The cylindrical wall is discretized in radial direction into a fixed grid system.

From RATE (t_o) the timestep is calculated that the grain surface needs to travel to the next grid point. The next time is then given by t_o + timestep.

2.3 Choice of discretization method

Equation (2.2) has to be solved with the boundary conditions discussed in section 2.2. Since eq. (2.2) is strictly nonlinear, the solution was found numerically.

Usually three types of numerical schemes are employed:

-1- explicit schemes, obtained by forward differences from the governing equation(s);

-2- implicit schemes, obtained by backward differences from the governing equation(s). Forward and backward refer to time.

-3- Crank-Nicolson schemes, essentially a combination of the former schemes.

Implicit schemes are absolutely stable but require much computer core storage. Explicit schemes can only be used with adequate stability criterions and usually require many iterations.
To gain experience and to have direct control of the accuracy of the solutions an explicit scheme was chosen for the first version of "TEMPROFIL".

The discretizing of eq. (2.2) for the inner solid with q = 0 yields

$$(2.8) \qquad T_i^{t+dt} = T_i^t + \frac{\lambda\, dt}{\rho\, C_p\, dr^2}\left(T_{i+1}^t - 2\,T_i^t + T_{i-1}^t\right) + \frac{\lambda\, dt}{2\,\rho\, C_p\, r_i\, dr}\left(T_{i+1}^t - T_{i-1}^t\right)$$

where it is understood that λ, ρ and C_p may depend on temperature. Index i refers to grid point i, and t to the previous time.

If convective heat transfer according to eq. (2.5) is assumed at boundary location, f, than the discretizing of eq. (2.2) with q ≠ 0 yields

$$(2.9) \qquad T_t^{t+dt} = T_f^t + \frac{2\,\beta\, dt}{\rho\, C_p\, dr}\left(T_{medium} - T_f^t\right) + \frac{2\,\lambda\, dt}{\rho\, C_p dr^2}\left(T_{f-1}^t - T_f^t\right) - \frac{\lambda\, dt}{\rho\, C_p\, r_f\, dr}\left(T_{f-1}^t - T_f^t\right)$$

For the inner boundary of the tube wall, indexed by i, a similar equation holds. The latter equation is obtained from eq. (2.9) by changing the sign of the last term, changing (f - 1) into i + 1 and f into i.
Boundary condition -B- (see section 2.2) is the trivial equation $T_i^{t + dt} = T_i^t$.

The boundary condition described by eq. (2.6) adds to the RHS of eq. (2.9) the term

$$(2.10) \qquad \frac{2\ \delta\ dt}{\rho\ C_p\ dr} \left(T_{medium}^4 - T_f^4\right)$$

in which δ denotes the products $\varepsilon_{surf}\ \varepsilon_{medium}\ \sigma$.

A criterion for stability of solutions of the above finite difference equations is (see Nogotov, 1978)

$$(2.11) \qquad dt \leq \frac{dr^2}{4\ \alpha}$$

in which α denotes thermal diffusivity.
These equations were used in the programme to be described in the next chapter.

3 PASCAL COMPUTER PROGRAMME

3.1 Description of the programme

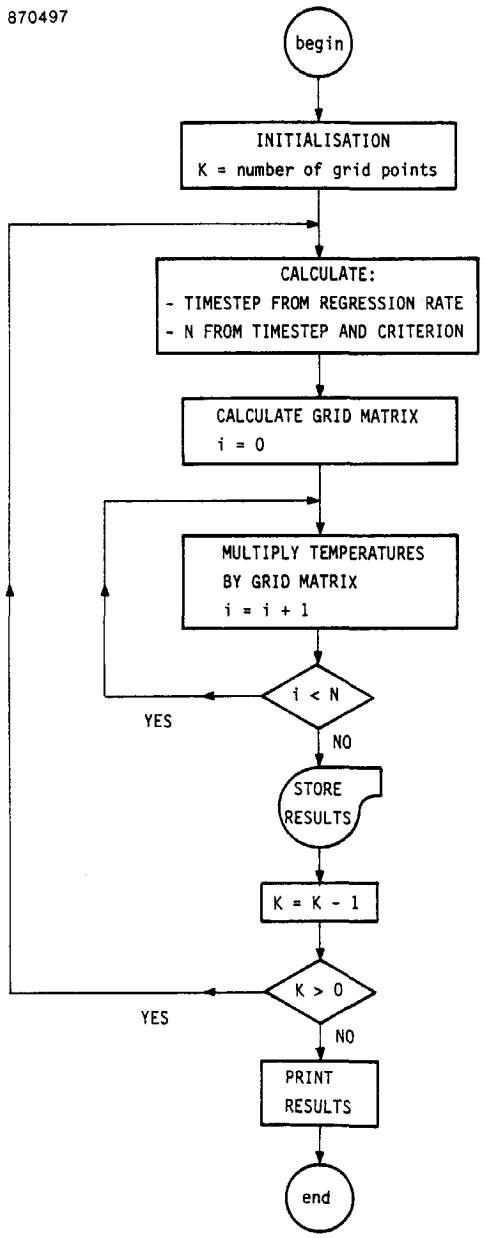

Figure 6

Flowchart of the programme "TEMPROFIL"

Based on equations (2.8) through (2.11), a computer programme named TEMPROFIL, was developed and run on the IBM 370/155 mainframe computer of Delft University of Technology. A flowchart is given in figure 6.

After initialisation, the first timestep, TIMESTEP, is calculated as the time required for the grain inner surface to reach the location corresponding to the second grid point. TIMESTEP depends on the momentary regression rate. The number of iterations, N, is then calculated with the aid of criterion (2.11). Note that N.dt = TIMESTEP. Next, the new temperatures $T_i^{t + dt}$ for all grid points i, are N times calculated, see eq. (2.8) and (2.9). For this purpose a matrix procedure is used. Subsequently, results are stored whereafter the next value of TIMESTEP is calculated, and so on and so forth. Finally results are printed or plotted.
Any arbitrary function of temperature can be used for C_p and λ, while regression rate can be a function of time.

3.2 Input and output variables

Input parameters are adjusted directly in the source, before compilation. No utility for interactive input of parameters was used.
Input parameters are:

- initial inner radius, R_i
- outer radius, R_f
- regression rate (as a function of time)
- c_p and λ (as a function of temperature)
- ρ
- K, the number of grid points (K ~ 400)
- choice of boundary conditions, via options (see also section 2.2)
- ambient temperatures if necessary
- Viewfactor, emissivity and absorption coefficient if necessary.

Output variables are:

- temperature distribution before each adjustment of the value of TIMESTEP
- elapsed time
- momentary position of grain inner surface

- configuration and input parameters (only once), e.g. number of grid points, R_i, and R_f.

3.3 Accuracy of the computations

The accuracy of the calculations with TEMPROFIL is mainly determined by the value of k, the number of grid points. Note that the timestep is limited by the condition (2.11). The dependance of temperature calculations on the value of k was investigated for two conditions:

-A- R_i = 3 mm ; R_f = 18 mm ; RATE = 0,15 mm/s

-B- R_i = 20 mm ; R_f = 35 mm ; RATE = 0,15 mm/s

Note that the effect of curvature of the wall is more pronounced for smaller vlues of R_i.
Figure 7 shows computational results for case A, and figure 8 for case 8.

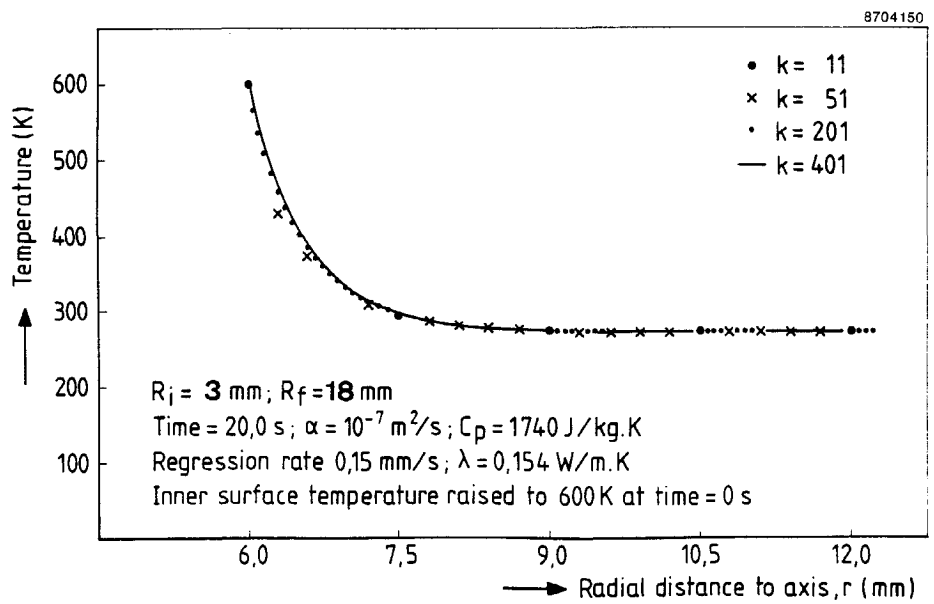

Figure 7

Convergence and accuracy study for R_i = 3 mm.

The temperatures are seen to gradually approach a limiting value if the value of k is increased. A typical value of k above which hardly any temperature differences are calculated is 201. The dependance of k on the thickness (R_f - R_i) and the thermal diffusivity, α, is represented by

$$k \propto \frac{(R_f - R_i)}{\sqrt{\alpha}} \qquad (3.1)$$

This can be deduced from the observartion, that thermal diffusion in a slab with a thickness dr is controlled by the Fourier number $Fo = \frac{\alpha t}{(dr)^2}$. To preserve the value of Fo for each "slab" (actually a shell) with thickness dr in the cylindrical grain, the value of K has to be adjusted according to eq. (3.1) if (R_f - R_i) or α is changed.

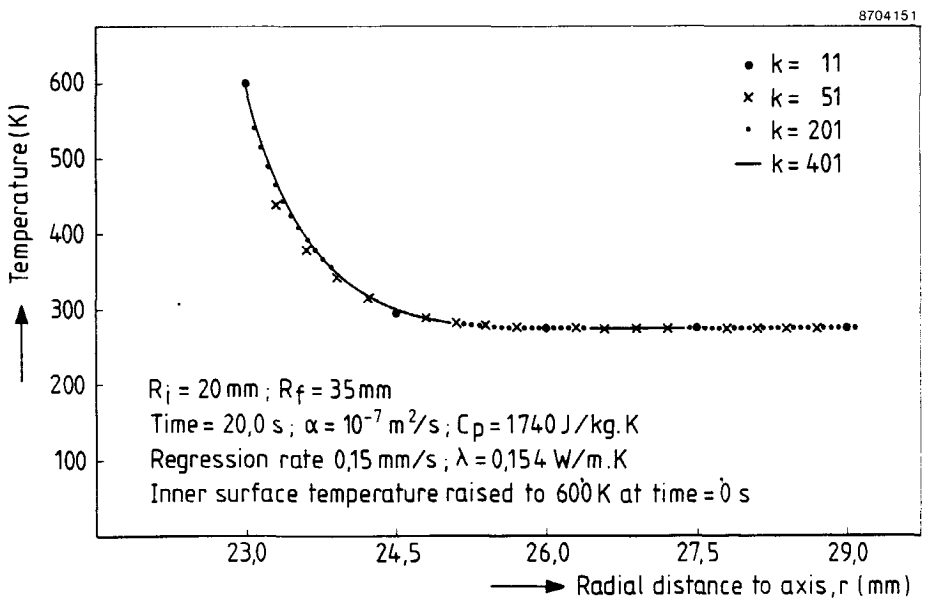

Figure 8

Convergence and accuracy study for R_i = 20 mm

A comparison of figures 7 and 8 confirms that eq. (3.1) is legitimate and useful an equation. It also shows that the larger the wall curvature, the

lower the temperatures inside the solid. This stands to reason: heat conduction in circumpheral direction hampers heat transfer in radial direction, and this effect is more pronounced if curvature is high. Figure 8 will be further discussed in section 3.4.

3.4 Validation with analytical results

Analytical solutions for the cooling or heating of a cylindrical grain can be found if regression is absent and material properties are independant of temperature. Usually solutions are represented as a sum over i of terms, each individual term of the form

$$f(\mu_i)\, e^{-Fo\, \mu_i^2} \qquad (3.2)$$

where Fo denotes a Fourier number and the values of μ_i correspond to solutions of an implicit equation involving boundary conditions (via Biot numbers). The temperature profile at time zero determines constants in the functional expression for f. The value of μ_i is of the order $i.\pi$, whence the sum rapidly converges.

Note that analytical solutions predict temperatures at any location and at any time.

Analytical solutions for heat transfer under the above mentioned restrictions were found and implemented by one of the authors (ref. 2). Although sources are not public domain property, they were used to validate computational results of TEMPROFIL.

Two cases were investigated for the purpose of validation:

-1- convective heat transfer to the inner boundary (see eq. 2.5) with β = 100 W/m^2K, uniform initial temperature of 303 K, ambient temperatures 353 K (inside the tube) and 323 K (outside the cylinder); convective heat transfer to outer surface with β = 10 W/m^2K;
R_i = 0,200 m; R_f = 0,215 m; ρ = 885 kg/m^3; λ = 0,154 W/mK;
C_p = 1740 J/kg K.

-2- uniform initial temperature of 273 K; inner surface temperature was raised to 872,6 K at time zero and kept the same value at subsequent times; no heat transfer to the outer boundary.
R_i, R_f, λ and C_p: see case -1-;
ρ = 1190 kg/m^3 K

As already mentioned, regression is zero and material properties are temperature independant.

Figure 9 shows the results for case -1-.

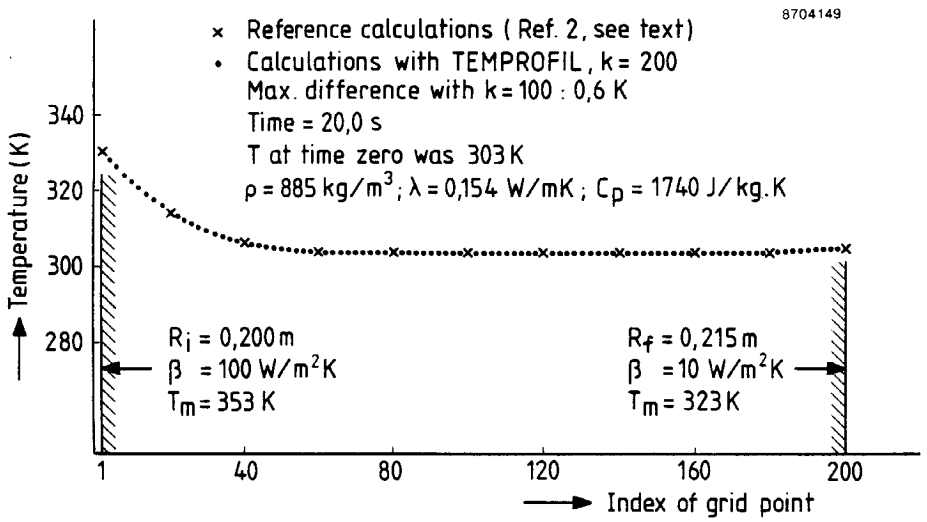

Figure 9

Validation of computational results for convective heat transfer.

It is clear that for k larger than zero the results with TEMPROFIL are in perfect agreement with the analytical results, of which only selected data are represented in the figure. The same conclusion is reached upon examining the results for case -2- (see figure 10).

If the value of R_i is large, the cylindrical wall can be approximated by an infinite slab of the same thickness. Only for slabs a moving coordinate system is allowed. If the regression rate, RATE, of a slab is constant, the temperature profile at any time, t, can be calculated in a coordinate system fixed to the regressing surface (see Carlslaw and Jaeger, ref. 1). With x = 0 at the surface, the temperature profile is given by

$$(3.3) \qquad T = T_o + \frac{1}{2} (T_{surface} - T_o) f(x,t) \quad .$$

$$f(x,t) = \operatorname{erfc}\left(\frac{-x-RATE.t}{2\sqrt{\alpha t}}\right) + \exp\left(\frac{RATE.x}{-\alpha}\right) \operatorname{erfc}\left(\frac{-x + RATE.t}{2\sqrt{\alpha t}}\right)$$

Here t denotes time, T_o the initially uniform temperature, and

$$(3.4) \qquad \operatorname{erfc}(y) = 1 - \frac{2}{\sqrt{\pi}} \int_o^y \exp(-t^2)\, dt$$

If these formulae are applied to the conditions of figure 8, a perfect agreement is observed with the results for k = 401.

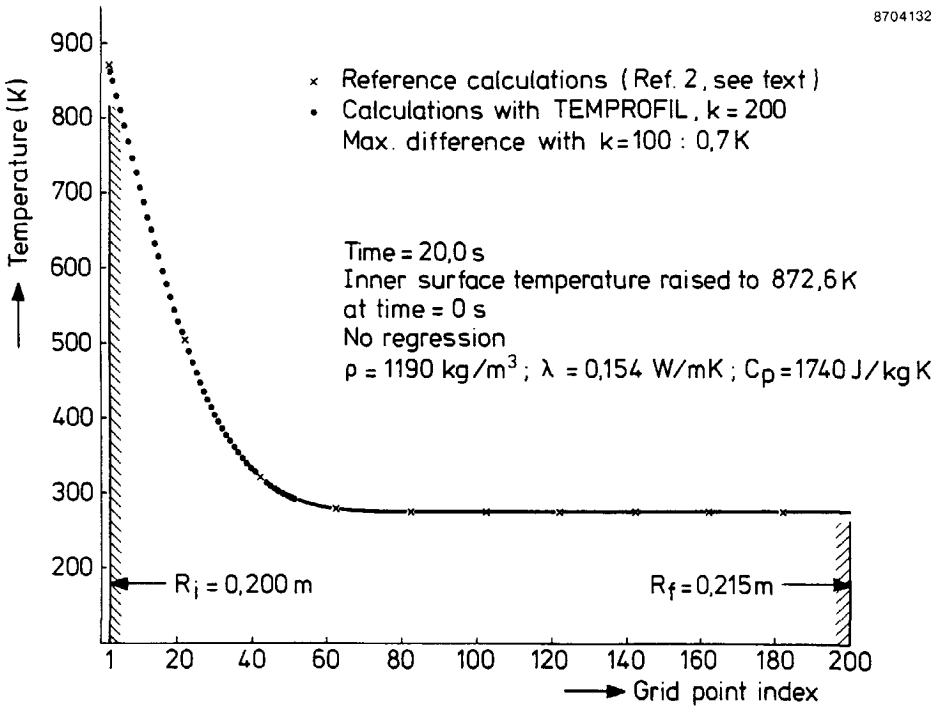

Figure 10

Validation of computational results with constant surface temperature.

4 TRANSIENT REGRESSION WITH LARGE TEMPERATURE GRADIENTS

Computational results of the programme TEMPROFIL for constant regression and temperature independant material properties were discussed in sections 3.3 and 3.4. Large temperature gradients in a cylindrical fuel grain are now accounted for by allowing for temperature dependance of material properties. Also a case of transient regression is studied in this chapter.

4.1 Time varying regression

Temperature profiles were computed for the case that regression rate, RATE, is the following function of time:

$$RATE = A + B \cos \omega t$$

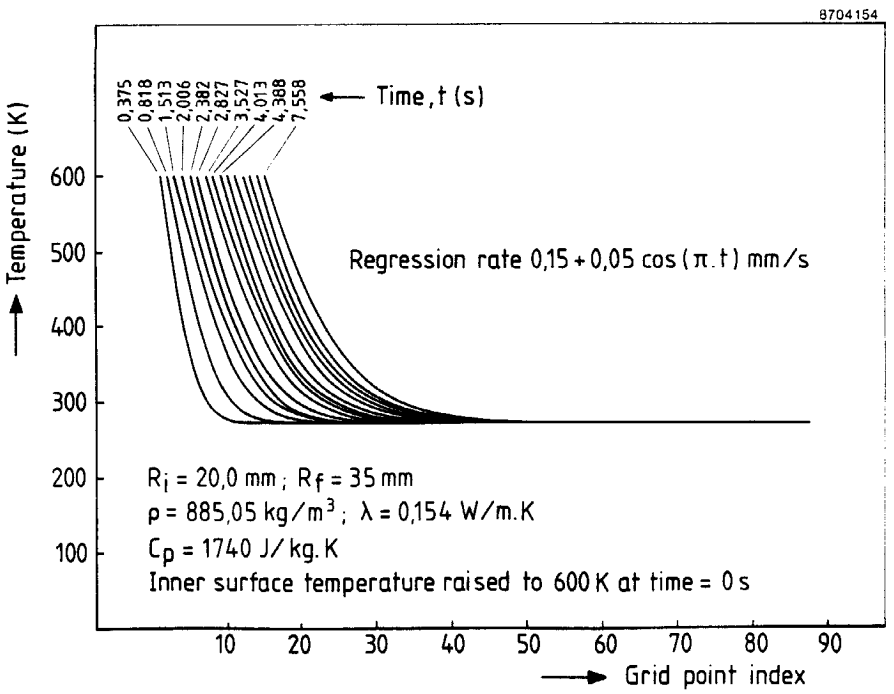

Figure 11

Temperature profiles with oscillatory regression rate.

with A = 0,15 mm/s, B = 0,05 mm/s, and $\omega = 2\pi\ \nu$, ν being 1/2 Hz. Remaining parameters were kept constant: ρ = 885,05 kg/m^3; λ = 0,154 W/mK; C_p = 1740 J/kg.K; R_i = 20 mm; R_f = 35 mm; ablation (inner surface) temperature 600 K; no heat transfer to the outer surface.

Figure 11 shows the results of various timesteps.

The effect of the superposition of an oscillatory regression component is oscillatory heat diffusion, as reflected by the temperature profiles of figure 11.

4.2 Temperature dependance of physical properties

For a constant regression rate, r = 0,15 mm/s, temperature profiles were computed for the following two cases:

-A- heat conduction coefficient, λ, given by

$\lambda = 0{,}154\ (1 + (T-273)/327)$ W/mK
(T in K)
while keeping C_p constant;

-B- heat capacity, C_p, given by
$C_p = 1740\ (1 + (T - 273)/327)$ J/kg K
(T in K)
while keeping λ constant.

Figure 12 shows the results for case -A- and figure 13 for case -B-.

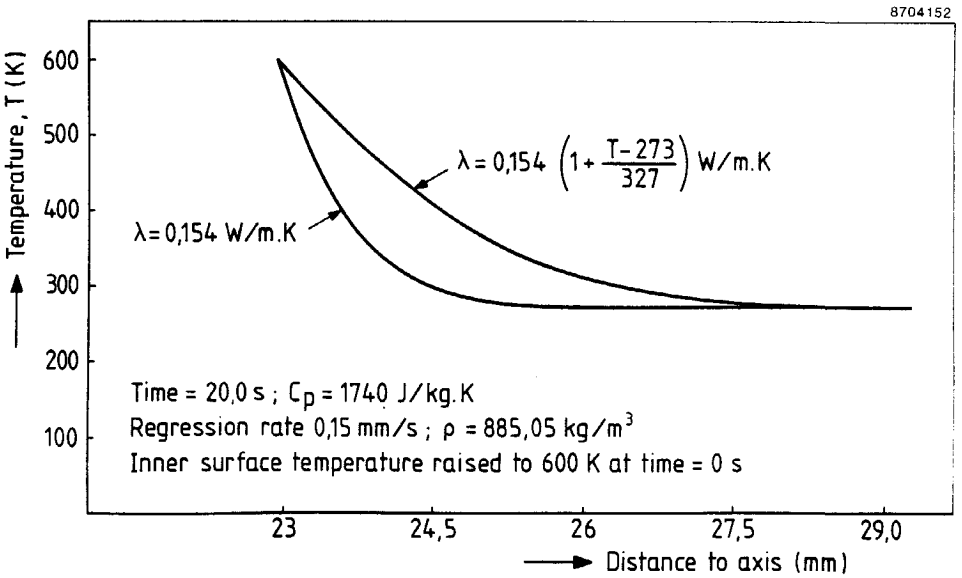

Figure 12

Temperature profiles for a linear dependancy of the heat conduction coefficient on temperature.

```
   WRITE('CP     AT 273 DEGREES         :',CP(273):11:6);
   WRITELN(' Joules/Kilogram/Kelvin.');
   WRITE('MASS DENSITY                  :',RHO:11:6,' Kilogrammes/');
   WRITELN('Squared meters.');WRITELN;
   WRITELN('DISCRETISATION PARAMETERS');
   WRITELN('-------------------------');
   WRITELN('NUMBER OF GRID POINTS        :',K:5);
   WRITELN('DISTANCE BETWEEN TWO GRID');
   WRITELN('GRID POINTS                  :',DX:11:6,' Meters.');
   WRITELN('INITIAL NUMBER OF ITERATIONS :',NSLAGEN:5);
   DTXNS := DT*NSLAGEN;
   WRITELN('INITIAL TIMESTEP             :',DTXNS:11:6,' Seconds.');
   IF REGR THEN
   BEGIN
      WRITE('REGRESSION RATE AT TIME = 0  :',RATE(0):11:6,' Meters/');
      WRITELN('Second.');
   END;
   WRITELN;
   WRITELN('BOUNDARY CONDITIONS');WRITELN('-------------------');
   IF OPTAR THEN
   BEGIN
      WRITELN('AT THE OUTER SURFACE:');WRITELN;
      WRITE('BETA                          :',BETAR:11:6,' Watts/');
      WRITELN('Squared meters/Kelvin.');
      WRITELN('TEMPERATURE                   :',UMR:11:6,' Kelvin.');
   END;
   IF OPTAL THEN
   BEGIN
      WRITELN('AT THE INNER SURFACE:');WRITELN;
      WRITE('BETA                          :',BETAL:11:6,' Watts/');
      WRITELN('Squared meters/Kelvin.');
      WRITELN('TEMPERATURE                   :',UML:11:6,' Kelvin.');
   END;
   IF OPTB THEN
   BEGIN
      WRITELN;
      WRITELN('A CONSTANT TEMPERATURE');
      WRITELN('AT THE INNER BOUNDARY OF     :',TABL:11:6,' Kelvin.');
   END;
   IF OPTD THEN
   BEGIN
      WRITELN('AT THE INNER SURFACE:');WRITELN;
      WRITE('DELTA                         :',DELTAL:11:9,' Watts/');
      WRITELN('Squared meters/Squared Kelvin/Squared Kelvin.');
      WRITE('SIGMA                         :',SIGMA:11:9,' Watts/');
      WRITELN('Squared meters/Squared Kelvin/Squared Kelvin.');
      WRITELN('EMISSION COEFFICIENT WALL    :',EPSW:11:6);
      WRITELN('EMISSION COEFFICIENT MEDIUM  :',EPSM:11:6);
      WRITELN('VIEWFAKTOR                    :',F:11:6);
      WRITELN('TEMPERATURE                   :',UML:11:6,' Kelvin.');
   END;
END;
%PAGE;
(*****************************************************************)
(*                                                               *)
(* PROCEDURE CALCMATRIX                                          *)
(*                                                               *)
(* This procedure fills the band-matrix 'A' with the             *)
(* coefficients for the present time-interval 'DT'.              *)
(*                                                               *)
(* LIST OF VARIABLES.                                            *)
(*                                                               *)
(*  R(K)     Radius at each grid point.          Meters          *)
(*  A(K,3)   Grid matrix.                                        *)
```

```
(*  DT       Time interval.                          Seconds          *)
(*  RHO      Mass density of the fuel.               Kilogrammes/     *)
(*                                                   Squared meters   *)
(*  DX       Distance between two neighbouring                        *)
(*           grid points.                            Meters           *)
(*  BETAR    The heat transfer coefficient at the                     *)
(*           right boundary.                         Watts/Squared    *)
(*                                                   meters/Kelvin    *)
(*  BETAL    The heat transfer coefficient at the                     *)
(*           left boundary.                          Watts/Squared    *)
(*                                                   meters/Kelvin    *)
(*  B        Indicates the grid point where the                       *)
(*           inner surface is located.                                *)
(*  NSLAGEN  The number of times that the vector U                    *)
(*           has to be multiplied by the matrix A                     *)
(*           multiplied by the matrix A to get the                    *)
(*           temperature distribution at the                          *)
(*           moment that the inner boundary has                       *)
(*           moved to the next grid point.                            *)
(*  OPT.     Options for different boundary                           *)
(*           conditions.                                              *)
(*                                                                    *)
(**********************************************************************)
PROCEDURE CALCMATRIX (VAR R                      : VECTOR;
                      VAR A                      : MATRIX;
                      VAR DT,RHO,DX,BETAR,BETAL  : REAL;
                      VAR B,NSLAGEN              : INTEGER;
                      VAR OPTAL,OPTAR,OPTB,OPTD  : BOOLEAN);
VAR I       : INTEGER;
    C1,C2,C3: REAL;
BEGIN
   A(.B,2.) := -(2/RHO/DX/DX+1/RHO/DX/R(.B.))*DT;
   A(.B,3.) := (2/RHO/DX/DX+1/RHO/DX/R(.B.))*DT;
   IF OPTD THEN
   BEGIN
      A(.B,1.) := 0;
   END;
   IF OPTAL THEN
   BEGIN
      A(.B,1.) := 2*BETAL/RHO/DX*DT;
   END;
   IF OPTB THEN
   BEGIN
      A(.B,1.) := 0;
      A(.B,2.) := 0;
      A(.B,3.) := 0;
   END;
   C2    := -2*DT/RHO/DX/DX;
   FOR I := B+1 TO K-1 DO
   BEGIN
      C1 := DT/RHO*(1-DX/(2*R(.I.)))/DX/DX;
      C3 := DT/RHO*(1+DX/(2*R(.I.)))/DX/DX;
      A(.I,1.) := C1;
      A(.I,2.) := C2;
      A(.I,3.) := C3;
   END;
   IF OPTAR THEN
   BEGIN
      A(.K,1.) := (2/RHO/DX/DX-1/RHO/DX/R(.K.))*DT;
      A(.K,2.) := -(2/RHO/DX/DX-1/RHO/DX/R(.K.))*DT;
      A(.K,3.) := 2*BETAR/RHO/DX*DT;
   END;
END;
%PAGE;
```

A comparison of figure 12 with the results given in figure 8 shows that more heat is absorbed by the solid material if a more realistic temperature dependance of λ is accounted for.

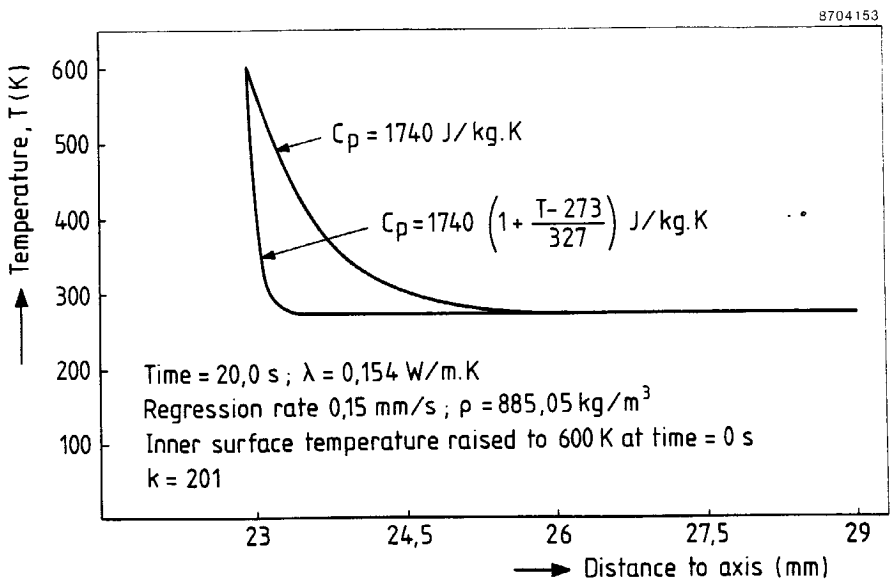

Figure 13

Temperature profiles for a linear dependance of the heat capacity on temperature.

A different conclusion is reached upon inspecting figure 13. Here the temperature, T, is plotted for time t = 20 s, and the comparison with the constant C_p-case shows smaller temperatures. Note that the product $C_p.T$ is a measure for the heat absorbed by a substance, and that the value of $\int_{R_i}^{R_f} dr.2\pi.r.C_p.T$ is much less different, about 2% after 20 s, than the temperatures themselves.

The above results are in agreement with common experience, although no experimental data were found to compare with.

5 SCOPE FOR FUTURE WORK

The programme "TEMPROFIL" may be extended by

- allowing the mass density to be temperature dependant;
- integrating $\int_{R_f} \rho\, C_p\, T\, 2\pi r\, dr$ in order to evaluate the total heat absorbed (only a simple integration routine was applied for testing, see chapter 4; much more accurate schemes are available);
- applying an implicit scheme in order to get rid of stability criteria and maybe to increase speed of computation;
- allowing heat diffusion to be three-dimensional. This requires an adaptation of the main equations, making the "matrix" in the programme essentially 3-D.

Furthermore it is possible to calculate the local speed of sound inside the solid, and to incorporate the programme in the analyzing procedure of an ultrasonic regression rate analyzer (see chapter 1 and figure 14).

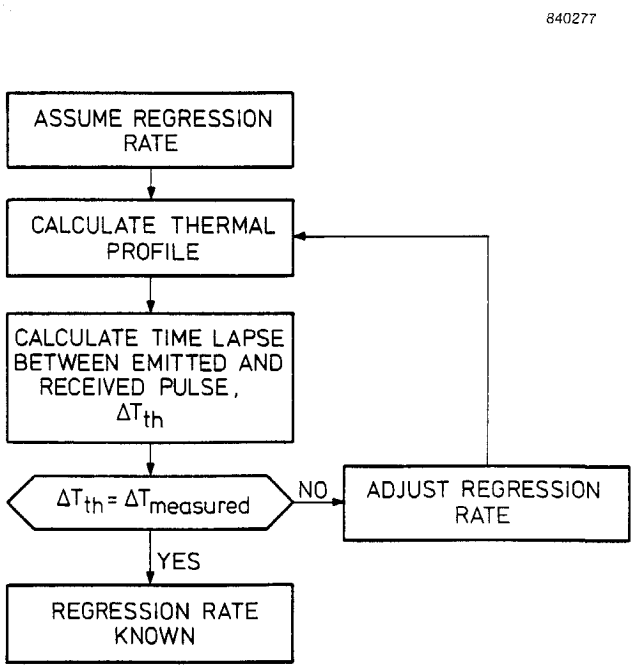

Figure 14

Flow chart of regression rate analysis.

However, the major uncertainty of this technique stems from the fact, that the boundary condition at the grain inner surface is unknown. This problem should be solved in order to really enhance accuracy of the regression analyzer. A not constant regression rate is accounted for by TEMPROFIL, but the assumption of a constant regression is not as important as the thermal boundary condition.

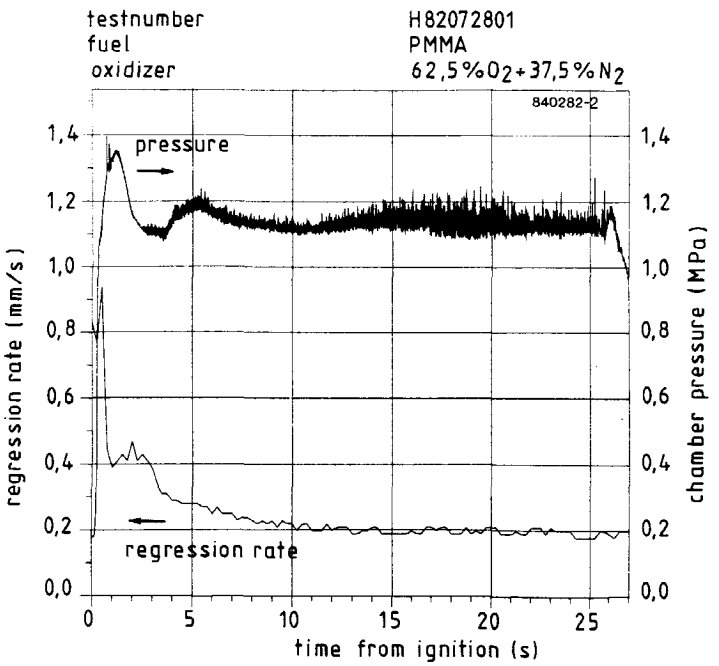

Figure 15

Specimen of ultrasonic regression rate measurement

The programme "TEMPROFIL" can also be used to calculate heat diffusion into a solid in combination with programmes that calculate flow and combustion inside a hollow fuel grain (see chapter 1). The great many of inside boundary conditions that TEMPROFIL can cope with makes the programme flexible enough for this application.

6 CONCLUSIONS

A PASCAL programme, named TEMPROFIL, was developed for the computation of transient temperature behaviour of cylindrical bodies under a variety of boundary conditions, amongst which transient regression of the inner surface, and temperature dependance of material properties.
Computational results were validated with analytical results, and the value of the number of grid points was related to the accuracy of the calculations.

Some practical cases of transient regression with large temperature gradients were investigated. If the heat flux due to convection or radiation to the inner surface is known at any time, the heat diffusion into the pyrolyzing body can be calculated. This allows for a coupling of TEMPROFIL with computational models that calculate flow and combustion inside a hollow cylindrical grain. Since temperature profiles are also calculated, the programme can be used to improve upon the results of ultrasonic regression rate analyzers. They basically assess travel times of sound pulses, and the speed of sound of commonly used materials is very dependant on temperature.

REFERENCES

(Alphabetic order)

1 Carslaw, H.S. and J.C. Jaeger
Conduction of heat in solids
oxford Univ. Press, sec. ed. 1960

2 Consultance agency "CWM"
Address: Bankven 40, 5052 BD Goirle, The Netherlands

3 Nogotov, E.F.
Applications of numerical heat transfer
McGraw-Hill, New York, 1978

4 Korting, P.A.O.G., H.F.R. Schöyer, C.W.M. van der Geld
On the use of ultrasonic pulse-echo technique for regression rate analysis. Part II: Some experimental results.
Report LR-453-II, PML-1985-C6, SFCC publication nr. 24, Delft Univ. of Technology, 1987.

5 Krauthkrämer, J. and H. Krauthkrämer
Ultrasonic testing of materials
Springer-Verlag, Berlin, sec. ed. 1977

6 North, A.M., R.A. Pethrick, D.W. Philips
Ultrasonic Studies on Polymers (PMMA)
Polymer J, Vol. 18, April 1977

APPENDIX 1

Source listing of "TEMPROFIL"

```
(*********************************************************************)
(*                                                                   *)
(* PROGRAMME: "TEMPROFIL"                                            *)
(*                                                                   *)
(* This Pascal programme calculates the temperature distribution     *)
(* in a regressing solid fuel of a hollow, cylindrical               *)
(* combustion chamber. The programme inputs are:                     *)
(*   radii of a cylindrical fuel grain,                              *)
(*   regression rate at every moment,                                *)
(*   Cp, rho and lambda versus temperature (e.g. polynomial          *)
(*   functions),                                                     *)
(*   number of grid points,                                          *)
(*   ablation and ambient temperatures.                              *)
(*                                                                   *)
(* VERSION: 1  PHYSICAL PROPERTIES DEPENDANT ON TEMPERATURE.         *)
(*             REGRESSION RATE DEPENDANT OF TIME.                    *)
(*                                                                   *)
(* LAST UPDATE: APRIL 1987                                           *)
(*                                                                   *)
(*********************************************************************)
(* This program allows for different types of boundary conditions.   *)
(* For each option some changes in procedure INITIALISE are          *)
(* necessary.                                                        *)
(* Some lines should be put outside comment blocks and others inside *)
(* a comment block.                                                  *)
(*                                                                   *)
(* In addition several values may need to be changed (See list of    *)
(* variables below):                                                 *)
(*                                                                   *)
(* K          RHO                                                    *)
(* RI         UML                                                    *)
(* RF         UMR                                                    *)
(*                                                                   *)
(*********************************************************************)
PROGRAM TEMPROFIL(INPUT,OUTPUT);
CONST K=101;
TYPE VECTOR  = ARRAY (.1..K.) OF REAL;
     MATRIX  = ARRAY (.1..K,1..3.) OF REAL;
%PAGE;
(*********************************************************************)
(*                                                                   *)
(* FUNCTION CP                                                       *)
(*                                                                   *)
(* CP as a function of temperature.                                  *)
(*                                                                   *)
(* CP in Joules/Kilogram/Kelvin.                                     *)
(* Y  in Kelvin.                                                     *)
(*                                                                   *)
(*********************************************************************)
FUNCTION CP(Y : REAL) : REAL;
BEGIN
   CP    := 1740;
END;
(*********************************************************************)
(*                                                                   *)
(* FUNCTION LAMBDA                                                   *)
(*                                                                   *)
(* LAMBDA as a function of temperature.                              *)
(*                                                                   *)
(* LAMBDA in Watts/Meters/Kelvin.                                    *)
(* Y      in Kelvin.                                                 *)
(*                                                                   *)
(*********************************************************************)
FUNCTION LAMBDA(Y : REAL) : REAL;
```

```
BEGIN
   LAMBDA := 0.154;
END;
(*************************************************************************)
(*                                                                       *)
(* FUNCTION REGRESSION RATE                                              *)
(*                                                                       *)
(* RATE as a function of time.                                           *)
(*                                                                       *)
(* RATE in Meters/Second.                                                *)
(* Y    in Seconds.                                                      *)
(*                                                                       *)
(*************************************************************************)
FUNCTION RATE(Y : REAL) : REAL;
BEGIN
   RATE  := 0.00015;
END;
(*************************************************************************)
(*                                                                       *)
(* FUNCTION TINIT                                                        *)
(*                                                                       *)
(* TINIT as a fuction of the radial coordinate.                          *)
(*                                                                       *)
(* TINIT in Kelvin.                                                      *)
(* Y     in Meters.                                                      *)
(*                                                                       *)
(*************************************************************************)
FUNCTION TINIT(Y : REAL) : REAL;
BEGIN
   TINIT := 273;
END;
%PAGE;
(*************************************************************************)
(*                                                                       *)
(* PROCEDURE CRITERION                                                   *)
(*                                                                       *)
(* This procedure computes which local temperature gives the             *)
(* severest demands for stability and adapts variables accordingly.      *)
(*                                                                       *)
(* LIST OF VARIABLES.                                                    *)
(*                                                                       *)
(*  K         Total number of grid points.                               *)
(*  U(K)      Initial temperature distribution.     Kelvin               *)
(*  B         Indicates the grid point where the                         *)
(*            inner surface is located.                                  *)
(*  NSLAGEN   The number of times that the vector U                      *)
(*            has to be multiplied by the matrix A                       *)
(*            to get the temperature distribution                        *)
(*            at the moment that the inner boundary                      *)
(*            has moved to the next grid point.                          *)
(*  RHO       Mass density of the fuel.             Kilogrammes/         *)
(*                                                  Squared meters       *)
(*  ALFA      Thermal diffusivity.                  Squared meters/      *)
(*                                                  Second               *)
(*  TIMESTEP  Time it takes to burn a slab of fuel                       *)
(*            with thickness DX with the momentary                       *)
(*            regression rate.                      Seconds              *)
(*  T         Time after ignition.                  Seconds              *)
(*  DT        Time interval.                        Seconds              *)
(*  DX        Distance between two neighbouring                          *)
(*            grid points.                          Meters               *)
(*                                                                       *)
(*************************************************************************)
PROCEDURE CRITERION(VAR U                          : VECTOR;
```

```
                    VAR B,NSLAGEN                    : INTEGER;
                    VAR RHO,ALFA,TIMESTEP,T,DT,DX : REAL);
VAR I                                  : INTEGER;
    TMIN,TMAX,ALFAMA,ALFAMI,DTMA,DTMI : REAL;
BEGIN
   TMIN      := U(.B.);
   TMAX      := U(.B.);
   FOR I     := B TO K DO
   BEGIN
      IF U(.I.) < TMIN THEN TMIN := U(.I.);
      IF U(.I.) > TMAX THEN TMAX := U(.I.);
   END;
   ALFAMA    := LAMBDA(TMAX)/RHO/CP(TMAX);
   ALFAMI    := LAMBDA(TMIN)/RHO/CP(TMIN);
(*********************************************************************)
(* TIMESTEP = Time it takes to burn a slab of fuel with thickness DX *)
(*            with the momentary burning rate.                       *)
(*********************************************************************)
   TIMESTEP := DX/RATE(T);
(*********************************************************************)
(* DT = The delta T should satisfy the following condition:          *)
(*                                                                   *)
(*             DX * DX                                               *)
(*       DT <= --------                                              *)
(*             4 * ALFA                                              *)
(*                                                                   *)
(* When this condition is satisfied the numerical solution of the    *)
(* differential equation is stable.                                  *)
(*********************************************************************)
   DTMA      := 1/(4*ALFAMA/DX/DX+1);
   DTMI      := 1/(4*ALFAMI/DX/DX+1);
   IF DTMA < DTMI THEN
   BEGIN
      ALFA   := ALFAMA;
      DT     := DTMA;
   END
   ELSE
   BEGIN
      ALFA   := ALFAMI;
      DT     := DTMI;
   END;
(*********************************************************************)
(* NSLAGEN = The number of times that the vector U has to be         *)
(*           multiplied by the matrix A to get the temperature       *)
(*           distribution at the moment that the inner boundary      *)
(*           has moved to the next grid point.                       *)
(*********************************************************************)
   NSLAGEN   := 1+TRUNC(TIMESTEP/DT);
   DT        := TIMESTEP/NSLAGEN;
END;
%PAGE;
(*********************************************************************)
(*                                                                   *)
(* PROCEDURE INITIALISE                                              *)
(*                                                                   *)
(* This procedure initialises constants and variables.               *)
(*                                                                   *)
(* LIST OF VARIABLES.                                                *)
(*                                                                   *)
(*  K         Total number of grid points.                           *)
(*  U(K)      Initial temperature distribution.     Kelvin           *)
(*  R(K)      Radius at each grid point.            Meters           *)
(*  A(K,3)    Grid matrix.                                           *)
(*  T         Time after ignition.                  Seconds          *)
```

```
(*  DT        Time interval.                          Seconds          *)
(*  RI        Inner radius.                           Meters           *)
(*  RF        Outer radius.                           Meters           *)
(*  RMOM      Momentary inner radius of fuel grain.   Meters           *)
(*  DX        Distance between two neighbouring                        *)
(*            grid points.                            Meters           *)
(*  TIMESTEP  Time it takes to burn a slab of fuel                     *)
(*            with thickness DX with the momentary                     *)
(*            regression rate.                        Seconds          *)
(*  BETAR     The heat transfer coefficient at the                     *)
(*            Inner boundary.                         Watts/Squared    *)
(*                                                    meters/Kelvin    *)
(*  BETAL     The heat transfer coefficient at the                     *)
(*            outer boundary.                         Watts/Squared    *)
(*                                                    meters/Kelvin    *)
(*  UMR       Ambient temperature at the inner                         *)
(*            boundary.                               Kelvin           *)
(*  UML       Ambient temperature at the outer                         *)
(*            boundary.                               Kelvin           *)
(*  RHO       Mass density of the fuel.               Kilogrammes/     *)
(*                                                    Squared meters   *)
(*  DELTAL    The radiant heat transfer coefficient                    *)
(*            at the inner boundary.                  Watts/Squared    *)
(*                                                    meters/Squared   *)
(*                                                    Kelvin/Squared   *)
(*                                                    Kelvin           *)
(*            DELTAL = SIGMA*EPSW*EPSM*F                               *)
(*    SIGMA   Stefan Boltzmann constant.                               *)
(*            SIGMA  = 5.6703E-08                     Watts/Squared    *)
(*                                                    meters/Squared   *)
(*                                                    Kelvin/Squared   *)
(*                                                    Kelvin           *)
(*    EPSW    Emission coefficient of the wall.       0 < EPSW < 1     *)
(*    EPSM    Emission coefficient of the medium.     0 < EPSM < 1     *)
(*    F       Viewfaktor.                             0 <  F   < 1     *)
(*  ALFA      Thermal diffusivity.                    Squared meters/  *)
(*                                                    Second           *)
(*  TABL      Ablation temperature.                   Kelvin           *)
(*  B         Indicates the grid point where the                       *)
(*            inner surface is located.                                *)
(*  NSLAGEN   The number of times that the vector U                    *)
(*            has to be multiplied by the matrix A                     *)
(*            to get the temperature distribution                      *)
(*            at the moment that the inner boundary                    *)
(*            has moved to the next grid point.                        *)
(*  OPT.      Options for different boundary                           *)
(*            conditions.                                              *)
(*  REGR      Option for a regression rate.                            *)
(*                                                                     *)
(***********************************************************************)
PROCEDURE INITIALISE(VAR U,R                              : VECTOR;
                     VAR A                                : MATRIX;
                     VAR T,DT,RI,RF,RMOM,DX,TIMESTEP,
                         BETAR,BETAL,UMR,UML,RHO,DELTAL,
                         SIGMA,EPSW,EPSM,F,ALFA,TABL      : REAL;
                     VAR B,NSLAGEN                        : INTEGER;
                     VAR OPTAL,OPTAR,OPTB,OPTD,REGR       : BOOLEAN);
VAR I,J  : INTEGER;
BEGIN
   BETAL  := 0;
   BETAR  := 0;
   DELTAL := 0;
   OPTAL  := FALSE;
   OPTAR  := FALSE;
```

```
   OPTB   := FALSE;
   OPTD   := FALSE;
(*************************************************************************)
(* Give dimensions of the cilinder.                                      *)
(*************************************************************************)
   RI     := 0.020;
   RF     := 0.035;
   RMOM   := RI;
   RHO    := 885.05747;
(*************************************************************************)
(* Give temperatures of the surrounding media.                           *)
(*************************************************************************)
   UML    := 600;
   UMR    := 273;
(*************************************************************************)
(* Regression rate is TRUE or FALSE.                                     *)
(*************************************************************************)
   REGR   := FALSE;
(*************************************************************************)
(* DX = Distance between two neighbouring grid points.                   *)
(*************************************************************************)
   DX     := (RF - RI)/(K - 1);
(*************************************************************************)
(* Give the entire slab an initial temperature and calculate the         *)
(* position of each grid point.                                          *)
(*************************************************************************)
   FOR I     := 1 TO K DO
   BEGIN
      R(.I.) := RI + (I - 1) * DX;
      U(.I.) := TINIT(R(.I.));
      FOR J  := 1 TO 3 DO A(.I,J.) := 0;
   END;
(*************************************************************************)
(* Define the boundary conditions.                                       *)
(*                                                                       *)
(* BETA.  is the convective heat transfer                                *)
(*        coefficient.                          Watts/Squared            *)
(*                                              meters/Kelvin            *)
(* DELTA. is the radiant heat transfer                                   *)
(*        coefficient.                          Watts/Squared            *)
(*                                              meters/Squared           *)
(*                                              Kelvin/Squared           *)
(*                                              Kelvin                   *)
(*************************************************************************)
(* OPTION A                                                              *)
(*                                                                       *)
(* At the inner boundary.                                                *)
(*                                                                       *)
(* This option solves convective heat transfer to the inner surface.     *)
(* It depends upon the temperature difference between the medium and     *)
(* the surface of the slab.                                              *)
(*************************************************************************)
(* BETAL  := 100;                                                        *)
(* OPTAL  := TRUE;                                                       *)
(*************************************************************************)
(* OPTION B                                                              *)
(*                                                                       *)
(* At the inner boundary.                                                *)
(*                                                                       *)
(* This option provides a constant temperature at the inner boundary.    *)
(*************************************************************************)
   TABL   := UML;
   U(.1.) := TABL;
   OPTB   := TRUE;
```

```
(*********************************************************************)
(* OPTION D                                                          *)
(*                                                                   *)
(* At the inner boundary.                                            *)
(*                                                                   *)
(* This option provides a radiant heat transfer to the inner surface *)
(* of the slab.                                                      *)
(* It depends upon the temperature difference between the medium and *)
(* the surface of the slab to the fourth power.                      *)
(*********************************************************************)
(* SIGMA  := 5.6703E-08;                                             *)
(* EPSW   := 0.75;                                                   *)
(* EPSM   := 0.83;                                                   *)
(* F      := 1.00;                                                   *)
(* DELTAL := SIGMA*EPSW*EPSM*F;                                      *)
(* OPTD   := TRUE;                                                   *)
(*********************************************************************)
(* OPTION A                                                          *)
(*                                                                   *)
(* At the outer boundary.                                            *)
(*                                                                   *)
(* This option solves convective heat transfer to the outer surface. *)
(* It depends upon the temperature difference between the medium and *)
(* the surface of the slab.                                          *)
(*********************************************************************)
   BETAR  := 0;
   OPTAR  := TRUE;
(*********************************************************************)
(* Set clock to T = 0.                                               *)
(* Put momentary inner radius B on grid point 1.                     *)
(*********************************************************************)
   T      := 0;
   B      := 1;
(*********************************************************************)
(* Determine the parameters to reach a stable solution of the        *)
(* differential equation.                                            *)
(*********************************************************************)
   CRITERION(U,B,NSLAGEN,RHO,ALFA,TIMESTEP,T,DT,DX);
END;
%PAGE;
(*********************************************************************)
(*                                                                   *)
(* PROCEDURE HEADER                                                  *)
(*                                                                   *)
(* This procedure prints the name of the program and the value       *)
(* of some variables.                                                *)
(*                                                                   *)
(*********************************************************************)
PROCEDURE HEADER(VAR NSLAGEN                                : INTEGER;
                 VAR DX,DT,BETAR,BETAL,UMR,UML,RHO,RI,RF,
                     DELTAL,SIGMA,EPSW,EPSM,F,TABL          : REAL;
                 VAR OPTAL,OPTAR,OPTB,OPTD,REGR             : BOOLEAN);
VAR DTXNS : REAL;
BEGIN
   WRITELN('PROGRAMME NAME                : TEMPROFIL');WRITELN;
   WRITELN;WRITELN('EXPLICIT METHOD WITH CENTRAL DIFFERENCES');
   WRITELN('---------------------------------------');WRITELN;
   WRITELN('IMPLEMENTED BY                : JEROEN MIES');WRITELN;
   WRITELN('MATERIAL CONSTANTS');
   WRITELN('------------------');
   WRITELN('INNER RADIUS                  :',RI:11:6,' Meters.');
   WRITELN('OUTER RADIUS                  :',RF:11:6,' Meters.');
   WRITE('LAMBDA AT 273 DEGREES         :',LAMBDA(273):11:6);
   WRITELN(' Watts/Meter/Kelvin.');
```

```
(*****************************************************************)
(*                                                               *)
(* PROCEDURE CALCPROFILE                                         *)
(*                                                               *)
(* This procedure calculates the new temperature distribution    *)
(* by multiplying the matrix 'A' with the vector 'U'.            *)
(*                                                               *)
(* LIST OF VARIABLES.                                            *)
(*                                                               *)
(*  U(K)      Initial temperature distribution.     Kelvin       *)
(*  UU(K)     New temperature distribution.         Kelvin       *)
(*  A(K,3)    Grid matrix.                                       *)
(*  B         Indicates the grid point where the                 *)
(*            inner surface is located.                          *)
(*  NSLAGEN   The number of times that the vector U              *)
(*            has to be multiplied by the matrix A               *)
(*            multiplied by the matrix A to get the              *)
(*            temperature distribution at the                    *)
(*            moment that the inner boundary has                 *)
(*            moved to the next grid point.                      *)
(*  BETAR     The heat transfer coefficient at the               *)
(*            right boundary.                       Watts/Squared*)
(*                                                  meters/Kelvin*)
(*  BETAL     The heat transfer coefficient at the               *)
(*            left boundary.                        Watts/Squared*)
(*                                                  meters/Kelvin*)
(*  UMR       Ambient temperature at the right                   *)
(*            boundary.                             Kelvin       *)
(*  UML       Ambient temperature at the left                    *)
(*            boundary.                             Kelvin       *)
(*  DT        Time interval.                        Seconds      *)
(*  DX        Distance between two neighbouring                  *)
(*            grid points.                          Meters       *)
(*  T         Time after ignition.                  Seconds      *)
(*  RHO       Mass density of the fuel.             Kilogrammes/ *)
(*                                                  Squared meters*)
(*  DELTAL    The radiant heat transfer coefficient              *)
(*            at the inner boundary.                Watts/Squared*)
(*                                                  meters/Squared*)
(*                                                  Kelvin/Squared*)
(*                                                  Kelvin       *)
(*  OPT.      Options for different boundary                     *)
(*            conditions.                                        *)
(*                                                               *)
(*****************************************************************)
PROCEDURE CALCPROFILE(VAR U,UU                          : VECTOR;
                      VAR A                             : MATRIX;
                      VAR B,NSLAGEN                     : INTEGER;
                      VAR BETAR,BETAL,UMR,UML,DT,DX,T,RHO,
                          DELTAL                        : REAL;
                      VAR OPTAL,OPTAR,OPTD              : BOOLEAN);
VAR I,J : INTEGER;
BEGIN
   FOR I := 1 TO NSLAGEN DO
   BEGIN
      UU(.B.) := A(.B,1.)*UML/CP(UML)+U(.B.)+
                 A(.B,2.)*U(.B.)*LAMBDA(U(.B.))/CP(U(.B.))+
                 A(.B,3.)*U(.B+1.)*LAMBDA(U(.B+1.))/CP(U(.B+1.));
      IF OPTAL THEN
      BEGIN
         UU(.B.) := UU(.B.)-2*BETAL/RHO/CP(U(.B.))/DX*DT*U(.B.);
      END;
      IF OPTD THEN
      BEGIN
```

```
      UU(.B.) := UU(.B.)+2*DELTAL*DT/RHO/CP(U(.B.))/DX*
                 (UML*UML*UML*UML-U(.B.)*U(.B.)*U(.B.)*U(.B.));
   END;
   FOR J   := B+1 TO K-1 DO
   UU(.J.) := A(.J,1.)*U(.J-1.)*LAMBDA(U(.J-1.))/CP(U(.J-1.))+
              (1+A(.J,2.)*LAMBDA(U(.J.))/CP(U(.J.)))*U(.J.)+
              A(.J,3.)*U(.J+1.)*LAMBDA(U(.J+1.))/CP(U(.J+1.));
   IF OPTAR THEN
   BEGIN
      UU(.K.) := A(.K,1.)*U(.K-1.)*LAMBDA(U(.K-1.))/CP(U(.K-1.))+
                 A(.K,2.)*U(.K.)*LAMBDA(U(.K.))/CP(U(.K.))-
                 2*BETAR/RHO/CP(U(.K.))/DX*DT*U(.K.)+U(.K.)+
                 A(.K,3.)*UMR/CP(UMR);
   END;
(*********************************************************************)
(* The vector U(.J.) optains the values of the new temperature       *)
(* distribution UU(.J.).                                             *)
(*********************************************************************)
   FOR J := B TO K DO U(.J.) := UU(.J.);
(*********************************************************************)
(* Total elapsed time is adjusted.                                   *)
(*********************************************************************)
   T     := T + DT;
 END;
END;
%PAGE;
(*********************************************************************)
(*                                                                   *)
(* PROCEDURE OUTPUT                                                  *)
(*                                                                   *)
(* This procedure prints the values of the temperatures at each      *)
(* grid point.                                                       *)
(*                                                                   *)
(* LIST OF VARIABLES.                                                *)
(*                                                                   *)
(*  UU(K)    New temperature distribution.        Kelvin            *)
(*  T        Time after ignition.                 Seconds           *)
(*  RMOM     Momentary inner radius of fuel grain. Meters           *)
(*  B        Indicates the grid point where the                     *)
(*           inner surface is located.                              *)
(*                                                                   *)
(*********************************************************************)
PROCEDURE OUTPUT(VAR UU             : VECTOR;
                 VAR T,RMOM,DX,TABL : REAL;
                 VAR B              : INTEGER;
                 VAR OPTB,REGR      : BOOLEAN);
VAR COUNT : INTEGER;
    STOP  : BOOLEAN;
BEGIN
   WRITELN;WRITELN('ELAPSED TIME           : ',T:12:7,' Seconds.');
   WRITELN('MOMENTARY INNER RADIUS : ',RMOM:12:7,' Meters.');
   WRITELN;
   WRITELN('GRID POINT - TEMPERATURE AT GRID LOCATION');WRITELN;
   COUNT := B+1;
   STOP  := FALSE;
   WRITE(B:3,' ',UU(.B.):8:5,' ',B+1:3,' ',UU(.B+1.):8:5,' ');
   IF REGR THEN
   BEGIN
      WRITELN;WRITE('INNER BOUNDARY HAS MOVED TO THE NEXT GRID POINT');
      WRITELN(' AT TIME =',T:11:6,' Seconds.');
      WRITE('INNER BOUNDARY NOW LIES AT                    ');
      WRITELN('       ',RMOM+DX:11:6,' Meters.');
      COUNT := COUNT - 1;
      IF OPTB THEN UU(.B+1.) := TABL;
```

```
   END;
   WHILE (STOP = FALSE) DO
   BEGIN
      COUNT := COUNT + 1;
      IF COUNT MOD 9 = 0 THEN WRITELN;
      WRITE(COUNT:3,' ',UU(.COUNT.):8:5,' ');
(*********************************************************************)
(* The next line may need to be modified as convenient.              *)
(*********************************************************************)
      IF UU(.COUNT+1.) < 0 THEN STOP := TRUE;
      IF (COUNT >= (K-1))  THEN STOP := TRUE;
   END;
   IF (COUNT = (K-1))THEN WRITELN(K:3,' ',UU(.K.):8:5);
   WRITELN;
END;
%PAGE;
(*********************************************************************)
(*                                                                   *)
(* PROCEDURE UPDATE                                                  *)
(*                                                                   *)
(* This procedure updates variables in preparation of the            *)
(* calculation of the new temperature distribution. Latter is the    *)
(* moment that the inner radius has moved to the next grid point.    *)
(* ( B := B+1 ) Then the elapsed time has increased with DT (which   *)
(* is dependent upon the momentary regression rate).                 *)
(*                                                                   *)
(* LIST OF VARIABLES.                                                *)
(*                                                                   *)
(*  U(K)      Initial temperature distribution.     Kelvin           *)
(*  A(K,3)    Grid matrix.                                           *)
(*  RMOM      Momentary inner radius of fuel grain. Meters           *)
(*  DX        Distance between two neighbouring                      *)
(*            grid points.                          Meters           *)
(*  TIMESTEP  Time it takes to burn a slab of fuel                   *)
(*            with thickness DX with the momentary                   *)
(*            regression rate.                      Seconds          *)
(*  T         Time after ignition.                  Seconds          *)
(*  DT        Time interval.                        Seconds          *)
(*  RHO       Mass density of the fuel.             Kilogrammes/     *)
(*                                                  Squared meters   *)
(*  ALFA      Thermal diffusivity.                  Squared meters/  *)
(*                                                  Second           *)
(*  TABL      Ablation temperature.                 Kelvin           *)
(*  B         Indicates the grid point where the                     *)
(*            inner surface is located.                              *)
(*  NSLAGEN   The number of times that the vector U                  *)
(*            has to be multiplied by the matrix A                   *)
(*            multiplied by the matrix A to get the                  *)
(*            temperature distribution at the                        *)
(*            moment that the inner boundary has                     *)
(*            moved to the next grid point.                          *)
(*                                                                   *)
(*********************************************************************)
PROCEDURE UPDATE(VAR U                                    : VECTOR;
                 VAR A                                    : MATRIX;
                 VAR RMOM,DX,TIMESTEP,T,DT,RHO,ALFA,TABL  : REAL;
                 VAR B,NSLAGEN                            : INTEGER);
BEGIN
(*********************************************************************)
(* In case of a moving inner boundary the inner radius must move by  *)
(* one grid point.                                                   *)
(* This happens after the instantaneous time interval TIMESTEP.      *)
(* TIMESTEP is equal to  DT * NSLAGEN.                               *)
(*********************************************************************)
```

```
   RMOM      := RMOM + DX;
   A(.B,1.) := 0;
   A(.B,2.) := 0;
   A(.B,3.) := 0;
   B         := B + 1;
   A(.B,1.) := 0;
   A(.B,2.) := 0;
   A(.B,3.) := 0;
   U(.B.)    := TABL;
(*************************************************************************)
(* Determine the parameters to reach a stable solution of the          *)
(* differential equation.                                                *)
(*************************************************************************)
   CRITERION(U,B,NSLAGEN,RHO,ALFA,TIMESTEP,T,DT,DX);
END;
%PAGE;
(*************************************************************************)
(*                                                                       *)
(* MAIN BODY OF THE PROGRAMME.                                           *)
(*                                                                       *)
(*************************************************************************)
VAR A                                                  : MATRIX;
    U,UU,R                                             : VECTOR;
    B,NSLAGEN                                          : INTEGER;
    T,DT,RI,RF,RMOM,ALFA,DX,TIMESTEP,BETAR,BETAL,
    UMR,UML,RHO,DELTAL,SIGMA,EPSW,EPSM,F,TABL          : REAL;
    OPTAL,OPTAR,OPTB,OPTD,REGR                         : BOOLEAN;
BEGIN
   INITIALISE(U,R,A,T,DT,RI,RF,RMOM,DX,TIMESTEP,BETAR,BETAL,UMR,UML,
              RHO,DELTAL,SIGMA,EPSW,EPSM,F,ALFA,TABL,B,NSLAGEN,
              OPTAL,OPTAR,OPTB,OPTD,REGR);
   HEADER(NSLAGEN,DX,DT,BETAR,BETAL,UMR,UML,RHO,RI,RF,DELTAL,SIGMA,
          EPSW,EPSM,F,TABL,OPTAL,OPTAR,OPTB,OPTD,REGR);
(*************************************************************************)
(*  POSSIBLE CRITERIA FOR STOPPING THE CALCULATIONS:                     *)
(*                                                                       *)
(*  WHILE RMOM <= RF    DO                                               *)
(*  WHILE RMOM <= 0.025 DO                                               *)
(*************************************************************************)
   WHILE T    <= 21    DO
   BEGIN
      CALCMATRIX(R,A,DT,RHO,DX,BETAR,BETAL,B,NSLAGEN,OPTAL,OPTAR,OPTB,
                 OPTD);
      CALCPROFILE(U,UU,A,B,NSLAGEN,BETAR,BETAL,UMR,UML,DT,DX,T,RHO,
                  DELTAL,OPTAL,OPTAR,OPTD);
      OUTPUT(UU,T,RMOM,DX,TABL,B,OPTB,REGR);
      IF REGR THEN UPDATE(U,A,RMOM,DX,TIMESTEP,T,DT,RHO,ALFA,TABL,B,
                          NSLAGEN);
   END
END.
```

APPENDIX 2

Specimen of output of "TEMPROFIL"

PROGRAMME NAME : TEMPROFIL

EXPLICIT METHOD WITH CENTRAL DIFFERENCES

IMPLEMENTED BY : JEROEN MIES

MATERIAL CONSTANTS

INNER RADIUS	: 0.020000	Meters.
OUTER RADIUS	: 0.035000	Meters.
LAMBDA AT 273 DEGREES	: 0.154000	Watts/Meter/Kelvin.
CP AT 273 DEGREES	:1740.000000	Joules/Kilogram/Kelvin.
MASS DENSITY	: 885.057470	Kilogrammes/Squared meters.

DISCRETISATION PARAMETERS

NUMBER OF GRID POINTS	: 101	
DISTANCE BETWEEN TWO GRID GRID POINTS	: 0.000150	Meters.
INITIAL NUMBER OF ITERATIONS	: 19	
INITIAL TIMESTEP	: 1.000000	Seconds.
REGRESSION RATE AT TIME = 0	: 0.000150	Meters/Second.

BOUNDARY CONDITIONS

AT THE OUTER SURFACE:

BETA	: 0.0	Watts/Squared meters/Kelvin.
TEMPERATURE	: 273.000000	Kelvin.
A CONSTANT TEMPERATURE AT THE INNER BOUNDARY OF	: 600.000000	Kelvin.

MOMENTARY INNER RADIUS = 0.020000 METERS.

GRID POINT - TEMPERATURE AT GRID LOCATION

```
  1 600.00000   2 514.41080
INNER BOUNDARY HAS MOVED TO THE NEXT GRID POINT AT TIME =    1.000000 Seconds.
INNER BOUNDARY NOW LIES AT                                   0.020150 Meters.
  2 600.00000   3 437.99794   4 376.69186   5 332.54725   6 304.06679   7 287.64059   8 279.19562
  9 275.33945  10 273.78264  11 273.23006  12 273.05884  13 273.01294  14 273.00241  15 273.00037  16 273.00005  17 273.0000
 18 273.00000  19 273.00000  20 273.00000  21 273.00000  22 273.00000  23 273.00000  24 273.00000  25 273.00000  26 273.0000
 27 273.00000  28 273.00000  29 273.00000  30 273.00000  31 273.00000  32 273.00000  33 273.00000  34 273.00000  35 273.0000
 36 273.00000  37 273.00000  38 273.00000  39 273.00000  40 273.00000  41 273.00000  42 273.00000  43 273.00000  44 273.0000
 45 273.00000  46 273.00000  47 273.00000  48 273.00000  49 273.00000  50 273.00000  51 273.00000  52 273.00000  53 273.0000
 54 273.00000  55 273.00000  56 273.00000  57 273.00000  58 273.00000  59 273.00000  60 273.00000  61 273.00000  62 273.0000
 63 273.00000  64 273.00000  65 273.00000  66 273.00000  67 273.00000  68 273.00000  69 273.00000  70 273.00000  71 273.0000
 72 273.00000  73 273.00000  74 273.00000  75 273.00000  76 273.00000  77 273.00000  78 273.00000  79 273.00000  80 273.0000
 81 273.00000  82 273.00000  83 273.00000  84 273.00000  85 273.00000  86 273.00000  87 273.00000  88 273.00000  89 273.0000
 90 273.00000  91 273.00000  92 273.00000  93 273.00000  94 273.00000  95 273.00000  96 273.00000  97 273.00000  98 273.0000
 99 273.00000 100 273.00000 101 273.00000
```

```
ELAPSED TIME            :     2.0000000 Seconds.
MOMENTARY INNER RADIUS  :     0.0201500 Meters.

GRID POINT - TEMPERATURE AT GRID LOCATION

  2 600.00000   3 528.79972
INNER BOUNDARY HAS MOVED TO THE NEXT GRID POINT AT TIME =    2.000000 Seconds.
INNER BOUNDARY NOW LIES AT                                   0.020300 Meters.
  3 600.00000   4 463.32196   5 407.48506   6 363.23563   7 330.53914   8 307.91695
  9 293.19308  10 284.13850  11 278.85955  12 275.93642  13 274.39900  14 273.63207  15 273.27008  16 273.10887  17 273.0413
 18 273.01471  19 273.00491  20 273.00153  21 273.00045  22 273.00012  23 273.00003  24 273.00001  25 273.00000  26 273.0000
 27 273.00000  28 273.00000  29 273.00000  30 273.00000  31 273.00000  32 273.00000  33 273.00000  34 273.00000  35 273.0000
 36 273.00000  37 273.00000  38 273.00000  39 273.00000  40 273.00000  41 273.00000  42 273.00000  43 273.00000  44 273.0000
 45 273.00000  46 273.00000  47 273.00000  48 273.00000  49 273.00000  50 273.00000  51 273.00000  52 273.00000  53 273.0000
 54 273.00000  55 273.00000  56 273.00000  57 273.00000  58 273.00000  59 273.00000  60 273.00000  61 273.00000  62 273.0000
 63 273.00000  64 273.00000  65 273.00000  66 273.00000  67 273.00000  68 273.00000  69 273.00000  70 273.00000  71 273.0000
 72 273.00000  73 273.00000  74 273.00000  75 273.00000  76 273.00000  77 273.00000  78 273.00000  79 273.00000  80 273.0000
 81 273.00000  82 273.00000  83 273.00000  84 273.00000  85 273.00000  86 273.00000  87 273.00000  88 273.00000  89 273.0000
 90 273.00000  91 273.00000  92 273.00000  93 273.00000  94 273.00000  95 273.00000  96 273.00000  97 273.00000  98 273.0000
 99 273.00000 100 273.00000 101 273.00000
```