EUR 3171.e

EUROPEAN ATOMIC ENERGY COMMUNITY - EURATOM

COSTANZA - CYLINDRICAL, A CYLINDRICAL ONE DIMENSIONAL DYNAMICS CODE FOR LIQUID-COOLED MULTI-CHANNEL NUCLEAR REACTORS

by

A. AGAZZI, G. FORTI and E. VINCENTI

1966

Joint Nuclear Research Center
Ispra Establishment - Italy

Reactor Physics Department
Reactor Theory and Analysis and Research Reactors

This report is on sale at the addresses listed on cover page 4

at the price of FF 8.50 FB 85.— DM 6.80 Lit. 1060 Fl. 6.20

When ordering, please quote the EUR number and the title, which are indicated on the cover of each report.

Printed by Vanmelle S.A.
Brussels, November 1966

EUR 3171.e

COSTANZA — CYLINDRICAL. A CYLINDRICAL ONE DIMENSIONAL DYNAMICS CODE FOR LIQUID-COOLED MULTI-CHANNEL NUCLEAR REACTORS
by A. AGAZZI, G. FORTI and E. VINCENTI

European Atomic Energy Community — EURATOM
Joint Nuclear Research Center
Ispra Establishment (Italy)
Reactor Physics Department
Reactor Theory and Analysis and Research Reactors
Brussels, November 1966 — 66 Pages — 5 Figures — FB 85

The FORTRAN 2 version 3 Code Costanza Cylindrical is a two groups reactor dynamics code in one spatial dimension (radial). Up to 10 different composition zones and 10 cooling channels are allowed. Each channel consists of a fuel zone, gap, cladding and coolant. The thermal

EUR 3171.e

COSTANZA — CYLINDRICAL. A CYLINDRICAL ONE DIMENSIONAL DYNAMICS CODE FOR LIQUID-COOLED MULTI-CHANNEL NUCLEAR REACTORS
by A. AGAZZI, G. FORTI and E. VINCENTI

European Atomic Energy Community — EURATOM
Joint Nuclear Research Center
Ispra Establishment (Italy)
Reactor Physics Department
Reactor Theory and Analysis and Research Reactors
Brussels, November 1966 — 66 Pages — 5 Figures — FB 85

The FORTRAN 2 version 3 Code Costanza Cylindrical is a two groups reactor dynamics code in one spatial dimension (radial). Up to 10 different composition zones and 10 cooling channels are allowed. Each channel consists of a fuel zone, gap, cladding and coolant. The thermal

EUR 3171.e

COSTANZA — CYLINDRICAL. A CYLINDRICAL ONE DIMENSIONAL DYNAMICS CODE FOR LIQUID-COOLED MULTI-CHANNEL NUCLEAR REACTORS
by A. AGAZZI, G. FORTI and E. VINCENTI

European Atomic Energy Community — EURATOM
Joint Nuclear Research Center
Ispra Establishment (Italy)
Reactor Physics Department
Reactor Theory and Analysis and Research Reactors
Brussels, November 1966 — 66 Pages — 5 Figures — FB 85

The FORTRAN 2 version 3 Code Costanza Cylindrical is a two groups reactor dynamics code in one spatial dimension (radial). Up to 10 different composition zones and 10 cooling channels are allowed. Each channel consists of a fuel zone, gap, cladding and coolant. The thermal

resistance of gap and the heat transfer coefficient at every axial level are variable functions of thermodynamic parameters. Inlet coolant temperatures and velocities may be given in input for each channel as tabulated functions of time as well as axial power distribution as function of space (kept constant during transients). Up to 10 delayed neutron groups are considered. Control rods are simulated as diluted thermal poison in each zone which may be varied in time independently according to any given law. In the initialization of each problem, criticality may be searched for by varying poison insertion in any wanted zone.

Feedback reactivities in each zone are considered for both fuel and coolant average temperatures by linear coefficients for p and $\Sigma_{a\ th}$. A free test routine may be used to insert new specifications when any wanted conditions are reached during transients.

resistance of gap and the heat transfer coefficient at every axial level are variable functions of thermodynamic parameters. Inlet coolant temperatures and velocities may be given in input for each channel as tabulated functions of time as well as axial power distribution as function of space (kept constant during transients). Up to 10 delayed neutron groups are considered. Control rods are simulated as diluted thermal poison in each zone which may be varied in time independently according to any given law. In the initialization of each problem, criticality may be searched for by varying poison insertion in any wanted zone.

Feedback reactivities in each zone are considered for both fuel and coolant average temperatures by linear coefficients for p and $\Sigma_{a\ th}$. A free test routine may be used to insert new specifications when any wanted conditions are reached during transients.

resistance of gap and the heat transfer coefficient at every axial level are variable functions of thermodynamic parameters. Inlet coolant temperatures and velocities may be given in input for each channel as tabulated functions of time as well as axial power distribution as function of space (kept constant during transients). Up to 10 delayed neutron groups are considered. Control rods are simulated as diluted thermal poison in each zone which may be varied in time independently according to any given law. In the initialization of each problem, criticality may be searched for by varying poison insertion in any wanted zone.

Feedback reactivities in each zone are considered for both fuel and coolant average temperatures by linear coefficients for p and $\Sigma_{a\ th}$. A free test routine may be used to insert new specifications when any wanted conditions are reached during transients.

EUR 3171.e

EUROPEAN ATOMIC ENERGY COMMUNITY - EURATOM

COSTANZA - CYLINDRICAL, A CYLINDRICAL ONE DIMENSIONAL DYNAMICS CODE FOR LIQUID-COOLED MULTI-CHANNEL NUCLEAR REACTORS

by

A. AGAZZI, G. FORTI and E. VINCENTI

1966

Joint Nuclear Research Center
Ispra Establishment - Italy

Reactor Physics Department
Reactor Theory and Analysis and Research Reactors

CONTENTS

SUMMARY

The FORTRAN 2 version 3 Code Costanza Cylindrical is a two groups reactor dynamics code in one spatial dimension (radial). Up to 10 different composition zones and 10 cooling channels are allowed. Each channel consists of a fuel zone, gap, cladding and coolant. The thermal resistance of gap and the heat transfer coefficient at every axial level are variable functions of thermodynamic parameters. Inlet coolant temperatures and velocities may be given in input for each channel as tabulated functions of time as well as axial power distribution as function of space (kept constant during transients). Up to 10 delayed neutron groups are considered. Control rods are simulated as diluted thermal poison in each zone which may be varied in time independently according to any given law. In the initialization of each problem, criticality may be searched for by varying poison insertion in any wanted zone.

Feedback reactivities in each zone are considered for both fuel and coolant average temperatures by linear coefficients for p and $\Sigma_{a\ th}$. A free test routine may be used to insert new specifications when any wanted conditions are reached during transients.

1. Purpose

The code studies the spatial dynamics of a cylindrical reactor in the two-groups approximation and one space dimension (r). A maximum number of 10 groups of delayed neutrons can be considered. The reactor model is an infinite cylinder, consisting of concentric regions. The regions can be ten at the most. In each region one cooling channel may or may not be taken into consideration.
This vertical channel, typical of the region, consists of one fuel rod, of a gap between rod and cladding, of the cladding and of the coolant corresponding to the rod. A subroutine calculates the temperatures along the radius of the rod, in the cladding and in the coolant, at various levels along the height of the channel. Optionally, instead of a cylindrical fuel rod, the channel can be in slab geometry. In output one can have a print of the temperatures of all the points so as to find the hot spot. The average temperatures in the fuel and in the coolant determine the temperature reaction. The channels are assumed to be thermically isolated from the external moderator, the temperature of which remains constant and has no influence on the reactivity.

The movement of the control rods can be simulated, in each region separately, by a diffused equivalent poison, which varies in time according to a table of values given in input. If some of the rods remain inserted, they can be simulated by a part of the poison which remains constant, while the rest of it varies with time.

The code determines first the critical value of the poison in any wanted region and the critical distribution of the fluxes and precursors. These can be normalized at any arbitrary value of the average thermal flux. The physical characteristics of the core, given in input, should be those of the initial conditions of the reactor. Then the code calculates the critical distribution of the temperatures at the corresponding power. Subsequently the temperatures reactions will be proportional to the differences between the actual temperatures and the initial temperatures.

2. Method

The quasi-linear system of the two-groups diffusion equations is:

$$(1)\quad D_1 \nabla^2 \Psi - \left(\frac{D_1}{\tau} + D_1 B^2\right)\Psi + K_\infty (1-\beta)\Sigma_a \phi + \sum_i^n \lambda_i C_i = \frac{1}{w}\frac{\partial \Psi}{\partial t}$$

$$(2)\quad D_2 \nabla^2 \phi - (\Sigma_a + \Sigma_{poison} + D_2 B^2)\phi + p.\frac{D_1}{\tau}.\Psi = \frac{1}{v}\frac{\partial \phi}{\partial t}$$

Each equation is transformed, with the finite-difference method, into a system of linear equations, by subdividing the radius of the reactor with a mesh of arbitrary meshwidths and the time in constant time steps.

Let be:

$$\frac{1}{w}\frac{d\Psi}{dt} = \frac{1}{w}\frac{\Psi - \Psi^*}{\Delta t} \qquad \text{and} \qquad \frac{1}{v}\frac{d}{dt} = \frac{1}{v}\frac{\phi - \phi^*}{\Delta t}$$

where Ψ^* and ϕ^* are the values of the fast and thermal flux calculated at the preceding time step. The equations (1) and (2) can be transformed into:

$$(3)\quad - D_1 \nabla^2 \Psi + A\cdot\Psi - B\ \phi - C = 0$$

$$(4)\quad - D_2 \nabla^2 \phi + E\cdot\phi - F\ \Psi - G = 0$$

where:

D_1 and D_2 are the fast and thermal diffusion coefficients,

$$A = \frac{D_1}{\tau} + D_1 B^2 + \frac{1}{w\Delta t} \quad ; \quad B = K_\infty (1 - \beta) \Sigma_a$$

$$C = \sum_i^n \lambda_i C_i + \frac{\Psi^*}{w\Delta t}$$

$$E = \Sigma_a + \Sigma_{poison} + D_2 B^2 + \frac{1}{v\Delta t} \qquad F = p \cdot \frac{D_1}{\tau}$$

$$G = \frac{\phi^*}{v\Delta t}$$

All the physical magnitudes, which appear in these coefficients, are functions of time and space. Their values can change at every time step, but remain constant during the time interval Δt.
The space is subdivided into cylindrical annuli each of which is included between two successive mesh-points of the radius.
The coefficients may have different values for each intervals, but are constant throughout the interval. We shall indicate with the index i the coefficients A_i , B_i , G_i corresponding to the interval between point i and point i+1, as indicated in figure.

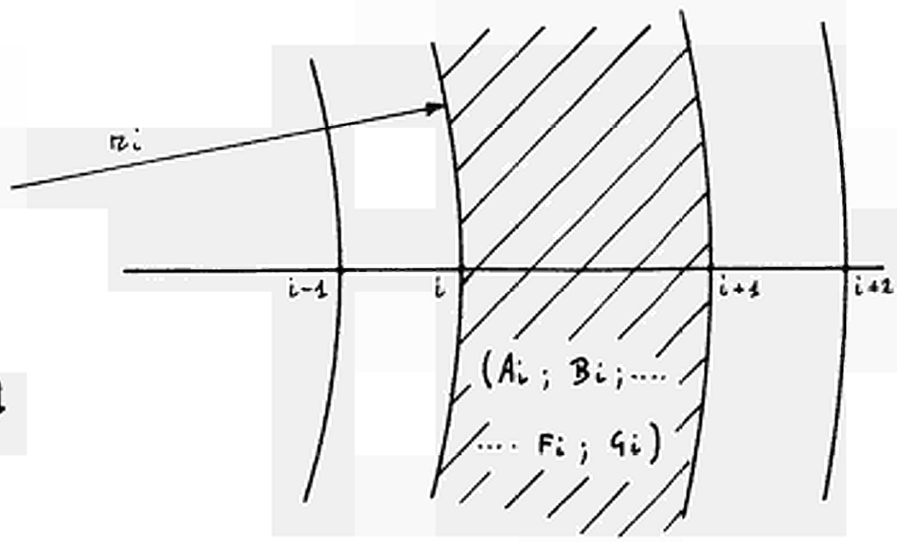


Fig. 1

Integrating the equation (3) and (4) over the annular region corresponding to point i (see fig. 2)

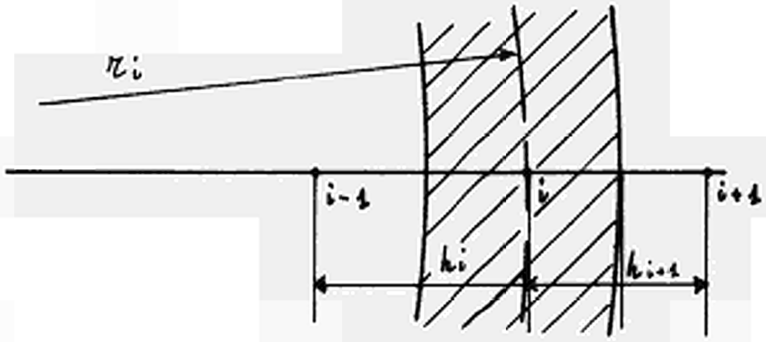


Fig. 2

and substituting to the integrals their finite difference expressions, we will obtain two systems of equations of the type:

5) $- a_{i-1} \Psi_{i-1} - a_i \Psi_{i+1} + e_i \Psi_i - b_i \phi_i - C_i = 0$

6) $- \alpha_{i-1} \phi_{i-1} - \alpha_i \phi_{i+1} + \varepsilon_i \phi_i - \beta_{1i} \Psi_i - \gamma_i = 0$

$(i = 1, 2, \dots n)$

where the coefficients are composed of geometrical terms, which are the same for both equations

$$s_i = \frac{r_i + \frac{h_{i+1}}{2}}{h_{i+1}} \qquad p_i = (r_i - \frac{h_i}{4})\frac{h_i}{2} \qquad q_i = (r_i + \frac{h_{i+1}}{4}) \frac{h_{i+1}}{4}$$

and of the terms containing the physical characteristics of the reactor:

$a_i = D_{1i} \cdot s_i$

$e_i = a_{i-1} + a_i + A_{i-1} \cdot p_i + A_i \cdot q_i$

$b_i = B_{i-1} \cdot p_i + B_i \cdot q_i$

$c_i = C_{i-1} \cdot p_i + C_i \cdot q_i$

$\alpha_i = D_{2\,i} \cdot S_i$

$\varepsilon_i = \alpha_{i-1} + \alpha_i + E_{i-1} \cdot p_i + E_i \cdot q_i$

$\beta_i = F_{i-1} \cdot p_i + F_i \cdot q_i$

$\gamma_i = G_{i-1} \cdot p_i + G_i \cdot q_i$

The two systems of linear equations 5) and 6), each of which is of n equations, where n is the number of the points of the lattice, are solved with an iterative method i.

In the source term $b_i\ \phi_i$ of system 5) we use the vector ϕ_i obtained from the calculation of the preceding time step; the system 5) is then solved in Ψ. The vector Ψ_i is put in the source term $\beta_i\ \Psi_i$ of 6) and a new vector ϕ_i is calculated. This is used in the term $b_i\ \phi_i$ of 5) and so on until it converges.

Each one of the two systems 5) and 6), having a tridiagonal positive definite matrix of coefficients and known terms containing the source terms, can be solved directly by a factorization method.

<u>Note</u>

The boundary condition $\frac{d\Psi}{dr} = 0$, at the axis of the cylindrical core, is obtained by putting the points 1 and 2 of the mesh symmatrically to the axis i.e. $r(1) = r(2)$, and imposing that $\Psi(1) = \Psi(2)$ and $\phi(1) = \phi(2)$. This is obtained by putting $a(1) = 0$ and $\alpha(1) = 0$. In fact the first equation of the systems 5) and 6) is of the type:

7) $-a(1).\Psi(1) - a(2).\Psi(3) + \varepsilon(2).\Psi(2) - b(2).\phi(2) - C(2) = 0$

if 8) $\Psi(1) = \Psi(2)$ then 7) can be written:

9) $a(2).\Psi(3) + [\varepsilon(2) - a(1)].\Psi(2) - b(2).\phi(2) - C(2) = 0$

as $a(1)$ is also contained in $\varepsilon(2)$, 9) can be obtained from 7) by putting $a(1) = 0$. The outer boundary condition is obtained by imposing $\Psi(n) = 0$ and $\phi(n) = 0$.

<u>The subroutine of the channel</u> (CANCIL and CANSL)

This subroutine calculates the temperatures at every point of the radius of the channel at all axial levels.
The radius of the channel is subdivided with a mesh of n mesh-points. The last three points correspond respectively to the coolant, to the cladding and to the gap between cladding and fuel. The remaining n-3 points belong to the fuel rod radius (or to the slab half thickness) and divide the radius in n-3 equal meshes. The radial lattice is the same at every axial level. The height of the channel is subdivided in m equal intervals ($m < 20$). The axial power distribution can be different for every channel, but remains constant during the transient (see Subroutine POT).

The calculation is based on the following equation:

$$K \Delta^2 T + P = C.\rho\frac{dT}{dt} \tag{10}$$

where:

K = thermal conductivity of the media

P = specific power (assumed constant along the radius of the rod)

C = specific heat

ρ = density

The finite-difference expression for the different media are as follows:

a) in the fuel, neglecting the heat conduction in axial direction we have:

$$P_i \cdot \pi(r_j^2-r_{j-1}^2).\Delta Z+K.2\pi.r_{j-1}.\Delta Z.\frac{T_{j-1}^i-T_j^i}{r_{j-1}} - K.2\pi.r_j.\Delta Z.\frac{T_j^i-T_{j+1}^i}{r_j} =$$

$$= \pi(r_j^2 - r_{j-1}^2) \ . \Delta Z.C.\rho.\frac{T_j^i - T_j^{i*}}{\Delta t} \tag{11}$$

where:

i   is the index of the axial level

j   is the index of the radial mesh-point

T_j^{i*} is the temperature at the point (i,j) resulting from the calculation at the preceding time step.

$\Gamma_j = \frac{1}{2} (\Delta r_j + \Delta r_{j+1})$

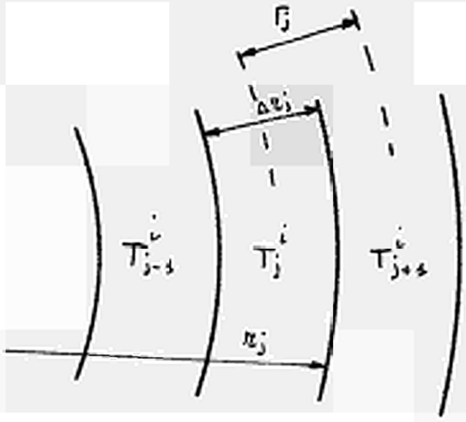


Fig. 3

b) in the cladding and gap we have the same equation with $P_i = 0$

c) in the coolant; considering that there is no heat flow outside the coolant channel we have:

$$h.S.(T_g - T_c) - CO \frac{dT_c}{dZ} = C.\rho.A \frac{dT_c}{dt} \qquad 12)$$

where:

h = heat transfer coefficient between cladding surface and coolant (see Note 2)

S = cladding surface per unit of height

T_g = surface temperature of cladding

T_c = average coolant temperature (*)

C = specific heat of coolant

Q = mass flow

ρ = coolant density

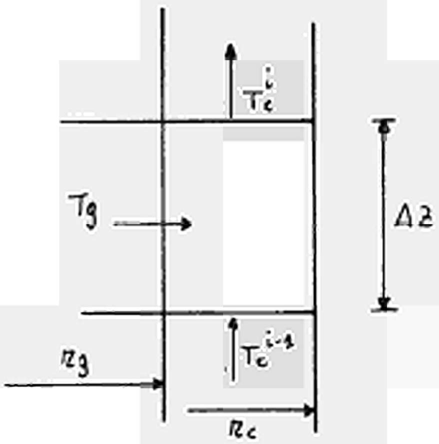
A = section of channel.

The finite-difference expression is:

$$13) \quad h.2\pi.r_g \left[T_g^i - \frac{1}{2}\left(T_c^{i-1}+T_c^i\right)\right] + \frac{CQ}{\Delta Z}\left(T_c^{i-1} - T_c^i\right) =$$

$$= C\,\rho\pi\left(r_c^2 - r_g^2\right) \cdot \frac{1}{\Delta t}\left[\frac{1}{2}\left(T_c^{i-1}+T_c^i\right) - \frac{1}{2}\left(T_c^{i-1*}+T_c^{i*}\right)\right]$$

Fig. 4



Solution of the system

For each value of i (height-level) we have a system of equations of the type:

$$14) \quad a_j\, T_{j-1} + \delta_j\, T_j + c_j\, T_{j+1} = d_j \qquad (j = 1,\ldots n)$$

(*). As it is conceived, this method is valid for forced circulation and turbulent coolant flow.

where the known term d_j contains:

T^{i*}_{j-1} , T^{i*}_{j}, T^{i*}_{j+1} = temperatures at the preceding time step

T^{i-1}_{n} = temperature of the coolant at the level i-1.

The boundary conditions are:

$\frac{dT}{dr} = 0$ at the axis of the rod

$h\frac{dT}{dr} = 0$ at the outer surface of the channel.

The system 5) has a coefficient matrix which is tridiagonal, definite positive. The system can be solved directly with the factorization method. The value T^{i}_{n} will be used to calculate the known term for the level i+1.

T^{1}_{n} is the inlet coolant temperature.

The above described calculation is done by the subroutine CANCIL, in cylindrical geometry. The subroutine CANSL does the same calculation for a slab geometry. In this case the formulas 2) and 4) become:

11bis) $$P_i . \Delta r_j . \Delta Z + K . \Delta Z . \frac{T^{i}_{j-2} - T^{i}_{j}}{r_{j-1}} - K . \Delta Z . \frac{T^{i}_{j} - T^{i}_{j+1}}{r_j} =$$

$$= C . \rho . \Delta r_j . \Delta Z \frac{T^{i}_{j} - T^{i*}_{j}}{\Delta t}$$

13bis) $$h . [T_g - \frac{1}{2} (T^{i-1}_{c} + T^{i}_{c}) + \frac{CQ}{\Delta Z} (T^{i-1}_{c} - T^{i}_{c}) =$$

$$= C . \rho . \Delta r . \frac{1}{\Delta t} [\frac{1}{2}(T^{i-1}_{c} + T^{i}_{c}) - \frac{1}{2}(T^{i-1*}_{c} + T^{i*}_{c})]$$

Note 1

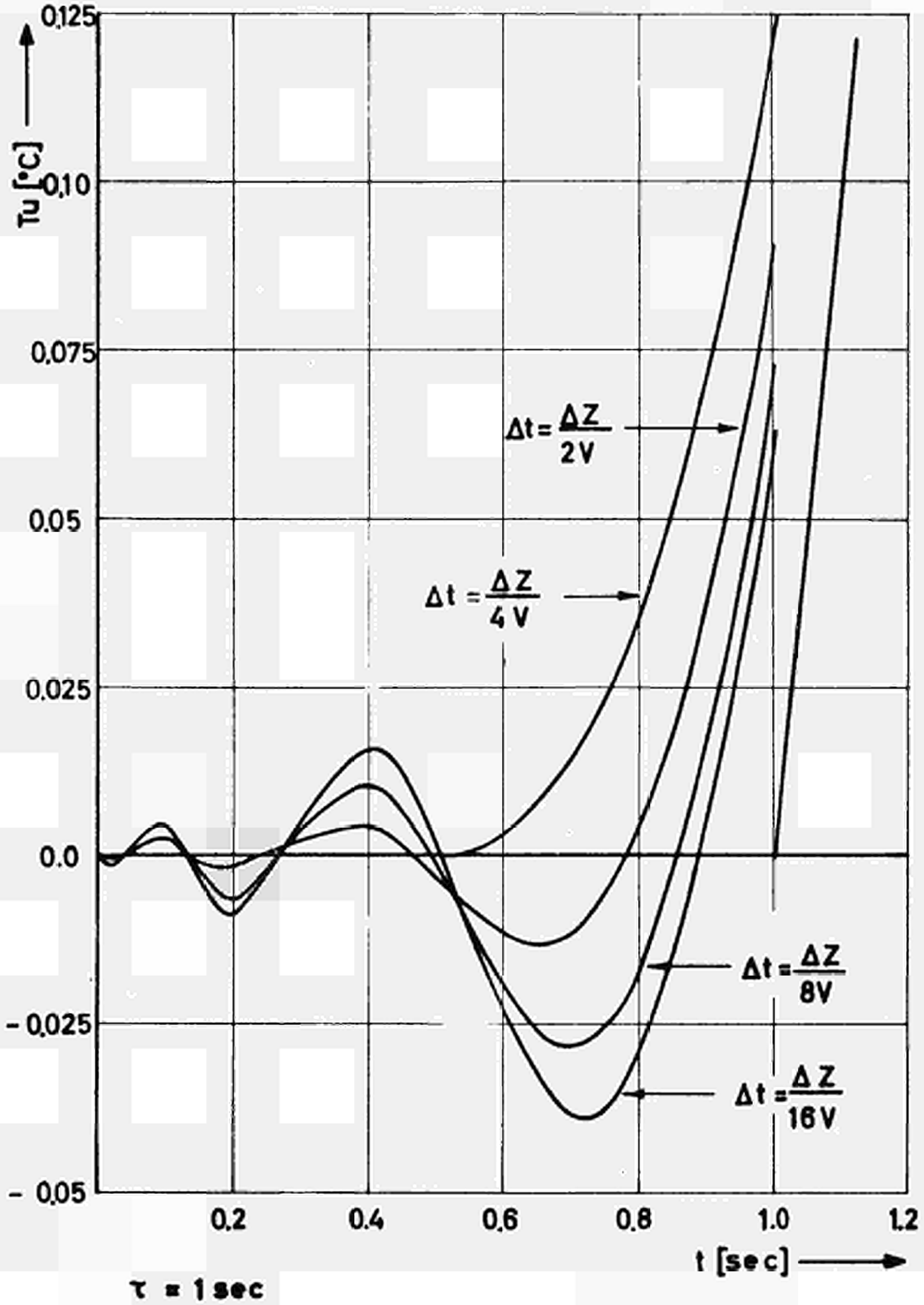
The choice of the mesh width ΔZ and of the time interval Δt should be made considering what follows.

The perturbation of the temperature in the coolant propagates along the channel with the velocity v of the coolant. The outlet temperature has over the inlet temperature a delay $\tau = L/v$ (L = length of the channel).
In the finite-difference representation the delay function is approximated and tends to the exact value with ΔZ tending to zero. For a given ΔZ, when the time interval is $\Delta t < \Delta Z/2.v$ there are oscillations during the initial time τ after the beginning of the perturbation. After the time τ the outlet temperature tends to its exact value with an error which decreases with decreasing Δt. The amplitude of the oscillations however are greater with decreasing Δt.
See fig. 5 for an inlet temperature varying as a ramp and for $\Delta Z = \frac{L}{5}$. To avoid the oscillations it should be

$$\text{(a)} \qquad \Delta t = \frac{\Delta Z}{2.v}$$

It may be advisable to take a ΔZ sufficiently small in order to reduce the error and determine Δt according to $\Delta t = \frac{\Delta Z}{2.v}$.

The time interval of the thermal calculation, determined in this way, results in any case to be much greater than the time interval of the neutronic calculation, which should be of the order of 10^{-3} sec. If, in the transient to be analysed, the power has a great variation and the temperature reaction is very strong, it may be convenient to repeat the thermal calculation more frequently. In that case Δt should be reduced and ΔZ determined according to formula (a) when the oscillations during τ are to be avoided.



Tu [°C]
0,125
0,10
0,075
0,05
0,025
0.0
- 0,025
- 0,05
0.2
0.4
0.6
0.8
1.0
1.2
t [sec]
τ = 1 sec
Δt = ΔZ/2V
Δt = ΔZ/4V
Δt = ΔZ/8V
Δt = ΔZ/16V

FIG. 5

Note 2

According to fig. 3, when we write equations 11 and 13 the temperature T_g^i is not actually the temperature of the surface of the cladding, but the average temperature of the cladding at the level i (T_{n-1}^i).

Therefore in the program the coefficient h is replaced by the following:

$$h' = \frac{1}{\frac{1}{h} + \frac{s}{2Kg}}$$

where:

s = radial thickness of the cladding
Kg = thermal conductivity of the cladding.

The same correction is introduced whenever we operate at the boundary between two different media. In our case we have:

- Fuel-gap

$$h = \frac{1}{\frac{\Delta r_{n-3}}{2.K_u} + \frac{1}{\beta_u}}$$

- Gap-cladding

$$h = \frac{1}{\frac{1}{\beta_g} + \frac{s}{2Kg}}$$

where:

K_u = uranium thermal conductivity
K_g = cladding thermal conductivity
β = heat trasnfer coefficient between uranium (cladding) surface and the medium in the gap
s = cladding thickness

This correction has a big influence when few points are used for the radial mesh.

Program composition

The code consists of the Main Program and 16 Subroutines.

1) Main Program

It reads the input data. It calculates the constant parts of the coefficients, and also the coefficients which are varied by the movement of the control rods. After each thermal calculation it determines the new values of the resonance escape probability and of the absorption cross section in each region according to the temperature coefficients of the fuel and the coolant.

2) Sub. MAT.

It determines all the temperature-dependent coefficients. It is called from MAIN and CRITIC. At the first calculation it determines the coefficients, according to the physical constants given in input, which must correspond to the initial condition of the reactor. Subsequently it is called after each thermal calculation and determines the coefficients with the corrected values of the physical constants.

3) Sub. BARRE.

It is called from MAIN at every time step, and calculates the new value of equivalent poison in each region. An increase in reactivity δK_∞, in any region, may also be simulated by this subroutine with a convenient negative poison.

4) Sub. FLUSSI

It finds the solution of the fast-group system 5) and of the thermal-group system 6) (page 6) separately with a factorization method. It iterates from one group to the other, varying the terms $b_i \phi_i$ and $\beta_i \Psi_i$ and keeping constant all other coefficients. It is called from Main at every time step.

5) Sub. INIZ.

It is called from Main at the beginning. It starts from a uniform distribution of fluxes, equal to the wanted value $\bar{\Phi}$ of the average thermal flux. It calls Subroutine Flussi a given number of times. Each time the fluxes are normalized in order to have the wanted $\bar{\Phi}$. All the coefficients of the system 5), 6) are kept constant (the time does not progress). The physical distribution of fluxes is thus obtained. Then it calculates the equilibrium precursor concentration corresponding to the obtained thermal flux distribution.
The value of $\bar{\Phi}$ has no influence on the thermal calculation, it is only a reference value, which can be put equal 1 for simplicity's sake.

6) Sub. CRITIC.

It is optionally called from Main at the beginning. It determines the critical value of the poison in any wanted region. It starts from a first-guess poison. By calling Flussi a given number of times, it determines the period, then changes the poison and repeats the operation until the reciprocal of the period is smaller than a given convergence parameter. The fluxes are then normalized at the wanted value of $\bar{\Phi}$.

7) Sub. STAMPA.

It prints the neutronics results. Three options are available:

1) TØ = Time ;
   PM1 = Average fast flux ;
   PM2 = Average thermal flux ;
   PER = Reactor period.

2) TØ = Time ;
   IT = Number of time steps ;
   PER = Reactor period ;

PINT = $\int_0^t \phi$ (t) dt

INT = Number of inner iteration of the Subroutine Flussi ;

R = Distance of each mesh point from the axis of the cylinder ;

P1 and P2 = Fast and thermal flux at each mesh point ;

C1, C2, ... , C10 = Precursor concentration at each mesh point.

3) TO ; PM1 ; PM2 ; PER as in option 1)

VELENI BARRE = Poison introduced by the control rod in each region ;

FLM1 and FLM2 = Fast and thermal flux, averaged in each region.

8) Sub. TEST.

This is a free Subroutine by means of which it is possible to introduce some new statement, without altering the rest of the program. It is called from Main at the first time step and subsequently at every time step if the index KTE >0 and has the same Common and Dimension as the Main program. The free memories of the vector DATA may be used to introduce any new parameter (for instance, it may be used to command the scram, with any given delay when $\bar{\phi}$ has reached a given threshold).

9) Sub. DCAN.

It reads all the input DATA relative to the thermal characteristics of the channels and calculates all the coefficients of the system 14) which do not change during the transient.

10) Sub. CANCIL. (for cylindrical geometry)
Sub. CANSL (for slab geometry)

Both subroutines calculate the variable coefficients. They solve the system of equations 14) by a factorization method for all the points of the channel.

11) Sub. VINIZ.

It is called from the subroutines of number 10) and calculates at every thermal time step the values of the inlet coolant temperature and of the coolant velocity. These magnitudes can vary as a step, or as a ramp, or as any function of time. In this last case it is possible to give up to ten values of the function at different instants. The subroutine makes a linear interpolation at every time step.

12) Sub. INTEGR.

It calculates the average temperatures of the fuel, cladding, and coolant to be used for the feed-back.

13) Sub. POT.

It reads the form factors of the power axial distribution of each channel and calculates at every time step the power corresponding at every axial level of each channel.

14) Sub. HTC.

This routine is optionally used to vary the heat transfer coefficient from cladding to coolant, according to the following formulae

(density) $$\rho_{coolant} = a_1 + a_2 \frac{1}{T_c} + a_3 \frac{1}{T_c^2} + a_4 \frac{1}{T_c^3}$$

(specific heat) $$C_{pcoolant} = a_5 + a_6 T_c$$

(viscosity) $$\eta_{coolant} = a_7 + a_8 \frac{1}{T_c} + a_9 \frac{1}{T_c^2} + a_{10} \frac{1}{T_c^3}$$

(thermal conductivity) $$\lambda_{coolant} = a_{11} + a_{12} T_c$$

(T_c = °C)

$$P_r = \frac{\eta \cdot C_p}{\lambda} \qquad Reyn = \frac{\rho\, v\, D}{\eta}$$

$$h = \frac{\eta}{D}\, a_{18}\, Reyn^{(a^{19})} \cdot P_r^{(a^{20})}$$

The a_i coefficient and the hydraulic diameter D are given in input.
The routine is called from CANCIL and CANSL, at every time step, for each axial level, in every channel.
(T_c = Temperature of coolant in °C).

15) Subroutine GAP (TPUR, TPS, TPG, RGAP)

This is a free subroutine, optionally used to vary the thermal resistance of the gap between fuel and cladding during the transient. The subroutine is called from CANCIL or CANSLAB, at every thermal calculation and at every axial level of the channel. One can build the subroutine to calculate the thermal resistance RGAP, according to his own formulae as a function of:

TPUR = average temperature of the section of the fuel rod at the corresponding axial level (°C)
TPS = fuel surface temperature at that axial level (°C)
TPG = cladding temperature at that axial level (°C).

16) Subroutine GAPI2 (TMA, TPG, ALF, BET, RGAP, CØST)

It is a free subroutine, to be used in the same occasion of GAP, which evaluates the thermal resistance RGAP of the gap in the initial static condition as a function of:

TMA = TPUR - TPS
TPG
ALF
BET = constants appearing in the relation

TPS - TPG = ALF + BET * RGAP

(The term ALF comes out from the fact that in the finite

difference representation the thermal resistance of half interval in fuel and cladding are added to the gap resistance).
They are calculated in CANCIL or CANSL for every level of the channel.

COST is a free constant; in the routine included as example COST is 4.185 if calories units are used and 1 if WATT units are used.

In the enclosed listing there is an example of these subroutines.

Input

It is possible to calculate several problems in one run. Each problem has to be preceded by a title card.

Title card

The columns 1 - 6 must be left blank, only in the last problem they must contain some positive number .

The columns 7 - 72 contain any alpha-numeric information.

A vector of 3500 memory positions DATA(1), DATA(2), ... DATA(3500) contains all the data. Since entire groups of memory positions are zero, it is possible to read only the sets of significant data. Each set must be preceded by a cart containing the integers $K_{i,1}$ and $K_{i,2}$ defining the number of decimal data to be read DATA ($K_{i,1}$); DATA ($K_{i,1}$+1);... ... DATA($K_{i,2}$).

In the last set the second integer must be preceded by the sign - (minus).

DATA(1) = (DELT) Integration time interval (DELT = 0.001 sec gives good results in normal cases).

DATA(2) = (TØ) Initial value of time.

DATA(3) = (IT) Initial time step number.

DATA(4) = (4 < IMAX < 100) Number of mesh points.

DATA(5) = (1 < NREG < 10) Number of regions

DATA(6) = (1 < NRIT < 10) Number of groups of delayed-neutron precursors.

DATA(7) = (INTE ≈ 10) Maximum number of inner iterations.

DATA(8) = (IDST ≈ 20) Number of initial iterations necessary to give the fluxes, their initial shapes and normalization.

DATA(9) = (ITCR = 20) Number of iterations to calculate the period corresponding to a given poison. If DATA(9) = zero, the critical search will not be made.

DATA(10)= (BU = $(\frac{\pi}{H})^2$) Value of axial buckling, common to all regions and groups.

DATA(11)= To be left blank

DATA(12)= (EPS = 0.0001) Convergence criterium for the inner iterations (EPS = relative variation of the average thermal flux).

DATA(13)= To be left blank.

DATA(14)= (SI) Normalization value of the average thermal flux.

DATA(15)= To be left blank.

DATA(16)= (KPC) The thermal calculation will be made every KPC^{th} neutronic time step (see note of page 11).

DATA(17)= (NCAN) Number of channels. If NCAN = 0 the thermal calculations will be bypassed, their corresponding input data ignored.

DATA(18)= (KMA1) The print of all the end temperatures will be made at each KMA1 thermal calculation.

DATA(19)= (KTME1) The print of the average temperatures in the fuel, cladding and coolant, will be made at each KTME1 thermal calculation.

DATA(21)......(30) = PØWER(M) (M = 1, 2, ...NREG). Thermal powers of each channel at the initial condition (cal./sec) or (WATT) (the thermal units can be calories or watt, providing the same unity be used for all the input data).

DATA(31)......(40) = (BETA(J) = β_j) (J=1,2,..NRIT) Yield of delayed neutron precursors per fission.

DATA(41)....(50) = (DL(I)= λ_i) (I=1,2,...NRIT) Decay constants of precursors.

DATA(61)....(71) = Interface indexes. Index 1 must always be given. Points 1 and 2 are symmetric to the cylinder-axis.

DATA(81)....(200) Contain 12 data for each region in the following order:

1) D fast
2) D thermal
3) K = $\nu \Sigma_f / \Sigma_a$
4) τ = Fermi age
5) Source: average flux in the region ($\frac{n}{cm^2 sec}$) due to the source
6) Σ_{ab} thermal
7) (SPR = $_{poison}$) This is not the poison of the movable control rods, it is an additional constant poison. It should be left blank when criticality search is requested.
8) To be left blank
9) W fast group neutron velocity
10) V thermal group neutron velocity
11) P = resonance escape probability
12) Zero if the region does not contain a channel. 1 if the region contains a channel.

DATA(201)....(300) Distances of the mesh points from the cylinder-axis. They are arbitrary, except for R(1) = = -R(2).

Control rod movement

The control rod movement is simulated in each region by an equivalent distributed poison variable in time. The function $\Sigma_{poison}(t)$ of each region is given in input in tabulated form; n points of the curve $(n < 30)$ must be conveniently chosen. The program interpolates linearly between the points.

DATA(301) (900) is a vector of 600 memories which contains all the points of all the curves of every region. It is subdivided in blocks of 60 memories, one for each region.

First region	DATA(301), (332),...(300+n)	Times of the chosen points
	DATA(331), (332),...(330+n)	Corresponding values of Σ_p.
Second region	DATA(361), (362),...(360+n)	
	DATA(391°, (392),...(390+n)	
Tenth region	DATA(841)......(840+n)	
	DATA(871)......(870+n)	

n, the number of points of the curve $\Sigma_p(t)$, may be different from region to region. If a region does not contain rods, the corresponding block of memory must be left blank.

DATA(300 + (i-1) * 60+1) where (i = 1, 2, ...NREG), must always be zero because it corresponds to the initial instant of the transient.

If a step of Σ_p has to be introduced at the time t, in the region number i, one must give the following data:

DATA(301 + (i-1) *60) = t - DELTA
DATA(302 + (i-1) *60) = t
DATA(331 + (i-1) *60) = 0
DATA(332 + (i-1) *60) = Σ_p

It is impossible to introduce a step at the very beginning of the transient t = 0, it can be introduced at the time t = DELTA at the earliest.

DATA(1301)......(1310) = CØU(M) (M=1,2,...NCAN)
Temperature coefficients of the fuel. They affect the resonance escape probability of each region containing a channel.
$P = P_o + P_o * CØU * (\bar{T}_{fuel} - \bar{T}_{o\ fuel})$

DATA(1311)......(1320) = CØR(M) (M=1, 2,...NCAN)
Temperature coefficient of coolant. They affect the absorption cross-section of each region containing a channel.
$\Sigma_{poison} = \Sigma_{o\ poison} - \Sigma_{absorption}\ CØR * \bar{T}_{coolant} - \bar{T}_{o\ coolant}]$

<u>Criticality search</u> (only if DATA(9) > 0)

a) DATA(1601)......(1610) = KV(I) Order number of the region, to be written if the region is poisoned; to be left blank if not.

b) DATA(1611) = (SPRG) Second guess of poison. It must be the same for all the checked regions. The first guess is zero.

c) DATA(1612) = DAPF = 10^{-4}) Convergence criterium for poison criticality search. The absolute value of the reciprocal period will be < DAPF.

d) DATA(1613) = (LF= 20) Maximum number of iterations for critical search.

<u>Print orders</u>

DATA(1851)......(1994)

Up to 24 cards (n=0, 1, 2..., 23) specifying the print orders may be given. Each card contains six data as follows:

DATA(1850+6.n+1) = (KTP) multiple of I2P. It is the number of time steps to be made until a new order of print will be read (next card).

DATA(1850+6.n+2) = (I1P) Each I1P time step a; print of IS1 type will be made.

DATA(1850+6.n+3) = (IS1) = 1, 2, 3 The print is of the type 1, or 2, or 3 accordingly (see Subroutine STAMPA page 16)

DATA(1850+6.n+4) = (I2P) multiple of I1P. Each I2P time step a print of IS2 type will be made.

DATA(1850+6.n+5) = (IS2) = 1, 2, 3 The print will be of type 1 or 2 or 3 accordingly (see Subroutine STAMPA page 16).

DATA(1850+6.n+6) = to be left blank.

The calculation stops at the time step, corresponding to the last given value of DATA(1850+6.n+1) = KTP and the following case will be started.

<u>Specification of the axial power distribution</u>

The axial power distribution may be different for each channel. It remains constant during the transient. DATA(2001)....(2200) is a vector of 200 memories subdivided in blocks of 20 memories, one block for each channel.

DATA(2000+n+(i-1) * 20) are the ordinates of the curve of the power distribution of the ith channel. These ordinates must be taken in correspondance of the medium points of the n axial zones, into which the channel has been subdivided ($n < 20$). Only the relative values of these ordinates are meaningful, because the program normalizes their absolute values at the power of the channel at every instant.

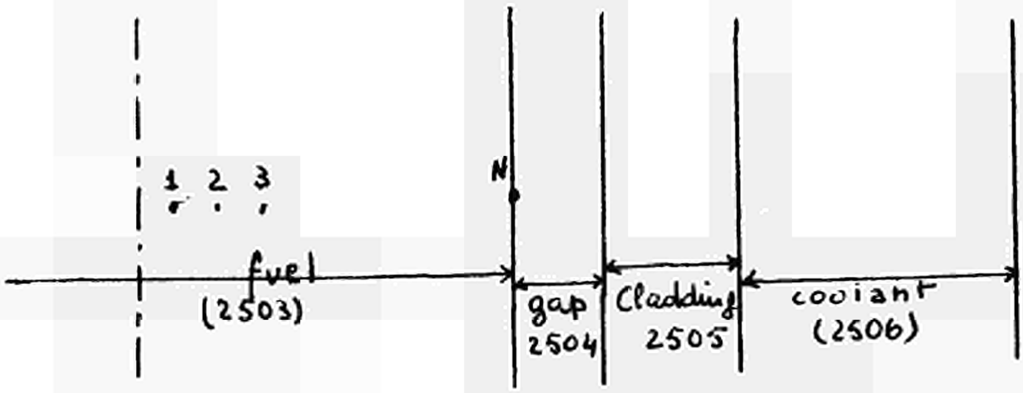
Data of the channels

If NCAN = DATA(17) = 0 the thermal calculations will be bypassed and their corresponding data ignored.

DATA(2500) = 1.0 if watt unities are used; 4.185 if calories are used. This datum is common for all the channels.

DATA(2501)......DATA(2530)

DATA(2501) = N ($N \leq 7$) Number of meshes into which the radius of the fuel rod (or the half thickness of the fuel slab) has been subdivided.



DATA(2502) = (NSV < 20) number of axial zones

DATA(2503) = Diameter of the fuel rod (thickness of the fuel slab) (cm)

DATA(2504) = Thickness of the gap (cm)

DATA(2505) = Thickness of the cladding (cm)

If there is no gap, the mesh-point N+2 is at the outer surface of the cladding and the mesh-point N+1 falls within the cladding.

DATA(2504) + DATA(2505) = Thickness of the cladding

DATA(2508) = DATA(2509)

DATA(2512) = DATA(2513)

DATA(2516) = DATA(2517)

DATA(2506) = Thickness of the coolant layer corresponding to the fuel element (cm)

DATA(2507) = Density of the fuel (g/cm^3) (g/cm^3)

DATA(2508) = Desnisty of the gas filling the gap

DATA(2509) = Density of the cladding

DATA(2510) = Density of the coolant.

DATA(2511) = Specific heat of fuel $(\frac{cal}{g\ °C})$ or $(\frac{Joule}{g\ °C})$

DATA(2512) = Specific heat of gas filling the gap

DATA(2513) = Specific heat of cladding

DATA(2514) = Specific heat of coolant.

DATA(2515) = Conductivity of fuel $(\frac{cal}{cm\ sec\ °C})$ or $(\frac{WATT}{cm\ °C})$

DATA(2516) = Conductivity of gas filling the gap

DATA(2517) = Conductivity of cladding

DATA(2518) = (VLR) Heat transfer coefficient between cladding surface and coolant. $(\frac{cal}{sec\ cm^2 °C})$ or $(\frac{WATT}{cm^2\ °C})$

If DATA(2516) = 0 the Subroutine GAP is called.

If DATA(2518) = 0 the Subroutine HTC is called.

DATA(2519) = Length of channel (cm)

DATA(2520) = Slab width (only in case of slab geometry; blank in case of cyclindrical geometry) (cm)

DATA(2521) = Initial value of coolant inlet temperature (°C) to be left blank if this temperature is given in tabulated form.

DATA(2522) = Only if DATA(2521) $\neq$ 0. Step of coolant inlet temperature (°C)

DATA(2523) = Only if DATA(2521) $\neq$ 0; value of $\frac{dT}{dt}$ in case of a ramp of coolant inlet temperature $(\frac{°C}{sec})$

DATA(2524) = Initial value of coolant velocity $(\frac{cm}{sec})$. To be left blank if the velocity is given in tabulated form

DATA(2525) = Only if DATA(2524) $\neq$ 0; step of coolant velocity $(\frac{cm}{sec})$

DATA(2526) = Only if DATA(2524) $\neq$ 0; value of $\frac{dv}{dt}$ in case of a ramp of coolant velocity $(\frac{cm}{sec^2})$

DATA(2530) = (IT1PØ) 1 for cylindrical geometry
2 for slab geometry.

DATA(2531)...(2560)
DATA(2561)...(2590)
DATA(2771)...(2800)

In the same order as before the data for the second, third, ...10th channel (Some regions may not contain a channel)

Variable inlet temperature (only if DATA(2521+(i-1)*30) = 0)

The coolant inlet temperature, of the channel number i(i < 10), can be whatever function of time $T_{inlet}(t)$ and can be given in input in tabulated form.

n points, of the curve $T_{inlet}(t)$, must be conveniently chosen (1 < n < 9). The program interpolates linearly between the points.

DATA(2900+n+(i-1)*10) are the times corresponding to the points of the curve of the ith channel (sec).
DATA(2900+1+(i-1) * 10), initial instant of the transient, and DATA(2900+10+(i-1) * 10) must always be zero or left blank. The last time given must be multiple of the thermal time step.

DATA(2800+n+(i-1) * 10) are the values of the inlet temperature at the selected points. DATA(2800+1+(i-1)* 10) is the initial value.

After the last point of the curve, the inlet temperature will maintain the value given at the last point.

<u>Variable coolant velocity</u> (only if DATA(2524+(i-1) *30)=0)

Same as for the variable inlet temperature where:

DATA(3100+n+(i-1) * 10) are the times

DATA(3000+n+(i-1) * 10) are the velocities.

<u>Variable value of heat transfer coefficient from cladding to coolant of the ith channel</u>

(Only if DATA(2518+(i-1)*30) = 0)

DATA 3200+20(i-1)+1 = a_1			$(g.cm^{-3})$
DATA 3200+20(i-1)+2 = a_2			$(g.cm^{-3}{\circ}C)$
DATA 3200+20(i-1)+3 = a_3			$(g.cm^{-3}{\circ}C^2)$
DATA 3200+20(i-1)+4 = a_4			$(g.cm^{-3}{\circ}C^3)$
DATA 3200+20(i-1)+5 = a_5	$(cal.g^{-1}{\circ}C^{-1})$	or	$(J.g^{-1}.{\circ}C^{-1})$
DATA 3200+20(i-1)+6 = a_6	$(cal.g^{-1})$	or	$(J.g^{-1})$
DATA 3200+20(i-1)+7 = a_7			$(g.cm^{-1}.sec^{-1})$
DATA 3200+20(i-1)+8 = a_8			$(g.cm^{-1}.sec^{-1}.{\circ}C)$
DATA 3200+20(i-1)+9 = a_9			$(g.cm^{-1}.sec^{-1}.{\circ}C^2)$
DATA 3200+20(i-1)+10= a_{10}			$(g.cm^{-1}.sec^{-1}.{\circ}C^3)$
DATA 3200+20(i-1)+11= a_{11}	$(cal\ sec^{-1}cm^{-2}{\circ}C^{-1})$	or	$(W.cm^{-2}\ {\circ}C^{-1})$
DATA 3200+20(i-1)+12= a_{12}	$(cal\ sec^{-1}cm^{-2})$	or	$(W.cm^{-2})$
DATA 3200+20(i-1)+17= hydraulic diameter			(cm)
DATA 3200+20(i-1)+18= a_{18}			
DATA 3200+20(i-1)+19= a_{19}			
DATA 3200+20(i-1)+20= a_{20}			

The coefficients a are defined in subroutine HTC.

References

1) "Difference Methods for Initial-Value Problems" Richtmyer, R.D. - Interscience Publishers, Inc., New York

2) "Matrix Iterative Analysis" Varga, R.S. - Prentice-Hall, Inc.

3) "Finite Difference Method for Solving the Spatio-Temporal Diffusion Equation in the Two-Group Approximation" Monterosso, R. Vincenti, E. - EUR 596.e

4) "Costanza - A Numerical Code for the Study of the Reactor Spatial Dynamics in Two Groups" Vincenti E., Monterosso R., Agazzi A. - EUR 2103.e

5) "A Digital Program for the Study of Reactor Dynamics (Part I - One Space Dimension)" Vincenti E., Guerri L. (non available)

```
C                    COSTANZA   -   CILINDRICO
C
C
      DIMENSION DR(15,10),RU(10),TMU(10,21)
      DIMENSION DATA(3500),ITIPO(10),MZ(10),MSV(10),RS(10),DX(10),VLR(10
     1),R(15,10),RO(15,10),SC(15,10),VL(15,10),TP(15,21,10),TMED(10,4)
      DIMENSION VR(10),FLIM2(10),POWER(10),PI(10),COU(10),CCR(10),TMUI(1
     10),TMC1I(10),TMC2I(10),TCI(10),ICAN(10),SPRI(10)
      DIMENSION D1(10),D2(10),VK(10),TAU(10),SA(10),SPR(10),CS(10),X(100
     1),T(100),          P1(100),P2(100),SOR(10)  ,BE(100),VOL1(100),VOL2(10
     20),VOL(100),AL2(100),AB2(100),EP2(100),TN1(100),TN2(100),
     3DER1(100),DER2(100),P1S(100),P2S(100),          SP(10),I1(10),I2(10
     4),W(10),V(10),AL1(100),BL1(100),BL2(100),          DEM1(100),DEM2(1
     500),           EP1(100),BI1(100),BI2(100),
     6         VIM(10),SORM(100),BE1(100),BE2(100),BETA(10),DL(10),DLDT(1
     70),DETA(10),CMC(10,100),COC(10,100),CCM(10,100),           C(10,1
     800),KV(10),AL22(100),P(10)
      DIMENSION ALFA(11)
      COMMON MZ,MSV,DATA,TP,TMED,VL,SC,RO,DR,DX,VLR,RS,R,RU,TMU
      COMMON KBA,KTE,KBAR,KS
      COMMON NREG,D1,D2,VK,TAU,SA,SPR,CS,X,T,C,P1,P2,P1S,P2S,SOR,IMAX,DL
     1DT,CL,DELT,IM1,NK,BE,VOL1,VOL2,VOL,VOLT,AL2,AB2,EP2,TN1,DLD,DER1,T
     2N2,DER2,CCC,IT,TO,KBA,KBI,SP,EPN,I1,I2,W,V,AL1,BL1,BL2,DLM,DEM1,DE
     3M2,CMC,BETA,EP1,BI1,BI2,DLD,PMPT,PM2,PM1,INT,EPS,INTE,PER,VIM,PINT
     4,P,SORM,BU,BUT,BE1,BE2,DETA,NRIT,SBETA,IDST,SI,ITCR,KV,SPRG,DAPF,L
     5F,L1,SP1,REP,SPCR,COM,AL22
      PINT =0.
  110 READ INPUT TAPE 5,20,LAST,ALFA
   20 FORMAT(I6,11A6)
      WRITE OUTPUT TAPE 6,23
   23 FORMAT (1H1,35X,      20HCOSTANZA  CILINDRICO//)
      WRITE OUTPUT TAPE 6,22, ALFA
   22 FORMAT (1H0,30X,12A6/////)
  100 READ INPUT TAPE 5,101,K1,K2,(DATA(I),I=K1,K2)
  101 FORMAT (2I12/(6E12.8))
      WRITE OUTPUT TAPE 6,102,(I,DATA(I),I=K1,K2)
  102 FORMAT (6(I5,E14.6))
      IF(K2)103,103,100
  103 CONTINUE
      PINT=0 0
      DELT=DATA(1)
      TO=DATA(2)
      IT=DATA(3)+0.0001
      IMAX=DATA(4)+0.0001
      NREG=DATA(5)+0.0001
      NRIT=DATA(6)+0.0001
      INTE=DATA(7)+0.0001
      IDST=DATA(8)+0.0001
      ITCR=DATA(9)+0.0001
      BU=DATA(10)
      BUT=DATA(11)
      EPS=DATA(12)
      EPN=DATA(13)
      SI=DATA(14)
      KPC=DATA(16)+0.1
```

```
      DT=DELT*FLOATF(KPC)
      NCAN=DATA(17)+0.1
      KMA1=DATA(18)+0.1
      KTME1=DATA(19)+0.1
      KTE=1
      KMAP=0
      KTMED=0
      KCAN=0
      DO 104 I=1,NRIT
      BETA(I)=DATA(I+30)
  104 DL(I)=DATA(I+40)
      DO 105 I=1,NREG
      I1(I)=DATA(I+60)+0.0001
      I2(I)=DATA(I+61)+0.0001
  105 CONTINUE
      IDF=80
      DO 107 M=1,NREG
      D1(M)=DATA(IDF+1)
      D2(M)=DATA(IDF+2)
      VK(M)=DATA(IDF+3)
      TAU(M)=DATA(IDF+4)
      SOR(M)=DATA(IDF+5)
      SA(M)=DATA(IDF+6)
      SPR(M)=DATA(IDF+7)
      CS(M)=DATA(IDF+8)
      W(M)=DATA(IDF+9)
      V(M)=DATA(IDF+10)
      P(M)=DATA(IDF+11)
      ICAN(M)=DATA(IDF+12)+0.1
      IDF=IDF+12
  107 CONTINUE
      DO 108 I=1,IMAX
      AB2(I)=0.
  108 X(I)=DATA(I+200)
      X(1)=-X(2)
      P1(IMAX)=0.
      P2(IMAX)=0.
      P1(1)=P1(2)
      P2(1)=P2(2)
      SBETA=0.
      DO 80 K=1,NRIT
      DLDT(K)=DL(K)*DELT
      DETA(K)=BETA(K)*DELT
   80 SBETA=SBETA+BETA(K)
      IM1=IMAX-1
      NK=IMAX-2
      DO 16 I=2,IMAX
      BE(I)=(X(I)+(X(I+1)-X(I))/2.)/(X(I+1)-X(I))
      VOL1(I)=(X(I)+(X(I+1)-X(I))/4.)*(X(I+1)-X(I))/2.
   16 VOL2(I)=(X(I)-(X(I)-X(I-1))/4.)*(X(I)-X(I-1))/2.
      VOL(2)=((X(2)+X(3))/2.)**2*3.1416
      DO 17 I=3,IM1
   17 VOL(I)=(VOL1(I)+VOL2(I))*6.2832
      VOLT=0.
      DO 8 I=2,IM1
    8 VOLT=VOLT+VOL(I)
```

```
      VOLT=VOLT+6.2832*VOL2(IMAX)
      CALL MAT
   31 DO 33 I=1,IM1
      P1(I)=SI
   33 P2(I)=SI
      CALL INIZ
      CALL STAMPA(2)
      IF(ITCR)35,35,34
   34 CALL CRITIC
      CALL MAT
      CALL INIZ
      CALL STAMPA(2)
   35 CALL STAMPA(3)
      IF(NCAN)1200,1200,1201
 1201 CONTINUE
      CALL CCAN(ITIPO)
      DO 1008 M=1,NREG
      ID=I2(M)-1
      IS=I1(M)+1
      VOLO=6.2832*VOL1(IS-1)
      FL2=P2(IS-1)*VOLO
      DO 1050 I=IS,ID
      VOLO=VOLO+VOL(I)
      FL2=FL2+P2(I)*VOL(I)
 1050 CONTINUE
      VR(M)=VOLO+6.2832*VOL2(ID+1)
      FLIM2(M)=(FL2+P2(ID+1)*6.2832*VOL2(ID+1))/VR(M)
      PI(M)=P(M)
      SPRI(M)=SPR(M)
 1008 CONTINUE
      N=0
      DO 1009 M=1,NREG
      POWER(M)=DATA(M+20)
      IF(ICAN(M))1009,1009,1010
 1010 N=N+1
      NSEL=ITIPO(N)
      GOTO(1100,1101),NSEL
 1100 CALL CANCIL (0.0,N,NS,NP3,TO,1.0,POWER(M))
      GOTO1103
 1101 CALL CANSL(0.0,N,NS,NP3,TO,1.0,POWER(M))
 1103 CONTINUE
      TMUI(M)=TMED(N,1)
      TMC1I(M)=TMED(N,2)
      TMC2I(M)=TMED(N,3)
      TCI(M)=TMED(N,4)
 1009 CONTINUE
      DO 1020 M=1,NCAN
      COU(M)=DATA(M+1300)
 1020 COR(M)=DATA(M+1310)
 1200 CONTINUE
      KS=0
      KST=1851
 1000 CONTINUE
      KTP=DATA(KST)+0.0001
      IF(KTP)106,106,127
  127 I1P=DATA(KST+1)+0.0001
```

```
      IS1=DATA(KST+2)+0.0001
      I2P=DATA(KST+3)+0.0001
      IS2=DATA(KST+4)+0.0001
      DO 13 KK =1,KTP,I2P
      DO 14 LL =1,I2P,I1P
      DO 15 MM =1,I1P
      IT=IT+1
      TO=TO+DELT
      IF(NCAN)1011,1011,1203
 1203 CONTINUE
      KCAN=KCAN+1
      IF(KCAN-KPC)1011,1012,1012
 1012 KCAN=0
      N=0
      KTMED=KTMED+1
      DO 1013 MR=1,NREG
      IF(ICAN(MR))1013,1013,1015
 1015 ID=I2(MR)-1
      IS=I1(MR)+1
      VOLO=6.2832*VOL1(IS-1)
      FL2=P2(IS-1)*VOLO
      DO 1014 I=IS,ID
 1014 FL2=FL2+P2(I)*VOL(I)
      FL2=(FL2+P2(ID+1)*6.2832*VOL1(ID+1))/VR(MR)
      POWER(MR)=DATA(MR+20)*FL2/FLIM2(MR)
      N=N+1
      NSEL=ITIPO(N)
      GO TO (1104,1105),NSEL
 1104 CALL CANCIL(1.0,N,NS,NP3,TO,DT,POWER(MR))
      GO TO 1106
 1105 CALL CANSL(1.0,N,NS,NP3,TO,DT,POWER(MR))
 1106 CONTINUE
      P(MR)=PI(MR)+PI(MR)*COU(N)*(TMED(N,1)-TMUI(MR))
      SPR(MR)=SPRI(MR)-SA(MR)*COR(N)*(TMED(N,4)-TCI(MR))
 1013 CONTINUE
      CALL MAT
      IF(KTMED-KTME1) 1011,1107,1107
 1107 KTMED=0
      WRITE OUTPUT TAPE 6,1150,TO
 1150 FORMAT (1H0///,21H  TEMPERATURE MEDIE   ,5X,4HTO =F10.4///,4X,1HM,
     110X,2HTU,12X,3HTG1,12X,3HTG2,13X,2HTR///)
      DO 1152 M=1,NCAN
      WRITE OUTPUT TAPE 6,1151,M,(TMED(M,I),I=1,4)
 1151 FORMAT (I5,4F15.2)
 1152 CONTINUE
      KMAP=KMAP+1
      IF(KMAP-KMA1) 1011,1108,1108
 1108 KMAP=0
      DO 1109 N=1,NCAN
      WRITE OUTPUT TAPE 6,1110,N,ITIPO(N),TO
 1110 FORMAT (1H0///9H CANALE NI3,5X,4HTIPOI2,5X,4HTO =F8.3//)
      DO 1111 J=1,NS
 1111 WRITE OUTPUT TAPE 6,1112,J,(TP(K,J,N),K=1,NP3)
 1112 FORMAT (1H0,I5,10F10.3)
 1109 CONTINUE
 1011 CONTINUE
```

```
      CALL BARRE
      DO 12 I=1,IMAX
      AL22(I)=AL2(I)+AB2(I)
      EP2(I)=BE2(I-1)+BE2(I)+AL22(I)*VOL1(I)+AL22(I-1)*VOL2(I)
      SRIT=0.
      IF(NRIT)81,81,82
   82 CONTINUE
      DO 73 K=1,NRIT
   73 SRIT=SRIT+DL(K)*C(K,I)
   81 CONTINUE
      TN1(I)=SRIT*(VOL1(I)+VOL2(I))+DER1(I)*P1(I)+SORM(I)
   12 TN2(I)=DER2(I)*P2(I)
      CALL FLUSSI
      PINT=PINT+PM2*DELT
      DO 30 I=2,IMAX
      DO 74 K=1,NRIT
   74 C(K,I)=C(K,I)-DLDT(K)*C(K,I)+COC(K,I)*P2(I)
   30 CONTINUE
      DO 75 K=1,NRIT
   75 C(K,1)=C(K,2)
      IF(KTE) 120,120,121
  121 CALL TEST
  120 CONTINUE
   15 CONTINUE
      CALL STAMPA(IS1)
   14 CONTINUE
      CALL STAMPA(IS2)
   13 CONTINUE
      KST=KST+6
      GO TO 1000
  106 CALL STAMPA(2)
      IF(LAST)110,110,9000
 9000 CALL EXIT
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
```

```
      SUBROUTINE MAT
C
      DIMENSION DR(15,10),RU(10),TMU(10,21)
      DIMENSION DATA(3500),          MZ(10),MSV(10),RS(10),DX(10),VLR(10
     1),R(15,10),RO(15,10),SC(15,10),VL(15,10),TP(15,21,10),TMED(10,4)
      DIMENSION D1(10),D2(10),VK(10),TAU(10),SA(10),SPR(10),CS(10),X(100
     1),T(100),        P1(100),P2(100),SOR(10) ,BE(100),VOL1(100),VOL2(10
     20),VOL(100),AL2(100),AB2(100),EP2(100),TN1(100),TN2(100),
     3DER1(100),DER2(100),P1S(100),P2S(100),          SP(10),I1(10),I2(10
     4),W(10),V(10),AL1(100),BL1(100),BL2(100),           DEM1(100),DEM2(1
     500),         EP1(100),BI1(100),BI2(100),
     6         VIM(10),SORM(100),BE1(100),BE2(100),BETA(10),CL(10),DLDT(1
     70),DETA(10),CMC(10,100),COC(10,100),COM(10,100),            C(10,1
     800),KV(10),AL22(100),P(10)
      DIMENSION SORD(100)
      COMMON MZ,MSV,DATA,TP,TMED,VL,SC,RO,DR,DX,VLR,RS,R,RU,TMU
      COMMON KBA,KTE,KBAR,KS
      COMMON NREG,D1,D2,VK,TAU,SA,SPR,CS,X,T,C,P1,P2,P1S,P2S,SOR,IMAX,DL
     1DT,CL,DELT,IM1,NK,BE,VOL1,VOL2,VOL,VOLT,AL2,AB2,EP2,TN1,DLD,DER1,T
     2N2,DER2,COC,IT,TO,KBA,KBI,SP,EPN,I1,I2,W,V,AL1,BL1,BL2,DLM,DEM1,DE
     3M2,CMC,BETA,EP1,BI1,BI2,DLD,PMPT,PM2,PM1,INT,EPS,INTE,PER,VIM,PINT
     4,P,SORM,BU,BUT,BE1,BE2,DETA,NRIT,SBETA,IDST,SI,ITCR,KV,SPRG,DAPF,L
     5F,L1,SP1,REP,SPCR,COM,AL22
C
      DO 1 M=1,NREG
      ID=I2(M)-1
      IS=I1(M)
      D1M=D1(M)
      D2M=D2(M)
      VKM=VK(M)
      TAUM=TAU(M)
      SAM=SA(M)
      SPRM=SPR(M)
      SORR=SOR(M)
      CSM=CS(M)
      WM=W(M)
      VM=V(M)
      PM=P(M)
C
C
      DO 2 I=IS,ID
      SORD(I)=SORR
      AL1(I)=D1M/TAUM+D1M*BU+1./(WM*DELT)
      AL2(I)=SAM+SPRM+D2M*BU+1./(VM*DELT)
      BL1(I)=VKM*(1.-SBETA)*SAM
      BL2(I)=PM*D1M/TAUM
      DEM1(I)=1./(WM*DELT)
      DEM2(I)=1./(VM*DELT)
      K=0
   10 K=K+1
      IF(K-NRIT)4,4,11
    4 CMC(K,I)=DETA(K)*VKM*SAM
      GO TO 10
   11 CONTINUE
      BE1(I)=D1M*BE(I)
```

```
      BE2(I)=D2M*BE(I)
    2 CONTINUE
    1 CONTINUE
C
      DO 3  I=1,IMAX
      EP1(I)=BE1(I-1)+BE1(I)+AL1(I)*VOL1(I)+AL1(I-1)*VOL2(I)
      BI1(I)=BL1(I)*VOL1(I)+BL1(I-1)*VOL2(I)
      BI2(I)=BL2(I)*VOL1(I)+BL2(I-1)*VOL2(I)
      DER1(I)=DEM1(I)*VOL1(I)+DEM1(I-1)*VOL2(I)
      DER2(I)=DEM2(I)*VOL1(I)+DEM2(I-1)*VOL2(I)
      SORM(I)=SORD(I)*VOL1(I)+SORD(I-1)*VOL2(I)
      K=0
   12 K=K+1
      IF(K-NRIT)5,5,3
    5 COC(K,I)=(CMC(K,I)*VOL1(I)+CMC(K,I-1)*VOL2(I))/(VOL1(I)+VOL2(I))
      GO TO 12
    3 CONTINUE
      RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
```

```
      SUBROUTINE BARRE
C
      DIMENSION DR(15,10),RU(10),TMU(10,21)
      DIMENSION DATA(3500),          MZ(10),MSV(10),RS(10),DX(10),VLR(10
     1),R(15,10),RO(15,10),SC(15,10),VL(15,10),TP(15,21,10),TMED(10,4)
      DIMENSION D1(10),D2(10),VK(10),TAU(10),SA(10),SPR(10),CS(10),X(100
     1),T(100),        P1(100),P2(100),SOR(10) ,BE(100),VOL1(100),VOL2(10
     20),VOL(100),AL2(100),AB2(100),EP2(100),TN1(100),TN2(100),
     3DER1(100),DER2(100),P1S(100),P2S(100),          SP(10),I1(10),I2(10
     4),W(10),V(10),AL1(100),BL1(100),BL2(100),          DEM1(100),DEM2(1
     500),        EP1(100),BI1(100),BI2(100),
     6          VIM(10),SORM(100),BE1(100),BE2(100),BETA(10),DL(10),DLDT(1
     70),DETA(10),CMC(10,100),COC(10,100),COM(10,100),          C(10,1
     800),KV(10),AL22(100),P(10)
      COMMON MZ,MSV,DATA,TP,TMED,VL,SC,RO,DR,DX,VLR,RS,R,RU,TMU
      COMMON KBA,KTE,KBAR,KS
      COMMON NREG,D1,D2,VK,TAU,SA,SPR,CS,X,T,C,P1,P2,P1S,P2S,SOR,IMAX,DL
     1DT,CL,DELT,IM1,NK,BE,VOL1,VOL2,VOL,VOLT,AL2,AB2,EP2,TN1,DLD,DER1,T
     2N2,DER2,COC,IT,TO,KBA,KBI,SP,EPN,I1,I2,W,V,AL1,BL1,BL2,DLM,DEM1,DE
     3M2,CMC,BETA,EP1,BI1,BI2,DLD,PMPT,PM2,PM1,INT,EPS,INTE,PER,VIM,PINT
     4,P,SORM,BU,BUT,BE1,BE2,DETA,NRIT,SBETA,IDST,SI,ITCR,KV,SPRG,DAPF,L
     5F,L1,SP1,REP,SPCR,COM,AL22
C
      DO 1 M=1,NREG
      IREG = (M-1)*60
      IF (DATA(IREG+302)-0.00001) 1,1,5
    5 DO 2 K=1,30
      IPP = IREG+K
      TOAV = DATA(IPP+301)
      IF (TOAV - 0.00001) 1,1,7
    7 IF (TO-TOAV) 3,3,2
    3 VELAV = DATA (IPP+331)
      TODI = DATA(IPP+300)
      VELDI = DATA(IPP+330)
      SP (M)=VELDI+(VELAV-VELDI)*(TO-TODI)/(TOAV-TODI)
      ID = I2(M)-1
      IS = I1(M)
      DO 4 I= IS,ID
    4 AB2(I)=SP (M)
      GO TO 1
    2 CONTINUE
    1 CONTINUE
      RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
```

```
      SUBROUTINE FLUSSI
      DIMENSION DR(15,10),RU(10),TMU(10,21)
      DIMENSION DATA(3500),           MZ(10),MSV(10),RS(10),DX(10),VLR(10
     1),R(15,10),RO(15,10),SC(15,10),VL(15,10),TP(15,21,10),TMED(10,4)
      DIMENSION D1(10),D2(10),VK(10),TAU(10),SA(10),SPR(10),CS(10),X(100
     1),T(100),          P1(100),P2(100),SOR(10) ,BE(100),VOL1(100),VOL2(10
     20),VOL(100),AL2(100),AB2(100),EP2(100),TN1(100),TN2(100),
     3DER1(100),DER2(100),P1S(100),P2S(100),              SP(10),I1(10),I2(10
     4),W(10),V(10),AL1(100),BL1(100),BL2(100),              DEM1(100),DEM2(1
     500),          EP1(100),BI1(100),BI2(100),GA1(100),AA(100),BB(100),G
     6A2(100),VIM(10),SORM(100),BE1(100),BE2(100),BETA(10),DL(10),DLDT(1
     70),DETA(10),CMC(10,100),COC(10,100),COM(10,100),               C(10,1
     800),KV(10),AL22(100),P(10)
      COMMON MZ,MSV,DATA,TP,TMED,VL,SC,RO,DR,DX,VLR,RS,R,RU,TMU
      COMMON KBA,KTE,KBAR,KS
      COMMON NREG,D1,D2,VK,TAU,SA,SPR,CS,X,T,C,P1,P2,P1S,P2S,SOR,IMAX,DL
     1DT,CL,DELT,IM1,NK,BE,VOL1,VOL2,VOL,VOLT,AL2,AB2,EP2,TN1,DLD,DER1,T
     2N2,DER2,COC,IT,TO,KBA,KBI,SP,EPN,I1,I2,W,V,AL1,BL1,BL2,DLM,DEM1,DE
     3M2,CMC,BETA,EP1,BI1,BI2,DLD,PMPT,PM2,PM1,INT,EPS,INTE,PER,VIM,PINT
     4,P,SORM,BU,BUT,BE1,BE2,DETA,NRIT,SBETA,IDST,SI,ITCR,KV,SPRG,DAPF,L
     5F,L1,SP1,REP,SPCR,COM,AL22
      PMPT=PM2
      INT=0
   11 CONTINUE
      PMP=PM2
C     FLUSSO RAPIDO
C
      DO 1 I=2,IMAX
    1 GA1(I)=TN1(I)+BI1(I)*P2(I)
      AA(1)=0.
      BB(1)=0.
      AA(2)=BE1(2)/EP1(2)
      BB(2)=GA1(2)/EP1(2)
      IM1=IMAX-1
      DO 2 I=3,IM1
      AA(I)=BE1(I)/(EP1(I)-BE1(I-1)*AA(I-1))
    2 BB(I)=AA(I)*(GA1(I)+BE1(I-1)*BB(I-1))/BE1(I)
      NK=IMAX-2
      DO 3 J=1,NK
      I=IMAX-J+1
    3 P1(I-1)=P1(I)*AA(I-1)+BB(I-1)
      P1(1)=P1(2)
      PM1=0
      DO 4 I=2,IM1
    4 PM1=PM1+P1(I)*VOL(I)
      PM1=PM1/VOLT
C
C     FLUSSO TERMICO
C
      DO 5 I=1,IMAX
    5 GA2(I)=TN2(I)+BI2(I)*P1(I)
      AA(1)=0
      BB(1)=0
      AA(2)=BE2(2)/EP2(2)
      BB(2)=GA2(2)/EP2(2)
```

```
      DO 6 I=3,IM1
      AA(I)=BE2(I)/(EP2(I)-BE2(I-1)*AA(I-1))
    6 BB(I)=AA(I)*(GA2(I)+BE2(I-1)*BB(I-1))/BE2(I)
      DO 7 J=1,NK
      I=IMAX-J+1
    7 P2(I-1)=P2(I)*AA(I-1)+BB(I-1)
      P2(1)=P2(2)
      PM2=0
      DO 8 I=2,IM1
    8 PM2=PM2+P2(I)*VOL(I)
      PM2=PM2/VOLT
      INT=INT+1
      DIF=ABSF((PMP-PM2)/PM2)
      IF(DIF-EPS)9,10,10
   10 IF(INT-INTE)11,11,9
    9 DO 12 I=1,IMAX
      P1S(I)=P1(I)
   12 P2S(I)=P2(I)
      RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
```

```
      SUBROUTINE INIZ
C
      DIMENSION DR(15,10),RU(10),TMU(10,21)
      DIMENSION DATA(3500),           MZ(10),MSV(10),RS(10),DX(10),VLR(10
     1),R(15,10),RO(15,10),SC(15,10),VL(15,10),TP(15,21,10),TMED(10,4)
      DIMENSION D1(10),D2(10),VK(10),TAU(10),SA(10),SPR(10),CS(10),X(100
     1),T(100),        P1(100),P2(100),SOR(10) ,BE(100),VOL1(100),VOL2(10
     20),VOL(100),AL2(100),AB2(100),EP2(100),TN1(100),TN2(100),
     3DER1(100),DER2(100),P1S(100),P2S(100),             SP(10),I1(10),I2(10
     4),W(10),V(10),AL1(100),BL1(100),BL2(100),            DEM1(100),DEM2(1
     500),          EP1(100),BI1(100),BI2(100),
     6          VIM(10),SORM(100),BE1(100),BE2(100),BETA(10),DL(10),DLDT(1
     70),DETA(10),CMC(10,100),COC(10,100),COM(10,100),              C(10,1
     800),KV(10),AL22(100),P(10)
      COMMON MZ,MSV,DATA,TP,TMED,VL,SC,RO,DR,DX,VLR,RS,R,RU,TMU
      COMMON KBA,KTE,KBAR,KS
      COMMON NREG,D1,D2,VK,TAU,SA,SPR,CS,X,T,C,P1,P2,P1S,P2S,SOR,IMAX,DL
     1DT,DL,DELT,IM1,NK,BE,VOL1,VOL2,VOL,VOLT,AL2,AB2,EP2,TN1,DLD,DER1,T
     2N2,DER2,COC,IT,TO,KBA,KBI,SP,EPN,I1,I2,W,V,AL1,BL1,BL2,DLM,DEM1,DE
     3M2,CMC,BETA,EP1,BI1,BI2,DLD,PMPT,PM2,PM1,INT,EPS,INTE,PER,VIM,PINT
     4,P,SORM,BU,BUT,BE1,BE2,DETA,NRIT,SBETA,IDST,SI,ITCR,KV,SPRG,DAPF,L
     5F,L1,SP1,REP,SPCR,COM,AL22
      DO 2 I=1,IMAX
    2 EP2(I)=BE2(I-1)+BE2(I)+AL2(I)*VOL1(I)+AL2(I-1)*VOL2(I)
      DO 1 LK=1,IDST
      DO 7 I=1,IMAX
      SRIT=0.
      IF(NRIT)20,20,21
   21 CONTINUE
      DO 8 L=1,NRIT
    8 SRIT=SRIT+DL(L)*C(L,I)
   20 CONTINUE
      TN1(I)=SRIT*(VOL1(I)+VOL2(I))+DER1(I)*P1(I)+SORM(I)
    7 TN2(I)=DER2(I)*P2(I)
      CALL FLUSSI
      SPH=0.
      DO 9 I=2,IM1
    9 SPH=SPH+P2(I)*VOL(I)
      FN=SI*VOLT/SPH
      DO 3 I=1,IMAX
      P2(I)=P2(I)*FN
    3 P1(I)=P1(I)*FN
C
      IF(NRIT)1,1,100
  100 CONTINUE
      DO 4 M=1,NREG
      ID=I2(M)-1
      IS=I1(M)
      VKM=VK(M)
      SAM=SA(M)
      DO 5 I=IS,ID
      DO 10 K=1,NRIT
   10 COM(K,I)=BETA(K)*VKM*SAM/DL(K)
    5 CONTINUE
    4 CONTINUE
```

```
C
      DO 6 I=1,IMAX
      DO 11 K=1,NRIT
   11 C(K,I)=P2(I)*(COM(K,I)*VOL1(I)+COM(K,I-1)*VOL2(I))/(VOL1(I)+VOL2(I
     1))
    6 CONTINUE
    1 CONTINUE
      PM1=0.
      PM2=0.
      DO 12 I=2,IM1
      PM1=PM1+P1(I)*VOL(I)
   12 PM2=PM2+P2(I)*VOL(I)
      PM2=PM2/VOLT
      PM1=PM1/VOLT
      RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
```

```
      SUBROUTINE CRITIC
C
      DIMENSION DR(15,10),RU(10),TMU(10,21)
      DIMENSION DATA(3500),              MZ(10),MSV(10),RS(10),DX(10),VLR(10
     1),R(15,10),RO(15,10),SC(15,10),VL(15,10),TP(15,21,10),TMED(10,4)
      DIMENSION D1(10),D2(10),VK(10),TAU(10),SA(10),SPR(10),CS(10),X(100
     1),T(100),          P1(100),P2(100),SOR(10) ,BE(100),VOL1(100),VOL2(10
     20),VOL(100),AL2(100),AB2(100),EP2(100),TN1(100),TN2(100),
     3DER1(100),DER2(100),P1S(100),P2S(100),              SP(10),I1(10),I2(10
     4),W(10),V(10),AL1(100),BL1(100),BL2(100),           DEM1(100),DEM2(1
     500),          EP1(100),BI1(100),BI2(100),
     6          VIM(10),SORM(100),BE1(100),BE2(100),BETA(10),DL(10),DLDT(1
     70),DETA(10),CMC(10,100),COC(10,100),COM(10,100),              C(10,1
     800),KV(10),AL22(100),P(10)
      COMMON MZ,MSV,DATA,TP,TMED,VL,SC,RO,DR,DX,VLR,RS,R,RU,TMU
      COMMON KBA,KTE,KBAR,KS
      COMMON NREG,D1,D2,VK,TAU,SA,SPR,CS,X,T,C,P1,P2,P1S,P2S,SOR,IMAX,DL
     1DT,DL,DELT,IM1,NK,BE,VOL1,VOL2,VOL,VOLT,AL2,AB2,EP2,TN1,DLD,DER1,T
     2N2,DER2,COC,IT,TO,KBA,KBI,SP,EPN,I1,I2,W,V,AL1,BL1,BL2,DLM,DEM1,DE
     3M2,CMC,BETA,EP1,BI1,BI2,DLD,PMPT,PM2,PM1,INT,EPS,INTE,PER,VIM,PINT
     4,P,SORM,BU,BUT,BE1,BE2,DETA,NRIT,SBETA,IDST,SI,ITCR,KV,SPRG,DAPF,L
     5F,L1,SP1,REP,SPCR,COM,AL22
C
      DO 1 I=1,NREG
    1 KV(I)=DATA(I+1600)
      SPRG=DATA(1611)
      DAPF=DATA(1612)
      LF=DATA(1613)+0.0001
      WRITE OUTPUT TAPE 6,204
  204 FORMAT (1H0///,15X,18H RICERCA CRITICITA)
      WRITE OUTPUT TAPE 6,30,(KV(I),I=1,NREG)
   30 FORMAT (1H //////,22H  REGIONI AVVELENATE  ,7I10)
      WRITE OUTPUT TAPE 6,31,SPRG,DAPF,LF,ITCR
   31 FORMAT (1H0///,10X,6HSPRG =E14.5,4X,6HDAPF =E14.5,4X,4HLF =I5,4X,6
     1HITCR =I5/////)
      SBETI=SBETA
      SBETA=0.
      CALL MAT
      L1=0.
      SP1=0.
      SPCR=0.
      ITCI=5
      DTG=DELT*FLOATF(ITCI)
C
 1000 CONTINUE
      L1=L1+1
      DO 4 M=1,NREG
      KVM=KV(M)
      IF(M-KVM)5,5,4
    5 SPR(M)=SPCR
    4 CONTINUE
      CALL MAT
      DO 6 I=1,IMAX
    6 EP2(I)=BE2(I-1)+BE2(I)+AL2(I)*VOL (I)+AL2(I-1)*VOL2(I)
      DO 7 L=1,ITCR,ITCI
```

```
      PMPC=PM2
      DO 77 J=1,ITCI
      DO 20 I=1,IMAX
      TN1(I)=DER1(I)*P1(I)
   20 TN2(I)=DER2(I)*P2(I)
   77 CALL FLUSSI
      DP=(PM2-PMPC)/DTG
      REP=(DP*2.)/(PM2+PMPC)
    7 CONTINUE
      IF(L1-1)14,14,15
   14 CONTINUE
      SP1=SPCR
      REP1=REP
      SPCR=SPRG
      GO TO 1000
   15 DAP=ABSF(REP)
      IF(DAPF-DAP)9,10,10
    9 IF(LF-L1)10,10,11
   11 TG=(SPCR-SP1)/(REP-REP1)
      SP1=SPCR
      SPCR=SPCR-TG*REP
      REP1=REP
      GO TO 1000
   10 DO 12 M=1,NREG
      KVM=KV(M)
      IF(M-KVM)13,13,12
   13 SPR(M)=SPCR
   12 CONTINUE
      WRITE OUTPUT TAPE 6,203
  203 FORMAT (1H0///,5X,10HITERAZIONI,14X,3HPM2,14X,3HREP,11X,6HVELENO)
      WRITE OUTPUT TAPE 6,3,L1,PM2,REP,SPCR
    3 FORMAT (1H ,I10,10X,3E16.5)
      SBETA=SBETI
      IF(NRIT)108,108,107
  107 CONTINUE
      DO 104 M=1,NREG
      ID = I2(M)-1
      IS = I1(M)
      VKM=VK(M)
      SAM=SA(M)
      DO 105 I=IS,ID
      DO 110 K=1,NRIT
  110 COM(K,I)=BETA(K)*VKM*SAM/DL(K)
  105 CONTINUE
  104 CONTINUE
      DO 106 I=1,IMAX
      DO 111 K=1,NRIT
  111 C(K,I)=P2(I)*(COM(K,I)*VOL1(I)+COM(K,I-1)*VOL2(I))/(VOL1(I)+VOL2(I
     1))
  106 CONTINUE
  108 CONTINUE
      RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
```

```
      SUBROUTINE STAMPA(IST)
      DIMENSION DR(15,10),RU(10),TMU(10,21)
      DIMENSION DATA(3500),            MZ(10),MSV(10),RS(10),DX(10),VLR(10
     1),R(15,10),RO(15,10),SC(15,10),VL(15,10),TP(15,21,10),TMED(10,4)
      DIMENSION D1(10),D2(10),VK(10),TAU(10),SA(10),SPR(10),CS(10),X(100
     1),T(100),          P1(100),P2(100),SOR(10) ,BE(100),VOL1(100),VOL2(10
     20),VOL(100),AL2(100),AB2(100),EP2(100),TN1(100),TN2(100),
     3DER1(100),DER2(100),P1S(100),P2S(100),              SP(10),I1(10),I2(10
     4),W(10),V(10),AL1(100),BL1(100),BL2(100),              DEM1(100),DEM2(1
     500),           EP1(100),BI1(100),BI2(100),
     6          VIM(10),SORM(100),BE1(100),BE2(100),BETA(10),CL(10),DLDT(1
     70),DETA(10),CMC(10,100),COC(10,100),COM(10,100),             C(10,1
     800),KV(10),AL22(100),P(10),SAV(10)
      DIMENSION FLM1(10),FLM2(10)
      COMMON MZ,MSV,DATA,TP,TMED,VL,SC,RO,DR,DX,VLR,RS,R,RU,TMU
      COMMON KBA,KTE,KBAR,KS
      COMMON NREG,D1,D2,VK,TAU,SA,SPR,CS,X,T,C,P1,P2,P1S,P2S,SOR,IMAX,DL
     1DT,CL,DELT,IM1,NK,BE,VOL1,VOL2,VOL,VOLT,AL2,AB2,EP2,TN1,DLD,DER1,T
     2N2,DER2,CCC,IT,TO,KBA,KBI,SP,EPN,I1,I2,W,V,AL1,BL1,BL2,DLM,DEM1,DE
     3M2,CMC,BETA,EP1,BI1,BI2,DLD,PMPT,PM2,PM1,INT,EPS,INTE,PER,VIM,PINT
     4,P,SORM,BU,BUT,BE1,BE2,DETA,NRIT,SBETA,IDST,SI,ITCR,KV,SPRG,DAPF,L
     5F,L1,SP1,REP,SPCR,COM,AL22
C
      GO TO (10,20,30),IST
   10 DP=(PM2-PMPT)/DELT
      PER=(PM2+PMPT)/(DP*2.)
      WRITE OUTPUT TAPE 6,1,TO,PM1,PM2,PER
    1 FORMAT (1H0////,4X,4HTO =,F10.5,4X,5HPM1 =,E12.5,4X,5HPM2 =,E12.5,
     14X,5HPER =,E12.5)
      GO TO 40
   20 DP=(PM2-PMPT)/DELT
      PER=(PM2+PMPT)/(DP*2.)
      WRITE OUTPUT TAPE 6,2,TO,IT,PER,PINT,INT
    2 FORMAT (1H0///////,4X,4HTO =,F10.5,3X,4HIT =,I6,4X,5HPER =,E12.5,
     14X,6HPINT =,E12.5,4X,5HINT =,I3)
      WRITE OUTPUT TAPE 6,3
    3 FORMAT (1H ///  ,15X,1HR,14X,2HP1,14X,2HP2,//)
      WRITE OUTPUT TAPE 6,4,(I,X(I),P1(I),P2(I),I=1,IMAX)
    4 FORMAT (1H ,I5,3E16.5)
      WRITE OUTPUT TAPE 6,5,PM1,PM2
    5 FORMAT (1H0,10X,11HVALORI MEDI,2E16.5)
      IF(NRIT)40,40,21
   21 CONTINUE
      WRITE OUTPUT TAPE 6,9
    9 FORMAT (1H ///,14X,2HC1,14X,2HC2,14X,2HC3,14X,2HC4,14X,2HC5,14X,2H
     1C6,//)
      DO 11 I=1,IMAX
   11 WRITE OUTPUT TAPE 6,12,I,(C(K,I),K=1,6)
   12 FORMAT (I5,6E16.5)
      IF(NRIT-6)100,100,101
  101 WRITE OUTPUT TAPE 6,102
  102 FORMAT (1H ///,14X,2HC7,14X,2HC8,14X,2HC9,14X,3HC10,//)
      DO 111 I=1,IMAX
  111 WRITE OUTPUT TAPE 6,12,I,(C(K,I),K=7,NRIT)
  100 CONTINUE
```

```
      GO TO 40
   30 DP=(PM2-PMPT)/DELT
      PER=(PM2+PMPT)/(DP*2.)
      DO 8 M=1,NREG
      ID=I2(M)-1
      IS=I1(M)+1
      VOLO=6.2832*VOL1(IS-1)
      FL1=P1(IS-1)*VOLO
      FL2=P2(IS-1)*VOLO
      DO 50 I=IS,ID
      VOLO=VOLO+VOL(I)
      FL1=FL1+P1(I)*VOL(I)
   50 FL2=FL2+P2(I)*VOL(I)
      VOLO=VOLO+VOL2(ID+1)
      FLM1(M)=(FL1+P1(ID+1)*VOL2(ID+1))/VOLO
      FLM2(M)=(FL2+P2(ID+1)*VOL2(ID+1))/VOLO
    8 SAV(M)=AB2(ID)
      WRITE OUTPUT TAPE 6,1,TO,PM1,PM2,PER
      WRITE OUTPUT TAPE 6,6,SAV,FLM1,FLM2
    6  FORMAT(13HOVELENI BARRE/10E12.4/9X,4HFLM1/10E12.4/9X,4HFLM2/10E
     112.4)
   40 CONTINUE
      RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
```

```
      SUBROUTINE TEST
      DIMENSION DR(15,10),RU(10),TMU(10,21)
      DIMENSION DATA(3500),          MZ(10),MSV(10),RS(10),DX(10),VLR(10
     1),R(15,10),RO(15,10),SC(15,10),VL(15,10),TP(15,21,10),TMED(10,4)
      DIMENSION D1(10),D2(10),VK(10),TAU(10),SA(10),SPR(10),CS(10),X(100
     1),T(100),         P1(100),P2(100),SOR(10) ,BE(100),VOL1(100),VOL2(10
     20),VOL(10C),AL2(100),AB2(100),EP2(100),TN1(100),TN2(1C0),
     3DER1(100),DER2(100),P1S(100),P2S(100),          SP(10),I1(10),I2(10
     4),W(10),V(10),AL1(100),BL1(100),BL2(100),          DEM1(100),DEM2(1
     500),        EP1(100),BI1(100),BI2(100),
     6        VIM(10),SORM(100),BE1(100),BE2(100),BETA(10),DL(10),DLDT(1
     70),DETA(10),CMC(10,100),COC(10,100),COM(10,100),            C(10,1
     800),KV(10),AL22(100),P(10)
      COMMON MZ,MSV,DATA,TP,TMED,VL,SC,RO,DR,DX,VLR,RS,R,RU,TMU
      COMMON KBA,KTE,KBAR,KS
      COMMON NREG,D1,D2,VK,TAU,SA,SPR,CS,X,T,C,P1,P2,P1S,P2S,SOR,IMAX,DL
     1DT,DL,DELT,IM1,NK,BE,VOL1,VOL2,VOL,VOLT,AL2,AB2,EP2,TN1,DLD,DER1,T
     2N2,DER2,COC,IT,TO,KBA,KBI,SP,EPN,I1,I2,W,V,AL1,BL1,BL2,DLM,DEM1,DE
     3M2,CMC,BETA,EP1,BI1,BI2,DLD,PMPT,PM2,PM1,INT,EPS,INTE,PER,VIM,PINT
     4,P,SORM,BU,BUT,BE1,BE2,DETA,NRIT,SBETA,IDST,SI,ITCR,KV,SPRG,DAPF,L
     5F,L1,SP1,REP,SPCR,COM,AL22
      KTE=0
      RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
```

```
      SUBROUTINE DCAN(ITIPO)
      DIMENSION DATA(3500),MZ(10),MSV(10),RS(10),RR(10),DX(10),VLR(10),I
     1TIPO(10),DRU(10),RU(10),RP(10)
      DIMENSION DR(15,10),R(15,10),RO(15,10),SC(15,10),VL(15,10)
      DIMENSION TP(15,21,10),TMED(10,4),TMU(10,21)
      COMMON MZ,MSV,DATA,TP,TMED,VL,SC,RO,DR,DX,VLR,RS,R,RU,TMU
      NCAN = DATA(17)+0.1
      DO 71 L1=1,NCAN
      NC=30*(L1-1)
      ITIPO(L1)=DATA(NC+2530)+0.0001
      MZ(L1)=DATA(NC+2501)+0.0001
      MSV(L1)=DATA(NC+2502)+0.0001
      N=MZ(L1)
      NSV=MSV(L1)
      NS=NSV+1
      NP1=N+1
      NP2=N+2
      NP3=N+3
      DO 62 J=1,NP3
   62 TP(J,1,L1)=0.
      FN=FLOATF(N)
      DRU(L1)=DATA(NC+2503)/(2.*FN)
      DO 12 I=1,N
   12 DR(I,L1)=DRU(L1)
      DR(NP1,L1)=DATA(NC+2504)
      DR(NP2,L1)=DATA(NC+2505)
      DR(NP3,L1)=DATA(NC+2506)
      R(1,L1)=DR(1,L1)
      DO 13 I=2,NP3
   13 R(I,L1)=R(I-1,L1)+DR(I,L1)
      DO 14 I=1,N
      RO(I,L1)=DATA(NC+2507)
      SC(I,L1)=DATA(NC+2511)
   14 VL(I,L1)=2.*DATA(NC+2515)/(DR(I,L1)+DR(I+1,L1))
      RO(NP1,L1)=DATA(NC+2508)
      RO(NP2,L1)=DATA(NC+2509)
      RO(NP3,L1)=DATA(NC+2510)
      SC(NP1,L1)=DATA(NC+2512)
      SC(NP2,L1)=DATA(NC+2513)
      SC(NP3,L1)=DATA(NC+2514)
      RU(L1)=DR(N,L1)/(2.*DATA(NC+2515))
      RP(L1)=DR(NP1,L1)/(2.*DATA(NC+2516))
      RS(L1)=DR(NP2,L1)/(2.*DATA(NC+2517))
      VL(N,L1)=1./(RU(L1)+RP(L1))
      VL(NP1,L1)=1./(RP(L1)+RS(L1))
      VLR(L1)=DATA(NC+2518)
      RR(L1)=1./VLR(L1)
      VL(NP2,L1)=1./(RS(L1)+RR(L1))
   71 DX(L1)=DATA(NC+2519)/DATA(NC+2502)
      RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
```

```
      SUBROUTINE CANCIL(FVI,ICAN,NS,NP3,TEM,DT,P)
      DIMENSION DATA(3500),AI(15),BI(15),CI(15),DI(15),A(15),B(15),C(15)
     1,TN(15),MZ(10),MSV(10),              RS(10),RR(10),DX(10),VLR(10)
      DIMENSION DR(15,10),R(15,10),RO(15,10),SC(15,10),VL(15,10),W(10,21
     1),TPP(21,10),RU(10),TMU(10,21),PZ(15)
      DIMENSION TP(15,21,10),TMED(10,4),TPI(10),WS(10)
      COMMON MZ,MSV,DATA,TP,TMED,VL,SC,RO,DR,DX,VLR,RS,R,RU,TMU
      NVI=FVI+1.0001
      P=P
      IC=ICAN
      NC=30*(IC-1)+1
      N=MZ(IC)
      NSV=MSV(IC)
      NS=NSV+1
      NP1=N+1
      NP2=N+2
      NP3=N+3
      T=TEM
      DO 61 I=1,NS
   61 TPP(I,IC)=TP(NP3,I,IC)
      CALL VINIZ (1,IC,T,TPI(IC),WS(IC))
      TP(NP3,1,IC)=TPI(IC)
      CALL POT (1,IC,N,NS,T,P,W)
      F=3.14*DR(NP3,IC)*(DR(NP3,IC)+2.*R(NP2,IC))*RO(NP3,IC)*WS(IC)
      CI(1)=VL(1,IC)*R(1,IC)
      DI(1)=FVI*RO(1,IC)*SC(1,IC)*R(1,IC)*R(1,IC)/(2.*DT)
      BI(1)=-(CI(1)+DI(1))
      DO 11 I=2,NP1
      AI(I)=VL(I-1,IC)*R(I-1,IC)
      CI(I)=VL(I,IC)*R(I,IC)
      DI(I)=FVI*RO(I,IC)*SC(I,IC)*DR(I,IC)*(R(I-1,IC)+0.5*DR(I,IC))/DT
   11 BI(I)=-(AI(I)+CI(I)+DI(I))
      AI(NP2)=VL(NP1,IC)*R(NP1,IC)
      CI(NP2)=0.5*VL(NP2,IC)*R(NP2,IC)
      DI(NP2)=FVI*RO(NP2,IC)*SC(NP2,IC)*DR(NP2,IC)*(R(NP1,IC)+0.5*DR(NP2
     1,IC))/DT
      BI(NP2)=-(AI(NP2)+2.*CI(NP2)+DI(NP2))
      AI(NP3)=VL(NP2,IC)*R(NP2,IC)
      DI(NP3)=FVI*RO(NP3,IC)*SC(NP3,IC)*DR(NP3,IC)*(R(NP2,IC)+0.5*DR(NP3
     1,IC))/DT
      G=SC(NP3,IC)*F/(3.14*DX(IC))
      E=0.5*(AI(NP3)-G+DI(NP3))
      BI(NP3)=-0.5*(AI(NP3)+G+DI(NP3))
      DO 17 I=2,NS
      I=I
      IF(DATA(NC+2515)*DATA(NC+2517)-1.0E-06)1,1,4
    1 IF(FVI)100,100,1010
  100 PZ(1)=W(1,I)
      DO 1000 K=2,N
 1000 PZ(K)=PZ(K-1)+W(K,I)
      TP(NP3,I,IC)=TP(NP3,I-1,IC)+2.0*PZ(N)/G
      IF(DATA(NC+2517)-1.0E-07)1001,1001,1002
 1001 CALL HTC(TP(NP3,I,IC),TPG,WS(IC),VLT,IC)
      RR(IC)=1.0/VLT
      IF(DATA(NC+2515)-1.0E-07)1002,1002,105
 1002 TPG=0.5*(TP(NP3,I,IC)+TP(NP3,I-1,IC))+PZ(N)*(RS(IC)+RR(IC))/R(NP2)
```

```
      KK=N-1
      TP(N,I,IC)=0.0
      DO 1003 K=1,KK
      L=N-K
 1003 TP(L,I,IC)=PZ(L)/CI(L)+TP(L+1,I,IC)
      SUM=TP(1,I,IC)*R(1,IC)*R(1,IC)
      DO 1004 J=2,KK
 1004 SUM=SUM+TP(J,I,IC)*(2.0*R(J-1,IC)+DR(J,IC))*DR(J,IC)
      TMA=SUM/(R(N,IC)*R(N,IC))
      ALF=PZ(N)*(RS(IC)/R(NP1,IC)+RU(IC)/R(N,IC))
      BET=PZ(N)*0.5*(1.0/R(NP1,IC)+1.0/R(N,IC))
      COST=DATA(2500)
      CALL GAPIZ(TMA,TPG,ALF,BET,RGAP,COST,IC)
      GO TO 102
 1010 IF(DATA(NC+2515)-1.0E-07)101,101,3
  101 CALL GAP(TMU(IC,I),TP(N,I,IC),TP(NP2,I,IC),RGAP,COST,IC)
  102 CONTINUE
      VL(N,IC)=1./(RU(IC)+0.5*RGAP)
      VL(NP1,IC)=1./(0.5*RGAP+RS(IC))
      DO 5 K=N,NP1
      AI(K)=VL(K-1,IC)*R(K-1,IC)
      CI(K)=VL(K,IC)*R(K,IC)
    5 BI(K)=-(AI(K)+CI(K)+DI(K))
      AI(NP2)=VL(NP1,IC)*R(NP1,IC)
      BI(NP2)=-(AI(NP2)+2.*CI(NP2)+DI(NP2))
    2 IF(DATA(NC+2517)-1.0E-07)3,3,4
    3 CALL HTC(TP(NP3,I,IC),TP(NP2,I,IC),WS(IC),VLT,IC)
  105 CONTINUE
      RR(IC)=1./VLT
      VL(NP2,IC)=1./(RS(IC)+RR(IC))
      CI(NP2)=0.5*VL(NP2,IC)*R(NP2,IC)
      BI(NP2)=-(AI(NP2)+2.*CI(NP2)+DI(NP2))
      AI(NP3)=VL(NP2,IC)*R(NP2,IC)
      E=0.5*(AI(NP3)-G+DI(NP3))
      BI(NP3)=-0.5*(AI(NP3)+G+DI(NP3))
    4 B(1)=BI(1)
      C(1)=CI(1)
      DO 18 K=2,NP2
      A(K)=AI(K)
      B(K)=BI(K)
   18 C(K)=CI(K)
      DO 72 K=1,N
   72 TN(K)=-DI(K)*TP(K,I,IC)-W(K,I)
      A(NP3)=AI(NP3)
      B(NP3)=BI(NP3)
      TN(NP1)=-DI(NP1)*TP(NP1,I   ,IC)
      TN(NP2)=-DI(NP2)*TP(NP2,I   ,IC)-CI(NP2)*TP(NP3,I-1,IC)
      TN(NP3)=E*TP(NP3,I-1,IC)-0.5*DI(NP3)*(TPP(I-1,IC)+TPP(I,IC))
C    RISOLUZIONE SISTEMA
      DO 19 K=2,NP3
      B(K)=B(K)-A(K)*C(K-1)/B(K-1)
   19 TN(K)=TN(K)-A(K)*TN(K-1)/B(K-1)
      TP(NP3,I,IC)=TN(NP3)/B(NP3)
      DO 20 K=1,NP2
      K1=NP3-K
   20 TP(K1,I,IC)=TN(K1)/B(K1)-TP(K1+1,I,IC)*C(K1)/B(K1)
```

```
17 CONTINUE
   CALL INTEGR(1,IC,N,NS,NP1,NP2,NP3)
   IF(FVI)31,31,80
31 WRITE OUTPUT TAPE 6,21,IC,T
33 CO 36 I=1,NS
36 WRITE OUTPUT TAPE 6,23,(TP(J,I,IC),J=1,NP3)
   WRITE OUTPUT TAPE 6,90,(TMED(IC,I),I=1,4)
21 FORMAT (1H0///9H CANALE NI3,5X,3HT =F8.3//)
23 FORMAT (1H0,10F10.3)
90 FORMAT (1H0//21H  TEMPERATURE MEDIE  //5X,4HTU =F10.3,5X,5HTG1 =F
  110.3,5X,5HTG2 =F10.3,5X,4HTR =F10.3)
80 RETURN
   END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
```

```
      SUBROUTINE CANSL(FVI,ICAN,NS,NP3,TEM,DT,P)
      DIMENSION DATA(3500),AI(15),BI(15),CI(15),DI(15),A(15),B(15),C(15)
     1,TN(15),MZ(10),MSV(10)               ,RS(10),RR(10),DX(10),VLR(10)
      DIMENSION DR(15,10),R(15,10),RO(15,10),SC(15,10),VL(15,10),W(10,21
     1),TPP(21,10),RU(10),TMU(10,21),PZ(15)
      DIMENSION TP(15,21,10),TMED(10,4),TPI(10),WS(10)
      COMMON MZ,MSV,DATA,TP,TMED,VL,SC,RO,DR,DX,VLR,RS,R,RU,TMU
      NVI=FVI+1.0001
      P=P
      L=0
      IC=ICAN
      NC=30*(IC-1)+1
      N=MZ(IC)
      NSV=MSV(IC)
      NS=NSV+1
      NP1=N+1
      NP2=N+2
      NP3=N+3
      T=TEM
      DO 61 I=1,NS
   61 TPP(I,IC)=TP(NP3,I,IC)
      CALL VINIZ (2,IC,T,TPI(IC),WS(IC))
      TP(NP3,1,IC)=TPI(IC)
      CALL POT (2,IC,N,NS,T,P,W)
      CI(1)=VL(1,IC)
      DI(1)=FVI*RO(1,IC)*SC(1,IC)*DR(1,IC)/DT
      BI(1)=-(CI(1)+DI(1))
      DO 11 I=2,NP1
      AI(I)=VL(I-1,IC)
      CI(I)=VL(I,IC)
      DI(I)=FVI*RO(I,IC)*SC(I,IC)*DR(I,IC)/DT
   11 BI(I)=-(AI(I)+CI(I)+DI(I))
      AI(NP2)=VL(NP1,IC)
      CI(NP2)=0.5*VL(NP2,IC)
      DI(NP2)=FVI*RO(NP2,IC)*SC(NP2,IC)*DR(NP2,IC)/DT
      BI(NP2)=-(AI(NP2)+2.*CI(NP2)+DI(NP2))
      AI(NP3)=VL(NP2,IC)
      DI(NP3)=FVI*RO(NP3,IC)*SC(NP3,IC)*DR(NP3,IC)/DT
      G=2.*SC(NP3,IC)*RO(NP3,IC)*WS(IC)*DR(NP3,IC)/DX(IC)
      E=0.5*(AI(NP3)-G+DI(NP3))
      BI(NP3)=-0.5*(AI(NP3)+G+DI(NP3))
      DO 17 I=2,NS
      I=I
      IF(DATA(NC+2515)*DATA(NC+2517)-1.0E-06)1,1,4
    1 IF(FVI)100,100,1010
  100 PZ(1)=W(1,I)
      DO 1000 K=2,N
 1000 PZ(K)=PZ(K-1)+W(K,I)
      TP(NP3,I,IC)=TP(NP3,I-1,IC)+2.0*PZ(N)/G
      IF(DATA(NC+2517)-1.0E-07)1001,1001,1002
 1001 CALL HTC(TP(NP3,I,IC),TPG,WS(IC),VLT,IC)
      RR(IC)=1.0/VLT
      IF(DATA(NC+2515)-1.0E-07)1002,1002,105
 1002 TPG=0.5*(TP(NP3,I,IC)+TP(NP3,I-1,IC))+PZ(N)*(RS(IC)+RR(IC))
      KK=N-1
      TP(N,I,IC)=0.0
```

```
      DO 1003 K=1,KK
      L=N-K
 1003 TP(L,I,IC)=PZ(L)/CI(L)+TP(L+1,I,IC)
      SUM=TP(1,I,IC)
      DO 1004 J=2,KK
 1004 SUM=SUM+TP(J,I,IC)
      TMA=SUM/FLOATF(N)
      ALF=PZ(N)*(RS(IC)+RU(IC))
      BET=PZ(N)
      COST=DATA(2500)
      CALL GAPIZ(TMA,TPG,ALF,BET,RGAP,COST,IC)
      GO TO 102
 1010 IF(DATA(NC+2515)-1.0E-07)101,101,3
  101 CALL GAP(TMU(IC,I),TP(N,I,IC),TP(NP2,I,IC),RGAP,COST,IC)
  102 CONTINUE
      VL(N,IC)=1.0/(RU(IC)+0.5*RGAP)
      VL(NP1,IC)=1.0/(0.5*RGAP+RS(IC))
      DO 5 K=N,NP1
      AI(K)=VL(K-1,IC)
      CI(K)=VL(K,IC)
    5 BI(K)=-(AI(K)+CI(K)+DI(K))
      AI(NP2)=VL(NP1,IC)
      BI(NP2)=-(AI(NP2)+2.0*CI(NP2)+DI(NP2))
    2 IF(DATA(NC+2517)-1.0E-07)3,3,4
    3 CALL HTC(TP(NP3,I,IC),TP(NP2,I,IC),WS(IC),VLT,IC)
  105 CONTINUE
      RR(IC)=1.0/VLT
      VL(NP2,IC)=1.0/(RS(IC)+RR(IC))
      CI(NP2)=0.5*VL(NP2,IC)
      BI(NP2)=-(AI(NP2)+2.0*CI(NP2)+DI(NP2))
      AI(NP3)=VL(NP2,IC)
      E=0.5*(AI(NP3)-G+DI(NP3))
      BI(NP3)=-0.5*(AI(NP3)+G+DI(NP3))
    4 B(1)=BI(1)
      C(1)=CI(1)
      DO 18 K=2,NP2
      A(K)=AI(K)
      B(K)=BI(K)
   18 C(K)=CI(K)
      DO 72 K=1,N
   72 TN(K)=-DI(K)*TP(K,I,IC)-W(K,I)
      A(NP3)=AI(NP3)
      B(NP3)=BI(NP3)
      TN(NP1)=-DI(NP1)*TP(NP1,I   ,IC)
      TN(NP2)=-DI(NP2)*TP(NP2,I   ,IC)-CI(NP2)*TP(NP3,I-1,IC)
      TN(NP3)=E*TP(NP3,I-1,IC)-0.5*DI(NP3)*(TPP(I-1,IC)+TPP(I,IC))
C    RISOLUZIONE SISTEMA
      DO 19 K=2,NP3
      B(K)=B(K)-A(K)*C(K-1)/B(K-1)
   19 TN(K)=TN(K)-A(K)*TN(K-1)/B(K-1)
      TP(NP3,I,IC)=TN(NP3)/B(NP3)
      DO 20 K=1,NP2
      K1=NP3-K
   20 TP(K1,I,IC)=TN(K1)/B(K1)-TP(K1+1,I,IC)*C(K1)/B(K1)
   17 CONTINUE
      CALL INTEGR(2,IC,N,NS,NP1,NP2,NP3)
```

```
      IF(FVI)31,31,80
31    WRITE OUTPUT TAPE 6,21,IC,T
33    DO 36 I=1,NS
36    WRITE OUTPUT TAPE 6,23,(TP(J,I,IC),J=1,NP3)
      WRITE OUTPUT TAPE 6,90,(TMED(IC,I),I=1,4)
21    FORMAT (1H0////9H CANALE NI3,5X,3HT =F8.3//)
23    FORMAT (1H0,10F10.3)
90    FORMAT (1H0//21H  TEMPERATURE MEDIE  //5X,4HTU =F10.3,5X,5HTG1 =F
     110.3,5X,5HTG2 =F10.3,5X,4HTR =F10.3)
80    RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
```

```
      SUBROUTINE VINIZ (JJ,IC,T,TPI,WS)
      DIMENSION DATA(3500),              DX(10),VLR(10),RS(10),MZ(10),
     1MSV(10)
      DIMENSION RU(10),TMU(10,21)
      DIMENSION VL(15,10),SC(15,10),RO(15,10),DR(15,10),R(15,10)
      DIMENSION TP(15,21,10),TMED(10,4),TPI(10),WS(10)
      COMMON MZ,MSV,DATA,TP,TMED,VL,SC,RO,DR,DX,VLR,RS,R,RU,TMU
      IC=IC
      ICC=30*(IC-1)
      IF(DATA(ICC+2521)-0.0001)9,9,15
   15 IF(T-1.0E-07) 50,50,51
   50 TPI=DATA(ICC+2521)
      GO TO 16
   51 TPI=DATA(ICC+2521)+DATA(ICC+2522)+DATA(ICC+2523)*T
      GO TO 16
    9 IAA=10*(IC-1)
      DO 1 I=1,10
      IPP = IAA+I
      TAV=DATA(IPP+2901)
      IF(T-TAV)2,3,1
    2 TRAV=DATA(IPP+2801)
      TRDI=DATA(IPP+2800)
      TDI=DATA(IPP+2900)
      TPI=TRDI+(T-TDI)*(TRAV-TRDI)/(TAV-TDI)
      GO TO 4
    3 TRAV = DATA(IPP+2801)
      TPI=TRAV
      GO TO 4
    1 CONTINUE
      TPI=TRAV
    4 CONTINUE
   16 IF(DATA(ICC+2524)-1.0E-06) 11,11,18
   18 IF(T-1.0E-10) 52,52,53
   52 WS=DATA(ICC+2524)
      GO TO 12
   53 WS = DATA(ICC+2524)+DATA(ICC+2525)+DATA(ICC+2526)*T
      GO TO 12
   11 IAA = 10*(IC-1)
      DO 5 I=1,10
      IPP = IAA+I
      TAV=DATA(IPP+3101)
      IF(T-TAV)6,7,5
    6 VAV=DATA(IPP+3001)
      VDI=DATA(IPP+3000)
      TDI=DATA(IPP+3100)
      WS=VDI+(T-TDI)*(VAV-VDI)/(TAV-TDI)
      GO TO 8
    7 VAV = DATA(IPP+3001)
      WS=VAV
      GO TO 8
    5 CONTINUE
      WS=VAV
    8 CONTINUE
   12 CONTINUE
      RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
```

```
      SUBROUTINE POT(JJ,IC,N,NS,TO,POWER,W)
      DIMENSION DATA(3500),FAS(21),PFAS(21),W(10,21),DR(15,10),R(15,10),
     1DX(10),VLR(10),RS(10),MZ(10),MSV(10),VL(15,10),SC(15,10),RO(15,10)
      DIMENSION TP(15,21,10),TMED(10,4),RU(10),TMU(10,21)
      COMMON MZ,MSV,DATA,TP,TMED,VL,SC,RO,DR,DX,VLR,RS,R,RU,TMU
      GO TO (11,12),JJ
   11 RUQ=3.1416*R(N,IC)**2*DX(IC)
      NC=(IC-1)*20
      SUM=0.0
      DO 1 I=2,NS
      IFAS=I+NC
      FAS(I)=DATA(IFAS+1999)
    1 SUM=SUM+FAS(I)
      DO 15 I=2,NS
      PFAS(I)=POWER*FAS(I)/(SUM*RUQ)
   15 PFAS(I)=0.5*PFAS(I)
      DO 2 I=2,NS
      W(1,I)=PFAS(I)*DR(1,IC)*DR(1,IC)
      IF(N-2)2,7,7
    7 DO 3 J=2,N
    3 W(J,I)=PFAS(I)*DR(J,IC)*(2.*R(J-1,IC)+DR(J,IC))
    2 CONTINUE
      GO TO 100
   12 NCC=(IC-1)*30
      RUQ=DATA(NCC+2503)*DATA(NCC+2520)*DX(IC)
      NC=(IC-1)*20
      SUM=0.0
      DO 4 I=2,NS
      IFAS=I+NC
      FAS(I)=DATA(IFAS+1999)
    4 SUM=SUM+FAS(I)
      DO 6 I=2,NS
    6 PFAS(I)=POWER*FAS(I)/(SUM*RUQ)
      DO 5 I=2,NS
      DO 5 J=1,N
    5 W(J,I)=PFAS(I)*DR(J,IC)
  100 RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
```

```
      SUBROUTINE INTEGR(JJ,IC,N,NS,NP1,NP2,NP3)
      DIMENSION MZ(10),MSV(10),DATA(3500),TP(15,21,10),TMED(10,4),VL(15,
     110),SC(15,10),RO(15,10),DR(15,10),DX(10),VLR(10),RS(10),R(15,10)
      DIMENSION  TMU(10,21),RU(10)
      COMMON MZ,MSV,DATA,TP,TMED,VL,SC,RO,DR,DX,VLR,RS,R,RU,TMU
      JJ=JJ
      NSM1=NS-1
      FNSV=FLOATF(NSM1)
      GO TO (10,20),JJ
   10 TMUR=0
      DO 2 I=2,NS
      SUMTP=TP(1,I,IC)*R(1,IC)*R(1,IC)
      DO 1 J=2,N
    1 SUMTP=SUMTP+TP(J,I,IC)*(2.*R(J-1,IC)+DR(J,IC))*DR(J,IC)
      TMU(IC,I)=SUMTP/(R(N,IC)*R(N,IC))
    2 TMUR=TMUR+TMU(IC,I)
      TMED(IC,1)=TMUR/FNSV
      GO TO 30
   20 TMUR=0
      DO 7 I=2,NS
      SUMTP=0
      DO 8 J=1,N
    8 SUMTP=SUMTP+TP(J,I,IC)*DR(J,IC)
      TMU(IC,I)=SUMTP/R(N,IC)
    7 TMUR=TMUR+TMU(IC,I)
      TMED(IC,1)=TMUR/FNSV
   30 TM2=0.
      DO 3 I=2,NS
    3 TM2=TM2+TP(NP1,I,IC)
      TMED(IC,2)=TM2/FNSV
      TM3=0
      DO 4 I=2,NS
    4 TM3=TM3+TP(NP2,I,IC)
      TMED(IC,3)=TM3/FNSV
      TM4=0.5*(TP(NP3,1,IC)+TP(NP3,NS,IC))
      DO 5 I=2,NSM1
    5 TM4=TM4+TP(NP3,I,IC)
      TMED(IC,4)=TM4/FNSV
      RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
```

```
      SUBROUTINE GAPIZ(TMA,TPG,ALF,BET,RGAP,COST,IC)
      DIMENSION DATA(3500),RU(10),        DX(10),VLR(10),RS(10),MZ(10),
     1MSV(10),TMU(10,21)
      DIMENSION VL(15,10),SC(15,10),RO(15,10),DR(15,10),R(15,10)
      DIMENSION TP(15,21,10),TMED(10,4)
      COMMON MZ,MSV,DATA,TP,TMED,VL,SC,RO,DR,DX,VLR,RS,R,RU,TMU
      AFUNF(RGAP)=9.7*COST*(AA+BB*RGAP)*(ALF+BET*RGAP)/((ALF+BET*RGAP+TA
     1G)**1.65-TT)-RGAP
      BFUNF(RGAP)=9.7*COST*5.0*(ALF+BET*RGAP)/((ALF+BET*RGAP+TAG)**1.65
     1-TT)-RGAP
      IC=IC
      TAG=TPG+273.0
      AA=-58.0+0.165*TAG-0.148(TMA+ALF)
      BB=-0.148*BET
      TT=TAG**1.65
      IF(AA-5.0)8,8,18
   18 RGAP=(5.0-AA)/BB
      X=AFUNF(RGAP)
      IF(X)19,4,8
   19 XV=X
      RV=0.0
    7 RV1=RGAP
      RGAP=0.5*(RGAP+RV)
      X=AFUNF(RGAP)
      IF(ABSF(X/RGAP)-0.001)4,3,3
    3 SIGN=X*XV
      XV=X
      IF(SIGN)5,5,7
    5 RV=RV1
      GO TO 7
    4 GIO=AA+BB*RGAP
      GO TO 9
    8 GIO=5.0
      RV=0.0
      XV=-1.0
      RGAP=0.01
  111 X=BFUNF(RGAP)
      IF(X)12,9,11
   11 RV=RGAP
      RGAP=2.0*RGAP
      GO TO 111
   12 RV1=RGAP
      RGAP=0.5*(RGAP+RV)
      X=BFUNF(RGAP)
      IF(ABSF(X/RGAP)-0.001)9,13,13
   13 SIGN=X*XV
      XV=X
      IF(SIGN)15,9,12
   15 RV=RV1
      GO TO 12
    9 WRITE OUTPUT TAPE 6,10,GIO,RGAP
   10 FORMAT (1H0/,5X,6H GIO =E12.5,5X,7H RGAP =E12.5)
      RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
```

```
      SUBROUTINE GAP(TPUR,TPS,TPG,RGAP,COST,IC)
      DIMENSION DATA(3500),RU(10),         DX(10),VLR(10),RS(10),MZ(10),
     1MSV(10),TMU(10,21)
      DIMENSION VL(15,10),SC(15,10),RO(15,10),DR(15,10),R(15,10)
      DIMENSION TP(15,21,10),TMED(10,4)
      COMMON MZ,MSV,DATA,TP,TMED,VL,SC,RO,DR,DX,VLR,RS,R,RU,TMU
      IC=IC
      GIO=-58.0+0.313*(TPG+273.0)-0.148*(TPUR+273.0)
      IF(GIO-5.C)1,2,2
    1 GIO=5.0
    2 RGAP= 9.7*GIO*(TPS-TPG)/((TPS+273.0)**1.65-(TPG+273.0)**1.65)*COST
      RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
```

```
      SUBROUTINE HTC(TPCO,TPG,WS,VLT,IC)
      DIMENSION DATA(3500),RU(10),        DX(10),VLR(10),RS(10),MZ(10),
     1MSV(10),TMU(10,21)
      DIMENSION VL(15,10),SC(15,10),RO(15,10),DR(15,10),R(15,10)
      DIMENSION TP(15,21,10),TMED(10,4),WS(10)
      COMMON MZ,MSV,DATA,TP,TMED,VL,SC,RO,DR,DX,VLR,RS,R,RU,TMU
      TPG=TPG
      NC=3200+20*(IC-1)
      DIAH=DATA(NC+17)
      A1=CATA(NC+1)
      A2=CATA(NC+2)
      A3=CATA(NC+3)
      A4=CATA(NC+4)
      A5=CATA(NC+5)
      A6=CATA(NC+6)
      A7=CATA(NC+7)
      A8=CATA(NC+8)
      A9=CATA(NC+9)
      A10=DATA(NC+10)
      A11=DATA(NC+11)
      A12=DATA(NC+12)
      A18=DATA(NC+18)
      A19=DATA(NC+19)
      A20=DATA(NC+20)
    2 TINV1=1.0/TPCO
      TINV2=TINV1*TINV1
      TINV3=TINV1*TINV2
      ROCO=A1+A2*TINV1+A3*TINV2+A4*TINV3
      CSCO=A5+A6*TPCO
      VISCO=A7+A8*TINV1+A9*TINV2+A10*TINV3
      CONCO=A11+A12*TPCO
      PRANDT=VISCO*CSCO/CONCO
      REYN=ROCO*DIAH*WS/VISCO
      VLT=A18*CONCO*REYN**A19*PRANDT**A20/DIAH
      RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
```

Sample output

COSTANZA CILINDRICO

SAMPLE PROBLEM COSTANZA CYLINDRICAL

```
   1  1.000000E-03    2 -0.              3 -0.              4  0.210000E 02    5  0.200000E 01    6  0.100000E 01
   7  0.100000E 02    8  0.200000E 02    9  0.200000E 02   10 -0.             11 -0.             12  1.000000E-04
  14  0.100000E 01   15 -0.             16  0.100000E 02   17  0.200000E 01   18  0.100000E 02   19  0.100000E 01
  20 -0.             21  0.200000E 06   22  0.180000E 07
  31  0.300000E-02
  41  1.000000E-02
  61  0.100000E 01   62  0.110000E 02   63  0.210000E 02
  81  0.130000E 01   82  0.800000E 00   83  0.100000E 01   84  0.130000E 03   85  0.             86  0.500000E-02
  87  0.             88 -0.             89  0.500000E 07   90  0.220000E 06   91  0.100000E 01   92  0.100000E 01
  93  0.130000E 01   94  0.800000E 00   95  0.110000E 01   96  0.130000E 03   97  0.             98  0.500000E-02
  99  0.            100 -0.            101  0.500000E 07  102  0.220000E 06  103  0.100000E 01  104  0.100000E 01
 201 -0.500000E 01  202  0.500000E 01  203  0.100000E 02  204  0.150000E 02  205  0.200000E 02  206  0.250000E 02
 207  0.300000E 02  208  0.350000E 02  209  0.400000E 02  210  0.450000E 02  211  0.500000E 02  212  0.550000E 02
 213  0.600000E 02  214  0.650000E 02  215  0.700000E 02  216  0.750000E 02  217  0.800000E 02  218  0.850000E 02
 219  0.900000E 02  220  0.950000E 02  221  0.100000E 03
 301  0.            302  0.100000E-00  303  0.101000E-00
 331  0.            332  0.500000E-04  333  0.
1301 -0.100000E-04 1302 -0.100000E-04
1311 -0.100000E-04 1312 -0.100000E-04
1602  0.200000E 01
1611  0.100000E-04 1612  1.000000E-04 1613  0.300000E 02
1851  0.100000E 02 1852  0.100000E 01 1853  0.300000E 01 1854  0.100000E 02 1855  0.200000E 01 1856 -0.
1857  0.900000E 02 1858  0.100000E 02 1859  0.300000E 01 1860  0.300000E 02 1861  0.200000E 01 1862 -0.
1863  0.400000E 03 1864  0.500000E 02 1865  0.300000E 01 1866  0.200000E 03 1867  0.200000E 01 1868 -0.
2001  0.400000E-00 2002  0.500000E 00 2003  0.600000E 00 2004  0.650000E 00 2005  0.700000E 00 2006  0.700000E 00
2007  0.650000E 00 2008  0.600000E 00 2009  0.500000E 00 2010  0.400000E-00
2021  0.500000E 00 2022  0.600000E 00 2023  0.650000E 00 2024  0.700000E 00 2025  0.700000E 00 2026  0.650000E 00
2027  0.600000E 00 2028  0.500000E 00
2500  0.100000E 01
2501  0.400000E 01 2502  0.100000E 02 2503  0.252000E 01 2504  0.300000E-01 2505  0.167000E-00 2506  0.243000E-00
2507  0.130000E 02 2508  0.166000E-03 2509  0.270000E 01 2510  0.850000E 00 2511  0.158000E-00 2512  0.523000E 01
2513  0.900000E 00 2514  0.265000E 01 2515  0.220000E-00 2516 -0.            2517  0.200000E 01 2518 -0.
2519  0.150000E 03 2520 -0.            2521  0.380000E 03 2522 -0.            2523 -0.            2524  0.950000E 03
2525 -0.            2526 -0.            2527 -0.            2528 -0.            2529 -0.            2530  0.100000E 01
2531  0.300000E 01 2532  0.800000E 01 2533  0.480000E-01 2534  0.197000E-01 2535  0.197000E-01 2536  0.152000E-00
2537  0.336000E 01 2538  0.270000E 01 2539  0.270000E 01 2540  0.110000E 01 2541  0.754000E 00 2542  0.962000E 00
2543  0.962000E 00 2544  0.418500E 01 2545  0.210000E 01 2546  0.220000E 01 2547  0.220000E 01 2548  0.350000E 01
2549  0.150000E 03 2550  0.950000E 02 2551  0.450000E 02 2552 -0.            2553 -0.            2554  0.720000E 03
2555 -0.            2556 -0.            2557 -0.            2558 -0.            2559 -0.            2560  0.200000E 01
3201 -0.372794E-00 3202  0.888738E 03 3203 -0.218204E 06 3204  0.191089E 08 3205  0.158400E 01 3206  0.242500E-02
3207  0.179876E-02 3208 -0.167242E 01 3209  0.879908E 03 3210 -0.559268E 05 3211  0.144200E-02 3212 -0.105000E-05
3217  0.354000E-00 3218  0.835000E-02 3219  0.900004E 05 3220 -0.
```

Input data

RICERCA CRITICITA

REGIONI AVVELENATE 0 2

SPRG = 0.10000E-04 DAPF = 1.00000E-04 LF = 30 ITCR = 20

ITERAZIONI	PM2		REP	VELENO
9	0.15539E-01	0.		-0.13191E-02

TO = -0. IT = 0 PER = 0.13422E 06 PINT = 0. INT = 1

Criticality search

	R	P1	P2
1	-0.50000E 01	0.74033E 00	0.14816E 01
2	0.50000E 01	0.74033E 00	0.14816E 01
3	0.10000E 02	0.74026E 00	0.14817E 01
4	0.15000E 02	0.74013E 00	0.14819E 01
5	0.20000E 02	0.73991E 00	0.14822E 01
6	0.25000E 02	0.73953E 00	0.14828E 01
7	0.30000E 02	0.73889E 00	0.14839E 01
8	0.35000E 02	0.73781E 00	0.14857E 01
9	0.40000E 02	0.73596E 00	0.14886E 01
10	0.45000E 02	0.73283E 00	0.14937E 01
11	0.50000E 02	0.72746E 00	0.15025E 01
12	0.55000E 02	0.71120E 00	0.14872E 01
13	0.60000E 02	0.67671E 00	0.14243E 01
14	0.65000E 02	0.62530E 00	0.13207E 01
15	0.70000E 02	0.55897E 00	0.11829E 01
16	0.75000E 02	0.48020E-00	0.10173E 01
17	0.80000E 02	0.39175E-00	0.83047E 00
18	0.85000E 02	0.29653E-00	0.62886E 00
19	0.90000E 02	0.19754E-00	0.41901E-00
20	0.95000E 02	0.97737E-01	0.20734E-00
21	0.10000E 03	0.	0.
	VALORI MEDI	0.48185E-00	1.00000E 00

	C1	C2	C3	C4	C5	C6
1	0.	0.	0.	0.	0.	0.
2	0.22223E-02	0.	0.	0.	0.	0.
3	0.22225E-02	0.	0.	0.	0.	0.
4	0.22228E-02	0.	0.	0.	0.	0.
5	0.22233E-02	0.	0.	0.	0.	0.
6	0.22243E-02	0.	0.	0.	0.	0.
7	0.22258E-02	0.	0.	0.	0.	0.
8	0.22285E-02	0.	0.	0.	0.	0.
9	0.22330E-02	0.	0.	0.	0.	0.
10	0.22406E-02	0.	0.	0.	0.	0.
11	0.23692E-02	0.	0.	0.	0.	0.
12	0.24539E-02	0.	0.	0.	0.	0.
13	0.23501E-02	0.	0.	0.	0.	0.
14	0.21792E-02	0.	0.	0.	0.	0.
15	0.19518E-02	0.	0.	0.	0.	0.
16	0.16786E-02	0.	0.	0.	0.	0.
17	0.13703E-02	0.	0.	0.	0.	0.
18	0.10376E-02	0.	0.	0.	0.	0.
19	0.69136E-03	0.	0.	0.	0.	0.
20	0.34211E-03	0.	0.	0.	0.	0.
21	-0.	0.	0.	0.	0.	0.

TO = 0.01000 PM1 = 0.48101E-00 PM2 = 0.99821E 00 PER =-0.25681E 01

VELENI BARRE	0.5000E-05	0.	0.	0.	0.	0.	0.	0.	0.	0.
FLM1	0.7355E 00	0.4197E-00	0.	0.	0.	0.	0.	0.	0.	0.
FLM2	0.1483E 01	0.8852E 00	0.	0.	0.	0.	0.	0.	0.	0.

Print n. 3

TO = 0.01000 IT = 10 PER =-0.25681E 01 PINT = 0.99936E-02 INT = 3

	R	P1	P2
1	-0.50000E 01	0.73834E 00	0.14772E 01
2	0.50000E 01	0.73834E 00	0.14772E 01
3	0.10000E 02	0.73830E 00	0.14773E 01
4	0.15000E 02	0.73820E 00	0.14776E 01
5	0.20000E 02	0.73803E 00	0.14781E 01
6	0.25000E 02	0.73771E 00	0.14788E 01
7	0.30000E 02	0.73714E 00	0.14800E 01
8	0.35000E 02	0.73615E 00	0.14819E 01
9	0.40000E 02	0.73441E 00	0.14852E 01
10	0.45000E 02	0.73138E 00	0.14905E 01
11	0.50000E 02	0.72613E 00	0.14996E 01
12	0.55000E 02	0.71000E 00	0.14847E 01
13	0.60000E 02	0.67565E 00	0.14221E 01
14	0.65000E 02	0.62438E 00	0.13189E 01
15	0.70000E 02	0.55819E 00	0.11814E 01
16	0.75000E 02	0.47956E-00	0.10161E 01
17	0.80000E 02	0.39125E-00	0.82949E 00
18	0.85000E 02	0.29617E-00	0.62813E 00
19	0.90000E 02	0.19730E-00	0.41854E-00
20	0.95000E 02	0.97621E-01	0.20711E-00
21	0.10000E 03	0.	0.
	VALORI MEDI	0.48101E-00	0.99821E 00

Print n. 2

	C1	C2	C3	C4	C5	C6
1	0.22223E-02	0.	0.	0.	0.	0.
2	0.22223E-02	0.	0.	0.	0.	0.
3	0.22225E-02	0.	0.	0.	0.	0.
4	0.22228E-02	0.	0.	0.	0.	0.
5	0.22233E-02	0.	0.	0.	0.	0.
6	0.22243E-02	0.	0.	0.	0.	0.
7	0.22258E-02	0.	0.	0.	0.	0.
8	0.22285E-02	0.	0.	0.	0.	0.
9	0.22330E-02	0.	0.	0.	0.	0.
10	0.22406E-02	0.	0.	0.	0.	0.
11	0.23692E-02	0.	0.	0.	0.	0.
12	0.24539E-02	0.	0.	0.	0.	0.
13	0.23501E-02	0.	0.	0.	0.	0.
14	0.21792E-02	0.	0.	0.	0.	0.
15	0.19518E-02	0.	0.	0.	0.	0.
16	0.16786E-02	0.	0.	0.	0.	0.
17	0.13703E-02	0.	0.	0.	0.	0.
18	0.10376E-02	0.	0.	0.	0.	0.
19	0.69136E-03	0.	0.	0.	0.	0.
20	0.34211E-03	0.	0.	0.	0.	0.
21	-0.	0.	0.	0.	0.	0.

TEMPERATURE MEDIE TO = 0.2000

M	TU	TG1	TG2	TR
1	746.72	451.93	405.25	399.24
2	70.44	69.98	69.49	53.49

Print of
average temperatures
every KTME 1 time steps

CANALE N 1 TIPO 1 TO = 0.200

1	0.	0.	0.	0.	0.	0.
2	818.238	775.933	691.342	564.739	435.538	385.526
3	889.587	836.707	730.979	572.856	432.235	389.555
4	974.912	911.455	784.560	594.500	434.995	394.380
5	1027.348	958.602	821.131	615.218	442.773	399.147
6	1079.887	1005.853	857.808	636.068	450.817	404.254
7	1084.394	1010.360	862.314	640.565	455.435	409.009
8	1040.465	971.719	834.248	628.337	456.300	413.073
9	996.159	932.702	805.818	615.979	457.617	416.713

CANALE N 2 TIPO 2 TO = 0.200

1	0.	0.	0.	0.	0.	45.000
2	59.526	59.469	59.357	59.074	58.677	46.651
3	64.156	64.088	63.953	63.613	63.136	48.654
4	67.725	67.651	67.504	67.135	66.617	50.868
5	71.523	71.444	71.285	70.887	70.327	53.314
6	74.097	74.018	73.859	73.460	72.898	55.835
7	75.266	75.192	75.045	74.674	74.150	58.257
8	76.295	76.226	76.090	75.747	75.262	60.561
9	75.712	75.655	75.541	75.255	74.849	62.556

Print of
temperatures map
every KMA 1 time steps

NOTICE TO THE READER

All Euratom reports are announced, as and when they are issued, in the monthly periodical **EURATOM INFORMATION,** edited by the Centre for Information and Documentation (CID). For subscription (1 year : US$ 15, £ 5.7) or free specimen copies please write to :

Handelsblatt GmbH
"Euratom Information"
Postfach 1102
D-4 Düsseldorf (Germany)

or

Office central de vente des publications
des Communautés européennes
2, Place de Metz
Luxembourg

To disseminate knowledge is to disseminate prosperity — I mean general prosperity and not individual riches — and with prosperity disappears the greater part of the evil which is our heritage from darker times.

Alfred Nobel

SALES OFFICES

All Euratom reports are on sale at the offices listed below, at the prices given on the back of the front cover (when ordering, specify clearly the EUR number and the title of the report, which are shown on the front cover).

PRESSES ACADEMIQUES EUROPEENNES

98, Chaussée de Charleroi, Bruxelles 6

Banque de la Société Générale - Bruxelles
compte N° 964.558,

Banque Belgo Congolaise - Bruxelles
compte N° 2444.141,

Compte chèque postal - Bruxelles - N° 167.37,

Belgian American Bank and Trust Company - New York
compte No. 22.186,

Lloyds Bank (Europe) Ltd. - 10 Moorgate, London E.C.2,

Postscheckkonto - Köln - Nr. 160.861.

OFFICE CENTRAL DE VENTE DES PUBLICATIONS DES COMMUNAUTES EUROPEENNES

2, place de Metz, Luxembourg (Compte chèque postal N° 191-90)

BELGIQUE — BELGIË
MONITEUR BELGE
40-42, rue de Louvain - Bruxelles
BELGISCH STAATSBLAD
Leuvenseweg 40-42 - Brussel

DEUTSCHLAND
BUNDESANZEIGER
Postfach - Köln 1

FRANCE
SERVICE DE VENTE EN FRANCE DES PUBLICATIONS DES COMMUNAUTES EUROPEENNES
26, rue Desaix - Paris 15e

ITALIA
LIBRERIA DELLO STATO
Piazza G. Verdi, 10 - Roma

LUXEMBOURG
OFFICE CENTRAL DE VENTE DES PUBLICATIONS DES COMMUNAUTES EUROPEENNES
9, rue Goethe - Luxembourg

NEDERLAND
STAATSDRUKKERIJ
Christoffel Plantijnstraat - Den Haag

UNITED KINGDOM
H. M. STATIONERY OFFICE
P. O. Box 569 - London S.E.1

EURATOM — C.I.D.
51-53, rue Belliard
Bruxelles (Belgique)