ornl

OAK RIDGE NATIONAL LABORATORY

MARTIN MARIETTA

NASA Contractor Report CR-191133
ORNL/TM-12703

Scoping Calculations of Power Sources for Nuclear Electric Propulsion

F. C. Difilippo

MANAGED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

NASA Contractor Report CR-191133
ORNL/TM-12703

Engineering Physics and Mathematics Division

SCOPING CALCULATIONS OF POWER SOURCES FOR NUCLEAR ELECTRIC PROPULSION

F. C. Difilippo

Date Published: May 1994

Research sponsored by the
National Aeronautics and
Space Administration

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
managed by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. Department of Energy
under contract DE-AC05-84OR21400

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

This work is the consequence of a collaborative agreement with the National Aeronautic and Space Administration (NASA), Lewis Research Center, and the Oak Ridge National Laboratory (ORNL). ORNL counterparts were B. A. Worley of the Engineering Physics and Mathematics Division (EPMD), R. Cooper and R. Holcomb of the Engineering Technology Division (ETD), and NASA counterparts were M. Doherty and J. Gilland from Lewis Research Center, Cleveland, Ohio. W. C. Jordan and J. C. Turner of the Computer Applications Division calculated the database for the fuel pin reactor design.

ABSTRACT

This technical memorandum describes models and calculational procedures used to fully characterize the nuclear island of power sources for nuclear electric propulsion. Two computer codes were written: one for the gas cooled NERVA derivative reactor and the other for liquid metal cooled fuel pin reactors. These codes are going to be interfaced by NASA with the balance of plant in order to make scoping calculations for mission analysis.

1.0 FOREWORD

Systems engineering efforts initiated by NASA's Lewis Research Center (LeRC) in FY92 under RTOP 593-72, for Nuclear Electric Propulsion (NEP), have enabled the development of detailed mathematical (computer) models to predict NEP subsystem performance and mass. The computer models are intended to help provide greater depth to NEP subsystem (and system) modeling, required for more accurately verifying performance projections and assessing the impact of specific technology developments.

The following **subsystem** models have been developed:

(1) Liquid-metal-cooled pin-type, and

(2) Gas-cooled NERVA (Nuclear Engine for Rocket Vehicle Applications) - derived **for reactor/shield**;

(3) Potassium-Rankine, and

(4) Brayton **for power conversion**;

(5) **Heat rejection general model** (includes direct Brayton, pumped loop Brayton, and shear flow condenser (Potassium-Rankine);

(6) **Power management and distribution (PMAD) general model**; and

(7) Magnetoplasmadynamic thruster for the **electric propulsion subsystem**.

These subsystem models for NEP were authored by the Oak Ridge National Laboratory (ORNL) for the reactor (NASA CR-191133, by the Rocketdyne Division of Rockwell International for Potassium Rankine (NASA CR-191134) and Brayton (NASA CR-191135) power conversion, heat rejection (NASA CR-191132), and power management and distribution (NASA CR-191137).

At the time of this writing, these eight VAX/FORTRAN source and executable codes are resident on one of LeRC's Scientific VAX computers.

2.0 SUMMARY

The modeling of a whole nuclear electric propulsion system for cargo and piloted missions is presently being developed at NASA Lewis Research Center. The vehicles would use either a Brayton direct conversion cycle using the heated helium from a NERVA type reactor or a potassium Rankine cycle with the working fluid heated at the secondary site of a heat exchanger with lithium at the primary site coming from a fast reactor.

This report describes the two computer codes written to fully characterize, for each of the options, the nuclear islands of the system. Given a set of input conditions, the codes calculate composition, dimensions, volumes, and masses of the core, reflector, control system, pressure vessel, neutron and gamma shields as well as the thermalhydraulic conditions of the coolant, clad and fuel. Input conditions are power, core life, burnup, pressure, and temperature of the coolant at the inlet of the core, and either the temperature of the coolant at the outlet of the core or the coolant mass flow.

Using state-of-the-art neutron cross sections and transport codes, a database was created for the neutronic performance of both reactor designs. The free parameters of the models are the moderating/fuel element ratio for the NERVA reactor and the enrichment and the pitch of the lattice for the fast reactor. Reactivity and energy balance equations are simultaneously solved to find the reactor design. Thermalhydraulic conditions are calculated by solving the one-dimensional versions of the equations of conservation of mass, energy, and momentum with compressible flow. The additional input conditions for the fluences and integrated doses at a cargo area of radius r located at distance z determines the dimensions, volumes, and masses of the neutron and gamma shields. The assumption was made to use either LiH in a stainless steel matrix or B_4C for the neutron shielding and tungsten for the gamma shielding.

3.0 INTRODUCTION

As a consequence of a collaborative agreement between NASA and ORNL, the Engineering Physics and Mathematics Division has been involved in the development of models and calculational procedures for the analysis (neutronic and thermalhydraulic) of power sources for nuclear electric propulsion.

The boundaries of the system to be modeled are the inlet and outlet plenums and the design variables of interest are the compositions, dimensions, volumes, and masses of the core, reflector, control system, pressure vessel, and neutron and gamma shields. The thermalhydraulic conditions of the coolant are also determined. Main input variables are power, core life, burnup, pressure, and temperature of the inlet coolant and its temperature at the outlet.

The computer codes developed are going to be coupled with the calculations of the balance of plant and ion generation and acceleration to perform scoping calculations for mission analysis. It was required then to develop codes that are fast running on PCs or workstations. For that reason, although the requested parameters are quite comprehensive, the models were kept as simple as possible.

The range of input conditions specified for this task were: a power range of 10 to 50 Mw(th); operating lifetimes of 2 to 10 years; and outlet coolant temperature ranging from 1200 to 2200° K.

3.1 TYPES OF REACTORS CONSIDERED

Two types of reactors were chosen for this analysis: the high temperature gas-cooled reactor of the NERVA derivative type and the lithium-cooled advanced fuel pin reactor, referred to as NERVA derivative and fuel pin for short.

The elements of the core of a NERVA derivative reactor are: (a) fuel elements with ZrC cladding, a dispersion of UC-ZrC in a graphite matrix, and 19 coolant holes (diameter 2.88 mm); and (b) support elements with ZrH_2 on an inconel tube with pyrolitic graphite and graphite as a thermal shield. Both elements are hexagonal with 1.913-cm flat-to-flat distance; by changing the ratio (S/F) of support to fuel elements the neutron spectrum can be modified considerably; the core is cooled by pumping He, which drives a turbine in a direct Brayton cycle. The reflector is made of beryllium, which contains the control drums that rotate the control boron carbide (B_4C) sheets. The pressure vessel is located at the periphery of the reflector.

The core of the fuel pin reactor consists of bundles of rods containing UN pellets. The coolant, liquid lithium, removes heat from the core to boil potassium at the secondary side of a heat exchanger which drives a turbine using a Rankine cycle. The fuel rods have 6.4 mm outside diameter (O.D.) with 0.635-mm-thick cladding of the tantalum alloy Astar. A tungsten liner, 0.122-mm-thick, lays between the clad and the UN pellet, 4.786 mm in diameter with a 0.025-mm-thick He gap surrounding the fuel pellet. The reflector is made of BeO which contains the control drums with the B_4C absorber. The pressure vessel is located between the core and the reflector.

Common to both reactors are the material of the reflector pressure vessel (Astar alloy), the neutron shielding materials (lithium hydride (LiH) in a stainless steel matrix or B_4C), and the gamma shielding material (tungsten).

4.0 NEUTRONIC CALCULATIONS

In order to find the volume of the core, a reactivity balance is performed where the reactivity effects of the reflector, burnup, pressure vessel, structural materials, and desired reactivity at end of life (EOL) are added to the multiplication constant of the bare core. It was then necessary to create a data base for all these parameters; this section describes the criterion and results. Both reactors are intended to be controlled with absorbing materials in the reflector. Thus, the excess reactivity is produced by the reflector with the bare core near critical state. Other factors like compact designs and low masses favor cylindrical shapes with the core diameter ($2R_c$) approximately the same as the core height (H_c) (i.e., $R_c/H_c = 0.541$, this is the ideal shape from a neutron economy standpoint). Many of the data were then generated for a critical core of ideal shape with the transport code XSDRNPM[1] using a very detailed cross section library.[2]

4.1 NERVA DERIVATIVE DATA BASE

The two free parameters of this design are the concentration of the fuel (highly enriched, 93%, ^{235}U) in the fuel element and the number of support-to-fuel-elements ratio. A plot of the critical mass of ^{235}U and the mass of the core plus the 24-cm reflector is presented in Fig. 1 showing that beyond ~500 g of ^{235}U per liter of fuel element the total mass is not substantially reduced (note that Fig. 1 is for an ideal shape, for odd shapes the situation might be different). It was decided to fix the fuel concentration to 500 g of ^{235}U/L fuel; the database depends then on S/F alone.

4.1.1 Neutronic Properties of the Core

Multiplication constants, k_∞, for the infinite lattice and migration lengths, M, are plotted in Fig. 2 as a function of S/F.

4.1.2 Reactivity Worth of Beryllium Reflector

The reactivity effects of a 30 cm thick radial reflector are shown in Fig. 3 as a function of S/F. It is defined as $\Delta k \equiv (k(\text{reflected}) - k(\text{bare})) \times 100$, where k's are the multiplication constants. The effects of the thickness of the Be reflector are shown in Fig. 4. This shape of the curve is used for all S/F although Fig. 4 corresponds to S/F = 0.195 and a fuel density of 374.7 g of ^{235}U/L of fuel.

4.1.3 Reactivity Worth of Burn-Up

The reactivity worths of the burnup, expressed as a percentage of the ^{235}U loading at beginning of life (BOL), are shown in Fig. 5. The values shown were calculated by depleting the number densities of ^{235}U by the percentage burnup and simulating the fission products as a lumped 1/v absorber calibrated at 50 barn per fission at a energy of 0.025 eV.

Because the reactor can have a thermal spectrum because of the presence of support elements, the effects of Xe and Sm at steady state and after transients were computed with formulas taken from Ref. 3. Effective cross sections for these formulas are obtained by collapsing a four-group cross section set and spectras used by the program; in this way, transition from thermal (where Xe poisoning is important) to epithermal spectras (where Xe is irrelevant) can be made gradually. Note that because of the way we computed its effects, 10% burnup means 10% depletion via fission and capture.

4.1.4 Depletion of ^{235}U

Figure 6 illustrates how the fission distributions change as a function of S/F. The change of spectra affects the values of the absorption to fission ratio by as much as 10%. This functional dependence is taken into account by the program.

4.1.5 Reactivity Worth of B_4C Sheet

The reactivity worth of 360° sheets of B_4C located between the core and the reflector are shown in Table 1, which was prepared for a core with S/F = 0 where the neutron spectra is the hardest. The maximum value would correspond to the removal of the reflector. Table 1 shows that a 2 mm thick sheet of natural B_4C is very effective, so it was decided to use it as control material. The reactivity worths of 360°, 2 mm thick natural B_4C sheets are shown in Fig. 7 as a function of S/F and for a 30 cm thick reflector; in general, as the moderation in the core is increased the reflector and anything within it is less important, including the control drums. The worth of the B_4C is then multiplied by the factor given by Fig. 4 to compensate for effects of reflector thickness. Figure 7 also shows the peaking factor in the radial direction of the power density at BOL, i.e., with fresh fuel and control drums in.

Table 1. Reactivity worth of B_4C sheets*

Sheet	Δk (%)	(Δk/Δk)max
2 mm Nat B	21.0	0.70
2 mm 80% B	24.0	0.80
5 mm 90% B	25.9	0.86

*Core: S/F = 0, R_c = 59.25 cm, H_c = 109.11. Reflector: 30 cm Be (Δk)max = 30.14% (removing the Be reflector).

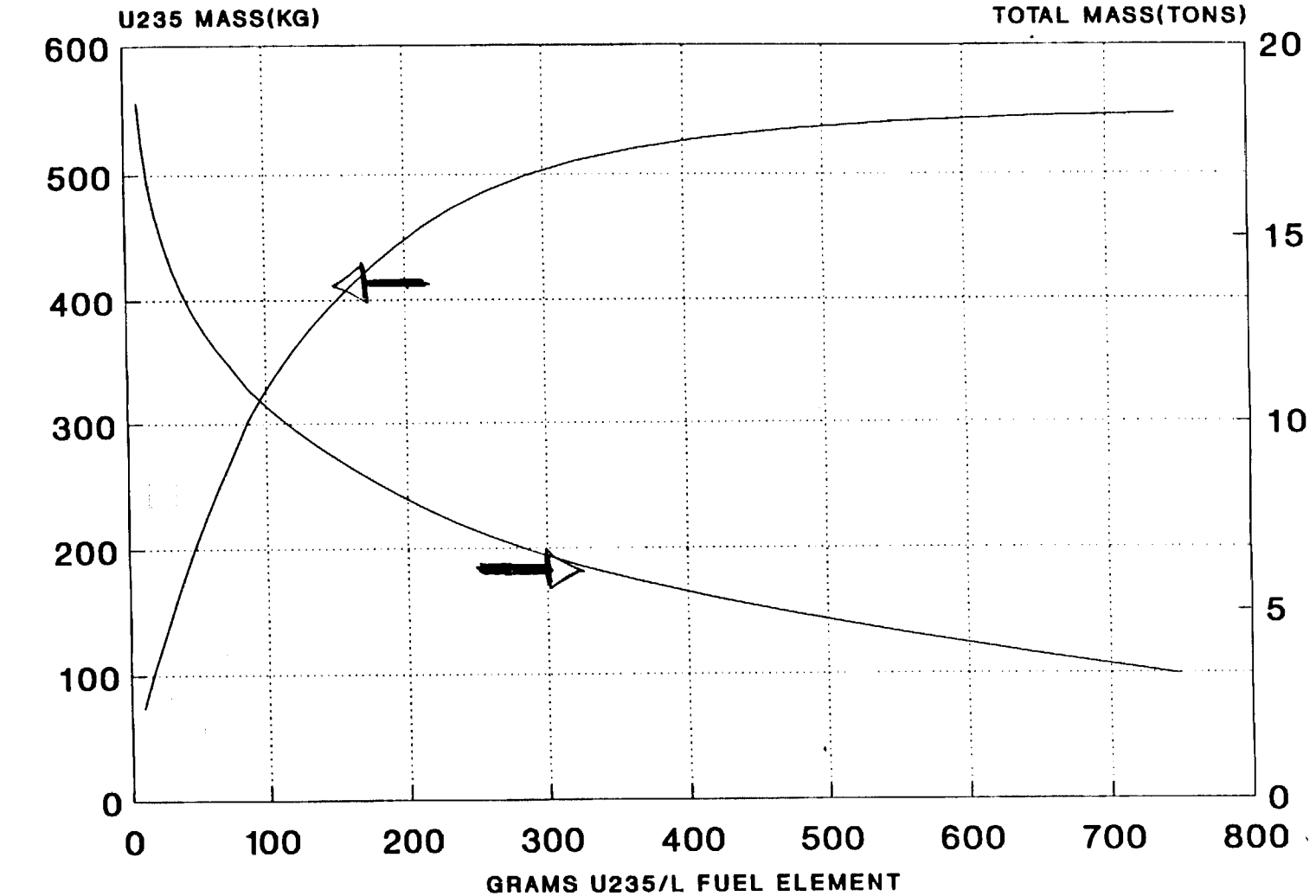

Fig. 1. ^{235}U critical mass of the core and mass of the core plus reflector for the NERVA derivative reactor.

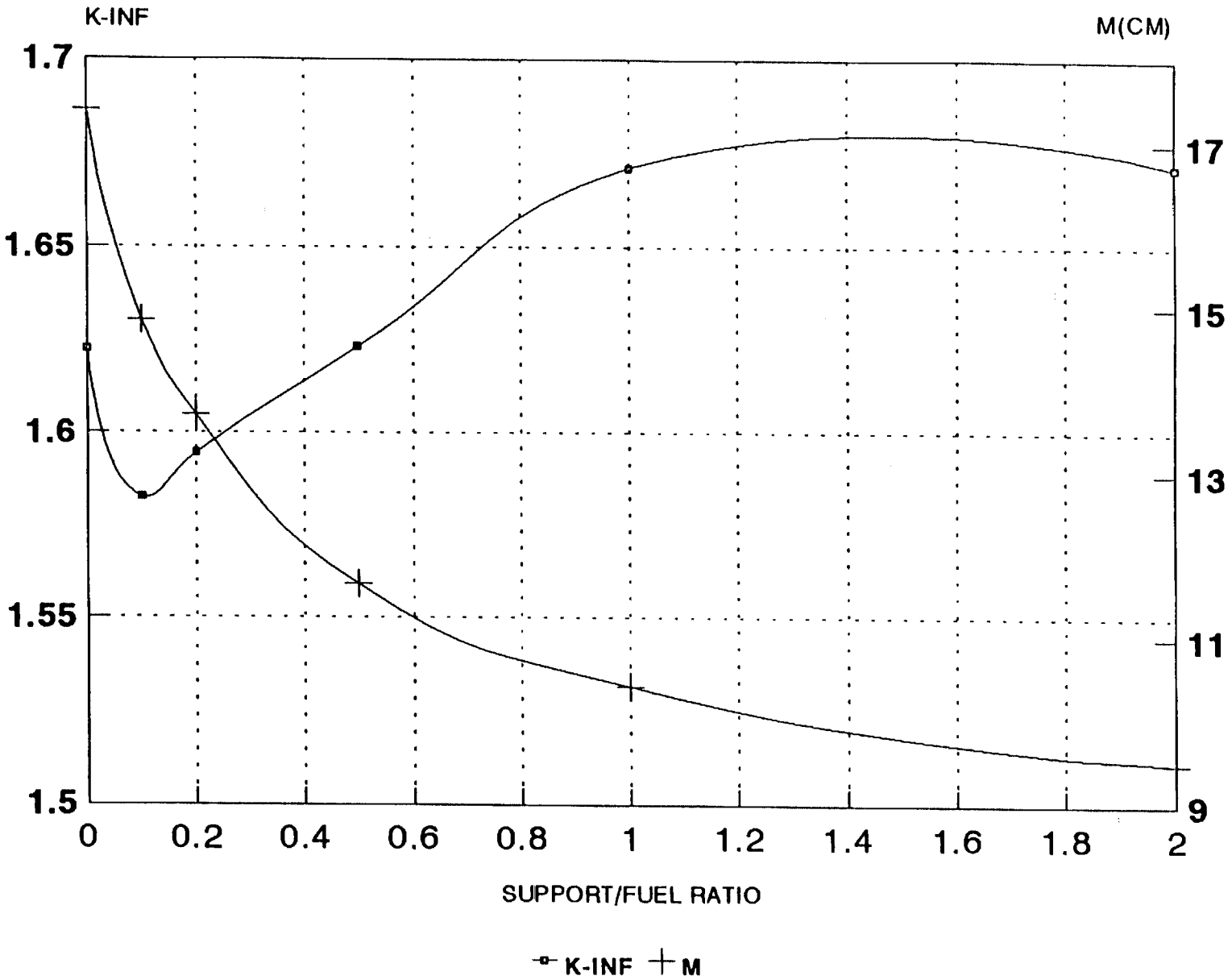

Fig. 2. k_∞ and migration length for the NERVA derivative core.

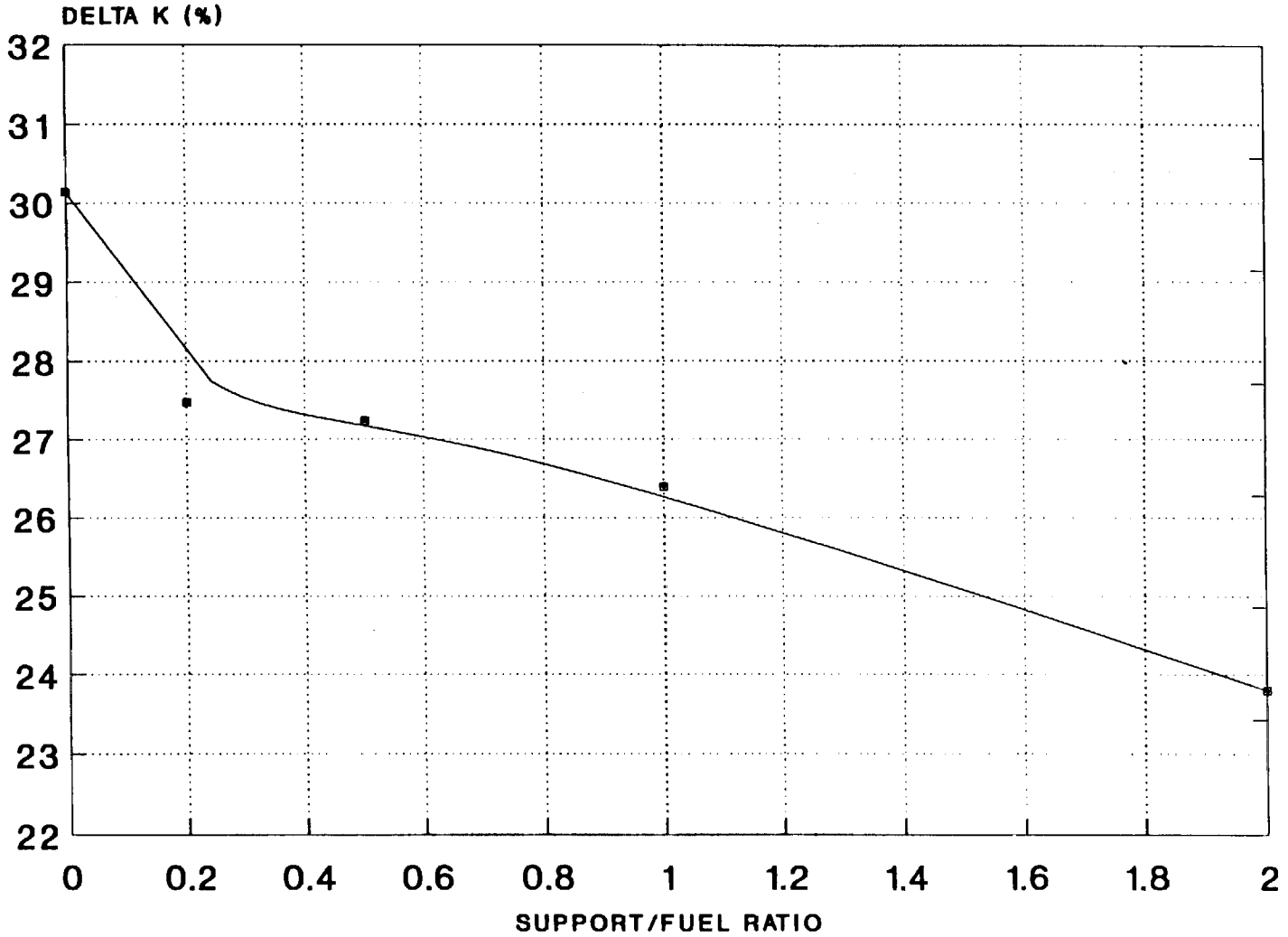

Fig. 3. Reactivity effects of a 30 cm thick radial Be reflector.

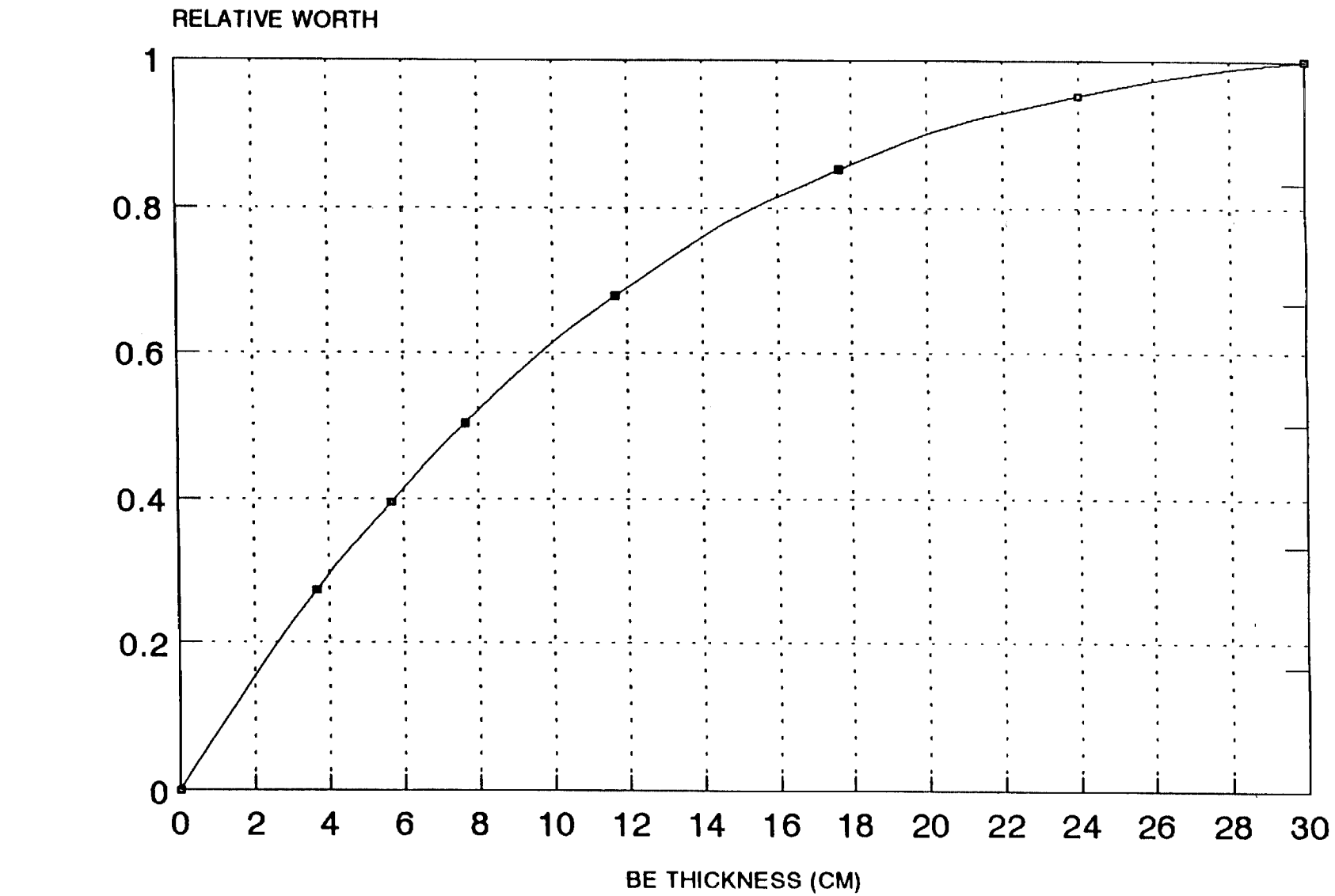

Fig. 4. Worth of the Be reflector (relative to 30 cm thickness) as a function of thickness.

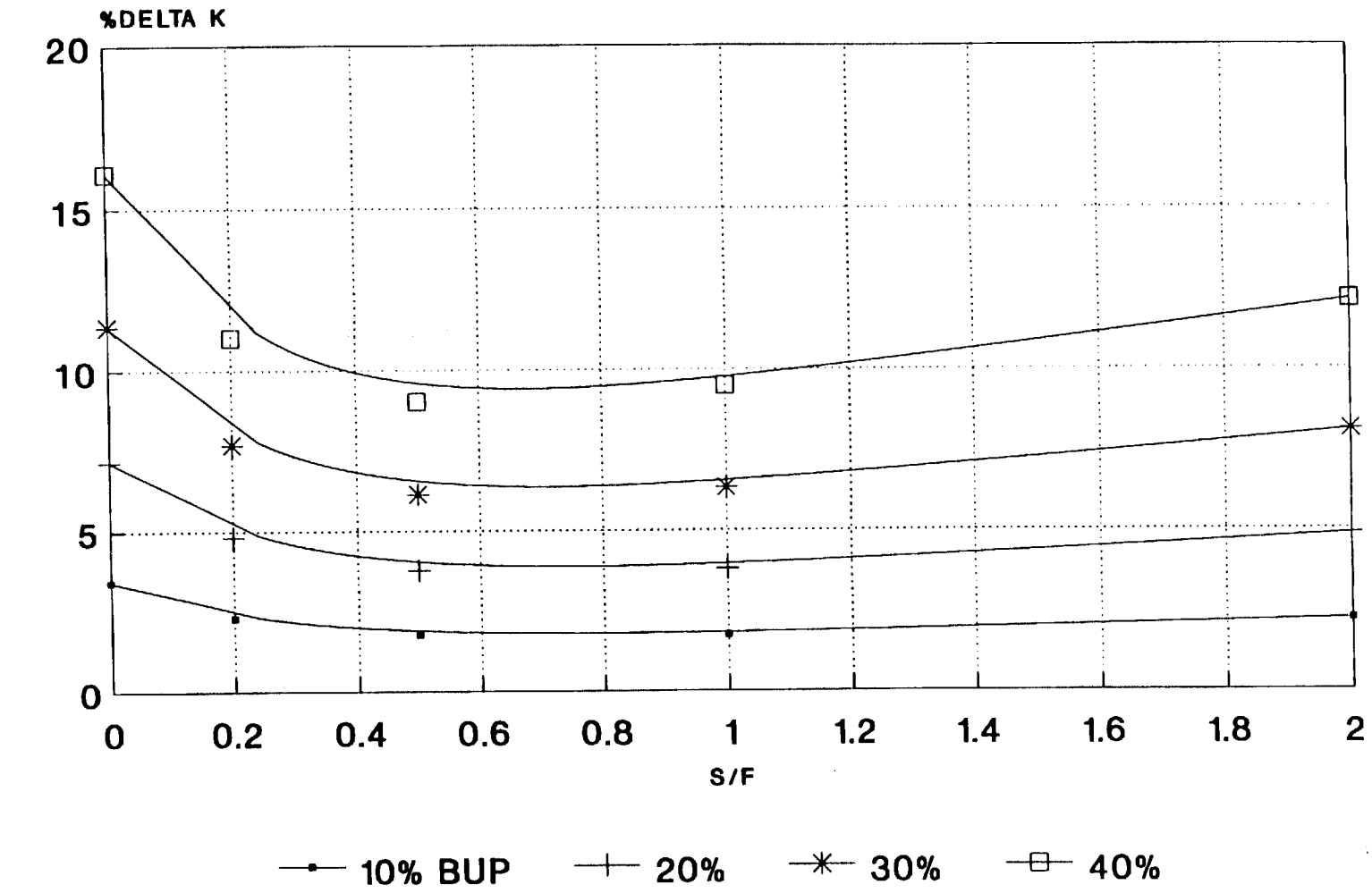

Fig. 5. Reactivity worth of burnup for the NERVA derivative reactor, excluding Xe and Sm.

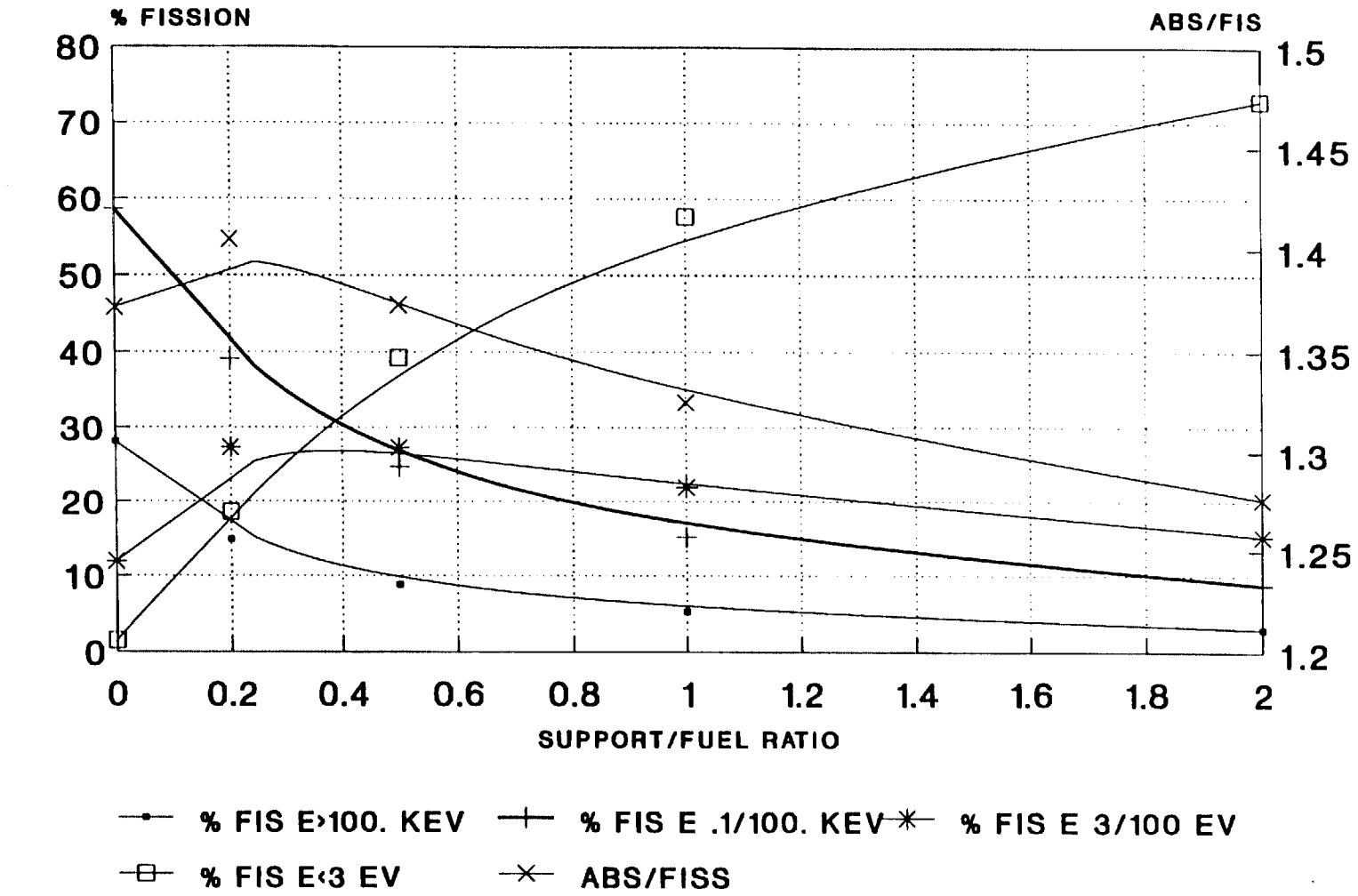

Fig. 6. Fission distributions and absorption over fission ratio as a function of S/F.

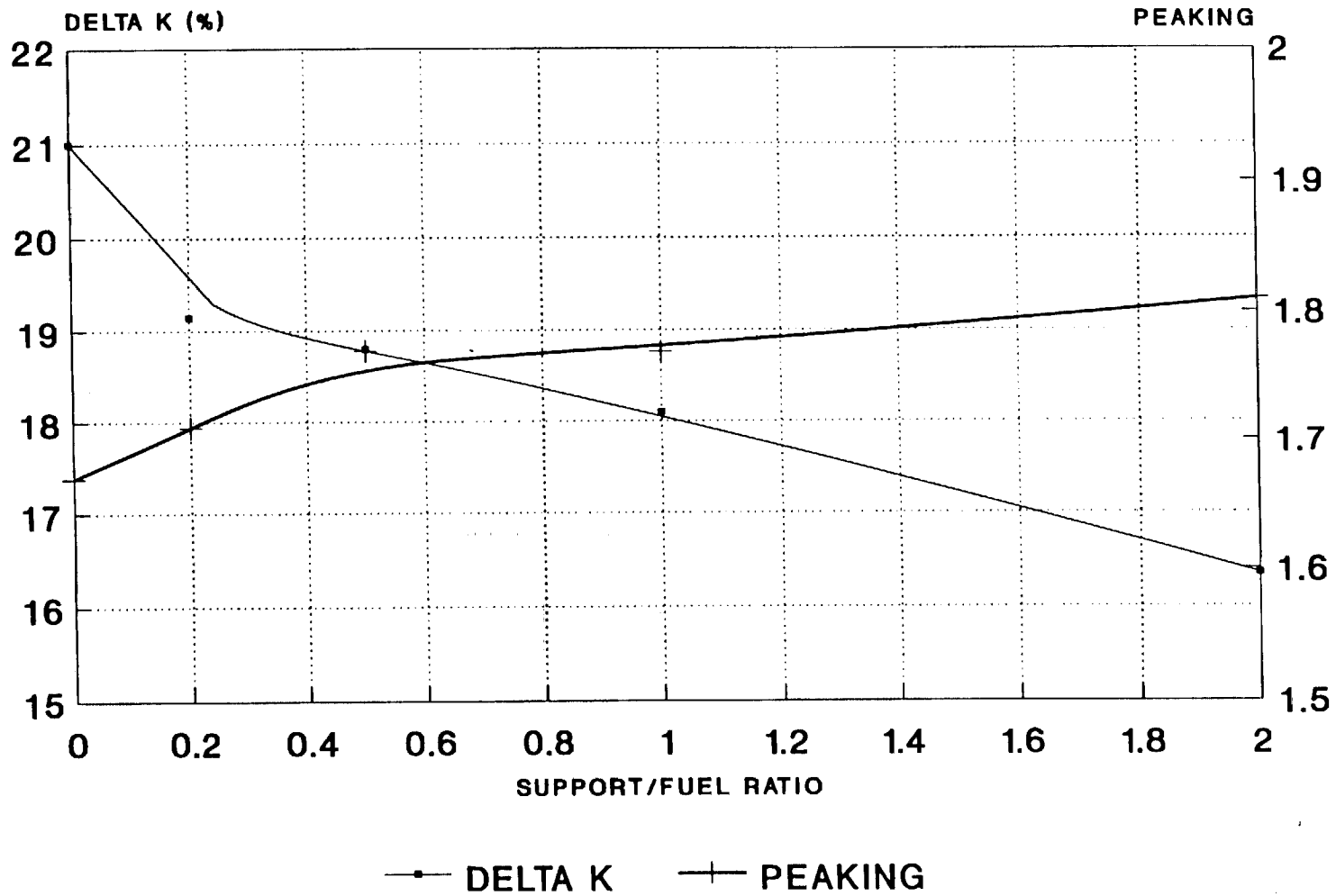

Fig. 7. Reactivity worth of a 360°, 2 mm thick, natural B_4C located between the core and reflector as a function of S/F, also radial peaking factors are shown.

4.2 FUEL PIN DATABASE

The two free parameters of this design are the enrichment and the pitch of the hexagonal lattice; a possible third one, the diameter of the fuel rod, was fixed to 6.4 mm.

4.2.1 Neutronic Properties of the Core

Figures 8 and 9 show k_∞ and critical bucklings as function of enrichment and pitch to diameter ratio, P/d. The corresponding critical masses for spherical bare reactors are shown in Fig. 10. The comparison of this figure with Fig. 1 shows that a considerably larger amount of fuel can be assembled with the fuel pin option. Figure 11 shows spectras averaged in the core for reflected spheres.

4.2.2 Reactivity Worth of Beryllium Oxide Reflector

The reactivity worth of a 25 cm thick BeO reflector is shown in Table 2 as function of enrichments and P/d. The worths were calculated as the difference between the multiplication constant of radially reflected and bare critical cylindrical cores with optimum shape. For a different reflector thickness the worth of the reflector is multiplied by the factor of Fig. 4 due to the incomplete database.

Table 2. Reactivity worth (%) of 25 cm BeO reflector for the fuel pin core

P/d	Enrichment (%)				
	93.0	**80.0**	**70.0**	**60.0**	**50.0**
1.1	19.26	17.26	15.28	12.78	9.57
1.2	18.57	16.30	14.11	11.40	8.04
1.3	17.65	15.18	12.85	10.02	6.58
1.4	16.61	13.99	11.55	8.65	5.22
1.5	15.50	12.76	10.25	7.32	3.95

4.2.3 Reactivity Worth of Burnup and Isotopics

Because of the lack of a database for this effect a simplified analytical approach was chosen. Assuming that the leakage does not change with burnup, the change in the multiplication constant because of the burnup (BU) is:

$$\Delta k(BU) \equiv k(BU) - k(0) = \frac{k_\infty(BU)}{k_\infty(0)} - 1 \qquad (1)$$

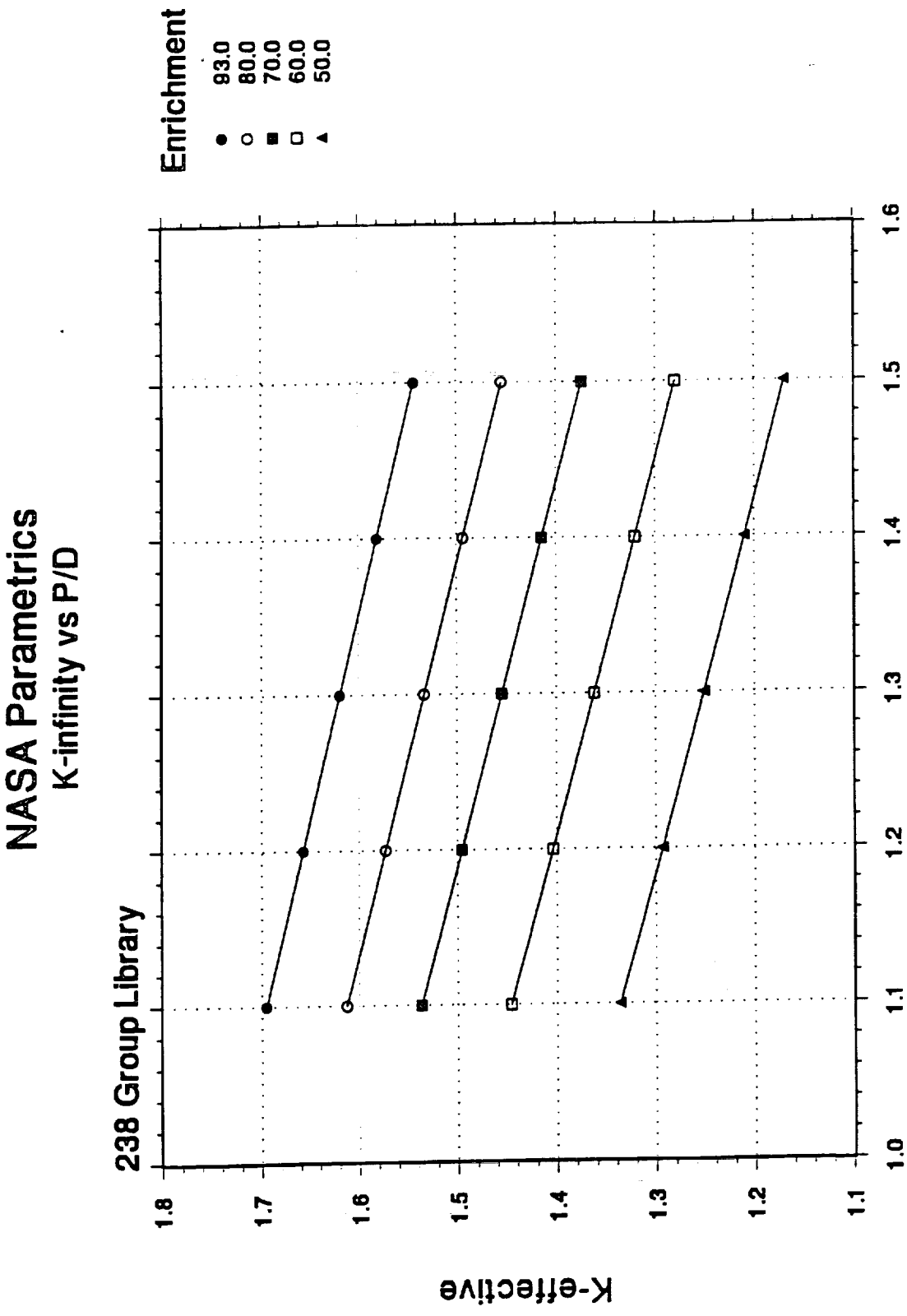

Fig. 8. Fuel pin reactor, k_∞ as a function of pitch and enrichment.

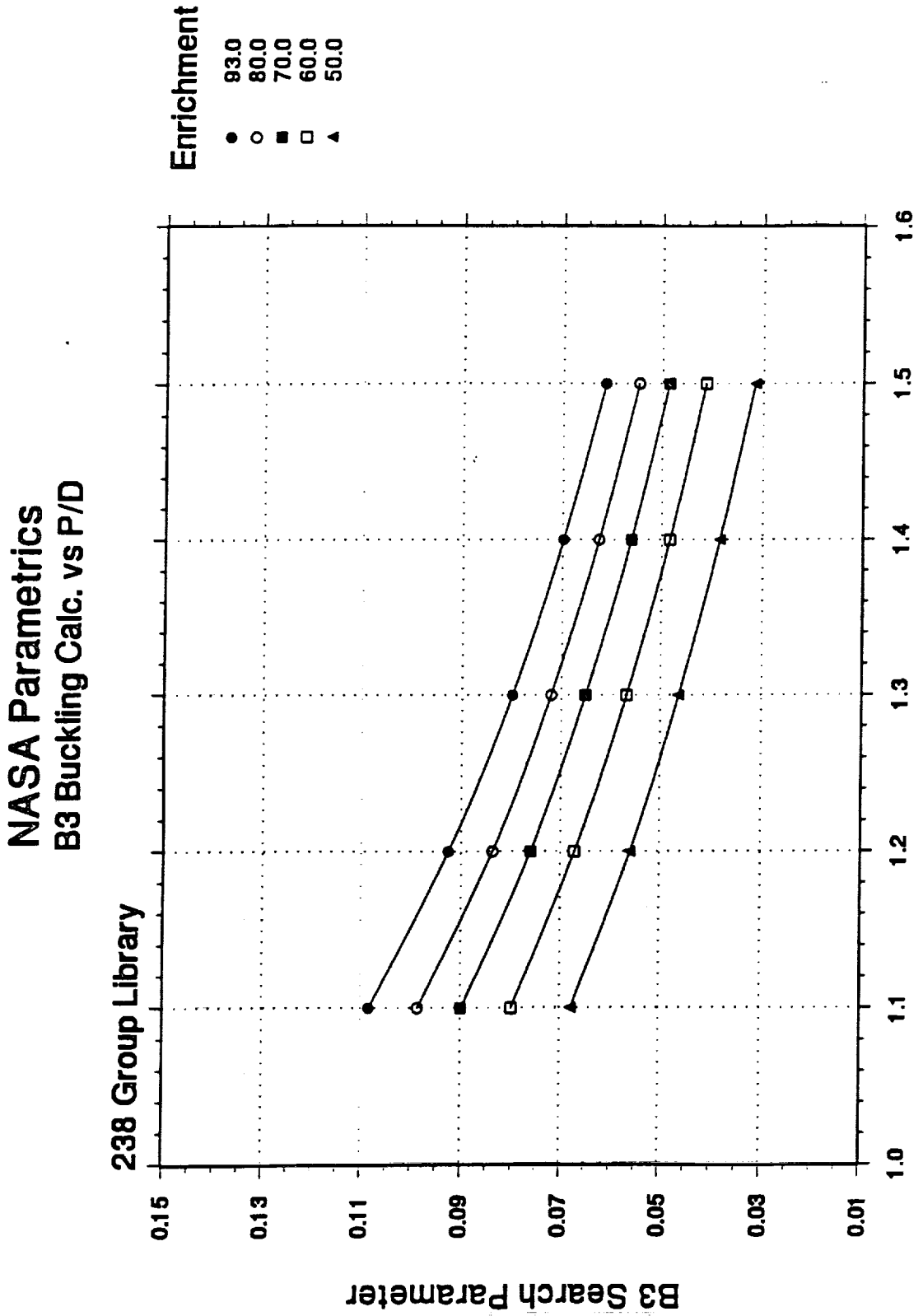

Fig. 9. Fuel pin reactor, critical buckling (cm^{-1}) as a function of pitch and enrichment.

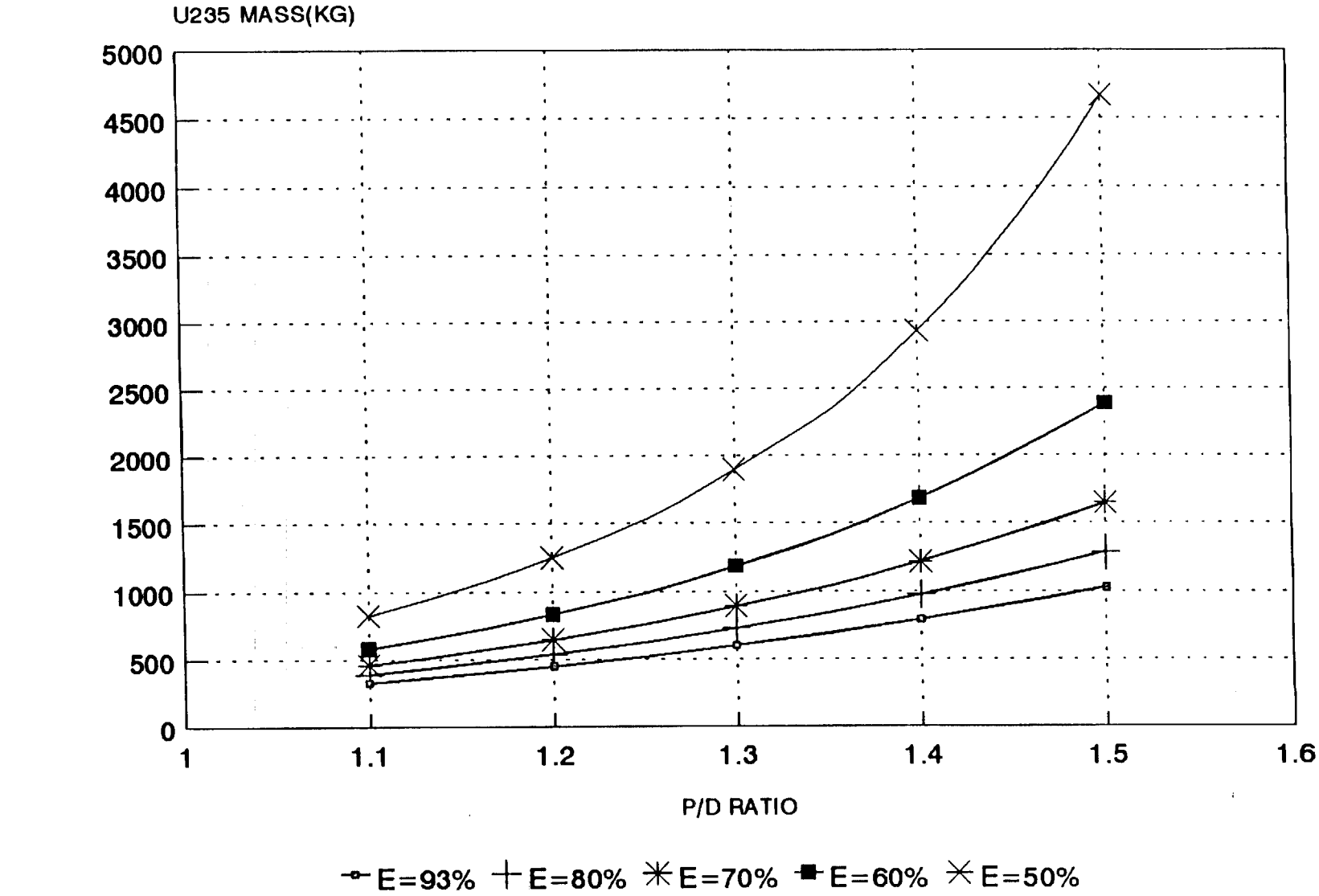

Fig. 10. Fuel pin reactor, critical masses for spherical bare cores.

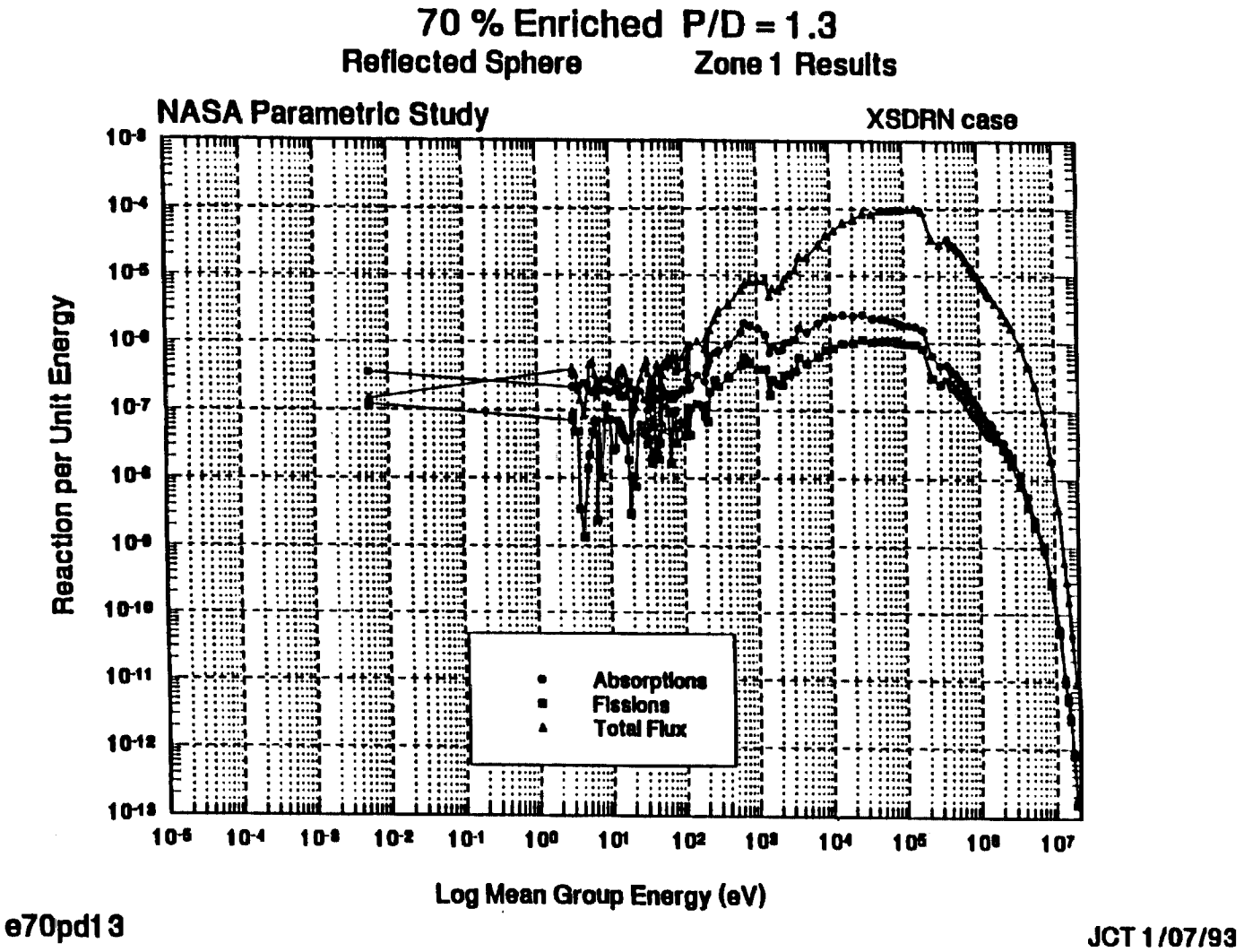

Fig. 11. Fuel pin reactor, spectras averaged in the core for reflected spheres.

where the multiplication constant for the infinite lattice is

$$k_\infty = \frac{\nu_5 n_5 \sigma_{f5} + \nu_8 n_8 \sigma_{f8} + \nu_9 n_9 \sigma_{f9}}{n_5 \sigma_{a5} + n_8 \sigma_{a8} + n_9 \sigma_{a9} + n_{fp} \sigma_{a,fp} + \Sigma_r} \quad (2)$$

where indexes 5, 8, and 9 stand for ^{235}U, ^{238}U, and ^{239}Pu, *fp* for fission products, and *r* for the rest of the mixture. The σ's stand for cross sections averaged in the neutron spectra and n_5, n_8, n_9, and n_{fp} are the isotopic concentrations which are functions of the burnup. The burnup is defined as the integrated number of fissions relative to the initial concentration of ^{235}U atoms,

$$BU \equiv \frac{f_5 + f_8 + f_9}{n_5(0)} \quad . \quad (3)$$

Note, that because of the way we compute the burnup this definition refers to depletion via fission only and is slightly different with respect to the case of the NERVA reactor.

Writing and solving the buildup and depletion equations for the isotopic and assuming that there is no Pu at BOL, we obtain

$$BU = \frac{\sigma_{f5}}{\sigma_{a8}}\left(1 - e^{-\sigma_{a5}\phi t}\right) + \frac{\sigma_{f8}}{\sigma_{a8}}\left(\frac{1}{\varepsilon} - 1\right)\left(1 - e^{-\sigma_{a8}\phi t}\right)$$
$$+ \frac{\sigma_{c8}}{\sigma_{a9} - \sigma_{a8}}\left(\frac{1}{\varepsilon} - 1\right)\left[\frac{\sigma_{f9}}{\sigma_{a8}}\left(1 - e^{-\sigma_{a8}\phi t}\right) - \frac{\sigma_{f9}}{\sigma_{a8}}\left(1 - e^{-\sigma_{a9}\phi t}\right)\right] \quad (4)$$

where indices f, a, and c stand for fission, absorption and capture, ε is the initial enrichment, and the σ's were taken from Ref. 4. For an input burnup, BU, Eq. 4 is solved for the fluence ϕt which is then used to find the isotopic concentrations necessary to evaluate Eqs. (1) and (2).

4.2.4 Reactivity Worth of the Pressure Vessel

Because in this design the pressure vessel is located between the core and the reflector it is relevant to the reactivity balance. Table 3 shows the reactivity worths as a function of enrichment and P/d.

Table 3. Reactivity worth (%) of 1 cm thick astar alloy pressure vessel*

P/d	Enrichment (%)				
	93.0	80.0	70.0	60.0	50.0
1.1	5.69	5.43	5.10	4.57	3.73
1.2	5.39	5.06	4.63	3.99	3.05
1.3	5.12	4.66	4.14	3.44	2.45
1.4	4.77	4.23	3.67	2.92	1.91
1.5	4.41	3.82	3.22	2.44	1.43

*Worth is negative, 25 cm BeO reflector, critical bare cylindrical core with optimum shape.

The worth of the pressure vessel is multiplied by the factor given by Fig. 4 to consider the effects of a different reflector thickness.

4.2.5 Reactivity worth of B_4C sheets

The reactivity worths of 360° 2 cm thick, 90% enriched B_4C sheets located between the pressure vessel and the BeO reflector are shown in Table 4.

Table 4. Reactivity worth (%) of 360°, 2 cm thick, 90% enriched B_4C sheets*

P/d	Enrichment (%)				
	93.0	80.0	70.0	60.0	50.0
1.1	6.73	5.98	5.23	4.29	3.12
1.2	6.64	5.74	4.92	3.91	2.69
1.3	6.37	5.43	4.55	3.50	2.24
1.4	6.07	5.08	4.15	3.06	1.81
1.5	5.73	4.69	3.73	2.63	1.39

*Worth is negative, 25 cm BeO reflector, critical bare cylindrical core with optimum shape and a 1 cm thick astar alloy pressure vessel.

The table shows low effectiveness of the B_4C sheet because of the shielding effects of the 1 cm thick pressure vessel. Better location for the sheet might be found, but no additional data is available. Corrections to the database because of different thicknesses for the pressure vessel and the B_4C sheets are discussed in Section 6.

5.0 SHIELDING CALCULATIONS

The materials for the shielding are LiH in a stainless steel matrix, or B_4C, for the neutrons, and tungsten for the gammas, in a shadow shield configuration. The input parameters for the designs are accepted values for fluences and integrated gamma doses for the entire duration of the mission at a cargo area located at z meters from the base of the reactor and with radial size r.

After its evaluation, the source term for each radiation is affected by the self shielding of the reactor materials. Removal cross sections and buildup factors are used to calculate the thickness of the shield and the radial sizes are chosen by assuming the shield plate is at the base of the reactor and that there is no line of sight from any point of the reactor to the cargo area. Both shielding requirements were calculated together because one type of shielding material shields, marginally, the other type of radiation.

5.1 NEUTRON SHIELDING

For an input power, the source term of fast neutrons is calculated assuming a value of 200 Mev per fission. The self shielding, f_s, for an uncollided current of fast neutrons in spherical geometry, and uniform composition within radius R, is

$$f_s = \frac{3}{a}\left\{\frac{1}{2} - \frac{1}{a^2}\left[1-e^{-a}(1+a)\right]\right\} , \qquad (5)$$

where $a = 2\Sigma R$ and Σ is the removal cross section. We had used an equivalent R for the total volume of the core, reflector and pressure vessel and a uniform density of materials with removal cross section from Ref. 5. Given the source term, the self shielding f_s, the accepted fluence, and the removal cross sections of the LiH-ss or B_4C, the thickness of the neutron shield is fixed.

5.2 GAMMA SHIELDING

The source term is calculated from the total number of fissions and captures, both determined by the power level. The capture reaction rate is then distributed into their material components with relative capture rates read from tables (case of NERVA derivative) or calculated with one-group capture cross sections (the case of the fuel pin). The next step is to multiply capture and fissions rates with the 9 group gamma production spectra per atom taken from Ref. 5.

The self shielding factor is given also by Eq. (5) but with the final value attenuated by the buildup factor due to the scattering of gamma rays. Removal cross sections and buildup factor parameters are also from Ref. 5. The definition of buildup factors for a mixture, even homogeneous, is not obvious; we have interpolated the tables with an equivalent atomic number for the mixture defined with the Z of the components weighted with their total gamma reaction rates. Given the source terms, the self shielding f_s and the accepted gamma dose, the thickness of the tungsten plate is determined after reading cross sections and buildup factors from the data base for the 9 energy groups.

6.0 THERMALHYDRAULICS

Thermalhydraulic conditions are calculated with the approximation of one-dimensional equations for the conservation of mass, energy and momentum. Three steps are involved in the calculations related with the calculations of the bulk conditions of the coolant, the temperature of the wall of the channel and the temperatures within the fuel.

Thermalhydraulic variables depend on power and flow conditions. Four are considered in this analysis and are described in Table 5.

Table 5. Power and flow conditions

Case	Power distribution		Flow distribution	Comments
	r,θ	z	r,θ	
1	Uniform	Uniform	Uniform	3-D fuel grading
2	Uniform	Cosine	Uniform	2-D (r,θ) fuel grading
3	Not uniform	Cosine	Proportional to local power	Uniform fuel, flow conditioning
4	Not uniform	Cosine	Uniform	Uniform fuel

Case 1 is the most relaxed and case 4 the most extreme; intermediate cases 2 and 3 give very similar results. Please note that fuel grading, cases 1 and 2, implies reactivity penalties not considered in the balance of reactivity in Section 3.

Power density, released as heat, at the location of the fissile material is

$$\rho_P = \bar{\rho}_P \frac{f_1}{f_2} F(r,\theta,z) \tag{6}$$

where $\bar{\rho}_p = P/\pi R_c^2 H_c$ is the fission power density averaged into core of radius R_c and height H_c, f_1 is the fraction of the fission power released as heat inside the core, f_2 is the volumetric fraction of the fissile material relative to the core volume and $F(r,\theta,z)$ is the power distribution of the homogenized core,

$$F(r,\theta,z) = \frac{\pi R_c^2 H_c \psi(r,\theta,z)}{\int_{core} \psi(r,\theta,z)\, dV} \tag{7}$$

We assume the distribution Ψ can be written as

$$\Psi(r,\theta,z) = R(r,\theta)\, Z(z) \quad . \tag{8}$$

Consequently

$$F(r,\theta,z) = f_z\, f_{r,\theta} \tag{9}$$

where

$$f_z = \frac{Z(z)}{\frac{1}{H_c}\int_0^{H_c} Z(z)dz} \tag{10}$$

and

$$f_{r,\theta} = \frac{R(r,\theta)}{\frac{1}{\pi R_c^2}\int_0^{R_c} rdr \int_0^{2\pi} R(r,\theta)\, d\theta} \quad . \tag{11}$$

We further assume that $Z(z)$ can be approximated as

$$Z(z) = \sin\frac{\pi z}{H_c'} \quad , \tag{12}$$

where $H_c' = H_c + 2\delta$, with δ being the extrapolated length and the fuel occupying the space between $z = \delta$ and $z = H_c + \delta$. Eq (10) can then be approximated as

$$f_z = \frac{\alpha}{\sin\alpha}\sin\frac{\pi z}{H_c'} \tag{13}$$

where $\alpha = \pi H_c/(2H_c')$.

Because of the even distribution of control drums around the reflector and because the assumed even movement of the absorbing sheets, $R(r,\theta)$ is a periodic function in θ with period $\theta_d = 2\pi/N_d$ where N_d is the number of drums, $R(r,\theta)$ can then be expanded in Fourier series, resulting in

$$f_{r,\theta} = a_0(r) + \sum_{n=1}^{\infty} \left[a_n(r) \cos n2\pi \frac{\theta}{\theta_d} + b_n(r) \sin n2\pi \frac{\theta}{\theta_d} \right] , \tag{14}$$

where

$$a_n = \frac{\bar{R}_n(r)}{\frac{1}{\pi R_c^2} \int_0^{R_c} \bar{R}_0(r) 2\pi r dr} \tag{15}$$

$$b_n = \frac{\bar{R}'_n(r)}{\frac{1}{\pi R_c^2} \int_0^{R_c} \bar{R}_0(r) 2\pi r dr} \tag{16}$$

$$\bar{R}_n(r) = \frac{1}{\theta_d} \int_0^{\theta_d} R(r,\theta) \cos\left(n2\pi \frac{\theta}{\theta_d}\right) d\theta \tag{17}$$

and

$$\bar{R}'_n(r) = \frac{1}{\theta_d} \int_0^{\theta_d} R(r,\theta) \sin\left(n2\pi \frac{\theta}{\theta_d}\right) d\theta \quad . \tag{18}$$

At **BOL** and with the absorbing sheets looking to the core, we assume that $f_{r,\theta}$ is maximum at the center of the core, where $a_n = b_n = 0$ for all $n \neq 0$, i.e.,

$$f_{r,\theta}]_{max} = a_0(r = 0) \tag{19}$$

and that $f_{r,\theta}$ is minimum at $r = R$ and in front of the absorbing sheet

$$f_{r,\theta}]_{min} = a_0(r = R_c) - \Delta_\theta \tag{20}$$

where $-\Delta_\theta$ is the sum of Eq. (14) evaluated at $r = R_c$ and the θ corresponding to the minimum, or

$$f_{r,\theta}]_{min} = a_0(R_c)\left[1 - \frac{\Delta_\theta}{a_0(R_c)}\right] . \tag{21}$$

We had tabulated $a_0(r)$ by using the approximation of considering a continuous 360^0 sheet of absorber rather than a collection of discrete strips around the core. Also the angular tilt in the bracket of Eq. (21) was left as an input variable in the programming. In this way, and for cases 3 and 4 of Table 5, the program computes the thermalhydraulic conditions for the two extreme channels.

6.1 BULK CONDITIONS OF COOLANT

For the case of the bulk conditions of the coolant and considering no gravitational effects and one phase flow, the equations of conservation are

$$\rho v = \rho_i v_i \tag{22}$$

$$h + \frac{v^2}{2} = h_i + \frac{v_i^2}{2} + \frac{P_i}{w} \tag{23}$$

and

$$w\frac{dv}{dz} + A\frac{dp}{dz} + \frac{1}{2}\rho v^2 f P_f = 0 \tag{24}$$

where the channel was subdivided into nodes, subindex i refers to the inlet conditions and the unsubscripted variables refer to the outlet conditions. Variables ρ, v, h and p are density velocity, enthalpy and pressure of the fluid, w is the constant mass flow through a constant area A, P_i is the power delivered to the node, f is the friction factor and P_f the wet perimeter.

6.1.1 Bulk Conditions of the Helium Coolant

Because of the compressible flow and the high speed of the coolant, Eqs. (22) to (24) have to be solved without further simplifications. Combining Eqs. (22) and (23) we obtain

$$F(p,T) \equiv h - h_i - \frac{P_i}{w} - \frac{v_i^2}{2}\left(1 - \frac{\rho_i^2}{\rho^2}\right) = 0 . \tag{25}$$

Using the equation of state, Eq. (24) can be written as

$$\frac{dp}{dz}\left[1 - \frac{v^2}{c^2}\right] + \frac{w^2}{A^2 p}\left[\frac{fP_f}{2A} - \frac{1}{\rho}\frac{\partial \rho}{\partial T}\frac{dT}{dz}\right] = 0 \qquad (26)$$

where $c = 1/(\partial \rho/\partial p)_T$ is the isothermal speed of sound. Integrating Eq. (26) along the node we have

$$G(p,T) \equiv p - p_i + \bar{F}_1 \Delta z + \bar{F}_2 (T - T_i) = 0 \qquad (27)$$

where $\bar{F}_1$ and $\bar{F}_2$ are the average values within the node of

$$F_1 = \frac{w^2 f P_f}{2A^3 \rho} \Big/ \left(1 - \frac{v^2}{c^2}\right) \qquad (28)$$

and

$$F_2 = \frac{w^2}{A^2 \rho^2}\frac{\partial \rho}{\partial T} \Big/ \left(1 - \frac{v^2}{c^2}\right) . \qquad (29)$$

The averages are approximated by weighing the inlet and outlet conditions

$$\bar{F}_e \equiv \frac{1}{1+x}\left[F_e(p_1,T_1) + x F_e(p,T)\right], \; e = 1,2 \qquad (30)$$

where x is the relative weight. The outlet conditions, p and T, are obtained by the simultaneous solutions of Eqs. (25) and (27). Because Eq. (25) is mainly dependent on T, a first estimation of the temperature $T^{(1)}$ is made by using p_i instead of p in Eq. (25). With $T^{(1)}$, Eq. (27) is solved for p to obtain a first estimation of $p, p^{(1)}$; then F and G are linearized around $p^{(1)}$, $T^{(1)}$ to find p, T. In the whole process x is an input value.

When v approaches sonic velocities the entire one dimensional approach is no longer valid. If this condition occurs, the program prints warnings and restarts calculations relaxing inlet and power conditions.

Friction factors, thermodynamics, and transport properties of the He were taken from Ref. 6. For Reynolds, Re, number below 2,000, ($Re = 4w/\mu P_f$, μ viscosity),

$$f = \frac{16}{Re} \quad , \tag{31}$$

and above $Re = 2000$ the Moody approximation to the Colebook-White correlation was used

$$f = 0.001375\left[1 + \left(20{,}000.\frac{\varepsilon}{d} + \frac{10^6}{Re}\right)^{1/3}\right] \quad , \tag{32}$$

which includes ε, the rugosity of the channel with diameter d.

6.1.2 Bulk Conditions of the Liquid Lithium

Considerable implications are possible in this case; because the dependence of the enthalpy with the pressure is negligible as well as the velocity terms, Eq. (25) is used to evaluate T which then defines ρ and the velocity. With the known velocity gradient and the average of the friction term in the node also known, Eq. (24) can be immediately integrated to find p.

The fuel rods are assembled in hexagonal bundles and they are separated with helicoidal spacing wires. One result of this arrangement is that there are three types of cells, central, lateral and at the corner with different flow fractional areas. The thermalhydraulics calculations are done for the 3 cells and the average conditions across the bundle are then calculated. For the calculation of the average pressure distribution in the bundle, we have used the Novendstern correlation taken from reference 7 which includes the effects of the spacing wire and the different geometries of the 3 cells within the bundle. The temperature of the hottest point in the coolant is monitored against the saturation temperature to warn the user about potential local boiling. Thermodynamics and transport properties of Li were taken from reference 8.

6.2 WALL TEMPERATURE

After the calculation of the bulk temperature of the coolant, the wall temperature, T_w, is calculated with the equation

$$T_w = T_b + \frac{j}{h} \quad , \tag{33}$$

where j is the power current and h is the heat transfer coefficient, obtained from correlation for the Nusselt number $Nu = hD/k$.

6.2.1 Wall Temperature of the NERVA Derivative Fuel Element

Although velocities of the He coolant are not very high for NEP, the program was prepared to deal with the case of high speed flow, like in the case of NTP. At very high speed, T_b in Eq. (33) is substituted by T_b' which is given by

$$F_R = \frac{T_b' - T_b}{T_s - T_b} \quad , \tag{34}$$

where the recovery factor, $F_R = \sqrt[3]{Pr}$ (Prandtl number, $Pr = C_p \mu / k$) and the stagnation temperature, T_s, is given by

$$h(T_s, p) = h(T_b, p) + v^2/2 \quad . \tag{35}$$

The correlation of Mc Eligot et. al.[6] was chosen for the Nusselt number. It is the well known Dittus-Boelter correlation with corrections because of entry effects and non-negligible differences between wall and bulk temperatures. The complications of all these corrections make the wall temperature an implicit variable so an iterative method is necessary.

6.2.2 Wall Temperature of the Fuel Rods

Because of the excellent heat conductivity of the liquid lithium and the higher density we do not have the complications of the previous case, namely aerodynamics and temperature gradient effects, so explicit calculations with Eq. (33) are possible and no iterations are necessary. The correlations for the Nusselt number in pages 189, 190 of Ref. 7 were used; the correlations depend mainly on the Peclet number, $P_e = RePr$, and are specially fitted for rods in a bundle. They are parametric in P/d and in the effective ratio of the eddy diffusivity of heat to momentum, Ψ, which depends on Prandtl and Reynolds numbers and P/d.

6.3 FROM WALL TO FUEL TEMPERATURES

Once the wall temperature is known, the temperature distributions inside the solid fuel element are calculated by solving the one dimensional (this time the radial direction) heat conduction equation. The maximum fuel temperature is then compared with the melting temperature of the fuel to warn the user to relax the input requirements.

6.3.1 Fuel Temperature of the NERVA Derivative Reactor

The coolant flows through holes in the hexagonal fuel element; centered in the coolant hole and for increasing radius r we find the wall at $r = r_0$, the fuel at $r = r_1$ and the boundary of the heat cell at $r = r_2$. Between r_0 and r_1 we have the ZrC clad, and between r_1 and r_2 a mixture of graphite and $UC - ZrC$; the radius of the cell, r_2, is determined by distributing the area of the mixture evenly between the nineteen holes.

The temperature distribution equations are subject to the boundary conditions $T(r = r_0) = T_w$ (from Section 5.2) and $\partial T / \partial r)_{r=r_2} = 0$.

The heat conductivity (k) of the different materials was taken from the available literature. If better values become available the subroutines could be reprogrammed easily. The k of the cladding was taken from Ref. 10, $k(ZrC) = 0.2077 \ \frac{watt}{cmK}$; the k of the mixture is calculated as

$$k(mix) = x(UC)\ k(UC) + x(ZrC)\ k(ZrC) + x(graph)\ k(graph) \tag{36}$$

where the $x's$ are the mass fractions in the mixture. The temperature dependent $k(UC)$ values were taken from Ref. 9. The k of graphite exhibits large anisotropic and irradiation effects;[10] no intentions were made to introduce these additional complications so average temperature dependent $k's$ without irradiation effects were used. Graphite melts at 3889^0K above the melting temperature of the $UC - ZrC$ mixture which depends on the relative concentration. Melting temperatures of the UC-ZrC mixture were taken from Ref. 10.

6.3.2 Fuel Temperature of the Fuel Pin Reactor

Centered in the fuel rod we have the UN fuel pin to $r = r_p$, the He gap to $r = r_c$ and the tungsten liner and astar alloy clad up to $r = r_r$. Heat conductivities were taken from reference 4 and the heat transfer of the He gap was computed as $k(He)/$ gap thickness, with $k(He)$ from Ref. 6. This value is a simplification; very detailed discussions about this parameters can be found in Ref. 7. Boundaries conditions are $T(r = r_r) = T_w$ (from Section 5.2.2) and $\partial T/\partial r)_{r=0} = 0$. Melting temperature of UN and W quoted in the program are from Ref. 11.

6.4 THERMALHYDRAULIC FEEDBACK VIA THE PRESSURE VESSEL

Strictly speaking the thermalhydraulic conditions have an effect over the neutronic calculations of Section 3. This feedback was not considered because it is a small effect in comparison with the scoping nature of the calculations and the uncertainties of the models. The exception is the pressure vessel for the case of the fuel pin reactor. The thickness d of the pressure vessel is determined in general by the equation[12]

$$d = \frac{pR}{\sigma} \tag{37}$$

where R is the external radius of the Be reflector for the NERVA derivative reactor or the radius of the core for the fuel pin reactor, p is the pressure, and σ is the maximum stress allowable for the pressure vessel. We have used a subroutine from the ALKASYS[4] code to calculate σ, determined by the temperature and the life of the reactor. The temperature of the pressure vessel is an input variable. If it is not specified, the code chooses the exit temperature of the coolant as the temperature of the pressure vessel.

Because the pressure vessel has an impact on the reactivity balance (see Table 3) its worth is included in the reactivity balance.

7.0 ORGANIZATION OF THE CODES

In order to find a design that meets the input demands two balance equations have to be simultaneously solved; one for the reactivity, the other for the energy. The major input variables are the reflector thickness, δ_R, the burnup at EOL, BU, the power level, P, the operation time, D, and the boundary conditions for the thermalhydraulics: inlet conditions of the coolant, p_i and T_i, and either its outlet temperature, T_o, or mass flow, w, and the temperature of the pressure vessel, T_{pv}.

The coolant outlet temperature or its flow is then calculated with an enthalpy balance equation neglecting the pressure drop along the reactor. If T_{pv} is not specified the code uses T_o as the temperature of the pressure vessel. T_{pv} and D are used then to compute σ, the maximum stress for the astar alloy.

7.1 SELECTION AND CALCULATION OF A DESIGN

In order to write the reactivity balance equation, the worth of each component is read from tables or computed. The reactivity worth of the radial reflector, $\Delta k_{(R)}$, is read from tables for a reference thickness and then modified by the factor of Fig. 4 because of the effects of input δ_R. The worth of axial reflectors, if any, are computed in the same way with the additional factor of 1/2, which is the ratio for axial over radial leakages for ideal shaped cylinders.

The reactivity worth of the burnup Δk(BU) is also read from tables or computed, and the reactivity worth of the pressure vessel Δk(PV) is parameterized in terms of σ and the radius of core; additional terms are the desired reactivity at EOL, Δk(EOL), and an estimation of the reactivity effects of structural parts Δk(SP) which are assumed input variables.

The reactivity balance equation is then written as:

$$k(EOL) \equiv 1 + \Delta k(EOL) = k_b + \Delta k(R) - \Delta k(BU) - \Delta k(PV) - \Delta k(SP) \qquad (38)$$

where the multiplication constant of the bare core is:

$$k_b = \frac{k_\infty}{1 + M^2 B^2} \qquad (39)$$

and the geometrical buckling is

$$B^2 = \left(\frac{\pi}{H_c}\right)^2 + \left(\frac{2.405}{R_c}\right)^2 . \qquad (40)$$

Equations (38) and (39) are then solved for the size of the core by specifying either ideal shape $(R_c/H_c = 0.54)$, R_c or H_c; in this process, some of the Δk's might be a function of the size so iterations are necessary (see next sections).

With R_c and H_c and the composition of the core, the total mass of fuel can then be calculated. Given this mass, the burnup and the scaling of 200 MeV per fission, the calculation of the available energy, Ea, that the reactor can release during time D is determined.

Figures 12 and 13 are examples of the available energy for both reactors. In order to match the requested energy Er = PD, the free parameters of each design are iterated until Ea = Er. If this cannot be done because of contradictions between the requests and what is possible, the codes relax, under options, the input conditions.

The organization of the codes follows the shapes of Figs. 12 or 13. The scheme for the exploration of possible designs is shown in Fig. 14. For a given set of input conditions, the possible designs are within the region of the (Ea, x) plane limited by parametric curves p_1 and p_n. For example, p_1 = S/F = 2, and p_n = S/F = 0, for the NERVA reactors and p_1 = enrichment = 93% and p_n = enr = 50% for the fuel pin reactors with P/d = 1.1. The parametric curves have a minimum corresponding to the ideal shape, vertical asymptotic lines A_1, A_n to the left and monotonic growth to the right. For example, if x is the radius of the core, to the left of the minimum we have cigar shape designs while pancake shapes correspond to the region to the right.

For each parameter, the codes calculate the asymptotic lines and the energy available for the optimum shapes, E_o; under a test option these values are printed together with *Ea* as function of R_c and H_c.

If the requested energy has a value, Er_1, below the minimum of the Eo, Eo_{min}, i.e., if Er is in region 1 of Fig. 14, there is a warning message and the codes, under option, proceed to change the requested input burnup, lowering it until Eo_{min} is smaller than Er_1 and choosing the ideal shape disregarding any input R_c or H_c.

If the requested energy has a value, Er_2, intermediate between the minimum and maximum of Eo, i.e., Er is in region 2 of Fig. 14, two things can happen: (1) if the ideal shape was chosen the codes proceed with their calculation, or (2) if the user chooses R_c or H_c then the codes compute the allowable ranges x_1 and x_2 in Fig. 14 which are then compared with the input dimension. If consistent, the codes proceed with the calculations; if not and under option the codes change the input dimension to 0.5 (x_1 + x_2).

If the requested energy has a value of Er_3 within region 3 of Fig. 14, designs are only possible between x_1 and x_2 and between x_3 and x_4 and ideal shape designs are not feasible. The codes then compute x_1, x_2, x_3, and x_4 and switch the option for the shape to a core with dimension x = 1.01 x_3 if allowed.

Assured now that the requested energy and shape are within curves p_1 and p_n of Fig. 14, designs with parameters p_1 and p_n bracket the demand so the codes iterate until a value p_r is found that produces the required energy Er.

The calculations are then continued with the computation of the number densities and masses of the core, the reflector and the pressure vessel. For the case of the fuel pin reactor, the entire process is made for P/d = 1.1, 1.2, 1.3, 1.4, and 1.5; one design is then chosen either by the user or by the code which chooses the design with the lowest total mass from those that have a pressure drop below a prescribed input value. With a design chosen, the codes proceed to calculate the geometries and masses of the neutron and gamma shieldings and the thermalhydraulic conditions for the most and least heated channels.

7.2 DESIGN OF THE CONTROL DRUMS

The number of control drums is, in principle, an input variable, n_d, but it can be changed by the program. It is compared with the maximum number $n_{d(max)}$ = Integer($2\pi/\Delta\theta$) where $\Delta\theta = 2 \arcsin(r_d/(R_r + r_d))$ is the central parallax of the drums, r_d is the radius of the drums (1/2 of the reflector thickness) and R_r is the internal radius of the reflector; if larger the program continues with $n_d = n_{d(max)}$. The reactivity worth per unit angle is calculated as $w_\theta = \Delta k(B_4C)/2\pi$ where the numerator is read from tables as indicated in Sections 3.1.5 and 3.2.5; the maximum reactivity of the drums, $w_\theta \Delta\theta n_{d(max)}$, is then compared with the reactivity to control $\Delta k_c = k(BOL) - 1. - \Delta k(SP) + \Delta k(SUB)$, where $\Delta k(SUB)$ is the desired subcriticality with the absorbers facing the core. If the maximum worth of the drums is smaller than Δk_c a warning message is printed and the program bypasses what follows.

The input number of drums is used to compute their maximum worth $\Delta k_d = w_\theta \Delta\theta n_d$ and if smaller than $\Delta k_c, n_d$ is increased to a value n_d' such that $\Delta k_d' = w_\theta \Delta\theta n_d' > \Delta k_c$. With n_d now fixed the central parallax of the absorbing part of the drum is computed as $\Delta k_c = w_\theta \Delta\theta_a n_d$ and then converted to local parallax (centered in the drums).

7.3 SOME DETAILS OF THE CODE NEPNERVA

The burnup contributions to the reactivity Eq. (38) is split as the sum of three terms $\Delta k(Xe)$, $\Delta k(Sm)$ and the Δk corresponding to ^{235}U depletion and other fission products as discussed in Section 3.1.3; this was done because as S/F increases the reactor becomes more moderated making necessary the special treatment for the large thermal neutron absorbers ^{135}Xe and ^{149}Sm. The transition of the importance of these isotopes, from irrelevant at S/F = 0 to very serious at S/F = 2, is made smoothly with cross sections that depend on S/F.

Steady and transient Xe and transient Sm reactivity effects depends on the flux; because for a given power the flux depends on the volume, Eq. (38) becomes then implicit on the size of the core and it has to be iterated. Equations for steady and transient reactivity effects for Xe and Sm were taken from Ref. 3.

7.4 SOME DETAILS OF THE CODE NEPPIN

Because of the fast spectrum, the effects described in the previous sections do not appear in this case, instead an iteration because of the pressure vessel becomes necessary when solving the reactivity Eq. (38). The reactivity effects of the pressure vessel of thickness d, around a core of radius R_c inside a reflector of thickness δ_r, was parameterized as

$$\Delta k(PV) = \Delta k_n(PV) \left(\frac{d}{d_n}\right)\left(\frac{R_{cn}}{R_c}\right) f\left(\frac{\delta_r}{\delta_{rn}}\right) \tag{41}$$

where $\Delta k_n(PV)$ are the values quoted in Table 3 at nominal values, $d_n = 1\,cm$, R_{cn} and $\delta_{rn} = 25\,cm$. The values of the function $f\left(\frac{\delta_r}{\delta_{rn}}\right)$ was taken from Fig. 4 due to the lack of better data. Because d is proportional to R_c (Eq. (37)), in general $\Delta k(PV)$ does not depend on R_c and the iteration is not necessary. There is a lower limit to the reflector thickness, 0.4 cm, a value from Ref. 4; when d = 0.4 cm an iteration on R_c is then necessary.

Because of the effects of the pressure vessel on the worth of the B_4C sheets a simple correlation was used to compute $\Delta k(B_4C)$ from the nominal values quoted in Table 4:

$$\Delta k(B_4C) = A\, e^{-\Sigma d}\left(1 - e^{-\Sigma_c d_c}\right) f\left(\frac{\delta_r}{\delta_{rn}}\right) , \tag{42}$$

where the first exponential refers to the effects of pressure vessel of thickness d and the second to the B_4C sheet of thickness d_c; the third factor is related to the changes due to reflector thickness, like in Eq. (41). Equation (42) was calibrated with the data of Table 4 that were calculated for d = 1 cm and d_c = 2 cm.

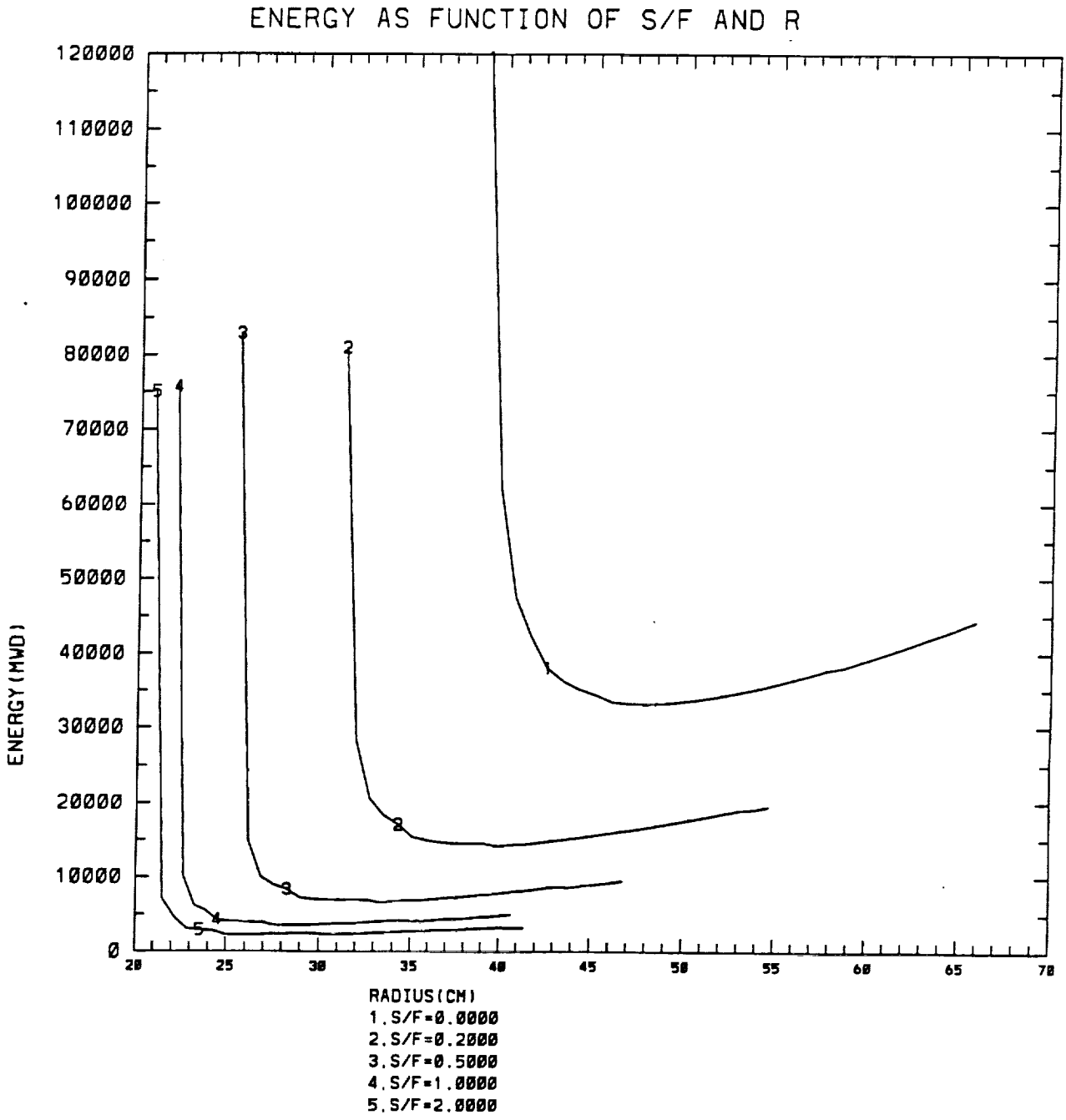

Fig. 12. Available energy in Mwd for the NERVA derivative reactor as a function of S/F, the radius of the core and for the following input conditions: 30 cm Be reflector, 2% reactivity worth for structural parts and desired end of life reactivity, and 15% burnup.

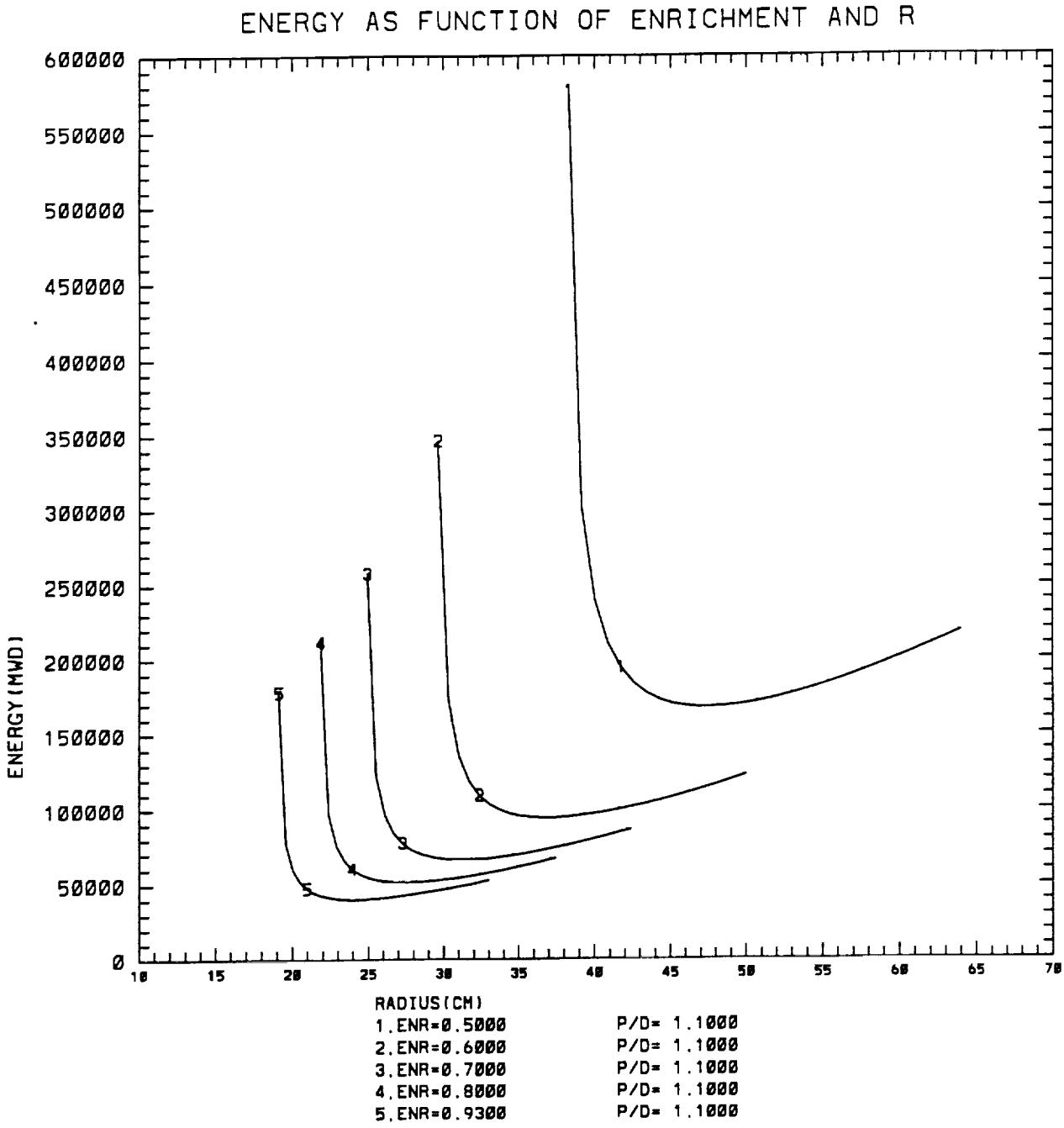

Fig. 13. Available energy in Mwd for the fuel pin reactor as a function of enrichment, the radius of the core and fixing P/d = 1.1. Other fixed input variables are 30 cm of BeO reflector, 2% reactivity worth for structural parts and end of life desired reactivity and 15% burnup.

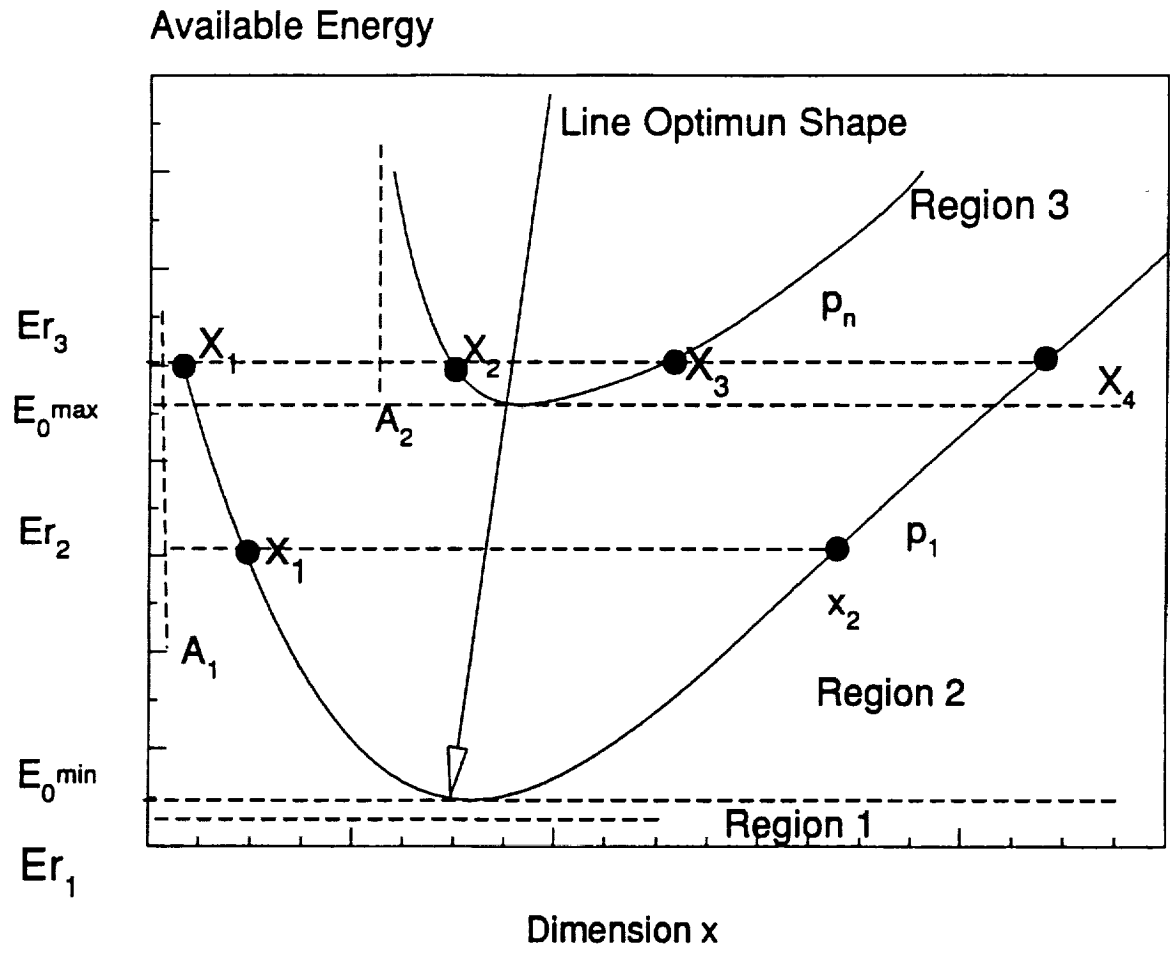

Fig. 14. Scheme for the exploration of possible designs, *Ea* is the available energy and *x* is one of the core dimensions (R_c or H_c).

8.0 INPUT AND OUTPUT DESCRIPTIONS

This section describes the input and output of the codes NEPNERVA and NEPPIN for, respectively, the helium cooled NERVA derivative and liquid lithium cooled fuel pin reactors. The names of the variables are those of the codes and in parenthesis some typical values.

8.1 INPUT TO NEPNERVA

itest/0,1/: Do not/run test, when itest = 1 the code prints the energy available for the set of input conditions.

ndt: Number of points, in the tabulation of the available energy as function of R_c or H_c, between the asymptotic and optimum values of R_c or H_c.

itmax: Maximum number of iterations.
icont/01/Do not/continue calculation after test.
nput: Total number of points in the tabulation of the available energy.
Pow (Mw): Fission power.
D (years): Time during which the reactor is on.
BU (%): Burnup at end of life, i.e., percentage of ^{235}U atoms that disappeared by fission and capture.
eps (0.001): Iterations are stopped within this relative change.
der (1.01): Relative increment to compute derivatives.
drr (cm): Thickness of the radial Be reflector.
dra1 (cm): Thickness of the top axial Be reflector.
dra2 (cm): Thickness of the bottom axial Be reflector.
dkstr (%,2): Reactivity of structural components.
dkeol (%,2): Reactivity at end of life.
dksu (%,2): Subcritical reactivity with drums in.

For test runs only the input might finish here.

ncr: number of drums.
nnodo: number of axial nodes for thermalhydraulic calculations ($\leq$ 50).
nfu: number of radial nodes for temperature calculations of the fuel mixture.
ishape/1,2,3: Shape of the core
1: ideal shape (R_c/H_c = 0.54)
2: input R_c
3: input H_c
iboun/0,1/: Input coolant flow/Outlet coolant temperature.
iopth/1,2,3,4/: Assumptions about the power and flow conditions
1: 3D uniform power distribution.
2: 2D(r,θ) uniform power distribution, cosine axial distribution (1 and 2 with uniform flow distribution).
3: r,θ power distribution corresponding to uniform fuel load, cosine axial distribution and uniform flow distribution.
4: Like 3, but with (r,θ) flow distribution proportional to power distribution.
iexpl/0,1/: No/Yes explicit thermalhydraulic, iexpl=1 eliminates the iterations of Section 5.1.1.
ifri/0,1/: No/Yes inclusion of friction factors in pressure calculations.
iditus/0,1: No/Yes Dittus correlation for heat transfer coefficient, if iditus=0 there is a correction because of radial gradient of temperatures.
iprth/0,1/: No/Yes print details of thermalhydraulic calculations.

iouth/1,2,3/: Printing of thermalhydraulic condition of
- 1: Maximum power density channel
- 2: Minimum power density channel
- 3: Both

iwrte/0,1/ No/Yes printing of iterations.
ishm/1,2/ Selection of neutron shielding
- 1: LiH stainless steel matrix
- 2: B_4C

ichoose /0,1/: For the case of contradictions between demand and availability;
- 0: Program chooses alternative demands.
- 1: Program stops for new demands from user (after printing some advices).

pin(MPa),tin(^oK): Pressure and temperature at the inlet of reactor.
ps(MPa),ts(^oK): Pressure and temperature in support elements, the thermalhydraulics of these elements are not calculated by the code, ps and ts are used just to compute the He density.
flowt(kg/s): Total coolant flow if iboun = 0.
toutl(^oK): Outlet temperature of coolant if iboun = 1.
ru: rugosity of the coolant channel.
xi(1.): x in Eq. (30).
fact(0.97): Factor for the conversion of fission power to thermal power.
ripple(0.1): Maximum ripple of the power distribution in the azimuthal direction (see Eq. (21)).
tempv(^oK): Temperature of the pressure vessel, if input <0 program chooses toutl as the value of this variable.
depvw(cm): Maximum allowed for pressure vessel thickness.
fastf($1/cm^2$): Maximum fast fluence at cargo bay.
dose(rad): Maximum gamma dose at cargo bay.
zload(m): Distance of cargo bay to base of reactor.
rload(m): Radius of cargo bay.
R(cm): Radius of core if ishape = 1.
H(cm): Length of core if ishape = 3.
The input file for the sample case follows.

```
Problem #1 NERVA Application to NEP

itest/# dR/itma/cont?/# of points in test      Format I5
    0   10  450    1   31
Pow(Mw)  /Life(y)  /Burnup(%)/Tolerance/Rel delta   E10.0
50.       2.        20.       0.001     1.01
Radial   /Top      /Bottom Reflector Thickness(cm)
30.       0.        0.
Struct.  /End Life /Subcrit     Reactivities(%)
2.        2.        2.
#drum/#axi/#rad/ishape:1,2,3;ideal,choose R,choose H
   12   50    3    1
ibou/iopth:1,2,3,4(Power,flow conditions);iboun:0,1 Boundary Condition
    1    3
iexpl/ifri/idit/iprt/iouth/explicit,friction,Dittus,PrintDetails,Max,Min,Both?
    0    1    0    1    3
iwrte/Nshield/Ichoose/wrte its,LiH-SS or B4C?, Ichoose or Program?
    0    1    0
Pin(MPa) /Tin(K)   /Pin      /Tin/Inlet Conditions Fuel,Support Elements
2.        400.      5.        400.
iboun=0 input total coolant flow(Kg/s),iboun=1 input outlet T(K)
1700.
rugosity /Implicit /Fraction/P Ripple Angular Power
0.01      1.        0.97      0.1
pressure vessel temperature/max permissible thickness(cm)
800.      10.
Max fast n/Dose(RAD)/z(m)    /r(m) Shielding at z,r for input dose
1.    e+13 1000000. 15.0      5.
ishape/1,2,3/ (skip,input R,input H(cm))
60.
```

8.2 OUTPUT FROM NEPNERVA

The output from running the code with the input file of the previous section follows.

```
Problem #1 NERVA Application to NEP


 INPUT TO THE PROGRAM

 Power(Mw)    50.00 Core life    2.000 years
 Energy Released(Mwday)  36525.00 Burn-up(%) 20.00

 Be reflector, radial thickness(cm) 30.00
               top          "         0.00
               bottom       "         0.00

 The following design can produce that energy within the relative
numerical precision of  0.1591E-03

 Minimum Core Volume Chosen: R/H= 0.541315

 And it has the following reactivity balance
 Delta k required(%) 11.46 with components
  Fission Products(%)  7.46
 " "(except Xe,Sm)(%)  6.52
        Structural(%)  2.00
       End of Life(%)  2.00
 keff at BOL(No structural mat) 1.1146 Components
                    Bare Reactor 0.8206
                       Reflector 0.2940


 Support/Fuel Ratio=  0.0554 Steady Xe reactivity(%)  0.45
 Xe Reactivity after trip(%)  0.45 Peak at time(hr)  1.30
 Sm Reactivity (%) Steady  0.46 Peak after trip(%)  0.49
 Sm saturates at days   64.80

 Neutron Flux(1/cm2sec)(core average) >100.Kev  0.4209E+15
    "       "      "         "     "     >100. ev  0.2291E+15
    "       "      "         "     "     >  3. ev  0.1133E+14
    "       "      "         "     "     <  3. ev  0.3727E+13



   12 Drums with absorbing angle(d)   55.83 each
 Control dk(%)=   11.46


 NUMBER DENSITIES AND MASSES OF NERVA DERIVATIVE REACTOR
         GENERAL DESCRIPTION

   265.412 Kg of U235. U enrichment 0.9300 Core Volume(L)     560.24
 Ratio Support/Fuel Elements   0.05542
 Core Radius(cm)     45.86 Height(cm)     84.78
 Number of elements  2085 Support   110 Fuel  1975
 Hexagonal Elements. Flat to flat distance(cm)    1.9130


           FUEL ELEMENTS(INDIVIDUAL)
 Molar UC fraction in UC-ZrC mix is 0.5000 ZrC clad thickness (mm) is 0.10
 There are  19 coolant holes of mm   2.79 diameter
 with He coolant at  2.000 MPa 1049.45 K

 Volume fractions in Fuel Element
```

```
     Coolant 0.3676
Clad Coolant 0.0545
Clad Fuel El 0.0208
Fuel Mix     0.5571
```

Volumetric Fractions in Fuel Mix * Density(g/cm3)

UC	0.07829	1.01545
ZrC	0.06328	0.42394
Graphite	0.85843	1.45934

Fuel microcell(Number Densities 10**24 atoms/cm3)
Excludes outer cladding

Coolant Radius(cm)	0.1397	He	0.00013774			
Clad Radius (cm)	0.1497	C	0.03908465		Zr	0.03908465
Cell Radius (cm)	0.2280	C	0.00494617		Zr	0.00247308
		Graphite	0.07316656		U235	0.00229997
					U238	0.00017312

FUEL ELEMENTS(ALL)
Volume of Fuel Elements(L) 530.685
Mass(kg) 1124.972 gU5/L(FE) 500.131

	Masses(kg) *	Fractions(%)
Graphite	431.466	38.353
UC	300.228	26.688
ZrC(in fuel)	125.343	11.142
ZrC(in Clad)	267.757	23.801
He(Coolant)	0.179	0.016

SUPPORT ELEMENTS(ALL)
Volume of Support Elements(L) 29.557
Mass(Kg) 91.755

	Masses(Kg) *	Fractions(%)
He	0.028	0.031
Inconel	10.430	11.368
ZrH2	59.067	64.374
Graphite	22.230	24.228

SUPERCELL CONFIGURATION(Number Densities 10**24atoms/cm3)

Support Element						
Coolant Radius(cm)	0.1500	He	0.00089003			
Inconel Radius(cm)	0.2500	Ni	0.06664123	Fe	0.00671785	
		Cr	0.01546157	Ti	0.00279728	
		Si	0.00477070			
ZrH2 Radius (cm)	0.6500	Zr	0.03616986	H	0.07233973	
Coolant Radius(cm)	0.7500	He	0.00089003			
Pyrographite R(cm)	0.8500	Gr	0.08523275			
Graphite Radii(cm)	1.0044	Gr	0.08523275			

Ring of Fuel Elements. Average Densities

Radius(cm)	4.3728	He	0.00005063	C	0.00569895
		Zr	0.00432113	Gr	0.04076312
		U5	0.00128138	U8	0.00009645

HOMOGENIZED DENSITIES IN CORE
Radius(cm) 45.8626

From Fuel Elements

He	0.00004796	C	0.00539829
Zr	0.00409316	Gr	0.03861255
U5	0.00121377	U8	0.00009136

From Support Elements

He	0.00000756	Ni	0.00013941
Fe	0.00001405	Cr	0.00003234
Ti	0.00000585	Si	0.00000998
Zr	0.00068097	H	0.00136193
Gr	0.00198940		

REFLECTOR
External Radius(cm) 75.8626

Be	0.12348490

Moderation by C. C/U235= 37.8985
Moderation by H. H/U235= 1.1221

VOLUMES AND MASSES OF REACTOR AND SHIELDINGS

	Volume(L)	Fraction(%)
Reactor	1563.72	42.65
N Shield	1811.01	49.39
G Shield	291.85	7.96
Total	3666.59	

	Mass(Kg)	Fraction(%)
Reactor	3533.29	32.06
N Shield	1854.48	16.83
G Shield	5632.66	51.11
Total	11020.43	

DIMENSION OF THE SHIELDINGS
LOAD AT Z(M)= 15.00 WITH RADIUS(M)= 5.00
G Shield Thickness(cm) 9.12 With Radius(cm) 99.69 102.19
N " " " 75.22 " " " 102.19 122.79
Pressure Vessel Evaluated at T(K)= 800.00 PRESS(MPa)= 2.00
and Full Power Life(y) 2.00

DISTRIBUTION OF MASSES AND VOLUMES IN THE REACTOR

Fraction of Core Occupied by Fuel(%Vol) 94.724

	Masses(Kg)	* Fractions(%)	Volume(L)	* Fraction(%)
Fuel Elem	1124.972	31.839	530.685	33.937
Support E	91.755	2.597	29.557	1.890
Reflector	1797.472	50.872	972.658	62.201
Pr Vessel	519.092	14.691	30.825	1.971

Pressure Vessel thickness(cm) 0.40

DETAILS OF THE SHIELDINGS

Neutron Source(1/sec) 0.3808E+19
Self Shielding by Reactor Materials 0.9953E-01
Equivalent R(cm)= 72.00 Sigma Removal 0.103711
2.00 Years Fast Fluence(1/cm2) without Any Shielding 0.7997E+18
" " " " with (n,gamma) " 0.1000E+14
" " " " Requested 0.1000E+14
All at 15.00 meters
Neutron Shielding Thickness(cm) 75.22
Neutron Shielding Thickness if no W is present, cm 89.53

NEUTRON SHIELDING: LiH-Stainless Steel Matrix

TOTAL PRODUCTION OF GAMMA RAYS(1/sec)

GROUP	1	2	3	4	5	6	7
	0-1Mev	1-2	2-3	3-5	5-7	7-9	>9
	2.084E+19	6.404E+18	2.452E+18	9.606E+17	2.708E+17	1.207E+16	3.077E+16

% FRACTION PRODUCED BY U235

98.09	95.67	94.01	63.32	24.66	0.00	79.38

Equivalent Z for Build up factors

34.65	26.86	27.32	31.76	35.89	39.13	41.82

Gamma Self shielding by Core, Reflector and PV

0.101662	0.193773	0.190531	0.212077	0.216133	0.215108	0.213580

The same but without Buildup factor

0.048616	0.095106	0.109257	0.143415	0.159791	0.168326	0.171213

Integrated Dose(Rad) without any shielding(n or gamma)

1.380E+09	1.984E+09	9.781E+08	5.971E+08	2.422E+08	1.352E+07	3.852E+07

Integrated Dose with W and Neutrons Shieldings

3.878E-02	9.709E+04	1.984E+05	4.868E+05	1.967E+05	6.884E+03	1.370E+04

Integrated Dose without Any Shielding,Total(Rad) 0.5233E+10
Integrated dose with Shieldings(W and Neutrons), Total (Rads) 0.10E+07
Tungsten Thickness(cm) 9.12
Tungsten Thickness(cm) 11.65 if no Neutron Shield Present

Requested Dose(Rad)1000000.00 at 15.00 meters

THERMALHYDRAULICS CALCULATIONS
Average Temperature of the Coolant at Core Exit(K) 1698.90

BOUNDARY CONDITION: OUTLET TEMPERATURE

****PROFILE OF CHANNEL WITH MAXIMUM POWER DENSITY****

Problem #1 Nerva Application to NEP

Thermodynamic profile of a He cooled channel
Inlet pressure(MPa)= 2.00000 Inlet temperature(K)= 400.00
Diameter(cm)= 0.28 Rugosity= 0.01000 der parm=1.0100 eps= 0.0010000
Total Power (Mw) 50.000 No. Orifices 37525
Total Flow(Kg/sec) 7.186
Factor that multiplies Power 0.9700000

Average Power Density(Mw/L of active mix) 0.1640 Channel Length(cm) 84.78

Cosine Axial Power Distribution=roav*Pfact*Cos(Pi*z/H)
Pfact(Axial*Radial*Azim)= 2.3949 Extrapolated Delta(cm) 5.00

Attention Implicit Calculation!! xi= 1.00000

Twall/Tbulk correction to Dittus-Boelter Correlation for Nu
Uniform Coolant Flow Assumed

Flow(g/s)= 0.191

Node	z(cm)	Po(w/cm)	Pr(MPa)	T(K)	V(m/s)	Re	Mach
1	0.00	6.02	2.00	400.00	13.06	3600.	0.011
2	1.70	8.04	2.00	411.99	13.46	3528.	0.011
3	3.39	10.02	2.00	427.39	13.97	3440.	0.011
4	5.09	11.98	2.00	446.15	14.59	3340.	0.012
5	6.78	13.90	2.00	468.21	15.31	3231.	0.012
6	8.48	15.77	1.99	493.51	16.15	3116.	0.012

7	10.17	17.60	1.99	521.97	17.08	2998.	0.013
8	11.87	19.37	1.99	553.49	18.13	2880.	0.013
9	13.57	21.07	1.99	587.97	19.27	2763.	0.013
10	15.26	22.72	1.99	625.31	20.50	2648.	0.014
11	16.96	24.29	1.98	665.39	21.84	2538.	0.014
12	18.65	25.78	1.98	708.08	23.26	2432.	0.015
13	20.35	27.19	1.98	753.24	24.77	2331.	0.015
14	22.04	28.52	1.98	800.74	26.36	2235.	0.016
15	23.74	29.75	1.97	850.42	28.04	2144.	0.016
16	25.43	30.89	1.97	902.13	29.79	2059.	0.017
17	27.13	31.94	1.97	955.70	31.58	1979.	0.017
18	28.83	32.88	1.97	1010.96	33.41	1904.	0.018
19	30.52	33.72	1.97	1067.74	35.28	1834.	0.018
20	32.22	34.45	1.97	1125.87	37.20	1768.	0.019
21	33.91	35.08	1.97	1185.14	39.16	1707.	0.019
22	35.61	35.59	1.97	1245.39	41.15	1650.	0.020
23	37.30	35.99	1.97	1306.41	43.16	1596.	0.020
24	39.00	36.28	1.97	1368.02	45.20	1547.	0.021
25	40.70	36.45	1.97	1430.03	47.25	1500.	0.021
26	42.39	36.51	1.97	1492.22	49.31	1457.	0.022
27	44.09	36.45	1.97	1554.42	51.37	1417.	0.022
28	45.78	36.28	1.97	1616.42	53.42	1379.	0.023
29	47.48	35.99	1.97	1678.02	55.46	1344.	0.023
30	49.17	35.59	1.96	1739.04	57.48	1312.	0.023
31	50.87	35.08	1.96	1799.27	59.48	1281.	0.024
32	52.57	34.45	1.96	1858.54	61.45	1253.	0.024
33	54.26	33.72	1.96	1916.65	63.38	1227.	0.025
34	55.96	32.88	1.96	1973.42	65.27	1203.	0.025
35	57.65	31.94	1.96	2028.66	67.11	1180.	0.025
36	59.35	30.89	1.96	2082.21	68.89	1159.	0.026
37	61.04	29.75	1.96	2133.90	70.61	1140.	0.026
38	62.74	28.52	1.96	2183.56	72.27	1122.	0.026
39	64.44	27.19	1.96	2231.04	73.86	1105.	0.027
40	66.13	25.78	1.96	2276.18	75.37	1090.	0.027
41	67.83	24.29	1.96	2318.85	76.80	1076.	0.027
42	69.52	22.72	1.96	2358.91	78.14	1064.	0.027
43	71.22	21.07	1.96	2396.23	79.40	1052.	0.028
44	72.91	19.37	1.96	2430.70	80.56	1042.	0.028
45	74.61	17.60	1.96	2462.20	81.62	1033.	0.028
46	76.30	15.77	1.96	2490.64	82.59	1025.	0.028
47	78.00	13.90	1.96	2515.92	83.45	1018.	0.028
48	79.70	11.98	1.96	2537.98	84.20	1012.	0.028
49	81.39	10.02	1.96	2556.73	84.85	1007.	0.029
50	83.09	8.04	1.96	2572.12	85.38	1002.	0.029
51	84.78	6.02	1.96	2584.10	85.80	999.	0.029

From the Coolant Channel to the Uranium Carbide
Radius(cm) Orificies, Clad, Cell 0.1397 0.1497 0.2280
Averages Densities(g/cm3) Volumetric Fraction

Total	2.8990	
UC	1.0155	0.0783
ZrC	0.4243	0.0633
Graphite	1.4592	0.8584

Number of fuel nodes 3
265411.72 grams of U235 in 530.68 Liters of Fuel Volume

Node	z(cm)	J(Kw/cm2)	Twall(K)	Tclad(K)	TcladAv	Tfuel(K)	TfuelAv
1	0.00	0.00686	467.46	467.78	467.63	468.08	467.99
2	1.70	0.00915	506.76	507.19	506.98	507.62	507.49
3	3.39	0.01142	559.58	560.11	559.85	560.70	560.52
4	5.09	0.01365	613.05	613.69	613.37	614.41	614.20
5	6.78	0.01583	668.55	669.28	668.92	670.17	669.90
6	8.48	0.01797	725.73	726.56	726.15	727.62	727.31
7	10.17	0.02005	783.95	784.89	784.43	786.13	785.76

8	11.87	0.02206	843.70	844.73	844.22	846.17	845.74
9	13.57	0.02401	904.15	905.27	904.72	906.91	906.42
10	15.26	0.02588	965.39	966.60	966.00	968.44	967.89
11	16.96	0.02767	1027.10	1028.38	1027.75	1030.45	1029.83
12	18.65	0.02937	1088.90	1090.26	1089.59	1092.56	1091.88
13	20.35	0.03098	1151.59	1153.03	1152.32	1155.54	1154.79
14	22.04	0.03249	1214.34	1215.85	1215.11	1218.55	1217.74
15	23.74	0.03389	1277.39	1278.97	1278.19	1281.85	1280.99
16	25.43	0.03519	1339.86	1341.50	1340.69	1344.57	1343.65
17	27.13	0.03638	1402.55	1404.24	1403.40	1407.49	1406.52
18	28.83	0.03746	1465.89	1467.63	1466.77	1471.07	1470.05
19	30.52	0.03841	1527.71	1529.50	1528.62	1533.13	1532.05
20	32.22	0.03925	1589.96	1591.79	1590.89	1595.61	1594.47
21	33.91	0.03996	1651.84	1653.70	1652.78	1657.70	1656.51
22	35.61	0.04055	1712.36	1714.24	1713.31	1718.39	1717.15
23	37.30	0.04100	1772.70	1774.61	1773.66	1778.87	1777.60
24	39.00	0.04133	1831.93	1833.85	1832.91	1838.23	1836.92
25	40.70	0.04153	1891.09	1893.02	1892.07	1897.49	1896.16
26	42.39	0.04159	1948.17	1950.11	1949.15	1954.66	1953.30
27	44.09	0.04153	2004.33	2006.26	2005.30	2010.89	2009.51
28	45.78	0.04133	2059.27	2061.19	2060.24	2065.88	2064.48
29	47.48	0.04100	2112.38	2114.28	2113.34	2119.02	2117.60
30	49.17	0.04055	2164.50	2166.39	2165.46	2171.15	2169.73
31	50.87	0.03996	2214.32	2216.18	2215.26	2220.95	2219.53
32	52.57	0.03925	2262.11	2263.94	2263.03	2268.66	2267.25
33	54.26	0.03841	2308.21	2310.00	2309.12	2314.67	2313.27
34	55.96	0.03746	2351.81	2353.55	2352.69	2358.13	2356.77
35	57.65	0.03638	2392.75	2394.44	2393.61	2398.93	2397.59
36	59.35	0.03519	2431.68	2433.31	2432.50	2437.68	2436.38
37	61.04	0.03389	2467.61	2469.18	2468.41	2473.42	2472.16
38	62.74	0.03249	2500.68	2502.20	2501.45	2506.28	2505.06
39	64.44	0.03098	2531.06	2532.50	2531.79	2536.42	2535.25
40	66.13	0.02937	2558.25	2559.62	2558.94	2563.35	2562.24
41	67.83	0.02767	2582.57	2583.86	2583.22	2587.39	2586.34
42	69.52	0.02588	2603.95	2605.16	2604.56	2608.48	2607.49
43	71.22	0.02401	2621.93	2623.04	2622.49	2626.13	2625.21
44	72.91	0.02206	2636.83	2637.86	2637.35	2640.71	2639.86
45	74.61	0.02005	2648.22	2649.15	2648.69	2651.75	2650.97
46	76.30	0.01797	2656.46	2657.29	2656.88	2659.62	2658.93
47	78.00	0.01583	2661.10	2661.83	2661.47	2663.89	2663.27
48	79.70	0.01365	2662.12	2662.75	2662.44	2664.52	2663.99
49	81.39	0.01142	2659.92	2660.45	2660.19	2661.93	2661.49
50	83.09	0.00915	2654.09	2654.51	2654.30	2655.70	2655.34
51	84.78	0.00686	2644.94	2645.26	2645.10	2646.14	2645.88

Axial Average Conditions in the Channel
Coolant Pressure(MPa) 1.970 Temperature(K) 1056.332
Give Density(g/cm3) 0.0008961
Average Density 0.0008960
Wall Temperature(K) 1815.332
Clad Temperature(K) 1816.019
Fuel Temperature(K) 1818.830

Mol Fraction of UC in UC-ZrC mix0.50000 Melting Temperature(K) 3343.50

HEAT BALANCE FOR THE CHANNEL

dQ: heat to coolant (joule/kg) 0.11343E+08
dH: change of enthalpy (joule/kg) 0.11340E+08
dKin change kinetic energy(joule/kg) 0.35955E+04
relative dKin/dQ 0.000317
Balance (dH+dKin)/dQ 1.0000000

PROFILE OF CHANNEL WITH MINIMUM POWER DENSITY

Problem #1 Nerva Application to NEP

Thermodynamic profile of a He cooled channel
Inlet pressure(MPa)= 2.00000 Inlet temperature(K)= 400.00
Diameter(cm)= 0.28 Rugosity= 0.01000 der parm=1.0100 eps= 0.0010000
Total Power (Mw) 50.000 No. Orifices 37525
Total Flow(Kg/sec) 7.186
Factor that multiplies Power 0.9700000

Average Power Density(Mw/L of active mix) 0.1640 Channel Length(cm) 84.78

Cosine Axial Power Distribution=roav*Pfact*Cos(Pi*z/H)
Pfact(Axial*Radial*Azim)= 0.6851 Extrapolated Delta(cm) 5.00

Attention Implicit Calculation!! xi= 1.00000

Twall/Tbulk correction to Dittus-Boelter Correlation for Nu
Uniform Coolant Flow Assumed

Flow(g/s)= 0.191

Node	z(cm)	Po(w/cm)	Pr(MPa)	T(K)	V(m/s)	Re	Mach
1	0.00	1.72	2.00	400.00	13.06	3600.	0.011
2	1.70	2.30	2.00	403.43	13.18	3579.	0.011
3	3.39	2.87	2.00	407.84	13.33	3552.	0.011
4	5.09	3.43	2.00	413.20	13.52	3520.	0.011
5	6.78	3.98	2.00	419.52	13.73	3484.	0.011
6	8.48	4.51	1.99	426.75	13.97	3443.	0.011
7	10.17	5.03	1.99	434.89	14.24	3399.	0.012
8	11.87	5.54	1.99	443.91	14.55	3351.	0.012
9	13.57	6.03	1.99	453.77	14.88	3301.	0.012
10	15.26	6.50	1.99	464.46	15.24	3249.	0.012
11	16.96	6.95	1.99	475.92	15.62	3195.	0.012
12	18.65	7.37	1.99	488.13	16.03	3140.	0.012
13	20.35	7.78	1.98	501.05	16.46	3084.	0.012
14	22.04	8.16	1.98	514.64	16.92	3028.	0.013
15	23.74	8.51	1.98	528.85	17.40	2972.	0.013
16	25.43	8.84	1.98	543.65	17.90	2916.	0.013
17	27.13	9.14	1.98	558.97	18.42	2861.	0.013
18	28.83	9.41	1.98	574.78	18.95	2806.	0.013
19	30.52	9.65	1.97	591.03	19.50	2753.	0.014
20	32.22	9.86	1.97	607.66	20.07	2701.	0.014
21	33.91	10.03	1.97	624.62	20.65	2650.	0.014
22	35.61	10.18	1.97	641.86	21.24	2601.	0.014
23	37.30	10.30	1.97	659.32	21.84	2554.	0.014
24	39.00	10.38	1.96	676.95	22.44	2508.	0.015
25	40.70	10.43	1.96	694.69	23.06	2464.	0.015
26	42.39	10.44	1.96	712.48	23.68	2421.	0.015
27	44.09	10.43	1.96	730.28	24.30	2381.	0.015
28	45.78	10.38	1.95	748.02	24.92	2342.	0.015
29	47.48	10.30	1.95	765.65	25.54	2305.	0.016
30	49.17	10.18	1.95	783.11	26.16	2269.	0.016
31	50.87	10.03	1.94	800.35	26.77	2235.	0.016
32	52.57	9.86	1.94	817.31	27.38	2203.	0.016
33	54.26	9.65	1.94	833.94	27.98	2173.	0.016
34	55.96	9.41	1.94	850.18	28.58	2145.	0.017
35	57.65	9.14	1.93	865.99	29.16	2118.	0.017
36	59.35	8.84	1.93	881.32	29.72	2092.	0.017
37	61.04	8.51	1.92	896.11	30.28	2068.	0.017
38	62.74	8.16	1.92	910.32	30.82	2046.	0.017
39	64.44	7.78	1.92	923.91	31.34	2025.	0.017
40	66.13	7.37	1.91	936.83	31.84	2006.	0.018
41	67.83	6.95	1.91	949.04	32.28	1988.	0.018
42	69.52	6.50	1.91	960.51	32.67	1972.	0.018
43	71.22	6.03	1.91	971.19	33.04	1957.	0.018
44	72.91	5.54	1.91	981.05	33.38	1944.	0.018

45	74.61	5.03	1.91	990.07	33.68	1931.	0.018
46	76.30	4.51	1.91	998.21	33.96	1921.	0.018
47	78.00	3.98	1.91	1005.44	34.21	1911.	0.018
48	79.70	3.43	1.91	1011.75	34.42	1903.	0.018
49	81.39	2.87	1.91	1017.12	34.61	1896.	0.018
50	83.09	2.30	1.91	1021.53	34.76	1890.	0.018
51	84.78	1.72	1.91	1024.95	34.88	1886.	0.018

From the Coolant Channel to the Uranium Carbide
Radius(cm) Orifices, Clad, Cell 0.1397 0.1497 0.2280
Averages Densities(g/cm3) Volumetric Fraction

Total	2.8990	
UC	1.0155	0.0783
ZrC	0.4243	0.0633
Graphite	1.4592	0.8584

Number of fuel nodes 3
265411.72 grams of U235 in 530.68 Liters of Fuel Volume

Node	z(cm)	J(Kw/cm2)	Twall(K)	Tclad(K)	TcladAv	Tfuel(K)	TfuelAv
1	0.00	0.00196	418.43	418.52	418.47	418.60	418.58
2	1.70	0.00262	428.86	428.98	428.92	429.09	429.06
3	3.39	0.00327	442.72	442.87	442.79	443.01	442.97
4	5.09	0.00390	456.75	456.93	456.84	457.10	457.05
5	6.78	0.00453	471.36	471.57	471.47	471.77	471.71
6	8.48	0.00514	486.64	486.88	486.76	487.11	487.04
7	10.17	0.00573	502.58	502.85	502.72	503.12	503.04
8	11.87	0.00631	519.20	519.49	519.35	519.79	519.70
9	13.57	0.00687	536.39	536.71	536.55	537.05	536.95
10	15.26	0.00740	553.97	554.32	554.15	554.69	554.58
11	16.96	0.00791	572.06	572.43	572.25	572.83	572.71
12	18.65	0.00840	590.41	590.81	590.61	591.24	591.11
13	20.35	0.00886	608.97	609.39	609.18	609.86	609.72
14	22.04	0.00929	627.86	628.29	628.08	628.79	628.64
15	23.74	0.00970	647.02	647.47	647.25	648.00	647.84
16	25.43	0.01007	666.19	666.66	666.43	667.22	667.06
17	27.13	0.01041	685.31	685.79	685.55	686.38	686.21
18	28.83	0.01072	704.51	705.01	704.77	705.63	705.45
19	30.52	0.01099	723.52	724.04	723.78	724.68	724.49
20	32.22	0.01123	742.50	743.02	742.76	743.69	743.49
21	33.91	0.01143	761.38	761.92	761.65	762.61	762.40
22	35.61	0.01160	779.88	780.42	780.15	781.14	780.92
23	37.30	0.01173	798.41	798.96	798.69	799.70	799.48
24	39.00	0.01182	816.44	816.99	816.72	817.75	817.52
25	40.70	0.01188	833.88	834.44	834.16	835.21	834.98
26	42.39	0.01190	851.21	851.76	851.49	852.55	852.31
27	44.09	0.01188	867.84	868.40	868.12	869.19	868.95
28	45.78	0.01182	884.27	884.82	884.54	885.61	885.38
29	47.48	0.01173	900.17	900.71	900.44	901.51	901.27
30	49.17	0.01160	915.21	915.75	915.48	916.55	916.31
31	50.87	0.01143	930.79	931.33	931.06	932.12	931.88
32	52.57	0.01123	944.52	945.04	944.78	945.83	945.59
33	54.26	0.01099	957.87	958.38	958.13	959.16	958.93
34	55.96	0.01072	970.33	970.83	970.58	971.59	971.36
35	57.65	0.01041	981.99	982.48	982.24	983.23	983.01
36	59.35	0.01007	992.99	993.46	993.23	994.19	993.97
37	61.04	0.00970	1003.13	1003.58	1003.36	1004.29	1004.08
38	62.74	0.00929	1012.38	1012.81	1012.59	1013.49	1013.29
39	64.44	0.00886	1020.71	1021.12	1020.92	1021.78	1021.58
40	66.13	0.00840	1028.09	1028.48	1028.29	1029.11	1028.92
41	67.83	0.00791	1034.51	1034.88	1034.69	1035.47	1035.29
42	69.52	0.00740	1040.09	1040.44	1040.27	1041.00	1040.83
43	71.22	0.00687	1044.67	1044.99	1044.83	1045.51	1045.36
44	72.91	0.00631	1048.23	1048.52	1048.38	1049.00	1048.86
45	74.61	0.00573	1050.91	1051.17	1051.04	1051.61	1051.48
46	76.30	0.00514	1052.46	1052.70	1052.58	1053.09	1052.97

47	78.00	0.00453	1053.04	1053.25	1053.15	1053.60	1053.49
48	79.70	0.00390	1052.65	1052.83	1052.74	1053.13	1053.04
49	81.39	0.00327	1051.20	1051.35	1051.27	1051.60	1051.52
50	83.09	0.00262	1048.73	1048.85	1048.79	1049.05	1048.99
51	84.78	0.00196	1045.25	1045.34	1045.29	1045.49	1045.44

Axial Average Conditions in the Channel
Coolant Pressure(MPa) 1.954 Temperature(K) 643.987
Give Density(g/cm3) 0.0014554
Average Density 0.0014553
Wall Temperature(K) 808.533
Clad Temperature(K) 808.729
Fuel Temperature(K) 809.296

Mol Fraction of UC in UC-ZrC mix0.50000 Melting Temperature(K) 3343.50

HEAT BALANCE FOR THE CHANNEL

dQ: heat to coolant (joule/kg) 0.32449E+07
dH: change of enthalpy (joule/kg) 0.32444E+07
dKin change kinetic energy(joule/kg) 0.52293E+03
relative dKin/dQ 0.000161
Balance (dH+dKin)/dQ 1.0000002

8.3 INPUT TO NEPPIN

itest/0,1/: Do not/run test, when itest = 1, the code prints the energy available for the set of input conditions.
ndt: Number of points, in the tabulation of the available energy as function of R_c or H_c, between the asymptotic and optimum values of R_c or H_c.
itmas: Maximum number of iterations.
icont/0,1/ Do not/continue calculation after test.
iboun/1,2/: Input coolant flow/outlet coolant temperature.
nput: Total of points in the tabulation of the available energy.
Pow(Mw): Fission power.
D(years): Time during which the reactor is on.
BU(%): Burnup at end of life, see Eq. (3).
eps(0.001): Iterations are stopped within this relative change.
der(1.01): Relative increment to compute derivatives.
drr(cm): Thickness of the radial Be reflector.
dra1(cm): Thickness of the top axial Be reflector.
dra2(cm): Thickness of the bottom axial Be reflector.
tduc(mm): Thickness of the duct of the bundle.
dkstr(%,2): Reactivity of structural components.
dkeol(%,2): Reactivity at end of life.
dksu(%,2): Subcritical reactivity with drums in.
pin(MPa),tin(°K): Pressure and temperature at the inlet of reactor.
pHe(MPa),THe(K): Pressure and temperature of the He in the gap of the fuel rod.
tempv(°K): Temperature of the pressure vessel, if input < 0 program chooses toutl as the value of this variable.
flowt(Kg/s): Total coolant flow if iboun = 1.
toutl(°K): Outlet temperature of coolant if iboun = 2.

User might stop input here for a test only run.

ncr: Number of drums.
nnodo: Number of axial nodes for thermalhydraulic calculations (≤ 50).
npin: Number of radial nodes for temperature calculations in fuel pin.
ishape/1,2,3/: Shape of the core.
- 1: Ideal shape ($R_c/H_c = 0.54$).
- 2: Input R_c
- 3: Input H_c

iopth/1,2,3,4/: Assumptions about the power and flow conditions.
- 1: 3D uniform power distributions.
- 2: 2D (r,θ) uniform power distribution, cosine axial distribution (1 and 2 with uniform flow distribution.
- 3: r,θ power distribution corresponding to uniform fuel load, cosine axial distribution and uniform flow distribution.
- 4: Like 3, but with (r,θ) flow distribution proportional to power distribution.

ifri/0,1/: No/Yes Inclusion of friction factors in pressure calculations.
iprth/0,1/: No/Yes Print details of thermalhydraulic calculations.
iouth/1,2,3/: Printing of thermalhydraulic condition of
- 1: Maximum power density channel.
- 2: Minimum power density channel.
- 3: Both.

iwrte/0,1/ No/Yes Printing of iterations.

ishn/1,2/ Selection of neutron shielding.

1: LiH in stainless steel matrix.

2: B_4C

ichoose/0,1/: For the case of contradictions between demand and availability:

0: Program chooses alternative demands.

1: Program stops for new demands from user (after printing some advices).

Ichoode/0,1...5/: Selection of a design

5 designs are calculated for P/d = 1.1, 1.2, 1.3, 1.4, and 1.5. If ichoode = 0 the code selects a design based on the Δp along the channel and the total mass; the user can override the selection process by choosing one of the five.

fact(0.97): Factor for the conversion of fission power to thermal power.

ripple(0.1): Maximum ripple of the power distribution in the azimuthal direction (see Eq. (21)).

hod: The lead of the helicoidal wire spacing divided the diameter of the fuel rod.

dcd(cm): Thickness of the B_4C control sheet.

delpr(Pa): Maximum pressure drop for the coolant, this value is used to select a design.

fastf($1/cm^2$): Maximum fast fluence at cargo bay.

dose(rad): Maximum gamma dose at cargo bay.

zload(m): Distance of cargo bay to base of reactor.

rload(m): Radius of cargo bay.

R(cm): Radius of core if ishape = 2.

H(cm): Length of core if ishape = 3.

The input file for the sample case follows:

```
Problem #1 Fuel Pin Application to NEP

itest/# dR/itma/cont?/ibound/# of points in test/ Format I5
    0   10  450    1    2   31
Pow(Mw)  /Life(y)  /Burnup(%)/Tolerance/Rel delta/  E10.0
50.       4.0       10.       0.001     1.010
Radial   /Top      /Bottom Reflector Thickness(cm)/duct thick(mm)
30.       0.        0.        1.0
Struct.  /End Life /Subcrit/    Reactivities(%)
2.        2.        2.
Pin(MPa) /Tin(K)   /PHe      /THe(Inlet and He gap Conditions)/press vess temp
2.        500.      0.1       300.      800.
iboun=1 input total coolant flow(Kg/s),iboun=2 input outlet T(K)
1200.
#drum/#axi/#rad/ishape:1,2,3(ideal,choose R,choose H)/# rods per Bundle
   12   30    3    1   19
iopth/1:3D Unif Power;2:2D Unif Power;3:Unif Load and Flow;4:Unif Load Flow~Pow
    3
ifri/iprth/iouth/friction Y/N;Print Details Y/N;Print Max,Min,Both?
    1    1    3
iwrte/Nshield/Ichoose/Ichoode;wrte its,LiH-SS/B4C, Ichoose or Program?/ design?
    0    1    0    0
Pow Fract/Ripple Angular Power/wirelead/d/B4C thick(cm) in drums
0.97      0.1       10.       2.
Max delta p acceptable(Pa)
51700.
Max fast n/Dose(RAD)/z(m)    /r(m) Shielding at z,r for input dose
1.    e+13 1000000. 15.0      5.
ishape(1,2,3) (skip,input R,input H(cm))
75.
```

8.4 OUTPUT FROM NEPPIN

The output from running the code with the input file of the previous section follow:

```
 ATTENTION
 Sampling p/d=  1.5000
 You requested Mwdays  73050.00
 And burn-up(%)10.00
 With radial Be reflector thickness(cm) 30.00
  "    top      "      "       "          0.00
  "    bottom   "      "       "          0.00
             End of Life delta k(%)        2.00
             Structural  delta k(%)        2.00

 Trying to satisfy reactivity balance,it happens your core
 is too large(and it would produce more energy you need).
 You might:
 1) Reduce the input burn-up, and/or
 2) Increase the reflector thickness,and/or
 3) Reduce delta k at end of life, and/or
 4) Reduce delta k structural


 Program will change burn-up for this p/d design
 and choose ishape=1

          BU        Mwday

       10.00     77233.00
        9.00     66788.17


Sampling of Designs
Mass=Masses of core,pressure vessel and reflector
 Max delta p(Pa) acceptable=   0.5170E+05

  I   p/d   Enr    BU  Mass(Kg)    U5(Kg)   DeltaP(Pa) coreR(cm) coreH(cm) ishape

  1 1.100 0.530 10.00    6424.6     769.0   0.1003E+04     39.96     73.27  1
  2 1.200 0.615 10.00    6064.9     768.3   0.1199E+04     40.56     74.26  1
  3 1.300 0.715 10.00    5750.4     768.2   0.1423E+04     40.90     75.01  1
  4 1.400 0.829 10.00    5476.4     769.4   0.5433E+04     41.12     75.37  1
  5 1.500 0.894  9.00    6010.7     854.8   0.6080E+04     43.70     79.94  1
Problem #1 Fuel Pin Application to NEP


 INPUT TO THE PROGRAM

 Power(Mw)   50.00 Core life   4.000 years
 Energy Released(Mwday)  73050.00 Burn-up(%) 10.00

 Be reflector, radial thickness(cm) 30.00
               top          "        0.00
               bottom       "        0.00

 The following design can produce that energy within the relative
numerical precision of  0.1000E-02

 Minimum Core Volume Option Chosen: R/H= 0.541315

 And it has the following reactivity balance
 Delta k required(%) 13.99 with components
  Fission Products(%)  8.23
   Pressure Vessel(%)  1.76
        Structural(%)  2.00
```

```
      End of Life(%)  2.00
 keff at BOL(No structural mat) 1.1399 Components
                   Bare Reactor 0.9941
                      Reflector 0.1458


 Enrichment=  0.8292 Pitch/Diameter=  1.40

 From Control
 Radius Drums(cm)   15.00 Central Parallax(degree)   30.78
 Thickness B4C sheet on Control Drums,cm    2.00


 You cannot control this reactor only with control
 drums in the reflector
 The reactivity worth of   11 drums(max value) is  6.17 %
                 and the reactivity to control is 13.99 %
Absorbing angle of each drum would be  149.22 degrees


 NUMBER DENSITIES AND MASSES OF FUEL PIN REACTOR
         GENERAL DESCRIPTION

Core Volume(L)     400.42 Active(UN) Volume(L)      80.94
 Core Radius(cm)      41.12 Height(cm)      75.37
 Number of rods  5871 Bundles   309( 19rods per bundle)


 Neutron Flux(1/cm2sec)(core and time average)          0.5748E+15
    "    Fluence(1/cm2)  "       "   after  4.00 years  0.7256E+23

 Masses(Kg) of Fissile Materials and Fission Products

                  BOL       EOL
     U-235    769.399   681.867
     U-238    160.488   157.337
    Pu-239      0.000     1.731     1.868 Kg produced
Fiss Prod       0.000    77.274
            FUEL RODS
 Rod diameter(mm) is 6.4000
 Astar Alloy clad thickness,mm 0.6350
 W Liner,mm 0.1270
 He gap,mm 0.0250
 UN pin diameter,mm 4.8260


 Volume fractions in Fuel Cell #1
      Coolant 0.5126
 Spacing Wire 0.0247
 Fuel Rod     0.4627


 Fuel Cell type 1 (Number Densities 10**24 atoms/cm3)
 Spacing Wire is diluted in clad

Region   Radius(cm) Isotope Concentration Volumetric Fraction

Coolant       0.4704     Li6  0.00307001  0.512619
  "                      Li7  0.03830476
Clad          0.3200   Ta181  0.05059388  0.176834
  "           0.3284(diluting spacing wire)
  "                    W182  0.00117159
  "                    W183  0.00063544
  "                    W184  0.00135207
  "                    W186  0.00125367
  "                   Re185  0.00020190
  "                   Re187  0.00034274
```

"		Hf	0.00041192	
Liner	0.2565	W182	0.01617544	0.028710
"		W183	0.00877312	
"		W184	0.01866725	
"		W186	0.01730863	
He Gap	0.2438	He	0.00002413	0.005480
Fuel Pin	0.2413	U235	0.02435551	0.263098
" "		U238	0.00501613	
" "		N14	0.02937164	

AVERAGES IN BUNDLE(INCLUDING DUCT)

Volumetric Fractions:

Rods	0.3555
Spacing Wire	0.0210
Li Coolant	0.5358
Duct	0.0878

Isotope Concentration
(Number Densities 10**24 atoms/cm3)

Li6	0.00164494
Li7	0.02052410
Ta181	0.01192968
W182	0.00063304
W183	0.00034334
W184	0.00073055
W186	0.00067738
Re185	0.00004761
Re186	0.00008082
Hf	0.00009713
He	0.00000010
U235	0.00492296
U238	0.00101391
N14	0.00593687

Moderation by Li, Li/U235= 4.5032

VOLUME(L) AND MASSES(KG) INSIDE THE CORE

Part	Volume	Fraction	Masses	Fraction
Rods	142.34	0.355479	2006.38	0.706037
Coolant	214.55	0.535811	102.33	0.036009
Duct	35.14	0.087755	591.74	0.208232
Spacing	8.39	0.020955	141.30	0.049723
Total	400.42		2841.75	

VOLUME(L) AND MASSES(KG) OF REACTOR

Part	Volume	Fraction	Masses	Fraction
Core	400.42	0.327118	2841.75	0.517700
Press Ves	12.16	0.009935	204.79	0.037308
Reflector	811.51	0.662947	2442.64	0.444992
Total	1224.09		5489.18	

VOLUMES AND MASSES OF REACTOR AND SHIELDINGS

	Volume(L)	Fraction(%)
Reactor	1224.09	38.60
N Shield	1691.15	53.33
G Shield	255.87	8.07
Total	3171.11	

	Mass(Kg)	Fraction(%)
Reactor	5489.18	45.82
N Shield	1731.73	14.46
G Shield	4759.17	39.73
Total	11980.08	

DIMENSION OF THE SHIELDINGS

LOAD AT Z(M)= 15.00 WITH RADIUS(M)= 5.00

G Shield Thickness(cm) 9.21 With Radius(cm) 92.76 95.32

N " " " 79.60 " " " 95.32 117.51

EVALUATION OF PRESSURE VESSEL

Evaluation at T(K)= 800.00 PRESS(MPa)= 2.00

and Full Power Life(y) 4.00

Pressure Vessel Thickness(cm) 0.40

DETAILS OF THE SHIELDINGS

Neutron Source(1/sec) 0.3808E+19

Self Shielding by Reactor Materials 0.8751E-01

Equivalent R(cm)= 66.36 Sigma Removal 0.128256

4.00 Years Fast Fluence(1/cm2) without Any Shielding 0.1415E+19

" " " " with (n,gamma) " 0.1000E+14

" " " " Requested 0.1000E+14

All at 15.00 meters

Neutron Shielding Thickness(cm) 79.60

Neutron Shielding Thickness if no W is present, cm 94.05

NEUTRON SHIELDING: LiH-Stainless Steel Matrix

TOTAL PRODUCTION OF GAMMA RAYS(1/sec)

GROUP	1	2	3	4	5	6	7
	0-1Mev	1-2	2-3	3-5	5-7	7-9	>9
	2.132E+19	7.140E+18	3.132E+18	1.474E+18	1.815E+17	1.493E+15	1.097E+16

% FRACTION PRODUCED BY U

89.14 75.35 62.24 32.78 35.62 0.00 99.51

Equivalent Z for Build up factors

54.97 45.51 46.08 51.17 55.29 58.22 60.50

Gamma Self shielding by Core, Reflector and PV

0.040546 0.098502 0.095939 0.101189 0.097540 0.093919 0.089193

The same but without Buildup factor

0.022162 0.050059 0.057265 0.070490 0.074171 0.074878 0.073508

Integrated Dose(Rad) without any shielding(n or gamma)

1.133E+09 2.262E+09 1.266E+09 8.799E+08 1.474E+08 1.471E+06 1.154E+07

Integrated Dose with W and Neutrons Shieldings

1.837E-02 8.058E+04 2.003E+05 6.097E+05 1.045E+05 6.534E+02 3.629E+03

Integrated Dose without Any Shielding,Total(Rad) 0.5701E+10

Integrated dose with Shieldings(W and Neutrons), Total (Rads) 0.10E+07

Tungsten Thickness(cm) 9.21

Tungsten Thickness(cm) 11.95 if no Neutron Shield Present

Requested Dose(Rad)1000000.00 at 15.00 meters

THERMALHYDRAULICS CALCULATIONS
Average Temperature of the Coolant at Core Exit(K) 1200.00

BOUNDARY CONDITION: OUTLET TEMPERATURE

****PROFILE OF CHANNEL WITH MAXIMUM POWER DENSITY****

Problem #1 Fuel Pin Application to NEP

Thermodynamic profile of a liquid Li cooled channel
****
* Remember Li melts at 453.70 K !!!! *
****
Inlet temperature(K)= 500.00
Diameter Rod(cm)= 0.64
Total Power(Mw) 50.000 No. Rods 5871 No.Bundles 309(19rods/bundle)
Total Flow(Kg/sec) 16.139
Active Volume(L of UN) 80.937
Factor that multiplies Power 0.9700000

Average Power Density(Mw/L(UN)) 0.5992 Channel Length(cm) 75.37

FLOW CONDITIONS:

Pitch/drod 1.400 Lead wire spacer/drod 10.000

Cell Type (per bundle)	1	2	3	Bundle
Number of Cells	24	12	6	
Flow(g/s) Total	24.2455	23.2914	4.6917	52.2286
Flow Area (*)	0.5126	0.9849	0.3968	0.5874
Flow Area(cm2),per cell	0.1782	0.3424	0.1379	9.2130
Flow Area(cm2),per rod	0.3564	0.6848	0.8276	0.4849
Wet Perimeter(cm) Total	1.1394	2.1024	1.0002	58.5741
Effective Diameter(cm)	0.6256	0.6514	0.5516	0.6291

(*) 1,2,3 in units of cell 1 area

Parameters of the bundle for friction correlations: X= 0.995 De1/DT= 0.994

Boundary Condition: Fix outlet flow temperature 1200.00
Cosine Axial Power Distribution=roav*Pfact*Cos(Pi*z/H)
Pfact(Axial*Radial*Azim)= 2.3752 Extrapolated Delta(cm) 5.00
Uniform Coolant Flow Assumed

AVERAGE CONDITIONS IN THE BUNDLE
****
********** Inlet Press(MPa) 2.000000 **********
****

Node	z(cm)	Po(w/cm)	Pr(KPa)	T(K)	(Tsat)	V(m/s)	Re
1	0.00	47.64	2000.00	500.00(	2129.52)	0.11	274.
2	2.51	71.06	2000.00	511.82(	2129.52)	0.11	296.
3	5.02	93.88	1999.99	528.40(	2129.52)	0.11	328.
4	7.54	115.90	1999.99	549.71(	2129.52)	0.11	371.
5	10.05	136.93	1999.99	575.68(	2129.52)	0.11	426.
6	12.56	156.79	1999.98	606.15(	2129.51)	0.11	493.

7	15.07	175.31	1999.98	640.90(2129.51)	0.11	572.
8	17.59	192.33	1999.98	679.65(2129.51)	0.11	664.
9	20.10	207.71	1999.97	722.07(2129.51)	0.12	767.
10	22.61	221.31	1999.97	767.79(2129.51)	0.12	880.
11	25.12	233.03	1999.97	816.40(2129.51)	0.12	1002.
12	27.63	242.75	1999.97	867.47(2129.51)	0.12	1130.
13	30.15	250.40	1999.96	920.54(2129.51)	0.12	1263.
14	32.66	255.91	1999.96	975.13(2129.51)	0.12	1399.
15	35.17	259.24	1999.95	1030.77(2129.51)	0.12	1534.
16	37.68	260.35	1999.95	1086.96(2129.51)	0.13	1668.
17	40.19	259.24	1999.95	1143.21(2129.51)	0.13	1799.
18	42.71	255.91	1999.94	1199.03(2129.51)	0.13	1926.
19	45.22	250.40	1999.94	1253.93(2129.51)	0.13	2048.
20	47.73	242.75	1999.94	1307.44(2129.51)	0.13	2162.
21	50.24	233.03	1999.93	1359.09(2129.51)	0.13	2270.
22	52.76	221.31	1999.93	1408.42(2129.51)	0.13	2370.
23	55.27	207.71	1999.93	1455.03(2129.51)	0.14	2462.
24	57.78	192.33	1999.92	1498.50(2129.51)	0.14	2545.
25	60.29	175.31	1999.92	1538.46(2129.51)	0.14	2620.
26	62.80	156.79	1999.91	1574.56(2129.51)	0.14	2686.
27	65.32	136.93	1999.91	1606.49(2129.51)	0.14	2743.
28	67.83	115.90	1999.91	1633.99(2129.51)	0.14	2792.
29	70.34	93.88	1999.90	1656.80(2129.51)	0.14	2832.
30	72.85	71.06	1999.90	1674.74(2129.50)	0.14	2862.
31	75.37	47.64	1999.90	1687.65(2129.50)	0.14	2884.

Thermodynamics Conditions of Cells

Cell Type	z Node	Max(T-<T>)	T Saturation	P(kPa)
1	31	434.21	2121.87	1937.51
2	31	-351.02	1336.63	10.78
3	31	-497.75	1189.91	1.91

Delta pressure(KPa) because T1>T(average) 1738.05

<c> in f=c/Re**0.25 is 0.25944

From the Coolant Channel to fuel pin temperature

Rod Diameter(mm): 6.4000
Clad+Liner Thickness("): 0.7620
He gap " ("): 0.0250
Fuel Pin Diameter ("): 4.8260

CONDITIONS FOR MAX HEATED ROD OF THE BUNDLE(Type 1 cell)

Node	z(cm)	J(Kw/cm2)	Twall(K)	Tclad(K)	TcladAv	Tfuel(K)	TfuelAv
1	0.00	0.02369	502.51	506.13	504.15	549.07	541.17
2	2.51	0.03534	519.82	525.22	522.27	588.24	576.46
3	5.02	0.04669	543.62	550.75	546.87	632.31	616.75
4	7.54	0.05764	573.89	582.69	577.89	680.96	661.75
5	10.05	0.06810	610.48	620.88	615.22	733.91	711.21
6	12.56	0.07798	653.17	665.08	658.59	790.85	764.86
7	15.07	0.08719	701.62	714.94	707.68	851.49	822.42
8	17.59	0.09566	755.42	770.03	762.06	915.47	883.59
9	20.10	0.10330	814.09	829.86	821.27	982.43	948.00
10	22.61	0.11007	877.11	893.92	884.76	1051.95	1015.26
11	25.12	0.11590	943.92	961.62	951.97	1123.57	1084.94
12	27.63	0.12073	1013.93	1032.36	1022.32	1196.79	1156.55
13	30.15	0.12454	1086.52	1105.54	1095.17	1271.09	1229.57
14	32.66	0.12728	1161.06	1180.49	1169.90	1345.90	1303.47
15	35.17	0.12893	1236.89	1256.58	1245.85	1420.64	1377.66
16	37.68	0.12949	1313.37	1333.14	1322.36	1494.71	1451.55
17	40.19	0.12893	1389.82	1409.51	1398.78	1567.53	1524.55
18	42.71	0.12728	1465.60	1485.03	1474.44	1638.47	1596.04

19	45.22	0.12454	1540.05	1559.06	1548.70	1706.94	1665.43
20	47.73	0.12073	1612.52	1630.95	1620.91	1772.36	1732.12
21	50.24	0.11590	1682.40	1700.10	1690.45	1834.16	1795.53
22	52.76	0.11007	1749.08	1765.89	1756.73	1891.80	1855.11
23	55.27	0.10330	1812.00	1827.77	1819.17	1944.76	1910.33
24	57.78	0.09566	1870.60	1885.21	1877.24	1992.58	1960.69
25	60.29	0.08719	1924.39	1937.70	1930.44	2034.81	2005.75
26	62.80	0.07798	1972.89	1984.80	1978.31	2071.07	2045.08
27	65.32	0.06810	2015.70	2026.10	2020.43	2101.02	2078.32
28	67.83	0.05764	2052.44	2061.25	2056.45	2124.36	2105.14
29	70.34	0.04669	2082.80	2089.93	2086.04	2140.86	2125.29
30	72.85	0.03534	2106.50	2111.90	2108.96	2150.34	2138.56
31	75.37	0.02369	2123.35	2126.97	2124.99	2152.68	2144.79

Axial Average Conditions in the Channel
Coolant Pressure(MPa) 2.000 Temperature(K) 1089.432
Wall Temperature(K) 1313.155 Heat Transfer(w/cm2K) 12.70
Clad Temperature(K) 1319.526
Fuel Temperature(K) 1416.165

HEAT BALANCE FOR THE BUNDLE

dQ: heat to coolant (joule/kg) 0.50566E+07
dH: change of enthalpy (joule/kg) 0.50566E+07
dKin change kinetic energy(joule/kg) 0.43164E-02
relative dKin/dQ 0.000000
Balance (dH+dKin)/dQ 0.9999961

PROFILE OF CHANNEL WITH MINIMUM POWER DENSITY

Problem #1 Fuel Pin Application to NEP

Thermodynamic profile of a liquid Li cooled channel
****
* Remember Li melts at 453.70 K !!!! *
****
Inlet temperature(K)= 500.00
Diameter Rod(cm)= 0.64
Total Power(Mw) 50.000 No. Rods 5871 No.Bundles 309(19rods/bundle)
Total Flow(Kg/sec) 16.139
Active Volume(L of UN) 80.937
Factor that multiplies Power 0.9700000

Average Power Density(Mw/L(UN)) 0.5992 Channel Length(cm) 75.37

FLOW CONDITIONS:

Pitch/drod 1.400 Lead wire spacer/drod 10.000

Cell Type (per bundle)	1	2	3	Bundle
Number of Cells	24	12	6	
Flow(g/s) Total	24.2455	23.2914	4.6917	52.2286
Flow Area (*)	0.5126	0.9849	0.3968	0.5874
Flow Area(cm2),per cell	0.1782	0.3424	0.1379	9.2130
Flow Area(cm2),per rod	0.3564	0.6848	0.8276	0.4849
Wet Perimeter(cm) Total	1.1394	2.1024	1.0002	58.5741
Effective Diameter(cm)	0.6256	0.6514	0.5516	0.6291

(*) 1,2,3 in units of cell 1 area

Parameters of the bundle for friction correlations: X= 0.995 Del/DT= 0.994

Boundary Condition: Fix outlet flow temperature 2152.68
Cosine Axial Power Distribution=roav*Pfact*Cos(Pi*z/H)
Pfact(Axial*Radial*Azim)= 0.5662 Extrapolated Delta(cm) 5.00
Uniform Coolant Flow Assumed

AVERAGE CONDITIONS IN THE BUNDLE

********** Inlet Press(MPa) 2.000000 **********

Node	z(cm)	Po(w/cm)	Pr(KPa)	T(K)	(Tsat)	V(m/s)	Re
1	0.00	11.36	2000.00	500.00(	2129.52)	0.11	274.
2	2.51	16.94	2000.00	502.81(	2129.52)	0.11	279.
3	5.02	22.38	1999.99	506.72(	2129.52)	0.11	287.
4	7.54	27.63	1999.99	511.71(	2129.52)	0.11	296.
5	10.05	32.64	1999.99	517.75(	2129.52)	0.11	308.
6	12.56	37.38	1999.98	524.79(	2129.51)	0.11	321.
7	15.07	41.79	1999.98	532.79(	2129.51)	0.11	337.
8	17.59	45.85	1999.97	541.69(	2129.51)	0.11	355.
9	20.10	49.51	1999.97	551.42(	2129.51)	0.11	375.
10	22.61	52.76	1999.97	561.89(	2129.51)	0.11	396.
11	25.12	55.55	1999.96	573.04(	2129.51)	0.11	420.
12	27.63	57.87	1999.96	584.76(	2129.51)	0.11	445.
13	30.15	59.69	1999.96	596.96(	2129.51)	0.11	472.
14	32.66	61.01	1999.96	609.52(	2129.51)	0.11	500.
15	35.17	61.80	1999.95	622.35(	2129.51)	0.11	529.
16	37.68	62.06	1999.95	635.33(	2129.51)	0.11	559.
17	40.19	61.80	1999.95	648.34(	2129.51)	0.11	589.
18	42.71	61.01	1999.94	661.27(	2129.51)	0.11	620.
19	45.22	59.69	1999.94	674.01(	2129.51)	0.11	650.
20	47.73	57.87	1999.94	686.44(	2129.51)	0.11	680.
21	50.24	55.55	1999.94	698.46(	2129.51)	0.12	709.
22	52.76	52.76	1999.93	709.95(	2129.51)	0.12	737.
23	55.27	49.51	1999.93	720.82(	2129.51)	0.12	764.
24	57.78	45.85	1999.93	730.96(	2129.51)	0.12	789.
25	60.29	41.79	1999.92	740.29(	2129.51)	0.12	812.
26	62.80	37.38	1999.92	748.73(	2129.51)	0.12	833.
27	65.32	32.64	1999.92	756.20(	2129.51)	0.12	851.
28	67.83	27.63	1999.92	762.63(	2129.51)	0.12	867.
29	70.34	22.38	1999.91	767.97(	2129.51)	0.12	881.
30	72.85	16.94	1999.91	772.17(	2129.51)	0.12	891.
31	75.37	11.36	1999.91	775.19(	2129.51)	0.12	899.

Thermodynamics Conditions of Cells

Cell Type	z Node	Max(T-<T>)	T Saturation	P(kPa)
1	31	101.98	877.17	0.01
2	31	-81.96	693.23	0.00
3	31	-116.03	659.16	0.00

Delta pressure(KPa) because T1>T(average) 0.01

<c> in f=c/Re**0.25 is 0.19946

From the Coolant Channel to fuel pin temperature
Rod Diameter(mm): 6.4000
Clad+Liner Thickness("): 0.7620
He gap " ("): 0.0250
Fuel Pin Diameter ("): 4.8260

CONDITIONS FOR MAX HEATED ROD OF THE BUNDLE(Type 1 cell)

Node	z(cm)	J(Kw/cm2)	Twall(K)	Tclad(K)	TcladAv	Tfuel(K)	TfuelAv
1	0.00	0.00565	500.60	501.46	500.99	511.74	509.86
2	2.51	0.00843	504.71	506.00	505.30	521.27	518.46
3	5.02	0.01113	510.32	512.02	511.10	532.10	528.39
4	7.54	0.01374	517.40	519.50	518.35	544.12	539.54
5	10.05	0.01623	525.90	528.38	527.03	557.25	551.84
6	12.56	0.01859	535.77	538.61	537.06	571.39	565.19
7	15.07	0.02079	546.93	550.11	548.38	586.43	579.50
8	17.59	0.02280	559.32	562.80	560.90	602.26	594.66
9	20.10	0.02463	572.82	576.58	574.53	618.76	610.55
10	22.61	0.02624	587.34	591.35	589.17	635.80	627.06
11	25.12	0.02763	602.76	606.98	604.68	653.27	644.06
12	27.63	0.02878	618.94	623.34	620.94	671.03	661.43
13	30.15	0.02969	635.76	640.29	637.82	688.93	679.04
14	32.66	0.03034	653.05	657.68	655.16	706.85	696.73
15	35.17	0.03074	670.68	675.37	672.81	724.63	714.39
16	37.68	0.03087	688.48	693.19	690.62	742.14	731.85
17	40.19	0.03074	706.30	710.99	708.44	759.24	748.99
18	42.71	0.03034	723.99	728.62	726.09	775.78	765.67
19	45.22	0.02969	741.38	745.91	743.44	791.63	781.73
20	47.73	0.02878	758.32	762.72	760.32	806.65	797.05
21	50.24	0.02763	774.67	778.89	776.59	820.71	811.50
22	52.76	0.02624	790.28	794.28	792.10	833.70	824.95
23	55.27	0.02463	805.00	808.76	806.71	845.50	837.29
24	57.78	0.02280	818.72	822.20	820.30	856.00	848.40
25	60.29	0.02079	831.31	834.48	832.75	865.11	858.18
26	62.80	0.01859	842.65	845.49	843.94	872.75	866.55
27	65.32	0.01623	852.66	855.13	853.78	878.84	873.43
28	67.83	0.01374	861.23	863.33	862.18	883.32	878.74
29	70.34	0.01113	868.30	870.00	869.07	886.14	882.43
30	72.85	0.00843	873.80	875.09	874.39	887.28	884.47
31	75.37	0.00565	877.68	878.55	878.08	886.71	884.83

Axial Average Conditions in the Channel
Coolant Pressure(MPa) 2.000 Temperature(K) 636.312
Wall Temperature(K) 688.931 Heat Transfer(w/cm2K) 10.20
Clad Temperature(K) 690.450
Fuel Temperature(K) 719.981

HEAT BALANCE FOR THE BUNDLE

dQ: heat to coolant (joule/kg) 0.12054E+07
dH: change of enthalpy (joule/kg) 0.12054E+07
dKin change kinetic energy(joule/kg) 0.72267E-03
relative dKin/dQ 0.000000
Balance (dH+dKin)/dQ 0.9999923

9.0 References

1. N. M. Greene, W. E. Ford III, L. M. Petrie, J. W. Arwood, "AMPX-77: A Modular Code System for Generating Coupled Multigroup Neutron-Gamma Cross Sections Libraries from ENDF/B -IV and/or ENDF/B-V," ORNL/CSD/TM-283 (October 1992).

2. W. E. Ford III, et. al., "ANSL-V: ENDF/B-V Based Multigroup Cross-Section Libraries for Advanced Neutron Source (ANS) Reactor Studies," ORNL-6618 (September 1990).

3. John R. Lamarsh, "Introduction to Nuclear Reactor Theory," Addison-Wesley Publ. Co. (1966).

4. J. C. Moyers and J. P. Nichols, "ALKASYS, A Computer Program for Studies of Rankine-Cyde Space Nuclear Power Systems," ORNL/TM-10427 (September 1987).

5. "Reactor Physics Constants," ANL-5800 Second Edition (July 1963).

6. G. Melese and R. Katz "Thermal and Flow Design of Helium - Cooled Reactors," American Nuclear Society (1984).

7. Y. S. Tang, R. D. Coffield, and R. A. Markley, "Thermal Analysis of Liquid-Metal Fast Breeder Reactor," American Nuclear Society (1978).

8. D. W. Jeppson, et. al., "Lithium Literature Review: Lithium's Properties and Interactions," HEDL-TME 78-15 UC-20, Hanford Engineering Development Laboratory (April 1978).

9. M. M. El-Wakil, "Nuclear Power Engineering," McGraw-Hill Book Co. (1962).

10. R. W. Bussard and R. D. DeLauer, "Fundamental of Nuclear Flight," McGraw-Hill Book Co. (1965).

11. "CRC Handbook of Chemistry and Physics," 48th Edition (1967-1968).

12. "Steam, its Generation and Use," Babcock and Wilcox, 38th Edition (1972).

APPENDIX A. LISTING OF CODE NEPNERVA

```
c program to estimate preliminary design of Nerva derivative reactor for
c space applications.
c Assumptions
c 1) fuel cell: He coolant,ZrC clad,UC-ZrC mix in Graphite matrix
c
c 2) support cell: He coolant,Inconel support for ZrH, coolant,Pyrolitic
c graphite and graphite (last two as thermal shield for ZrH
c
      character*80 tit,aldum
c
      dimension buw(5,5),bup(5),aux(5),wbeth(8),th(8),pfr(5),sof1(5)
     *,pfrm(5),Sogam(7),Pr5(7),Emin(5),Ras(5),Has(5),Rop(5),Hop(5)
c
      common /Soze/ its,epss
c
      common /MaBaSi/ R,H,ishape
c
      common /Bala/ Ereq,buwo(5),relr,rela1,rela2,dkeol,dkstr,BU,Vcor,
     *U5m,dkbu,dkxes,dkxep,dksms,dksmp,dkbe,rkre,eps,itmax,dcr
     *,itest,aofis
c
      common /cont/ Rad,drr,dref,rasp,refec,ncr,dksu,dkest,relar
c
      common /Fluxo/ Pow,Vc,sofr,fluxn(4),xedks,xedkt,tmax,smdks,smdkt,
     *tsm,D,iwxe
c
      common/thermo/ tit,nnodo,iexpl,ifri,imax,norif,iditus,nfu,pin,tin
     *,dcm,ru,xi,der,epso,Powt,rjs,flowt,GU235,Vact,Enr,PUC,Rouc,RoZrC
     *,RGraf,dclad,Hact,Pfacr,Daxi,fact,pf,tf,fe,iprth,iopth,iouth,
     *Pfacaz,iprob,flowal,iboun
c
      common /fphe/ imaxt,epsi,dero
c
      common /Gamsh/ coBe,coTa,coC,coZr,coU,coH,coNi,coFe,coCr,coTi,coSi
     *,dist,Req,shg(7),shgnb(7),Zeq(7),Dosewg(7),Doseg(7),itmag,epsg
     *,aten,Dosen,Doseng(7)
c
      Common /CROSS/ AU,BBU,signe,ishin,ishig
c
      open(unit=7,file='nepnerva3.i',status='old')
      open(unit=8,file='nepnerva3.out',status='unknown')
c
      read(7,11) tit
11    format(a80)
c
      read(7,11) aldum
      read(7,20) itest,ndt,itmax,icont,nput
      read(7,11) aldum
      read(7,10) Pow,D,BU,eps,der
      read(7,11) aldum
      read(7,10) drr,dra1,dra2
      read(7,11) aldum
      read(7,10) dkstr,dkeol,dksu
      dkest=dkstr
      if(icont.eq.0) go to 2378
      read(7,11) aldum
      read(7,20) ncr,nnodo,nfu,ishape
      read(7,11) aldum
      read(7,20) iboun,iopth
      read(7,11) aldum
      read(7,20) iexpl,ifri,iditus,iprth,iouth
      read(7,11) aldum
      read(7,20) iwrte,ishn,ichoose
      read(7,11) aldum
```

```
      read(7,10) pin,tin,ps,ts
      read(7,11) aldum
      if(iboun.eq.0) read(7,10) flowt
      if(iboun.eq.1) read(7,10) toutl
      read(7,11) aldum
      read(7,10) ru,xi,fact,Ripple
      read(7,11) aldum
      read(7,10) tempv,depvw
      read(7,11) aldum
      read(7,10) fastf,Dose,zload,rload
      read(7,11) aldum
      if(ishape.eq.1) go to 2378
      if(ishape.eq.2) read(7,10) R
      if(ishape.eq.3) read(7,10) H
2378  continue
c
c
      epso=eps
      epsi=eps
      epsg=eps
      epss=eps
      dero=der
      Powt=Pow
      Daxi=5.
c
      Ereq=Pow*D*365.25
c
c
c        INPUT DESCRIPTION FOR REALS
c
c
c Pow(Mw): power;  D(years): core life; BU(at%): percent U235 atoms
c burned;
c Inlet flow conditions:
c pin,tin(MPa,K)) pressure and temperature of the inlet coolant(fuel el
c ps,ts           pressure and temperature coolant(sup el), just for
c                 book-keeping
c drr,dra1,dra2 (cm) thickness of Be reflector:radial,top-bottom axial
c dkstr(%): reactivity structural materials(estimated)
c dkeol(%):     "       at end of life(desired)
c dksu (%):desired subcritical reactivity required at BOL with drums in
c eps: tolerance for  iterations (like energy)
c ru: rugosity coolant channel; xi: /0., 1.0/ fully explicit to implicit
c der=1.01 (compute derivatives)
c fact:fraction fission energy/thermal
c flowr(kg/s)total coolant flow(applicable if iboun=0)
c Daxi(cm): extrapolation length axial cos power distribution
c Ripple:Relative amplitude of power distribution as function of angle
c at R=Rcore
c tempv(K): Temperature of Pressure Vessel; if <0. program chooses tout
c depvw(cm):Max thickness of the pressure vessel acceptable
c toutl(K): outlet temperature of coolant(applicable if iboun=1)
c fastf(1/cm2): fast fluence (during time D) at payload area
c zload(m) : axial distance from the base of reactor to payload plane
c rload(m) : radius of load area
c Dose(Rad): integrated gamma dose(during time D) required for payload
c ddt(cm): delta r and delta h for test curve Eavailable=f(R) or f(H)
c
10    format(7e10.0)
c
      imax=itmax
      imaxt=itmax
      itmag=itmax
      its=itmax
c
c
c       INPUT DESCRIPTION FOR INTEGERS
```

```
c
c
c itmax: max number iterations for reactor search
c ncr: number of control drums
c ishape/1,2,3/ Shape: ideal(minimum volume),R input,H input
c iwrte=1 write iterations
c itest=0/1/ no,yes run and print test to know your options
c nnodo number of nodes in coolant channel
c iexpl/0,1/ implicit,explicit solution of coolant channel
c ifri/0,1/ no,yes friction factors;iditus/0,1/ no,yes Dittus Correlation
c nfu; number of nodes inside fuel mix
c iprth/0,1/no,yes print details cooling channel
c iopth=1 3-D uniform power density assumed
c       =2 2-D (r,theta) uniform power density assumed
c       =3 Uniform Fuel load/uniform coolant flow assumed
c       =4 Uniform Fuel load/Coolant flow proportional to power density
c          assumed
c iouth=1 Thermodynamic profile at Max power density channel is printed
c iouth=2          "          "    " Min    "       "       "     "   "
c iouth=3          "          "    " Max and Min   "       "       "     "   "
c iboun=0 Input Coolant Flow
c iboun=1 Input Outlet Temperature of Coolant
c ishn=1,2  Choose Neutron Shielding LiH+SS/ B4C
c icont/0,1/ no,yes continue calculation after test
c ndt: number of intervals between Xas and Xop to compute Energy=f(R)
c or f(H), nput total number of points in tabulation
c ichoose=0,1 no,yes correction,by program,of incompatible request
c
c
c
20     format(14i5)
c
       if(BU.gt.50.) write(8,32) BU
c
32     format(//' YOUR VALUE FOR BURN UP',f8.2,' IS TOO LARGE FOR'/
      *          ' A REASONABLE EXTRAPOLATION OF THE DATA BASE'//
      *          ' PROGRAM STOPS '//)
c
       if(BU.gt.50.) stop
c
       if(drr.gt.40..or.dra1.gt.40..or.dra2.gt.40.)
      *write(8,31) drr,dra1,dra2
31     format(//' YOUR VALUE OF REFLECTOR THICKNESS(CM)',3f8.2/
      *          ' ARE TOO LARGE FOR A REASONABLE EXTRAPOLATION'/
      *          ' OF THE DATA BASE'//
      *          ' PROGRAM STOPS '//)
c
       if(drr.gt.40..or.dra1.gt.40..or.dra2.gt.40.) stop
c
c
       data enr,ff,dh,dc,puc,dref,nh/.93,1.913,2.794,0.1,0.5,30.,19/
c
c enr: enrichment(at %);
c fuel element; ff(cm): flat to flat distance hexagonal element
c dh(mm) coolant hole; dc(mm) cladding thickness
c puc: mole fraction of UC in mix UC-ZrC
d dref(cm): thickness of Be reflector(for data base)
c nh number of coolant holes per fuel element(data base calculated,nh=19)
c
       dcm=0.1*dh
       dclad=0.1*dc
       data r1,r2,r3,r4,r5/0.15,0.25,0.65,0.75,0.85/
c
c Support Element, radius of different regions(cm)
c r1,Coolant;r2,Inconel;r3,ZrH;r4,Coolant;r5,PyroGraphite
c data base calculated with these values
c
```

```
      data pee,Av,ABe,xop/3.1415926,0.602202,
     *                    9.01219,0.541315/
c
c xop:optimum shape for cylindrical reactors; Av: Avogadro's number
c ABe: Be at-weight;pisq3=pi*sqrt(3)
c
      data au5,ah,ahe,azr,ac,ani,acr,afe,ati,asi,au8/235.044,1.00797,4.0
     *026,91.22,12.01115,58.71,51.996,55.847,47.90,28.086,238.0508/
c
c atomic weights
c au5:U235; ah:H; ahe:He; azr: Zr; ac: C; ani: Ni; acr: Cr; afe: Fe
c ati: Ti; asi: Si;au8: U238
c
      data xh1,xh2,xc12,xc13/.99985,0.00015,.9889,.0111/
c
c atomic fractions
c xh1,xh2: H and deuterium in nat H
c xc12,xc13: C12 and C13 in nat C (not used in present version)
c
      data rouc,rozrc,rgraf,rgrap,roin,rozrh/12.97,6.7,1.7,1.7,8.90,5.6/
c
c densities (g/cm3)
c rouc: UC; rozrc: ZrC; rgraf: Graphite; rgrap: pyrolytic Graphite;
c roin: inconel alloy; rozrh:ZrH2
c
      data xni,xcr,xfe,xTi,xsi,roBe,RoAst,RoW,RoLis,RoBC
     */0.73,0.15,0.07,0.025,0.025,1.848,16.84,19.3,1.024,2.92/
c
c weight fraction inconel alloy
c xni: Ni; xcr: Cr; xfe: Fe; xTi: Ti;xsi: Si
c Densities(g/cm3): Ro...
c roBe: Be;RoAst: Astar Alloy ; RoLis:LiH+Stainless Steel N'Shield
c RoBC: B4C(N' Shield)
c
      data wfi,rnu,remH,remBe,remC,remFe,remNi,remZr,remU,remTa
     */3.2042e-11,2.44,1.,1.07,0.72,1.98,1.89,2.36,3.6,3./
c
c wfi: joules/fission(200Mev); rnu: number of neutrons per fission
c rem H,Be,C,Fe,Ni,Zr,U,Ta : removal cross sections(barns) for fast neut
c
      data (pfr(i),i=1,5)/1.67,1.71,1.77,1.77,1.81/
c
c pfr radial maximum power factor at BOL(These are number with control
c drums in, then at center line and for uniform load)
c
      data (pfrm(i),i=1,5)/0.577,0.423,0.311,0.311,0.257/
c
c pfrm minimum radial peaking factors at BOL( at border core)
c same restrictions as pfr
c
      data (buw(1,i),i=1,5)/0.,0.,0.,0.,0./
      data (buw(2,i),i=1,5)/3.42,2.28,1.75,1.73,2.21/
      data (buw(3,i),i=1,5)/7.17,4.82,3.77,3.80,4.86/
      data (buw(4,i),i=1,5)/11.35,7.70,6.14,6.33,8.10/
      data (buw(5,i),i=1,5)/16.09,11.01,9.03,9.50,12.13/
      data (bup(i),i=1,5)/0.,10.,20.,30.,40./
c
c buw reactivity worth(% delta k) at bup(atom % burn up) for
c the following values of Support/Fuel elements
c
      data (sof1(i),i=1,5)/0.0,0.2,0.5,1.0,2.0/
c
c sof1: Support/Fuel ratio
c
c
      data (wbeth(i),i=1,8)/0.,0.2728,0.3947,0.5026,0.6778,0.8524,
     *                      0.9517,1.000/
```

```
c wbeth: relative(to 30 cm) worth of Be as function of its thickness
c th(cm)(input next)
c
      data (th(i),i=1,8)/0.,3.67,5.67,7.67,11.67,17.67,24.,30./
c
c burn-up worths for input BU,stored at buwo()
c
      RoAst=RoAst/1000.
      isof1=5
      ibup=5
      icart=1
1274  continue
      do 101 i=1,isof1
      do 102 j=1,ibup
102   aux(j)=buw(j,i)
      call Inter(bup,aux,ibup,BU,y)
101   buwo(i)=y
c
c relative worth of Be reflectors: relr, radial;rela1,rela2:axial
c
      ibe=8
      call Inter(th,wbeth,ibe,drr,relr)
      relar=relr
      call Inter(th,wbeth,ibe,dra1,rela1)
      call Inter(th,wbeth,ibe,dra2,rela2)
      rela1=0.5*rela1
      rela2=rela2*0.5
c
c if itest=1 print your options:
c
c  TTTTT EEEEE SSSSS TTTTT     1) Energy available with optimum shape
c    T   E     S       T       2) Radius of infinite cylinder
c    T   EEE    S      T       3) Height of infinite "pancake"
c    T   E        S    T       4) Energy available as function of R and
c    T   EEEE  SSSSS   T          H
c                               (2 or 3 would produce "infinite" energy)
c
c
c
c
      ish=ishape
      RRR=r
      HHH=h
c
c
      itpr=itest
      if(itpr.eq.1)
     *write(8,6133) Ereq,BU,drr,dr1,dr2,dkeol,dkstr
6133  format(//' INPUT'/' You requested Mwdays',f10.2/
     *                    ' And burn-up(%)',f5.2/
     *' With radial Be reflector thickness(cm)',f6.2/
     *'  "   top     "     "       "          ',f6.2/
     *'  "   bottom  "     "       "          ',f6.2/
     *'            End of Life delta k(%)     ',f6.2/
     *'            Structural  delta k(%)     ',f6.2/)
      if(itpr.eq.1)
     *write(8,1632)
c
1632  format(' Run Test, Energy Available (Mwd) for Optimum Shape'/
     *       ' with corresponding Rop and Hop'/
     *       '           Critical Radius(cm) for Infinite Reactor'/
     *       '           Critical Height(cm) for Infinite Reactor'/
     *       ' For your input conditions'/
     *'S/F         Energy        Rop        Hop      Rc          Hc'/)
c
c
c compute Energy available for ideal shape Emin()
c and asymptotic radius Ras() and height Has()
```

```
c
      itest=1
      do 1633 is=1,isof1
      ishape=1
      sofac=sof1(is)
      Ene=Balance(ibu,sofac)
      Emin(is)=Ene
      Rop(is)=R
      Hop(is)=H
      Ropo=R
      Hopo=H
      ishape=2
      Edu=Balance(ibu,sofac)
      Rcri=dcr
      Ras(is)=Rcri
      ishape=3
      Edu=Balance(ibu,sofac)
      Hcri=dcr
      Has(is)=Hcri
c
c
      if(itpr.eq.1)   write(8,1634) sofac,Ene,Ropo,Hopo,Rcri,Hcri
1633  continue
c
1634  format(f6.3,f12.1,4f12.2)
c
      itest=itpr
      if(itest.eq.0) go to 1631
c
c compute curves Eavailable as function of radius and Height of core and
c S/F in order to illustrate the user about the options.
c
      do 5782 is=1,isof1
      sofac=sof1(is)
c
c
      write(8,5783) sofac,Ereq
c
5783  format(//' AVAILABLE ENERGY(MWDAYS) AS FUNCTION OF S/F AND R,H'/
     */'S/F=',f8.4/' Required Energy(Mwd)',f10.2//
     *'    I',4x,'R(CM)',6x,'R/H',7x,'EAVAI',4x,'H(CM)',6x,'R/H',,7x,'EA
     *VAI'//)
c
      Rs0=Ras(is)*der
      Hs0=Has(is)*der
      rnt=ndt
      ddr=(Rop(is)-Rs0)/rnt
      ddh=(Hop(is)-Hs0)/rnt
      ndtt=ndt+1
      do 5782 idt=1,nput
      ridt=idt-1
      ishape=2
      Rs=Rs0+ridt*ddr
      R=Rs
      Ener=Balance(ibu,sofac)
      xra=R/H
      ishape=3
      Hs=Hs0+ridt*ddh
      H=Hs
      EneH=Balance(ibu,sofac)
      xh=R/H
c
c
      write(8,5784) idt,Rs,xra,Ener,Hs,xh,EneH
c
5784  format(i5,2(f9.2,f9.3,f12.0))
c
```

```
5782  continue
c
c
c     EEEEE   N   N  D       TTTTT EEEEE SSSSS TTTTT
c     E       NN  N  D D       T   E     S       T
c     EEE     N N N  D  D      T   EEE    S      T
c     E       N  NN  D D       T   E        SS   T
c     EEEEE   N   N  D         T   EEEE  SSSSS   T
c
c 1631 bypass prints and some checks on energy
c
1631  continue
c
c Stop here if your intentions are only to run the test
c
      if(icont.eq.0) stop
      ishape=ish
      r=RRR
      h=HHH
      itest=0
c
c check if you are requesting too little energy
c
      Emino=Emin(isof1)
       if(Ereq.ge.Emino) go to 337
      if(icart.eq.1)
     *write(8,5133) Ereq,BU,drr,dr1,dr2,dkeol,dkstr
5133  format(///' ATTENTION'/' You requested Mwdays',f10.2/
     *                        ' And burn-up(%)',f5.2/
     *' With radial Be reflector thickness(cm)',f6.2/
     *'  "   top     "     "       "           ',f6.2/
     *'  "   bottom  "     "       "           ',f6.2/
     *'            End of Life delta k(%)      ',f6.2/
     *'            Structural  delta k(%)      ',f6.2//
     *' Trying to satisfy reactivity balance,it happens your core'/
     *' is too large(and it would produce more energy you need).'/
     *' You might:'/
     *' 1) Reduce the input burn-up, and/or'/
     *' 2) Increase the reflector thickness,and/or'/
     *' 3) Reduce delta k at end of life, and/or'/
     *' 4) Reduce delta k structural')
      if(ichoose.eq.1) write(8,1424)
1424  format(///' I choose=1, Program stops')
      if(ichoose.eq.1) stop
      if(icart.eq.1) write(8,1426)
1426  format(///' Program will change burn-up'//
     *'          BU       Mwday'/)
      icart=0
      write(8,1425) BU,Emino
1425  format(2f12.2)
      BUA=BU
      BU=BU-1.
      if(BU.le.0.) BU=BUA/2.
      go to 1274
337    continue
c
c check if there is enough energy for ideal shape(ishape=1)
c
      iwae=0
      if(Ereq.le.Emin(1)) go to 357
      iwae=1
c
c set the warning flag iwae=1 for potential strange shapes:cigar/pancake
c and calculate R1,R2,R3,R4 and H1,H2,H3,H4 which are the limits for R
c and H for the input conditions(reactivity,required Energy) in the
c bifurcation region
c
```

```
      itest=1
      ishape=2
      sofac=sof1(isof1)
      Das=Ras(isof1)
      Dop=Rop(isof1)
      call Size(Das,Dop,Ereq,sofac,R11,R41)
      sfr14=sofac
      hr4=h
      hr1=Vcor/(pee*r11**2)
      vr14=Vcor/1000.
      sofac=sof1(1)
      Das=Ras(1)
      Dop=Rop(1)
      call Size(Das,Dop,Ereq,sofac,R21,R31)
      sfr23=sofac
      hr3=h
      hr2=Vcor/(pee*R21**2)
      vr23=Vcor/1000.
      ishape=3
      sofac=sof1(isof1)
      Das=Has(isof1)
      Dop=Hop(isof1)
      call Size(Das,Dop,Ereq,sofac,H1,H4)
      sfh14=sofac
      rh4=R
      rh1=sqrt(Vcor/(pee*H1))
      vh14=Vcor/1000.
      sofac=sof1(1)
      Das=Has(1)
      Dop=Hop(1)
      call Size(Das,Dop,Ereq,sofac,H2,H3)
      sfh23=sofac
      rh3=R
      rh2=sqrt(Vcor/(pee*H2))
      vh23=Vcor/1000.
      write(8,369) Ereq,Emin(1),R11,R21,R31,R41,sfr14,vr14,hr1,hr4,sfr23
     *,vr23,hr2,hr3
369   format(//' ATTENTION:'/
     *' You requested Mwdays',f10.2,' larger than',f10.2/
     *' which is the largest energy available for ideal shape.'/
     *' Strange shapes(cigar/pancake) can appear as solutions,'/
     *' the following designs are possible:'/
     *'  ishape=2 with R(cm) between'/
     *'                                R1',f8.2,' and R2',f8.2/
     *'                or  R(cm) between'/
     *'                                R3',f8.2,' and R4',f8.2/
     *'R1 and R4 corresponds to S/F=',f8.3,' with core Vol(L)',f9.1/
     *'          with H(cm),Respectively=',2f10.2/
     *'R2 and R3 corresponds to S/F=',f8.3,' with core Vol(L)',f9.1/
     *'          with H(cm),Respectively=',2f10.2/)
      write(8,412) H1,H2,H3,H4,sfh14,vH14,rh1,rh4,sfh23,vh23,rh2,rh3
412   format(
     *' Or  ishape=3 with H(cm) between'/
     *'                                H1',f8.2,' and H2',f8.2/
     *'                  or  H(cm) between'/
     *'                                H3',f8.2,' and H4',f8.2/
     *'H1 and H4 corresponds to S/F=',f8.3,' with core Vol(L)',f9.1/
     *'          with R(cm),Respectively=',2f10.2/
     *'H2 and H3 corresponds to S/F=',f8.3,' with core Vol(L)',f9.1/
     *'          with R(cm),Respectively=',2f10.2/)
      write(8,7133) BU,drr,dr1,dr2,dkeol,dkstr
7133  format(/' The previous data corresponds to your input conditions'/
     */' Burn-up(%)',f6.2/
     *' Radial Be reflector thickness(cm)',f6.2/
     *'  Top     "     "       "          ',f6.2/
     *'  Bottom  "     "       "          ',f6.2/
     *'        End of Life delta k(%)     ',f6.2/
```

```
     *'        Structural  delta k(%)      ',f6.2//
     *' If awkward values of Radius/Height appear YOU MIGHT:'///
     *' 1) Increase the burn-up, and/or'/
     *' 2) Decrease the reflector thickness,and/or'/
     *' 3) Increase delta k at end of life, and/or'/
     *' 4) Increase delta k structural'//
     *' #1 is very effective,#2,3 and 4 not so much'/
     *' If problem persist better try other kind of reactor'//)
      itest=0
      ishape=ish
      R=RRR
      H=HHH
357   continue
      if(ishape.eq.1.or.iwae.eq.1) go to 434
c
c begin check of input R or H if E requested is not in bifurcation
c region
c
      itest=1
      if(ishape.eq.2)  then
      sofac=sof1(isof1)
      Das=Ras(isof1)
      Dop=Rop(isof1)
      call Size(Das,Dop,Ereq,sofac,R11,R21)
      R=RRR
      if(R11.lt.R.and.R.lt.R21) go to 443
      if(ichoose.eq.1) write(8,4551) ichoose,R,R11,R21
4551  format(//' I choose Flag=',i5,' Your R=',f8.2/
     *       '                          Outside Possible Range',2f10.2/
     *       ' Program stop'//)
      if(ichoose.eq.1) stop
      Rin=R
      R=0.5*(R11+R21)
      write(8,4552) ichoose,Rin,R11,R21,R
4552  format(//' I choose Flag=',i5,' Your R=',f8.2/
     *       '                          Outside Possible Range',2f10.2/
     *       ' Program continues with average, R=',f8.2//)

443   continue
      else
      sofac=sof1(isof1)
      Das=Has(isof1)
      Dop=Hop(isof1)
      call Size(Das,Dop,Ereq,sofac,H1,H2)
      H=HHH
      if(H1.lt.H.and.H.lt.H2) go to 454
      if(ichoose.eq.1) write(8,4553) ichoose,H,H1,H2
4553  format(//' I choose Flag=',i5,' Your H=',f8.2/
     *       '                          Outside Possible Range',2f10.2/
     *       ' Program stop'//)
      if(ichoose.eq.1) stop
      Hin=H
      H=0.5*(H1+H2)
      write(8,4554) ichoose,Hin,H1,H2,H
4554  format(//' I choose Flag=',i5,' Your H=',f8.2/
     *       '                          Outside Possible Range',2f10.2/
     *       ' Program continues with average, H=',f8.2//)

454   continue
      endif
      itest=0
434   continue
      if(iwae.eq.0) go to 1004
      go to(1001,1002,1003),ishape
1001  continue
      if(ichoose.eq.1) write(8,455) ichoose
455   format(//'I choose Flag=',i3,' Not enough available energy'/
```

```
     *          '                          for ishape=1 option, program stop')
      if(ichoose.eq.1) stop
      ishape=2
      R=R31*1.01
      write(8,4555) R
4555  format(//
     *' Not enough energy for ishape=1(input), program continues with'/
     *' with R=',f8.2,' toward pancake shape'//)
      go to 1004
1002  continue
      if(R11.lt.R.and.R.lt.R21) go to 1004
      if(R31.lt.R.and.R.lt.R41) go to 1004
      if(ichoose.eq.1) write(8,4556) ichoose,R
4556  format(//
     *'I choose Flag=',i3,' Your input R=',f8.2,' Out Possible Range'/
     *' Program stop')
      if(ichoose.eq.1) stop
      Rin=R
      R=R31*1.01
      write(8,4557) Rin,R
4557  format(//
     *' Your input R=',f8.2,' is outside possible range, program'/
     *' continues with R=',f8.2,' toward pancake shape'//)
      go to 1004
1003  continue
      if(H1.lt.H.and.H.lt.H2) go to 1004
      if(H3.lt.H.and.H.lt.H4) go to 1004
      if(ichoose.eq.1) write(8,4558) ichoose,H
4558  format(//
     *'I choose Flag=',i3,' Your input H=',f8.2,' Out Possible Range'/
     *' Program stop')
      if(ichoose.eq.1) stop
      Hin=H
      H=H2*0.99
      write(8,4559) Hin,H
4559  format(//
     *' Your input H=',f8.2,' is outside possible range, program'/
     *' continues with H=',f8.2,' toward pancake shape'//)
1004  continue
c
c
      ibu=0
      sofa1=sof1(1)
      bal1=Balance(ibu,sofa1)
      if(ibu.eq.1) write(8,597)
597   format(' Stop trying to bracket demand')
      if(ibu.eq.1) stop
      sofa2=sof1(isof1)
      bal2=Balance(ibu,sofa2)
      if(ibu.eq.1) write(8,597)
      if(ibu.eq.1) stop
c
c if we successfully arrived to this point,it means we bracket the demand
c Then:
c bisect  Balance until Eavailable=Erequested
c
      it=0
145   sofa=0.5*(sofa1+sofa2)
      Erema=Balance(ibu,sofa)
      if(ibu.eq.1) write(8,597)
      if(ibu.eq.1) stop
      test=abs(Erema)/Ereq
      if(test.lt.eps) go to 144
      it=it+1
      if(iwrte.eq.1) write(8,219)
219   format('  it',7x,'sofa1',8x,'sofa',7x,'sofa2',8x,'bal1',7x,'Erem',
     *8x,'bal2')
```

```
      if(iwrte.eq.1) write(8,218)it,sofa1,sofa,sofa2,ball,Erema,bal2
218   format(i4,3f12.6,3f12.3)
      call bisect(sofa1,sofa2,sofa,ball,bal2,Erema,iwa)
      if(it.le.itmax.and.iwa.eq.0) go to 145
144   continue
      if(it.gt.itmax.or.iwa.eq.1) then
      write(8,146) sofa1,ball,sofa2,bal2,it,iwa
146   format(///' Problems trying to bisect  Balance'/
     *          ' sof1=',e12.4,' ball=',e12.4/
     *          ' sof2=',e12.4,' bal2=',e12.4/
     *'                              iteration=',i5/
     *'                                    iwa=',i5/)
      stop
      else
      continue
      endif
c
c successful end of reactor search, calculate number densities and masses
c
      GU235=U5m
      Vact=0.001*Vcor/(1.+sofa)
      U5m=U5m/1000.
      rasp=sofa
      sofr=sofa
      Vc=Vcor
c
c with power level ,s/f and volume of the core, compute fluxes and
c reactivities of Xe and Sm
      Call Flux
c choose R as near possible to ideal R with integer number of elements
      ae=0.86602540*ff**2
      ne=pee*R**2/ae
      re=ne
      if(rasp.eq.0.) then
      nf=ne
      rf=nf
      nse=0
      rse=0.
      go to 1975
      else
      rf=re/(1.+rasp)
      nf=rf
      nse=ne-nf
      rse=nse
      endif
1975  R=sqrt(ne*ae/pee)
      H=Vcor/(ne*ae)
      Hact=H
c
c
c fuel microcell, calculation of geometry and concentrations
c assume clad around coolant hole and around fuel element
c aae is the active area of ae, ac1 is the area of the clad around hole
c and ac2 is the area of the clad around fuel element
c aco is the area of the coolant hole
c
      dcc=0.1*dc
      rh=dh*0.1/2.
      ac1=pee*dcc*(2.*rh+dcc)*nh
      b=ff/1.7320508
      bp=b-dcc/0.86602540
      ac2=3.*(b+bp)*dcc
      aco=pee*rh**2*nh
      aae=ae-ac1-ac2-aco
c
c volumetric fractions
c
```

```
      fe=aae/ae
      f1=ac1/ae
      f2=ac2/ae
      fco=aco/ae
c
c radius of microcell: rm1,coolant; rm2,clad; rm3 microcell
c
      rm1=rh
      rm2=rh+dcc
      amicro=pee*rm2**2+aae/nh
      rm3=sqrt(amicro/pee)
c
c Calculate outlet condition, ie. if iboun=0(flow is input)>houtlet
c                                if iboun=1(tout is input)>flow
c See: 1) no velocities in energy conservation (see effects in detailed
c         of the channel)
c      2) tout=f(hout,pinlet) (reasonable approximation)
c      3) from 1) and 2) tout is reasonable.
c
      pino=pin*1.e+6
      call phe(hin,pino,tin,rhor,vis,cp,pr,a,rrk,xx,0)
      if(iboun.eq.0) go to 391
      call phe(hout,pino,tout1,rhor,vis,cp,pr,a,rrk,xx,0)
      flowt=fact*Pow*1.e+6/(hout-hin)
c
c alternative flow in case of problem in thermo
c
      call phe(hina,1.e+7,20.,rh,visa,cpa,pra,aa,rka,xxa,0)
      call phe(houta,1.e+7,tout1,rh,visa,cpa,pra,aa,rka,xxa,0)
      flowal=fact*Pow*1.e+6/(houta-hina)
391   hout=hin+fact*Pow*1.e+6/flowt
      call phe(hout,pino,tout,rhor,vis,cp,pr,a,rrk,xx,1)
c
c calculate maximum stress(sigma) for the Astar Pressure Vessel
c
      if(tempv.lt.0.) tempv=tout
      call desstr(tempv,1,D,sigma,sigrod,itmax,ipresv)
c
c number densities; for He: estimate taverage as the next statement
      tav=0.5*(tin+tout)
      rhe=rho(pino,tav)
      hen=0.001*rhe*Av/Ahe
      rf=nf
      ru5=1000.*u5m/(rf*aae*H)
      u5n=ru5*Av/Au5
      u8n=u5n*(1./enr-1.)
c
c Assume UC+ZrC (puc molar fraction)
c
      un=u5n+u8n
      cn2=(1./puc-1.)*un
      zrn2=cn2
      rzrca=(cn2*ac+zrn2*azr)/av
      cn2=cn2+un
c Calculate space occupied by the graphite
      rouca=(u5n*au5+u8n*au8+un*ac)/av
c Volumetric fraction in UC-ZrC-Graphite mix
      xuc=rouca/rouc
      xzrc=rzrca/rozrc
      xgraf=1.-xuc-xzrc
      rgraa=xgraf*rgraf
      grafn=rgraa*av/ac
c densities in clad
      cn1=rozrc*av/(ac+azr)
      zrn1=cn1
c Average densities in fuel element(including outer clad)
      hena=hen*fco
```

```
      cna=cn1*(f1+f2)+cn2*fe
      zrna=zrn1*(f1+f2)+zrn2*fe
      grafa=grafn*fe
      u5na=u5n*fe
      u8na=u8n*fe
      Vcore=Vcor/1000.
c
c
      write(8,11) tit
c
      write(8,51) Pow,D,Ereq,BU,drr,dra1,dra2,test
c
51    format(//' INPUT TO THE PROGRAM'//' Power(Mw)',f8.2,' Core life',f
     *8.3,' years'/' Energy Released(Mwday)',f10.2,' Burn-up(%)',f6.2//
     *' Be reflector, radial thickness(cm)',f6.2/
     *'                top          "       ',f6.2/
     *'                bottom       "       ',f6.2//
     *' The following design can produce that energy within the relative
     *'/'numerical precision of',e12.4)
c
      if(ishape.eq.1) write(8,317) xop
317   format(/' Minimum Core Volume Chosen: R/H=',f9.6/)
c
      if(ishape.eq.2) write(8,318) R
318   format(/' Core Radius Chosen: R(cm)=',f7.2/)
c
      if(ishape.eq.3) write(8,319) H
319   format(/' Core Height Chosen: H(cm)=',f7.2/)
c
      dkfp=dkbu+dkxep+dksmp
      dkreq=dkfp+dkstr+dkeol
      dkbe=dkbe/100.
      refec=rkre+dkbe
c
c
      write(8,52) dkreq,dkfp,dkbu,dkstr,dkeol,refec,rkre,dkbe
c
52    format(' And it has the following reactivity balance'/
     *' Delta k required(%)',f6.2,' with components'/
     *'  Fission Products(%)',f6.2/
     *' " "(except Xe,Sm)(%)',f6.2/
     *'         Structural(%)',f6.2/
     *'       End of Life(%)',f6.2/
     *' keff at BOL(No structural mat)',f7.4, ' Components'/
     *'                    Bare Reactor',f7.4/
     *'                       Reflector',f7.4//)
c
c
      write(8,521)sofr,xedks
c
521   format(' Support/Fuel Ratio=',f8.4,' Steady Xe reactivity(%)'f6.2)
      if(iwxe.eq.0) write(8,1521) xedkt,tmax
1521  format(
     *' Xe Reactivity after trip(%)',f6.2,' Peak at time(hr)',f6.2)
      if(iwxe.eq.1) write(8,1522)
1522  format(' No Xe peak after shutdown')
      write(8,5211)smdks,smdkt,tsm,(fluxn(i),i=1,4)
5211  format(
     *' Sm Reactivity (%) Steady',f6.2,' Peak after trip(%)',f6.2/
     *' Sm saturates at days',f8.2/
     */' Neutron Flux(1/cm2sec)(core average) >100.Kev',e12.4/
     * '    "       "      "        "     "     >100. ev',e12.4/
     * '    "       "      "        "     "     >  3. ev',e12.4/
     * '    "       "      "        "     "     <  3. ev',e12.4//)
c
      Rad=R
c
```

```
c call control to design control drums
c
      Call Control
c
c
      write(8,30) u5m,enr,Vcore,rasp,R,H,ne,nse,nf,ff
c
30    format(' NUMBER DENSITIES AND MASSES OF NERVA DERIVATIVE REACTOR'/
     *       '          GENERAL DESCRIPTION'/
     */f10.3,' Kg of U235. U enrichment',f7.4,' Core Volume(L)',f10.2/
     *' Ratio Support/Fuel Elements',f10.5/
     *' Core Radius(cm)',f10.2,' Height(cm)',f10.2/
     *' Number of elements',i6,' Support',i6,' Fuel',i6/
     *' Hexagonal Elements. Flat to flat distance(cm)',f10.4//)
c
c
      write(8,40) puc,dc,nh,dh,pin,tav
c
40    format('            FUEL ELEMENTS(INDIVIDUAL)'/
     *' Molar UC fraction in UC-ZrC mix is',f7.4,' ZrC clad thickness
     *(mm) is',f6.2/' There are',i4,' coolant holes of mm',f7.2,' diamet
     *er'/' with He coolant at',f7.3,' MPa',f8.2,' K'/)
c
c
      write(8,50) fco,f1,f2,fe
c
50    format(' Volume fractions in Fuel Element'/'      Coolant',f7.4/
     *                                        ' Clad Coolant',f7.4
     *                                       /' Clad Fuel El',f7.4
     *                                       /' Fuel Mix    ',f7.4)
c
c
      write(8,41) xuc,rouca,xzrc,rzrca,xgraf,rgraa
c
41    format(//' Volumetric Fractions in Fuel Mix * Density(g/cm3)'/
     *         '                        UC',f10.5,f13.5/
     *         '                       ZrC',f10.5,f13.5/
     *         '                  Graphite',f10.5,f13.5//)
c
c
      write(8,60) rm1,hen,rm2,cn1,zrn1,rm3,cn2,zrn2,grafn,u5n,u8n
c
60    format(//' Fuel microcell(Number Densities 10**24 atoms/cm3)'/
     *         ' Excludes outer cladding'/
     *         ' Coolant Radius(cm)',f7.4,' He',f12.8/
     *         ' Clad Radius   (cm)',f7.4,'  C',f12.8,'   Zr',f12.8/
     *         ' Cell Radius   (cm)',f7.4,'  C',f12.8,'   Zr',f12.8/
     *         '                     Graphite',f12.8,' U235',f12.8/
     *         '                                      U238',f12.8)
c
c Total mass of the fuel elements
      Vact=rf*ae*H/1000.
      Gfuel=Vact*(hena*ahe+cna*ac+zrna*azr+grafa*ac+u5na*au5+u8na*au8)/a
     *v
      Ggraf=Vact*grafa*ac/av
      GZrCc=Vact*(f1+f2)*rozrc
      GHe=Vact*hena*ahe/av
      GZrCf=Vact*fe*rzrca
      GUC=Vact*fe*rouca
      wUC=GUC*100./Gfuel
      wZrCf=GZrCf*100./Gfuel
      wGraf=Ggraf*100./Gfuel
      wZrCc=GZrCc*100./Gfuel
      wHe=GHe*100./Gfuel
      rou5=u5m*1000./Vact
c
c
```

```
      write(8,71) Vact,Gfuel,rou5,Ggraf,wGraf,GUC,wUC,GZrCf,wZrCf,GZrCc,
     *wZrCc,Ghe,wHe
c
71    format(//'                    FUEL ELEMENTS(ALL)'/
     *         ' Volume of Fuel Elements(L)',f10.3/
     *         '                    Mass(kg)',f10.3,' gU5/L(FE)',f10.3//
     *         '                  Masses(kg) * Fractions(%)'/
     *         '          Graphite',f10.3,f13.3/
     *         '                UC',f10.3,f13.3/
     *         '      ZrC(in fuel)',f10.3,f13.3/
     *         '      ZrC(in Clad)',f10.3,f13.3/
     *         '       He(Coolant)',f10.3,f13.3)
c
c Number densities and masses of Support Element
c
      r6=sqrt(ae/pee)
      a1=pee*r1**2
      a2=pee*r2**2-a1
      a3=pee*r3**2-(a1+a2)
      a4=pee*r4**2-(a1+a2+a3)
      a5=pee*r5**2-(a1+a2+a3+a4)
      a6=ae-(a1+a2+a3+a4+a5)
      f1=a1/ae
      f2=a2/ae
      f3=a3/ae
      f4=a4/ae
      f5=a5/ae
      f6=a6/ae
      p=ps*1.e6
      rhe=rho(p,ts)
      ches=0.001*rhe*av/ahe
      cni=roin*xni*av/ani
      cfe=roin*xfe*av/afe
      ccr=roin*xcr*av/acr
      cti=roin*xti*av/ati
      csi=roin*xsi*av/asi
      czr=rozrh*av/(azr+2.*ah)
      ch=2.*czr
      cgraf=rgraf*av/ac
      cgrap=rgrap*av/ac
c averages in support Elements
      chesa=(f1+f4)*ches
      cnisa=cnis*f2
      cfea=cfe*f2
      ccra=ccr*f2
      ctia=cti*f2
      csia=csi*f2
      czra=czr*f3
      cha=ch*f3
      cgras=f5*cgrap+f6*cgraf
c Volume and masses of Support Elements
      Vsup=nse*ae*H*0.001
      Ghes=Vsup*(f1+f4)*0.001*rhe
      GInc=Vsup*f2*roin
      GZrh=Vsup*f3*rozrh
      Ggras=Vsup*(f5*rgrap+f6*rgraf)
      Gsup=Ghes+GInc+GZrh+Ggras
      fhes=1.
      fInc=1.
      fZrh=1.
      fGras=1.
      if(rasp.eq.0.) go to 1763
      fhes=100.*Ghes/Gsup
      fInc=100.*GInc/Gsup
      fZrh=100.*GZrh/Gsup
      fGras=100.*Ggras/Gsup
c
```

```
c
1763  write(8,73) Vsup,Gsup
c
73    format('             SUPPORT ELEMENTS(ALL)'/
     *         'Volume of Support Elements(L)',f10.3/
     *       '                      Mass(Kg)',f10.3//)
c
c
      write(8,74) Ghes,fhes,GInc,fInc,GZrh,fZrh,Ggras,fgras
c
74    format('           Masses(Kg) * Fractions(%)'/
     *       '         He',f10.3,f13.3/
     *       '    Inconel',f10.3,f13.3/
     *       '       ZrH2',f10.3,f13.3/
     *       '  Graphite',f10.3,f13.3//)
c
c
      write(8,636)
c
636   format(///' SUPERCELL CONFIGURATION(Number Densities 10**24atoms/c
     *m3)'//)
c
      if(rasp.eq.0.) go to 1349
c
c
      write(8,72)r1,ches,r2,cni,cfe,ccr,cti,csi,r3,czr,ch,r4,ches,r5,cgr
     *ap,r6,cgraf
c
72    format('Support Element'/
     *       'Coolant Radius(cm)',f7.4,' He',f12.8/
     *       'Inconel Radius(cm)',f7.4,' Ni',f12.8,' Fe',f12.8/
     *       '                  ',7x,' Cr',f12.8,' Ti',f12.8/
     *       '                  ',7x,' Si',f12.8/
     *       'ZrH2 Radius   (cm)',f7.4,' Zr',f12.8,'  H',f12.8/
     *       'Coolant Radius(cm)',f7.4,' He',f12.8/
     *       'Pyrographite R(cm)',f7.4,' Gr',f12.8/
     *       'Graphite Radii(cm)',f7.4,' Gr',f12.8//)
c
      rsup=sqrt(r6**2+rf*ae/(pee*rse))
c
c
      write(8,70)rsup, hena,cna,zrna,grafa,u5na,u8na
c
70    format('Ring of Fuel Elements. Average Densities'/
     *       '         Radius(cm)',f7.4,' He',f12.8,'  C',f12.8/
     *       '                  ',7x,  ' Zr',f12.8,' Gr',f12.8/
     *       '                  ',7x,  ' U5',f12.8,' U8',f12.8)
c
c Homogenized Densities in Supercell
1349  ff=rf/re
      fsup=rse/re
      hena=hena*ff
      cna=cna*ff
      zrna=zrna*ff
      grafa=grafa*ff
      u5na=u5na*ff
      u8na=u8na*ff
      ches=ches*(f1+f4)*fsup
      cni=cni*f2*fsup
      cfe=cfe*f2*fsup
      ccr=ccr*f2*fsup
      cti=cti*f2*fsup
      csi=csi*f2*fsup
      czr=czr*f3*fsup
      ch=ch*f3*fsup
      cgra=(f5*cgrap+f6*cgraf)*fsup
      cBe=roBe*Av/Abe
```

```
      RBe=R+drr
c
c
      write(8,91)R,hena,cna,zrna,grafa,u5na,u8na,ches,cni,cfe,ccr,cti,
     *csi,czr,ch,cgra,RBe,cBe
c
91    format(///' HOMOGENIZED DENSITIES IN CORE'/
     *'Radius(cm)',f9.4/
     *'From Fuel Elements'/
     * '                          He',f12.8,'  C',f12.8/
     * '                          Zr',f12.8,' Gr',f12.8/
     * '                          U5',f12.8,' U8',f12.8/
     */'From Support Elements'/
     * '                          He',f12.8,' Ni',f12.8/
     * '                          Fe',f12.8,' Cr',f12.8/
     * '                          Ti',f12.8,' Si',f12.8/
     * '                          Zr',f12.8,'  H',f12.8/
     * '                          Gr',f12.8//
     *' REFLECTOR'/
     *'External Radius(cm)',f9.4/
     * '                          Be',f12.8)
c
      rec=(grafa+cgra+cna)/u5na
      reh=ch/u5na
c
c
      write(8,92) rec,reh
c
92    format(///' Moderation by C. C/U235=',f10.4/
     *          ' Moderation by H. H/U235=',f10.4)
c
c Compute pressure vessel (if(ipresv.ne.0))
c
      Volpv=0.
      Gpv=0.
      delpv=0.
      Rcr=R+drr
      Hcr=H+dra1+dra2
      if(ipresv.eq.0) go to 590
      delpv=(pin/sigma)*Rcr
      if(delpv.lt.0.4) delpv=0.4
      if(delpv.gt.depvw) ipresv=0
      if(ipresv.eq.0) go to 590
      Volpv=2.*pee*(Rcr+delpv)**2*delpv
      Volpv=Volpv+pee*((Rcr+delpv)**2-Rcr**2)*Hcr
      Gpv=Volpv*RoAst
      Volpv=Volpv/1000.
590   continue
      Vref=pee*((R+drr)**2-R**2)*H/1000.
      Vref=Vref+pee*(R+drr)**2*(dra1+dra2)/1000.
      Gref=Vref*roBe
      Grea=Gsup+Gfuel+Gref+Gpv
      Vres=Vsup+Vact+Vref+Volpv
      fsup=100.*Gsup/Grea
      ffue=100.*Gfuel/Grea
      fref=100.*Gref/Grea
      fmpv=100.*Gpv/Grea
      fvc=100.*Vact/Vres
      fvpv=100.*Volpv/Vres
      fvs=100.*Vsup/Vres
      fre=100.*Vref/Vres
      Vcor=Vact+Vsup
      fcor=100.*Vact/Vcor
      fcore=(fvs+fvc)*0.01
      Hrea=Hcr+2.*delpv
      Rrea=Rcr+delpv
c
```

```
c calculation of neutron shielding
c
c compute fast source Sn
c
      Sn=Pow*1.e+6*rnu/wfi
c
c compute shielding by core,reflector and pressure vessel: fsh
c concentrations smothered in Vol=Vsup+Vfuel+Vref+Vpv
c
      coBe=cBe*fre*0.01
      coTa=fvpv*Av*10.*RoAst/181.
      coC=fcore*(cna+grafa+cgra)
      coZr= fcore*(Zrna+cZr)
      coU=fcore*(U5na+U8na)
      coH=fcore*ch
      coNi=fcore*cni
      coFe=fcore*cfe
      coCr=fcore*ccr
      coTi=fcore*cTi
      coSi=fcore*cSi
      sigre=coBe*remBe+coTa*remTa+coC*remC+coZr*remZr+coU*remU+
     *     +coH*remH+(coNi+coCr)*remNi+(coFe+coTi+coSi)*remFe
      Req=(750.*Vres/pee)**0.33333333
      sire=sigre*Req*2.
      fsh=3.*(0.5-(1.-(1.+sire)*exp(-sire))/sire**2)/sire
      Snsh=Sn*fsh
c Fast fluence without shielding is flufa,at distance dist
      dist=100.*zLoad+0.5*Hrea
      flufa=Snsh/(4.*pee*dist**2)
c Multiply by core life, 1 julian year=31,557,600. sec
      tiempo=D*3.15576e+7
      flufa=flufa*tiempo
c
c Calculate Neutron Shield for zero gamma Shield
c
      tne0=shieldn(flufa,fastf,ishn,0.00,igam)
      if(ishin.eq.0) tne0=0.
c
c Now, Calculation Gamma Source Sogam()
c
      Call Gamso(Sn,aofis,sofr,Sogam,Pr5)
c
c First, Gamma Shield with no Neutron Shield
c
      Call Shieldga(Sogam,Dose,Doses,Dosew,tiempo,tgam0,0.,igam1,ishn,0)
c
c Now, Coupled with n Shield
c
      Call Shieldga(Sogam,Dose,Doses,Dosew,tiempo,tgam,tne,igam2,ishn,1)
c
      if(igam2.eq.0) tne=tne0
      if(igam2.eq.0) tgam=0.
c
c Volume and Masses of shieldings
c
      Tgt=Tan((100.*rload-Rrea)/(100.*zload+Hrea))
      rr1=Hrea*Tgt+Rrea
      rr2=(Hrea+tgam)*Tgt+Rrea
      rr3=(Hrea+tgam+tne)*Tgt+Rrea
      h1=rr1/Tgt
      h2=rr2/Tgt
      h3=rr3/Tgt
      V1=pee*RR1**2*h1*0.33333333
      V2=pee*rr2**2*h2*0.33333333
      V3=pee*rr3**2*h2*0.33333333
      Volg=(v2-v1)/1000.
      Voln=(v3-v2)/1000.
```

```
      Gg=RoW*Volg
      if(ishn.eq.1) Ron=RoLiS
      if(ishn.eq.2) Ron=RoBC
      Gne=Voln*Ron
      Vols=Vres+Voln+Volg
      Gsis=Grea+Gne+Gg
      fresv=100.*Vres/Vols
      fnev=100.*Voln/Vols
      fgv=100.*Volg/Vols
      fresg=100.*Grea/Gsis
      fneg=100.*Gne/Gsis
      fgg=100.*Gg/Gsis
c
c
      write(8,789) Vres,fresv,Voln,fnev
c
789   format(
     *'         VOLUMES AND MASSES OF REACTOR AND SHIELDINGS'//
     *'           Volume(L) Fraction(%)'/
     *' Reactor ',f10.2,f10.2/
     *' N Shield',f10.2,f10.2)
c
c
      write(8,796) Volg,fgv
796   format(' G Shield',2f10.2)
c
c
      write(8,797) Vols
c
797   format(' Total    ',f10.2//)
c
c
      write(8,798) Grea,fresg,Gne,fneg
c
798   format(
     *'             Mass(Kg) Fraction(%)'/
     *' Reactor ',f10.2,f10.2/
     *' N Shield',f10.2,f10.2)
c
c
      write(8,796) Gg,fgg
c
c
      write(8,806) Gsis
c
806   format(' Total     ',f10.2//)
c
c
      if(igam2.eq.0) write(8,808)
808   format(' There were numerical problem with the calculation of'/
     *' gamma shield, calculations skipped'/)
      if(ishig.eq.0) write(8,809)
      if(ishin.eq.0) write(8,810)
809   format(/' No need of gamma shield'/)
810   format(/' No need of neutron shield'/)
c
      rr21=rr2
c
c
      write(8,811)  zload,rload,tgam,rr1,rr2,tne,rr21,rr3
c
811   format(
     *'                  DIMENSION OF THE SHIELDINGS'/
     *' LOAD AT Z(M)=',F8.2,' WITH RADIUS(M)=',F8.2/
     *' G Shield Thickness(cm)',f8.2,' With Radius(cm)',2f8.2/
     *' N    "          "     " ',f8.2,'    "      "     " ',2f8.2)
c
```

```
c
      write(8,758) tempv,pin,D
c
758   format(
     *' Pressure Vessel Evaluated at T(K)=',f8.2,' PRESS(MPa)=',f8.2/
     *'                             and Full Power Life(y)',f8.2//)
c
c
      if(ipresv.eq.0) write(8,626)
626   format(' You have problems to contain your reactor, pressure Vesse
     *l'/'calculations skipped, not included in shielding or balance'//)
c
c
      write(8,75) fcor,Gfuel,ffue,Vact,fvc,Gsup,fsup,Vsup,fvs,
     *Gref,fref,Vref,fre
c
75    format('DISTRIBUTION OF MASSES AND VOLUMES IN THE REACTOR'//
     *' Fraction of Core Occupied by Fuel(%Vol)',f10.3//
     *'          Masses(Kg) * Fractions(%)  Volume(L) * Fraction(%)'/
     *'Fuel Elem',f10.3,f13.3,5x,f10.3,f13.3/
     *'Support E',f10.3,f13.3,5x,f10.3,f13.3/
     *'Reflector',f10.3,f13.3,5x,f10.3,f13.3)
c
c
      if(ipresv.ne.0) write(8,637) Gpv,fmpv,Volpv,fvpv,delpv
637   format('Pr Vessel',f10.3,f13.3,5x,f10.3,f13.3//
     *' Pressure Vessel thickness(cm)',f8.2//)
c
      flush=flufa
      if(ishin.eq.1) flush=fastf
      if(ishig.eq.1.and.igam2.ne.0) flush=flufa*aten
c
c
      write(8,757) Sn,fsh,Req,sigre,D,flufa,flush,fastf,Zload,tne,tne0
c
757   format(
     *'      DETAILS OF THE SHIELDINGS'//
     *       ' Neutron Source(1/sec)',e12.4/' Self Shielding by Reactor
     *Materials',e12.4/' Equivalent R(cm)=',f8.2,' Sigma Removal',f12.6/
     *f6.2,' Years Fast Fluence(1/cm2) without Any Shielding',e12.4
     */    '   "      "     "    "     with (n,gamma)    "  ',e12.4
     */    '   "      "     "    "    Requested             ',e12.4
     */' All  at',f6.2,' meters'
     */'Neutron Shielding Thickness(cm)',f10.2/
     *'Neutron Shielding Thickness if no W is present, cm',f10.2//)
c
c
      if(ishn.eq.1) write(8,767)
767   format(' NEUTRON SHIELDING: LiH-Stainless Steel Matrix'//)
c
c
      if(ishn.eq.2) write(8,768)
768   format(' NEUTRON SHIELDING: B4C'//)
c
c
      write(8,771)
771   format(' TOTAL PRODUCTION OF GAMMA RAYS(1/sec)'//
     *'GROUP    1          2          3          4          5          6
     *    7'/
     *'    0-1Mev       1-2        2-3        3-5        5-7        7-9
     *   >9'/)
c
c
      write(8,772) (sogam(i),i=1,7)
c
772   format(1P7e10.3/)
c
```

```
c
      write(8,*) ' % FRACTION PRODUCED BY U235'
c
c
      write(8,773) (Pr5(i),i=1,7)
c
773   format(7f10.2)
c
c
      write(8,*) ' Equivalent Z for Build up factors'
c
c
      write(8,773) (Zeq(i),i=1,7)
c
c
      write(8,*) ' Gamma Self shielding by Core, Reflector and PV'
c
c
      write(8,774) (shg(i),i=1,7)
c
774   format(7f10.6)
c
c
      write(8,*) ' The same but without Buildup factor'
c
c
      write(8,774) (shgnb(i),i=1,7)
c
c
      write(8,*)'Integrated Dose(Rad) without any shielding(n or gamma)'
c
c
      write(8,772) (Dosewg(i),i=1,7)
c
c
      if(ishig.eq.0.and.ishin.eq.1)
     *write(8,*)'Integrated Dose  Neutrons Shieldings(and no W)'
c
c
      if(ishig.eq.0.and.ishin.eq.1) write(8,772) (DosenG(i),i=1,7)
c
      if(ishig.eq.1.and.igam2.ne.0)
     *write(8,*)'Integrated Dose with W and Neutrons Shieldings'
c
c
      if(ishig.eq.1.and.igam2.ne.0) write(8,772) (DoseG(i),i=1,7)
c
      write(8,775) Dosew
c
775   Format(' Integrated Dose without Any Shielding,Total(Rad)',e12.4)
c
c
      if(ishig.eq.1.and.igam2.ne.0) Dosos=Doses
      if(ishig.eq.0) Dosos=Dosen
      write(8,776) Dosos,tgam,tgam0
c
776   format(' Integrated dose with Shieldings(W and Neutrons), Total (R
     *ads)',e12.2/
     *' Tungsten Thickness(cm)',f10.2/
     *' Tungsten Thickness(cm)',f10.2,' if no Neutron Shield Present'//)
c
c
      write(8,804) Dose,Zload
c
804   format(/' Requested Dose(Rad)',f10.2,' at ',f6.2,' meters'/)
c
c  SOLVE THERMALHYDRAULIC
```

```
c
      norif=nh*nf
c
c
      write(8,2367) tout
c
2367  format(//' THERMALHYDRAULICS CALCULATIONS'/
     *         ' Average Temperature of the Coolant at Core Exit(K)',f10
     *.2//)
c
c
      if(iboun.eq.0) write(8,3478)
3478  format(/' BOUNDARY CONDITION: INPUT FLOW'//)
c
c
      if(iboun.eq.1) write(8,3479)
3479  format(/' BOUNDARY CONDITION: OUTLET TEMPERATURE'//)
c
c check if this is a calculation assuming uniform power distribution
c
      if(iopth.gt.2) go to 674
c here if 3D or 2D power density is constant
      Pfacr=1.
      Pfacaz=1.
      Call Thermohe
c program end
      go to 677
674   continue
c here if we have uniform loading ie spacial dependent power density
      if(iouth.eq.2) go to 671
c
c Thermo at Max power density
c
      Pfacaz=1.
c
c
      write(8,687)
c
687   format(' ****PROFILE OF CHANNEL WITH MAXIMUM POWER DENSITY****'//)
c
      Call Inter(sof1,pfr,isof1,sofa,Pfacr)
      Call Thermohe
      if(iprob.eq.1) stop
671   continue
      if(iouth.eq.1) go to 677
c
c Thermo at Min power density
c
      Pfacaz=1.-Ripple
c
c
      write(8,697)
c
697   format(///'***PROFILE OF CHANNEL WITH MINIMUM POWER DENSITY***'//)
c
      Call Inter(sof1,pfrm,isof1,sofa,Pfacr)
      Call Thermohe
677   continue
      Stop
      End
c
c He density(kg/m3) function of p(Pa) and t(K)
c from page 405, ANS book "Thermal and Flow Design of Helium-Cooled
c Reactors",G. Melese and R. Katz
c
      Function rho(p,t)
      R=2077.22
```

```
      rho=p/(r*t+p*b(t))
      return
      end
c
c Function b(t)(used by rho)
c
      Function b(t)
      data c1,c2,c3,c4,c5/9.489433e-4,9.528079e-4,3.420680e-2,
     *                    2.739470e-3,9.409120e-4/
      b=c1+c2/(1.-c3*t)+c4/(1.+c5*t)
      return
      end
c
c Subroutine: Interpolate Table
c
      Subroutine Inter(x,y,np,xi,yi)
      Dimension       x(20),y(20)
      if(xi.lt.x(1)) go to 10
      if(xi.ge.x(np)) go to 20
      i=1
11    continue
      if(x(i).gt.xi) go to 12
      i=i+1
      go to 11
12    continue
      go to 30
10    i=2
      go to 30
20    i=np
30    yi=y(i-1)+((y(i)-y(i-1))/(x(i)-x(i-1)))*(xi-x(i-1))
      return
      end
c
c  Compute available energy or
c  difference between available and requested energies
c  as function of S/F(other thing too)
c
      Function Balance(ibu,sofa)
c
c For input S/F=sofa and when itest=0:
c this function calculate the volume of the core that satisfies the
c reactivity balance; and then the energy balance(Mwday)
c Balance=Eavailable-Erequested. If ishape=2 or 3 (input R or H) the
c function checks if R(or) H is compatible with reactivity balance
c if not, set flag ibu=1 and RETURN
c
c if itest=1 and for input sofa calculate
c available energy
c radius infinite core that satisfies reactivity balance if ishape=2
c height of infinite "   "       "          "        "    "    "  =3
c
c
      common /Bala/ Ereq,buwo(5),relr,rela1,rela2,dkeol,dkstr,BU,Vcor,
     *U5m,dkbu,dkxes,dkxep,dksms,dksmp,dkbe,rkre,eps,itmax,dcr
     *,itest,aofi
c
      common /MaBaSi/ R,H,ishape
c
      common /Fluxo/ Pow,Vc,sofr,fluxn(4),xedks,xedkt,tmax,smdks,smdkt
     *,tsm,D,iwxe
c
      dimension rm(8),sof(8),sof1(5),aof(5),bew(5)
     *,rki(8)
c
      data (rm(i),i=1,8)/17.40,14.86,13.71,11.65,10.41,9.499,8.958,8.76/
c
c rm (cm) migration length of mixture with   500 gU235/L fuel element
```

```
c for the following Support/Fuel element ratio
c
      data (sof(i),i=1,8)/0.0,0.1,0.2,0.5,1.0,2.,3.,4./
c
      data (rki(i),i=1,8)/1.62228,1.58272,1.59454,1.62315,1.67140,
     *                    1.67186,1.62773,1.56044/
c
c rki k-infinite as function of S/F
c
      data (aof(i),i=1,5)/1.372,1.407,1.373,1.325,1.276/
c
c aof: absorption to fission ratio in U235 for the following S/F ratios
c
      data (sof1(i),i=1,5)/0.0,0.2,0.5,1.0,2.0/
c
c sof1: Support/Fuel ratio
c
      data (bew(i),i=1,5)/30.14,27.47,27.24,26.40,23.80/
c
c bew(): % delta k; reactivity worth of 30 cm Be radial reflector as
c function of S/F (sof1(i),i=1,5)
c
      data dext,rofu /2.29,0.5/
c
c dext(cm) extrapolation length for bare core
c rofu (g/cm3) density of U235 corresponding to data base
c
      data pee,xop,pisq3/3.1415926,0.541315,5.441398/
c
c xop:optimum shape for cylindrical reactors=Radius/Height
c pisq3=pi*sqrt(3)
c
      sofr=sofa
      ibu=0
      isof=8
      isof1=5
      ibup=4
      ibe=8
      sofo=sofa
c
c compute kinf and migration length
c
      call Inter(sof,rki,isof,sofo,rkin)
      call Inter(sof,rm,isof,sofo,rmi)
c
c compute A/F for U235 at S/F=sofo
c
      call Inter(sof1,aof,isof1,sofo,aofi)
c
c compute delta k BU
c
      call Inter(sof1,buwo,isof1,sofo,dkbu)
c
c compute delta k of Be reflectors
c
      call Inter(sof1,bew,isof1,sofo,wbe)
      wber=wbe*relr
      wbea1=wbe*rela1
      wbea2=wbe*rela2
      dkbe=wber+wbea1+wbea2
c
c Reactivity effects of Xe and Sm
c Simultaneous solution of dk(Xe,Sm)=g(Vc) and Vc=f(dk(Xe,Sm))
c bisect function h=g-h where h is inverse f
c
      it=0
      rkrem=1.+(DKBU+dkstr+dkeol-dkbe)/100.
```

```
c buck buck(0.Xe)
      buck=sqrt(rkin/rkrem-1.)/rmi
      go to (2001,2002,2003),ishape
c
c if ishape=1, optimum shape----->2001
2001  hp=pisq3/buck
      rp=xop*hp
      R=rp-dext
      H=hp-dext*2.
      Vc1=pee*R**2*H
      Vc=Vc1
      Call Flux
      Fu1=xedkt+smdkt
      Vc2=Vc1
728   Vc2=Vc2+Vc1
      it=it+1
      H=(Vc2/(pee*xop**2))**0.33333333
      R=xop*H
      bge=(2.405/(R+dext))**2+(pee/(H+2.*dext))**2
      h2=rkin/(1.+rmi**2*bge)-rkrem
      h2=100.*h2
      Vc=Vc2
      Call Flux
      Fu2=xedkt+smdkt-h2
      if((Fu1*Fu2).gt.0..and.it.le.itmax) go to 728
      if(it.lt.itmax) go to 739
      write(8,201)
      stop
739   continue
748   Vca=(Vc2+Vc1)*0.5
      it=it+1
      H=(Vca/(pee*xop**2))**0.33333333
      R=xop*H
      bge=(2.405/(R+dext))**2+(pee/(H+2.*dext))**2
      ha=rkin/(1.+rmi**2*bge)-rkrem
      ha=100.*ha
      Vc=Vca
      Call Flux
      Fua=xedkt+smdkt-ha
      call bisect(Vc1,Vc2,Vca,Fu1,Fu2,Fua,iwa)
      test=abs(Vc1-Vca)/Vca
      if(test.gt.eps.and.iwa.eq.0) go to 748
      if(it.lt.itmax.and.iwa.eq.0) go to 749
      write(8,201)
      stop
749   continue
c
c End iteration for the case optimum shape.RESULT: Volume of Core(Vc)
c that satisfies reactivity balance
c
      go to 2004
c
c go next if ishape=2----->input Radius of core
c
2002  buca=buck**2-(2.405/(R+dext))**2
      dcr=2.405/buck-dext
      if(buca.le.0.) ibu=1
c
c set flag ibu=1 if R(input) is too small to satisfy reactivity balance
c
      if(ibu.eq.1) return
      hp=pee/sqrt(buca)
      H=hp-2.*dext
      Vc1=pee*R**2*H
      Vc=Vc1
      Call Flux
      Fu1=xedkt+smdkt
```

```
      Vc2=Vc1
828   Vc2=Vc2+Vc1
      it=it+1
      H=Vc2/(pee*R**2)
      bge=(2.405/(R+dext))**2+(pee/(H+2.*dext))**2
      h2=rkin/(1.+rmi**2*bge)-rkrem
      h2=100.*h2
      Vc=Vc2
      Call Flux
      Fu2=xedkt+smdkt-h2
      if((Fu1*Fu2).gt.0..and.it.le.itmax) go to 828
      if(it.lt.itmax) go to 839
      write(8,201)
      stop
839   continue
848   Vca=(Vc2+Vc1)*0.5
      it=it+1
      H=Vca/(pee*R**2)
      bge=(2.405/(R+dext))**2+(pee/(H+2.*dext))**2
      ha=rkin/(1.+rmi**2*bge)-rkrem
      ha=ha*100.
      Vc=Vca
      Call Flux
      Fua=xedkt+smdkt-ha
      call bisect(Vc1,Vc2,Vca,Fu1,Fu2,Fua,iwa)
      test=abs(Vc1-Vca)/Vca
      if(test.gt.eps.and.iwa.eq.0) go to 848
      if(it.lt.itmax.and.iwa.eq.0) go to 849
      write(8,201)
      stop
849   continue
c
c End iteration for the case of input Radius.RESULT: Volume of Core(Vc)
c that satisfies reactivity balance
c
      go to 2004
c
c go next if ishape=3------->input Height of the core
c
2003  bucr=buck**2-(pee/(H+2.*dext))**2
      dcr=pee/buck-dext*2.
      if(bucr.le.0.) ibu=1
c
c set flag ibu=1 if H(input) is too small to satisfy reactivity balance
c
      if(ibu.eq.1) return
      rp=2.405/sqrt(bucr)
      R=rp-dext
      Vc1=pee*R**2*H
      Vc=Vc1
      Call Flux
      Fu1=xedkt+smdkt
      Vc2=Vc1
928   Vc2=Vc2+Vc1
      it=it+1
      R=sqrt(Vc2/(pee*H))
      bge=(2.405/(R+dext))**2+(pee/(H+2.*dext))**2
      h2=rkin/(1.+rmi**2*bge)-rkrem
      h2=h2*100.
      Vc=Vc2
      Call Flux
      Fu2=xedkt+smdkt-h2
      if((Fu1*Fu2).gt.0..and.it.le.itmax) go to 928
      if(it.lt.itmax) go to 939
      write(8,201)
      stop
939   continue
```

```
948   Vca=(Vc2+Vc1)*0.5
      it=it+1
      R=sqrt(Vca/(pee*H))
      bge=(2.405/(R+dext))**2+(pee/(H+2.*dext))**2
      ha=rkin/(1.+rmi**2*bge)-rkrem
      ha=100.*ha
      Vc=Vca
      Call Flux
      Fua=xedkt+smdkt-ha
      call bisect(Vc1,Vc2,Vca,Fu1,Fu2,Fua,iwa)
      test=abs(Vc1-Vca)/Vca
      if(test.gt.eps.and.iwa.eq.0) go to 948
      if(it.lt.itmax.and.iwa.eq.0) go to 949
c
c
      write(8,201)
c
201   format(/' stop in Balance, trying to compute Xe, Sm'/)
      stop
c
949   continue
c
c End iteration for the case of input Height.RESULT: Volume of Core(Vc)
c that satisfies reactivity balance
c
2004  continue
c
c End of iterations to obtain Volume of core that satisfies reactivity
c balance (that includes Xe and Sm reactivities)
c
      ibu=0
      dkxes=xedks
      dkxep=xedkt
      dksms=smdks
      dksmp=smdkt
c
c recompute required dk from bare core
c
      rkre=1.+(DKBU+dkxep+dksmp+dkstr+dkeol-dkbe)/100.
c
c check available energy
c
      Vcor=Vc
      Vact=Vcor/(1.+sofa)
      U5m=Vact*rofu
c
c for U235 at 200 Mev/fission
c
      Eavai=(U5m*BU)/(105.2*aofi)
      Balance=Eavai-Ereq
      if(itest.eq.1) Balance=Eavai
      return
      end
c
c
      subroutine bisect(x1,x2,xav,f1,f2,fav,iwa)
      if((f1*f2).le.0.) go to 11
      write(8,1)
1     format(//' f1*f2 >0.'//)
      iwa=1
      return
11    iwa=0
      if((fav*f1).le.0.) go to 10
      f1=fav
      x1=xav
      return
10    f2=fav
```

```
      x2=xav
      return
      end
c
c compute control drums
c
      subroutine Control
      dimension cr(5),sof(5)
      common /cont/ R,drr,dref,rasp,refec,ncr,dksu,dkstr,relr
c
c R:radius core, drr:thickness reflector; dref: reference reflector
c thickness(with which worths were calculated);rasp:S/F element ratio
c refec:keff to control; ncr:number of control rods;dksu(%):subcritical-
c lity required for drums in.
c dkstr(%): dk structural part of reactor,relr:relative worth of reflector
c
      data (cr(i),i=1,5)/21.0,19.15,18.8,18.1,16.35/
c
c cr: %delta k; max worth of control rods in Be reflector
c calculated with reflector thickness dref(reference)
c
      data (sof(i),i=1,5)/0.0,0.2,0.5,1.0,2.0/
      if(ncr.eq.0) write(8,48)
48    format(' Your input for the number of control drums is zero'/
     *' Subroutine Control bypassed'//)
      if(ncr.eq.0) return
c
      pee=3.1415926
      isof=5
      Call Inter(sof,cr,isof,rasp,cdr)
c
c worth of 2*pee sheet(cdr) is corrected because drr might not be =dref
c correction factor relative worth of reflector(with respect to dref)
      cdr=cdr*relr
c rea(%) reactivity to control
      rea=(refec-1.)*100.+dksu-dkstr
c
c compute maximum number of drums=nmax
c
      rd=0.5*drr
      par=2.*asin(rd/(R+rd))
      pard=par*180./pee
c
c rd and par are the radius and the central parallax of the drums
c
      nmax=2.*pee/par
      cdrm=cdr*nmax*par/(2.*pee)
      if(cdrm.ge.rea) go to 10
      write(8,20) nmax,cdrm,rea,pard
20    format(//' You cannot control this reactor only with control'/
     *          ' drums in the reflector'/
     *' The reactivity worth of',i5,' drums(max value) is',f6.2,' %'/
     *'                    and the reactivity to control is',f6.2,' %'/
     *' Parallax of each drum would be',f8.2,' degrees'//)
      return
10    continue
      if(ncr.gt.nmax) write(8,33) ncr,nmax
33    format(//' Your input for the number of drums is',i5/
     *' larger than the max possible',i5,' continue with max'//)
      if(ncr.gt.nmax) ncr=nmax
      rcr=ncr
      cdrm=cdr*rcr*par/(2.*pee)
      if(cdrm.lt.rea) write(8,34) ncr,nmax
34    format(//' Your input for the number of drums',i5,' is too small'/
     *'(max possible is',i5,')program is going to increase it')
      if(cdrm.lt.rea) then
101   ncr=ncr+1
```

```
      cdrm=cdr*ncr*par/(2.*pee)
      if(cdrm.lt.rea) go to 101
      rcr=ncr
      else
      endif
c
c para is the central parallax of the absorbing part of one drum
c
      para=(rea/rcr)*(2.*pee/cdr)
c pass to local parallax(seen from center of drum)---> dtcr
      bb=(R+rd)*cos(0.5*para)
      cc=R**2+2.*R*rd
      xx=bb-sqrt(bb**2-cc)
      dtcr=2.*asin(xx*sin(0.5*para)/rd)
      dtcd=180.*dtcr/pee
      write(8,31) ncr,dtcd,rea
31    format(//i5,' Drums with absorbing angle(d)',f8.2,' each'/
     *          ' Control dk(%)=',f8.2//)
      return
      end
c
c Four groups cross section to compute fluxes,Xe and Sm Effects
c
      Subroutine Flux
      common /Fluxo/ Pow,Vc,sofr,fluxn(4),xedks,xedkt,tmax,smdks,smdkt,
     *tsm,D,iwxe
      dimension aux(5),spec(4,5),f5(4,5),sif(4),sp(4),sof(5)
c
      data (sof(i),i=1,5) /0.0,0.2,0.5,1.0,2.0/
c
c Neutron spectra as function of S/F and Neutron Energy
c
      data (spec(i,1),i=1,4) /0.641,0.348,0.009,0.002/
      data (spec(i,2),i=1,4) /0.611,0.335,0.038,0.015/
      data (spec(i,3),i=1,4) /0.581,0.321,0.061,0.037/
      data (spec(i,4),i=1,4) /0.553,0.305,0.078,0.063/
      data (spec(i,5),i=1,4) /0.519,0.288,0.090,0.102/
c
c U235 fission cross sections as function of S/f and Energy
c averaged in core volume(supp+fuel)
c
      data (f5(i,1),i=1,4) /1.28,4.92,37.33,25.55/
      data (f5(i,2),i=1,4) /1.27,6.08,37.33,62.13/
      data (f5(i,3),i=1,4) /1.27,6.45,37.33,90.85/
      data (f5(i,4),i=1,4) /1.27,6.63,37.33,121.5/
      data (f5(i,5),i=1,4) /1.27,6.74,37.3,157.4/
c
      data rou5,A5,Av,wfi,xe23,xe4,gi,gx,xel,rnu/0.5,235.044,0.602202,
     *3.2042e-11,623.,1.126e+6,0.061,0.003,2.09e-5,2.44/
c
c wfi: joules/fission, xe23,xe4: capture cross section Xe
c gi,gx: Yield I135 and Xe135 per fission, rnu=number neutrons/fission
c
      data rli,pml,sm23,sm4,gp/2.87e-5,3.56e-6,278.,29982.,0.0113/
c
c xel,rli,pml: decay constants Xe135,I135 and Pm149
c gp:Yield Pm149/fission, sm23,sm4: capture cross section Sm149
c
      Vact=Vc/(1.+sofr)
      ro5=rou5*Vact/Vc
c ro5 is U235 density average in core(support+fuel elements)
      u5n=ro5*Av/A5
      isof=5
      ng=4
      do 10 i=1,ng
      do 20 j=1,isof
20    aux(j)=spec(i,j)
```

```
      call Inter(sof,aux,isof,sofr,spe)
      sp(i)=spe
      do 30 j=1,isof
30    aux(j)=f5(i,j)
      call Inter(sof,aux,isof,sofr,sig)
      sif(i)=sig
10    continue
      sigf=0.
      do 40 i=1,ng
40    sigf=sigf+sp(i)*sif(i)
      sigf=sigf*u5n
      flun=Pow*1.e+6/(Vc*wfi*sigf)
c flun :average flux in core
      do 50 i=1,ng
50    fluxn(i)=flun*sp(i)
      xec=xe23*(sp(2)+sp(3))+xe4*sp(4)
      fluxe=xel*1.e+24/xec
      xedks=100.*(gi+gx)*flun/(rnu*(flun+fluxe))
c
c compute max xe after trip(iwxe=0)
c if no max appears set flag iwxe=1, and make xedkt=xedks
c
      iwxe=0
      flumi=fluxe*gx/gi
      xedkt=xedks
      if(flun.le.flumi) iwxe=1
      if(iwxe.eq.1) go to 1910
      fluxi=rli*1.e+24/xec
      tmax=1.+(gi+gx)*(fluxi-fluxe)/(gi*(fluxe+flun))
      tmax=alog(rli/(xel*tmax))/(rli-xel)
      xedkt=exp(-xel*tmax)*(gi+gx)*flun/(fluxe+flun)+
     1     (exp(-xel*tmax)-exp(-rli*tmax))*gi*flun/(fluxi-fluxe)
      xedkt=xedkt*100./rnu
      tmax=tmax/3600.
1910  continue
      smc=sm23*(sp(2)+sp(3))+sm4*sp(4)
      smc=smc*1.e-24
      flusm=pml/smc
      tsm=1./(smc*flun)
      tvida=D*3.15576e+7
c
c compute sm after trip
c
      tsmo=tvida/tsm
      etsm=0.
      if(tsmo.lt.50.) etsm=exp(-tsmo)
      smdks=gp*(1.-etsm)/rnu
      smdkt=smdks+gp*flun/(rnu*flusm)
      smdks=100.*smdks
      smdkt=100.*smdkt
      tsm=tsm/86400.
      return
      end
c
c solve thermalhydraulic
c
      subroutine thermohe
      real*4 k
c
      dimension pow(51),pres(51),tbulk(51),ve(51),rmach(51),z(51),rj(51)
     *,tw(51),Tclad(51),Tfuel(51),Tclaa(51),Tfua(51)
c
c
      common/thermo/ aln,nnodo,iexpl,ifri,itmax,norif,iditus,nfu,pin,tin
     *,d,ru,xi,der,eps,Powt,rjs,flowt,GU235,Vact,Enr,PUC,Rouc,RoZrC,
     *RoGraf,dclad,Hact,Pfacr,Daxi,fact,phav,thav,fe,iprth,iopth,iouth
     *,Pfacaz,itry,flowal,iboun
```

```
      common /fGu/ ifro,deri,pf,ar,dho,rug,dz
      common /fphe/ imaxt,epsi,dero
      data pee/3.1415926/,rgas/4124.2/
      character*80 aln
c
c nnode: number of nodes; iexpl /0,1/ No, Yes explicit
c ifri.ne.1 : switch off friction factor
c itmax: max iters, norif: number of orifices, iditus/0,1/ no Dittus
c( means  correction for Twall/Tbulk in Nussel correlation),1 Yes Dittus
c nfu: number of nodes inside fuel
c
      itry=0
      iexplo=iexpl
2013  dero=der
      ifro=ifri
      deri=der
      dho=d*0.01
      rug=ru
      imaxt=itmax
      epsi=eps
c
c pin(MPa), tin( K) inlet conditions
c d(cm), ru : channel diameter and rugosity
c xi /0.,1./ xi=0. fully explicit; xi=1. fully implicit
c der: factor to multiply variable to compute derivative numerically
c typical value 1.01; eps: tolerance to finish iterations; Powt(Mw)
c total power; rjs(Kw/cm2) scale for plot heat rate
c
c
c
c flowt(Kg/sec) :  Total flow
c flow(g/sec)   :  Channel flow
c
      flow=1000.*flowt/norif
      Vamix=Vact*fe
c
c Vact:Volume of fuel elements; Vamix: volume occupied by UC-ZrC mix in
c graphite matrix; exclude coolant holes and clads
c
c GU235 grams of U235 in  active mix Volume Vamix(Liters) U is
c Enriched (at fraction), PUC mole fraction of UC in UC-ZrC mix
c RoUC, RoZrC and RoGraf (g/cm3) densities of UC, Zrc and Graphite
c dclad(cm): clad thickness
c
      RaUC=(1.+(1./Enr-1.)*1.012798+0.05110173/Enr)*GU235/(1000.*Vamix
     *)
      RaZrc=(1./PUC-1.)*RaUC*0.4178467
      xUC=RaUC/RoUC
      xZrC=RaZrC/RoZrC
      xGraf=1.-xUC-xZrC
      RaGraf=xGraf*RoGraf
      Ra=RaGraf+RaZrC+RaUC
c
      npu=nnodo+1
c  Hact(cm):active length; Pfact:Radial Peaking factor at r position
c  assume cos in z and  Daxi(cm) axial extrapol.
      dzet=Hact/nnodo
      roav=Powt*fact/Vamix
c Pfraz : Power factor radial-azimuthal
      Pfraz=1.
      if(iopth.gt.2) Pfraz=Pfacr*Pfacaz
c
c recompute flow  according to option
c
      if(iopth.eq.4) flow=flow*Pfraz
      Hext=Hact+2.*Daxi
      we=Hext/Hact
```


C-2.

```
      rih=pee/Hext
      Pfact=Pfraz*pee/(2.*we*sin(0.5*pee/we))
      do 182 i=1,npu
      zzzz=Daxi+(i-1)*dzet
      if(iopth.gt.1)  pow(i)=roav*Pfact*sin(rih*zzzz)
      if(iopth.eq.1)  pow(i)=roav
182   z(i)=zzzz-Daxi
c
c See: pow is power per unit volume of active mix(UC-ZrC-Graphite)
c transform pow(i)(Mw/L) to power per unit length (watt/cm)
c
      r0=d/2.
      r1=r0+dclad
      r2=sqrt(r1**2+Vamix*1000./(pee*Hact*norif))
      facto=1000.*pee*(r2**2-r1**2)
      do 123 i=1,npu
123   pow(i)=pow(i)*facto
c
c z(i) (cm) boundaries of the nodes
c
      write(8,12) aln
12    format(a80)
      write(8,21) pin,tin,d,ru,der,eps,Powt,norif,flowt,fact
21    format(//' Thermodynamic profile of a He cooled channel'/
     *' Inlet pressure(MPa)=',f10.5,' Inlet temperature( K)=',f10.2/
     *' Diameter(cm)=',f7.2,' Rugosity=',f8.5,' der parm=',f6.4,
     *' eps=',f10.7/' Total Power   (Mw)',f10.3,' No. Orifices',i10/
     *              ' Total Flow(Kg/sec)',f10.3/
     *' Factor that multiplies Power',f10.7//)
      write(8,162)roav,Hact
162   format(' Average Power Density(Mw/L of active mix)',f8.4,
     *' Channel Length(cm)',f7.2/)
      if(iopth.eq.1) write(8,1264)
1264  format(' 3-D UNIFORM POWER DISTRIBUTION ASSUMED'/)
      if(iopth.eq.2) write(8,1266)
1266  format(' 2-D (R-THETA) UNIFORM POWER DISTRIBUTION ASSUMED'/)
      if(iopth.ge.2)  write(8,164) Pfact,Daxi
164   format(' Cosine  Axial Power Distribution=roav*Pfact*Cos(Pi*z/H)'/
     *' Pfact(Axial*Radial*Azim)=',f7.4,' Extrapolated Delta(cm)',f7.2)
c
      if(iexpl.eq.1.or.xi.eq.0.) write(8,22)
22    format(//' Attention Fully Explicit Calculation!!'//)
c
      if(iexpl.ne.1.and.xi.ne.0.) write(8,122) xi
122   format(//' Attention Implicit Calculation!!  xi=',f10.5//)
c
      if(ifri.ne.1) write(8,23)
23    format('  friction was assumed zero'//)
c
      if(iditus.eq.1) write(8,24)
24    format(' Dittus-Boelter Correlation for Nu (No Twall/Tbulk correct
     *ion)')
      if(iditus.eq.0) write(8,376)
376   format(' Twall/Tbulk correction to Dittus-Boelter Correlation for
     *Nu')
c
      pres(1)=pin*10.**6
      tbulk(1)=tin
      ar=pee*(dho/2.)**2
      pf=pee*dho
c
      if(iopth.eq.4) write(8,1293)
1293  format(' Coolant Flow Proportional to Power Density'/)
      if(iopth.ne.4) write(8,1295)
1295  format(' Uniform Coolant Flow Assumed'/)
      write(8,31) flow
31    format(' Flow(g/s)=',f10.3//
```

```
     *' Node     z(cm)    Po(w/cm)    Pr(MPa)       T(K)     V(m/s)        Re
     *      Mach'//)
c
      g=flow*0.001/ar
      flow=flow*0.001
3001  p=pres(1)
      t=tbulk(1)
      call phe(h,p,t,rho,vis,cp,pr,a,k,x,0)
      pi=p
      ti=t
      rhoi=rho
      hi=h
      Re=dho*g/vis
      Rei=Re
      ve(1)=flow/(ar*rho)
c remember for future use inlet conditions
      hinlet=h
      echannel=0.
      rmach(1)=ve(1)/a
      i=1
      pre=pres(1)*10.**(-6)
      write(8,41) i,z(i),pow(1),pre,tbulk(i),ve(i),Re,rmach(1)
      f=0.
c branching: if fully explicit continue, if not go to 2001
      if(iexpl.eq.0.and.xi.gt.0.) go to 2001
      do 30 i=2,npu
      dz=z(i)-z(i-1)
      po= 0.5*(pow(i)+pow(i-1))*dz
      echannel=echannel+po
      h=hi+po/flow
      p=pi
      call phe(h,p,t,rho,vis,cp,pr,a,k,x,1)
      tbulk(i)=t
      d1=drdp(pi,ti,der)
      d2=drdt(pi,ti,der)
      if(ifri.eq.1) f=fri(Rei,ru)
      f1=1.-d1*(g/rhoi)**2
      f2=(f*pf/(2.*ar*rhoi))*g*g
      f3=-d2*(g/rhoi)**2
      dz=0.01*dz
      pres(i)=pres(i-1)-f2*dz/f1-f3*(tbulk(i)-tbulk(i-1))/f1
      p=pres(i)
      t=tbulk(i)
      call phe(h,p,t,rho,vis,cp,pr,a,k,x,0)
      if(i.eq.npu) houtlet=h
      Re=dho*g/vis
      ve(i)=flow/(ar*rho)
      rmach(i)=ve(i)/a
      pre=pres(i)*10.**(-6)
      pi=p
      ti=t
      rhoi=rho
      hi=h
      Rei=Re
      if(iprth.eq.1.or.i.eq.npu)
     *write(8,41) i,z(i),pow(i),pre,tbulk(i),ve(i),Re,rmach(i)
30    continue
      gamma=cp/(cp-rgas)
41    format(i5,5f10.2,f10.0,f10.3)
c end calculation fully explicit
      go to 4321
2001  continue
c
c here begins the implicit calculation
c
      do 531 i=2,npu
      dz=z(i)-z(i-1)
```

```
      po= 0.5*(pow(i)+pow(i-1))*dz
      echannel=echannel+po
      po=po/flow
c
c First find an estimation of toutlet  explicitly
c
      h=hi+po
      p=pi
      call phe(h,p,t,rho,vis,cp,pr,a,k,x,1)
c
c find where Gu changes sign as function p outlet node
c
      taprox=t
      dz=0.01*dz
      p1=pi
      it=0
      Gu1=Gu(pi,taprox,pi,ti,xi,g)
      p2=pi
42    p2=p2/der
      it=it+1
      Gu2=Gu(p2,taprox,pi,ti,xi,g)
      if((Gu1*Gu2).gt.0..and.it.le.itmax) go to 42
      if(it.gt.itmax) then
      write(8,43) i,p1,Gu1,p2,Gu2
43    format(//' implicit calculation, problem trying to find p2 in node
     *',i3,' p1 Gu1=',2e14.6//' p2 Gu2=',2e14.6/' go to explicit'//)
      iexpl=1
      go to 3001
      else
      endif
      it=0
c
c start bisect to find p at outlet node
c
45    pav=0.5*(p1+p2)
      Guav=Gu(pav,taprox,pi,ti,xi,g)
      test=abs(Guav)/p
      if(test.lt.eps) go to 44
      it=it+1
      call bisect(p1,p2,pav,Gu1,Gu2,Guav,iwa)
      if(it.le.itmax.and.iwa.eq.0) go to 45
44    continue
      if(it.gt.itmax.or.iwa.eq.1) then
      write(8,46) i,p1,Gu1,p2,Gu2,it,iwa
46    format(//' implicit calculation, problem trying bisect Gu  in node
     *',i3,' p1 Gu1=',2e14.6//' p2 Gu2=',2e14.6,' its=',i4,' iwa=',i4/
     *' go to explicit'//)
      iexpl=1
      go to 3001
      else
      endif
c
c we have here pav,tprox values which are good estimations.
c let us improved it by Taylor expansions of Fu(p,t) and Gu(p,t)
c
      vi=ve(i-1)
      Fu1=Fu(pi,ti,pav,taprox,po,vi)
      pprima=der*pav
      Fu2=Fu(pi,ti,pprima,taprox,po,vi)
      dfup=(Fu2-Fu1)/(pprima-pav)
      tprima=der*taprox
      Fu2=Fu(pi,ti,pav,tprima,po,vi)
      dfut=(Fu2-Fu1)/(tprima-taprox)
      Gu1=Gu(pav,taprox,pi,ti,xi,g)
      Gu2=Gu(pprima,taprox,pi,ti,xi,g)
      dgup=(Gu2-Gu1)/(pprima-pav)
      Gu2=Gu(pav,tprima,pi,ti,xi,g)
```

```
      dgut=(Gu2-Gu1)/(tprima-taprox)
      delta=dfup*dgut-dgup*dfut
      delp=(-Ful*dgut+Gul*dfut)/delta
      delt=(-Gul*dfup+Ful*dgup)/delta
c     write(8,3478) ful,dfup,dfut,gul,dgup,dgut,delta,delp,delt
3478  format(' Ful Df/Dp Df/dt',3e14.5/' Gul Dg/Dp Dg/Dt',3e14.5/
     *' delta delp delt',3e14.5/)
      p=pav+delp
      t=taprox+delt
      call phe(h,p,t,rho,vis,cp,pr,a,k,x,0)
      if(i.eq.npu) houtlet=h
      pre=p*10.**(-6)
      pres(i)=p
      tbulk(i)=t
      ve(i)=flow/(ar*rho)
      Re=dho*g/vis
      rmach(i)=ve(i)/a
      if(iprth.eq.1.or.i.eq.npu)
     *write(8,41) i,z(i),pow(i),pre,tbulk(i),ve(i),Re,rmach(i)
      hi=h
      pi=p
      ti=t
531   continue
4321  continue
c
c End calculation coolant conditions
c
c Check if we did not pass speed of sound anywhere , if yes change
c condition at inlet and set flag iprob=1
c
      iprob=0
      do 2297 i=2,npu
      if(pres(i).gt.pres(i-1).or.pres(i).lt.0.) iprob=1
2297  continue
      if(iprob.eq.0) go to 2303
c
c Hoping to fix the mess we change input conditions in order to relax
c thermalhydraulic
c
      if(itry.eq.1) write(8,2308)
2308  format(' No success with the change. Program Ends')
      if(itry.eq.1) return
      ifri=0
      Pfacr=1.
      Pfacaz=1.
      iopth=1
      pin=10.
      tin=20.
      itry=1
      iexpl=iexplo
      if(iboun.ne.0) flowt=flowal
      write(8,2314) pin,tin
2314  format(//' With previous power level and input conditions'/
     *'flow velocities reach speed of sound. To relax thermalhydraulic'/
     *'we change to following condtions:'/
     *'                                1) Uniform 3D power density.'/
     *'                                2) Assume no friction in channel.'/
     *'                                3) Inlet p(MPa)=',f8.2/
     *'                                4) Inlet T(K)  =',f8.2//)
      go to 2013
c
c Beginning calculation wall temperature
c
2303  iwar=0
      do 302 i=1,npu
      p=pres(i)
      t=tbulk(i)
```

```
      ti=t
      call phe(h,p,t,rho,vis,cp,pr,a,k,x,0)
      Rei=dho*g/vis
      Pri=Pr
      Rki=k/dho
      h=h+0.5*ve(i)**2
      call phe(h,p,t,rho,vis,cp,pr,a,k,x,1)
      Tfs=t
c Apply recovery factor (Rc) to stagnation temperature Tfs
      Rc=(Pri)**0.33333333
      Tfa=ti+Rc*(Tfs-ti)
c First estimation wall temperature
      Zr=z(i)/(100.*dho)
      RNus=RNu(Rei,Pri,1.,Zr)
      hco=RNus*Rki
      rjq=pow(i)*100./(pee*dho)
      rj(i)=rjq*10.**(-7)
      Tw1=Tfa+rjq/hco
      Tw(i)=Tw1
      if(iditus.eq.1) go to 3121
c     write(8,143) Tw1,rjq,hco,Rc,Tfa,Tfs
      it=0
      Tr1=Tw1/ti
      Fq1=Fq(rjq,Zr,Rei,Pri,Rki,Tfa,Tr1,Tw1)
c     write(8,144) Tw1,Fq1
      Tw2=Tw1
422   Tw2=Tw2*der
      Tr2=Tw2/ti
      it=it+1
      Fq2=Fq(rjq,Zr,Rei,Pri,Rki,Tfa,Tr2,Tw2)
c     write(8,144) Tw2,Fq2
      if((Fq1*Fq2).gt.0..and.it.le.itmax) go to 422
      if(it.gt.itmax) then
      write(8,431) i,Tw1,Fq1,Tw2,Fq2
143   format(' first estimation, Tw1=',f10.3,' Rjq hc',2e14.5/
     *' Rc Tfa Tfs',3f14.3/12x,'Tw',12x,'Fq')
144   format(2e14.5)
431   format(//' implicit calculation, problem trying to find Tw2in node
     *',i3,'Tw1 Fq1=',2e14.6//'Tw2 Fq2=',2e14.6/' bypass wall   '//)
461   format(//' implicit calculation, problem trying bisect Fq  in node
     *',i3,'Tw1 Fq1=',2e14.6//'Tw2 Fq2=',2e14.6,' its=',i4,' iwa=',i4/
     *' by pass wall  '//)
      iditus=1
      iwar=1
      go to 3121
      else
      endif
      it=0
c
c start bisect to find T wall
c
452   Twa=0.5*(Tw1+Tw2)
      Tra=Twa/ti
      Fqav=Fq(rjq,Zr,Rei,Pri,Rki,Tfa,Tra,Twa)
      test=abs(Fqav)/rjq
      if(test.lt.eps) go to 442
      it=it+1
      call bisect(Tw1,Tw2,Twa,Fq1,Fq2,Fqav,iwa)
      if(it.le.itmax.and.iwa.eq.0) go to 452
442   continue
      if(it.gt.itmax.or.iwa.eq.1) then
      write(8,461) i,Tw1,Fq1,Tw2,Fq2,it,iwa
      iditus=1
      iwar=1
      go to 3121
      else
      endif
```

```
      Tw(i)=Twa
c From Twa to Tfuel
c
3121  continue
      if(xGraf.lt.0.) go to 302
      cclad=RkZrC(Twa)
      rjc=rjq*10.**(-4)
      Poa=rjc*2.*r0/r2**2
      Qh=Poa/(1.-(r1/r2)**2)
c See only one node for the clad
c
      Tclad(i)=Tw(i)+rjc*r0*alog(r1/r0)/cclad
      culo=1.+dclad/r0
      Tclaa(i)=Tw(i)+rjc*r0**2*(culo*alog(culo)-(culo-1.))/(dclad*cclad)
      dr=(r2-r1)/nfu
      Tinl=Tclad(i)
      rinl=r1
      Tfa=0.
      do 3021 in=1,nfu
      Cfuel=(RaUC*RkUC(Tinl)+RaZrC*RkZrC(Tinl)+RaGraf*RkGraf(Tinl))/R
     *a
      rout=rinl+dr
      Tout=Tinl+(0.25*Qh*rinl**2/Cfuel)*(2.*(r2/rinl)**2*alog(rout/r
     *inl)+1.-(rout/rinl)**2)
      Tfa=Tfa+(Tout*rout+Tinl*rinl)*dr
      Tinl=Tout
      rinl=rout
3021  continue
      Tfuel(i)=Tout
      Tfua(i)=Tfa/(r2**2-r1**2)
302   continue
      write(8,978) r0,r1,r2,Ra,RaUC,xUC,RaZrC,xZrC,RaGraf,xGraf,nfu
     *,GU235,Vact
978   format(//' From the Coolant Channel to the Uranium Carbide'/
     *' Radius(cm) Orifices, Clad, Cell',3f10.4/
     *' Averages Densities(g/cm3)',10x,' Volumetric Fraction'/
     *'                      Total',f10.4/
     *'                         UC',2f10.4/
     *'                        ZrC',2f10.4/
     *'                   Graphite',2f10.4//' Number of fuel nodes',i4/
     *f10.2,' grams of U235 in',f10.2,' Liters of Fuel Volume')
      if(iwar.eq.1) write(8,427)
427   format(//'Due to iterations difficulties we switch to Dittus Correl
     *ation')
      write(8,976)
976   format(///' Node     z(cm) J(Kw/cm2)  Twall(K)  Tclad(K)  TcladAv
     *  Tfuel(K)  TfuelAv '//)
      i=1
      write(8,977) i,z(i),rj(i),tw(i),Tclad(i),Tclaa(i),Tfuel(i),Tfua(i)
      do 1506 i=2,npu
      if(iprth.eq.1.or.i.eq.npu)
     *write(8,977) i,z(i),rj(i),tw(i),Tclad(i),Tclaa(i),Tfuel(i),Tfua(i)
1506  continue
977   format(i5,f10.2,f10.5,5f10.2)
c
c calculation of axial averages
c See: <T>=<t*rho>/<rho> i.e. <T> is defined according number of He molecules
c
      zav=0
      rohav=0.
      thav=0.
      phav=0.
      twa=0.
      tca=0.
      tfa=0.
      p=pres(1)
      t=tbulk(1)
```

```
      call phe(h,p,t,rho,vis,cp,pr,a,k,x,0)
      roin=rho
      do 1436 i=2,npu
      dz=z(i)-z(i-1)
      zav=zav+dz
      phav=phav+0.5*(pres(i)+pres(i-1))*dz
      twa=twa+0.5*(tw(i)+tw(i-1))*dz
      tca=tca+0.5*(tclaa(i)+tclaa(i-1))*dz
      tfa=tfa+0.5*(Tfua(i)+Tfua(i-1))*dz
      p=pres(i)
      t=tbulk(i)
      call phe(h,p,t,rho,vis,cp,pr,a,k,x,0)
      roout=rho
      rohav=rohav+0.5*(roin+roout)*dz
      thav=thav+0.5*(tbulk(i)*roout+tbulk(i-1)*roin)*dz
1436  roin=roout
      rohav=0.001*rohav/zav
      thav=thav/zav
      thav=thav*0.001/rohav
      phav=phav/zav
      twa=twa/zav
      tca=tca/zav
      tfa=tfa/zav
      p=phav
      t=thav
      call phe(h,p,t,rho,vis,cp,pr,a,k,x,0)
      roav=rho*0.001
      pre=phav*0.000001
      Write(8,1486) pre,thav,roav,rohav,twa,tca,tfa
1486  Format(//' Axial Average Conditions in the Channel'/
     *'    Coolant Pressure(MPa)',f8.3,' Temperature(K)',f9.3/' Give Den
     *sity(g/cm3)',f10.7/'              Average Density',f10.7/
     *'            Wall Temperature(K)',f9.3/
     *'                 Clad Temperature(K)',f9.3/
     *'                      Fuel Temperature(K)',f9.3)
      Tcrit=Tmelt(PUC)
      Write(8,384) PUC,Tcrit
384   format(//' Mol Fraction of UC in UC-ZrC mix',f7.5,' Melting Temper
     *ature(K)',f10.2)
      dh=houtlet-hinlet
      dq=echannel/flow
      dkin=0.5*(ve(npu)**2-ve(1)**2)
      rel=dkin/dq
      bal=(dh+dkin)/dq
      write(8,987) dq,dh,dkin,rel,bal
987   format(//' HEAT BALANCE FOR THE CHANNEL'//
     * 'dQ: heat to coolant (joule/kg)',e14.5/'dH: change of enthalpy
     1(joule/kg)',e14.5/'dKin change kinetic energy(joule/kg)',e14.5/
     2' relative dKin/dQ',f10.6/' Balance (dH+dKin)/dQ',f14.7)
c
      return
      end
c
c friction factor for one phase flow
c
      function fri(Re,ru)
c where Re is the Reynold's number and ru=e/d (rugosity)
c Colebrook-White correlation, page 115 ANS book about Gas Cooled Rs.
      if(Re.le.2000.) then
      fri=16./Re
      else
      fri=0.001375*(1.+(20000.*ru+1000000./Re)*0.33333333)
      endif
      return
      end
c
c calculate drho/dp
```

```
c
      function drdp(pe,te,der)
      p=pe
      t=te
      call phe(h,p,t,rho,vis,cp,pr,a,k,x,0)
      p1=p
      rho1=rho
      p=der*p
      call phe(h,p,t,rho,vis,cp,pr,a,k,x,0)
      drdp=(rho1-rho)/(p1-p)
      return
      end
c
c calculate drho/dt
c
      function drdt(pe,te,der)
      p=pe
      t=te
      call phe(h,p,t,rho,vis,cp,pr,a,k,x,0)
      t1=t
      rho1=rho
      t=der*t
      call phe(h,p,t,rho,vis,cp,pr,a,k,x,0)
      drdt=(rho1-rho)/(t1-t)
      return
      end
c
c function Fu to solve in order to fin t outlet node
c
      function Fu(pi,ti,pe,te,pon,vi)
      t=ti
      p=pi
      call phe(h,p,t,rho,vis,cp,pr,a,k,x,0)
      hi=h
      rhoi=rho
      t=te
      p=pe
      call phe(h,p,t,rho,vis,cp,pr,a,k,x,0)
      Fu=h-hi-pon-0.5*vi**2*(1.-(rhoi/rho)**2)
      return
      end
c
c function Gu to solve in order to find p outlet node
c
      function Gu(po,to,pi,ti,xi,g)
      common /fGu/  ifri,der,pf,ar,d,ru,dz
c evaluate everything at outlet p,t
      p=po
      t=to
      call phe(h,p,t,rho,vis,cp,pr,a,k,x,0)
      d1=drdp(po,to,der)
      d2=drdt(po,to,der)
      f1=1.-d1*(g/rho)**2
      f3=-d2*(g/rho)**2
      Re=d*g/vis
      f=0
      if(ifri.eq.1) f=fri(Re,ru)
      f2=0.5*f*Pf*g*g/(rho*ar)
      f1o=f2/f1
      f2o=f3/f1
c evaluate everything at inlet pi,ti
      p=pi
      t=ti
      call phe(h,p,t,rho,vis,cp,pr,a,k,x,0)
      d1=drdp(pi,ti,der)
      d2=drdt(pi,ti,der)
      f1=1.-d1*(g/rho)**2
```

```
      f3=-d2*(g/rho)**2
      Re=d*g/vis
      f=0
      if(ifri.eq.1) f=fri(Re,ru)
      f2=0.5*f*Pf*g*g/(rho*ar)
      f1i=f2/f1
      f2i=f3/f1
      f1a=(f1i+xi*f1o)/(1.+xi)
      f2a=(f2i+xi*f2o)/(1.+xi)
      Gu=po-pi+f1a*dz+f2a*(to-ti)
      return
      end
c
c Nussel Number for one Phase Flow
c
      function RNu(Re,Pr,Tr,Zr)
c Where Re and Pr and Reynolds and Prandt numbers evaluated at bulk conditions
c Tr: Twall/Tbulk and Zr=z/D. Correlation Mc Eilgot page 93 ANS book about
c gas cooled reactors. The correlation is not applicable for Zr<=5., for
c this value of Zr we use the value of Nu for Zr=5.
      if(Zr.lt.5.) Zr=5.
      RNu=0.021*Re**0.8*Pr**0.4*Tr**(-0.5)*(1.+Zr**(-0.7))
      return
      end
c
c Function to solve in order to find wall temperature
c
      function Fq(rjq,Zr,Rei,Pri,Rki,Tfa,Tr,Tw)
      RNus=RNu(Rei,Pri,Tr,Zr)
      hco=RNus*Rki
      Fq=rjq-hco*(Tw-Tfa)
      return
      end
c
c Function: Thermal Conductivity of UC
c Source: Book El-Wakil
c
      Function RkUc(Te)
c T(K) RkUC (watt/cmK)
      Dimension       T(20),Rk(20)
      np=14
      Data (T(i),i=1,14)/366.5 ,422.0 ,477.6 ,533.2 ,588.7, 644.2,
     *699.7 ,755.2 ,810.8, 922.0, 1033.1, 1144.3, 1255.4, 1366.5/
      Data (Rk(i),i=1,14)/0.2557, 0.2435, 0.2333, 0.2254, 0.2193,
     *0.2145, 0.2110, 0.2081, 0.2062, 0.2046, 0.2036, 0.2025,
     *0.2020, 0.2003/
      call Inter(T,Rk,14,Te,Cond)
      RkUC=Cond
      Return
      End
c
c Function: Thermal Conductivity of ZrC
c Source: Book Bussard-DeLauer
c
      Function RkZrC(Te)
c T(K) RkZrC(watt/cmK)
      RkZrC=0.2077
      Return
      End
c
c Function: Thermal Conductivity of Graphite
c Source: Book busaard-DeLauer
c
      Function RkGraf(Te)
c T(K) RkGraf (watt/cmK)
      Dimension       Tp(20),Rc(20)
      np=12
```

```
      Data (Tp(i),i=1,12)/277.8 ,416.7 ,555.5 ,833.3 ,1111.1 ,1388.9,
     *1666.7 ,1944.4 ,2222.2 ,2500. ,2777.8 ,3333.3/
      Data (Rc(i),i=1,12)/1.743, 1.414, 1.097, 0.817, 0.6339, 0.5485,
     *0.4632, 0.4145, 0.3657, 0.3413, 0.3169, 0.2926/
      call Inter(Tp,Rc,np,Te,Cond)
      RkGraf=Cond
      Return
      End
c
c Function: Melting Temperature of Uc-ZrC mix
c Source: Book busaard-DeLauer
c
      Function Tmelt(PUC)
c  Tmelt(K), PUc mol fraction of Uc in UC-ZrC mix
      Dimension xmol(20),T(20)
      np=6
      Data (xmol(i),i=1,6) / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0/
      Data (t(i),i=1,6)/ 3758., 3594., 3437., 3250., 3022., 2687. /
      call Inter(xmol,T,np,PUC,Tm)
      Tmelt=Tm
      Return
      End
c
c Properties of He
c
      subroutine phe(h,p,t,rho,vis,cp,pr,a,rk,x,iflag)
      common /fphe/ nit,eps,der
c Thermodynamic and Transport Properties of Helium
c from Appendix B, page 405, Thermal and Flow Design of Helium-Cooled
c Reactors by G. Melese and R.Katz American Nuclear Society, 1984.
      cp=5193.
      R=2077.22
      x=1.
c
c go to 10 if iflag=1 (ie input is p,h)
c
      if(iflag.ne.0) go to 10
1     rho=p/(R*t+p*b(t))
      h=cp*t+(b(t)-t*c(t))*p
      z=1.+p*b(t)/(R*t)
      a=z*sqrt(1.666*R*t)
      vis=3.953e-7*t**0.687
      rk=2.774e-3*t**0.701
      pr=0.740*t**(-0.014)
      return
10    continue
c
c use bisect method to find t from p and h
c
      rc=der
      t1=h/cp
      en1=ent(h,p,t1)
      if(en1.lt.0.) rc=1./rc
      t2=t1
      it=0
30    t2=t2*rc
      it=it+1
      en2=ent(h,p,t2)
      if(en1*en2.lt.0) go to 20
      if(it.lt.nit) go to 30
      write(6,50) h,p
50    format(//'Program stop at phe(1). Unable to find T for input h=',
     *e14.5,' and p=',e14.5//)
      stop
20    continue
      it=0
      iwa=0
```

```
452    ta=0.5*(t1+t2)
       ena=ent(h,p,ta)
       test=abs(ena)/h
       if(test.lt.eps)go to 442
       it=it+1
       call bisect(t1,t2,ta,en1,en2,ena,iwa)
       if(it.le.nit.and.iwa.eq.0) go to 452
442    continue
       if(it.gt.nit.or.iwa.eq.1)
      *write(6,51) h,p
51     format(//'Program stop at phe(2). Unable to find T for input h=',
      *e14.5,' and p=',e14.5//)
       if(it.gt.nit.or.iwa.eq.1) stop
       t=ta
       go to 1
       end
c
c c(t) used to compute enthalpy
c
       function c(t)
       data c2,c3,c4,c5 /9.528079e-4,3.420680e-2,
      *                  2.739470e-3,9.409120e-4/
       c=c2*c3/(1.-c3*t)**2-c4*c5/(1.+c5*t)**2
       return
       end
       function ent(h,p,Te)
       ent=h-5193.*Te-(b(Te)-Te*c(Te))*p
       return
       end
c
c
c    ***** DESIGN STRENGTH SUBROUTINE *****
c Calculation of the Stress that Astar alloy can stand
c subroutine from Alkasys
c tt(K) Temperature, n=1 or 2 (it does not matter for sigpv)
c fpl (years) power life, itma Max number iterations
c ipr 1,0 Ok, no Ok pressure vessel
c sigpv(MPa) max stress of ASTAR alloy for pressure vessel
c
  subroutine desstr(tt,n,fpl,sigpv,sigrod,itma,ipr)

  real tt,nr,tmat,fpl,sigpv,sigrod,rho
  real b,c,v,sigma,theta,thet,dtds
        data tmat/1100./
  integer ll, n
        ipr=1
        if(tt.ne.0.) then
   if (tt.gt.tmat) then
    rho=.604
    b=67375.0
    c=3.548E9
    nr=1943
    v=.25875
   else
    rho=.31
    b=72614.8
    c=1.995E10
    nr=3768
    v=.03652
   endif
   do ll=1,n
    if (ll.eq.1) then
     thet=fpl*8766.139
    else
     thet=fpl*8766.139/2.
    endif
    sigma=1000
```

```
    theta=1/c*exp((b-nr*log(sigma)-v*sigma)/tt)
          ite=0
    dowhile ((abs((theta-thet)/thet)).gt.0.001)
     dtds=-theta*(nr+v*sigma)/sigma/tt
     sigma=sigma+(thet-theta)/dtds
           ite=ite+1
           if(ite.gt.itma) ipr=0
           if(ite.gt.itma) return
     if (sigma.lt.100) sigma=100
     theta=1/c*exp((b-nr*log(sigma)-v*sigma)/tt)
    enddo
    if (ll.eq.1) then
     sigpv=sigma*6.894757e-3
    else
     sigrod=sigma*6.894757e-3
    endif
   enddo
   sigpv=sigpv/1.5
  endif
  return
  end       !SUBROUTINE DESSTR
c
c function to compute thickness neutron shielding
c
      Function Shieldn(fluin,fluout,ish,tw,igam)
      common /cross/ AA,BB,signe,ishin,ishig
c
c Fluin,fluout, fast fluxes in/out; ish=1,  LiH+SS
c ish=2, B4C; tw(cm) gamma shielding (Tungstene)(if igam.ne.0)
c
      data remLi,remH,remFe,remW,remBC
     *    /1.01,1.00,1.98,3.13,5.1/
c removal cross section for fast neutrons of Li,H,Fe,W,B4C
c
      data Av,roLiH,roW,roBC,roFe,roLiss,ALiH,AFe,ABC,AW
     *    /0.602202,0.82,19.3,2.52,7.874,1.024,7.95,55.85,55.26,183.85/
      sigW=remW*Av*roW/AW
      ishin=1
      go to(10,20),ish
10    xLiH=(roLiss-roFe)/(roLiH-roFe)
      xFe=1.-xLiH
      sigre=(remLi+remH)*Av*roLiH*xLiH/ALiH+remFe*Av*roFe*xFe/AFe
      go to 30
20    sigre=remBC*Av*roBC/ABC
30    aa=alog(fluin/fluout)/sigre
      if(aa.le.0.) ishin=0
      bb=sigW/sigre
      signe=sigre
      Shieldn=aa-bb*tw
      return
      end
c
c Subroutine to calculate Gamma source
c
      Subroutine Gamso(Sn,aof,sof,sogam,Pr5)
c
c Input: Sn(1/s): neutron production; sof=S/F Elements
c aof=A/F in U235 for sof
c sogam(1): Gamma source between "0" and 1 Mev
c      (2):                       1  and 2
c      (3):                       2  and 3
c      (4):                       3  and 5
c      (5):                       5  and 7
c      (6):                       7  and 9
c      (7):                       > around 9 Mev
      Dimension Sogam(7),Pf(7),Fp(7),GNi(7),GH(7),GZr(7),GBe(7),
     *GU(7),GFe(7),GCr(7),Pr5(7)
```

```
c
c Pf: Prompt Fission Gammas; Fp: Fission Products Gammas(Both, Per Fiss)
c GNi to GCr:Captures Gammas(per Event) for indicated Elements Ni,..Cr
c
      Data(Pf(i),i=1,7)/3.1,2.11,.73,.26,.0428,0.,0./
      Data(Fp(i),i=1,7)/9.03,1.137,.4,0.,0.,0.,0./
      Data(GNi(i),i=1,7)/.84,.40,.23,.23,.34,.62,0./
      Data(GH(i),i=1,7)/0.,0.,1.,0.,0.,0.,0./
      Data(GZr(i),i=1,7)/0.,0.,0.,1.13,.35,.04,0./
      Data(GBe(i),i=1,7)/0.,0.,0.,.54,.73,.0,.0/
      Data(GU(i),i=1,7)/2.54,1.78,.91,.34,.0,.0,0.041/
      Data(GFe(i),i=1,7)/.75,.60,.27,.23,.25,.38,.02/
      Data(GCr(i),i=1,7)/.85,.41,.21,.12,.23,.39,.06/
      Data FNi2,FH2,FZr2,FBe2,FU82,FFe2,FCr2
     *     /.12,.083,.088,.051,.045,.016,.008/
      Data FNi0,FH0,FZr0,FBe0,FU80,FFe0,FCr0
     *     /0.,0.,.076,.087,.069,0.,0./
      Data Rnu/2.44/
c
c See: Total number of Absorptions= Sn(Total production)
c    F: total number of fission
c    A5: Total number of absorptions(capture+fission) in U235
c    C5: total number of captures in U235
c    Ct:   "     "     " captures
      F=Sn/Rnu
      A5=F*aof
      C5=A5-F
      Ct=Sn-F
      CNi=(FNi0+(FNi2-FNi0)*sof/2.)*Ct
      CH=(FH0+(FH2-FH0)*sof/2.)*Ct
      CZr=(FZr0+(FZr2-FZr0)*sof/2.)*Ct
      CBe=(FBe0+(FBe2-FBe0)*sof/2.)*Ct
      CU8=(FU80+(FU82-FU80)*sof/2.)*Ct
      CFe=(FFe0+(FFe2-FFe0)*sof/2.)*Ct
      CCr=(FCr0+(FCr2-FCr0)*sof/2.)*Ct
      Do 10 i=1,7
      Gam5=F*(Pf(i)+Fp(i))+C5*GU(i)
      Sogam(i)=Gam5+CU8*GU(i)+CNi*GNi(i)+CH*GH(i)
     *        +CZr*GZr(i)+CBe*GBe(i)+CFe*GFe(i)+CCr*GCr(i)
      Pr5(i)=100.*Gam5/Sogam(i)
10    continue
      return
      end
c
c
      Subroutine Shieldga(Sogam,Dose,Dosea,Dosew,tiempo,tgam,ten,igam,
     *ishn,iten)
      Common /CROSS/ AA,BBB,signe,ishin,ishig
c
c Shieldga: Computes gamma shielding
c Sogam(i),i=1,7 Gamma Source (photons/sec); Dose(rad): Integrated
c Dose at load area; Dosew(rad): Dose without the shielding
c Dosea: dose for tgam
c igam:flag if =1 problems with iterations
c Tgam(cm): Thickness of Tungsten
c ten(cm) thickness of n shielding, ishn=1, LiH, ishn=2, B4C
c iten/0,1/ /ten input,ten iterated by this subroutine/
c tgam:iterated,if necessary,to have Dose=input
c Index(1): Gamma Processes between "0" and 1 Mev
c      (2):                        1  and 2
c      (3):                        2  and 3
c      (4):                        3  and 5
c      (5):                        5  and 7
c      (6):                        7  and 9
c      (7):                        > around 9 Mev
      common /Gamsh/ coBe,coTa,coC,coZr,coU,coH,coNi,coFe,coCr,coTi,coSi
     *,dist,Req,shg(7),shgnb(7),Zeq(7),Dosewg(7),Doseg(7),itmag,epsg
```

```
     *,aten,Dosen,Doseng(7)
      dimension Sogam(7),GBe(7),GW(7),Gc(7),GZr(7),GU(7),GH(7),GFe(7),
     *GTi(7),GSi(7),b(4,7),a3(4,7),a4(4,7),zb(4),a(4),a1(4),a2(4),rh(7)
     *,ega(7),sign(7),An(7),A1n(7),A2n(7)
c
      data (GBe(i),i=1,7) /.0773,.0459,.0394,.0266,.0211,.0180,.0161/
      data (GW(i),i=1,7) /.125,.0492,.0437,.0402,.0418,.0438,.0465/
      data (GC(i),i=1,7) /.087,.0518,.0444,.0304,.0245,.0213,.0194/
      data (GZr(i),i=1,7) /.0851,.0467,.0414,.0349,.0344,.0349,.0359/
      data (GU(i),i=1,7) /.176,.0548,.0484,.044,.0455,.0479,.0511/
      data (GH(i),i=1,7) /.173,.103,.0876,.0579,.0446,.0371,.0321/
      data (GFe(i),i=1,7) /.0828,.0485,.0424,.033,.0304,.0295,.0294/
      data (GTi(i),i=1,7) /.0876,.0518,.0451,.0338,.0302,.0285,.0280/
      data (GSi(i),i=1,7) /.0869,.0517,.0447,.0323,.0277,.0254,.0243/
c
c GBE...GSi are the total mass attenuation coefficients(cm**2/g)
c for the 7 gamma groups. Fe(z=26),Ni(z=28) and Cr(z=24) can be lumped
c together; Ta(z=73) lumped with W(z=74);GZr(z=40) is indeed GMo(z=42)
c and GTi(z=22) is indeed GCa(z=20)
c Also Li(Z=3) and B(z=5) are lumped with Be(z=4)
c
      data (b(1,j),j=1,7)/8.,5.5,4.5,3.8,3.1,2.3,2.25/
      data (b(2,j),j=1,7)/10.,8.,5.5,3.75,2.9,2.35,2.0/
      data (b(3,j),j=1,7)/3.3,2.9,2.7,2.05,1.2,.7,.6/
      data (b(4,j),j=1,7)/1.65,2.45,2.15,1.65,.96,.67,.5/
c
      data (a3(1,j),j=1,7)/.11,.082,.074,.066,.064,.062,.060/
      data (a3(2,j),j=1,7)/.0948,.0895,.0788,.075,.0825,.0833,.095/
      data (a3(3,j),j=1,7)/.043,.069,.086,.118,.171,.205,.212/
      data (a3(4,j),j=1,7)/.032,.045,.097,.123,.175,.204,.214/
c
      data (a4(1,j),j=1,7)/.044,.093,.116,.130,.152,.150,.128/
      data (a4(2,j),j=1,7)/.012,.04,.07,.082,.075,.0546,.0116/
      data (a4(3,j),j=1,7)/.148,.188,.134,.070,.0,.052,.144/
      data (a4(4,j),j=1,7)/.296,.178,.103,.064,.059,.067,.08/
c
      data (zb(i),i=1,4)/13.,26.,74.,82./
c
c b(i,j),a2(i,j) and a4(i,j) are build up parameters for groups j
c and atomic number Z=Zb(i)
c
      data (rh(i),i=1,7)/4.5e+5,5.5e+5,7.e+5,8.e+5,8.5e+5,9.e+5,1.e+6/
c
c rh(i) Gamma Flux Mev/cm2sec that produces 1 Roentgen/hour
c
      data (Ega(i),i=1,7)/0.5,1.5,2.5,4.,6.,8.,10./
c
c ega(i) Mev Average Energy of gamma group i
c
      data ABe,ATa,AW,AC,AZr,AU,AH,ANi,AFe,ACr,ATi,ASi,Av,RoW
     */9.0122,180.948,183.85,12.011,91.22,235.044,1.00797,58.71,55.847,
     *51.996,47.90,28.086,.602202,19.3/
      data ZBe,ZTa,ZW,ZC,ZZr,ZU,ZH,ZNi,ZFe,ZCr,ZTi,ZSi,pee,ZLi,ZBo
     */4.,73.,74.,6.,40.,92.,1.,28.,26.,24.,22.,14.,3.141593,3.,5./
c
      data roLiH,roBC,RoSS,RoLiS /0.82,2.92,7.874,1.024/
      igam=1
      ishig=1
      xLiH=(RoLiS-RoSS)/(RoLiH-RoSS)
      xSS=1.-xLiH
c
c  Attenuation*Build Up factor for N Shield(which also shield gammas)
c
      do 101 j=1,7
      go to(111,222),ishn
c ishn=1 LiH+SS
111   sig=RoLiS*(xLiH*0.8732*GBe(j)+xLiH*0.1268*GH(j)+xSS*GFe(j))
```

```
      sigZ=RoLiS*(ZLi*xLiH*0.8732*GBe(j)+ZH*xLiH*0.1268*GH(j)+ZFe*xSS*GF
     *e(j))
      ZZ=sigZ/sig
      go to 333
c ish=2 B4C
222   sig=RoBC*(0.7826*GBe(j)+0.2174*GC(j))
      sigZ=RoBC*(ZBo*0.7826*GBe(j)+ZC*0.2174*GC(j))
      ZZ=sigZ/sig
c ZZ is "equivalent Z for buildup factor
333   do 444 i=1,4
      a(i)=b(i,j)
      al(i)=a3(i,j)
444   a2(i)=a4(i,j)
c interpolate in table as function of Z,
      Call Inter(zb,a,4,zz,am)
      Call Inter(zb,a1,4,zz,al1)
      Call Inter(zb,a2,4,zz,al2)
      An(j)=am
      Aln(j)=-al1
      A2n(j)=al2
      sign(j)=sig
101   continue
c end Calculation Atenuation/build up factor for n shield
c
c Start Calculation Self Shielding in Reactor
c
      do 10  j=1,7
c
      do 20 i=1,4
      a(i)=b(i,j)
      al(i)=a3(i,j)
20    a2(i)=a4(i,j)
      sig=coBe*Abe*Gbe(j)     +coTa*ATa*GW(j)       +coC*AC*GC(j)
     *   +coZr*AZr*GZr(j)     +coU*AU*GU(j)         +coH*AH*GH(j)
     *   +coFe*AFe*GFe(j)     +coNi*ANi*GFe(j)      +coCr*ACr*GFe(j)
     *   +coTi*ATi*GTi(j)     +coSi*ASi*GSi(j)
      sig=sig/Av
      ZZ=coBe*Abe*Gbe(j)*ZBe +coTa*ATa*GW(j)*ZTa   +coC*AC*GC(j)*ZC
     *   +coZr*AZr*GZr(j)*ZZr +coU*AU*GU(j)*ZU      +coH*AH*GH(j)*ZH
     *   +coFe*AFe*GFe(j)*ZFe +coNi*ANi*GFe(j)*ZNi  +coCr*ACr*GFe(j)*ZCr
     *   +coTi*ATi*GTi(j)*ZTi +coSi*ASi*GSi(j)*ZSi
      ZZ=ZZ/Av
      zz=zz/sig
      Call Inter(zb,a,4,zz,am)
      Call Inter(zb,a1,4,zz,al1)
      Call Inter(zb,a2,4,zz,al2)
      Zeq(j)=zz
      b1=2.*sig*Req*(1.-al1)
      b2=2.*sig*Req*(1.+al2)
      exp1=0.
      exp2=0.
      if(b1.lt.50.) exp1=exp(-b1)
      if(b2.lt.50.) exp2=exp(-b2)
      shg(j)=3.*Am*(0.5-(1.-exp1*(1.+b1))/b1**2)/b1+
     *       3.*(1-Am)*(0.5-(1.-exp2*(1.+b2))/b2**2)/b2
      bb=2.*sig*Req
      expo=0.
      if(bb.lt.50.) expo=exp(-bb)
      shgnb(j)=3.*(0.5-(1.-expo*(1.+bb))/bb**2)/bb
10    continue
c
c End Calculation Self shielding
c
c
c Dosew and Dosewg() are the doses without any shield(n or gamma)
c Dosen and Doseng() are the doses without W(gamma shield) but inclu-
c ding the shielding effects of the neutron shield(either input or itera-
```

```
c ted
      time=tiempo/3600.
      Dosew=0.
      Dosen=0.
      if(iten.eq.1)  ten=AA
      if(iten.eq.1.and.ishin.eq.0) ten=0.
      do 30 i=1,7
      tsn=Sign(i)*ten
      Shin=exp(-tsn)*(An(i)*exp(-tsn*A1n(i))+(1.-An(i))*exp(-tsn*A2n(i))
     *     )
      Dosewg(i)=time*Sogam(i)*shg(i)*Ega(i)/(4.*pee*dist**2*rh(i))
      Doseng(i)=Shin*Dosewg(i)
      Dosew=Dosew+Dosewg(i)
30    Dosen=Dosen+Doseng(i)
      if(Dosen.le.Dose) ishig=0
      if(ishig.eq.0) tgam=0.
      if(ishig.eq.0) return
c
c Return if there is no need for gamma shield (thick W=0.)
c
      t1=0.
      Dose1=Dosen
      f1=Dose1-Dose
c
c Starts iteration on tgam:
c First look for
c t2 such that Dose(t2) <Dose
c
      it=0
      sig=GW(1)*RoW
      t2=alog(Dosewg(1)/Dose)/sig
 51   dose2=0.
      do 40 i=1,7
      am=b(3,i)
      al1=-a3(3,i)
      al2=a4(3,i)
      sig=GW(i)*RoW
      BFA=am*exp(-al1*sig*t2)+(1.-am)*exp(-al2*sig*t2)
40    dose2=dose2+Dosewg(i)*exp(-sig*t2)*BFA
      if(dose2.lt.dose) go to 50
      t2=1.01*t2
      it=it+1
      if(it.le.itmag) go to 51
      igam=0
      write(8,*) ' Problem with Calculation of Gamma Shielding'
50    continue
      f2=Dose2-Dose
      if(igam.eq.0) return
      it=0
61    tgam=0.5*(t1+t2)
      dosea=0.
      do 41 i=1,7
      am=b(3,i)
      al1=-a3(3,i)
      al2=a4(3,i)
      sig=GW(i)*RoW
      BFA=am*exp(-al1*sig*tgam)+(1.-am)*exp(-al2*sig*tgam)
c
c recompute neutron shielding(ten) because W also shield neutrons
c see: if ten<0. program set ten to 0.(ie no need of n shielding
c because W is enough also if ishin=0(fluin lt prescribed fluout)
c
      if(iten.eq.1)  ten=AA-BBB*tgam
      if(ten.lt.0.) ten=0.
      if(iten.eq.1.and.ishin.eq.0) ten=0.
      tsn=Sign(i)*ten
      Shin=exp(-tsn)*(An(i)*exp(-tsn*A1n(i))+(1.-An(i))*exp(-tsn*A2n(i))
```

```
     *     )
      Doseg(i)=Dosewg(i)*exp(-sig*tgam)*BFA*Shin
41    dosea=dosea+Doseg(i)
c recompute n shielding, because it might be smaller than input max flu-
c ence
      eco=(ten+BBB*tgam)*signe
      aten=exp(-eco)
      fa=Dosea-Dose
      rel=abs(dosea-dose)/dose
      if(rel.le.epsg) go to 60
      call Bisect(t1,t2,tgam,f1,f2,fa,iwa)
      it=it+1
      if(it.le.itmag.and.iwa.ne.1) go to 61
      igam=0
      write(8,*) ' Problems with calculation of Gamma Shielding'
60    continue
      return
      end
c
c Compute sizes for input Energy and S/F
c
      Subroutine Size(das,dop,E,sof,d1,d2)
c
      common /Soze/ its,epss
c
      common /MaBaSi/ R,H,ishape
c
      ri=1.
      it=0
10    x1=das*(1.+0.01/ri)
      if(ishape.eq.2) R=x1
      if(ishape.eq.3) H=x1
      e1=Balance(ibu,sof)-E
      if(e1.gt.0.) go to 20
      ri=ri+1.
      it=it+1
      if(it.gt.its) go to 100
      go to 10
20    x2=dop
      if(ishape.eq.2) R=x2
      if(ishape.eq.3) H=x2
      e2=Balance(ibu,sof)-e
30    it=0
35    xav=0.5*(x1+x2)
      if(ishape.eq.2) R=xav
      if(ishape.eq.3) H=xav
      Ea=Balance(ibu,sof)-e
      test=abs(Ea)/E
      if(test.le.epss) go to 50
      call bisect(x1,x2,xav,E1,E2,Ea,iwa)
      it=it+1
      if(iwa.eq.1) go to 100
      if(it.gt.its) go to 50
      go to 35
50    d1=xav
c
c right side limit
c
      x1=dop
      if(ishape.eq.2) R=x1
      if(ishape.eq.3) H=x1
      E1=Balance(ibu,sof)-e
21    it=0
41    x2=x1*(1.+0.10)**(it+1)
      if(ishape.eq.2) R=x2
      if(ishape.eq.3) H=x2
      e2=Balance(ibu,sof)-e
```

```
      if(e2.gt.0.) go to 31
      it=it+1
      if(it.gt.its) go to 100
      go to 41
31    it=0
36    xav=0.5*(x1+x2)
      if(ishape.eq.2) R=xav
      if(ishape.eq.3) H=xav
      Ea=Balance(ibu,sof)-e
      test=abs(Ea)/E
      if(test.le.epss) go to 51
      call bisect(x1,x2,xav,E1,E2,Ea,iwa)
      it=it+1
      if(it.gt.its.or.iwa.eq.1) go to 100
      go to 36
51    d2=xav
      return
100   write(8,101) it
101   format(//' Program stop at subroutine Size'/' iterations=',i10/)
      stop
      end
```

APPENDIX B. LISTING OF CODE NEPPIN

```
c program to estimate preliminary design of fuel pin,liquid Li reactor
c for space applications.
c Assumptions
c 1) fuel cell: liquid Li(coolant),Astar alloy clad,W liner,He gap and
c   UN pellet.
c
c 2) Astar pressure vessel, OBe reflector and B4C(90%B10) drums in
c    reflector.
c
c 3) Gamma Shield: W; Neutron Shield:B4C or LiH in SS matrix
c
c 4) Data base parameters: Enrichment and pitch/drod for drod=6.4mm
c
      character*80 tit,aldum
c'
      dimension aux(5),bux(5),wbeth(8),th(8),pfr(3,3),U5md(5),isha(5),
     *pfrm(3,3),Sogam(7),PrU(7),Emin(5,5),Ras(5,5),Has(5,5),Rop(5,5)
     *,Hop(5,5),enri(5),podi(5),BUal(5),Rd(5),Hd(5),dpvd(5),delp(5),
     *GRea(5),dkbup(5),dkpvd(5),rkbare(5),enrd(5),dkobe(5),Nbun(5),
     *podip(3),enrip(3)
c
      common /Soze/ its,epss
c
      common /MaBaSi/ R,H,ishape
c
      common /mabu/ c5e,c8e,c9e,p9,cfpe,ftin
c
c U5,U8,Pu9,FP concentrations at EOL (integrated fluence=ftin)
c all from subrou bupo.Per unit input volume
c
      common /Bala/ Ereq,relr,relal,rela2,dkeol,dkstr,BU,Vcor,
     *U5m,dkbu,dkbe,rkre,dcr,sigma,pin,dpv,dkpv,tduco,
     *eps,itest
c
      common /cont/ Rad,dvp,drr,enr,pod,refec,ncr,dksu,dkest,relar,dcd
c
c
      common/thermo/ tit,nnodo,ifri,npin,pint,tin,toutl,deltapa
     *,Pow,rjs,flowt,podt,tduc,Rc
     *,Hact,Pfacr,Daxi,fact,iprth,iopth,ichoose
     *,Pfacaz,iboun,nrod,nbu,nrodbu,hod
c
c
      common /Conce/ coBe,coTa,coU5,coU8,coLi,coW,coRe,coHf,coHe,coN,coO
      common /Gamsh/
     *dist,Req,shg(7),shgnb(7),Zeq(7),Dosewg(7),Doseg(7),itmag,epsg
     *,aten,Dosen,Doseng(7)
c
      Common /CROSS/ AU,BBU,signe,ishin,ishig
c
      Tsa(ps)=8143./(8.-alog10(ps/0.133322))
      Psa(ts)=0.133322*10.**(8.00-8143./ts)
c
c Tsa(ps) is saturation temperature of Li(in K) as function of ps(KPa)
c Psa(ts) is      "       pressure    of Li(in KPa)"      "     of ts(K)
c
      open(unit=7,file='neppin1.i',status='old')
      open(unit=8,file='neppin1.out',status='unknown')
c
c
      TLim=453.7
      read(7,11) tit
11    format(a80)
c
```

```
      read(7,11) aldum
      read(7,20) itest,ndt,itmax,icont,iboun,nput
      read(7,11) aldum
      read(7,10) Pow,D,BU,eps,der
      read(7,11) aldum
      read(7,10) drr,dra1,dra2,tduc
      read(7,11) aldum
      read(7,10) dkstr,dkeol,dksu
      dkest=dkstr
      read(7,11) aldum
      read(7,10) pin,tin,pHe,THe,tempv
      read(7,11) aldum
      if(iboun.eq.1) read(7,10) flowt
      if(iboun.eq.2) read(7,10) toutl
      if(icont.eq.0) go to 2379
      read(7,11) aldum
      read(7,20) ncr,nnodo,npin,ishape,nrodbu
      read(7,11) aldum
      read(7,20) iopth
      read(7,11) aldum
      read(7,20) ifri,iprth,iouth
      read(7,11) aldum
      read(7,20) iwrte,ishn,ichoose,ichoode
      read(7,11) aldum
      read(7,10) fact,Ripple,hod,dcd
      read(7,11) aldum
      read(7,10) delpr
      read(7,11) aldum
      read(7,10) fastf,Dose,zload,rload
      read(7,11) aldum
      if(ishape.eq.1) go to 2379
      if(ishape.eq.2) read(7,10) R
      if(ishape.eq.3) read(7,10) H
c
c check if T inlet>T melt Li
c
2379  continue
      if(tin.le.TLim) write(8,17) tin,TLim
17    format('     Inlet Temperatures is(K)',f8.2,' smaller than'/
     *       ' Melting Temperature of Li(K)',f8.2/
     *       'PROGRAM STOP')
      if(tin.le.TLim) stop
c
c compute flowt(Kg/s) or Toutlet according to option iboun
c
      pint=pin
      go to(111,222),iboun
111   call pli(hin,pin,tin,rhoi,vis,cp,pr,k,x,0)
      hout=hin+Pow*fact*1.e+6/flowt
      call pli(hout,pin,toutl,rhoo,vis,cp,pr,k,x,1)
      go to 333
222   call pli(hin,pin,tin,rhoi,vis,cp,pr,k,x,0)
      call pli(hout,pin,toutl,rhoo,vis,cp,pr,k,x,0)
      flowt=Pow*fact*1.e+6/(hout-hin)
333   continue
      roLi=0.5*(rhoi+rhoo)/1000.
c
c check degree of subcooling at outlet
c
      Ts=Tsa(pin*1000.)
      if(toutl.ge.Ts) write(8,245) pin,Ts,toutl
245   format(//' Your reactor is underpressurized'/
     *' At inlet pressure(MPa)',f8.3,' Li boils at K',f10.2/
     *'              smaller than outlet temperature',f10.2/
     *' PROGRAM STOPS')
      if(toutl.ge.Ts) stop
c
```

```
c
      if(tempv.lt.0) tempv=toutl
c
c
c compute max stress (sigma) for pressure vessel
c
  call  desstr(tempv,1,D,sigma,sigrod,itmax,ipresv)
c
c
c
      epso=eps
      epsi=eps
      epsg=eps
      epss=eps
      dero=der
      Daxi=5.
      tduco=tduc
c
      Ereq=Pow*D*365.25
c
c Core life in sec,multiply by 1 julian year=31,557,600. sec
      tiempo=D*3.15576e+7
c
c
c        INPUT DESCRIPTION FOR REALS
c
c
c Pow(Mw): power;  D(years): core life; BU(at%): percent U235 atoms
c burned;
c Inlet flow conditions:
c pin,tin(MPa,K)) pressure and temperature of the inlet coolant(fuel el
c pHe,THe(MPa,K) initial"   "      "    of He gap in rod.
c drr,dra1,dra2 (cm) thickness of Be reflector:radial,top-bottom axial
c dkstr(%): reactivity structural materials(estimated)
c dkeol(%):     "      at end of life(desired)
c dksu (%):desired subcritical reactivity required at BOL with drums in
c eps: tolerance for  iterations (like energy)
c der=1.01 (compute derivatives)
c fact:fraction fission energy/thermal
c flowr(kg/s)total coolant flow(applicable if iboun=1)
c Daxi(cm): extrapolation length axial cos power distribution
c Ripple:Relative amplitude of power distribution as function of angle
c at R=Rcore
c tduc(mm): thickness of the duct of the bundle of fuel elements
c hod: led of the spacing wire of the fuel rod/rod diameter
c dcd(cm): thickness B4C sheet of control drums
c tempv(K): Temperature of Pressure Vessel; if <0. program chooses tout
c delpr(Pa): max delta pressure for core
c toutl(K): outlet temperature of coolant(applicable if iboun=2)
c fastf(1/cm2): fast fluence (during time D) at payload area
c zload(m) : axial distance from the base of reactor to payload plane
c rload(m) : radius of load area
c Dose(Rad): integrated gamma dose(during time D) required for payload
c ddt(cm): delta r and delta h for test curve Eavailable=f(R) or f(H)
c
10    format(7e10.0)
c
      imax=itmax
      imaxt=itmax
      itmag=itmax
      its=itmax
c
c
c       INPUT DESCRIPTION FOR INTEGERS
c
c
c itmax: max number iterations for reactor search
```

```
c ncr: number of control drums
c ishape/1,2,3/ Shape: ideal(minimum volume),R input,H input
c nrodbu: number of fuel rods per bundle
c iwrte=1 write iterations
c itest=0/1/ no,yes run and print test to know your options
c nnodo number of nodes in coolant channel
c ifri/0,1/ no,yes friction factors;iditus/0,1/ no,yes Dittus Correlatio
c npin; number of nodes inside fuel pin
c iprth/0,1/no,yes print details cooling channel
c iopth=1 3-D uniform power density assumed
c      =2 2-D (r,theta) uniform power density assumed
c      =3 Uniform Fuel load/uniform coolant flow assumed
c      =4 Uniform Fuel load/Coolant flow proportional to power density
c         assumed
c iouth=1 Thermodynamic profile at Max power density channel is printed
c iouth=2       "           "    "  Min   "      "        "     "   "
c iouth=3       "           "    " Max and Min   "       "     "   "
c iboun=1 Input Coolant Flow
c iboun=2 Input Outlet Temperature of Coolant
c ishn=1,2  Choose Neutron Shielding LiH+SS/ B4C
c icont/0,1/ no,yes continue calculation after test
c ndt:number of intervals between Xas and Xop for test curve Energy=f(R)
c or f(H) nput is the number of points in tabulation
c ichoose=0,1 no,yes correction,by program,of incompatible request
c ichoode=/1,2,3,4,5/ choose design with p/d=1.1,1.2,1.3,1.4,1.5
c if different from /1,2,3,4,5/ program choose one for you
c
c
c
20    format(14i5)
c
c
      if(drr.gt.40..or.dra1.gt.40..or.dra2.gt.40.)
     *write(8,31) drr,dra1,dra2
31    format(//' YOUR VALUE OF REFLECTOR THICKNESS(CM)',3f8.2/
     *         ' ARE TOO LARGE FOR A REASONABLE EXTRAPOLATION'/
     *         ' OF THE DATA BASE'//
     *         ' PROGRAM STOPS '//)
c
      if(drr.gt.40..or.dra1.gt.40..or.dra2.gt.40.) stop
c
c
      data dc,dl,dhe,dpin,drod,dref/0.635,0.127,0.025,4.826,6.4,25./
c
c dc,dl,dhe(mm) clading,liner and He gap thickness
c dpin(mm),drod  pin rod diameters
c TLim(K) melting temperature of Li
d dref(cm): thickness of Be reflector(for data base)
c
      dclad=0.1*dc
c
      data pee,Av,ABe,xop/3.1415926,0.602202,
     *                    9.01219,0.541315/
c
c xop:optimum shape for cylindrical reactors; Av: Avogadro's number
c ABe: Be at-weight;pisq3=pi*sqrt(3)
c
      data AU5,AU8,APu,AN,AHe/235.044,238.0508,239.0522,14.00307,4.0026/
      data AW2,AW3,AW4,AW6/181.9483,182.9503,183.9510,185.9543/
      data ATa,ARe5,ARe7,ALi6,ALi7/180.9480,184.9530,186.9560,6.01513,
     *7.01601/
      data AW,ARe,AHf,AO/183.85,186.2,178.49,15.9994/
c
c atomic weights
c A+chemical symbol+1 digit for isotopes
c
      data xLi6,xLi7,xRe5,xRe7/0.0742,0.9258,.3707,.6293/
```

```
      data xW2,xW3,xW4,xW6/0.2655,0.1440,0.3064,0.2841/
c
c atomic fractions
c x+chemical symbol+digit for isotope
c 0.37% of N15 ,0.0132% of Ta180 and  0.14% of W180 ignored
c
      data roun,row,roAst,roobe/12.172,18.6,16.84,3.01/
c
c densities (g/cm3)
c roun: UN; row: W; roAst: Astar alloy
c roobe: OBe
c
c Average densities of the fuel rod
c
      rrod=0.1*drod/2.
      rpin=0.1*dpin/2.
      rclad=rrod-dc*0.1
      rliner=rclad-dl*0.1
      Arod=pee*rrod**2
      fclad=pee*(rrod**2-rclad**2)/Arod
      fliner=pee*(rclad**2-rliner**2)/Arod
      fHe=pee*(rliner**2-rpin**2)/Arod
      fpin=pee*rpin**2/Arod
      rorod=fpin*roUN+fliner*roW+fclad*roAst
c
c
      data xTa,xW,xRe,XHf,RoLis,RoBC
     */0.90275,0.08,0.01,0.00725,1.024,2.92/
c
c weight fraction Astar   alloy
c xTa: Ta; xW: W; xRe: Re; xHf: Hf
c Densities(g/cm3): Ro...
c RoLis:LiH+Stainless Steel N'Shield
c RoBC: B4C(N' Shield)
c
      data wfi,rnu,remLi,remBe,remO,remU,remW
     */3.2042e-11,2.44,1.01,1.07,0.92,3.6,3.13/
c
c wfi: joules/fission(200Mev); rnu: number of neutrons per fission
c rem Li,Be,O,U,Ta : removal cross sections(barns) for fast neutrons
c
      data (pfr(1,j),j=1,3)/1.80,1.85,1.91/
      data (pfr(2,j),j=1,3)/1.68,1.72,1.76/
      data (pfr(3,j),j=1,3)/1.61,1.63,1.65/
c
c pfr(i,j)radial maximum power factor at BOL(with control drums in)
c (then at center line)for uniform load as function of enr i,p/d j
c
      data (pfrm(1,j),j=1,3)/0.353,0.318,0.276/
      data (pfrm(2,j),j=1,3)/0.433,0.410,0.382/
      data (pfrm(3,j),j=1,3)/0.490,0.498,0.472/
c
c pfrm(i,j) minimum radial power factors at BOL( at border core)
c same restrictions as pfr
c
c pfr and pfrm are for enrip(i) and podip(j) bellow
      data (enrip(i),i=1,3)/0.5,0.70,0.93/
      data (podip(j),j=1,3)/1.1,1.3,1.5/
c
c
      data (enri(i),i=1,5)/0.5,0.6,0.7,0.8,0.93/
      data (podi(i),i=1,5)/1.1,1.2,1.3,1.4,1.5/
c
c enri(i): enrichment, podi(i) pitch/diameter of the data base
c
      data (wbeth(i),i=1,8)/0.,0.2728,0.3947,0.5026,0.6778,0.8524,
     *                      0.9517,1.000/
```

```
c wbeth: relative(to 30 cm) worth of OBe as function of its thickness
c th(cm)(input next)
c
      data (th(i),i=1,8)/0.,3.67,5.67,7.67,11.67,17.67,24.,30./
c
      nenr=5
      npod=5
c
c relative worth of OBe reflectors: relr, radial;rela1,rela2:axial
c
      ibe=8
      call Inter(th,wbeth,ibe,drr,relr)
      relar=relr
      call Inter(th,wbeth,ibe,dra1,rela1)
      call Inter(th,wbeth,ibe,dra2,rela2)
      rela1=0.5*rela1
      rela2=rela2*0.5
c
c if itest=1 print your options:
c
c  TTTTT EEEEE SSSSS TTTTT     1) Energy available with optimum shape
c    T   E     S       T       2) Radius of infinite cylinder
c    T   EEE    S      T       3) Height of infinite "pancake"
c    T   E        S    T       4) Energy available as function of R and
c    T   EEEE  SSSSS   T          H
c                              (2 or 3 would produce "infinite" energy)
c
c
c
      ish=ishape
      RRR=r
      HHH=h
c
c
      itpr=itest
      if(itpr.eq.1)
     *write(8,6133) Ereq,BU,drr,dr1,dr2,dkeol,dkstr
6133  format(//' INPUT'/' You requested Mwdays',f10.2/
     *                    ' And burn-up(%)',f5.2/
     *' With radial OBe reflector thickness(cm)',f6.2/
     *'  "   top      "     "       "          ',f6.2/
     *'  "   bottom   "     "       "          ',f6.2/
     *'             End of Life delta k(%)     ',f6.2/
     *'             Structural  delta k(%)     ',f6.2/)
      if(itpr.eq.1)
     *write(8,1632)
c
1632  format(' Run Test, Energy Available (Mwd) for Optimum Shape'/
     *       ' with corresponding Rop and Hop'/
     *       '           Critical Radius(cm) for Infinite Reactor'/
     *       '           Critical Height(cm) for Infinite Reactor'/
     *       ' For your input conditions')
      itest=1
      do 299 ipod=1,npod
      pod=podi(ipod)
      if(itpr.eq.1)   write(8,302) pod
302   format(/'p/d=',f6.4)
      if(itpr.eq.1)   write(8,301)
301   format(/
     *'enr         Energy       Rop       dpv       Hop    Rc        dpv
     *       Hcr')
c
c
c compute Energy available for ideal shape Emin()
c and asymptotic radius Ras() and height Has()
c
      do 1633 ienr=1,nenr
```

```
      ishape=1
      enr=enri(ienr)
      podt=pod
      Ene=Balance(ibu,enr,pod)
      Emin(ienr,ipod)=Ene
      Rop(ienr,ipod)=R
      Hop(ienr,ipod)=H
      Ropo=R
      Hopo=H
      dpvo=dpv
      ishape=2
      Edu=Balance(ibu,enr,pod)
      Rcri=dcr
      dpvc=dpv
      Ras(ienr,ipod)=Rcri
      ishape=3
      Edu=Balance(ibu,enr,pod)
      Hcri=dcr
      Has(ienr,ipod)=Hcri
c
c
      if(itpr.eq.1) write(8,1634) enr,Ene,Ropo,dpvo,Hopo,Rcri,dpvc,Hcri
1633  continue
299   continue
c
1634  format(f5.3,f12.1,6f10.2)
c
      itest=itpr
      if(itest.eq.0) go to 1631
c
c compute curves Eavailable as function of radius and Height of core and
c enrichment and p/d in order to illustrate the user about the options.
c
      do 5682 ipod=1,npod
      pod=podi(ipod)
      do 5782 ienr=1,nenr
      enr=enri(ienr)
c
c
      write(8,5783) enr,pod,Ereq
c
5783  format(//
     *' AVAILABLE ENERGY(MWDAYS) AS FUNCTION OF ENRICHMENT,P/D,R AND H'/
     */'ENR=',f6.4,10x,'P/D=',f7.4/' Required Energy(Mwd)',f10.2//
     *'    I',4x,'R(CM)PVcm  R/H',7x,'EAVAI',4x,'H(CM)',
     *'PVcm  R/H',7x,'EAVAI'//)
c
      Rs0=Ras(ienr,ipod)*der
      Hs0=Has(ienr,ipod)*der
      rdt=ndt
      delr=(Rop(ienr,ipod)-Rs0)/rdt
      delH=(Hop(ienr,ipod)-Hs0)/rdt
      do 5782 idt=1,nput
      ishape=2
      rit=idt-1
      Rs=Rs0+rit*delr
      R=Rs
      Ener=Balance(ibu,enr,pod)
      dpvr=dpv
      xra=R/H
      ishape=3
      Hs=Hs0+rit*delH
      H=Hs
      EneH=Balance(ibu,enr,pod)
      dpvh=dpv
      xh=R/H
c
```

```
c
      write(8,5784) idt,Rs,dpvr,xra,Ener,Hs,dpvh,xh,EneH
c
5784  format(i5,2(f9.2,f4.1,f5.2,f12.0))
c
5782  continue
5682  continue
c
c
c     EEEEE   N   N  D       TTTTT EEEEE SSSSS TTTTT
c     E       NN  N  D D       T   E     S       T
c     EEE     N N N  D  D      T   EEE    S      T
c     E       N  NN  D D       T   E        SS    T
c     EEEEE   N   N  D         T   EEEE  SSSSS   T
c
c 1631 bypass prints and some checks on energy
c
1631  continue
c
c Stop here if your intentions are only to run the test
c
      if(icont.eq.0) stop
c
      icart=1
c
c
      drodc=0.1*drod
c
c
      do 467 ipod=1,npod
      isha(ipod)=ish
467   bual(ipod)=bu
c
c
      do 1391 ipod=1,npod
c
c start loop of designs by fixing first p/d
c up to statement # 1391
c
      pod=podi(ipod)
      BU=bual(ipod)
      ishape=isha(ipod)
c
c Abu and Abund(cm2) area of 19 rods bundle with and without the duct
      Abu=3.4641016*drodc**2*(pod*2.73205081-0.5+tduc/drod)**2
      Abund=3.4641016*drodc**2*(pod*2.73205081-0.5)**2
c duct fraction
      fduct=(Abu-Abund)/Abu
c f1,2,3 are flow fractions for cell types 1,2,3(with respect to acell1)
      x=pod
      f1=1.-0.906900*(1.+(x-1.)**2/3.)/x**2
      f2=2.30940*((x-.5)*x-.392699*(1.+.5*(x-1.)**2))/x**2
      f3=2.30940*(.577350*(x-.5)**2-0.130900)/x**2
c  area of cell(cm2) is acell
      pitch=pod*drodc
      acell=0.433013*pitch**2
c a1,a2,a3(cm2) are area flow of cell types 1,2,3
      a1=f1*acell
      a2=f2*acell
      a3=f3*acell
      At=24.*a1+12.*a2+6.*a3
c flow fraction
      fLi=At/Abu
      Arod=pee*rrod**2*19.
c rods fraction
      frod=Arod/Abu
c spacing wire fraction
```

```
      fwire=(Abund-Arod-At)/Abu
c core density, averaged within bundle ,is rocore
      rocore=fduct*RoAst+fLi*roLi+frod*rorod+fwire*RoAst
c
      r=RRR
      h=HHH
      itest=0
c
c check if you are requesting too little energy
c
      iflamin=0
      Emino=Emin(nenr,ipod)
       if(Ereq.ge.Emino) go to 337
      iflamin=1
      write(8,5133) pod,Ereq,BU,drr,dr1,dr2,dkeol,dkstr
5133  format(//' ATTENTION'/' Sampling p/d=',f8.4/
     *                      ' You requested Mwdays',f10.2/
     *                      ' And burn-up(%)',f5.2/
     *' With radial Be reflector thickness(cm)',f6.2/
     *'  "   top     "     "       "           ',f6.2/
     *'  "   bottom  "     "       "           ',f6.2/
     *'            End of Life delta k(%)     ',f6.2/
     *'            Structural  delta k(%)     ',f6.2//
     *' Trying to satisfy reactivity balance,it happens your core'/
     *' is too large(and it would produce more energy you need).'/
     *' You might:'/
     *' 1) Reduce the input burn-up, and/or'/
     *' 2) Increase the reflector thickness,and/or'/
     *' 3) Reduce delta k at end of life, and/or'/
     *' 4) Reduce delta k structural')
      if(ichoose.eq.1) write(8,1424)
1424  format(//' I choose=1, Program stops')
      if(ichoose.eq.1) stop
        write(8,1426)
1426  format(//' Program will change burn-up for this p/d design'/
     *' and choose ishape=1'//
     *'          BU       Mwday'/)
      icart=0
      itest=1
c
c Recompute Energy available for ideal shape Emin() by changing burnup
c
      enr=enri(nenr)
      ishape=1
4644  Ene=Balance(ibu,enr,pod)
      Emin(nenr,ipod)=Ene
      write(8,1425) BU,Ene
1425  format(2f12.2)
      if(Ereq.ge.Ene) go to 4633
      BUA=BU
      BU=BU-1.
      if(BU.le.0.) BU=BUA/2.
      BUal(ipod)=bu
      go to 4644
4633  continue
      itest=0
337    continue
c
c check if there is enough energy for ideal shape(ishape=1)
c
      iwae=0
      Emaxo=Emin(1,ipod)
      if(Ereq.le.Emaxo) go to 357
      iwae=1
c
c set the warning flag iwae=1 for potential strange shapes:cigar/pancake
c and calculate R1,R2,R3,R4 and H1,H2,H3,H4 which are the limits for R
```

```
c and H for the input conditions(reactivity,required Energy) in the
c bifurcation region
c
      itest=1
      ishape=2
      enr=enri(nenr)
      Das=Ras(nenr,ipod)
      Dop=Rop(nenr,ipod)
      call Size(Das,Dop,Ereq,enr,pod,R1,R4)
      enrr14=enr
      hr4=h
      hr1=Vcor/(pee*r1**2)
      vr14=Vcor/1000.
      enr=enri(1)
      Das=Ras(1,ipod)
      Dop=Rop(1,ipod)
      call Size(Das,Dop,Ereq,enr,pod,R2,R3)
      enrr23=enr
      hr3=h
      hr2=Vcor/(pee*R2**2)
      vr23=Vcor/1000.
      ishape=3
      enr=enri(nenr)
      Das=Has(nenr,ipod)
      Dop=Hop(nenr,ipod)
      call Size(Das,Dop,Ereq,enr,pod,H1,H4)
      enrh14=enr
      rh4=R
      rh1=sqrt(Vcor/(pee*H1))
      vh14=Vcor/1000.
      enr=enri(1)
      Das=Has(1,ipod)
      Dop=Hop(1,ipod)
      call Size(Das,Dop,Ereq,enr,pod,H2,H3)
      enrh23=enr
      rh3=R
      rh2=sqrt(Vcor/(pee*H2))
      vh23=Vcor/1000.
      write(8,369)pod,Ereq,Emaxo,R1,R2,R3,R4,enrr14,vr14,hr1,hr4,enrr23
     *,vr23,hr2,hr3
369   format(//' ATTENTION:'/
     *' Designs with p/d=',f8.5/
     *' You requested Mwdays',f10.2,' larger than',f10.2/
     *' which is the largest energy available for ideal shape.'/
     *' Strange shapes(cigar/pancake) can appear as solutions,'/
     *' the following designs are possible:'/
     *'  ishape=2 with R(cm) between'/
     *'                              R1',f8.2,' and R2',f8.2/
     *'               or  R(cm) between'/
     *'                              R3',f8.2,' and R4',f8.2/
     *'R1 and R4 corresponds to Enr=',f8.3,' with core Vol(L)',f9.1/
     *'          with H(cm),Respectively=',2f10.2/
     *'R2 and R3 corresponds to Enr=',f8.3,' with core Vol(L)',f9.1/
     *'          with H(cm),Respectively=',2f10.2/)
      write(8,412) H1,H2,H3,H4,enrh14,vH14,rh1,rh4,enrh23,vh23,rh2,rh3
412   format(
     *' Or  ishape=3 with H(cm) between'/
     *'                              H1',f8.2,' and H2',f8.2/
     *'                  or  H(cm) between'/
     *'                              H3',f8.2,' and H4',f8.2/
     *'H1 and H4 corresponds to Enr=',f8.3,' with core Vol(L)',f9.1/
     *'          with R(cm),Respectively=',2f10.2/
     *'H2 and H3 corresponds to Enr=',f8.3,' with core Vol(L)',f9.1/
     *'          with R(cm),Respectively=',2f10.2/)
      write(8,7133) BU,drr,dr1,dr2,dkeol,dkstr
7133  format(/' The previous data corresponds to your input conditions'/
     */' Burn-up(%)',f6.2/
```

```
     *' Radial Be reflector thickness(cm)',f6.2/
     *'  Top     "     "       "          ',f6.2/
     *'  Bottom  "     "       "          ',f6.2/
     *'        End of Life delta k(%)     ',f6.2/
     *'        Structural  delta k(%)     ',f6.2//
     *' If awkward values of Radius/Height appear YOU MIGHT:'///
     *' 1) Increase the burn-up, and/or'/
     *' 2) Decrease the reflector thickness,and/or'/
     *' 3) Increase delta k at end of life, and/or'/
     *' 4) Increase delta k structural'//
     *' #1 is very effective,#2,3 and 4 not so much'/
     *' If problem persist better try other kind of reactor'//)
      itest=0
      ishape=ish
      R=RRR
      H=HHH
357   continue
      if(ishape.eq.1.or.iwae.eq.1) go to 434
c
c begin check of input R or H if E requested is not in bifurcation
c region
c
      itest=1
      if(ishape.eq.2)  then
      enr=enri(nenr)
      Das=Ras(nenr,ipod)
      Dop=Rop(nenr,ipod)
      call Size(Das,Dop,Ereq,enr,pod,R1,R2)
      R=RRR
      if(R1.lt.R.and.R.lt.R2) go to 443
      if(ichoose.eq.1) write(8,4551)pod,ichoose,R,R1,R2
4551  format(//' Sampling p/d=',f6.3/
     *//' I choose Flag=',i5,' Your R=',f8.2/
     *       '                          Outside Possible Range',2f10.2/
     *       ' Program stop'//)
      if(ichoose.eq.1) stop
      Rin=R
      R=0.5*(R1+R2)
      write(8,4552) pod,ichoose,Rin,R1,R2,R
4552  format(//' Sampling p/d=',f6.3
     */' I choose Flag=',i5,' Your R=',f8.2/
     *       '                          Outside Possible Range',2f10.2/
     *       ' Program continues with average, R=',f8.2//)
443   continue
      else
      enr=enri(nenr)
      Das=Has(nenr,ipod)
      Dop=Hop(nenr,ipod)
      call Size(Das,Dop,Ereq,enr,pod,H1,H2)
      H=HHH
      if(H1.lt.H.and.H.lt.H2) go to 454
      if(ichoose.eq.1) write(8,4553) pod,ichoose,H,H1,H2
4553  format(//' Sampling p/d=',f6.3
     */' I choose Flag=',i5,' Your H=',f8.2/
     *       '                          Outside Possible Range',2f10.2/
     *       ' Program stop'//)
      if(ichoose.eq.1) stop
      Hin=H
      H=0.5*(H1+H2)
      write(8,4554) pod,ichoose,Hin,H1,H2,H
4554  format(//' Sampling p/d=',f6.3
     */' I choose Flag=',i5,' Your H=',f8.2/
     *       '                          Outside Possible Range',2f10.2/
     *       ' Program continues with average, H=',f8.2//)
454   continue
      endif
      itest=0
```

```
434   continue
      if(iwae.eq.0) go to 1004
      go to(1001,1002,1003),ishape
1001  continue
      if(ichoose.eq.1) write(8,455) ichoose
455   format(//'I choose Flag=',i3,' Not enough available energy'/
     *        '                          for ishape=1 option, program stop')
      if(ichoose.eq.1) stop
      ishape=2
      R=R3*1.01
      write(8,4555) R
4555  format(//
     *' Not enough energy for ishape=1(input), program continues with'/
     *' with R=',f8.2,' toward pancake shape'//)
      go to 1004
1002  continue
      if(R1.lt.R.and.R.lt.R2) go to 1004
      if(R3.lt.R.and.R.lt.R4) go to 1004
      if(ichoose.eq.1) write(8,4556) ichoose,R
4556  format(//
     *'I choose Flag=',i3,' Your input R=',f8.2,' Out Possible Range'/
     *' Program stop')
      if(ichoose.eq.1) stop
      Rin=R
      R=R3*1.01
      write(8,4557) Rin,R
4557  format(//
     *' Your input R=',f8.2,' is outside possible range, program'/
     *' continues with R=',f8.2,' toward pancake shape'//)
      go to 1004
1003  continue
      if(H1.lt.H.and.H.lt.H2) go to 1004
      if(H3.lt.H.and.H.lt.H4) go to 1004
      if(ichoose.eq.1) write(8,4558) ichoose,H
4558  format(//
     *'I choose Flag=',i3,' Your input H=',f8.2,' Out Possible Range'/
     *' Program stop')
      if(ichoose.eq.1) stop
      Hin=H
      H=H2*0.99
      write(8,4559) Hin,H
4559  format(//
     *' Your input H=',f8.2,' is outside possible range, program'/
     *' continues with H=',f8.2,' toward pancake shape'//)
1004  continue
c
c
c enr1 (higher enrichment) has balance<0.
      ibu=0
      enr1=enri(nenr)
      bal1=Balance(ibu,enr1,pod)
      if(ibu.eq.1) write(8,597)
597   format(' Stop trying to bracket demand')
      if(ibu.eq.1) stop
c loock for enrichment that produces balance>0.
      if(ishape.eq.1.or.iwae.eq.1) then
      enr2=enri(1)
      bal2=Balance(ibu,enr2,pod)
      if(ibu.eq.1) write(8,597)
      if(ibu.eq.1) stop
      else
      it=0
      enr2=enr1
790   enr2=enr2/der
      it=it+1
      bal2=Balance(ibu,enr2,pod)
      if(ibu.eq.1) write(8,597)
```

```
      if(ibu.eq.1) stop
      if(bal2.gt.0.) go to 890
      if(it.lt.itmax) go to 790
      write(8,794) it,enr2,pod
794   format(' stop trying to find Bal>0. iterations=',i5,' enr=',f10.3/
     *'p/d=',f10.3)
      stop
890   continue
      endif
c
c if we successfully arrived to this point,it means we bracket the demand
c Then:
c bisect  Balance until Eavailable=Erequested
c
      it=0
145   enra=0.5*(enr1+enr2)
      Erema=Balance(ibu,enra,pod)
      if(ibu.eq.1) write(8,597)
      if(ibu.eq.1) stop
      test=abs(Erema)/Ereq
      if(test.lt.eps) go to 144
      it=it+1
      if(iwrte.eq.1.and.it.eq.1) write(8,219) pod,bu
219   format(//'Sampling p/d=',f6.3,' BU(%)=',f7.2//
     *'Iterations on enrichment'/
     *'  it',7x,' enr1',8x,'enra',7x,' enr2',8x,'bal1',7x,'Erem',
     *8x,'bal2')
      if(iwrte.eq.1) write(8,218)it,enr1,enra,enr2,bal1,Erema,bal2
218   format(i4,3f12.6,3f12.3)
      call bisect(enr1,enr2,enra,bal1,bal2,Erema,iwa)
      if(it.le.itmax.and.iwa.eq.0) go to 145
144   continue
      if(it.gt.itmax.or.iwa.eq.1) then
      write(8,146) enr1,bal1,enr2,bal2,it,iwa
146   format(///' Problems trying to bisect  Balance'/
     *          ' enr1=',e12.4,' bal1=',e12.4/
     *          ' enr2=',e12.4,' bal2=',e12.4/
     *'                              iteration=',i5/
     *'                                    iwa=',i5/)
      stop
      else
      continue
      endif
c
c successful end of reactor search, calculate  masses
c change R such area fit integer number of bundles
      Nbu=pee*R**2/Abu
      Rbu=pee*R**2/Abu
      Rnbu=nbu
      if((Rbu-Rnbu).gt.0.5) Nbu=Nbu+1
      R=sqrt(Nbu*Abu/pee)
c
c recompute H
      H=Vcor/(pee*R**2)
c
c compute mass core+pressure vessel+reflector=Greactor
c
      Volpv=0.
      Gpv=0.
      Volpv=2.*pee*(R+dpv)**2*dpv
      Volpv=Volpv+pee*((R+dpv)**2-R**2)*H
      Gpv=Volpv*RoAst
      Rpv=R+dpv
      Hpv=H+dpv
      Vref=pee*((Rpv+drr)**2-Rpv**2)*Hpv
      Vref=Vref+pee*(Rpv+drr)**2*(dra1+dra2)
      Gref=Vref*roOBe
```

```
      GReactor=Vcor*rocore+Gpv+Gref
c
c compute delta pressure across core=delp
c
      Hact=H
      call thermoli(1)
c remember some data for future selection
      enrd(ipod)=enra
      Rd(ipod)=R
      Hd(ipod)=H
      dpvd(ipod)=dpv
      GRea(ipod)=GReactor
      delp(ipod)=deltapa
      dkbup(ipod)=dkbu
      dkpvd(ipod)=dkpv
      rkbare(ipod)=rkre
      dkobe(ipod)=dkbe
      Nbun(ipod)=Nbu
      u5md(ipod)=U5m/1000.
      isha(ipod)=ishape
1391  continue
c
c at 1391 end of loop on p/d
c
c Now choose design using Greactor and delta p as criteria
c
      ich=1
      Rmref=GRea(1)/1000.
      write(8,849) delpr
849   format(//'Sampling of Designs'/
     *'Mass=Masses of core,pressure vessel and reflector'/
     *' Max delta p(Pa) acceptable=',e13.4//
     *'  I   p/d   Enr    BU  Mass(Kg)    U5(Kg)   DeltaP(Pa) coreR(cm)
     *coreH(cm) ishape'/)
      do 783 ipod=1,npod
      delpa=delp(ipod)
      Rmass=GRea(ipod)/1000.
      enro=enrd(ipod)
      buon=bual(ipod)
      rdo=Rd(ipod)
      Hdo=Hd(ipod)
      U5m=U5md(ipod)
      ishu=isha(ipod)
      write(8,855)ipod,podi(ipod),enro,buon,Rmass,U5m,delpa,Rdo,Hdo,ishu
855   format(i3,2f6.3,f6.2,2f10.1,e13.4,2f10.2,i3)
      if(delpa.gt.delpr) go to 783
      if(Rmass.lt.Rmref) ich=ipod
      if(Rmass.lt.Rmref) Rmref=Rmass
783   continue
c
c chosen design is ipod=ich
c unless you had decided other thing
      if(ichoode.ge.1.and.ichoode.le.5) ich=ichoode
      R=Rd(ich)
      H=Hd(ich)
      dpv=dpvd(ich)
      enr=enrd(ich)
      pod=podi(ich)
      dkbu=dkbup(ich)
      dkpv=dkpvd(ich)
      rkre=rkbare(ich)
      dkbeo=dkobe(ich)
      Nbu=Nbun(ich)
      bu=bual(ich)
      Vcor=pee*R**2*H
      ishape=isha(ich)
      write(8,11) tit
```

```
c
      write(8,51) Pow,D,Ereq,BU,drr,dra1,dra2,eps
c
51    format(//' INPUT TO THE PROGRAM'//' Power(Mw)',f8.2,' Core life',f
     *8.3,' years'/' Energy Released(Mwday)',f10.2,' Burn-up(%)',f6.2//
     *' Be reflector, radial thickness(cm)',f6.2/
     *'               top         "       ',f6.2/
     *'               bottom      "       ',f6.2//
     *' The following design can produce that energy within the relative
     *'/'numerical precision of',e12.4)
c
      if(ishape.eq.1) write(8,317) xop
317   format(/' Minimum Core Volume Option Chosen: R/H=',f9.6/)
c
      if(ishape.eq.2) write(8,318) R
318   format(/' Core Radius Option Chosen: R(cm)=',f7.2/)
c .
      if(ishape.eq.3) write(8,319) H
319   format(/' Core Height Option Chosen: H(cm)=',f7.2/)
c
      dkfp=dkbu
      dkreq=dkfp+dkstr+dkeol+dkpv
      dkbeo=dkbeo/100.
      refec=rkre+dkbeo
c
c
      write(8,52) dkreq,dkfp,dkpv,dkstr,dkeol,refec,rkre,dkbeo
c
52    format(' And it has the following reactivity balance'/
     *' Delta k required(%)',f6.2,' with components'/
     *'  Fission Products(%)',f6.2/
     *'   Pressure Vessel(%)',f6.2/
     *'        Structural(%)',f6.2/
     *'       End of Life(%)',f6.2/
     *' keff at BOL(No structural mat)',f7.4, ' Components'/
     *'                    Bare Reactor',f7.4/
     *'                       Reflector',f7.4//)
c
c
      write(8,521) enr,pod
c
521   format(' Enrichment=',f8.4,' Pitch/Diameter=',f6.2)
c
      Rad=R
c
c call control to design control drums
c
      dvp=dpv
      Call Control
c
c
      nrod=Nbu*nrodbu
      Vcore=Vcor/1000.
c  densities inside the pin
      AU=enr*AU5+(1.-enr)*AU8
      AUN=AU+AN
      roN=roun*AN/AUN
      roU=roun*AU/AUN
      rou5=enr*AU5*rou/AU
      rou8=(1.-enr)*AU8*roU/AU
      cU5=roU5*Av/AU5
      cU8=roU8*Av/AU8
      Vact=nrod*pee*(drodc/2.)**2*H*fpin/1000.
      u5m=Vact*roU5
      u8m=Vact*rou8
      Pum=0.
      Fpm=0.
```

```
      write(8,30) Vcore,Vact,R,H,nrod,nbu,nrodbu
c
30    format(' NUMBER DENSITIES AND MASSES OF FUEL PIN REACTOR'/
     *       '          GENERAL DESCRIPTION'/
     *'Core Volume(L)',f10.2,' Active(UN) Volume(L)',f10.2/
     *' Core Radius(cm)',f10.2,' Height(cm)',f10.2/
     *' Number of rods',i6,' Bundles',i6,'(',i3,'rods per bundle)'//)
c
c
      bup=0.01*bu
      Call bupo(bup,1.50,cu5,cu8,dkbu)
c
c see: 1.50 and dkbu are dummies
c
      rou5e=c5e*AU5/Av
      rou8e=c8e*AU8/Av
      roPue=c9e*APu/Av
      roFpe=cfpe*0.5*AU/Av
      U5me=rou5e*Vact
      U8me=rou8e*Vact
      Pume=roPue*Vact
      Pup=(p9*APu/Av)*Vact
      Fpme=roFpe*Vact
      ftin=ftin*1.e+24
      fluxn=ftin/tiempo
      write(8,5211)fluxn,D,ftin
5211  format(
     *'/' Neutron Flux(1/cm2sec)(core and time average)          ',e12.4/
     *' '     "    Fluence(1/cm2)  "       "   after',f6.2,' years',e12.4)
c
      write(8,986) u5m,u5me,u8m,u8me,pum,pume,Pup,fpm,fpme
986   format(//' Masses(Kg) of Fissile Materials and Fission Products'//
     *'                    BOL       EOL'/
     *'     U-235',2f10.3/
     *'     U-238',2f10.3/
     *'    Pu-239',3f10.3,' Kg produced'/
     *'Fiss Prod ',2f10.3)
c
      write(8,40) drod,dc,dl,dHe,dpin
c
40    format('             FUEL RODS'/
     *' Rod diameter(mm) is',f7.4/' Astar Alloy clad thickness,mm',f7.4/
     *' W Liner,mm',f7.4/' He gap,mm',f7.4/' UN pin diameter,mm',f7.4)
c
c
      x=pod
      pitch=x*drodc
      fcool=1.-0.90689968*(1.+(x-1.)**2/3.)/x**2
      frod=0.90689968/x**2
      fwire=1.-fcool-frod
      write(8,50) fcool,fwire,frod
c
50    format(//' Volume fractions in Fuel Cell #1'/'      Coolant',f7.4
     *                                          /' Spacing Wire',f7.4
     *                                          /' Fuel Rod    ',f7.4)
c
c
c
      Rco=0.52503757*pitch
      ALi=ALi6*xLi6+ALi7*xLi7
      coLi=roLi*Av/ALi
      coLi6=coLi*xLi6
      coLi7=coLi*xLi7
      roTa=roAst*xTa
      coTa=roTa*Av/ATa
      roTun=roAst*xW
      coTun=roTun*Av/AW
```

```
      cW2c=coTun*xW2
      cW3c=coTun*xW3
      cW4c=coTun*xW4
      cW6c=coTun*xW6
      roRe=roAst*xRe
      coRe=roRe*Av/ARe
      cRe185=coRe*xRe5
      cRe187=coRe*xRe7
      roHf=roAst*xHf
      cHf=roHf*Av/AHf
      fclac=(fclad+fwire)*frod
      Rrodw=Rrod*sqrt(1.+fwire/frod)
      coW=roW*Av/AW
      cW2l=coW*xW2
      cW3l=coW*xW3
      cW4l=coW*xW4
      cW6l=coW*xW6
      flic=fliner*frod
      pHep=pHe*1000000.
      roHe=0.001*rho(pHep,THe)
      coHe=roHe*Av/AHe
      fHec=fHe*frod
      cN=roN*Av/AN
      fpinc=fpin*frod
      write(8,60) Rco,coLi6,fcool,coLi7,
     *            Rrod,coTa,fclac,
     *            Rrodw,
     *                cW2c,cW3c,cW4c,cW6c,
     *                cRe185,cRe187,
     *                cHf,
     *            Rclad,cW2l,flic,
     *            cW3l,cW4l,cW6l
      write(8,1160)
     *            Rliner,coHe,fHec,
     *            Rpin,cU5,fpinc,
     *            cU8,cN
c
60     format(//
     *' Fuel Cell type 1 (Number Densities 10**24 atoms/cm3)'/
     *' Spacing Wire is diluted in clad'//
     *'Region   Radius(cm) Isotope Concentration Volumetric Fraction'//
     *'Coolant  ',f10.4,'     Li6',f12.8,f10.6/
     *'  "      ',10x,  '     Li7',f12.8/
     *'Clad     ',f10.4,'   Ta181',f12.8,f10.6/
     *'  "      ',f10.4,'(diluting spacing wire)'/
     *'  "      ',10x,  '    W182',f12.8/
     *'  "      ',10x,  '    W183',f12.8/
     *'  "      ',10x,  '    W184',f12.8/
     *'  "      ',10x,  '    W186',f12.8/
     *'  "      ',10x,  '   Re185',f12.8/
     *'  "      ',10x,  '   Re187',f12.8/
     *'  "      ',10x,  '      Hf',f12.8/
     *'Liner    ',f10.4,'    W182',f12.8,f10.6/
     *'  "      ',10x,  '    W183',f12.8/
     *'  "      ',10x,  '    W184',f12.8/
     *'  "      ',10x,  '    W186',f12.8)
1160   format(
     *'He Gap   ',f10.4,'      He',f12.8,f10.6/
     *'Fuel Pin ',f10.4,'    U235',f12.8,f10.6/
     *' "    "  ',10x,  '    U238',f12.8/
     *' "    "  ',10x,  '     N14',f12.8//)
c
c AVERAGES IN BUNDLE:
c
c Abu and Abund(cm2) area of 19 rods bundle with and without the duct
      Abu=3.4641016*drodc**2*(pod*2.73205081-0.5+tduc/drod)**2
      Abund=3.4641016*drodc**2*(pod*2.73205081-0.5)**2
```

```
c duct fraction
      fduct=(Abu-Abund)/Abu
c f1,2,3 are flow fractions for cell types 1,2,3(with respect to acell1)
      f1=1.-0.906900*(1.+(x-1.)**2/3.)/x**2
      f2=2.30940*((x-.5)*x-.392699*(1.+.5*(x-1.)**2))/x**2
      f3=2.30940*(.577350*(x-.5)**2-0.130900)/x**2
c  area of cell(cm2) is acell
      pitch=pod*drodc
      acell=0.433013*pitch**2
c a1,a2,a3(cm2) are area flow of cell types 1,2,3
      a1=f1*acell
      a2=f2*acell
      a3=f3*acell
      At=24.*a1+12.*a2+6.*a3
c flow fraction
      fLi=At/Abu
      Arod=pee*(drodc/2.)**2*19.
c rods fraction
      frod=Arod/Abu
c spacing wire fraction
      fwire=(Abund-Arod-At)/Abu
c core density, averaged within bundle ,is rocore
      rocore=fduct*RoAst+fLi*roLi+frod*rorod+fwire*RoAst
c
      fclw=fwire+frod*fclad+fduct
      flin=frod*fliner
      fpinb=frod*fpin
c
      Vact=Vcore*fpinb
c
      coLi6=coLi6*fLi
      coLi7=coLi7*fLi
      coTa=coTa*fclw
      cW2=cW2c*fclw+cW2l*flin
      cW3=cW3c*fclw+cW3l*flin
      cW4=cW4c*fclw+cW4l*flin
      cW6=cW6c*fclw+cW6l*flin
      cRe185=cRe185*fclw
      cRe187=cRe187*fclw
      cHf=cHf*fclw
      coHe=coHe*frod*fHe
      fHec=fHe*frod
      cU5=cU5*fpinb
      cU8=cU8*fpinb
      cN=cN*fpinb
      write(8,61) frod,fwire,fLi,fduct
      write(8,65) coLi6,coLi7,
     *            coTa,
     *                cW2,cW3,cW4,cW6,
     *                cRe185,cRe187,
     *                cHf,
     *            coHe,
     *            cU5,
     *            cU8,cN
c
61    format(//'AVERAGES IN BUNDLE(INCLUDING DUCT)'/
     *' Volumetric Fractions:         Rods',f7.4/
     *'                       Spacing Wire',f7.4/
     *'                         Li Coolant',f7.4/
     *'                              Duct',f7.4//)
65    format(
     *' Isotope Concentration '/
     *'(Number Densities 10**24 atoms/cm3)'//
     *'     Li6',f12.8/
     *'     Li7',f12.8/
     *'   Ta181',f12.8/
     *'    W182',f12.8/
```

```
     *'    W183',f12.8/
     *'    W184',f12.8/
     *'    W186',f12.8/
     *'   Re185',f12.8/
     *'   Re186',f12.8/
     *'      Hf',f12.8/
     *'      He',f12.8/
     *'    U235',f12.8/
     *'    U238',f12.8/
     *'     N14',f12.8//)
      rLi=(coLi6+coLi7)/cU5
      write(8,92) rLi
c
92    format(///' Moderation by Li, Li/U235=',f10.4)
c
c compute masses and volumes inside the core
c
      Gcore=rocore*Vcore
      Vrod=Vcore*frod
      Grod=Vrod*rorod
      frodm=Grod/Gcore
      VLi=Vcore*fLi
      GLi=VLi*roLi
      fLim=GLi/Gcore
      Vduct=Vcore*fduct
      Gduct=Vduct*roAst
      fducm=Gduct/Gcore
      Vwire=Vcore*fwire
      Gwire=vWire*roAst
      fwim=Gwire/Gcore
      write(8,1093) Vrod,frod,Grod,frodm,
     *              Vli,fLi,GLi,fLim,
     *              Vduct,fduct,Gduct,fducm,
     *              Vwire,fwire,Gwire,fwim,
     *              Vcore,Gcore
1093  format(//
     *' VOLUME(L) AND MASSES(KG) INSIDE THE CORE'//
     *'Part      Volume    Fraction  Masses    Fraction'//
     *'Rods     ',2(f10.2,f10.6)/
     *'Coolant  ',2(f10.2,f10.6)/
     *'Duct     ',2(f10.2,f10.6)/
     *'Spacing  ',2(f10.2,f10.6)/
     *'Total    ',2(f10.2,10x))
c
c compute mass core+pressure vessel+reflector=Greactor
c
      Volpv=2.*pee*(R+dpv)**2*dpv
      Volpv=Volpv+pee*((R+dpv)**2-R**2)*H
      Volpv=Volpv/1000.
      Gpv=Volpv*RoAst
      Rpv=R+dpv
      Hpv=H+dpv*2.
      Vref=pee*((Rpv+drr)**2-Rpv**2)*Hpv
      Vref=Vref+pee*(Rpv+drr)**2*(dra1+dra2)
      Vref=Vref/1000.
      Gref=Vref*roOBe
      GReactor=Gcore+Gpv+Gref
      Vreactor=Vcore+Volpv+Vref
      fvc=Vcore/Vreactor
      fvpv=Volpv/Vreactor
      fvr=Vref/Vreactor
      fmc=Gcore/GReactor
      fmpv=Gpv/GReactor
      fmr=Gref/GReactor
      write(8,1095) Vcore,fvc,Gcore,fmc,
     *              Volpv,fvpv,Gpv,fmpv,
     *              Vref,fvr,Gref,fmr,
```

```
     *              VReactor,GReactor
1095  format(//
     *' VOLUME(L) AND MASSES(KG) OF REACTOR'//
     *'Part      Volume    Fraction  Masses    Fraction'//
     *'Core     ',2(f10.2,f10.6)/
     *'Press Ves',2(f10.2,f10.6)/
     *'Reflector',2(f10.2,f10.6)/
     *'Total    ',2(f10.2,10x))
      Hrea=Hpv+dra1+dra2
      Rrea=Rpv+drr
c
c calculation of neutron shielding
c
c compute fast source Sn
c
      Sn=Pow*1.e+6*rnu/wfi
c
c compute shielding by core,reflector and pressure vessel: fsh
c concentrations smothered in Vol=Vcore+Volpv+Vref
c
      coLi=(coLi6+coLi7)*fvc
      coTa=coTa*fvc+xTa*fvpv*roAst*Av/ATa
      coW=(cW2+cW3+cW4+cW6)*fvc+xW*fvpv*roAst*Av/AW
      coRe=(cRe185+cRe186)*fvc+xRe*fvpv*roAst*Av/ARe
      coHf=cHf*fvc+              xHf*fvpv*roAst*Av/AHf
      coHe=coHe*fvc
      coU5=cU5*fvc
      coU8=cU8*fvc
      coU=coU5+coU8
      coN=cN*fvc
      robe=fvr*roobe*av/(AO+ABe)
      coO=robe
      coBe=robe
      sigre=coBe*remBe+remW*(coTa+coW+coRe+coHf)+coU*remU+
     *     +remLi*(coLi+coHe)+(coN+coO)*remO
      Req=(750.*Vreactor/pee)**0.33333333
      sire=sigre*Req*2.
      fsh=3.*(0.5-(1.-(1.+sire)*exp(-sire))/sire**2)/sire
      Snsh=Sn*fsh
c Fast fluence without shielding is flufa,at distance dist
      dist=100.*zLoad+0.5*Hrea
      flufa=Snsh/(4.*pee*dist**2)
      flufa=flufa*tiempo
c
c Calculate Neutron Shield for zero gamma Shield
c
      tne0=shieldn(flufa,fastf,ishn,0.00,igam)
      if(ishin.eq.0) tne0=0.
c
c Now, Calculation Gamma Source Sogam()
c
      Call Gamso(Sn,Sogam,PrU,fvr)
c
c First, Gamma Shield with no Neutron Shield
c
      Call Shieldga(Sogam,Dose,Doses,Dosew,tiempo,tgam0,0.,igam1,ishn,0)
c
c Now, Coupled with n Shield
c
      Call Shieldga(Sogam,Dose,Doses,Dosew,tiempo,tgam,tne,igam2,ishn,1)
c
      if(igam2.eq.0) tne=tne0
      if(igam2.eq.0) tgam=0.
c
c Volume and Masses of shieldings
c
      Tgt=Tan((100.*rload-Rrea)/(100.*zload+Hrea))
```

```
      rr1=Hrea*Tgt+Rrea
      rr2=(Hrea+tgam)*Tgt+Rrea
      rr3=(Hrea+tgam+tne)*Tgt+Rrea
      h1=rr1/Tgt
      h2=rr2/Tgt
      h3=rr3/Tgt
      V1=pee*RR1**2*h1*0.33333333
      V2=pee*rr2**2*h2*0.33333333
      V3=pee*rr3**2*h2*0.33333333
      Volg=(v2-v1)/1000.
      Voln=(v3-v2)/1000.
      Gg=RoW*Volg
      if(ishn.eq.1) Ron=RoLiS
      if(ishn.eq.2) Ron=RoBC
      Gne=Voln*Ron
      Vols=Vreactor+Voln+Volg
      Gsis=Greactor+Gne+Gg
      fresv=100.*Vreactor/Vols
      fnev=100.*Voln/Vols
      fgv=100.*Volg/Vols
      fresg=100.*Greactor/Gsis
      fneg=100.*Gne/Gsis
      fgg=100.*Gg/Gsis
c
c
      write(8,789) Vreactor,fresv,Voln,fnev
c
789   format(
     *'         VOLUMES AND MASSES OF REACTOR AND SHIELDINGS'//
     *'           Volume(L) Fraction(%)'/
     *' Reactor ',f10.2,f10.2/
     *' N Shield',f10.2,f10.2)
c
c
      write(8,796) Volg,fgv
796   format(' G Shield',2f10.2)
c
c
      write(8,797) Vols
c
797   format(' Total    ',f10.2//)
c
c
      write(8,798) Greactor,fresg,Gne,fneg
c
798   format(
     *'            Mass(Kg) Fraction(%)'/
     *' Reactor ',f10.2,f10.2/
     *' N Shield',f10.2,f10.2)
c
c
      write(8,796) Gg,fgg
c
c
      write(8,806) Gsis
c
806   format(' Total     ',f10.2//)
c
c
      if(igam2.eq.0) write(8,808)
808   format(' There were numerical problem with the calculation of'/
     *' gamma shield, calculations skipped'/)
      if(ishig.eq.0) write(8,809)
      if(ishin.eq.0) write(8,810)
809   format(/' No need of gamma shield'/)
810   format(/' No need of neutron shield'/)
c
```

```
      rr21=rr2
c
c
      write(8,811)  zload,rload,tgam,rr1,rr2,tne,rr21,rr3
c
811   format(
     *'              DIMENSION OF THE SHIELDINGS'/
     *' LOAD AT Z(M)=',F8.2,' WITH RADIUS(M)=',F8.2/
     *' G Shield Thickness(cm)',f8.2,' With Radius(cm)',2f8.2/
     *' N   "          "     " ',f8.2,'   "     "    " ',2f8.2)
c
c
      write(8,758) tempv,pin,D
c
758   format(//'EVALUATION OF PRESSURE VESSEL'/
     *'Evaluation at T(K)=',f8.2,' PRESS(MPa)=',f8.2/
     *'          and Full Power Life(y)',f8.2//)
      write(8,1366) dpv
1366  format('Pressure Vessel Thickness(cm)',f6.2//)
c
c
      if(ipresv.eq.0) write(8,626)
626   format(' You have problems to contain your reactor, pressure Vesse
     *l'/'calculations skipped, not included in shielding or balance'//)
c
c
c
      flush=flufa
      if(ishin.eq.1) flush=fastf
      if(ishig.eq.1.and.igam2.ne.0) flush=flufa*aten
c
c
      write(8,757) Sn,fsh,Req,sigre,D,flufa,flush,fastf,Zload,tne,tne0
c
757   format(
     *'      DETAILS OF THE SHIELDINGS'//
     *       ' Neutron Source(1/sec)',e12.4/' Self Shielding by Reactor
     *Materials',e12.4/' Equivalent R(cm)=',f8.2,' Sigma Removal',f12.6/
     *f6.2,' Years Fast Fluence(1/cm2) without Any Shielding',e12.4
     */    '   "      "     "    "     with (n,gamma)    "  ',e12.4
     */    '   "      "     "    "    Requested             ',e12.4
     *'/' All  at',f6.2,' meters'
     *'/'Neutron Shielding Thickness(cm)',f10.2/
     *'Neutron Shielding Thickness if no W is present, cm',f10.2//)
c
c
      if(ishn.eq.1) write(8,767)
767   format(' NEUTRON SHIELDING: LiH-Stainless Steel Matrix'//)
c
c
      if(ishn.eq.2) write(8,768)
768   format(' NEUTRON SHIELDING: B4C'//)
c
c
      write(8,771)
771   format(' TOTAL PRODUCTION OF GAMMA RAYS(1/sec)'//
     *'GROUP    1          2          3          4          5          6
     *    7'/
     *'    0-1Mev       1-2        2-3        3-5        5-7        7-9
     *   >9'/)
c
c
      write(8,772) (sogam(i),i=1,7)
c
772   format(1P7e10.3/)
c
c
```

```
      write(8,*) ' % FRACTION PRODUCED BY U'
c
c
      write(8,773) (PrU(i),i=1,7)
c
773   format(7f10.2)
c
c
      write(8,*) ' Equivalent Z for Build up factors'
c
c
      write(8,773) (Zeq(i),i=1,7)
c
c
      write(8,*) ' Gamma Self shielding by Core, Reflector and PV'
c
c
      write(8,774) (shg(i),i=1,7)
c
774   format(7f10.6)
c
c
      write(8,*) ' The same but without Buildup factor'
c
c
      write(8,774) (shgnb(i),i=1,7)
c
c
      write(8,*)'Integrated Dose(Rad) without any shielding(n or gamma)'
c
c
      write(8,772) (Dosewg(i),i=1,7)
c
c
      if(ishig.eq.0.and.ishin.eq.1)
     *write(8,*)'Integrated Dose  Neutrons Shieldings(and no W)'
c
c
      if(ishig.eq.0.and.ishin.eq.1) write(8,772) (DosenG(i),i=1,7)
c
      if(ishig.eq.1.and.igam2.ne.0)
     *write(8,*)'Integrated Dose with W and Neutrons Shieldings'
c
c
      if(ishig.eq.1.and.igam2.ne.0) write(8,772) (DoseG(i),i=1,7)
c
      write(8,775) Dosew
c
775   Format(' Integrated Dose without Any Shielding,Total(Rad)',e12.4)
c
c
      if(ishig.eq.1.and.igam2.ne.0) Dosos=Doses
      if(ishig.eq.0) Dosos=Dosen
      write(8,776) Dosos,tgam,tgam0
c
776   format(' Integrated dose with Shieldings(W and Neutrons), Total (R
     *ads)',e12.2/
     *' Tungsten Thickness(cm)',f10.2/
     *' Tungsten Thickness(cm)',f10.2,' if no Neutron Shield Present'//)
c
c
      write(8,804) Dose,Zload
c
804   format(/' Requested Dose(Rad)',f10.2,' at ',f6.2,' meters'/)
c
c  SOLVE THERMALHYDRAULIC
c
```

```
c
c
      podt=pod
      Hact=H
      write(8,2367) toutl
c
2367  format(//' THERMALHYDRAULICS CALCULATIONS'/
     *         ' Average Temperature of the Coolant at Core Exit(K)',f10
     *.2//)
c
c
      if(iboun.eq.1) write(8,3478)
3478  format(/' BOUNDARY CONDITION: INPUT FLOW'//)
c
c
      if(iboun.eq.2) write(8,3479)
3479  format(/' BOUNDARY CONDITION: OUTLET TEMPERATURE'//)
c
c check if this is a calculation assuming uniform power distribution
c
      if(iopth.gt.2) go to 674
c here if 3D or 2D power density is constant
      Pfacr=1.
      Pfacaz=1.
      Call Thermoli(0)
c program end
      go to 677
674   continue
c here if we have uniform loading ie spacial dependent power density
      if(iouth.eq.2) go to 671
c
c Thermo at Max power density
c
      Pfacaz=1.
c
c
      write(8,687)
c
687   format(' ****PROFILE OF CHANNEL WITH MAXIMUM POWER DENSITY****'//)
c
c  calculation of Pfacr(radial power factor)MAX
      nenrp=3
      npodp=3
      do 2577 j=1,npodp
      do 2578 i=1,nenrp
2578  aux(i)=pfr(i,j)
      call Inter(enrip,aux,nenrp,enr,rmm)
2577  bux(j)=rmm
      call Inter(podip,bux,npodp,pod,Pfacr)
      Call Thermoli(0)
671   continue
      if(iouth.eq.1) go to 677
c
c Thermo at Min power density
c
      Pfacaz=1.-Ripple
c
c
      write(8,697)
c
697   format(///'***PROFILE OF CHANNEL WITH MINIMUM POWER DENSITY***'//)
c
c calculation of min power radial profile
      do 3577 j=1,npodp
      do 3578 i=1,nenrp
3578  aux(i)=pfrm(i,j)
      call Inter(enrip,aux,nenrp,enr,rmm)
```

```
3577  bux(j)=rmm
      call Inter(podip,bux,npodp,pod,Pfacr)
      Call Thermoli(0)
677   continue
      Stop
      End
c
c
c
c
c
c
c
c He density(kg/m3) function of p(Pa) and t(K)
c from page 405, ANS book "Thermal and Flow Design of Helium-Cooled
c Reactors",G. Melese and R. Katz
c
      Function rho(p,t)
      R=2077.22
      rho=p/(r*t+p*b(t))
      return
      end

c
c Function b(t)(used by rho)
c
      Function b(t)
      data c1,c2,c3,c4,c5/9.489433e-4,9.528079e-4,3.420680e-2,
     *                    2.739470e-3,9.409120e-4/
      b=c1+c2/(1.-c3*t)+c4/(1.+c5*t)
      return
      end

c
c  Compute available energy or
c  difference between available and requested energies
c  as function of S/F(other thing too)
c
      Function Balance(ibu,enr,pod)
c
c For input enr(ichment),pod(p/d) and when itest=0:
c this function calculates 1) the volume of the core that satisfies the
c reactivity balance; and then 2)the energy balance(Mwday)
c Balance=Eavailable-Erequested. If ishape=2 or 3 (input R or H) the
c function checks if R(or) H is compatible with reactivity balance
c if not, set flag ibu=1 and RETURN
c
c if itest=1 and for inputs enr and pod it calculates
c available energy(for ishape=1,2 or 3),
c radius infinite core that satisfies reactivity balance if ishape=2
c height of infinite "   "       "          "        "     "    "  =3
c
      common /Bala/ Ereq,relr,relal,rela2,dkeol,dkstr,BU,Vc,
     *U5m,dkbu,dkbe,rkrem,dcr,sigma,pin,dpv,dkpv,tduc,
     *eps,itest
c
      common /MaBaSi/ R,H,ishape
c
      dimension rm(5,5),enri(5),podi(5),bew(5,5)
     *,rki(5,5),prv(5,5),aux(5),bux(5),dext(5)
c
      data (rm(5,j),j=1,5)/7.695,8.762,9.841,10.927,12.010/
      data (rm(4,j),j=1,5)/7.945,9.032,10.131,11.230,12.325/
      data (rm(3,j),j=1,5)/8.149,9.248,10.357,11.470,12.571/
      data (rm(2,j),j=1,5)/8.356,9.467,10.586,11.706,12.811/
      data (rm(1,j),j=1,5)/8.563,9.681,10.805,11.929,13.037/
c
```

```
c rm(i,j) (cm) migration length of cell with enrichment i and p/d j
c for the following enrichments and p/d
c
      data(enri(i),i=1,5)/0.5,0.6,0.7,0.8,0.93/
      data(podi(i),i=1,5)/1.1,1.2,1.3,1.4,1.5/
c
      data(rki(5,j),j=1,5)/1.6953,1.6574,1.6194,1.5814,1.5432/
      data(rki(4,j),j=1,5)/1.6127,1.5728,1.5333,1.4938,1.4544/
      data(rki(3,j),j=1,5)/1.5366,1.4953,1.4545,1.4141,1.3740/
      data(rki(2,j),j=1,5)/1.4460,1.4035,1.3618,1.3207,1.2802/
      data(rki(1,j),j=1,5)/1.3364,1.2931,1.2510,1.2098,1.1695/
c
c rki k-infinite as function of i(enrichment) and j(p/d)
c
c
      data (bew(5,j),j=1,5)/19.26,18.57,17.65,16.61,15.50/
      data (bew(4,j),j=1,5)/17.26,16.30,15.18,13.99,12.76/
      data (bew(3,j),j=1,5)/15.28,14.11,12.85,11.55,10.25/
      data (bew(2,j),j=1,5)/12.78,11.40,10.02,8.65,7.32/
      data (bew(1,j),j=1,5)/9.57,8.04,6.58,5.22,3.95/
c
c bew(i,j): % delta k; reactivity worth of 25 cm OBe radial reflector as
c function of i(enr) and j(p/d)
c
      data (prv(5,j),j=1,5)/5.69,5.39,5.12,4.77,4.41/
      data (prv(4,j),j=1,5)/5.43,5.06,4.66,4.23,3.82/
      data (prv(3,j),j=1,5)/5.10,4.63,4.14,3.67,3.22/
      data (prv(2,j),j=1,5)/4.57,3.99,3.44,2.92,2.44/
      data (prv(1,j),j=1,5)/3.73,3.05,2.45,1.91,1.43/
      data dref/1./
c
c prv(i,j): % delta k; reactivity worth Astar as pressure Vessel
c function of i(enr) and j(p/d) at reference thickness dref
c
      data (dext(j),j=1,5) /3.165,3.565,3.954,4.360,4.737/
c
c dext(j)(cm) extrapolation length for bare core(mainly function of p/d)
c
      data pee,xop,pisq3,bo/3.1415926,0.541315,5.441398,2.945511/
c
c xop:optimum shape for cylindrical reactors=Radius/Height
c pisq3=pi*sqrt(3)
c
      data roun,drodc,fpin,wfi/12.172,0.64,0.56861025,0.9501647/
c roun: density UN, fpin: fraction volume of pin inside rod,dpin=4.826mm
c drod=6.4mm, wfi=Mwd produced by the total fission of 1 g of U235
c at 200 Mev/fission
      data AU5,AU8,AN,Av/235.044,238.0508,14.00307,0.602202/
c
c
c
c
      ibu=0
      nenr=5
      npod=5
c
c compute dexto(cm) extrapolated length
c
      call Inter(podi,dext,npod,pod,dexto)
c
c compute kinf
c
      do 1577 j=1,npod
      do 1578 i=1,nenr
1578  aux(i)=rki(i,j)
      call Inter(enri,aux,nenr,enr,rkk)
1577  bux(j)=rkk
```

```
      call Inter(podi,bux,npod,pod,rkin)
c
c compute migration length
c
      do 2577 j=1,npod
      do 2578 i=1,nenr
2578  aux(i)=rm(i,j)
      call Inter(enri,aux,nenr,enr,rmm)
2577  bux(j)=rmm
      call Inter(podi,bux,npod,pod,rmi)
c
c compute dk OBe reflector
c
      do 3577 j=1,npod
      do 3578 i=1,nenr
3578  aux(i)=bew(i,j)
      call Inter(enri,aux,nenr,enr,bmm)
3577  bux(j)=bmm
      call Inter(podi,bux,npod,pod,wbe)
      wber=wbe*relr
      wbea1=wbe*rela1
      wbea2=wbe*rela2
      dkbe=wber+wbea1+wbea2
c
cb compute dk Astar pressure vessel ( 1. cm thick, reference value)
c
      do 4577 j=1,npod
      do 4578 i=1,nenr
4578  aux(i)=prv(i,j)
      call Inter(enri,aux,nenr,enr,bmm)
4577  bux(j)=bmm
      call Inter(podi,bux,npod,pod,wpv)
      Rref=bo*rmi/sqrt(rkin-1.)-dexto
      drefp=Rref*(pin/sigma)
      apv=relr*0.01*wpv*(0.4/dref)*Rref
      dkpv=(drefp/dref)*relr*wpv
c
c compute delta k BU
c see: c50 and c80 are number densities inside the pin
c
      AU=enr*AU5+(1.-enr)*AU8
      AUN=AU+AN
      cu=roun*Av/AUN
      c50=cu*enr
      c80=cu*(1.-enr)
      bup=BU*0.01
      Call bupo(bup,rkin,c50,c80,dkbu)
c
c compure U235 density diluted in the core,rou5
c
      rkrm=1.+(DKBU+dkstr+dkeol-dkbe)/100.
      rkrem=rkrm+dkpv/100.
      go to (2001,2002,2003),ishape
c
c if ishape=1, optimum shape----->2001
2001  Rp=rmi*bo/sqrt(rkin/rkrem-1.)
      Hp=Rp/xop
      R=rp-dexto
      dpv=(pin/sigma)*R
      H=hp-dexto*2.
      Vc=pee*R**2*H
      if(dpv.gt.0.4) go to 2004
      dpv=0.4
      aa=(rmi*bo)**2+rkin*apv*dexto/rkrm**2
      bb=0.5*apv*rkin/(rkrm**2*aa)
      cc=(rkin/rkrm-1.)/aa
      riv=-bb+sqrt(bb**2+cc)
```

```
      rp=1./riv
      hp=rp/xop
      R=rp-dexto
      H=hp-dexto*2.
      Vc=pee*R**2*H
      dkpv=100.*apv/R
      rkrem=rkrm+dkpv/100.
c     write(8,1699) enr,pod,r,h,dpv,rkin,rmi,apv,rkrm,dexto,bo
1699  format('enr ',f6.3,' pod ',f6.3,' r ',f6.2,' h ',f6.2/
     *'dpv ',f6.2,' rin ',f7.5,' m ',f6.2,/
     *'apv ',e10.4,' rkrm ',f7.5,' dex ',f6.2,' bo ',f7.4//)
      go to 2004
c
c go next if ishape=2----->input Radius of core
c
2002  continue
      buck=sqrt(rkin/rkrem-1.)/rmi
      Rcr=2.405/buck-dexto
      dprs=Rcr*(pin/sigma)
      if(dprs.gt.0.4) dcr=Rcr
      if(dprs.gt.0.4) go to 1703
      aa=(rmi*2.405)**2+rkin*apv*dexto/rkrm**2
      bb=0.5*apv*rkin/(rkrm**2*aa)
      cc=(rkin/rkrm-1.)/aa
      riv=-bb+sqrt(bb**2+cc)
      rp=1./riv
      dcr=rp-dexto
1703  continue
      if(R.le.dcr) ibu=1
c
c set flag ibu=1 if R(input) is too small to satisfy reactivity balance
c
      if(ibu.eq.1) return
      dpv=(pin/sigma)*R
      if(dpv.ge.0.4) go to 1823
      dpv=0.4
      dkpv=100.*apv/R
1823  rkrem=rkrm+dkpv/100.
      buca=(rkin/rkrem-1.)/rmi**2-(2.405/(R+dexto))**2
      hp=pee/sqrt(buca)
      H=hp-dexto*2.
      Vc=pee*R**2*H
      go to 2004
c
c go next if ishape=3------->input Height of the core
c
2003  dcr=pee*rmi/sqrt(rkin/rkrem-1.)-dexto*2.
      if(H.le.dcr) ibu=1
c
c set flag ibu=1 if H(input) is too small to satisfy reactivity balance
c
      if(ibu.eq.1) return
      buca=rkin/rkrem-1.-(pee*rmi/(H+dexto*2.))**2
      rp=rmi*2.405/sqrt(buca)
      R=rp-dexto
      dpv=(pin/sigma)*R
      Vc=pee*R**2*H
      if(dpv.gt.0.4) go to 2004
      dpv=0.4
      aa=(rmi*2.405)**2+rkin*apv*dexto/rkrm**2
      bb=0.5*apv*rkin/(rkrm**2*aa)
      cc=(rkin/rkrm-1.-(rmi*pee/(H+dexto*2.))**2)/aa
      riv=-bb+sqrt(bb**2+cc)
      rp=1./riv
      R=rp-dexto
      Vc=pee*R**2*H
      dkpv=100.*apv/R
```

```
      rkrem=rkrm+dkpv/100.
2004  continue
c
c Compute average density of U235
c Abu (cm2) area of 19 rods bundle with the duct
      Abu=3.4641016*drodc**2*(pod*2.73205081-0.5+0.1*tduc/drodc)**2
c rods fraction
      Arod=19.*pee*(drodc/2.)**2
      frod=Arod/Abu
      rouno=roun*frod*fpin
      rou=rouno*(AU/AUN)
      rou5=enr*rou*AU5/AU
c
c check available energy
c
      U5m=Vc*rou5
c
c Eavai(Mwd) is the total energy produced by the fission of U235,U238and
c Pu239 at 200 Mev per fission. See,here BU=Total fission/Initial number
c of U235 atoms; ie U5 at EOL is NOT BU*U5(BOL)
c
      Eavai=U5m*BU*wfi/100.
      Balance=Eavai-Ereq
      if(itest.eq.1) Balance=Eavai
      return
      end

c
c
c compute control drums
c
      subroutine Control
      dimension cr(5,5),enri(5),podi(5),aux(5),bux(5),bew(5,5),prv(5,5)
      common /cont/ R,dpv,drr,enr,pod,refec,ncr,dksu,dkstr,relr,dcd
c
c R:radius core, drr:thickness reflector; dref: reference reflector
c thickness(with which worths were calculated);rasp:S/F element ratio
c refec:keff to control; ncr:number of control rods;dksu(%):subcritica-
c lity required for drums in.
c dkstr(%): dk structural part of reactor,relr:relative worth of reflecto
c
      data dref,dpvref,dcdref/25.,1.,2./
c
c dref,dpvref,dcdref:reference values for calculation worth of reflector
c PV and B4C in Control drums(cd)
      data (cr(5,j),j=1,5)/6.73,6.64,6.37,6.07,5.73/
      data (cr(4,j),j=1,5)/5.98,5.74,5.43,5.08,4.69/
      data (cr(3,j),j=1,5)/5.23,4.92,4.55,4.15,3.73/
      data (cr(2,j),j=1,5)/4.29,3.91,3.50,3.06,2.63/
      data (cr(1,j),j=1,5)/3.12,2.69,2.24,1.81,1.39/
c Cr(i,j)
c  %delta k; max worth of control rods in OBe reflector
c for enrichment i and p/d j and
c calculated with reflector thickness dref(reference)
c
      data (bew(5,j),j=1,5)/19.26,18.57,17.65,16.61,15.50/
      data (bew(4,j),j=1,5)/17.26,16.30,15.18,13.99,12.76/
      data (bew(3,j),j=1,5)/15.28,14.11,12.85,11.55,10.25/
      data (bew(2,j),j=1,5)/12.78,11.40,10.02,8.65,7.32/
      data (bew(1,j),j=1,5)/9.57,8.04,6.58,5.22,3.95/
c
c bew(i,j): % delta k; reactivity worth of 25 cm OBe radial reflector as
c function of i(enr) and j(p/d)
c
      data (prv(5,j),j=1,5)/5.69,5.39,5.12,4.77,4.41/
      data (prv(4,j),j=1,5)/5.43,5.06,4.66,4.23,3.82/
      data (prv(3,j),j=1,5)/5.10,4.63,4.14,3.67,3.22/
```

```
      data (prv(2,j),j=1,5)/4.57,3.99,3.44,2.92,2.44/
      data (prv(1,j),j=1,5)/3.73,3.05,2.45,1.91,1.43/
c
c prv(i,j): % delta k; reactivity worth Astar as pressure Vessel
c function of i(enr) and j(p/d) at reference thickness dref
c
      data (enri(i),i=1,5)/0.5,0.6,0.7,0.8,0.93/
      data (podi(j),j=1,5)/1.1,1.2,1.3,1.4,1.5/
      if(ncr.eq.0) write(8,48)
48    format(' Your input for the number of control drums is zero'/
     *' Subroutine Control bypassed'//)
      if(ncr.eq.0) return
c
      pee=3.1415926
c dcd(cm): thickness of B4C in Control drums
c
c worth of drums for input enr,pod
c
      npod=5
      nenr=5
      do 2577 j=1,npod
      do 2578 i=1,nenr
2578  aux(i)=cr(i,j)
      call Inter(enri,aux,nenr,enr,crr)
2577  bux(j)=crr
      call Inter(podi,bux,npod,pod,cdr)
c
c worth of 2*pee sheet(cdr) is corrected because drr might not be =dref
c correction factor relative worth of reflector(with respect to dref)
c cdr,next, is for conditions:  PV=1cm thick,B4C=2cm thick
      cdr=cdr*relr
c
c Correction for PV.ne.1cm or B4C.ne.2cm:
c
c worth=AM*exp(-sigpv*dpv)*(1.-exp(-sigcd*dcd)
c where: sigpv and sigcd are removal cross section in PV and Drum
c        dpv,dcd are thicknesses of PV and B4C
c
c 1) Compute sigpv looking to worth of PV at "infinite" and dpvref=1.
c
c a)worth of reflector=worth of infinite thick PV=wpvi
c
      do 3577 j=1,npod
      do 3578 i=1,nenr
3578  aux(i)=bew(i,j)
      call Inter(enri,aux,nenr,enr,crr)
3577  bux(j)=crr
      call Inter(podi,bux,npod,pod,wpvi)
      wpvi=wpvi*relr
c
c b)worth of dpvref cm thick PV=wpv1
c
      do 4577 j=1,npod
      do 4578 i=1,nenr
4578  aux(i)=prv(i,j)
      call Inter(enri,aux,nenr,enr,crr)
4577  bux(j)=crr
      call Inter(podi,bux,npod,pod,wpv1)
      wpv1=wpv1*relr
c
c compute exponent of flux attenuation in PV
c
      sigpv=-alog(1.-wpv1/wpvi)/dpvref
c
c 2) Compute sigcd and AM looking at worth of B4C at "infinite" and 2.cm
c worth of "infinite" thick B4C=wpvi(Reflector)-wpv1(PV)
      wcdi=wpvi-wpv1
```

```
      AM=wcdi*exp(sigpv*dpvref)
      sigcd=-alog(1.-cdr/(AM*exp(-sigpv*dpvref)))/dcdref
c
c worth of 2 pi's sheet of B4C(thick=dcd) beside the PV(thick=dpv)
c
      cdr=AM*exp(-sigpv*dpv)*(1.-exp(-sigcd*dcd))
c
c
c rea(%) reactivity to control
c
c
      rea=(refec-1.)*100.+dksu-dkstr
c
c compute maximum number of drums=nmax
c
      Rpv=R+dpv
      rd=0.5*drr
      par=2.*asin(rd/(Rpv+rd))
      pard=par*180./pee
      write(8,23) rd,pard,dcd
23    format(//' From Control'/
     *' Radius Drums(cm)',f8.2,' Central Parallax(degree)',f8.2/
     *' Thickness B4C sheet on Control Drums,cm',f8.2/)
c
c rd and par are the radius and the central parallax of the drums
c
      nmax=2.*pee/par
      cdrm=cdr*nmax*par/(2.*pee)
      if(cdrm.ge.rea) go to 10
c pass to local parallax(seen from center of drum)---> dtcr,dtcd
      bb=(Rpv+rd)*cos(0.5*par)
      xx=bb
      dtcr=2.*asin(xx*sin(0.5*par)/rd)
      dtcd=180.*dtcr/pee
      write(8,20) nmax,cdrm,rea,dtcd
20    format(//' You cannot control this reactor only with control'/
     *         ' drums in the reflector'/
     *' The reactivity worth of',i5,' drums(max value) is',f6.2,' %'/
     *'                    and the reactivity to control is',f6.2,' %'/
     *'Absorbing angle of each drum would be',f8.2,' degrees'//)
      return
10    continue
      if(ncr.gt.nmax) write(8,33) ncr,nmax
33    format(//' Your input for the number of drums is',i5/
     *' larger than the max possible',i5,' continue with max'//)
      if(ncr.gt.nmax) ncr=nmax
      rcr=ncr
      cdrm=cdr*rcr*par/(2.*pee)
      if(cdrm.lt.rea) write(8,34) ncr,nmax
34    format(//' Your input for the number of drums',i5,' is too small'/
     *'(max possible is',i5,')program is going to increase it')
      if(cdrm.lt.rea) then
101   ncr=ncr+1
      cdrm=cdr*ncr*par/(2.*pee)
      if(cdrm.lt.rea) go to 101
      rcr=ncr
      else
      endif
c
c para is the central parallax of the absorbing part of one drum
c
      para=(rea/rcr)*(2.*pee/cdr)
c pass to local parallax(seen from center of drum)---> dtcr,dtcd
      bb=(Rpv+rd)*cos(0.5*para)
      cc=Rpv**2+2.*Rpv*rd
      xx=bb-sqrt(bb**2-cc)
      dtcr=2.*asin(xx*sin(0.5*para)/rd)
```

```
      dtcd=180.*dtcr/pee
      write(8,31) ncr,dtcd,rea
31    format(//i5,' Drums with absorbing angle(d)',f8.2,' each'/
     *           ' Control dk(%)=',f8.2//)
      return
      end


c
c solve thermalhydraulic
c
      subroutine thermoli(iop)
      real*4 k
      dimension pow(51),pres(51),tbulk(51),ve(51),z(51),rj(51)
     *,tw(51),Tclad(51),Tfuel(51),Tclaa(51),Tfua(51),Tbuma(51)
c
      common/thermo/ aln,nnodo,ifri,npin,pin,tin,tout,deltapa
     *,Powt,rjs,flowt,pod,tduc,Rc
     *,Hact,Pfacr,Daxi,fact,iprth,iopth,ichoose
     *,Pfacaz,iboun,nrod,nbu,nrodbu,hodo
c
      common /fric/ podf,hod,xx,drel,coefri
c
      character*80 aln
c
c
      Tsa(ps)=8143./(8.-alog10(ps/0.133322))
      Psa(ts)=0.133322*10.**(8.00-8143./ts)
c
c Tsa(ps) is saturation temperature of Li(in K) as function of ps(KPa)
c Psa(ts) is     "      pressure    of Li(in KPa)"     "    of ts(K)
c
      data pee/3.1415926/
c
      data drod,dclad,dliner,dHe,TLim /0.64,0.0635,0.0127,0.0025,453.7/
      data rpin/0.2413/
      data TmTa,TmUN/3269.,2903./
c
c cm diameter rod,clad,liner,He thickness
c
c
c
c nnode: number of nodes;
c iboun /1,2/ boundaries conditions : read flow or T outlet
c ifri.ne.1 : switch off friction factor
c nclad: number nodes inside clad
c npin: number of nodes inside fuel
c icheck/0,1/ no,print some parameters to check them
c
c pin(MPa), tin( K) inlet conditions
c pod=p/d,
c fact: factor to multiply the power
c total power; rjs(Kw/cm2) scale for plot heat rate
c Vact(L) active volume of the core, ie Volume occupied by UN
c tduc(mm) duct thickness; hod lead of the spacing wire(in unit of drod)
c
c
c
c flowt(Kg/sec) :  Total flow
c flow1,2,3(g/sec)   :  Channel flow for cell types 1,2,3
c
c
c pin(MPa), tin( K) inlet conditions
c total power; rjs(Kw/cm2) scale for plot heat rate
c
c
      npu=nnodo+1
```

```
      hod=hodo
      podf=pod
c
c  Hact(cm):active length; Pfact:Radial Peaking factor at r position
c  assume cos in z and  Daxi(cm) axial extrapol.
c
      Abu=3.46410*drod**2*(pod*2.732051-0.5+.1*tduc/drod)**2
      Abund=3.46410*drod**2*(pod*2.732051-0.5)**2
c flow per bundle flow(g/s)
      flow=1000.*flowt/Nbu
      d=drod*0.01
c f1,2,3 are flow fractions for cell types 1,2,3(with respect to acell1)
      x=pod
      f1=1.-0.906900*(1.+(x-1.)**2/3.)/x**2
      f2=2.30940*((x-.5)*x-.392699*(1.+.5*(x-1.)**2))/x**2
      f3=2.30940*(.577350*(x-.5)**2-0.130900)/x**2
c  area of cell(m2) is acell
      pitch=pod*d
      acell=0.433013*pitch**2
c a1,a2,a3(m2) are area flow of cell types 1,2,3
      a1=f1*acell
      a2=f2*acell
      a3=f3*acell
c wet perimeters (m) are pf1,2,3
      pf1=pee*d*(2.+x)/6.
      pf2=.25*d*(pee+x*(pee+4.))
      pf3=d*(pee+6.92820*(x-.5))/6.
c d1,2,3 are the equivalent diameters(m) for cell types 1,2,3
      d1=4.*a1/pf1
      d2=4.*a2/pf2
      d3=4.*a3/pf3
c total area flow,At and equivalent diameter for 19 rods bundle
      At=24.*a1+12.*a2+6.*a3
      pft=24.*pf1+12.*pf2+6.*pf3
      dt=4.*At/pft
      drel=D1/Dt
c XX factor for bundle pressure correlation
      XX=At/(24.*A1+12.*A2*(D2/D1)**0.714+6.*A3*(D3/D1)**0.714)
c geometry of fuel rod:
      Rr=0.01*drod/2.
      Rcl=Rr-(dclad+dliner)*0.01
      Rp=Rcl-0.01*dHe
c
c d,Rr,Rc,Rp(meter) are diameter and radius of rod,internal radius of
c clad+liner and radius of pin
c
c     distribute flow in cell types according to area
      flow1=flow*24.*A1/At
      flow2=flow*12.*A2/At
      flow3=flow*6.*A3/At
c estimation of delta p across bundle
c
      Gt=0.001*flow/At
c viscosity at average conditions
      p=pin*1.e+6
      call pli(h,p,tin,rhoi,visi,cp,pr,k,x,0)
      vein=Gt/rhoi
      call pli(h,p,tout,rhoo,viso,cp,pr,k,x,0)
      veout=Gt/rhoo
      t=0.5*(tin+tout)
      call pli(h,p,t,rhoa,visa,cp,pr,k,x,0)
      Rea=Dt*Gt/visa
      frica=fri(Rea)
      vea=Gt/rhoa
      facfra=rhoa*Vea**2*frica*Pft/2.
      deltapa=Gt*(veout-vein)+facfra*Hact/At
      if(iop.eq.1) return
```

```
c
c if iop=1 return after calculating 1 node estimation of delta P
c
      dzet=Hact/nnodo
      Vact=nrod*pee*rpin**2*Hact/1000.
      roav=Powt*fact/Vact
      write(8,12) aln
12    format(a80)
      write(8,21) TLim,tin,drod,Powt,nrod,Nbu,nrodbu,flowt,Vact,fact
21    format(//' Thermodynamic profile of a liquid Li cooled channel'/
     *'*********************************************************'/
     *'*          Remember Li melts at',f8.2,' K !!!!          *'/
     *'*********************************************************'/
     *' Inlet temperature( K)=',f10.2/
     *' Diameter Rod(cm)=',f7.2/
     *' Total Power(Mw)',f10.3,' No. Rods',i10,' No.Bundles',i5,'(',i3,'
     *rods/bundle)'/
     *              ' Total Flow(Kg/sec)',f10.3/
     *'Active Volume(L of UN)',f10.3/
     *' Factor that multiplies Power',f10.7//)
      write(8,162)roav,Hact
162   format('Average Power Density(Mw/L(UN))',f8.4,' Channel Length(cm)
     *',f7.2/)
      A1c=A1*10000.
      A2c=A2*10000.
      A3c=A3*10000.
      Atc=At*10000.
      A1cr=A1c/0.5
      A2cr=A2c/0.5
      A3cr=A3c*6.
      Atcr=Atc/19.
      pf1c=pf1*100.
      pf2c=pf2*100.
      pf3c=pf3*100.
      pftc=pft*100.
      d1c=d1*100.
      d2c=d2*100.
      d3c=d3*100.
      dtc=dt*100.
      ft=Atc/Abund
      n1=24
      n2=12
      n3=6
      write(8,190) pod,hod
190   format(//'FLOW CONDITIONS:'//
     *' Pitch/drod',f8.3,' Lead wire spacer/drod',f8.3//
     *'Cell Type (per bundle) ',11x,'1',11x,'2',11x,'3',6x,'Bundle'//)
      write(8,191) n1,n2,n3,flow1,flow2,flow3,flow,
     *             f1,f2,f3,ft,
     *             A1c,A2c,A3c,Atc,
     *             A1cr,A2cr,A3cr,Atcr,
     *             pf1c,pf2c,pf3c,pftc,
     *             d1c,d2c,d3c,dtc,
     *             XX,drel
191   format(
     *'Number of Cells        ',3i12/
     *'Flow(g/s) Total        ',4f12.4/
     *'Flow Area (*)          ',4f12.4/
     *'Flow Area(cm2),per cell',4f12.4/
     *'Flow Area(cm2),per rod ',4f12.4/
     *'Wet Perimeter(cm) Total',4f12.4/
     *'Effective Diameter(cm) ',4f12.4//
     *' (*) 1,2,3 in units of cell 1 area '//
     *' Parameters of the bundle for friction correlations:'
     *' X=',f6.3,' Del/DT=',f6.3//)
      if(iboun.eq.1) write(8,237) tout
      if(iboun.eq.2) write(8,238) tout
```

```
237   format(' Boundary Condition: Fix coolant flow; Temp. outlet',f8.2)
238   format(' Boundary Condition: Fix outlet flow temperature',f8.2)
      flow=0.001*flow
      Gt=flow/At
      flow1=0.001*flow1/24.
      flow2=0.001*flow2/12.
      flow3=0.001*flow3/6.
c
c
c Pfraz : Power factor radial-azimuthal
      Pfraz=1.
      if(iopth.gt.2) Pfraz=Pfacr*Pfacaz
c
c recompute flow  according to option
c
      if(iopth.eq.4) flow=flow*Pfraz
c
c
      Hext=Hact+2.*Daxi
      we=Hext/Hact
      rih=pee/Hext
      Pfact=Pfraz*pee/(2.*we*sin(0.5*pee/we))
      do 182 i=1,npu
      zzzz=Daxi+(i-1)*dzet
      if(iopth.gt.1)  pow(i)=roav*Pfact*sin(rih*zzzz)
      if(iopth.eq.1)  pow(i)=roav
182   z(i)=zzzz-Daxi
c
c
c z(i) (cm) boundaries of the nodes
c
c
      if(iopth.eq.1) write(8,1264)
1264  format(' 3-D UNIFORM POWER DISTRIBUTION ASSUMED'/)
c
       if(iopth.eq.2) write(8,1266)
1266  format(' 2-D (R-THETA) UNIFORM POWER DISTRIBUTION ASSUMED'/)
c
      if(iopth.ge.2)  write(8,164) Pfact,Daxi
164   format(' Cosine  Axial Power Distribution=roav*Pfact*Cos(Pi*z/H)'/
     *' Pfact(Axial*Radial*Azim)=',f7.4,' Extrapolated Delta(cm)',f7.2)
c
c
c
      if(iopth.eq.4) write(8,1293)
1293  format(' Coolant Flow Proportional to Power Density'/)
c
      if(iopth.ne.4) write(8,1295)
1295  format(' Uniform Coolant Flow Assumed'/)
c
c
c
c
c End calculation coolant conditions
c
c
c transform pow(i)(Mw/L) to power per unit length (watt/cm)
c  PER ROD
      facto=1.e+7*pee*Rp**2
      do 123 i=1,npu
123   pow(i)=pow(i)*facto
c
c z(i) (cm) boundaries of the nodes
ci
c
c
      if(ifri.ne.1) write(8,23)
```

```
23    format('  friction was assumed zero'//)
c
c
c
c
      write(8,253) pin
253   format(//' AVERAGE CONDITIONS IN THE BUNDLE'/
     *'*************************************************************'/
     *'***********   Inlet Press(MPa)',f9.6,'  ***********************'/
     *'*************************************************************'
     *///)
       write(8,31)
31    format(
     *' Node   z(cm)Po(w/cm)   Pr(KPa)    T(K)  (Tsat)  V(m/s)      Re')
c
c for bundle
      pres(1)=pin*10.**6
      tbulk(1)=tin
      Tbuma(1)=tin
      p=pres(1)
      t=tbulk(1)
      call pli(h,p,t,rho,vis,cp,pr,k,x,0)
c Inlet conditions for cell types 1,2,3
      hi1=h
      hi2=h
      hi3=h
c Average conditions for bundle
      rhoi=rho
      Re=Dt*Gt/vis
      frici=fri(Re)
      ve(1)=Gt/rho
c remember for future use inlet conditions
      hin=h
      echannel=0.
      i=1
      pre=pres(1)*10.**(-3)
      Ts=Tsa(pre)
      write(8,41) i,z(i),pow(1),pre,tbulk(i),Ts,ve(i),Re
      facfr=0.
c
c
c start analysis bulk conditions
c
c
c change units for the flow in each type of cell
      dt1m=0.
      dt2m=0.
      dt3m=0.
      isat=0
      coeRe=coefri
c
      do 30 i=2,npu
      dz=z(i)-z(i-1)
c po is # of watts per rod in dz
      po= 0.5*(pow(i)+pow(i-1))*dz
      echannel=echannel+po*19.
c toutlet of node for cell types 1,2,3 T1,T2,T3
c cell 1 1/2 rod
      h=hi1+po/(2.*flow1)
      p=pres(i-1)
      call pli(h,p,t,rho,vis,cp,pr,k,x,1)
      T1=t
      Tbuma(i)=T1
      hi1=h
      hout=24.*flow1*h
c cell 2 1/2 rod
      h=hi2+po/(2.*flow2)
```

```
      call pli(h,p,t,rho,vis,cp,pr,k,x,1)
      T2=t
      hi2=h
      hout=hout+12.*flow2*h
c cell 3 1/6 rod
      h=hi3+po/(6.*flow3)
      call pli(h,p,t,rho,vis,cp,pr,k,x,1)
      T3=t
      hi3=h
      hout=hout+6.*flow3*h
      hout=hout/flow
c compute average T in bundle
      call pli(hout,p,t,rho,vis,cp,pr,k,x,1)
      tbulk(i)=T
c remember max  T of cells
      dt1=T1-T
      if(abs(dt1).gt.dt1m) i1m=i
      if(abs(dt1).gt.dt1m) dt1m=dt1
      dt2=T2-T
      if(abs(dt2).gt.dt2m) i2m=i
      if(abs(dt2).gt.dt2m) dt2m=dt2
      dt3=T3-T
      if(abs(dt3).gt.dt3m) i3m=i
      if(abs(dt3).gt.dt3m) dt3m=dt3
c compute voutlet
      ve(i)=ve(i-1)*rhoi/rho
c compute pressure
      Re=Dt*Gt/vis
      if(ifri.eq.0) go to 267
      frico=fri(Re)
      coeRe=coeRe+coefri
      facfr=0.5*(rhoi*Ve(i-1)**2*frici+rho*ve(i)**2*frico)*Pft/2.
267   continue
      dz=0.01*dz
      pres(i)=pres(i-1)-Gt*(ve(i)-ve(i-1))-facfr*dz/At
      rhoi=rho
      frico=frici
      pre=pres(i)*1.e-03
      Ts=Tsa(pre)
      if(i.eq.npu.or.iprth.eq.1)
     *write(8,41) i,z(i),pow(i),pre,tbulk(i),Ts,ve(i),Re
      if(Ts.le.T1) isat=1
30    continue
      coeRe=coeRe/npu
c
c end t bulk calculations
c
41    format(i4,2f8.2,f10.2,f8.2,'(',f8.2,')',f8.2,f8.0)
      T1=T+dt1m
      T2=T+dt2m
      T3=T+dt3m
      p1=psa(t1)
      p2=psa(t2)
      p3=psa(t3)
      delp=p1-psa(T)
      write(8,360) i1m,dt1m,t1,p1,i2m,dt2m,t2,p2,i3m,dt3m,t3,p3,delp,coe
     *Re
360   format(' Thermodynamics Conditions of Cells'/
     *   'Cell Type   z Node  Max(T-<T>)           T Saturation P(kPa)'/
     *       ',        '1',i9,3f12.2/
     *       ',        '2',i9,3f12.2/
     *       ',        '3',i9,3f12.2//
     *' Delta pressure(KPa) because T1>T(average)',f10.2//
     *' <c> in f=c/Re**0.25 is',f10.5//)
c
c Check If reactor is underpressurized ,isat=1
c  , if yes write warning
```

```
c
      if(isat.eq.0) go to 350
      write(8,351) T1,Ts
351   format('Although ,in a BULK sense,the pressurization is correct'/
     *'  the temperature of cell type 1',f10.2/
     *' reaches or passes boiling temperature',f10.2/
     *' so some flushing is possible.'/
     *'You might:'/
     *'1)Increase p at inlet'/
     *'2)Reduce T outlet(if iboun=2), or increase flow(if iboun=1)'/
     *'3)Reduce T inlet'/
     *'4)Assume 2D uniform power distribution.'/
     *'5)Assume flow grading(iopth=4)'/
     *'6)Use larger bundle(not available yet)'//)
      if(ichoose.eq.1) stop
2515  write(8,2516)
2516  format(//
     *'*****************************************************************'/
     *'************** WARNING *** WARNING *** WARNING ***************'/
     *'*****************************************************************'//
     *'Despite previous caveats, Program continues with nominal values'/
     *)
350   continue
c
c calculation wall temperature and rod temperatures for MAX HEATED ROD
c IN bundle which is type 1 cell
c
c Initialization of function RNu
      Dummy=RNu(Re,1.,1,pod)
      dg=dHe*0.01
      G1=flow1/a1
c
c loop on z nodes
c
      have=0.
      TTa=0.
      TUN=0.
      do 302 i=1,npu
      p=pres(i)
      t=tbuma(i)
c
c SEE: we have chosen the highest T (cell 1) Tbuma(i)
c
      call pli(h,p,t,rho,vis,cp,pr,k,x,0)
      Re=G1*D1/vis
c define Peclet number
      Pe=Re*Pr
      RNus=RNu(Re,Pe,0,pod)
      hco=RNus*k/Dt
      have=have+hco
c rjq=watt/m2
      rjq=pow(i)*100./(pee*0.01*drod)
c rj(i)=Kw/cm2
      rj(i)=rjq*10.**(-7)
c Qh=watts/m3
      Qh=100.*pow(i)/(pee*Rp**2)
      Tw(i)=T+rjq/hco
c From Twa to Tfuel
c Clad+liner
      cclad=Rkclad(Twa)
c See only one node for the clad
c
      Tclad(i)=Tw(i)+rjq*Rr*alog(Rr/Rcl)/cclad
      Tclaa(i)=Tw(i)+rjq*Rr*(0.5-alog(Rr/Rcl)*Rcl**2/(Rr**2-Rcl**2))/
     *cclad
      if(Tclad(i).gt.TTa) TTa=Tclad(i)
c delta T across He gap
```

```
      Tg=Tclad(i)
      hg=Rkgap(Tg,dg)
      Tgap=Tclad(i)+rjq/hg
c temperatures inside pin , npin nodes
      dr=Rp/npin
      Tinl=Tgap
      rinl=Rp
      Tfa=0.
      do 3021 in=1,npin
      Cfuel=RkUN(Tinl)
      rout=rinl-dr
      Tout=Tinl+0.25*Qh*(rinl**2-rout**2)/Cfuel
      Tfa=Tfa+0.5*(Tout+Tinl)*(rinl**2-rout**2)
      Tinl=Tout
      rinl=rout
3021  continue
      Tfuel(i)=Tout
      if(Tout.gt.TUN) TUN=Tout
      Tfua(i)=Tfa/Rp**2
302   continue
      have=have/(10000.*npu)
c
c end loop on z nodes pin
      drd=10.*drod
      dcl=(dclad+dliner)*10.
      deHe=1000.*dg
      dpin=drd-(dcl+deHe)*2.
      write(8,978) drd,dcl,deHe,dpin
978   format(//' From the Coolant Channel to fuel pin temperature'/
     *'         Rod Diameter(mm):',f8.4/
     *' Clad+Liner Thickness( "):',f8.4/
     *' He gap          "    ( "):',f8.4/
     *' Fuel Pin Diameter   ( "):',f8.4//
     *' CONDITIONS FOR MAX HEATED ROD OF THE BUNDLE(Type 1 cell)')
      write(8,976)
976   format(///' Node     z(cm) J(Kw/cm2)  Twall(K)  Tclad(K)  TcladAv
     *  Tfuel(K)  TfuelAv '//)
      do 2583 i=1,npu
      if(i.eq.1.or.i.eq.npu.or.iprth.eq.1)
     *write(8,977)i,z(i),rj(i),tw(i),Tclad(i),Tclaa(i),Tfuel(i),Tfua(i)
2583  continue
977   format(i5,f10.2,f10.5,5f10.2)
c
c calculation of axial averages
c
      if(TTa.ge.TmTa.or.TUN.ge.TmUN) write(8,2633) TTa,TUN,TmTa,TmUN
2633  format(
     *'*************************************************************'/
     *'******WARNING******WARNING***********WARNING**************'/
     *'*************************************************************'//
     *'Max. Temp(K) clad is',f9.2,' Max. Temp(K) UN is',f9.2/
     *' Ta melts at(K)     ',f9.2,'  UN melts at(K)   ',f9.2/
     *' Relax inlet, outlet conditions or assume uniform power or'/
     *' flow grading'//)
      zav=0
      thav=0.
      phav=0.
      twa=0.
      tca=0.
      tfa=0.
      do 1436 i=2,npu
      dz=z(i)-z(i-1)
      zav=zav+dz
      phav=phav+0.5*(pres(i)+pres(i-1))*dz
      twa=twa+0.5*(tw(i)+tw(i-1))*dz
      tca=tca+0.5*(tclaa(i)+tclaa(i-1))*dz
      tfa=tfa+0.5*(Tfua(i)+Tfua(i-1))*dz
```

```
      thav=thav+0.5*(tbulk(i)+tbulk(i-1))*dz
1436  roin=roout
      thav=thav/zav
      phav=phav/zav
      twa=twa/zav
      tca=tca/zav
      tfa=tfa/zav
      pre=phav*0.000001
      Write(8,1486) pre,thav,twa,have,tca,tfa
1486  Format(//' Axial Average Conditions in the Channel'/
     *'    Coolant Pressure(MPa)',f8.3,' Temperature(K)',f9.3/
     *'         Wall Temperature(K)',f9.3,' Heat Transfer(w/cm2K)',f8.2/
     *'                Clad Temperature(K)',f9.3/
     *'                     Fuel Temperature(K)',f9.3)
      dh=hout-hin
      dq=echannel/flow
      dkin=0.5*(ve(npu)**2-ve(1)**2)
      rel=dkin/dq
      bal=(dh+dkin)/dq
      write(8,987) dq,dh,dkin,rel,bal
987   format(//' HEAT BALANCE FOR THE BUNDLE'//
     * 'dQ: heat to coolant (joule/kg)',e14.5/'dH: change of enthalpy
     1(joule/kg)',e14.5/'dKin change kinetic energy(joule/kg)',e14.5/
     2' relative dKin/dQ',f10.6/' Balance (dH+dKin)/dQ',f14.7)
      return
      end


c
c friction factor for bundle cooled by liquid metal
      function fri(Re)
      common /fric/ pod,hod,xx,drel,coefri
c where Re is the Reynold's number
c Novendstern correlation, page 131 ANS book about LMFBR
      if(Re.le.1000.) then
      fri=21./Re
      coefri=21./Re**0.75
      else
      coefri=(1.034/pod**.124+29.7*pod**6.94*Re**.086/hod**2.239)**.855
      coefri=0.079*coefri*xx**1.75/drel**1.25
      fri=coefri/Re**0.25
      endif
      return
      end


c
c
      subroutine bisect(x1,x2,xav,f1,f2,fav,iwa)
      if((f1*f2).le.0.) go to 11
      write(8,1)
1     format(//' f1*f2 > 0.'//)
      iwa=1
      return
11    iwa=0
      if((fav*f1).le.0.) go to 10
      f1=fav
      x1=xav
      return
10    f2=fav
      x2=xav
      return
      end


c
```

```
c Nussel Number for one Phase Flow of liquid metal along fuel elements
c
      function RNu(Re,Pe,ini,pod)
      dimension B(4),x(4),cc(6),xx(6)
      data (b(i),i=1,4)/7.25,10.65,12.,12.85/
      data (x(i),i=1,4)/1.1,1.2,1.3,1.4/
      data (cc(i),i=1,6)/.00293,.00253,.002078,.001607,.001194,.000862/
      data (xx(i),i=1,6)/1.01,1.2,1.4,1.7,2.2,3./
c Where Re and Pe and Reynolds and Peclet numbers evaluated at bulk
c conditions,ini=1 initialize according pod and return. ini=0 compute
c Nussel number after the initialization. Correlation from page 189 ANS
c book about LMFBR's
      if(ini.ne.1) go to 10
      call Inter(xx,cc,6,pod,C)
      if(pod.le.1.35) then
      al=0.80
      call Inter(x,b,4,pod,C1)
      c2=0.025
      else
      al=0.86
      c1=6.66+3.126*pod+1.184*pod**2
      c2=0.0155
      endif
10    Pr=Pe/Re
      eon=C*Re**0.915
      psi=1.-1.82/(Pr*eon**1.4)
      if(psi.lt.0.) psi=0.
      RNu=C1+C2*(psi*Pe)**al
      return
      end


c
c Function: Thermal Conductivity of UN
c Source: Alkasys Manual
c
      Function RkUN(Te)
c
c T(K) RkUC (watt/mK)
c
      RkUN=24.
      Return
      End
c
c
c Function: Thermal Conductivity of He gap
c Source: Alkasys Manual
c
      Function Rkgap(T,d)
c
c T(K),d(m) Rkgap (watt/m2K)
c Rkgap=k(He)/d(He),k(He) from ANS book about gas cooled reactors
      Rkgap=0.002774*T**0.701/d
      Return
      End
c
c
c Function: Thermal Conductivity of Astar clad and W liner
c Source: Alkasys Manual
c
      Function Rkclad(Te)
c
c T(K) RkUC (watt/cmK)
c
      Rkclad=57.
      Return
      End
```

```
c Subroutine: Interpolate Table
c
      Subroutine Inter(x,y,np,xi,yi)
      Dimension       x(20),y(20)
      if(xi.lt.x(1)) go to 10
      if(xi.ge.x(np)) go to 20
      i=1
11    continue
      if(x(i).gt.xi) go to 12
      i=i+1
      go to 11
12    continue
      go to 30
10    i=2
      go to 30
20    i=np
30    yi=y(i-1)+((y(i)-y(i-1))/(x(i)-x(i-1)))*(xi-x(i-1))
      return
      end


c
c Properties of Li
c
      subroutine pli(h,p,t,rho,vis,cp,pr,rk,x,iflag)
c23456789012345678901234567890123456789012345678901234567890123456789012
c Thermodynamic and Transport Properties of Lithium
c from "Lithium Literature Review:Lithium's Properties and Interactions"
c HEDL-TME 78-15, April 1978.
c
c SI units, T(K)
      x=1.
c
c go to 10 if iflag=1 (ie input is p,h)
c
      if(iflag.ne.0) go to 10
c density rho
1     rho=0.515-0.000101*((T-273.15)-200.)
c from g/cm3 to Kg/m3
      rho=rho*1000.
c enthalpy h
      h=-5.075+1.0008*(T-273.15)-5173./(T-273.15)
c from cal/g to Joules/Kg
      h=h*4186.8
c specific heat cp
      cp=4186.8*(1.0008+5173./(T-273.15)**2)
c viscosity vis
      vis=726.07/T-1.3380
      vis=10.**(vis)
c from centipoise to N/M2sec
      vis=0.001*vis
c heat conductivity rk
      rk=9.59+0.00455*(T-273.15)
c from cal/sec-m-C
      rk=rk*4.1868
c Prandtl number
      pr=vis*cp/rk
      return
10    continue
c
c find T from h
c
      aa=4190.15
      bb=-(h+21248.01)
      cc=-21658316.
      T=(-bb+sqrt(bb*bb-4.*aa*cc))/(2.*aa)
```

```
      T=T+273.15
      go to 1
      end


c
c    ***** DESIGN STRENGTH SUBROUTINE *****
c Calculation of the Stress that Astar alloy can stand
c subroutine from Alkasys
c tt(K) Temperature, n=1 or 2 (it does not matter for sigpv)
c fpl (years) power life, itma Max number iterations
c ipr 1,0 Ok, no Ok pressure vessel
c sigpv(MPa) max stress of ASTAR alloy for pressure vessel
c
  subroutine desstr(tt,n,fpl,sigpv,sigrod,itma,ipr)

  real tt,nr,tmat,fpl,sigpv,sigrod,rho
  real b,c,v,sigma,theta,thet,dtds
        data tmat/1100./
  integer ll, n
        ipr=1
        if(tt.ne.0.) then
   if (tt.gt.tmat) then
    rho=.604
    b=67375.0
    c=3.548E9
    nr=1943
    v=.25875
   else
    rho=.31
    b=72614.8
    c=1.995E10
    nr=3768
    v=.03652
   endif
   do ll=1,n
    if (ll.eq.1) then
     thet=fpl*8766.139
    else
     thet=fpl*8766.139/2.
    endif
    sigma=1000
    theta=1/c*exp((b-nr*log(sigma)-v*sigma)/tt)
          ite=0
    dowhile ((abs((theta-thet)/thet)).gt.0.001)
     dtds=-theta*(nr+v*sigma)/sigma/tt
     sigma=sigma+(thet-theta)/dtds
           ite=ite+1
           if(ite.gt.itma) ipr=0
           if(ite.gt.itma) return
     if (sigma.lt.100) sigma=100
     theta=1/c*exp((b-nr*log(sigma)-v*sigma)/tt)
    enddo
    if (ll.eq.1) then
     sigpv=sigma*6.894757e-3
    else
     sigrod=sigma*6.894757e-3
    endif
   enddo
   sigpv=sigpv/1.5
  endif
  return
  end       !SUBROUTINE DESSTR


c
c function to compute thickness neutron shielding
```

```
c
      Function Shieldn(fluin,fluout,ish,tw,igam)
      common /cross/ AA,BB,signe,ishin,ishig
c
c Fluin,fluout, fast fluxes in/out; ish=1,  LiH+SS
c ish=2, B4C; tw(cm) gamma shielding (Tungstene)(if igam.ne.0)
c
      data remLi,remH,remFe,remW,remBC
     *    /1.01,1.00,1.98,3.13,5.1/
c removal cross section for fast neutrons of Li,H,Fe,W,B4C
c
      data Av,roLiH,roW,roBC,roFe,roLiss,ALiH,AFe,ABC,AW
     *    /0.602202,0.82,19.3,2.52,7.874,1.024,7.95,55.85,55.26,183.85/
      sigW=remW*Av*roW/AW
      ishin=1
      go to(10,20),ish
10    xLiH=(roLiss-roFe)/(roLiH-roFe)
      xFe=1.-xLiH
      sigre=(remLi+remH)*Av*roLiH*xLiH/ALiH+remFe*Av*roFe*xFe/AFe
      go to 30
20    sigre=remBC*Av*roBC/ABC
30    aa=alog(fluin/fluout)/sigre
      if(aa.le.0.) ishin=0
      bb=sigW/sigre
      signe=sigre
      Shieldn=aa-bb*tw
      return
      end


c
c Subroutine to calculate Gamma source
c
      Subroutine Gamso(Sn,sogam,PrU,fvr)
      common /Conce/ coBe,coTa,coU5,coU8,coLi,coW,coRe,coHf,coHe,coN,coO
c
c Input: Sn(1/s): neutron production,fvr: volume fraction of reflector
c Output:
c sogam(1): Gamma source between "0" and 1 Mev
c      (2):                        1  and 2
c      (3):                        2  and 3
c      (4):                        3  and 5
c      (5):                        5  and 7
c      (6):                        7  and 9
c      (7):                        > around 9 Mev
c and PrU(): fractions of gammas produced in U
      Dimension Sogam(7),Pf(7),Fp(7),GLi(7),GO(7),GTa(7),GBe(7),
     *GU(7),GRe(7),GHf(7),GW(7),GN(7),PrU(7)
c
c Pf: Prompt Fission Gammas; Fp: Fission Products Gammas(Both, Per Fiss)
c GNi to GCr:Captures Gammas(per Event) for indicated Elements Ni,..Cr
c Data from ANL-5800 page 631. For O: the 8 Mev released for the capture
c of one neutron was uniformly distributed in range 0-10 Mev.
c See:capture in O is almost 0.0; Remember B10 has (n,alpha) rather(n,g)
      Data(Pf(i),i=1,7)/3.1,2.11,.73,.26,.0428,0.,0./
      Data(Fp(i),i=1,7)/9.03,1.137,.4,0.,0.,0.,0./
      Data(GLi(i),i=1,7)/0.,0.,0.,0.,0.4,0.6,0./
      Data(GTa(i),i=1,7)/1.37,0.99,0.66,0.55,0.05,0.,0./
      Data(GW(i),i=1,7)/0.68,0.82,0.59,0.53,0.15,0.005,0./
      Data(GRe(i),i=1,7)/1.24,0.88,0.62,0.51,0.105,0.,0./
      Data(GHf(i),i=1,7)/1.37,1.37,0.85,0.52,0.12,0.005,0./
      Data(GN(i),i=1,7)/0.0,0.,0.,0.54,0.11,0.15,0.12/
      Data(GO(i),i=1,7)/0.16,0.16,0.16,0.32,0.32,0.32,0.16/
      Data(GBe(i),i=1,7)/0.,0.,0.,.54,.73,.0,.0/
      Data(GU(i),i=1,7)/2.54,1.78,.91,.34,.0,.0,0.041/
      Data Rnu/2.52/
      Data scLi,scTa,scW,scRe,scHf,scN,scO,scBe,scU5,scU8
```

```
     */0.000024,0.3,0.25,0.5,0.5,0.0002,0.0,0.0,0.2246,0.1613/
      Data sfU5,sfU8/1.4400,0.11201/
c
c See: Total number of Absorptions= Sn(Total production)
c    F: total number of fission
c    F5:  "      "       "      in U235
c    F8:  "      "       "      in U238
c    C5: total number of captures in U235
c    C8:   "      "      "    "     in U238
c   Ct:   "      "      " captures(code assumes reactor is well reflected)
      F=Sn/Rnu
      sif=coU5*sfU5+coU8*sfU8
      F5=F*coU5*sfU5/sif
      F8=f*coU8*sfU8/sif
      C5=F5*(scU5/sfU5)
      C8=F8*(scU8/sfU8)
      Ct=Sn-F
c
c distribute Ct in rest of materials
c assume average flux in core,pressure vessel and half that flux for the
c reflector, then disadvantage factor are
c
      disc=1./(1.-0.5*fvr)
      disr=0.5*disc
      den=disc*(coLi*scLi+coTa*scTa+coW*scW+coRe*scRe+coHf*scHf+coN*scN+
     *   coU5*scU5+coU8*scU8)+disr*(coO*scO+coBe*scBe)
      CLi=Ct*(coLI*scLi*disc/den)
      CTa=Ct*(coTa*scTa*disc/den)
      CW=Ct*(coW*scW*disc/den)
      CRe=Ct*(coRe*scRe*disc/den)
      CHf=Ct*(coHf*scHf*disc/den)
      CN=Ct*(coN*scN*disc/den)
      CO=Ct*(coO*scO*disr/den)
      CBe=Ct*(coBe*scBe*disr/den)
      Do 10 i=1,7
      GamU=F*(Pf(i)+Fp(i))+(C5+C8)*GU(i)
      Sogam(i)=GamU+CLi*GLi(i)+CW*GW(i)
     *    +CTa*GTa(i)+CBe*GBe(i)+CRe*GRe(i)+CO*GO(i)+CHf*GHf(i)+CN*GN(i)
      PrU(i)=100.*GamU/Sogam(i)
10    continue
      return
      end


c
c
      Subroutine Shieldga(Sogam,Dose,Dosea,Dosew,tiempo,tgam,ten,igam,
     *ishn,iten)
      Common /CROSS/ AA,BBB,signe,ishin,ishig
c
c Shieldga: Computes gamma shielding
c Sogam(i),i=1,7 Gamma Source (photons/sec); Dose(rad): Integrated
c Dose at load area; Dosew(rad): Dose without the shielding
c Dosea: dose for tgam
c igam:flag if =1 problems with iterations
c Tgam(cm): Thickness of Tungsten
c ten(cm) thickness of n shielding, ishn=1, LiH, ishn=2, B4C
c iten/0,1/ /ten is input,ten is iterated by this subroutine/
c tgam:iterated,if necessary,to have Dose=input
c Index(1): Gamma Processes between "0" and 1 Mev
c      (2):                        1  and 2
c      (3):                        2  and 3
c      (4):                        3  and 5
c      (5):                        5  and 7
c      (6):                        7  and 9
c      (7):                        > around 9 Mev
      common /Conce/ coBe,coTa,coU5,coU8,coLi,coW,coRe,coHf,coHe,coN,coO
```

```
      common /Gamsh/
     *dist,Req,shg(7),shgnb(7),Zeq(7),Dosewg(7),Doseg(7),itmag,epsg
     *,aten,Dosen,Doseng(7)
      dimension Sogam(7),GBe(7),GW(7),GO(7),GU(7),GH(7),GC(7),GFe(7),
     *b(4,7),a3(4,7),a4(4,7),zb(4),a(4),a1(4),a2(4),rh(7)
     *,ega(7),sign(7),An(7),A1n(7),A2n(7)
c
      data (GH(i),i=1,7)/0.173,0.103,0.0876,0.0579,0.0446,0.0371,0.0321/
      data (GC(i),i=1,7)/0.087,0.0518,0.0444,0.0304,0.0245,0.0213,.0194/
      data (GFe(i),i=1,7)/.0828,.0485,.0424,.033,.0304,.0295,.0294/
      data (GBe(i),i=1,7) /.0773,.0459,.0394,.0266,.0211,.0180,.0161/
      data (GW(i),i=1,7) /.125,.0492,.0437,.0402,.0418,.0438,.0465/
      data (GO(i),i=1,7) /.087,.0518,.0445,.0309,.0254,.0224,.0206/
      data (GU(i),i=1,7) /.176,.0548,.0484,.044,.0455,.0479,.0511/
c
c GBE...GU are the total mass attenuation coefficients(cm**2/g)
c for the 7 gamma groups. Li(Z=3)and B(z=5) are lumped with Be(z=4)

c Ta(z=73),Re(Z=75) and Hf(Z=72) are lumped with W(z=74)
c N(z=7) is lumped with O(Z=8)
c
      data (b(1,j),j=1,7)/8.,5.5,4.5,3.8,3.1,2.3,2.25/
      data (b(2,j),j=1,7)/10.,8.,5.5,3.75,2.9,2.35,2.0/
      data (b(3,j),j=1,7)/3.3,2.9,2.7,2.05,1.2,.7,.6/
      data (b(4,j),j=1,7)/1.65,2.45,2.15,1.65,.96,.67,.5/
c
      data (a3(1,j),j=1,7)/.11,.082,.074,.066,.064,.062,.060/
      data (a3(2,j),j=1,7)/.0948,.0895,.0788,.075,.0825,.0833,.095/
      data (a3(3,j),j=1,7)/.043,.069,.086,.118,.171,.205,.212/
      data (a3(4,j),j=1,7)/.032,.045,.097,.123,.175,.204,.214/
c
      data (a4(1,j),j=1,7)/.044,.093,.116,.130,.152,.150,.128/
      data (a4(2,j),j=1,7)/.012,.04,.07,.082,.075,.0546,.0116/
      data (a4(3,j),j=1,7)/.148,.188,.134,.070,.0,.052,.144/
      data (a4(4,j),j=1,7)/.296,.178,.103,.064,.059,.067,.08/
c
      data (zb(i),i=1,4)/13.,26.,74.,82./
c
c b(i,j),a2(i,j) and a4(i,j) are build up parameters for groups j
c and atomic number Z=Zb(i)
c
      data (rh(i),i=1,7)/4.5e+5,5.5e+5,7.e+5,8.e+5,8.5e+5,9.e+5,1.e+6/
c
c rh(i) Gamma Flux Mev/cm2sec that produces 1 Roentgen/hour
c
      data (Ega(i),i=1,7)/0.5,1.5,2.5,4.,6.,8.,10./
c
c ega(i) Mev Average Energy of gamma group i
c
      data ABe,ATa,AW,AO,AHf,AU5,AU8,ALi,ARe,ANi,Av,RoW
     */9.0122,180.948,183.85,15.9994,178.49,235.044,238.0508,6.939,186.2
     *,14.0067,0.602202,19.3/
      data ZBe,ZTa,ZW,ZU,pee,ZLi,ZBo,ZRe,ZHf,ZN,ZO
     */4.,73.,74.,92.,3.141593,3.,5.,75.,72.,7.,8./
c
      data roLiH,roBC,RoSS,RoLiS /0.82,2.92,7.874,1.024/
      igam=1
      ishig=1
      xLiH=(RoLiS-RoSS)/(RoLiH-RoSS)
      xSS=1.-xLiH
c
c  Attenuation*Build Up factor for N Shield(which also shield gammas)
c
      do 101 j=1,7
      go to(111,222),ishn
c ishn=1 LiH+SS
111   sig=RoLiS*(xLiH*0.8732*GBe(j)+xLiH*0.1268*GH(j)+xSS*GFe(j))
```

```
      sigZ=RoLiS*(ZLi*xLiH*0.8732*GBe(j)+ZH*xLiH*0.1268*GH(j)+ZFe*xSS*GF
     *e(j))
      ZZ=sigZ/sig
      go to 333
c ish=2 B4C
222   sig=RoBC*(0.7826*GBe(j)+0.2174*GC(j))
      sigZ=RoBC*(ZBo*0.7826*GBe(j)+ZC*0.2174*GC(j))
      ZZ=sigZ/sig
c ZZ is "equivalent Z for buildup factor
333   do 444 i=1,4
      a(i)=b(i,j)
      a1(i)=a3(i,j)
444   a2(i)=a4(i,j)
c interpolate in table as function of Z,
      Call Inter(zb,a,4,zz,am)
      Call Inter(zb,a1,4,zz,a11)
      Call Inter(zb,a2,4,zz,a12)
      An(j)=am
      A1n(j)=-a11
      A2n(j)=a12
      sign(j)=sig
101   continue
c end Calculation Attenuation/build up factor for n shield
c
c Start Calculation Self Shielding in Reactor
c
      do 10  j=1,7
c
      do 20 i=1,4
      a(i)=b(i,j)
      a1(i)=a3(i,j)
20    a2(i)=a4(i,j)
      sig=coBe*ABe*GBe(j)      +coTa*ATa*GW(j)        +coO*AO*GO(j)
     *   +coHf*AHf*GW(j)       +coU5*AU5*GU(j)        +coU8*AU8*GU(j)
     *   +coRe*ARe*GW(j)       +coLi*ALi*GBe(j)       +coW*AW*GW(j)
     *   +coN*ANi*GO(j)
      sig=sig/Av
      ZZ =coBe*ABe*GBe(j)*ZBe +coTa*ATa*GW(j)*ZTa    +coO*AO*GO(j)*ZO
     *   +coHf*AHf*GW(j)*ZHf  +coU5*AU5*GU(j)*ZU     +coU8*AU8*GU(j)*ZU
     *   +coRe*ARe*GW(j)*ZRe  +coLi*ALi*GBe(j)*ZLi   +coW*AW*GW(j)*ZW
     *   +coN*ANi*GO(j)*ZO
      ZZ=ZZ/Av
      zz=zz/sig
      Call Inter(zb,a,4,zz,am)
      Call Inter(zb,a1,4,zz,a11)
      Call Inter(zb,a2,4,zz,a12)
      Zeq(j)=zz
      b1=2.*sig*Req*(1.-a11)
      b2=2.*sig*Req*(1.+a12)
      exp1=0.
      exp2=0.
      if(b1.lt.50.) exp1=exp(-b1)
      if(b2.lt.50.) exp2=exp(-b2)
      shg(j)=3.*Am*(0.5-(1.-exp1*(1.+b1))/b1**2)/b1+
     *      3.*(1-Am)*(0.5-(1.-exp2*(1.+b2))/b2**2)/b2
      bb=2.*sig*Req
      expo=0.
      if(bb.lt.50.) expo=exp(-bb)
      shgnb(j)=3.*(0.5-(1.-expo*(1.+bb))/bb**2)/bb
10    continue
c
c End Calculation Self shielding
c
c
c Dosew and Dosewg() are the doses without any shield(n or gamma)
c Dosen and Doseng() are the doses without W(gamma shield) but inclu-
c ding the shielding effects of the neutron shield(either input or itera-
```

```
c ted
      time=tiempo/3600.
      Dosew=0.
      Dosen=0.
      if(iten.eq.1)  ten=AA
      if(iten.eq.1.and.ishin.eq.0) ten=0.
      do 30 i=1,7
      tsn=Sign(i)*ten
      Shin=exp(-tsn)*(An(i)*exp(-tsn*Aln(i))+(1.-An(i))*exp(-tsn*A2n(i))
     *     )
      Dosewg(i)=time*Sogam(i)*shg(i)*Ega(i)/(4.*pee*dist**2*rh(i))
      Doseng(i)=Shin*Dosewg(i)
      Dosew=Dosew+Dosewg(i)
30    Dosen=Dosen+Doseng(i)
      if(Dosen.le.Dose) ishig=0
      if(ishig.eq.0) tgam=0.
      if(ishig.eq.0) return
c
c Return if there is no need for gamma shield (thick W=0.)
c
      t1=0.
      Dose1=Dosen
      f1=Dose1-Dose
c
c Starts iteration on tgam:
c First look for
c t2 such that Dose(t2) <Dose
c
      it=0
      sig=GW(1)*RoW
      t2=alog(Dosewg(1)/Dose)/sig
 51   dose2=0.
      do 40 i=1,7
      am=b(3,i)
      al1=-a3(3,i)
      al2=a4(3,i)
      sig=GW(i)*RoW
      BFA=am*exp(-al1*sig*t2)+(1.-am)*exp(-al2*sig*t2)
40    dose2=dose2+Dosewg(i)*exp(-sig*t2)*BFA
      if(dose2.lt.dose) go to 50
      t2=1.01*t2
      it=it+1
      if(it.le.itmag) go to 51
      igam=0
      write(8,*) ' Problem with Calculation of Gamma Shielding'
50    continue
      f2=Dose2-Dose
      if(igam.eq.0) return
      it=0
61    tgam=0.5*(t1+t2)
      dosea=0.
      do 41 i=1,7
      am=b(3,i)
      al1=-a3(3,i)
      al2=a4(3,i)
      sig=GW(i)*RoW
      BFA=am*exp(-al1*sig*tgam)+(1.-am)*exp(-al2*sig*tgam)
c
c recompute neutron shielding(ten) because W also shield neutrons
c see: if ten<0. program set ten to 0.(ie no need of n shielding
c because W is enough also if ishin=0(fluin lt prescribed fluout)
c
      if(iten.eq.1)  ten=AA-BBB*tgam
      if(ten.lt.0.) ten=0.
      if(iten.eq.1.and.ishin.eq.0) ten=0.
      tsn=Sign(i)*ten
      Shin=exp(-tsn)*(An(i)*exp(-tsn*Aln(i))+(1.-An(i))*exp(-tsn*A2n(i))
```

```
     *     )
      Doseg(i)=Dosewg(i)*exp(-sig*tgam)*BFA*Shin
41    dosea=dosea+Doseg(i)
c recompute n shielding, because it might be smaller than input max flu-
c ence
      eco=(ten+BBB*tgam)*signe
      aten=exp(-eco)
      fa=Dosea-Dose
      rel=abs(dosea-dose)/dose
      if(rel.le.epsg) go to 60
      call Bisect(t1,t2,tgam,f1,f2,fa,iwa)
      it=it+1
      if(it.le.itmag.and.iwa.ne.1) go to 61
      igam=0
      write(8,*) ' Problems with calculation of Gamma Shielding'
60    continue
      return
      end
c


c Compute the two solutions,d1 and d2, that produce energy E(Mwd)
c inputs: asymptotic(das) and optimum(dop) sizes for enrichment enr
c and p/d pod
c output: d1 and d2
c
      Subroutine Size(das,dop,E,enr,pod,d1,d2)
c
      common /Soze/ its,epss
c
      common /MaBaSi/ R,H,ishape
c
c left side limit
c
      ri=1.
      it=0
10    x1=das*(1.+0.01/ri)
      if(ishape.eq.2) R=x1
      if(ishape.eq.3) H=x1
      e1=Balance(ibu,enr,pod)-E
      if(e1.gt.0.) go to 20
      ri=ri+1.
      it=it+1
      if(it.gt.its) go to 100
      go to 10
20    x2=dop
      if(ishape.eq.2) R=x2
      if(ishape.eq.3) H=x2
      e2=Balance(ibu,enr,pod)-e
30    it=0
35    xav=0.5*(x1+x2)
      if(ishape.eq.2) R=xav
      if(ishape.eq.3) H=xav
      Ea=Balance(ibu,enr,pod)-e
      test=abs(Ea)/E
      if(test.le.epss) go to 50
      call bisect(x1,x2,xav,E1,E2,Ea,iwa)
      it=it+1
      if(iwa.eq.1) go to 100
      if(it.gt.its) go to 50
      go to 35
50    d1=xav
c
c right side limit
c
      x1=dop
      if(ishape.eq.2) R=x1
```

```
      if(ishape.eq.3) H=x1
      E1=Balance(ibu,enr,pod)-e
21    it=0
41    x2=x1*(1.+0.10)**(it+1)
      if(ishape.eq.2) R=x2
      if(ishape.eq.3) H=x2
      e2=Balance(ibu,enr,pod)-e
      if(e2.gt.0.) go to 31
      it=it+1
      if(it.gt.its) go to 100
      go to 41
31    it=0
36    xav=0.5*(x1+x2)
      if(ishape.eq.2) R=xav
      if(ishape.eq.3) H=xav
      Ea=Balance(ibu,enr,pod)-e
      test=abs(Ea)/E
      if(test.le.epss) go to 51
      call bisect(x1,x2,xav,E1,E2,Ea,iwa)
      it=it+1
      if(it.gt.its.or.iwa.eq.1) go to 100
      go to 36
51    d2=xav
      return
100   write(8,101) it
101   format(//' Program stop at subroutine Size'/' iterations=',i10/)
      stop
      end


c
c Subroutine bupo
c
      Subroutine bupo(bup,rkin,c50,c80,dka)
      common /mabu/ c5,c8,c9,p9,cfp,ftin
c
c input: bup(burnup,fraction),rkin(k Infinity),c50,c80(1/cmbarn) initial
c concentration of U235 and U238
c
c output: dka(%, and modulus) reactivity effects of burnup
c
c c5,c8,c9,cfp, concentrations(U235,U238,Pu239 and Fission Prod) at EOL
c ftin: fluence at EOL, p9: total production of Pu239
      dimension flutin(20),buta(20)
      data rnu5,rnu8,rnu9/2.52,2.61,2.91/
      data sa5,sa8,sa9,safp/1.6646,0.2733,2.11,0.57/
      data sf5,sf8,sf9/1.4400,0.11201,1.70/
      data sc8/0.1613/
c compute table BU vs integrated flux
      enr=c50/(c50+c80)
      e5=sf5/sa5
      e8=sf8/sa8
      a8=e8*(1./enr-1.)
      a9=sc8*(1./enr-1.)/(sa9-sa8)
      a91=sf9/sa8
      e9=sf9/sa9
      nbu=20
      dft=1./(sa5*19)
      do 20 i=1,nbu
      ftin=(i-1)*dft
      flutin(i)=ftin
20    buta(i)=e5*(1.-exp(-sa5*ftin))+(A8+A9*A91)*(1.-exp(-sa8*ftin))-
     *     A9*e9*(1.-exp(-sa9*ftin))
      call Inter(buta,flutin,nbu,bup,ftin)
      srest=c50*(rnu5*e5/rkin-1.)*sa5+c80*(rnu8*e8/rkin-1.)*sa8
      c5=c50*exp(-sa5*ftin)
      c8=c80*exp(-sa8*ftin)
```

```
      c9=(sc8*c80/(sa9-sa8))*(exp(-sa8*ftin)-exp(-sa9*ftin))
      p9=(c80-c8)*sc8/sa8
c     cfp=2.*(c50*sf5+c80*sf8)*ftin
      cfp=2.*((c50-c5)*e5+(c80-c8)*e8+(p9-c9)*e9)
      rkinbu=(rnu5*c5*sf5+rnu8*c8*sf8+rnu9*c9*sf9)/
     *      (c5*sa5+c8*sa8+c9*sa9+cfp*safp+srest)
      dka=rkinbu/rkin-1.
      dka=-100.*dka
      return
      end
```

INTERNAL DISTRIBUTION

1. C. E. Oliver
2. R. Cooper
3-7. F. C. Difilippo
8. R. S. Holcomb
9. G. E. Michaels
10. W. C. Jordan
11. R. B. Perez
12. R. T. Primm, III
13. J. P. Reiner
14. J. C. Turner
15. R. C. Ward
16-20. B. A. Worley
21. EPMD Reports Office
22-26. Laboratory Records Department
27. Laboratory Records, ORNL-RC
28. Document Reference Section
29. Central Research Library
30. ORNL Patent Section

EXTERNAL DISTRIBUTION

31. S. Anghaie, 202 NCS, University of Florida, Gainesville, FL 32611.
32. Gary Bennett, NASA Headquarter, Code RST, Washington, DC 20546.
33. Jeff George, NASA Lewis Research Center, MS AAC-2, 21000 Brookpark Road, Cleveland, OH 44135.
34. Nils J. Diaz, INSPI, University of Florida, Gainesville, FL 32611.
35. Dean Dobranovich, Sandia National Laboratory, P.O. Box 5800, Albuquerque, NM 87185-5800.
36-40. M. P. Doherty, NASA Lewis Research Center, MS-500/322, 21000 Brookpark Road, Cleveland, OH 44135.
41. Al Juhasz, NASA Lewis Research Center, MS 301-5, 21000 Brookpark Road, Cleveland, OH 44135.
42. A. C. Zuppero, DOE-INEL, P.O. Box 1625, MS-3413, Idaho Falls, ID 83415.
43. R. W. Brockett, Harvard University, Pierce Hall, 29 Oxford Street, Cambridge, MA 02138.
44. D. J. Dudziak, Department of Nuclear Engineering, 110B Burlington Engineering Labs, North Carolina State University, Raleigh, NC 27695-7909.
45. J. E. Leiss, Rt. 2, Box 142C, Broadway, VA 22815.
46. N. Moray, Department of Mechanical and Industrial Engineering, University of Illinois, 1206 West Green Street, Urbana, IL 61801.
47. M. F. Wheeler, Department of Mathematics, Rice University, P.O. Box 1892, Houston, TX 77251.
48. Office of Assistant Manager for Energy Research and Development, Department of Energy, Oak Ridge Operations Office, P.O. Box 2008, Oak Ridge, TN 37831.
49-50. Office of Scientific and Technical Information, U.S. Department of Energy, P.O. Box 62, Oak Ridge, TN 37831.
51. Report Control, NASA Lewis Research Center, 21000 Brookpark Road, Mail Stop 60/1, Cleveland, OH 44135.
52-53. Library, NASA Lewis Research Center, 21000 Brookpark Road, Mail Stop 60/3, Cleveland, OH 44135.
54. Technology Utilization Office, NASA Lewis Research Center, 21000 Brookpark Road, Mail Stop 7/3, Cleveland, OH 44135.
55-79. NASA Center for Aerospace Information, ATTN: Accession Branch, P.O. Box 8757, Baltimore, MD 21240-0757.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1994	Contractor Report

4. TITLE AND SUBTITLE
Scoping Calculations of Power Sources for Nuclear Electric Propulsion

5. FUNDING NUMBERS

6. AUTHOR(S)
Felix C. Difilippo

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Oak Ridge National Laboratory
Engineering Physics and Mathematics Division
P.O. Box 2008, Bldg. 6035
Oak Ridge, TN 37831-6363

8. PERFORMING ORGANIZATION REPORT NUMBER
ORNL/TM-12703

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Lewis Research Center
21000 Brookpark Road
Cleveland, OH 44137

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CR-191133

11. SUPPLEMENTARY NOTES
Felix C. Difilippo, Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, TN 37831-6363 (615) 574-6188.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

This technical memorandum describes models and calculational procedures to fully characterize the nuclear island of power sources for nuclear electric propulsion. Two computer codes were written: one for the gas-cooled NERVA derivative reactor and the other for liquid metal-cooled fuel pin reactors. These codes are going to be interfaced by NASA with the balance of plant in order to making scoping calculations for mission analysis.

14. SUBJECT TERMS
Nuclear electric propulsion, Reactor modeling, Liquid metal cooled reactor, Gas cooled reactor

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	None