Digital Simulation of Spatial Xenon Oscillations

Olsson, Gustaf

1969

Document Version:
Publisher's PDF, also known as Version of record

Link to publication

Citation for published version (APA):
Olsson, G. (1969). *Digital Simulation of Spatial Xenon Oscillations*. (Research Reports TFRT-3014). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
1

DIGITAL SIMULATION OF SPATIAL XENON OSCILLATIONS

Gustaf Olsson

ABSTRACT

A nonlinear model of xenon spatial oscillations in one dimension based on one group diffussion theory and finite differences is presented. The process has been simulated on a digital computer.

In the report is discussed the domain of linearity of the model. The influence of different core parameters on reactor stability and on amplitude of the oscillations is presented. The results are compared to other models.

Influence of nonlinear terms, such as temperature feedback and control rod, indicate that periodic solutions can appear. This has been predicted earlier with a simple two point model. The rod movement has a very big influence on amplitude and character of the oscillations. It can be explained from a very simple model of the core.

The simulations have indicated suitable approximations to get a space independent nonlinear model of the process. Physical interpretation and drawbacks of this model are discussed.

TABLE OF CONTENTS

Table of contents
Contd.

1. INTRODUCTION

Xenon spatial instability is a problem in large power reactors. It depends on the fission product xenon, which has a tremendously large neutron cross section. The reactivity feedback from xenon can cause oscillations in the power spatial distribution, which have to be avoided.

The fundamental equations are rather complicated and it is impossible to treat them analytically as they stand. In an earlier report [5] is described different mathematical models, and an extensive study has been made of a simple two point model. This model has given physical insight in the problem.

In this report the equations are simulated on a digital computer. The study includes the xenon instability problem along the core axis. The importance of axial oscillations is discussed in [5].

The simulations were started at Swedish State Power Board, Stockholm, where the main part of the program TRAXEN was written. It was necessary to know the xenon stability for the Marviken reactor.

The purpose of the report is moreover:

- to study the influence of nonlinear terms and compare the result to previous linear model studies,
- to examine the influence from a control rod on stability,
- to calculate the amplitudes for the transients of different disturbances.

In the next section the results are summed up.

SUMMARY OF THE RESULTS

The report deals with the axial problem of xenon spatial instability. In [5] is discussed the reasons for choosing the axial direction.

The dynamical behaviour is governed by a nonlinear one group diffussion equation for the neutron flux, which is coupled to the ordinary nonlinear differential equations for xenon and iodine. The equations are presented in 2.1. The solution method is described in 2.2. Due to the nonlinear character it is necessary to know time and space distribution of the disturbances. The most probable ones are presented in 2.3. In 2.4 is the choice of core parameters reported.

The simulations have been divided into two main fields, small and large disturbances.

In chapter 3 is discussed the region for linearity. Inside this domain the results from the simulations can be compared to those of linear models. In 3.2 is discussed how stability can be determined from the trajectories. Critical height as function of different core parameters is presented in 3.3. The results are compared to linear finite difference models and modal expansion models with good agreement.

Of technological reasons the amplitude of the transients is important to know. It depends strongly on core height. The maximum flux deviation as function of core height for step disturbances is presented in 3.4. It is compared to a two point model from [5]. The agreement is rather good.

For large disturbances there are mainly two nonlinear terms, which will affect the stability of the solutions, the absorbtion term and the temperature coefficient. Depending on rod insertion length in equilibrium rod absorbtion, amplitude and direction of the disturbance the effect of the rod is very different. In 4.1 is discussed the rod movement from a very simple model. The results from the simulations are then compared to those of the simple model and the agreement is surprisingly good.

The trajectories can be damped or amplified by the rod. Due to this fact an unstable limit cycle occurs in some cases. For rod control is shown in 4.2 that there are stable trajectories for small disturbances but unstable ones for large disturbances.

In 4.2 is also shown that stable periodic solutions can appear. This type of performance was predicted with a simple two point model in [5]. In the nonlinear case the amplitude of the transients are no longer proportional to the disturbances.

In 4.3 is shown that the rod, the temperature coefficient as well as the direction of the disturbance are very important for the amplitude.

In chapter 5 is derived a space independent nonlinear xenon model. Similar models have been derived by other authors. It is possible to give a nice physical interpretation of the equations. However, it is shown that the model describes badly what happens for big disturbances.

2. MATHEMATICAL MODEL

The xenon process is described by the coupling between the neutron distribution equation and the radioactive decay differential equations of xenon and iodine.

Since the xenon oscillations appear only in large thermal reactors the neutron distribution is treated by one group diffussion theory. Due to the long period of the oscillations the neutron flux can be regarded stationary, why the flux distribution is completely determined by the time dependent material buckling.

The control rod is simplified to have space independent absorbtion.

In 2.2 is briefly described the solution method.

The type of disturbances which are relevant are presented in 2.3, while the choice of core parameters are discussed in 2.4.

2.1 FUNDAMENTAL EQUATIONS OF THE XENON PROCESS

2.1.1 NEUTRON DISTRIBUTION

The fundamental equations and the conditions for the xenon process are described in [5] but are repeated here by convenience. The motivation to study the axial oscillations is also found there, why we directly describe the one group diffusion equation in one dimension:

$$\frac{\partial}{\partial z}\left[D(z,t)\frac{\partial}{\partial z}\Phi(z,t)\right] + (\nu\,\Sigma_f - \Sigma_a)\,\Phi(z,t) = 0 \qquad (1)$$

where $\Phi(z,t)$ is the neutron thermal flux and
$D(z,t)$ is the time and space dependent diffusion.

After division with a suitable mean value of D, called D^o, equation (1) is transformed to:

$$\frac{\partial}{\partial z}\left[E(z,t)\frac{\partial}{\partial z}\Phi(z,t)\right] + B^2(z,t)\,\Phi(z,t) = 0 \qquad (2)$$

where $B^2(z,t)$ is the material buckling and
$E(z,t)$ is a normalized diffusion parameter.

The boundary conditions are:

$$\Phi(z,0) = \Phi^o(z)$$
$$\Phi(0,t) = \Phi(H,t) = 0 \qquad (3)$$

$$\int_z K(z)\,\Phi(z,t)\,dz = P(t) \qquad (4)$$

We approximate the space derivatives by finite differences and get from (2):

$$\left[(E_{k+1} - E_k)(\Phi_{k+1} - \Phi_k) + E_k(\Phi_{k+1} - 2\Phi_k + \Phi_{k-1})\right] + h^2\,B_k^2\,\Phi_k = 0$$
$$k = 1, \ldots, N$$

where the subscript means space point

or

$$\Phi_{k+1} \cdot E_{k+1} - \Phi_k(E_{k+1} + E_k) + \Phi_{k-1} E_k + h^2 B_k^2 \Phi_k = 0 \qquad (5)$$

$$k = 1, \ldots, N$$

where we have defined:

$$h = \frac{H}{N+1}$$

The boundary conditions are described in discrete form as:

$$\Phi_o(t) = \Phi_{N+1}(t) = 0$$

$$\Phi_k(0) = \Phi_k^o \qquad k = 1, \ldots, N \qquad (6)$$

$$\sum_{\nu=1}^{N} K_\nu \cdot \Phi_\nu(t) = P(t)$$

The buckling can be expanded into two parts, one equilibrium part, B^{2*}, and one perturbed part,

$$B_k^2(t) = B_k^{2*} + \alpha_k \cdot \left(\Phi_k(t) - \Phi_k^o\right) + \beta \cdot \left(X_k(t) - X_k^o\right) +$$

$$+ c_k(t) + u_k(t) \qquad (7)$$

The coefficients α_k and β express the dependence of buckling on changes in flux (and temperature) and xenon respectively. The term c_k is the influence on buckling from control rod movement and u_k is a general control term, available for the operator.

2.1.2 XENON AND IODINE EQUATIONS

Xenon concentration is built up mainly by radioactive decay of iodine and a smaller part by the fission. It is destroyed by capture of neutrons and by radioactive decay. Iodine is also got by fission and is destroyed by radioactive decay to xenon.

The xenon and iodine differential equations thus read:

$$\frac{dX_k}{dt} = - \lambda_x X_k(t) + \lambda_i I_k(t) + \gamma_x \sigma_x \Phi_k(t) - \sigma_x X_k(t) \Phi_k(t) \qquad (8)$$

$$\frac{dI_k}{dt} = - \lambda_i I_k(t) + \gamma_i \sigma_x \Phi_k(t) \qquad k = 1, \ldots, N \qquad (9)$$

The boundary conditions are:

$$\begin{aligned} X_o(t) &= X_{N+1}(t) = 0 \\ I_o(t) &= I_{N+1}(t) = 0 \\ X_k(0) &= X_k^o \qquad k = 1, \ldots, N \\ I_k(0) &= I_k^o \qquad k = 1, \ldots, N \end{aligned} \qquad (10)$$

2.1.3 ABSORBTION TERM IN BUCKLING

The absorbtion or control term c in the buckling (7) represents an absorbtion which must be added or subtracted in order to maintain criticality of the reactor.

As shown in [5], chapter 2.7, only one parameter is necessary to describe uniquely the absorbtion distribution c_k ($k = 1, \ldots, N$). Physically we have discussed three alternatives, which we call:

- rod control with variable insertion length,
- rod control with variable absorbtion,
- homogeneous control.

(i) ROD CONTROL WITH VARIABLE INSERTION LENGTH

We assume the absorbtion constant along the rod, called c^1. The insertion length is called λ, where $0 \leq \lambda \leq 1$, and λ determines uniquely the absorbtion. Of computational reasons λ can exceed these limits in order to maintain criticality. Then we have the two cases:

$$\text{A.} \quad c(z,t) = \lambda(t) \cdot c^1 \text{ if } \begin{cases} \lambda < 0 \\ \lambda > 1 \end{cases} \qquad 0 \leq z \leq H$$

or in discrete form

$$c_k(t) = \lambda(t) \cdot c^1 \qquad \begin{cases} \lambda < 0 \\ \lambda > 1 \end{cases} \qquad k = 1, \ldots, N \tag{11}$$

and

$$\text{B.} \quad c(z,t) = \begin{cases} c^1 & 0 \leq z \leq \lambda H \\ 0 & \lambda H < z \leq H \end{cases} \qquad \text{if } 0 \leq \lambda(t) \leq 1$$

or in discrete form

$$c_k(t) = \begin{cases} c^1 & \text{for } k < [N\lambda] + 1 \\ \{N\lambda - [N\lambda]\} \cdot c^1 & \text{for } k = [N\lambda] + 1 \\ 0 & \text{else} \end{cases} \qquad k = 1, \ldots, N \quad 0 \leq \lambda(t) \leq 1 \tag{12}$$

where $[y]$ assigns integer part of y.

(ii) ROD CONTROL WITH VARIABLE ABSORBTION

Now the insertion is assumed to be constant and the absorbtion is variable. Physically this case can be interpreted as if many fine control rods are inserted from top to this insertion length. We regard all fine rods as one big rod with variable absorbtion. The absorbtion along this rod cluster is assumed to increase or decrease if some fine rod is moved in or out. We determine a point K^* inside the core and get:

$$c_k(t) = \lambda(t) \cdot c^1 \qquad \text{if } k \leq K^* \leq N \tag{13}$$

$$c_k = 0 \qquad \text{else}$$

The parameter λ now determines the absorbtion along the rod.

(iii) HOMOGENEOUS CONTROL

The absorbtion is constant in the whole core, and we get this case by setting $K^* = N$ in (ii)

$$c_k = \lambda(t) \cdot c^1 \qquad k = 1, \ldots, N \tag{14}$$

Physically this control may be regarded as an absorbtion control by a liquid or gaseous absorber.

2.2 SOLUTION METHOD

A Fortran program TRAXEN (TRAnsients of XENon) is written to solve the system equations (5), (8), (9), for all types of disturbances. The program is described in more detail in appendix 3.

As input data we must give geometrical data, mean flux, nuclear constants, control configuration, time and spatial distribution of disturbances, core parameters, such as spatial distribution of buckling, temperature coefficient and diffussion, power condition and desired accuracy. The program calculates both equilibrium flux distribution and transients of the flux, xenon and iodine distributions.

As the neutron diffussion equation (5) is always stationary, the program calculates iteratively the new flux distribution in every time step from the known value of the state in the previous time step.

The equations are integrated with a Runge - Kutta method, corrected with Richardson extrapolation. The numerical methods are described in appendix 2.

In introductory simulations have been tried different time step lengths. It was found that one hour was suitable, and regarding the period of about 24 hours it is accurately enough.

The program is made for maximum 50 node points. As the computing time increases as about N^2, it is necessary to compromize between accuracy and computing time. In [5] chapter 4.2.1 we found that the stability limit could be accurately determined with 20 space points, and for the following simulations we have chosen N = 20 throughout.

2.3 TYPE OF DISTURBANCES

For the nonlinear analysis it is very important to know the type of disturbances which are relevant. As we want to know the conditions for xenon instability, in order to build the reactor inherently stable, we want to know the most serious disturbance in order to be able to predict, if the actual reactor is stable or not for all possible disturbances.

2.3.1 SPATIAL DISTRIBUTION OF THE DISTURBANCE

Due to the complexity of the problem we cannot analytically derive the most serious disturbance. However, we can regard the xenon problem as mainly first overtone oscillation. Therefore it seems natural to give the flux a disturbance mainly in the first overtone by moving reactivity from one half of the core to the other.

The disturbance u(z,t) (7) is assumed to be separable in space and time:

$$u(z,t) = r(t) \cdot R(z) \qquad (15)$$

We have standardized the distribution R(z) to be constant in every half, as in figure 1.

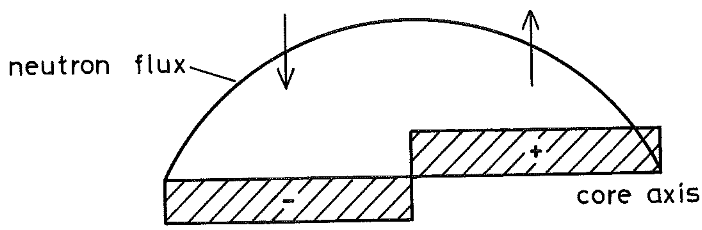


Fig. 1: Space distribution of the reactivity disturbance of the neutron flux.

The shattered areas mark the movement of reactivity. We call the variable r(t) the amplitude of the disturbance. It can, of course, be both positive and negative.

As an example of a disturbance with a distribution of this shape we will regard a refuelling process.

We assume the total power to be constant. The refuelling process takes place during the operation of the reactor. Before the refuelling the rod (rods) may be in a position like figures 2a or 2b.

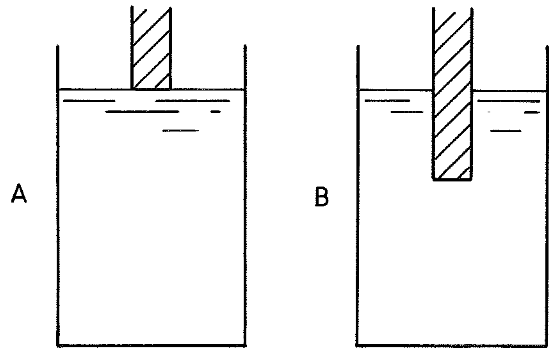


Fig. 2: Examples of rod positions before refuelling

When new fuel elements are inserted, the reactivity in the core increases, why the rods must be inserted in order to hold the power constant (see fig. 3: a, b)

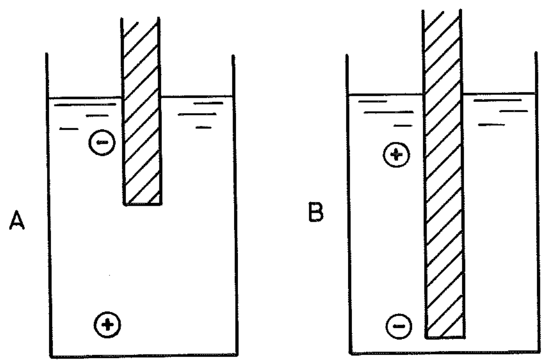


Fig. 3: Examples of position of control rod after refuelling. The plus and minus signs stand for changes in reactivity.

Figure 3 shows the sign of the disturbances in the two halves of the core for these two standard cases.

2.3.2 TIME DISTRIBUTION OF THE DISTURBANCE

In the linear and nonlinear stability analysis the zero solution is disturbed during a finite time. We have chosen the time distribution $r(t)$ to be a rectangular pulse of one or two hours duration.

Except the stability analysis we want to know the amplitude of the transient, which is caused by a reactivity disturbance.

The time function $r(t)$ has been chosen to a step in some cases. This disturbance may be relevant e.g. for a refuelling process.

The amplitude of the disturbances has been chosen between 10 and 1000 pcm in the simulations. The biggest disturbance may occur in a reactor when a fuel element accidently falls down in the reactor. During a refuelling process the movement of reactivity is likewise considerable.

2.4 CHOICE OF REACTOR CORE PARAMETERS

All parameters and their values are found in appendix 1. The equations are valid for both heavy water and light pressurized water reactors. In these simulations is used only heavy water reactor datas. They are much standardized but several core parameters are taken from the Marviken reactor, [4], which was the first object for this study.

The most interesting parameters affecting the stability are:

- core height H
- temperature (or flux) coefficient α
- mean flux level $\bar{\Phi}$
- flux shape $\bar{\Psi}$, B^{2*}
- absorbtion configuration c
- type of disturbance

The rod absorbtion c^1 (12) is very important, as the insertion of the rod is dependent of c^1. Likewise the rod movement during an oscillation is dependent of c^1, which will have big influence on stability, as we will discuss in section 4.2.

In Marviken a shim rod will hold 1000 pcm. As our rod may represent several fine rods we have chosen 500 pcm as a representative value of the rod absorbion, when it is inserted in the whole core. In this study we are only interested in the general behaviour for a standard reactor, but in a case study this absorbtion should be carefully taken into account.

Even if the TRAXEN program is prepared for hydrodynamic studies with a variable diffussion constant we do not take the void into account in this study. Thus:

E_k = constant = 1 for all k.

Likewise, we have incomplete information about the space distribution of α (eq. (7)) and have treated α as space independent.

The mean value of the flux is calculated as:

$$\bar{\Phi} = \frac{1}{N+1} \sum_{k=1}^{N} \Phi_k \tag{16}$$

When the numerical values of the parameters are not specially mentioned, we have used the values from appendix 1.

3. SMALL DISTURBANCES

Several previous reports on the xenon problem have treated the linearized equations, just in order to find the stability boundaries. An extensive reference list is found in [5].

By studying the nonlinear equations for different disturbances from equilibrium it is possible to get a good feeling for the domain where the linear approximation is valid with reasonable accuracy. This is discussed in 3.1.

In 3.2 is defined stability criteria and in 3.3 is shown how critical height depends on different core parameters.

The amplitude of the transient is an important measure of performance of a reactor. We measure the maximum flux deviation from equilibrium in the core. The amplitude depends on several core parameters, mainly on core geometry. This relationship is discussed in 3.4. It is possible to get an estimation of the amplitude as function of core height from a two point model. A comparison is made between simulation results and an analytical derivation of the amplitude.

3.1 THE DOMAIN OF LINEARITY

One purpose of this study was to find the importance of the nonlinear terms in different situations. It is impossible to give a general criterion of the boundaries of the linear region. We will, however, give some hints where reasonable accuracy of the superposition principle is to be found.

An easy criterion to check is the rod movement. From (2:7) we find that this is a small term. In [5] chapter 2 was found that the control term had no influence on linear stability for a symmetric two space point model or for a linear finite difference multipoint model with flat flux. It is reasonable to assume that the rod movement at small disturbances has small influence on stability even for other flux forms.

In the simulations we use the rod movement as the primary indicator to determine whether the disturbance can be called small or large.

Further, the amplitude of the transient shall be linearly related to the amplitude of the disturbance.

The permitted domain for small disturbances depends strongly on the core parameters, and it is, of course, larger for smaller core sizes and more stable reactors (stability is defined in 3.3).

We show some numerical examples to illustrate the linear domain. We have measured the first maximum of the amplitude of some transients for different disturbances and have checked the linearity by the superposition principle. As a measure of transient amplitude we use the maximum flux deviation from equilibrium in the core. The result is described in table 1.

Table 1: Maximum amplitude of neutron flux deviation transients related to different disturbances. The flux form is found in fig. 1:E. The control is homogeneous. $H_{crit} = 7.5$ m.

Core height	Step disturbance r(t) pcm	Normalized values Disturbance	Ampl. of transient
7.5 m	10	1	1
	20	2	2.00
	50	5	4.98
	100	10	9.64
7.0 m	20	2	2
	50	5	5.00
	100	10	9.80
6.0 m	20	2	2
	100	10	10.04

As table 1 shows we can regard 50 pcm as a small disturbance (within 1% accuracy) for a reactor near the stability limit, while 100 pcm gives a rather good linear relationship for a 6 m reactor for this special flux, representative for the Marviken reactor (see fig. 1:D). When studying the linear stability we have used 10 - 50 pcm as disturbances.

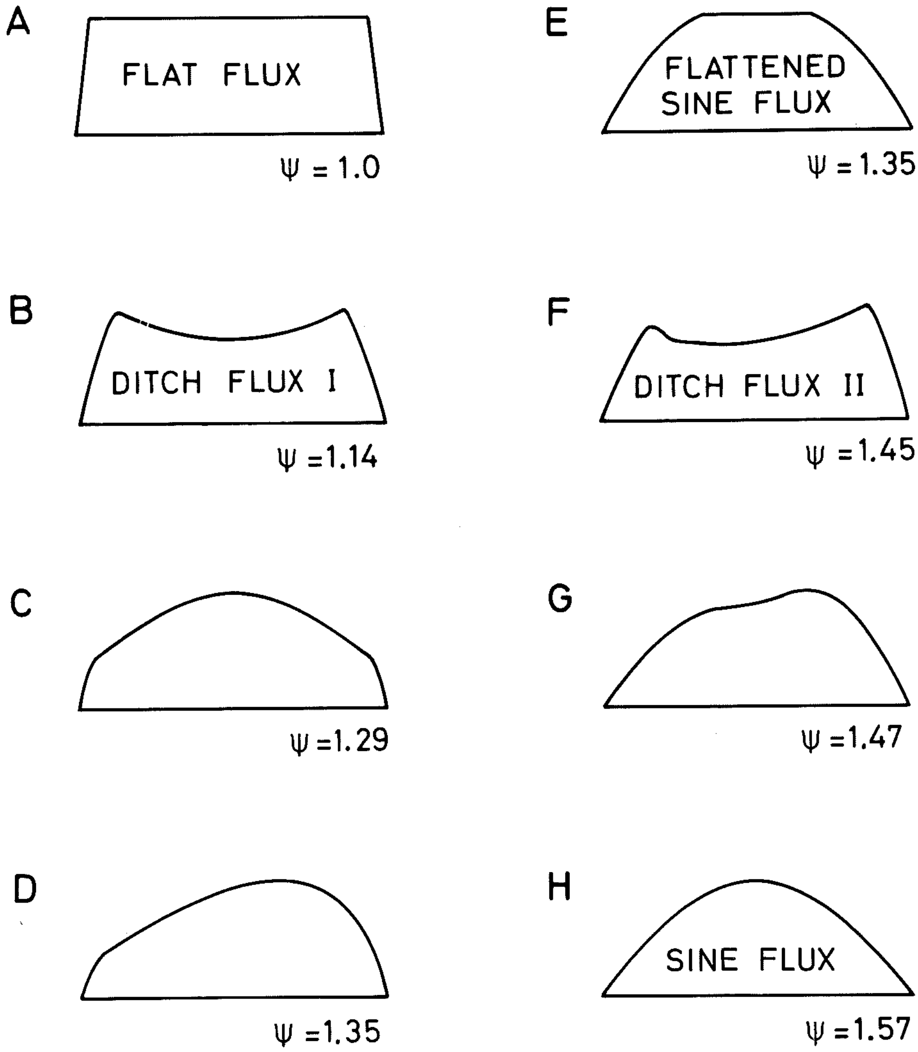


Fig. 1 - Different neutron flux axial distributions. Fluxes A,B,C,E,H, are symmetric.

3.2 STABILITY IN CASE OF SMALL DISTURBANCES

We use the definition of stability in the sense of Lyapunov, which means that a trajectory is stable (asymptoticly) when the deviation of the state after a disturbance of the zero solution converges to zero. Otherwise it is unstable.

The stability can be measured or calculated in several ways. In [5] we regarded the eigenvalues of the system matrix. Here we will study the convergence of the trajectories of every space point.

It is sufficient to check the stability in one point. This is realized by studying (2:5), which couples all the state variables strongly together. The system can be described by:

$$\dot{x} = A \cdot x$$

where x is the state vector and A is a 2N x 2N matrix.

Eq. (2:5) causes every state variable to be coupled to its neighbours in A. Thus if one state variable is unstable, all the state variables must be unstable, due to the strong coupling in the system matrix.

It is even sufficient to decide convergence or divergence near the stability limit by studying only two consecutive extremum values of the transient. This is important, as the computation takes much time.

If a transient seems to be at the stability limit when looking at two peaks, one must be sure that no slowly varying unstable oscillation is added to the dominating oscillation of 24-hours period. In [5] sect. 4.2.2 is calculated the eigenvalues for two standard fluxes at the stability limits, and this shows that no unstable oscillation is added to the dominating oscillation.Only fast decaying oscillations appear beside the undamped oscillation.

A number of simulations of about twenty periods (> 500 h) for different fluxes have confirmed this statement.

One observation may be done. We have neglected influence from the fission product samarium, which has a cross section about a hundred of that of xenon and may cause small oscillations of a periodicy of several days.

We will often express the stability in a significant core parameter and define the critical height as the core height where the stability limit is reached.

3.3 CRITICAL HEIGHT AS FUNCTION OF DIFFERENT CORE PARAMETERS

As pointed out in 3.1 the rod movement can be neglected in all the linear studies. Thus we need not distinguish between rod control and homogeneous control in the stability analysis or at amplitude calculations. Simulations have confirmed the statement.

Generally the stability is decreasing when core size, temperature coefficient or mean flux is increasing. For very high flux levels, however, the stability is increasing again. A slight asymmetry in the flux shape can also make stability better compared to a symmetric flux shape. Moreover, a flat flux has a lower critical height than a sinusodial flux (see also [5], chapter 4).

The flux shape may be described by the form factor Ψ, the symmetry or it can be characterized by a ditch form (flux B and F in fig. 1). Generally the critical height is bigger for a bigger form factor. In table 2 are compared the critical heights as function of the form factors for some common flux shapes.

Table 2: Critical height H(m) for different form factors Ψ, calculated with TRAXEN for 20 space points. ($\bar{\Phi}$ = 1; α = -0.0514)
The flux forms are shown in figure 1.
The critical height is the same for rod control and homogeneous control.

Ψ	Flux	H_{crit}(m)
1.0	A	5.36
1.14	B	5.15
1.29	C	7.25
1.35	D	7.3
1.35	E	7.5
1.45	F	5.05
1.47	G	7.7
1.57	H	8.89

The table also shows that the form factor is no unique measure of flux shape.

Fluxes B and F have a lower critical height than A, depending on the ditch shape along the axis. As the xenon oscillations are mainly affecting the first overtone, the loose coupling between the two halves of the core for a ditch flux shape causes a less stable flux. These results are compared to other models in [5], chapter 4.3 and the accuracy is satisfactory.

The dependence of the temperature coefficient α has been calculated. For a flat flux (fig 1:A) the following values of critical height H(m) have been computed ($\bar{\Phi}$ = 1, homogeneous control).

α	H_{crit}(m)
-0.0514	5.36
0	5.02

In [5] is calculated the quotient

$$K = \frac{\Delta H}{H_o^1} \cdot \frac{\alpha^1}{\Delta\alpha}$$

for a two point model, where we found:

K = 0.062.

Here we find:

K = 0.063.

In [5] was proved for a two point model that the critical height is lower for a symmetric than for an asymmetric flux shape. This is verified by computation with the TRAXEN model. In table 2 we see that flux D has a higher critical height than flux C (see fig. 1). The fuel distributions in the fluxes C and D are the same. Flux D is got from flux C only by moving absorbtion from one half to the other in order to get an asymmetric equilibrium flux.

We conclude, that the control rod movement does not affect the value of the critical height. On the other hand we see, that the rod insertion length in equilibrium may affect the stability. It will influence the flux distribution, which in turn influences stability.

3.4 AMPLITUDE OF THE TRANSIENTS

For small disturbances the type of the input signal is irrelevant for stability tests. We have therefore used steps as disturbances, as they are rather common in real reactors (see 2.3).

Besides stability boundaries there are bounds on the amplitude of transients in a power reactor. The flux deviation must not deviate more than some 5 or 10% from equilibrium. As the amplitude of the transient is related to stability, we use even the maximum amplitude of the transient as a stability measure. In order to be able to compare different reactors we choose a step in reactivity as a standard disturbance. The disturbance consists of a stepwise movement of 100 pcm from one half to the other half of the core.

We treat only stable reactors, which means that the maximum of a step response appears in the first overshoot.

3.4.1 SIMULATION RESULTS

Due to the nature of the oscillations the most serious point of the core is situated around the center of one of the two core halves, i.e. at the coordinates $z = 0.75$ H or $z = 0.25$ H. The simulations show, that for the standard flux shape of figure 1:C the most serious point is $z = 0.810$ H or $z = 0.190$ H.

After a step disturbance we get a transient in every space point like figure 2. We are now interested in the amplitude of the first overshoot.

Table 3 shows the result of a number of simulations. Flux shape is that of figure 1:C, the disturbance is 100 pcm and the control is homogeneous.

Table 3: Amplitude of first overshoot as function of core height for the flux C of figure 1.
$H_{crit} = 7.25$ m; $\alpha = -0.0514$; $\bar{\Phi} = 1$;
Control is homogeneous. Disturbance 100 pcm.

Core height (m)	Amplitude $\varphi(z = 0.81\ H)(\%)$
5.0	5.47
6.0	10.54
6.5	13.7
7.0	19.29

Figure 2 shows the biggest flux transients for two of the fluxes after a 100 pcm disturbance, the 6.0 and 6.5 m cores.

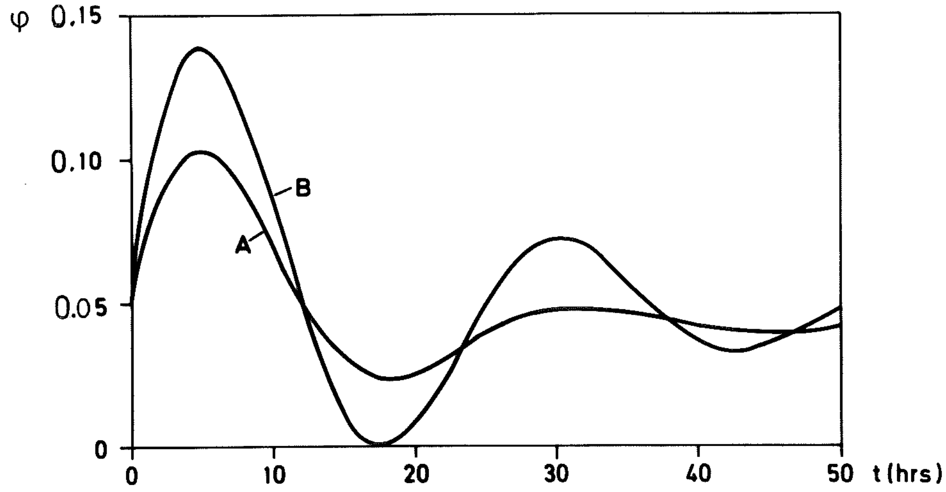


Fig. 2 - Neutron flux transients after a 100 pcm step disturbance in the most serious space point z = 0.81 H. The flux form is shown in figure 1:C

$\Phi = 1$

$\alpha = -0.0514$

Rod control

Core height: A) 6.0 m

B) 6.5 m

The relationship between core height and amplitude is presented in figure 3. We can see, that the amplitude is increasing very fast with core height.

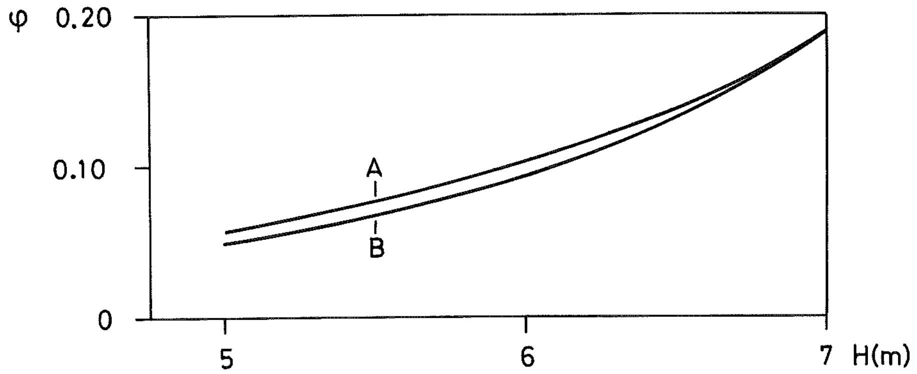


Fig. 3 - The maximum amplitude of the flux deviation after a 100 pcm step disturbance as function of core height. Comparison between simulations with TRAXEN (A) and analytical results with a two point model (B). The flux form is shown in figure 1:C

$\Phi = 1$

$\alpha = -0.0514$

Rod control (same result as homogeneous control)

3.4.2 COMPARISON WITH A SIMPLE TWO POINT MODEL

In [5] is presented a xenon model when the diffussion equation is given in only two core points, called the two point model. From this model is derived in appendix 4 a linear two point model of the form:

$$\frac{dx}{dt} = Ax + Bu$$

$$y = Cx + Du$$

where $y = \varphi_1$ = flux deviation
u = absorbtion input.

The transfer function G(s) is of second order. It is easy to derive the step response and its maximum values analytically.

For the case:

$\alpha = -0.0514$

$\bar{\Phi} = 1$

giving $H_{crit} = 6.93$ m

the values of maximum flux deviation are compared to the simulations in figure 3. The values are also shown in table 4.

Table 4: Amplitude of the first overshoot as function of core height for a two point reactor model.
$\bar{\Phi} = 1$; $\alpha = -0.0514$; $H_{crit} = 6.93$ m
Disturbance 100 pcm.

Core height (m)	Amplitude φ(%)
5.0	4.97
5.5	6.78
6.0	9.32
6.5	13.03
7.0	18.79

The comparison can only be qualitative as the flux shapes and critical heights are quite different, but the agreement between the simulations and the two point model is all the time within 10%. Near the stability limit the difference is only 2.6%.

4. LARGE DISTURBANCES. INFLUENCE OF NONLINEAR TERMS.

In this chapter is discussed the influence of different nonlinear terms. The most important ones are the absorbtion term (e.g. from the rod) in the buckling, the temperature coefficient and the quadratic term in the xenon equation.

If a rod is used for control, the influence may be very strong of this nonlinearity. In 4.1 is discussed qualitatively the influence from a very simplified model. It is shown, that this model can explain all the different types of rod movement, that have appeared in the simulations.

The next important nonlinear term is the temperature coefficient α. It has influence both on the linear stability and on the nonlinear character of the solutions, and a more negative α has a strong stabilizing effect on the oscillations.

In 4.2 is discussed appearance of periodic solutions, limit cycles, both with rod control and with homogeneous control. Soft self excitation and hard self excitation have been shown. The periodic solutions were discovered with a simple two point model, [5], chapter 5, and have been verified by simulations here.

In 4.3 is discussed influence of nonlinear terms on the amplitude of the transients. As a criterion of nonlinear influence we calculate the accuracy of the superposition principle in different cases.

4.1 INFLUENCE OF ROD MOVEMENT ON STABILITY

As the absorbtion c(z,t) is completely determined by one parameter, one of the constants in (1:11) or (1:12) can be arbitrarily chosen, e.g. the "thickness" of the rod. The rod insertion length and the movement are related to the thickness, and, as we will show, the stability in turn is dependent of rod movement.

Now, we will discuss, what causes the absorbtion to increase or decrease, or alternatively what causes the rod to move up and down. From some very simple examples we will demonstrate the rod movements for different phases of the oscillations.

4.1.1 GENERAL DISCUSSION

Case I

Regard a flux during an oscillation. We will study the special condition when the oscillation passes a symmetric shape as in fig. 1. For simplicity we assume the buckling to be space independent in the two halves of the core. The buckling is called B_1^2 and B_2^2.

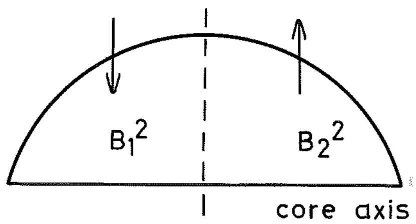


Figure 1: Symmetric flux distribution.

Assume, that the flux deviation will be as in figure 1. The buckling B_1^2 decreases and the B_2^2 increases.

In appendix 5 is shown, that, in order to satisfy all the boundary conditions, we must have:

$$\left|\Delta B_1^2\right| > \left|\Delta B_2^2\right| \tag{1}$$

where ΔB^2 is the total finite change in buckling.

Now the change of buckling is caused by four different terms (2:7). We assume, that the mean value of xenon and flux deviations are equal but of opposite sign in the two core halves, as the total power is constant. Thus the contribution to ΔB^2 from the xenon and temperature coupling are approximately equal. Further we assume the disturbance u is of the same amplitude but opposite sign in the two halves.

In order to satisfy (1) we must add absorbtion to the core, which can be done in a couple of ways:

- in the case homogeneous control the same absorbtion is added to both halves which directly makes (1) to be satisfied,
- in the case rod control we must either decrease B_1^2 by inserting the rod in the left part, or we can decrease B_2^2 by inserting it into the right part, or a combination of these movements.

An analogous discussion is valid for the opposite flux movement to that in fig. 1.

To sum up, absorbtion must always be added when the flux deviates from its symmetric shape. The deviation may be caused by disturbance of an equilibrium flux or a free oscillation or both these changes.

Case II

We regard the flux during an oscillation or in equilibrium when it is asymmetric and we assume

$B_1^2 < B_2^2$

where B_i^2 are assumed to be space independent bucklings.

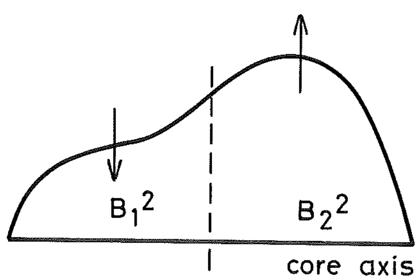


Figure 2: Asymmetric flux

If the deviation has a direction as in fig. 2, we get deviations of the buckling called ΔB_1^2 and ΔB_2^2. In appendix 5 is proved that for the simple case, that B_1^2 and B_2^2 are space independent

$$\left|\Delta B_1^2\right| > \left|\Delta B_2^2\right|$$

If we further assume as in case I the xenon deviations to be almost equal in the two parts we must even here add absorbtion to the core, e.g. insert a rod.

Case III

For the flux movement in figure 3, where $B_1^2 < B_2^2$ we can distinguish between two cases. If the deviation of the flux is so small, that it does not reach the symmetric shape we have (see appendix 5)

$$\left|\Delta B_1^2\right| > \left|\Delta B_2^2\right|$$

where ΔB_i^2 are the changes of the bucklings and absorbtion must be subtracted, or the rod is moved out.

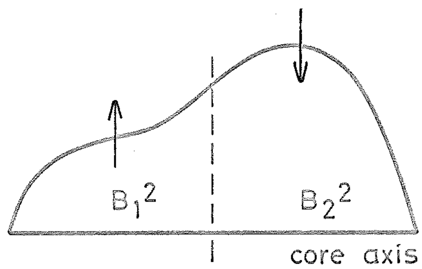


Figure 3: Asymmetric flux

If the deviation is so big, that the flux shape passes the symmetric shape (see fig. 4) the movement can be divided into two parts. For the first part, up to the symmetric shape, fig. 4, we have to subtract absorbtion (to draw out the rod). For the second part of the movement, we have exactly case I and absorbtion must be added again (the rod is inserted again).

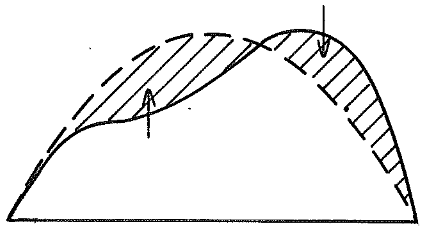


Figure 4: Deviation of flux from asymmetric to symmetric shape.

Simulations have shown that this reasoning is valid for a great number of different flux forms. A contradiction has not been shown for neither rod control nor homogeneous control during a single simulation.

4.1.2 SIMULATION RESULTS

In a couple of numerical examples is demonstrated the variation of the absorbtion. In the first example we have homogeneous control, and in the second and third examples we use rod control. For the two latter cases is demonstrated the effect of the rod on stability, depending on direction of the disturbance.

Example 1: H o m o g e n e o u s   c o n t r o l

Figure 5 shows a flat flux oscillation (fig. 3:1:A) during 30 hours. Core height is 5.40 m, 0.04 m above the critical height. The disturbance is a 1000 pcm pulse, moved from "left" to "right" in the flux during 2 hours. After this time the oscillation is free. The pulse causes the flux to deviate directly from equilibrium and absorbtion is added ($\Delta c < 0$). We observe the flux deviation after the pulse has finished at t=2 (B). When this flux form is oscillating to the symmetric shape (C), absorbtion is taken away (Δc is growing in the upper curve). In the next phase, from C to D, we add absorbtion at the same time as the almost symmetric flux shape C oscillates to D, and then a similar sequence takes place in the continuation of the transient.

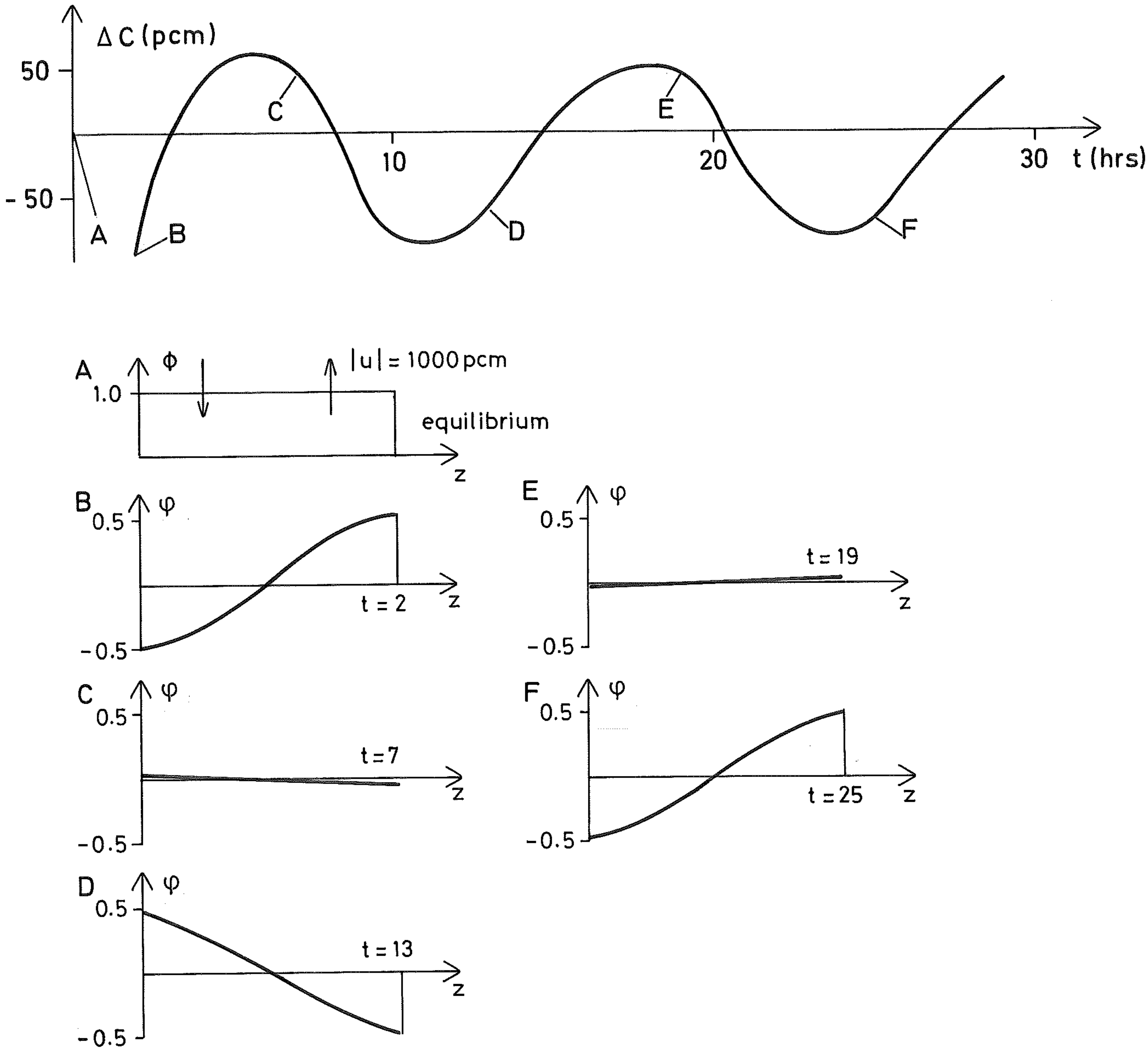


Fig. 5 - Variation of control term C during an oscillation for a flat flux with homogeneous control

H = 5.40 m   Φ = 1.0

α = -0.0514

Disturbance 1000 pcm during 2 hours with direction shown in A. The curves B-F show the spatial distribution of flux deviation. Coincident points of time (B-F) are marked.

Thus we observe that the absorbtion must oscillate with double frequency compared to the flux or xenon oscillations.

Example 2 - 3: R o d   c o n t r o l

A similar oscillation of the absorbtion as in example 1 is observed in the rod control case. During an oscillation the rod has to move to and fro in order to maintain constant power and this will have a strong influence on stability and on the amplitude of the transients. We will study two different disturbances on a flat flux, the same flux as in example 1.

Figures 6 and 7 show the result of two different signs of a disturbance on the same flux and rod configuration. The total absorbtion varies in the same way but with different amplitudes and is also similar to that of homogeneous control, fig. 5. It can be explained from the general discussion of the preceeding section exactly as example 1.

Thus the rod movement must have different influence on the transients in fig. 6 and 7. Compare the curves B in fig. 6 and 7. In the former case the amplitude of the flux deviation is smaller than in the latter case depending on the rod. In both cases the rod has moved to right, into the right core half, and has caused a damping of the first flux (fig. 6) and an amplification of the second flux (fig. 7).

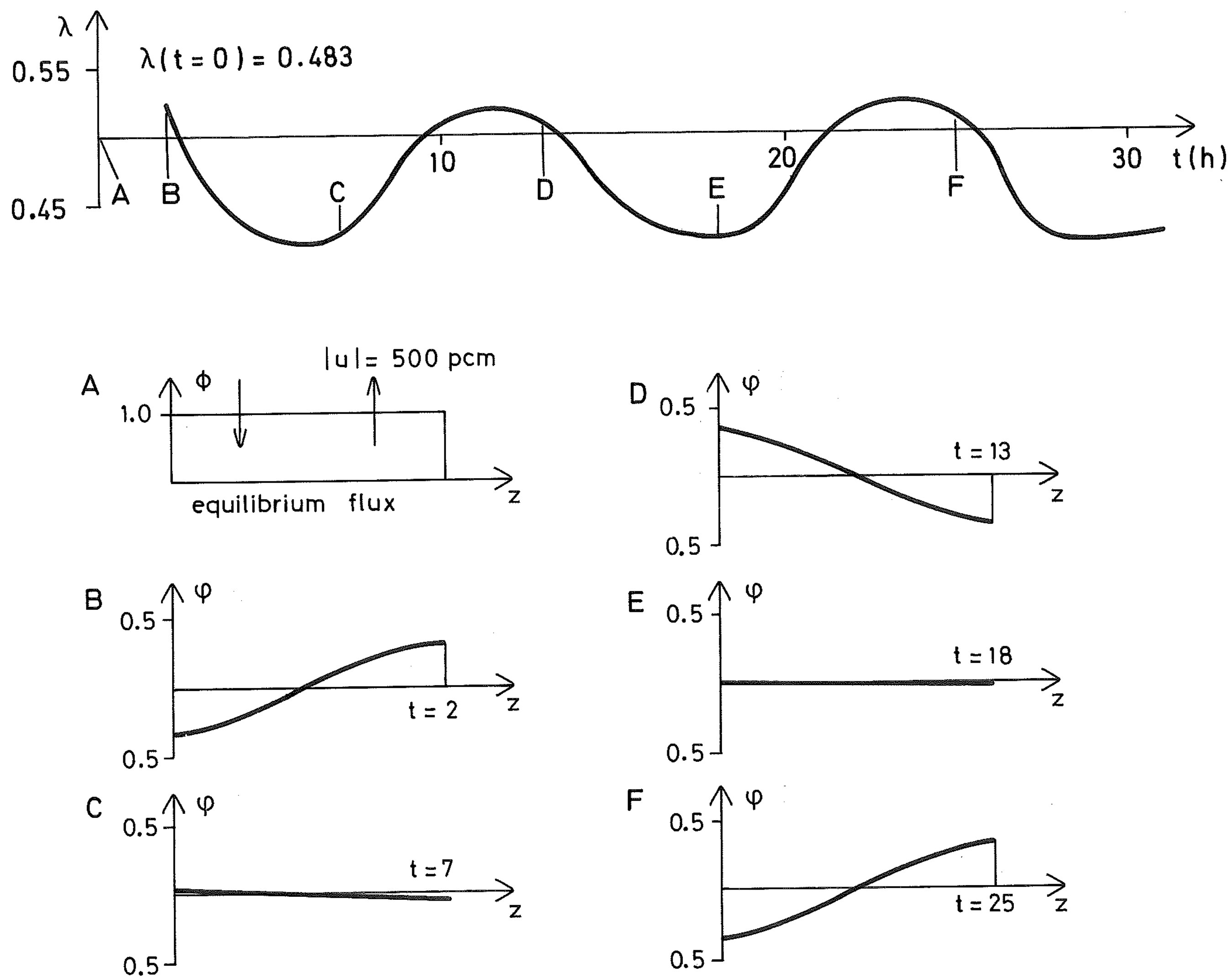
We can see from fig. 6 and 7 that every second time the rod moves in, it causes amplification and every second time it causes damping (curves B, D, F).

The transient in figure 7, however, are all the time bigger than those of figure 6. This depends on what has happened during the first two hours, when the disturbance was acting. In both cases it was inserted, but in figure 7 it caused already here an amplification. The flux is more sensitive to the disturbance of figure 7 than that of figure 6 for this rod configuration.

We realize immediately, that if the rod from the beginning is inserted only in a small part of the core, we get the same type of variation as in figure 6 and 7, but the mean value of the insertion is now $\lambda_0 < 0.5$. Thus the disturbance like figure 6 can be amplified instead of damped by the rod as the absorbtion increases in the left part only, where the flux decreases. The opposite effect is got for the opposite disturbance. If the rod has smaller absorbtion it must be moved a longer way in order to maintain criticality during an oscillation. Then we intuitively realize that the maximum influence of the rod is got for an insertion variation over a whole core half.

We have from this discussion a good explanation to the strong asymmetry of the two point model oscillations in ref. [5] chapter 5.1.2. For one half period the transient is amplified (fig. 5:3, 4 in [5]) and for the other half it is strongly damped by the "rod". In the two point model we have all the time a maximum influence of the "rod" as it all the time acts in the whole "core half", namely in one of the two space points.

In 4.2 and 4.3 are shown some more simulations where the rod has influenced the trajectories considerably.



<u>Fig. 6</u> - Variation of rod insertion λ during a xenon oscillation for a flat flux. Rod absorbtion = 500 pcm maximum.

H = 5.40 m   Φ = 1.0

α = -0.0514

Disturbance 500 pcm during 2 hours with a direction as in A. The curves B-F show the spatial distribution of flux deviation. Coincident points of time (B-F) are marked.

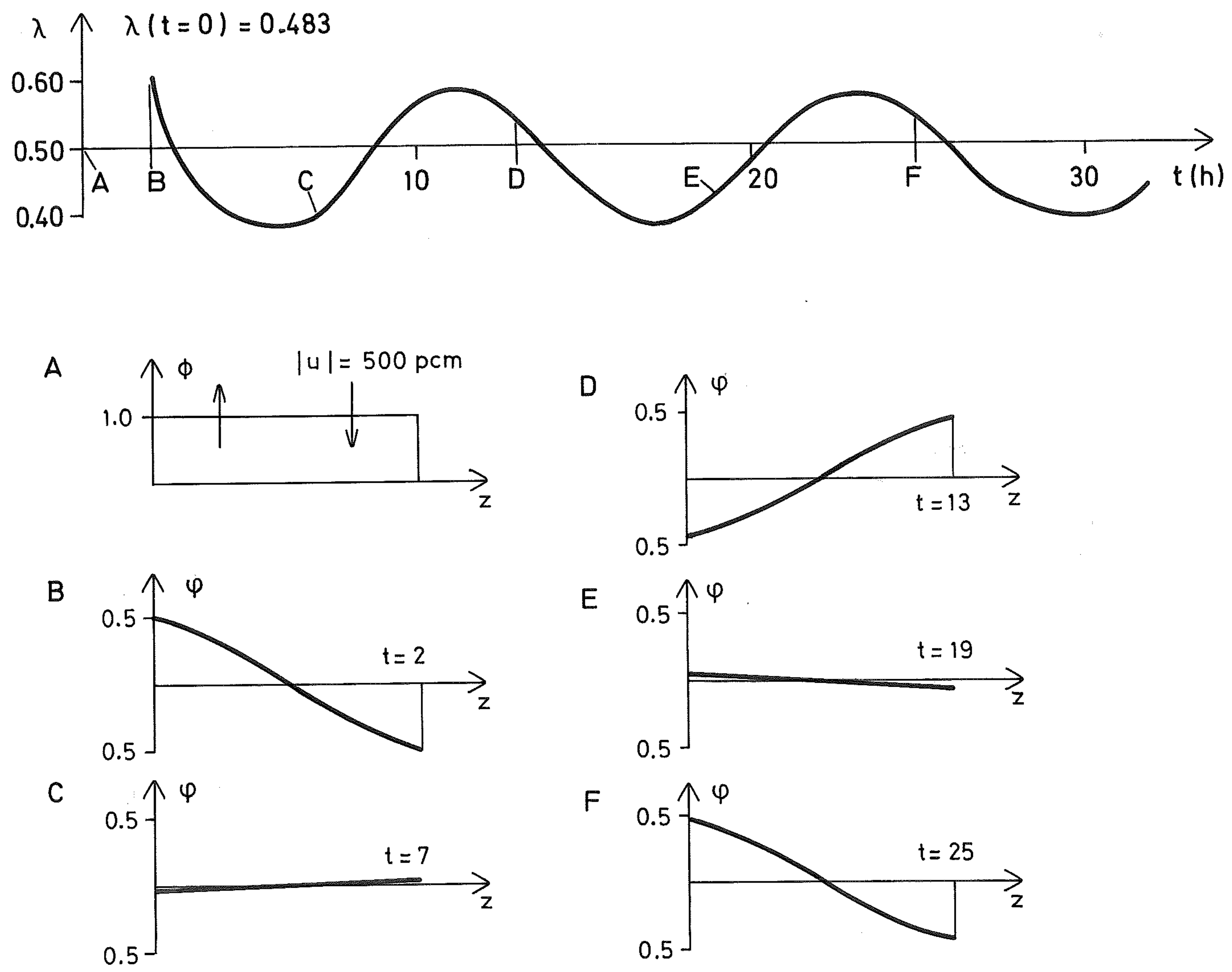


Fig. 7 - Variation of rod insertion λ during a xenon oscillation for a flat flux. Rod absorbtion = 500 pcm maximum.

H = 5.40 m   Φ = 1.0

α = -0.0514

Disturbance 500 pcm during 2 hours with a direction as in A.

The curves B-F show the spatial distribution of flux deviation.

Coincident points of time (B-F) are marked.

4.2 STABILITY. PERIODIC SOLUTIONS

4.2.1 CRITERION OF STABILITY IN CASE OF LARGE DISTURBANCES

In 3.2 we defined stability in case of small disturbances. When studying the nonlinear trajectories we must care about every single trajectory.

Let us define the amplitude of a transient to be $||x||$, where $||x|| = 0$ in equilibrium.

We call the trajectory stable if:

$$\lim_{t \to \infty} ||x|| < M$$

where M is a finite number.

If

$$\lim_{t \to \infty} ||x|| = 0$$

the trajectory is asymptoticly stable.

4.2.2 ROD CONTROL

In chapter 5.1 in [5] is reported the appearance of periodic solutions for a nonlinear two point model. It is shown that the temperature coefficient α has a significant influence on the character of the solutions.

With the complex model it is even possible to show a strong influence of the flux shape. In chapter 3.3 was found that a smaller form factor gave a less stable flux or a lower critical height.

Here is shown two cases which confirm what is shown in chapter 5.1 in [5] about periodic solutions. It is possible to get stable periodic solutions for both $\alpha = 0$ and $\alpha < 0$ with a core height above the critical height. The small disturbance solutions diverge to the limit cycle, while big disturbance trajectories converge to the same limit cycle.

In [5] was also shown the appearance of unstable limit cycles. It has not been possible to confirm this result for flat fluxes, but for a ditch flux we have got unstable limit cycles.

Example 1: S t a b l e   p e r i o d i c   s o l u t i o n s

Figure 8 shows trajectories for a flat flux reactor, bigger than critical height. The solutions converge to a stable periodic solution. The curves B and C show the same oscillations as in fig. 6 and 7. The upper curve, A, shows a small disturbance trajectory of a single space point. The transient in A diverges, while B and C, which are caused by big disturbances converge to a stable periodic solution, whose period time is about 24.1 hours. The trajectories are rather symmetric around the time-axis, because the rod movement is not very big (see fig. 6 and 7). Compare [5] chapter 5.1.

Fig. 8 - Trajectories of flux and xenon deviations in one space point z = 0.048 H of a flat flux (fig. 3:1:A) reactor with rod control (the same as in figures 6 and 7)

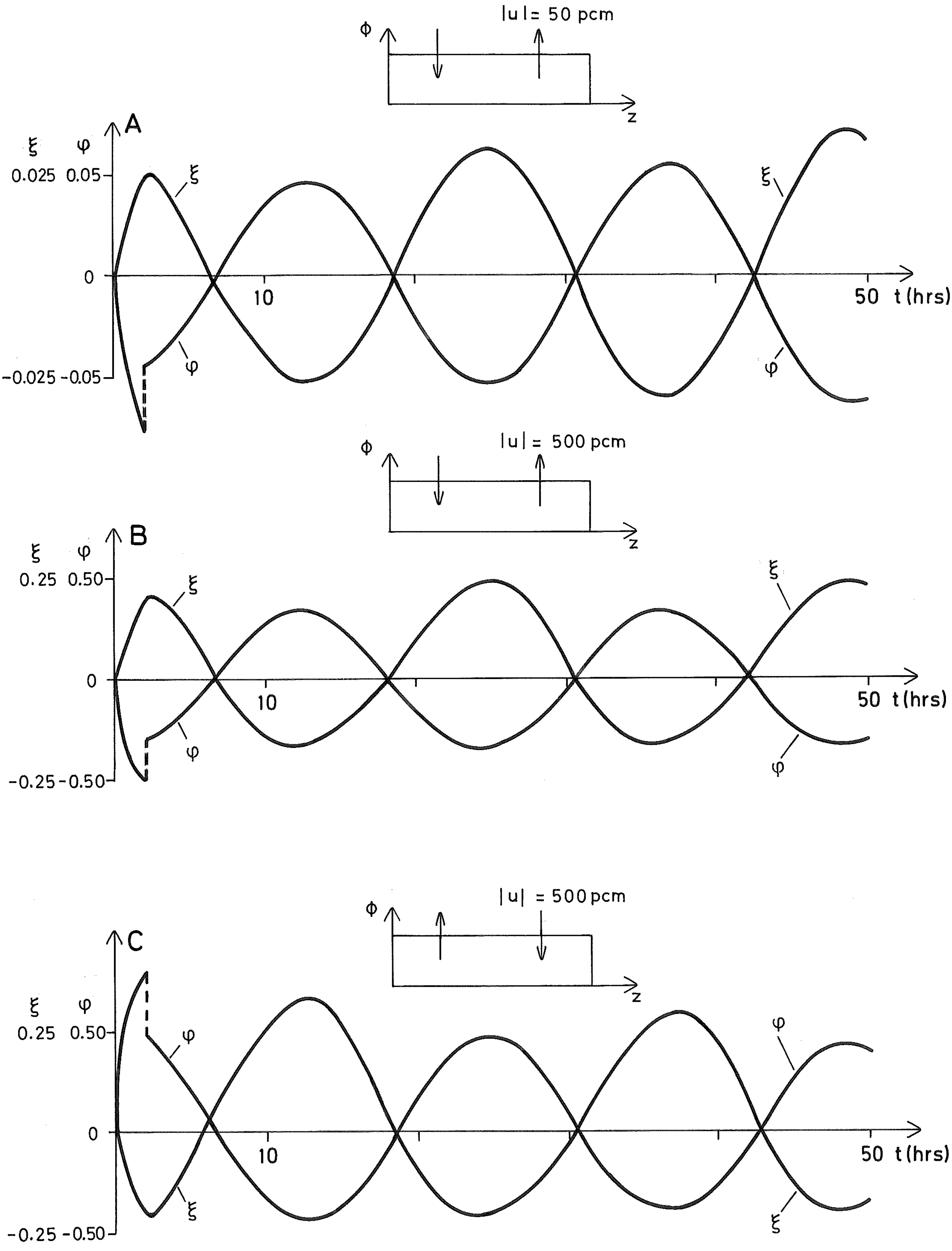
$H = 5.40$ m   ($H_{crit} = 5.36$ m)

$\alpha = -0.0514$

$\Phi = 1.0$

The direction of the disturbance is shown in the small figures. The amplitude is

A.	50 pcm	Divergent trajectories
B.	500 "	Convergent "
C.	500 "	" "


|u| = 50 pcm
Φ
z
A
ξ
φ
0.025 0.05
0
-0.025 -0.05
10
50 t(hrs)
|u| = 500 pcm
B
0.25 0.50
-0.25 -0.50
C

Example 2: U n s t a b l e   p e r i o d i c   s o l u t i o n s

For a ditch flux, fig. 9 (the same as fig. 3:1:F), was found an unstable limit cycle. The core height was 5.0 m, and as the critical height is found to be 5.05 m (section 3.3) it is stable for small disturbances. The flux was disturbed by 100 pcm reactivity moved from one side to the other, as in figure 9. Unstable trajectories were found.

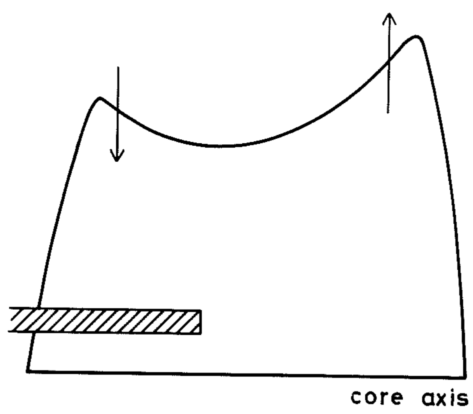


Figure 9: Ditch flux distribution.
H = 5.0 m
The rod at equilibrium is shown.

As mentioned earlier it has been rather difficult to verify unstable periodic solutions. This depends very much on rod configuration. As mentioned before, the "rod" can cause much bigger amplifications in the two point case. Perhaps we have not found the most serious rod configuration in the simulations.

4.2.3 HOMOGENEOUS CONTROL

With homogeneous control the nonlinear terms are still damping large trajectories very powerful, why periodic solutions appear even here (compare [5], chapter 5.1). The periodic solutions exist for both positive and negative values of α. As a negative α has a damping effect on the trajectories, the amplitude of the periodic solutions are smaller for negative than for positive α.

We will regard a case when $\alpha = 0$. The core height is $H = 5.02$ m, 1 cm over the critical height. Thus the trajectories for small disturbances are unstable, but those for big disturbances are stable, as in figure 10. The period of the periodic solution is about 24.5 hours.

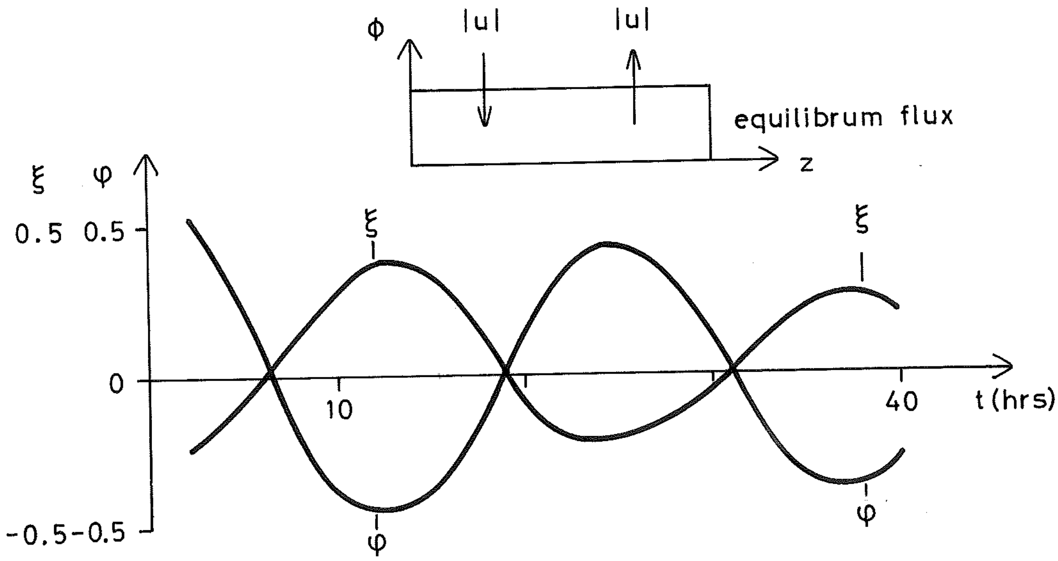


Fig. 10 - Trajectories of flux and xenon deviations in one space point, z = 0.952 H, of a flat flux (fig 3:1:A) reactor with homogeneous control.

$H = 5.02$ m $\quad (H_{crit} = 5.01$ m)

$\alpha = 0$

$\Phi = 1.0$

The direction of the disturbance is shown in the small figure. The transient is convergent althought the reactor is above the critical height.

4.3 AMPLITUDE OF THE TRANSIENTS

4.3.1 INTRODUCTION

As mentioned before, not only the critical height but also the amplitude of the transients are very important of technological reasons. After a disturbance of the reactivity we want to know when the first maximum positive deviation of the flux appears and how big it is. In fig. 8 we can see that it appears at the end of the pulse disturbance at t = 2. Depending on the direction of the disturbance, the most serious point will be in the upper or lower core half.

In 2.3 we mentioned that step disturbances may also be relevant in order to study the amplitudes.

In next section we will study the influence of the control rod on transient amplitude. In 4.3.3 we use homogeneous control. In this case the nonlinear character is mainly determined by temperature coefficient and xenon feedback in reactivity.

4.3.2 ROD CONTROL INFLUENCE ON AMPLITUDE

In 4.2 is already studied two cases, figures 6 and 7, where the rod has a big influence on the first maximum flux deviation.

<u>Example</u>

A further example is studied below. A reactor with 7.0 m core height (figure 11) was disturbed by a step of 500 pcm reactivity. The critical height is 7.3 m. Figure 12 shows the trajectories at the points z = 0.81 H and z = 0.19 H respectively, where the flux deviation is largest. In the figure is compared three different cases. Curves A and C are for rod control, where the rod is inserted to about the core center at equilibrium. The direction of the disturbance is opposite in the two cases, curve C as marked in figure 11. Curve B is the result with homogeneous control and is shown for comparison. We can see that the rod has a considerable influence on the amplitude.

The behaviour can be explained as in previous sections.

As shown in 4.2 we have to add absorbtion during the first part of the oscillation. With a rod we must insert it for both directions of disturbance. Therefore it will damp the movement, caused by a disturbance as in figure 11 (curve C in fig. 12) and will amplify the opposite step answer (curve A). With homogeneous control we have a medium amplitude (curve B).

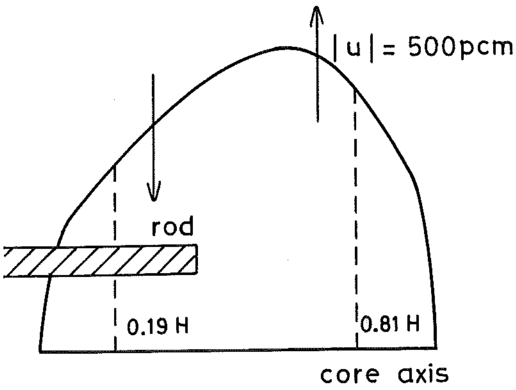


Figure 11: Flux distribution.
H = 7.0 m, Ψ = 1.33

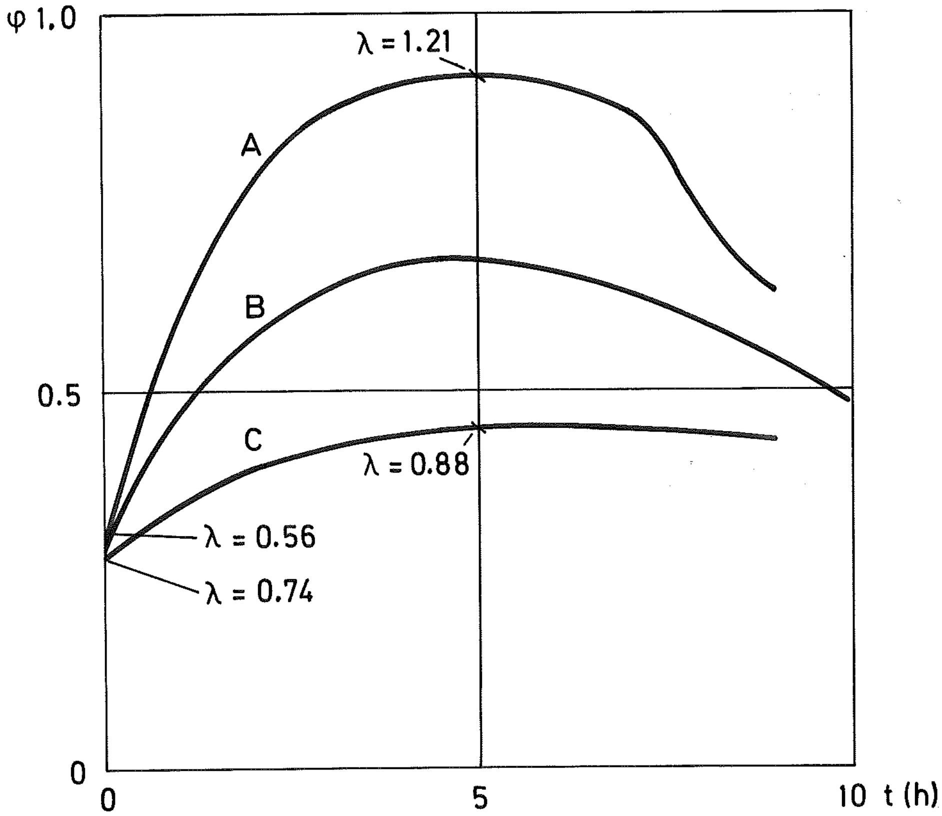


Fig. 12 - Trajectories for maximum flux deviation in one space point of the flux shown in figure 11.

$H = 7.0$ m ($H_{crit} = 7.3$ m)

$\alpha = -0.0514$

$\Phi = 1.0$

The disturbance is a 500 pcm step. The direction is as in fig.11 for B, C and opposite for A.

Rod insertion length λ is shown for its maximum deviation.

	Control	Variable	λ(t=0)	λ(t=0+)
A	rod	φ(z=0.19H)	0.54	0.56
B	homogeneous	φ(z=0.81H)	-	-
C	rod	φ(z=0.81H)	0.54	0.74

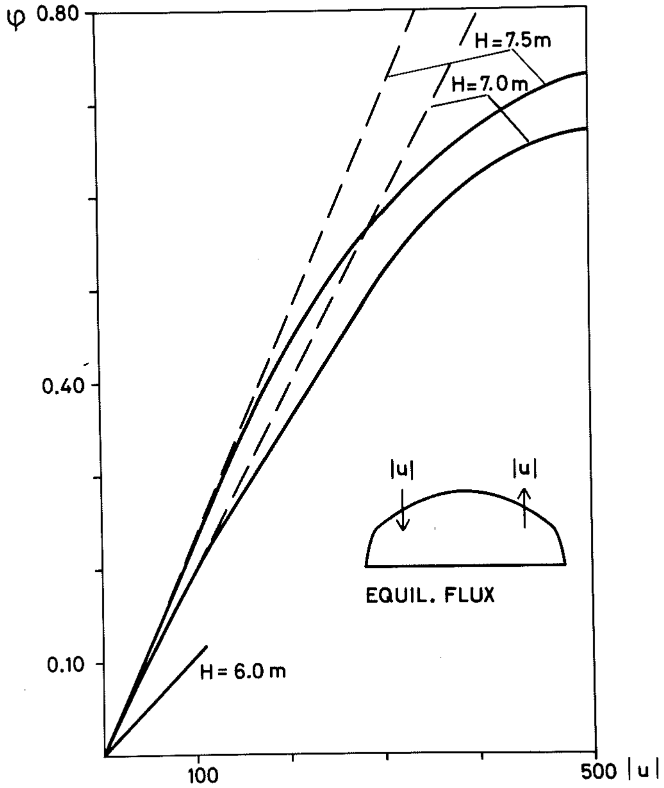


Fig. 13 - Maximum flux deviation at one space point z = 0.81 H. The reactivity disturbance is a step of u pcm with a direction as in the figure.

Core height 6,7,7.5 m

$\Phi = 7.0$

$\alpha = -0.0514$

Homogeneous control

A similar result as that in figure 12 is got for a big number of cases. Especially the fluxes D, F, G of fig. 3:1 have been disturbed by a 500 pcm step disturbance of both directions. For all fluxes absorbtion had to be added during the first 6 - 8 hours of the transient. The rod caused an amplification for one direction and damping for the other.

As mentioned in 3.4 the sensitivity of the fluxes for disturbances increases with core height. Thus a 500 pcm disturbance on an 8 m reactor, 0.65 m over critical height in case D, fig. 3:1, causes a much bigger rod movement than in a 7 m core.

If the rod moves mainly in "left" part during the transient it causes the disturbance shown in figure 11 to be amplified, while it is damped if the rod moves mainly in "right" part. If the rod moves through the whole core it is impossible without simulation to predict the result.

The opposite effect of rod is got for opposite direction of the disturbance.

4.3.3 INFLUENCE ON AMPLITUDE OF OTHER NONLINEAR TERMS.

AMPLITUDE OF DISTURBANCE.

In a linear system the superposition principle is valid. For big disturbances in the xenon process, the nonlinear terms, besides the rod, has a damping effect on the amplitude of the trajectories.

Figure 13 shows the sensitivity to different step disturbances and core heights for a flux with homogeneous control. The maximum flux deviation for the most sensitive point is registrated, and appears after 5 - 6 hours. For disturbances above 100 pcm the linearity is bad (compare 3.1).

FLUX SHAPE

An asymmetric flux has different sensitivity depending on the sign of disturbances.

Example (Homogeneous control)

The flux in fig. 14 (H = 8.0 m, 0.3 m over critical height) was disturbed by a 500 pcm reactivity step as in figure 14 with maximum flux deviation:

$\Delta\Phi(z = 0.762\ H) = 0.689$ at $t \approx 6$ hours,

while the opposite direction of the disturbance resulted in:

$\Delta\Phi(z = 0.238\ H) = 0.938$ at $t \approx 6$ hours.

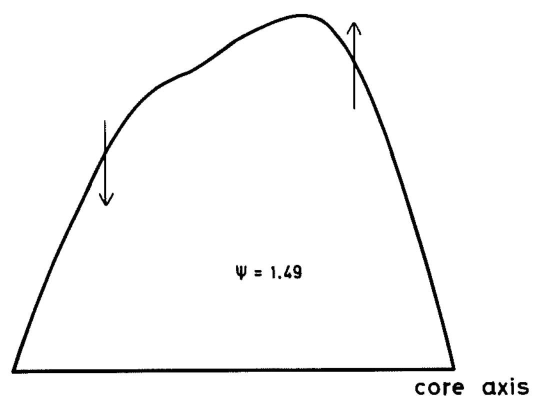


Figure 14 Flux distribution.
H = 8.0 m

5. A SPACE INDEPENDENT NONLINEAR MODEL

Simulations have shown, that a rather simple, almost linear relationship holds between neutron flux and xenon deviations in every space point. This condition is made use of to get a simpler model. The diffussion equation is replaced by a simple linear condition, which causes the xenon process to be described by a second order nonlinear ordinary differential equation.

This model is compared in 5.2 to a two-point model and is analysed in 5.3. It is shown that the space independent model is only valid for small disturbances.

The result is compared to other space independent models in 5.4.

5.1 RELATIONSHIP BETWEEN FLUX AND XENON DEVIATIONS

In the simulations of small disturbances we have observed, that flux and xenon concentration in one core point varies approximately in opposite direction during an oscillation. Margolis [3] showed for a two region core that the transfer function

$$G(s) = \frac{\delta\Phi(s)}{\delta X(s)}$$

has a phase which is approximately $-\Pi$, why G(s) is approximately real and negative.

In figure 1 is shown the flux and xenon deviations for a flat flux in the point $z = 0.05$ H. The variables oscillates to and fro along the line during an oscillation. The oscillation is caused by a small disturbance, 50 pcm, which has been moved between the core halves during 2 hours. In other space points we have similar lines with the same slope but other length (amplitude of the oscillations).

From figure 1 we state the relationship:

$$\varphi = -b \cdot \xi \qquad (1)$$

where $\varphi = \Phi - \Phi^{o}$, $\xi = X - X^{o}$ and b = constant.

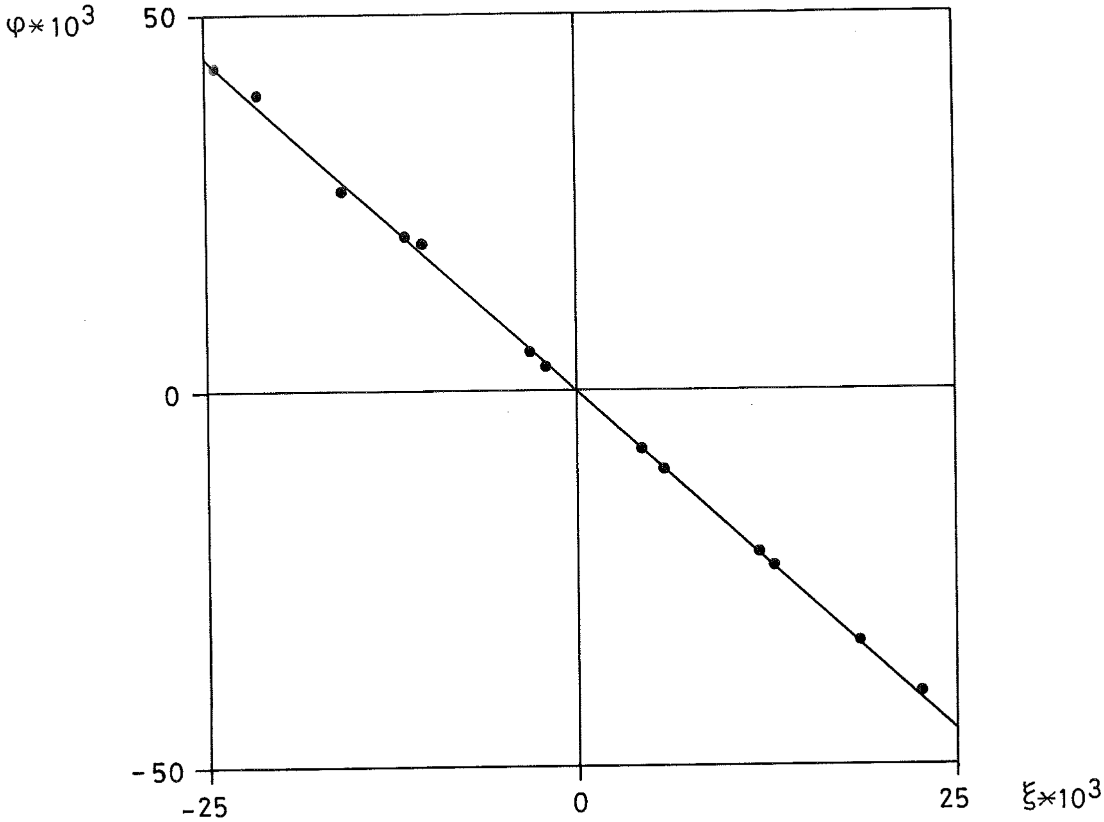


Fig. 1 - Flux and xenon deviation at z = 0.05 H for a flat flux (fig 3:1A)
H = 5.40 (4 cm above crit. height)
Homogeneous control. Disturbance: 50 pcm in 2 hours

$\varphi = -1.78\ \xi$

The same constant b is valid for every space point.

Now the same good linear relationship is found in several cases at small disturbances (< 100 pcm). A number of fluxes just around the critical heights have been observed for small disturbances, and the linear approximation (1) is valid with good accuracy. In table 1 is summed up the constants b (eq. (1)) for some cases.

Table 1 Linear relationship between flux and xenon $\varphi = -b\xi$ The fluxes are shown in figure 3:1.

Case	H(m)	H_{crit}	Flux	α	b
1	4.95	5.01	A	0	1.73
2	5.02	5.01	A	0	1.76
3	5.40	5.36	A	-0.0514	1.78 x)
4	8.90	8.89	H	-0.0514	1.79
5	7.0	7.35	D	-0.0514	1.71
6	7.5	7.35	D	-0.0514	1.73

x) (in fig. 1)

We define:

$$\xi = X - X^o$$
$$\eta = I - I^o$$
$$\varphi = \Phi - \Phi^o$$

and write the xenon and iodine equations in incremental form

from (2:8), (2:9) and neglect the subscript for the space point. Thus the following equations are valid:

$$\frac{d\xi}{dt} = -\lambda_x \xi + \lambda_i \eta + \gamma_x \sigma_x \varphi - \sigma_x X^o \varphi - \sigma_x \Phi^o \xi - \sigma_x \xi \varphi \qquad (2)$$

$$\frac{d\eta}{dt} = \gamma_i \sigma_x \varphi - \lambda_i \eta \qquad (3)$$

We insert (1) in (2) and (3) and get the dynamic system:

$$\frac{dx}{dt} = \begin{bmatrix} -\lambda_x - \sigma_x \Phi^o + b \cdot \sigma_x (X^o - \gamma_x) & \lambda_i \\ -\gamma_i \sigma_x b & -\lambda_i \end{bmatrix} x + \begin{bmatrix} b\sigma_x x_1^2 \\ 0 \end{bmatrix}$$

or

$$\frac{dx}{dt} = Ax + \begin{bmatrix} b\sigma_x x_1^2 \\ 0 \end{bmatrix} \qquad (4)$$

where

$$x = \begin{bmatrix} \xi \\ \eta \end{bmatrix}$$

5.2 COMPARISON WITH A TWO-POINT MODEL

In [5], chapter 2, is derived a model of a two point reactor. In the linear approximation of the symmetric flux we found (section 2.7 in [5]) the relationship

$\varphi_1 = b_1 \cdot (x_1 - x_3)$ for rod control (2:48) and

$\varphi_1 = b_2 \cdot (x_1 - \frac{1}{2} x_3)$ for homogeneous control (2:50)

where $x_1 = \xi_1$, $x_3 = \xi_1 + \xi_2$ and b_1, b_2 are core constants. Now, if we have a symmetric disturbance of the flux, the initial conditions are:

$$x_3 = \xi_1 + \xi_2 = 0$$

$$x_4 = \eta_1 + \eta_2 = 0$$

and (2:49) and (2:52) in [5] show that x_3 and x_4 are identically zero all the time. In this case we get:

$$\varphi_1 = b_1 x_1 = b_2 x_1$$

for both homogeneous and rod control, where

$$b_1 = \frac{\beta \, \Phi^o \, h^2}{2 - h^2 \, \alpha \, \Phi^o} = \frac{\beta \, \Phi^o \, H^2}{18 - H^2 \, \alpha \, \Phi^o} = b_2 \qquad (5)$$

For the critical height for $\Phi^o = 1$, $\alpha = -0.0514$
we find
$H^o = 6.93$ m
$b_1 = -1.71$ or

$$\varphi = -1.71 \, \xi \qquad (6)$$

in the two point symmetric model.

For $\alpha = 0$ we have the critical height

$H^o = 6.498$ m

for the two point model [5] which gives

$b_1 = -1.71$

Compare b_1 to the values of b in table 1 where all the fluxes are situated around the different critical heights. We can also see that b is very insensitive to variations in α.

Further in the two point model, the dynamic behaviour of the process for the symmetric initial conditions above is determined from a 2 x 2 submatrix.

$$A = \begin{bmatrix} -\lambda_x - \sigma_x \Phi^o - b_1\sigma_x(X^o - \gamma_x) & \lambda_i \\ \gamma_i \sigma_x b_1 & -\lambda_i \end{bmatrix} \quad (7)$$

(see (2:53, 54, 56) in [5])

We see directly, that the linear part of (4) is identical to (7). The matrix A (7) is got as an approximation of the symmetric two point model when the control term is neglected [5].

5.3 ANALYSIS OF THE SPACE INDEPENDENT MODEL

First we find the singular points of (4), which are (0,0) and

$$X_1^o = \frac{1}{\sigma_x b} \{\lambda_x + \sigma_x \Phi^o + b\, \sigma_x(1 - X^o)\}, \text{ as } \gamma_x + \gamma_i = 1$$

$$X_2^o = -\frac{\gamma_i}{\lambda_i} \{\lambda_x + \sigma_x \Phi^o + b\, \sigma_x(1 - X^o)\} \qquad (8)$$

Now we choose b = 1.7, which causes (0,0) to be a stable focus and the other singular point to be a saddle point. (The choice of b is not critical. We get one positive real eigenvalue for all values of b in this singular point.)

Figure 2 shows the phase plane of (4). The saddle point (8) is marked in the figure. The trajectories for large deviations from origo show some odd details.

Trajectories near origo are stable and oscillatory. For bigger disturbances, however, the trajectories will start outside an unstable limit cycle. At first they are diverging in a spiral, but later they are nonoscillatory unstable. For very big disturbances it is possible to get unstable solutions without any oscillation.

These nonperiodic unstable solutions have not been verified by any other space dependent model.

The character of the phase plane is not influenced by the value of the temperature coefficient (see table 1) when α varies between 0 and -0.05. This contradicts the results from the two point model in [5], chapter 5, as well as the TRAXEN simulations.

To sum up, the space independent model (4) is not sufficient to describe what happens for large amplitudes of the state variables. Then we must include higher order terms in the relationship between flux and xenon.

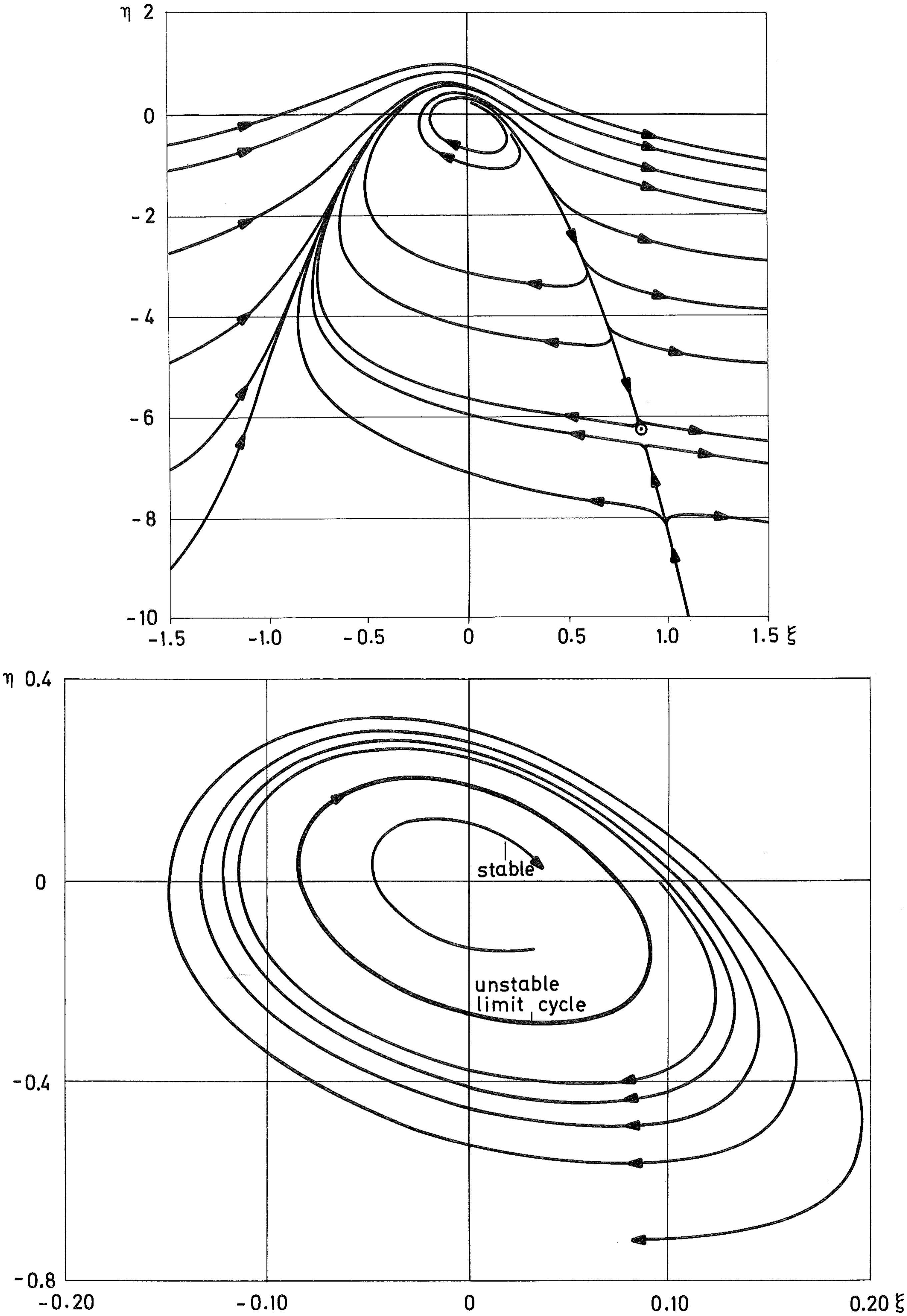


Fig. 2 - The phase plane of the nonlinear space independent xenon model. The lower figure is a detail of the upper one.

5.4 COMPARISON WITH OTHER SPACE INDEPENDENT MODELS

A state independent model of the same structure as presented is proposed by Sha [8]. He assumes the relationship (1) and gets some stability criteria out of Lyapunov theory. However, it is not presented any trajectories.

Chernick et. al [1] have a similar model. Nonoscillatory unstable solutions are got for positive temperature coefficients and this also contradicts the result from the nonlinear two space point model, [5], ch.5, as well as TRAXEN simulations, ch. 4.2.

ACKNOWLEDGEMENT

The program TRAXEN was first written by the author at Swedish State Power Board, Stockholm. The author wishes to express his gratitude to professor K.J. Åström for his suggestions and criticism. Mrs. G. Christensen and Miss L. Jönsson typed the manuscript and Mrs. B. Tell drew the figures.

REFERENCES

[1] J. Chernick, G. Lellouche, and W. Wollman: The Effect of Temperature on Xenon Instability, Nucl. Sci. Eng., 10,120-31(1961)

[2] S. Kaplan: The Property of Finality and the Analysis of Problems in Reactor Space-Time Kinetics by Various Modal Expansions, Nucl. Sci. Eng. , 9, 357 (1961)

[3] S.G. Margolis: A Nyqvist Criterion for Spatial Xenon Stability, Trans. Am. Nucl. Soc. 3(2):437 (Dec., 1960)

[4] Marviken Power Station: Design Status Report, AB Atomenergi, S-353, 1966, Sweden (not for publication)

[5] G. Olsson: Spatial Xenon Instability in Thermal Reactors, Report 6910, Div. of Aut. Contr., Inst. of Techn., Lund, 1969

[6] G. Olsson: Digitalt program TRAXEN för transientberäkningar av xenonsvängningar i en axiell reaktormodell, Swedish State Power Board, Stockholm, 1966, Report E-53/66

[7] G. Olsson: Digital Simulation of Axial Xenon Instability in Reactors, Swedish State Power Board, Stockholm, Part I: Report V-116/65, 1965, Part II: Report E-8/66, 1966

[8] W.T. Sha: Stability in the Large of Xenon Oscillations, Trans. Am. Nucl. Soc., 10, 572 (1967)

Appendix 1

DEFINITION OF SYMBOLS AND THEIR NUMERICAL VALUES

Symbol	First def. in equation	Explanation	Numerical value
$B^2(z,t)$	2:2	Material buckling	
$(B^2)^*(z)$	2:7	Material buckling, equilibrium value	
$c(z,t)$	2:7	Absorbtion term	
$c^1(z,t)$	2:11	Rod absorbtion	
$D(z,t)$	2:1	Absolute diffusion	
$E(z,t)$	2:1	Relative diffusion	
H	2:3	Extrapolated core height (m)	
h	2:5	Distance between two node points; $h = \frac{H}{N+1}$	
H_{crit}		Critical core height	
$I(z,t)$	2:8	Iodine concentration, measured with the xenon equilibrium concentration at infinite flux as basis	
$I^o(z)$	2:10	Equilibrium value of iodine concentration	
$\eta(z,t)$		$I(z,t) - I^o(z)$	
$K(z)$	2:4	Weight function in expression for total power	
K^*	2:13	Rod insertion	
N	2:5	Number of node points	
$P(t)$	2:4	Total power	
$r(t), R(z)$	2:15	Time and space distribution of reactivity disturbance	
t	2:1	Time in hours	
$u(z,t)$	2:7	Control term in buckling	

Symbol	First def. in equation	Explanation	Numerical value
$X(z,t)$	2:8	Xenon concentration, measured with the xenon equilibrium concentration at infinite flux as basis	
$X^o(z)$	2:10	Equilibrium value of xenon concentration	
$\xi(z,t)$		$X(z,t) - X^o(z)$	
z	2:1	Space coordinate	
$\alpha(z)$	2:7	Temperature coefficient, expressed as reactivity bounded in fuel temperature increase above the moderator at mean flux and infinite gitter.	-0.226%
		Normalization to mean flux $\bar{\Phi}$=5.65*10^{13}, M^2= 440 cm^2, multiply with $\frac{1}{0.0440}$	α = -0.0514
β	2:7	Xenon influence on changes in buckling (-3.2% on reactivity) at saturation	-0.73
γ_x	2:8	Fraction of xenon yield (relative to xenon + iodine yield)	0.05
γ_i	2:9	Fraction of iodine yield (relative to xenon + iodine yield)	0.95
$\Phi(z,t)$	2:1	Neutron flux, normalized to 5.65*10^{13} neutr/cm^2*sec.	
$\bar{\Phi}$	2:16	Mean flux	
$\Phi^o(z)$	2:3	Equilibrium flux	
$\varphi(z,t)$		$\phi(z,t) - \phi^o(z)$	
λ_x	2:8	Xenon disintegration constant	0.0756 h^{-1}
λ_i	2:8	Iodine disintegration constant	0.1058 h^{-1}
λ_1 c^1	2:11	Rod insertion length, rod absorbtion	
ψ	ch. 3.3	Form factor, $\Phi_{max}/\bar{\Phi}$	
σ_x	2:8	Microscopic xenon cross section normalized to $\bar{\Phi}$ = 5.65 10^{13} and time base in hours	2.29*10^{-18} cm^2 0.0465
Σ_f	2:1	Macroscopic fission cross area	
Σ_a	2:1	Macroscopic absorbtion cross area	
ν	2:1		

FLUX SHAPES AND DISTRIBUTIONS

The table shows the distribution in 20 space points of the undisturbed buckling.
The fluxes are shown in figure 3:1.
If only 10 values are written, the flux is symmetric.

A. Flat flux

$\Psi = 1.05$

h^2B^2	1.0	0	0	0	0	0	0	0	0	0	(symmetric)

B. Ditch flux 1

$\Psi = 1.31$ $\phi min/\phi max = 0.740$

h^2B^2 0.08155 ... 0.08155 (1,...7); -0.07727 ... -0.07727 (8,...10) (symm)

C. $\Psi = 1.289$

h^2B^2	0.7979	0.01015	0.00857	0.00755	0.00666
	0.00610	0.00567	0.00533	0.00512	0.00512

D. $\Psi = 1.35$

h^2B^2	0.8011	0.0124	0.0107	0.00933	0.00833	0.00755
	0.00688	0.00633	0.00600	0.00567	0.00722	0.00522
	0.0177	0.0178	0.0180	0.0184	0.0190	0.0200
	0.0212	0.8088				

E. Flattened sine flux

$\Psi = 1.35$

h^2B^2	0.0643	0.0519	0.0467	0.0439	0.0423	0.0415
	0.0411	0.0411	0.0796	0.0796 - - - - -		(symmetric)

F. Ditch flux 2

$\Psi = 1.45$ $\phi min/\phi max = 0.66$

h^2B^2 0.0752 ... 0.0752 (1,...6); 0.0790; -0.0773 -0.0773 (8,...13)

0.0816 ... 0.0816 (14,...20)

G. $\Psi = 1.47$

	(1,...7)			(8,...13)		
h^2B^2	0.0331	...	0.0331	-0.0126	...	-0.0126
	0.0457	...	0.0457			
	(14,...20)					

H. Sine flux

$\Psi = 1.57$

h^2B^2 0.02234 (constant)

Appendix 2

NUMERICAL METHODS

In order to simulate the xenon process, we have to solve eq. (2:5, 8, 9) with (2:7, 11, 12, 13) and the boundary conditions (2:6, 10) inserted. The solution can be divided into two different parts, here called equilibrium flux and transient calculations.

In the equilibrium state we put the derivatives of xenon and iodine equal to zero in (2:8, 9) and the equilibrium values are found to be:

$$X_k^o = \frac{1}{1 + \frac{\lambda_x}{\sigma_x \, \Phi_k^o}} \tag{1}$$

$$I_k^o = \frac{\gamma_i \, \sigma_x}{\lambda_i} \, \Phi_k^o \qquad k = 1, \ldots, N \tag{2}$$

In the xenon process the xenon and iodine concentrations are state variables and are uniquely determined by the differential equations.

A2.1 CALCULATION OF NEUTRON FLUX

We will solve the diffussion equation in every time step by writing eq. (2:5) in matrix form:

$$E \cdot \Phi + h^2(B^{2*} + \Delta B^2)\Phi = 0 \qquad (3)$$

where

$$E = \begin{bmatrix} -(E_1 + E_2) & E_2 & 0 & - & - & - & - & - & 0 \\ E_2 & -(E_2 + E_3) & E_3 & 0 & - & - & - & & \\ 0 & E_3 & -(E_3 + E_4) & E_4 & - & - & - & & \\ | & & & \ddots & & & & & \\ | & & & & \ddots & & & & \\ | & & & & & \ddots & & & \\ | & & & & & & \ddots & & E_N \\ | & & & & & & & \ddots & \\ 0 & - & - & - & - & 0 & E_N & & -(E_N+E_{N+1}) \end{bmatrix}$$

$$B^{2*} = \text{diag}\ (B_1^{2*},\ B_2^{2*},\ \ldots,\ B_N^{2*})$$

$$\Delta B^{2*} = \text{diag}\ (B_1^2 - B_1^{2*},\ \ldots,\ B_N^2 - B_N^{2*})$$

$$\Phi = (\phi_1,\ \phi_2,\ \phi_3,\ \ldots,\ \phi_N)^T$$

$$h^2 = (H/N + 1)^2$$

The symbols are explained in appendix 1. See also eq. (2:5).

The flux Φ is solved from (3)

$$\Phi = - E^{-1}(Q + \Delta Q)\Phi \qquad (4)$$

where $Q = h^2\ B^{2*}$

$\Delta Q = h^2(\Delta\ B^{2*})$

and ΔQ is a function of Φ.

In equilibrium we shall solve (4) with (1) inserted in Q, while $\Delta Q = 0$. During the transient, the xenon concentration is found from the differential equations.

We define a new matrix:

$$H = -E^{-1}(Q + \Delta Q) \tag{5}$$

where H is a function of Φ and X.

Our diffussion equation is thus formulated as:

$$\Phi = H \cdot \Phi \tag{6}$$

In order to solve (6), we use an iterative technique as it is impossible to get an explicit solution of Φ. We regard an eigenvalue problem:

$$H\ \Phi = \mathcal{H}\ \Phi \tag{7}$$

If we can find a real solution for the eigenvalue $\mathcal{H} = 1$, we have also found the solution of (6).

Now we know [2] that the eigenvalue problem for the one group diffussion equation in discrete form:

$$E\ \phi + \mu\ h^2\ B^2\ \phi = 0 \tag{8}$$

has a fundamental solution ϕ^o for the smallest eigenvalue, μ, which is proportional to the flux distribution of the poisoned reactor.

We see that (8) is transformed to:

$$- E^{-1}\ h^2\ B^2\ \phi = \frac{1}{\mu} \cdot \phi \quad \text{or}$$

$$H\ \phi = \frac{1}{\mu} \cdot \phi \tag{9}$$

Thus, we have to find the largest eigenvalue of (7) and its eigenvector ϕ.

As ϕ is implicitly defined in H, we must iterate. The elements of H are unknown depending on three terms in (2:7) which are included in Q (4) for every space point. In the equilibrium case we have $\Delta Q = 0$ and it remains to determine xenon equilibrium concentration and absorbtion c (or rod insertion λ) in equilibrium.

A2.2 CALCULATION OF THE EQUILIBRIUM FLUX. Order of operations.

The computation advances as follows:

In brackets are shown the names of Fortran subroutines, described in app. 3:

{1} Guess a start value of ϕ, e.g. (EGENV)

$$\phi^{(o)} = A \sin\left(\frac{\Pi}{H} \cdot z\right)$$

and a rod position $\lambda^{(o)}$ (FLOW) (CONT)

{2} The amplitude A of $\phi^{(o)}$ is normed by the power condition (2:6) (XNORM)

{3} The xenon equilibrium first value $X^{(o)}$ is determined from (1) and $\phi^{(o)}$, and $X^{(o)}$ and $\lambda^{(o)}$ are inserted in Q [eq. (3) - (4)]. (MATR) (RAND)

{4} The first iteration (iteration parameter = i) of H = $= H^{(i)}$ (i = 0) is then determined. (MATR) (FFGG)

{5} With the potense method is determined the largest eigenvalue $\mathcal{H}_{(i)}$ to the matrix $H^{(i)}$, i = 0, 1, 2, ... (iteration variable = ν)

$$\mathcal{H}^{(\nu)}_{(i)} = \frac{||H^{(i)} \cdot \phi^{(\nu)}_{(i)}||}{||\phi^{(\nu)}_{(i)}||} \qquad \nu = 0, 1, 2, \ldots \qquad \text{(XNORM)}$$

where we define

$$\phi_{(i)}^{(\nu+1)} = H^{(i)} \cdot \phi_{(i)}^{(\nu)} \qquad \text{(GFi)}$$

The norms are calculated from the power condition (2:6)

$$||\phi_{(i)}^{(\nu)}|| = B * \sum_{k=1}^{N} K_k \cdot \phi_k^{(\nu)}$$

$$||H^{(i)} \cdot \phi_{(i)}^{(\nu)}|| = ||\phi_{(i)}^{(\nu+1)}|| = B * \sum_{1}^{N} K_k \cdot \phi_{k(i)}^{(\nu+1)}$$

why

$$\mathcal{H}_{(i)}^{(\nu)} = \frac{\sum_{1}^{N} K_k \cdot \phi_{k(i)}^{(\nu+1)}}{\sum_{1}^{N} K_k \cdot \phi_{k(i)}^{(\nu)}} \qquad \text{(XNORM)}$$

$\mathcal{H}_{(i)}$ is accepted as an eigenvalue if

$$\left| \frac{\mathcal{H}_{(i)}^{(\nu+1)}}{\mathcal{H}_{(i)}^{(\nu)}} - 1 \right| \leq \varepsilon_3 \qquad \text{(XKAPPA)}$$

{6} The new eigenvector $\phi_{(i)}$ is accepted as the new approximation of ϕ for the i:th iteration. (EGENV,TEST)

{7} Points 2 - 5 are repeated now for the next iteration. In points 3 and 4 we shall put in the new value $\phi_{(i+1)}$ to get the matrix $H_{(i+1)}$. In order to avoid numerical instability it was necessary to use a relaxation method. Instead of $\phi_{(i+1)}$ we put in a value $\phi^{*}_{(i+1)}$ in the matrix H in order to get $H_{(i+1)}$, where $\phi^{*}_{(i+1)}$ is found by the formula:

$$\phi^{*}_{(i+1)} = \phi_{(i+1)} + \vartheta(\phi_{(i)} - \phi_{(i+1)}) \qquad \text{(TEST)}$$

where $0 \leq \vartheta \leq 1$.

The fastest computations were found for:

$\vartheta = 0.2 - 0.3$

{8} We accept $\phi_{(i+1)}$ as the right eigenvector for the eigenvalue $\mathcal{H}_{(i+1)}$ if:

$$\frac{\left|\left|\phi_{(i+1)} - \phi_{(i)}\right|\right|}{\left|\left|\phi_{(i+1)}\right|\right|} < \varepsilon_2 \qquad \text{(TEST)}$$

Now we have got a solution of the eigenvalue problem (7) for a certain value $\lambda^{(o)}$ of the insertion length of the rod. The eigenvalue $\mathcal{H} \neq 1$ and now λ shall be adjusted $\lambda^{(i)}$, $i = 0, 1, \ldots,$ until we have found the value $\mathcal{H} = 1$ and corresponding eigenvector ϕ which is the solution of the problem.

{9} We guess arbitrarily a new value of λ, called $\lambda^{(1)}$ and get a new solution ϕ and $\mathcal{H}^{(\mu)}$ by proceeding through points 2 - 8 again (iteration parameter μ). (FLOW)

{10} The next value of λ, $\lambda^{(\mu+1)}$, is calculated by linear regula falsi from previous values: (FLOW)

$$\lambda^{(\mu+1)} = \lambda^{(\mu)} - \frac{\left(\mathcal{H}^{(\mu)} - 1\right)\left(\lambda^{(\mu)} - \lambda^{(\mu-1)}\right)}{\mathcal{H}^{(\mu)} - \mathcal{H}^{(\mu-1)}}$$

and proceed through 2 - 8 and 10.

{11} When $\left[\mathcal{H}^{(\mu+1)} - 1\right] < \varepsilon_1$ we accept ϕ as the right flux. (FLOW)

{12} From the beginning we have choosen ε_2 and ε_3 between 10^{-2} and 10^{-3}. Now we make them much smaller, 10^{-6} to 10^{-8} have been found to be acceptable and increase the accuracy of the calculations. (FLOW)

A2.3 CALCULATION OF THE TRANSIENT

As the flux is assumed to be stationary all the time, we have to resolve the diffusion equation as the xenon concentration varies.

The differential equations (2:8) and (2:9) are integrated with the Runge - Kutta method, where the initial values are found from (1) and (2) after the equilibrium flux calculation. Richardson extrapolation is used in order to increase the accuracy.

The neutron flux is then calculated for every time step as described above. The following changes are made:

{1} As initial guess of the flux and rod position we use the values of the previous time step.

{2} The xenon concentration of previous time step is inserted.

Appendix 3

SHORT DESCRIPTION OF THE TRAXEN PROGRAM

A Fortran program package called TRAXEN (TRAnsients of XENon) has been written [6] in order to simulate the xenon process.

It was written initially for IBM 7090, but has later been changed to CD 3300 and CD 3600. The general numerical methods are described in appendix 2.

Figure 1 is a general chart of the subroutines. The inputs and outputs are described briefly and in detail in [6].

The calculation time grows approximately with N^2 where N is number of meshpoints. Calculation of an equilibrium flux for N = 20 takes about 1 - 4 seconds on a CDC 3600, depending on the buckling and flux shape.

The computing time for a transient depends much on stability or the amplitude of the transients. For bigger deviations for every time step we have either to decrease step length or increase the number of iterations.

The computing time for time step of a transient with 20 space points is 2 - 3 seconds.

Maximum number of meshpoints is 50.

INPUTS

The MAIN program reads in the following datas in groups (see appendix 1 for the therminology).

N, H

γ_i, σ_x, λ_x, λ_i

$\beta(1)$, ..., $\beta(N)$

λ, c^1, K [eq. (2:11-13)]

MVOID = 1 if hydraulics shall be calculated (not used here)

STL5 = 1 if a xenon control rod should be used (not used in the report)

P(t), r(t) in polygone chains

E(1), ..., E(N)

Q(1), ..., Q(N) (Q = unpoisoned buckling)

$\alpha(1)$, ..., $\alpha(N)$

R(1), ..., R(N)

stl2 = 1 for power control [eq. (2:6)]

stl2 = 2 for constant flux in one space point

K(1), ..., K(N) [coefficients in eq. (2:6)]

Printing instructions

Accuracy ε for the iterations

Initial values for flux, xenon and iodine (arbitrarily)

STAT if equilibrium flux is to be calculated

TRAN if transient flux is to be calculated

For a new calculation it is only necessary to read in the changed datas.

The input formats are described in detail in [6].

OUTPUTS

There is possible to print out a number of test values in the subroutines (see [6]).

In subroutine TRY is printed out:

time, λ, Ψ, $\bar{\phi}$, P, r(t), $\mu = \frac{1}{\mathcal{H}}$,

ϕ (normalized to $\phi_{max} = 1$),
ϕ in physical units,
X, I, φ, $\frac{d\xi}{dt}$, ξ,
} for every space point

fourier coefficients for fundamental mode and two overtones (sine waves).

CALCULATION OF EQUILIBRIUM FLUX

1. The MAIN program calls subroutine STAT where a few parameters are set.

2. STAT calls subroutine RAND with the rod insertion λ as a parameter.

3. In RAND is first calculated actual power P(t) and disturbance u(z,t), (zero in this case) with RRPP. If a xenon control rod is used its reactivity value is calculated (ROD). RAND calls FLOW with the rod insertion length as argument.

4. In FLOW is the iteration of rod insertion λ made. First is a flux calculated for the rod position, which was guessed. Then the routine iterates in λ until the eigenvalue $\mathcal{H}$ (eq. (A2:7)) is close to 1 (A2; points 1, 9 - 12).

5. FLOW calls the function EGENV and the subroutine MATR.
In MATR is defined the matrixes E, Q, ΔQ and H (eq. (A2:3 - 5)), see A2 points 3 - 4.
In EIGENV and its subroutine TEST is iteration made of the flux until the flux vector has converged (A2; points 1, 6, 7, 8).

6. EGENV calls the function XKAPPA. In XKAPPA is the biggest eigenvalue $\mathcal{H}$ of the eq. (A2:7) calculated with the potense method (A2, point 5). XKAPPA uses GFI to calculate the product $H * \Phi$

and XNORM to calculate the norm (A2, point 5).
XKAPPA accepts $\mathcal{H}$ as an eigenvalue when the iteration has converged.

CALCULATION OF THE TRANSIENT FLUX

1. The MAIN program calls subroutine TRAN, where the actual time is defined. If the power is zero TRAN calls subroutine EFFO in order to calculate the transients analytically until power gets positive. If the power is positive it calls RK2.
2. In RK2 is step length of the integration determined. A Richardson extrapolation is also made. The routine calls RK1, which is an ordinary Runge - Kutta subroutine.
3. RK1 calls DERI which defines the right hand side of the equations.
4. DERI in turn calls RAND. The following sequences are the same as in the equilibrium case from point 3. There are some differences, which are pointed out in appendix 2.

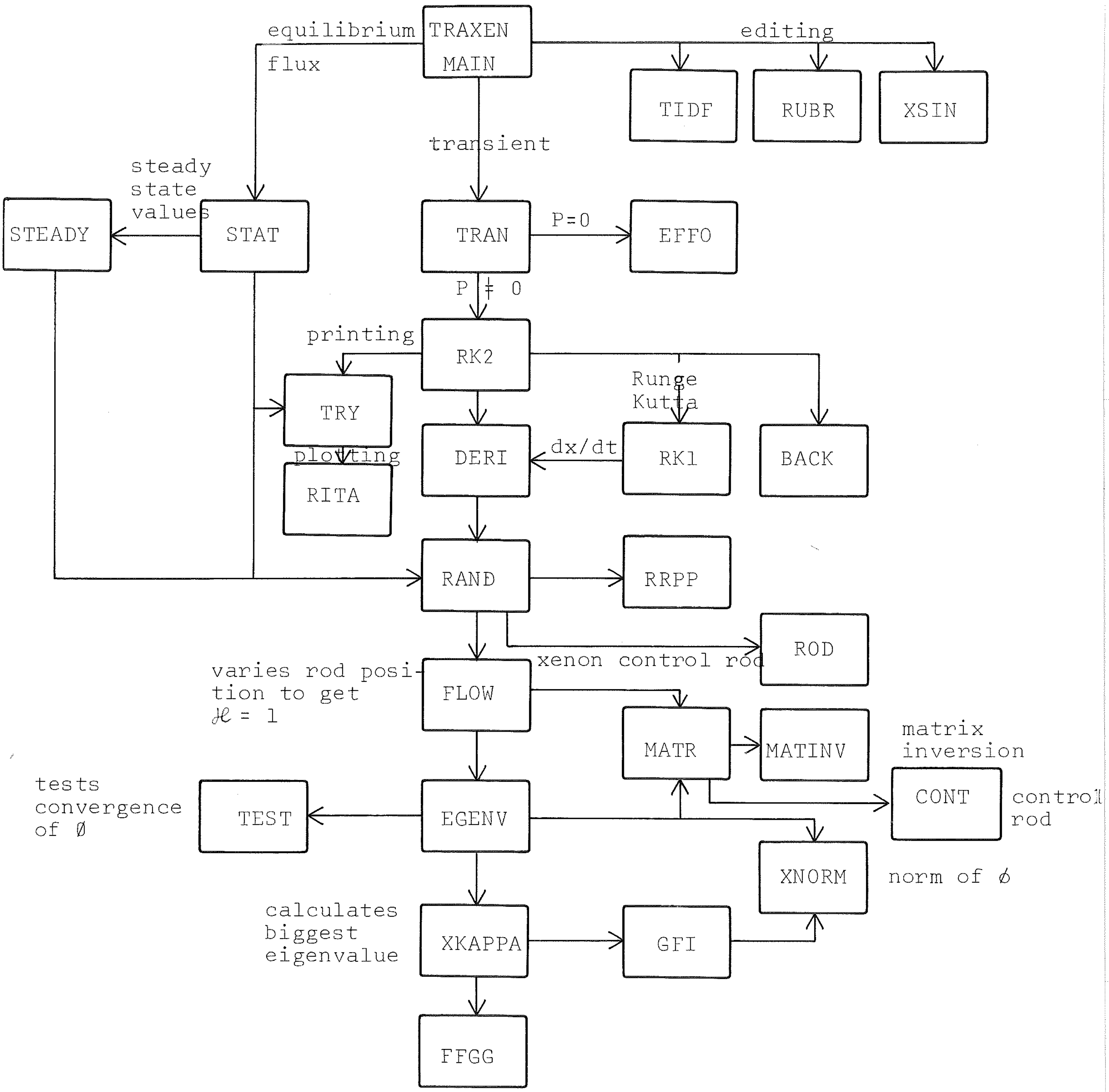


Figure 1. Blockdiagram of the TRAXEN subroutines

```
      PROGRAM TRAXEN
C     AUTHOR     GUSTAF OLSSON, DIVISION OF AUTOMATIC CONTROL LUND
C     THE PROGRAM CALCULATES EQUILIBRIUM NEUTRON FLUX DISTRIBUTION AND XENON
C     TRANSIENTS FOR A NONLINEAR FINITE DIFFERENCE XENON MODEL
C
C     REF.   GUSTAF OLSSON.  DIGITAL SIMULATION OF SPATIAL XENON OSCILLATIONS
C     DIV OF AUTOMATIC CONTROL LUND  REPORT 6911
C*
C     THE MATHEMATICAL MODEL IS DESCRIBED IN CHAPTER 2
C     THE NUMERICAL METHODS ARE DESCRIBED IN APP 2
C*
C     INPUT VARIABLES
C         VARIABLES AND FORMATS ARE DESCRIBED IN
C         G. OLSSON,  DIGITALT PROGRAM TRAXEN FOR TRANSIENTBERAKNINGAR
C                     AV XENONSVANGNINGAR I EN AXIELL REAKTORMODELL
C         SWEDISH STATE POWER BOARD, STOCKHOLM 1966  REPORT E-53/66
C*
C     OUTPUT
C         ALL INPUTS ARE REPEATED
C         TEST VALUES CAN BE PRINTED IF THE VARIABLES NTRY ARE 1
C         ALL OTHER OUTPUTS ARE PRINTED IN SUBROUTINE TRY
C*
C             TIME               ROD INSERTION (LAMBDA)        FORM FACTOR (PSI)
C             POWER              R = AMPLITUDE OF DISTURBANCE
C             NY = EIGENVALUE OF FUNDAMENTAL MODE  (EQ. A2.8)
C     IF STL5 = 1
C             AN EXTRA XENON CONTROL ROD IS DEFINED BY   ROD, IC, ICU, ICL
C             THIS ROD CAN BE INSERTED IN ORDER TO DAMP THE OSCILLATIONS
C             IC = CENTRE OF THE ROD     ICU= UPPER BOUNDARY
C             ICL = LOWER BOUNDARY
C
C             FI = NEUTRON FLUX
C*
C             FOR EVERY SPACE POINT IS PRINTED
C     IF NTRY1 = 2 IS ONLY DELTAFI AND DELTAX IN EVERY SPACE POINT CALCULATED
C             FI (NORM) = FI/FIMAX
C             FI(ABS) = FI IN PHYSICAL UNITS
C             XE = XENON CONC, MEASURED WITH THE XE EQUILIBRIUM CONC.
C                 AT INFINITE FLUX AS BASIS
C             JOD = IODINE CONC WITH SAME BASIS AS XE
C             DELTAFI(NORM)= (FI(T) - FI(0)/MEAN FLUX
C             DXDT = XENON TIME DERIVATIVE
C             DELTAX = XE(T) - XE(0)
C             FOURIER COEFF. FOR FLUX AND XENON FOR FUNDAMENTAL MODE AND
C             TWO OVERTONES, THE MODES ARE SINE WAVES
C
C
C     SUBROUTINES AND FUNCTIONS
C         TRY        STAT       STEADY     TRAN       EFFO
C         RK2        RK1        DERI       BACK       XSIN
C         RAND       FLOW       MATR       RRPP       EGENV
C         CONT       XKAPPA     XNORM      TEST       FFGG
C         GFI        MATINV     TIDF       RUBR       RITA
C         ROD        HYDRO      VOID    (HYDRO AND VOID ARE NOT USED IN THESE
C                                        CALCULATIONS, AS THE VOID IS NOT TAKEN
C                                        ACCOUNT FOR)
```

```
C
C
C     THE MAIN PROGRAM READS IN ALL DATA AND PRINT THEM OUT
C
C     IF A TRANSIENT IS TO BE CALCULATED SUBROUTINE TRAN IS CALLED
C     IF AN EQUILIBRIUM FLUX SHALL BE CALCULATED IT CALLS SUBROUTINE STAT
C     FOR EDITING IT CALLS SUBROUTINES TIDF AND RUBR
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),           CO       1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),      CO       2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP, CO       3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,  CO       4
     4 TID0, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),    CO       5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,      CO       6
     6 AX(20), TETA                                                     CO       7
      COMMON  NRIT, KURV, NSTANS,                                       CO       8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,           CO       9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                              CO      10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,          CO      11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                CO      12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                              CO      13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                            CO      14
      INTEGER AX
      REAL JOD,  JOD0,     LAMDAX,  LAMDAI,  LAMBDA,  K1                XECO    11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                      XECO    12
      DIMENSION  P(30), RT(30)                                          MAIN
      READ (5,25)  AX                                                   MAIN0003
      WRITE  (6,26) AX                                                  MAIN0004
      STL1 = 0
 1000 CALL RUBR ( I )                                                   MAIN0005
      GOTO ( 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14), I          MAIN0006
    1 READ ( 5, 100 ) N, H                                              MAIN0007
      HZ = H / FLOAT ( N+1 )                                            MAIN0008
      WRITE ( 6, 101) N, H, HZ                                          MAIN0009
      GOTO 1000                                                         MAIN0010
2     READ (5,102) GAMMAI, SIGMAX, LAMDAX, LAMDAI                       MAIN  11
      WRITE (6,103) GAMMAI, SIGMAX, LAMDAX, LAMDAI                      MAIN  12
      WRITE (6,1003)                                                    MAIN1012
403   READ (5,1004)  I1004, BEX, I1005, BEY, I1006, BEZ                 MAIN1013
      WRITE (6,1005) I1004, BEX, I1005, BEY, I1006, BEZ                 MAIN1014
      IF (I1004.EQ.(-1)) 1000, 31                                       MAIN
31    IF (I1004.GT.0)  34, 32                                           MAIN
34    BETA (I1004) = BEX                                                MAIN
32    IF (I1005.GT.0)  35, 33                                           MAIN
35    BETA (I1005) = BEY                                                MAIN
33    IF (I1006.GT.0)  36, 37                                           MAIN
36    BETA (I1006) = BEZ                                                MAIN
37    CONTINUE                                                          MAIN
      GOTO  403                                                         MAIN1019
3    READ(5,104) STL3, LAMBDA, C, K, MVOID                              MAIN  14
     WRITE(6,105)STL3, LAMBDA, C, K, MVOID                              MAIN  15
      IF (MVOID.EQ.1)  83, 84                                           MAIN
83    CALL VOID
84    CONTINUE                                                          MAIN
      READ(5,401) STL5, KSTYR, PART,PROD,DROD,ICONT,ABSO,XD,EPSX,EPSDX MAIN
      WRITE(6,402) STL5, KSTYR, PART,PROD,DROD,ICONT,ABSO,XD,EPSX,EPSDXMAIN
```

```
      GO TO 1000                                                       MAIN0016
    4 READ(5, 108) TMAX, HT                                            MAIN0017
      WRITE (6, 110) TMAX, HT                                          MAIN0018
      WRITE (6,111)                                                    MAIN0019
      DO 113 I113 = 1,30                                               MAIN
   50 READ (5,112)          TTT, PPP                                   MAIN0021
      WRITE (6,114) I113, TTT, PPP                                     MAIN0022
      T(I113) = TTT                                                    MAIN0023
      IF (TTT.LT.0.)  200, 51                                          MAIN
51    P(I113) = PPP                                                    MAIN
  113 CONTINUE                                                         MAIN0026
  200 CALL TIDF (PI,P)                                                 MAIN0027
      WRITE (6, 115)                                                   MAIN0028
      DO 116 I116 = 1,30                                               MAIN
      READ  (5, 117)  TTX,  RRR                                        MAIN0030
      WRITE  (6, 118) I116, TTX, RRR                                   MAIN0031
      T (I116) = TTX                                                   MAIN0032
      IF (TTX.LT.0.) 201,52                                            MAIN
52    CONTINUE                                                         MAIN
      RT (I116) =  RRR                                                 MAIN0034
  116 CONTINUE                                                         MAIN0035
  201 CALL  TIDF (RI, RT)                                              MAIN0036
      WRITE (6,160) (I,TI(I,1), TI(I,2),PI(I,1), PI(I,2) , RI(I,1) ,   MAIN0037
     C RI(I,2) , I= 1, M )                                             MAIN0038
      GOTO 1000                                                        MAIN0039
    5 WRITE (6, 121)                                                   MAIN0040
   70 READ (5, 120)  I116, EX, I117, EY,  I118, EZ                     MAIN0041
      WRITE (6, 122) I116, EX, I117, EY, I118, EZ                      MAIN0042
      IF (I116.EQ.(-1)) 1000,54                                        MAIN
54    IF (I116.GT.0)  61,55                                            MAIN
61    E(I116) = EX                                                     MAIN
55    IF (I117.GT.0)  62,56                                            MAIN
62    E(I117)  = EY                                                    MAIN
56    IF (I118.GT.0)  63,57                                            MAIN
63    E(I118) =EZ                                                      MAIN
57    CONTINUE                                                         MAIN
      GOTO  70                                                         MAIN0047
    6 CONTINUE                                                         MAIN0048
      WRITE (6, 126)                                                   MAIN0049
   60 READ (5, 128)   I128, XQ,  XALFA,  XR                            MAIN0050
      IF  (I128.EQ.(-1)) 1000, 65                                      MAIN
65    Q(I128) =XQ                                                      MAIN
      ALFA (I128) = XALFA                                              MAIN0053
      R(I128)= XR                                                      MAIN0054
      WRITE (6, 130)  I128,  Q(I128), ALFA(I128), R(I128)              MAIN0055
      GOTO 60                                                          MAIN0056
    7 READ (5, 132)  STL2, K1, FIREF, HN                               MAIN0057
      WRITE (6,134) STL2, K1, FIREF,  HN                               MAIN0058
      WRITE (6, 137)                                                   MAIN0059
  211 READ (5 , 136)  I135, BX, I136, BY, I137, BZ                     MAIN0060
      WRITE (6, 138) I135, BX, I136, BY, I137, BZ                      MAIN0061
      IF(I135.EQ.(-1)) 1000,71                                         MAIN
71    IF (I135.GT.0) 74,72                                             MAIN
74    B(I135) = BX                                                     MAIN
72    IF (I136.GT.0) 75,73                                             MAIN
75    B(I136) = BY                                                     MAIN
```

```
73      IF (I137.GT.0) 76, 77                                              MAIN
76      B(I137) = BZ                                                       MAIN
77      CONTINUE                                                           MAIN
        GOTO 211                                                           MAIN0066
8       READ (5,140) IT, NRIT, KURV, NTRY1, NTRY2, NTRY3, NTRY4, NTRY5,    MAIN0067
     1 NTRY6, NTRY7, NTRY8, NTRY9, NSTANS                                  MAIN0068
        WRITE (6,142)IT, NRIT, KURV, NTRY1,NTRY2, NTRY3, NTRY4, NTRY5,     MAIN0069
     1 NTRY6, NTRY7, NTRY8, NTRY9, NSTANS                                  MAIN0070
        GOTO 1000                                                          MAIN0071
9       READ (5, 144) EPS1, EPS2, EPS21, EPS3, EPS31, EPS32,               MAIN0072
     1 ITE1, ITE2, ITE3, TETA                                              MAIN0073
        WRITE (6, 146) EPS1, EPS2, EPS21, EPS3, EPS31, EPS32,              MAIN0074
     1 ITE1, ITE2, ITE3, TETA                                              MAIN0075
        GOTO 1000                                                          MAIN0076
   10   WRITE  (6, 152)                                                    MAIN0077
  310   READ (5, 150)  I150,  XXF,  YYX, ZZJ                               MAIN0078
        WRITE (6, 154)  I150,  XXF, YYX,  ZZJ                              MAIN0079
        IF (I150.EQ.(-1)) 1000, 81                                         MAIN
81      FI(I150) = XXF                                                     MAIN
        FI1(I150) = FI(I150)                                               MAIN0082
        XE0(I150) = YYX                                                    MAIN0083
        JODO(I150)= ZZJ                                                    MAIN0085
        GOTO 310                                                           MAIN0086
   11   STL1=1                                                             MAIN0087
        GOTO  1000                                                         MAIN0088
   12   STL1 =2                                                            MAIN0089
        GOTO  1000                                                         MAIN0090
   13   CONTINUE                                                           MAIN0091

C     THE SUBROUTINE XSIN CALCULATES DIFFERENT SINE FUNCTIONS FOR
C     THE FOURIER COEFFICIENT CALCULATION IN SUBROUTINE TRY
C
        CALL  XSIN(N)                                                      MAIN1091
        IF (STL1.EQ.2) 20, 1020                                            MAIN
1020    CALL STAT                                                          MAIN
        IF (STL1.EQ.1) 21, 1021                                            MAIN
1021    IF (EPS32.NE.0.) 1023, 20                                          MAIN
1023    EPS31 = EPS32                                                      MAIN
   20   CALL TRAN                                                          MAIN0095
   21   CONTINUE                                                           MAIN0096
        STL1 = 0                                                           MAIN *95
        GOTO  1000                                                         MAIN0097
   14   STOP                                                               MAIN0098
25      FORMAT (20A4)                                                      MAIN
26      FORMAT (20A4)                                                      MAIN
 100    FORMAT ( I10, F10.0 )                                              MAIN0101
 101    FORMAT ( 10X, 1HN, I4, 4X, 1HH, F8.2, 4X, 2HHZ, F8.3 )             MAIN0102
102     FORMAT  (4E10.0)                                                   MAIN 103
103     FORMAT (10X, 7HGAMMAI=, F6.2, 4X,  7HSIGMAX=, F10.5, 4X,           MAIN 104
     1 7HLAMDAX=, F10.5, 4X,  7HLAMDAI=,  F10.5)                           MAIN 105
104     FORMAT (I10, 2E10.0, 2I10)                                         MAIN 106
105     FORMAT (10X, 5HSTL3=,I4, 4X, 7HLAMBDA=, F10.4,4X, 2HC=, F10.4,     MAIN 107
     14X, 2HK=, I4, 4X, 6HMVOID=, I2 )                                     MAIN 108
 108    FORMAT ( 2E10.0)                                                   MAIN0109
 110    FORMAT (10X, 7HT(MAX)=, F10.4, 4X, 7HDELTAT=, F10.4)               MAIN0110
 111    FORMAT (13X,1HI, 4X, 4HT(I), 5X, 6HEFFEKT)                         MAIN0111
```

```
  112   FORMAT  (2E10.0)                                                    MAIN0112
  114   FORMAT (10X, I4, F10.4, F10.4)                                      MAIN0113
  115   FORMAT  (13X, 1HI, 3X,  4HT(I), 3X, 10HBUKT,STORN )                 MAIN0114
  117   FORMAT (2F10.0)                                                     MAIN0115
  118   FORMAT (10X,  I4, F10.4, F10.4 )                                    MAIN0116
  160   FORMAT (/12X,  1HI, 3X,  7HTI(I,1), 3X, 7HTI(I,2), 3X, 7HPI(I,1),MAIN0117
     C 3X, 7HPI(I,2), 3X, 7HRI(I,1), 3X, 7HRI(I,2) / (I13, 6F10.4))         MAIN0118
  121   FORMAT (14X, 1HI, 4X, 4HE(I), 11X, 1HJ, 4X, 4HE(J),11X, 1HK, 4X, MAIN0119
     C 4HE(K) )                                                             MAIN0120
  120   FORMAT (I10, E10.0, I10, E10.0, I10, E10.0)                         MAIN0121
  122   FORMAT (10X, I5, F10.4, 5X, I5, F10.4, 5X, I5, F10.4 )              MAIN0122
  126   FORMAT (10X,  9X,  4HQ(I), 5X,  7HALFA(I), 3X, 4HR(I))              MAIN0123
  128   FORMAT  (I10, 3E10.0)                                               MAIN0124
  130   FORMAT (10X, I5, 3F10.4 )                                           MAIN0125
  132   FORMAT  (I10, 3E10.0)                                               MAIN0126
  134   FORMAT (10X, 5HSTL2=,I3,2X, 3HK1= , F10.4, 3X, 6HFIREF= , F10.5, MAIN0127
     C 3X,  3HHN=, F10.4 )                                                  MAIN0128
  137   FORMAT  (14X, 1HI, 4X, 4HB(I),11X, 1HJ, 4X, 4HB(J), 11X, 1HK, 4X,MAIN0129
     C 4HB(K) )                                                             MAIN0130
  136   FORMAT ( I10, E10.0, I10, E10.0, I10, E10.0 )                       MAIN0131
  138   FORMAT (10X, I5, F10.4, 5X, I5, F10.4, 5X, I5, F10.4 )              MAIN0132
140     FORMAT  (13I5 )                                                     MAIN0133
142     FORMAT (10X, 3HIT=, I3, 2X, 5HNRIT=, I2, 2X, 5HKURV=,I2, 2X,        MAIN0134
     1 5HNTRY=, 9I4, 2X, 7HNSTANS=, I4 )                                    MAIN1134
144     FORMAT (6E10.0 / 3I10, F10.0)                                       MAIN0135
146     FORMAT (10X, 5HEPS1=, E12.3, 3X, 5HEPS2=, E12.3, 3X, 6HEPS21=,      MAIN0136
     1 E12.3, 3X, 5HEPS3=, E12.3, 3X, 6HEPS31=,  E12.3                      MAIN0137
     2,3X, 6HEPS32=, E12.3  / 10X, 5HITE1=                                  MAIN1137
     3 I4, 3X, 5HITE2=, I4, 3X, 5HITE3=,  I4, 3X, 5HTETA=, F7.2)            MAIN0138
  152   FORMAT  (23X, 2HFI,  10X, 2HXE,  9X, 3HJOD )                        MAIN0139
  150   FORMAT (I10,  3E10.0)                                               MAIN0140
  154   FORMAT  (10X, I5, 3X, 3F12.5 )                                      MAIN0141
401     FORMAT (2I5, 3F5.0, I5, 2F10.0, 2E10.0 )                            MAIN
402     FORMAT (10X,5HSTL5=,I3,3X,6HKSTYR=,I3,3X,5HPART=,F6.3,3X,5HPROD=,MAIN
     1 F6.4,3X,5HDROD=,F6.4,3X,6HICONT=,I4,3X,5HABSO=,F7.4,3X,3HXD=,        MAIN
     2 F7.4/ 10X,5HEPSX=,E12.4,3X,6HEPSDX=,E12.4 )                          MAIN
1003    FORMAT (14X,1HI,4X,4HBETA,11X,1HJ,4X,4HBETA,11X,1HK,4X,             MAIN 143
     1 4HBETA )                                                             MAIN 144
1004    FORMAT (I10,E10.0, I10, E10.0, I10, E10.0)                          MAIN 145
1005    FORMAT (10X,I5, F10.4, 5X, I5, F10.4, 5X, I5, F10.4)                MAIN 146
        END                                                                 MAIN0142
```

```
      SUBROUTINE TRY                                                    TRY 0001
C
C    TRY IS A PRINT OUT SUBROUTINE
C
C    THE SUBROUTINE IS CALLED BY RK2, STAT, EFFO
C    CALLS SUBROUTINE RITA IN ORDER TO PLOT THE SPATIAL DISTRIBUTION
C        OF FLUX, XE, AND IODINE
C
C
C        TIDO = ACTUAL TIME               LAMBDA = ROD POSITION
C        FFI = FORM FACTOR                SUM = MEAN FLUX
C        PP = POWER                       RR = AMPLITUDE OF DISTURBANCE
C        XNY = INVERTED VALUE OF THE EIGENVALUE YK1
C        IC = XENON CONTROL ROD CENTRE
C        ICU = UPPER BOUNDARY             ICL = LOWER BOUNDARY OF THE ROD
C        FI2 = FLUX NORMALIZED TO FI(MAX) = 1
C        FI = FLUX IN PHYSICAL UNITS   XE0 = XENON CONCENTRATION
C        JODO = IODINE CONCENTRAION    DELTFI = FLUX DEVIATION
C        DXDT = TIME DERIVATIVE OF XE
C        A1,A2,A3,X1,X2,X3 = FLUX AND XENON FOURIER COEFF.
C
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),           CO       1
     1 JODO(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),      CO       2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP, CO       3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,   CO       4
     4 TIDO, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),    CO       5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,      CO       6
     6 AX(20), TETA                                                     CO       7
      COMMON  NRIT, KURV, NSTANS,                                       CO       8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,           CO       9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                              CO      10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,          CO      11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                CO      12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                              CO      13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                            CO      14
      INTEGER AX
      REAL JOD,  JODO,      LAMDAX,  LAMDAI,  LAMBDA,  K1               XECO    11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                      XECO    12
      DIMENSION DELTFI(50)                                              TRY
      SUM= 0                                                            TRY 0003
      FIMAX =  0                                                        TRY 0004
C                                                                       TRY 0005
C    THE MAX VALUE OF THE FLUX IS CALCULATED
C
    1 DO 2  I2= 1,N                                                     TRY 0007
      SUM = SUM  + FI(I2)                                               TRY 0008
      FIMAX = AMAX1 (FIMAX, FI(I2))                                     TRY 0009
    2 CONTINUE                                                          TRY 0010
      IF (SUM.EQ.0.) 6,7                                                TRY
    7 CONTINUE                                                          TRY
C                                                                       TRY 0012
C    THE MEAN FLUX IS CALCULATED
C
      SUM = SUM/ FLOAT (N+1 )                                           TRY 0014
```

EXEMPEL PÅ PRODUKTFUNKTIONSANALYS FÖR DAMMSUGARE

Behovsfunktion:	Städa heltäckningsmatta
Huvudfunktion:	Avlägsna partiklar
Delfunktioner nivå 1:	Frigöra partiklar
	Transportera partiklar och luft
	Separera partiklar och luft
	Magasinera partiklar
	Avleda luft
	Åstadkomma förflyttning
Stödfunktioner:	Ge beröringsskydd
	Ge låg bullernivå
	Underlätta hantering
	Reducera dammlukt
	Ge attraktivitet

Se funktionsträd i figur 34.

690901 / Fredy Olsson / Fo

```
      FFI = FIMAX / SUM                                               TRY 0015
      FFIX = 1,0/SUM
C                                                                     TRY 0016
C    XKAPPA  ( OR YK1) IS INVERTED (SEE FLOW)
C
      XNY =  1, / YK1                                                 TRY 0018
C
C    THE MAXIMUM FLUX IS NORMALIZED TO 1,0
C
      DO  5  I5= 1,N                                                  TRY 0020
      DELTFI (I5) = (FI(I5) - FI1(I5)) *FFIX                          TRY
      FI2 (I5) = FI(I5) /FIMAX                                        TRY 0021
    5 CONTINUE                                                        TRY 0022
      GOTO 10                                                         TRY 0023
    6 FFI  = 0                                                        TRY 0024
      XNY = 0                                                         TRY 0025
      DO 8 I8 = 1, N                                                  TRY 0026
      FI2(I8) = 0                                                     TRY 0027
    8 CONTINUE                                                        TRY 0028
   10 CONTINUE                                                        TRY 0029
      WRITE  (6,3) TIDO, LAMBDA, FFI, SUM, PP, RR, XNY                TRY 0030
C
C    STL5 = 1   THE XENON CONTROL ROD IS USED
C
      IF (STL5,EQ,1) 50,51                                            TRY
50    WRITE (6,103) IC,ICU, ICL                                       TRY
51    CONTINUE                                                        TRY
      IF (TIDO,NE,0,0,AND,NTRY1,EQ,2) 202,201
201   WRITE (6,4)  (I4, FI2(I4), FI(I4), XEO(I4), JODO(I4), DELTFI(I4),TRY   34
    1 DXDT(I4),I4=1,N )                                               TRY
      GOTO 210
202   WRITE (6,400) (DELTFI(I4),I4=1,N)
210   CONTINUE
C
C    THE XENON DEVIATION IS CALCULATED
C    W = 1,0 ASSIGNS THAT TRANSIENT SHALL BE CALCULATED
C
      IF ( STL5, EQ , 1. AND , W , EQ , 1,0 ) 70,72                   TRY   N1
70    DO 71 I71 = 1,N                                                 TRY   N2
71    DELTFI( I71 ) = XEO( I71 ) - XSS( I71 )                         TRY   N3
      WRITE( 6,75 )  ( DELTFI( II ), II = 1,N )                       TRY   N4
72    CONTINUE                                                        TRY   N6
C
C    FOURIER COEFFICIENTS ARE CALCULATED  IF NTRY1 NOT = 2
C
      IF (NTRY1,EQ,2) 222,220
220   CONTINUE
      IF (STL2,EQ,2) 16,60                                            TRY
60    CONTINUE                                                        TRY
      DO 15 I15 = 1, N                                                TRY  137
15     FI2 (I15) = FI(I15) /SUM                                       TRY  237
      GOTO 18                                                         TRY  337
16    DO 17 I17 = 1, N                                                TRY  437
17    FI2(I17) = FI(I17)/ FIREF                                       TRY  537
18    CONTINUE
     A1 = 0.                                                          TRY   44
```

```
      X1 = 0.                                                          TRY  *44
     A2 = 0.                                                           TRY   45
      X2 = 0.                                                          TRY  *45
     A3 = 0.                                                           TRY   46
      X3 = 0.                                                          TRY  *46
      DO 25 I25 = 1, N                                                 TRY   47
      A1 = A1  * FI2(I25) * S(I25)                                     TRY   48
      X1 = X1 *  XE0 (I25) * S(I25)                                    TRY  *48
      A2 = A2  * FI2 (I25) * S(2*I25)                                  TRY   49
      X2 =X2 *  XE0 (I25) * S(2*I25)                                   TRY  *49
      A3 = A3  * FI2 (I25) * S(3*I25)                                  TRY   50
      X3 = X3*  XE0 (I25) * S(3* I25)                                  TRY  *50
25    CONTINUE                                                         TRY   51
      GOTO (26,36), STL2
26    WRITE (6,30) A1,A2, A3
      WRITE (6,40) X1, X2, X3                                          TRY 1055
      RETURN
36    WRITE (6,35) A1,A2, A3
      WRITE (6,40) X1,X2, X3                                           TRY 1355
      RETURN
30     FORMAT(//10X,29HFOURIERKOEFF FI (FIMED = 1,0)  //10X, 3HA1=,    TRY   56
     1E15.5, 5X, 3HA2=, E15.5, 5X, 3HA3=, E15.5 )                      TRY   57
35    FORMAT (//10X, 29HFOURIERKOEFF FI (FIREF = 1.0)  //10X,3HA1=,    TRY   58
     C E15.5, 5X, 3HA2=, E15.5, 5X, 3HA3=, E15.5 )                     TRY   59
40    FORMAT (//10X, 18HFOURIERKOEFF XENON  // 10X, 3HX1=, E15.5,      TRY  159
     1 5X, 3HX2=,  E15.5, 5X, 3HX3=,  E15.5 )                          TRY  259
    3 FORMAT ( //// 10X, 4HTIME ,F7.3, 3X, 6HLAMBDA,F12.5, 4X,3HPSI,   TRY 0031
     C F8.4, 4X, 5HFIMED, F8.3, 4X, 5HPOWER , F8.3, 4X, 1HR, F8.3 ,    TRY 0032
     C 4X,  2HNY , F8.4 )                                              TRY 0033
103   FORMAT (10X, 3HROD, 3X, 3HIC=, I4, 2X, 4HICU=, I4, 2X, 4HICL=,   TRY 3301
     1 I4 )                                                            TRY 3302
4     FORMAT  (// 12X, 1HI, 2X, 8HFI(NORM), 9X, 7HFI(ABS), 13X, 2HXE,  TRY 0035
     1 13X,3HJOD,11X,13HDELTAFI(NORM),6X,4HDXDT//(10X,I3,F10.6,E20.6,  TRY
     2 2F15.6,F20.6,F15.6))                                            TRY
75    FORMAT ( 10X, 6HDELTAX, 10F10.6 / (16X, 10F10.6 ))               TRY   N5
400   FORMAT (/10X,6HDELTFI,10F10.6/(16X,10F10.6))
222   RETURN
      END                                                              TRY   61
```

```
      SUBROUTINE  STAT                                                  STAT0001
C
C     REF,   GUSTAF OLSSON,  DIGITAL SIMULATION OF SPATIAL XENON OSCILLATIONS
C     DIV OF AUTOMATIC CONTROL LUND  REPORT 6911
C
C     THE SUBROUTINE IS CALLED BY MAIN
C
C     STAT  CALLS SUBROUTINE RAND IN ORDER TO CALCULATE THE EQUILIBRIUM
C     FLUX
C     THE STATIONARY VALUES ARE PRINTED OUT BY SUBROUTINE TRY
C     IF THE HYDRAULICS IS TO BE CALCULATED MVOID = 1,  SUBR HYDRO IS CALLED
C     IF THE VARIABLE STL5 = 1  THE STEADY STATE VALUES  AFTER A STEP
C          ARE CALCULATED WITH THE SUBROUTINE STEADY
C     THE SPATIAL DISTRIBUTION OF FLUX, XE, IODINE ARE PLOTTED BY RITA
C     W = 0 ASSIGNS THAT EQUILIBRIUM SHALL BE CALCULATED
C
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),           CO       1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),      CO       2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP, CO       3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,  CO       4
     4 TID0, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),    CO       5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,      CO       6
     6 AX(20), TETA                                                     CO       7
      COMMON  NRIT, KURV, NSTANS,                                       CO       8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,           CO       9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                              CO      10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,          CO      11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                CO      12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                              CO      13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                            CO      14
      INTEGER AX
      REAL JOD,  JOD0,      LAMDAX,  LAMDAI,  LAMBDA,  K1               XECO    11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                      XECO    12
      WRITE (6, 12)    AX                                               STAT0013
      W =  0                                                            STAT0002
      ITID = 1                                                          STAT0003
      STL 4 = 1                                                         STAT0004
      TID0  = 0                                                         STAT0005
      CALL  RAND (LAMBDA)                                               STAT0006
      DO 2  I2 = 1, N                                                   STAT0007
      FI1(I2) = FI(I2)                                                  STAT0008
      XE0 (I2)  = XE(I2)                                                STAT0009
      JOD0 (I2) = JOD(I2)                                               STAT0011
    2 CONTINUE                                                          STAT0012
      WRITE (6, 10)                                                     STAT0015
      DO  30  I30 = 1,N                                                 STAT
30    DXDT(I30) = 0.0                                                   STAT
      CALL TRY                                                          STAT0017
      IF (MVOID.EQ.1) 14,15                                             STAT
14    CALL HYDRO                                                        STAT
15    IF (NRIT.GT.0) 20, 16                                             STAT
16    IF (STL5.EQ.1) 50,17                                              STAT
50    CALL  STEADY                                                      STAT
17    RETURN
```

```
20        WRITE (6,25)  TIDO
          CALL RITA (FI, XEO, JODO, KURV, N)                               STAT0020
          GOTO 16                                                          STAT0022
12        FORMAT (10X,20A4)
   10     FORMAT (10X, 16HSTATIONART FLODE )                               STAT
25        FORMAT  (10X,  19HKURVA RITAD,  TIME= , F8,3 )                   STAT0016
          END                                                              STAT0021
                                                                           STAT0024
```

```
          SUBROUTINE  STEADY
C
C    THE SUBROUTINE IS CALLED BY STAT
C
C    CALCULATES THE EQUILIBRIUM VALUES AFTER A STEP DISTURBANCE IN REACTIVITY
C    CALLS RAND IN ORDER TO CALCULATE EQUILIBRIUM FLUX
C    THE STEADY STATE VALUES ARE PRINTED OUT BY SUBROUTINE TRY
C*
C*
          COMMON FI(50),FI1(50),FI2(50),XE(50), XEO(50), JOD(50),          CO      1
        1 JODO(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),      CO      2
        2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP, CO      3
        3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,   CO      4
        4 TIDO, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),    CO      5
        5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,      CO      6
        6 AX(20), TETA                                                     CO      7
          COMMON  NRIT, KURV, NSTANS,                                      CO      8
        1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,           CO      9
        2 STL1, STL2, STL3, STL4, STL5, STL6,                              CO     10
        3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,          CO     11
        4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                CO     12
        5 DEDA(20),DBDA(19), XALF1 (20),MVOID                              CO     13
          COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                           CO     14
          INTEGER AX
          REAL JOD,  JODO,     LAMDAX,  LAMDAI,  LAMBDA,  K1               XECO   11
          INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                     XECO   12
          W= 1,0
          STL4 = 1
          ITID= M                                                          STEAD
          TIDO = TI(ITID,1)                                                STEAD
          TID = TIDO                                                       STEAD
          CALL RAND (LAMBDA)                                               STEAD
C         REDEFINITION OF XE AND IODINE BEFORE PRINTING
          DO 2 I2 = 1,N                                                    STEAD
          XSS(I2) = XEO(I2)
          XEO(I2) = XE(I2)
          DIDT(I2) = JODO(I2)
2         JODO(I2) = JOD(I2)
          WRITE (6, 100)                                                   STEAD
100       FORMAT (10X, 12HSTEADY STATE   )                                 STEAD
          CALL  TRY                                                        STEAD
          DO 4 I4 = 1,N
          XEO (I4) = XSS(I4)
          JODO(I4) = DIDT(I4)
          DIDT(I4) = 0,0
4         XSS(I4) = XE(I4)
          RETURN
          END
```

```
      SUBROUTINE TRAN
C
C*
C    REF.   GUSTAF OLSSON.  DIGITAL SIMULATION OF SPATIAL XENON OSCILLATIONS
C    DIV OF AUTOMATIC CONTROL LUND  REPORT 6911
C
C    THE SUBROUTINE IS CALLED BY MAIN
C
C    TRAN IS THE MAIN SUBROUTINE FOR THE TRANSIENT CALCULATIONS
C    DEFINES THE ACTUAL TIME FOR THE INTEGRATION
C    IF POWER = 0  IT CALLS SUBROUTINE EFFO
C    IF POWER NOT EQUAL  0 IT CALLS RK2
C    W = 1.0 ASSIGNS THAT TRANSIENT SHALL BE CALCULATED
C
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),           CO       1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),      CO       2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP, CO       3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,   CO       4
     4 TID0, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),    CO       5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,      CO       6
     6 AX(20), TETA                                                     CO       7
      COMMON  NRIT, KURV, NSTANS,                                       CO       8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,           CO       9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                              CO      10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,          CO      11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                CO      12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                              CO      13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                            CO      14
      INTEGER AX
      REAL JOD,  JOD0,     LAMDAX,  LAMDAI,  LAMBDA,  K1               XECO  11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                      XECO  12
      STL4 = 2                                                          TRAN0002
      W = 1.                                                            TRAN0003
      DO 1 I1 = 2, M                                                    TRAN0004
      ITID = I1                                                         TRAN0005
      TID0 = TI(ITID, 1)                                                TRAN0006
      IF (PI(ITID,1).EQ.0..AND. PI(ITID,2).EQ.0.) 5, 3                  STAT
3     CALL RK2                                                          STAT
      IF(ABS(TMAX - TID0 ).LE. 1.E-4 ) 4,1                              STAT
    5 CALL EFFO                                                         TRAN0011
    1 CONTINUE                                                          TRAN0012
    4 RETURN                                                            TRAN0013
      END                                                               TRAN0014
```

```
      SUBROUTINE  EFF0
C
C    THE SUBROUTINE IS CALLED BY TRAN
C
C    IF POWER = 0 IT IS POSSIBLE TO ANALYTICALLY CALCULATE THE XENON
C        AND IODINE PROPAGATION.
C    SUBROUTINE EFF0 USES THESE ANALYTICAL EXPRESSIONS IN ORDER TO CALCULATE
C    XENON AND IODINE TRANSIENTS UNTIL THE POWER IS POSITIVE AGAIN.
C
C    EFF0 CALLS SUBROUTINE TRY IN ORDER TO PRINT OUT THE VALUES
C
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),               CO       1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),          CO       2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP,     CO       3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,      CO       4
     4 TID0, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),        CO       5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,          CO       6
     6 AX(20), TETA                                                         CO       7
      COMMON  NRIT, KURV, NSTANS,                                           CO       8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,               CO       9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                                  CO      10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,              CO      11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                    CO      12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                                  CO      13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                                CO      14
      INTEGER AX
      REAL JOD,  JOD0,      LAMDAX,  LAMDAI,  LAMBDA,  K1                   XECO    11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                          XECO    12
    1 DO  2 I2 = 1,N                                                        EFF00002
      FI(I2) = 0                                                            EFF00003
    2 CONTINUE                                                              EFF00004
      LAMBDA = 0                                                            EFF00005
      CALL TRY                                                              EFF00006
      YYY =  AMIN1 (TI(ITID,2) , TMAX )                                     EFF00007
      XXX = LAMDAI/ (LAMDAX- LAMDAI )                                       EFF00008
    4 HT  =  AMIN1 (HT, (YYY - TID0))                                       EFF00009
      DO 3 I3 = 1, N                                                        EFF00010
      JOD (I3) =  JOD0 (I3) * EXP (-LAMDAI * HT)                            EFF00011
      XE0(I3) = ( XE0(I3) - JOD0(I3)*XXX ) *EXP(- LAMDAX * HT) +            EFF00012
     C XXX * JOD (I3)                                                       EFF00013
      JOD0 (I3) =  JOD(I3)                                                  EFF00014
      XE (I3)  =  XE0 (I3)                                                  EFF00015
    3 CONTINUE                                                              EFF00016
      TID0 = TID0 + HT                                                      EFF00017
      IF  (ABS(YYY- TID0 ).LE. 1.E-4) 6, 10                                 EFF00018
10    CALL TRY                                                              EFF00019
      GOTO  4                                                               EFF00020
    6 CALL TRY                                                              EFF00021
      RETURN                                                                EFF00022
      END                                                                   EFF00023
```

```
      SUBROUTINE RK2
C
C
C     REF.   GUSTAF OLSSON. DIGITAL SIMULATION OF SPATIAL XENON OSCILLATIONS
C     DIV OF AUTOMATIC CONTROL LUND  REPORT 6911
C         APPENDIX 2
C
C     THE SUBROUTINE IS CALLED BY TRAN
C
C     RK2 DETERMINES TIME STEP AND MAKES RICHARDSON EXTRAPOLATION (USING BACK)
C     THE ROUTINE DETERMINES WHETHER TO PRINT OR NOT BY CALLING TRY
C     ITITIATES RUNGE KUTTA TIME STEP BY CALLING RK1
C
C     RK2 CALLS THE SUBROUTINES RK1, DERI, HYDRO, VOID, TRY, BACK, RITA
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),             CO       1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),        CO       2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP,   CO       3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,    CO       4
     4 TID0, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),      CO       5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,        CO       6
     6 AX(20), TETA                                                       CO       7
      COMMON  NRIT, KURV, NSTANS,                                         CO       8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,             CO       9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                                CO      10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,            CO      11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                  CO      12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                                CO      13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                              CO      14
      INTEGER AX
      REAL JOD,  JOD0,     LAMDAX,  LAMDAI,  LAMBDA,  K1                  XECO    11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                        XECO    12
      DIMENSION  DXE(50), DJOD(50)                                        RK2
      NR0 = NRIT                                                          RK2
      CALL DERI                                                           RK2 0003
      CALL TRY                                                            RK2 0004
      HT1=HT
      IF (MVOID.EQ.1)  60, 61                                             RK2
60    CALL  HYDRO                                                         RK2
61    CONTINUE                                                            RK2
      XXX =  AMIN1  (TI (ITID , 2) , TMAX )                               RK2 0005
1     HT  =  AMIN1  (HT, (XXX-  TID0 ))                                   RK2 0006
      IF (HT.EQ.0.) 1000, 2                                               RK2
2     X = LAMBDA                                                          RK2
      XYZ = RK1(X)                                                        RK2
C                                                                         RK2 0008
C     XYZ IS A TRUNCATION ERROR IN RUNGE-KUTTA
C                                                                         RK2 0010
      IF (NTRY7.EQ.1) 4,6                                                 RK2
4     WRITE (6,100) XYZ                                                   RK2
6     DO 10 I10 =1,N                                                      RK2
        DXE (I10)  = XE0 (I10)                                            RK2 0013
10      DJOD(I10)  = JOD0(I10)                                            RK2 0014
      CALL TRY                                                            RK2 0015
      IF (MVOID.EQ.1)  62, 63                                             RK2
```

```
62      CALL  HYDRO                                                        RK2
63      CONTINUE                                                           RK2
C                                                                          RK2 0016
C    THE SAME INTEGRATION AS BEFORE BUT WITH HALF THE TIME STEP LENGTH
C                                                                          RK2 0018
        CALL  BACK                                                         RK2 0019
        HT =  0,5 * HT                                                     RK2 0020
        CALL DERI                                                          RK2 0021
        DO 12 I12 = 1,2                                                    RK2 0022
        XYZ  = RK1 (X)                                                     RK2 0023
12      CONTINUE                                                           RK2 0024
        IF (NTRY7,EQ,1) 13,15                                              RK2
13      WRITE (6,100) XYZ                                                  RK2
C                                                                          RK2 0026
C    THE DIFFERENCE BETWEEN THE CALCULATIONS WITH DIFFERENT STEP LENGTHS
C                                                                          RK2 0028
15      DO 14 I14 = 1,N                                                    RK2
        DXE (I14) = (XE0(I14) -DXE(I14))* 1,06667                          RK2
        DJOD(I14) = (JOD0(I14)-DJOD (I14))*1,06667                         RK2
14      CONTINUE                                                           RK2 0032
        IF (NTRY8,EQ,1) 16,17                                              RK2
16      WRITE (6,110) (DXE(I) ,I=1,N ), (DJOD(K), K=1,N )                  RK2
C                                                                          RK2 0035
C    INTEGRATION WITH THE FIRST STEP LENGTH CONTINUES
C                                                                          RK2 0037
17      HT = HT + HT                                                       RK2
        IF (ABS(XXX - TID0),LE,1,E-4) 30,18                                RK2
18      DO  22  I22 =  1, IT                                               RK2 0040
        NRIT = NRIT - 1                                                    RK2  140
        HT  = AMIN1 ( HT, (XXX - TID0))                                    RK2 0041
        XYZ  = RK1 (X)                                                     RK2 0042
        IF (NTRY7,EQ,1) 26,27                                              RK2
26      WRITE (6,100) XYZ                                                  RK2
27      DO 24  K=1,N                                                       RK2
       XE0 (K) = XE0 (K) + DXE (K)                                         RK2 0045
24     JOD0(K) = JOD0 (K) + DJOD (K)                                       RK2 0046
        IF (ABS(XXX - TID0),LE,1,E-4) 30, 22                               RK2
22     CONTINUE                                                            RK2 0048
       CALL TRY                                                            RK2 0049
        IF (MVOID,EQ,1)  64, 65                                            RK2
64      CALL  HYDRO                                                        RK2
65      CONTINUE                                                           RK2
        IF (NRIT,EQ,0) 50, 18                                              RK2
30     CALL TRY                                                            RK2 0051
        IF (MVOID,EQ,1)  66, 67                                            RK2
66      CALL  HYDRO                                                        RK2
67      CONTINUE                                                           RK2
100     FORMAT  ( 5X,  3HRK2  /   15X,  E12,4 )                            RK2 0052
110    FORMAT (10X, 3HRK2, 5X, 14HTRUNKATIONSFEL /  (10X, 10F11,6))        RK2 0053
        HT=HT1
1000    RETURN                                                             RK2   54
50      WRITE (6, 55)  TID0                                                RK2 0055
55      FORMAT (10X, 19HKURVA RITAD,  TIME=   , F8,3)                      RK2 0056
        CALL RITA (FI, XE0, JOD0, KURV, N)                                 RK2 0057
        NRIT =  NR0                                                        RK2 0058
        GOTO  18                                                           RK2 0059

        END                                                                RK2 0060
```

```
      FUNCTION RK1( X)                                                  RK1 0001
C
C    REF.   GUSTAF OLSSON, DIGITAL SIMULATION OF SPATIAL XENON OSCILLATIONS
C    DIV OF AUTOMATIC CONTROL LUND  REPORT 6911
C       APPENDIX 2
C
C    THE FUNCTION IS CALLED BY RK2
C
C    RK1 IS A GENERAL RUNGE-KUTTA ROUTINE. INTEGRATES ONE TIME STEP AT A TIME
C    THE VARIABLE RK1 IS AN ERROR WHICH IS CALCULATED  OUT OF THE RUNGE-KUTTA
C    RK1 CALLS SUBROUTINE DERI IN ORDER TO CALCULATE THE RIGHT HAND SIDE
C
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),           CO       1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),      CO       2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP, CO       3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,  CO       4
     4 TID0, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),    CO       5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,      CO       6
     6 AX(20), TETA                                                     CO       7
      COMMON  NRIT, KURV, NSTANS,                                       CO       8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,           CO       9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                              CO      10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,          CO      11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                CO      12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                              CO      13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                            CO      14
      INTEGER AX
      REAL JOD,  JOD0,     LAMDAX,  LAMDAI,  LAMBDA,  K1                XECO  11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                      XECO  12
    1 DO 2 I2 =1, N                                                     RK1 0002
      XE(I2) = XE0(I2)                                                  RK1 0003
      JOD(I2) = JOD0(I2)                                                RK1 0004
    2 CONTINUE                                                          RK1 0005
   11 TID = TID0                                                        RK1 0006
      IF(NTRY2.EQ.1) 20,21                                              RK1
20    WRITE(6, 100) TID, (XE(I100), JOD(I100), I100= 1,N )              RK1
21    CONTINUE                                                          RK1
  100 FORMAT ( 10X, 3HRK1 / 10X, F10.6 / ( 10X, 6E16.6 ))               RK1 0009
      DO 4 I4=1, N                                                      RK1 0010
      XK1(I4) = HT*0.33333333*DXDT(I4)                                  RK1 0011
      NPI = N+I4                                                        RK1 0012
      XK1(NPI) = HT*0.33333333*DIDT(I4)                                 RK1 0013
      XK(I4) = XK1(I4)                                                  RK1 0014
      XK(NPI) = XK1(NPI)                                                RK1 0015
      XEPS(I4) = XK1(I4)                                                RK1 0016
      XEPS(NPI) = XK1(NPI)                                              RK1 0017
      XE(I4) = XE0(I4) +XK1(I4)                                         RK1 0018
      JOD(I4) = JOD0(I4) + XK1(NPI)                                     RK1 0019
    4 CONTINUE                                                          RK1 0020
   12 TID = TID0 + 0.33333333*HT                                        RK1 0021
      IF (NTRY2.EQ.1) 30,31                                             RK1
30    WRITE(6, 100) TID, (XE(I100), JOD(I100), I100= 1,N )              RK1
31    CONTINUE                                                          RK1
      CALL DERI                                                         RK1 0024
```

```
      DO 5 I5 =1, N                                                     RK1 0025
      XK2(I5) = HT*0,33333333*DXDT(I5)                                  RK1 0026
      NPI = N+I5                                                        RK1 0027
      XK2(NPI) = HT*0,33333333*DIDT(I5)                                 RK1 0028
      XE(I5) = XE0(I5) + 0,5*XK1(I5) +0,5*XK2(I5)                       RK1 0029
      JOD(I5) = JOD0(I5) + 0,5*XK1(NPI) + 0,5*XK2(NPI)                  RK1 0030
    5 CONTINUE                                                          RK1 0031
      IF (NTRY2.EQ.1) 40,41                                             RK1
   40 WRITE(6, 100) TID, (XE(I100), JOD(I100), I100= 1,N )              RK1
   41 CONTINUE                                                          RK1
   13 CALL DERI                                                         RK1 0034
      DO 6 I6 = 1, N                                                    RK1 0035
      XK2(I6) = HT*0,33333333*DXDT(I6)                                  RK1 0036
      NPI = N + I6                                                      RK1 0037
      XK2(NPI) = HT*0,33333333*DIDT(I6)                                 RK1 0038
      XEPS(I6) = XEPS(I6) - 4,5*XK2(I6)                                 RK1 0039
      XEPS(NPI) = XEPS(NPI) - 4,5*XK2(NPI)                              RK1 0040
      XE(I6) = XE0(I6) + 0,375*XK1(I6) + 1,125*XK2(I6)                  RK1 0041
      JOD(I6) = JOD0(I6) + 0,375*XK1(NPI) + 1,125*XK2(NPI)              RK1 0042
    6 CONTINUE                                                          RK1 0043
   14 TID = TID0 + 0,5*HT                                               RK1 0044
      IF (NTRY2.EQ.1) 50,51                                             RK1
   50 WRITE(6, 100) TID, (XE(I100), JOD(I100), I100= 1,N )              RK1
   51 CONTINUE                                                          RK1
      CALL DERI                                                         RK1 0047
      DO 7 I7 = 1, N                                                    RK1 0048
      XK4(I7) = HT*0,33333333*DXDT(I7)                                  RK1 0049
      NPI = N + I7                                                      RK1 0050
      XK4(NPI) = HT*0,33333333*DIDT(I7)                                 RK1 0051
      XK(I7) = XK(I7) + 4,*XK4(I7)                                      RK1 0052
      XK(NPI) = XK(NPI) + 4,*XK4(NPI)                                   RK1 0053
      XEPS(I7) = XEPS(I7) + 4,*XK4(I7)                                  RK1 0054
      XEPS(NPI) = XEPS(NPI) + 4,*XK4(NPI)                               RK1 0055
      XE(I7) = XE0(I7) + 1,5*XK1(I7) - 4,5*XK2(I7) + 6,*XK4(I7)         RK1 0056
      JOD(I7) = JOD0(I7) + 1,5*XK1(NPI) - 4,5*XK2(NPI) + 6,*XK4(NPI)    RK1 0057
    7 CONTINUE                                                          RK1 0058
   15 TID = TID0+ HT                                                    RK1 0059
      CALL DERI                                                         RK1 0060
      DO 8 I8 = 1, N                                                    RK1 0061
      XK4(I8) = HT*0,33333333*DXDT(I8)                                  RK1 0062
      NPI = N + I8                                                      RK1 0063
      XK4(NPI) = HT*0,33333333*DIDT(I8)                                 RK1 0064
      XK(I8) = XK(I8) + XK4(I8)                                         RK1 0065
      XK(NPI) = XK(NPI) + XK4(NPI)                                      RK1 0066
      XEPS(I8) = XEPS(I8) - 0,5*XK4(I8)                                 RK1 0067
      XEPS(NPI) = XEPS(NPI) - 0,5*XK4(NPI)                              RK1 0068
      XE0(I8) = XE0(I8) + 0,5*XK(I8)                                    RK1 0069
      JOD0(I8) = JOD0(I8) + 0,5*XK(NPI)                                 RK1 0070
    8 CONTINUE                                                          RK1 0071
      TID0 = TID                                                        RK1 0072
      EPSMAX = 0                                                        RK1 0073
      N2 = 2*N                                                          RK1 0074
      DO 9 I9 = 1, N2                                                   RK1 0075
      EPSMAX = AMAX1(EPSMAX, ABS( XEPS(I9)))                            RK1 0076
    9 CONTINUE                                                          RK1 0077
      RK1 = 0,2*EPSMAX                                                  RK1 0078
```

```
      TID = TIDO                                                        RK1 0079
      DO 16 I16 = 1, N                                                  RK1 0080
      XE(I16)= XE0(I16)                                                 RK1 0081
      JOD(I16) = JOD0(I16)                                              RK1 0082
   16 CONTINUE                                                          RK1 0083
      IF (NTRY2.EQ.1) 60,61                                             RK1
60    WRITE(6, 100) TID, (XE(I100), JOD(I100), I100= 1,N )              RK1
61    CONTINUE                                                          RK1
      CALL DERI                                                         RK1 0086
      RETURN                                                            RK1 0087
      END                                                               RK1 0088



      SUBROUTINE DERI                                                   DERI0001
C
C    REF.   GUSTAF OLSSON. DIGITAL SIMULATION OF SPATIAL XENON OSCILLATIONS
C    DIV OF AUTOMATIC CONTROL LUND  REPORT 6911
C       APPENDIX 2
C
C    THE SUBROUTINE IS CALLED BY RK2 AND RK1
C
C    DERI CALCULATES THE RIGHT HAND SIDE OF THE DIFFERENTIAL EQUATIONS (2.8-9)
C
C    DERI CALLS SUBROUTINE RAND
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),           CO     1
    1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),       CO     2
    2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP,  CO     3
    3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,   CO     4
    4 TIDO, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),     CO     5
    5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,       CO     6
    6 AX(20), TETA                                                      CO     7
      COMMON  NRIT, KURV, NSTANS,                                       CO     8
    1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,            CO     9
    2 STL1, STL2, STL3, STL4, STL5, STL6,                               CO    10
    3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,           CO    11
    4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                 CO    12
    5 DEDA(20),DBDA(19), XALF1 (20),MVOID                               CO    13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                            CO    14
      INTEGER AX
      REAL JOD,  JOD0,      LAMDAX,  LAMDAI,  LAMBDA,  K1               XECO  11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                      XECO  12
      GAMMAX  =  1. - GAMMAI                                            DERI0002
      CALL  RAND  (LAMBDA)                                              DERI0003
      DO 1 I1= 1, N                                                     DERI0004
      DXDT (I1)  = SIGMAX *GAMMAX* FI(I1) + LAMDAI * JOD(I1)  -         DERI0005
     C LAMDAX * XE(I1) - SIGMAX * FI(I1) * XE(I1)                       DERI0006
      DIDT(I1) = GAMMAI * SIGMAX * FI(I1) -  LAMDAI * JOD (I1)          DERI0007
    1 CONTINUE                                                          DERI0008
      IF (NTRY2.EQ.1) 10,11                                             DERI
10    WRITE (6,100) (DXDT (I2),DIDT(I2), I2=1,N)                        DERI
11    RETURN                                                            DERI
  100 FORMAT ( 10X, 4HDERI / ( 10X, 6F10.6 ))                           DERI0010
      END                                                               DERI0012
```

```
      SUBROUTINE BACK
C
C    THE SUBROUTINE IS CALLED BY RK2
C
C    BACK ROUTINE BACKS THE PROCESS ONE TIME STEP IN ORDER TO MAKE
C    A RICHARDSON EXTRAPOLATION,
C
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),           CO       1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),      CO       2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP, CO       3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,  CO       4
     4 TID0, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),    CO       5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,      CO       6
     6 AX(20), TETA                                                     CO       7
      COMMON  NRIT, KURV, NSTANS,                                       CO       8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,           CO       9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                              CO      10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,          CO      11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                CO      12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                              CO      13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                            CO      14
      INTEGER AX
      REAL JOD, JOD0,      LAMDAX, LAMDAI, LAMBDA,  K1                  XECO    11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                      XECO    12
    1 TID0 = TID0 - HT                                                  BACK0002
      DO 2 I2 = 1, N                                                    BACK0003
      XE0(I2) = XE0(I2) - XK(I2)* 0,5                                   BACK0004
      NPI = N + I2                                                      BACK0005
      JOD0(I2) = JOD0(I2) - XK(NPI) * 0,5                               BACK0006
    2 CONTINUE                                                          BACK0007
      RETURN                                                            BACK0008
      END                                                               BACK0009
```

```
      SUBROUTINE XSIN(NX)
C
C    THE SUBROUTINE IS CALLED BY MAIN
C
C    THE SUBROUTINE CALCULATES SINE FUNCTIONS FOR THE FOURIER COEFFICIENT
C        CALCULATION IN SUBROUTINE TRY
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),           CO       1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),      CO       2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP, CO       3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,  CO       4
     4 TID0, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),    CO       5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,      CO       6
     6 AX(20), TETA                                                     CO       7
      COMMON  NRIT, KURV, NSTANS,                                       CO       8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,           CO       9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                              CO      10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,          CO      11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                CO      12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                              CO      13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                            CO      14
      INTEGER AX
      REAL JOD,  JOD0,      LAMDAX, LAMDAI, LAMBDA,  K1                 XECO    11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                      XECO    12
      N3 = 3 * N                                                        XSIN     1
      V11=  1,/ FLOAT (N+1)                                             XSIN     2
      V = 3,1416 * V11                                                  XSIN     3
      V11= 2,* V11                                                      XSIN     4
     DO 20 I20 = 1, N3
   20 S(I20) = SIN (FLOAT(I20) * V ) * V11                              XSIN     6
      RETURN
      END
```

```
      SUBROUTINE  RAND (X)
C
C     REF.   GUSTAF OLSSON,  DIGITAL SIMULATION OF SPATIAL XENON OSCILLATIONS
C     DIV OF AUTOMATIC CONTROL LUND  REPORT 6911
C         APPENDIX 2  POINT 3
C
C     THE SUBROUTINE IS CALLED BY STAT, STEADY AND DERI
C
C     CALLS RRPP TO GET MOMENT VALUES OF POWER AND DISTURBANCE
C     CALLS ROD TO GET ACTUAL REACTIVITY VALUES FOR THE XENON CONTROL ROD
C     CALLS FLOW FOR CALCULATION AF THE ACTUAL VALUES OF FI, XE AND IODINE
C
C     X = GUESSED VALUE OF ROD POSITION. THE VALUE IS READ IN AS DATA FIRST TIME
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),            CO      1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),       CO      2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP,  CO      3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,   CO      4
     4 TID0, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),     CO      5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,       CO      6
     6 AX(20), TETA                                                      CO      7
      COMMON  NRIT, KURV, NSTANS,                                        CO      8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,            CO      9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                               CO     10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,           CO     11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                 CO     12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                               CO     13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                             CO     14
      INTEGER AX
      REAL JOD,  JOD0,     LAMDAX,  LAMDAI,  LAMBDA,  K1                 XECO   11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                       XECO   12
      CALL RRPP                                                          RAND0002
      IF (STL5.NE.1) 11,10                                               RAND
10    CALL ROD                                                           RAND 209
11    CALL FLOW(X)                                                       RAND    3
      IF (STL4.EQ.2) 20,14                                               RAND
14    DO 2 I2 = 1,N                                                      RAND
      JOD (I2) = SIGMAX * FI(I2) * GAMMAI / LAMDAI                       RAND0006
    2 CONTINUE                                                           RAND0008
20    RETURN                                                             RAND
      END                                                                RAND0010
```

```
      SUBROUTINE FLOW (X)
C
C
C     REF,   GUSTAF OLSSON,  DIGITAL SIMULATION OF SPATIAL XENON OSCILLATIONS
C     DIV OF AUTOMATIC CONTROL LUND  REPORT 6911
C         APPENDIX 2  POINT 1,9 - 12
C
C
C     THE SUBROUTINE IS CALLED BY RAND
C
C     FLOW CALLS THE SUBROUTINE MATR AND FUNCTION EGENV
C
C     LA0 = THE OLD VALUE OF THE ROD POSITION
C     LA1 = A NEW VALUE OF THE ROD POSITION
C     BY CALLING MATR AND EGENV THE ITERATION OF ROD POSITION PROCEEDS
C         UNTIL A VALUE OF THE ROD POSITION CAUSES THE EIGENVALUE FOR THE
C         FLUX TO BE CLOSE TO 1
C
C
C     YK0 = OLD EIGENVALUE              YK1 = NEW EIGENVALUE
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),           CO       1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),       CO       2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP,  CO       3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,   CO       4
     4 TID0, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),    CO       5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,       CO       6
     6 AX(20), TETA                                                      CO       7
      COMMON  NRIT, KURV, NSTANS,                                        CO       8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,            CO       9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                               CO      10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,            CO      11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                  CO      12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                                CO      13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                              CO      14
      INTEGER AX
      REAL JOD,  JOD0,      LAMDAX,  LAMDAI,  LAMBDA,  K1                 XECO   11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                        XECO   12
      REAL LA0, LA1, LA2                                                  FLOW0002
      LA0 = X                                                             FLOW0003
      IF (STL4,EQ,1) 2, 3                                                 FLOW
 3    CONTINUE                                                            FLOW
      CALL MATR (LA0)                                                     FLOW0004
 2    YK0 = EGENV (LA0)                                                   FLOW0005
      IF (ABS(YK0- 1,),LE, EPS1)  20,6                                    FLOW
 6    IF (YK0,LE,0,8,OR,YK0,GE,1,2) 10, 8                                 FLOW
 8    CONTINUE                                                            FLOW
      LA1  = LA0 + 0,5 * (YK0 - 1,)                                       FLOW0008
      GOTO 15                                                             FLOW0009
C                                                                         FLOW0010
C     LAMBDA IS DECREASED OR INCREASED WITH 0,1
C                                                                         FLOW0012
 10   LA1 = SIGN (0,1, YK0 - 1,) + LA0                                    FLOW0013
 15   DO 1 I1 = 1, ITE1                                                   FLOW0014
      J1 = I1                                                             FLOW0015
```

```
      IF(STL4,EQ,1) 4, 5                                                FLOW
5     CALL MATR (LA1)                                                   FLOW
4     YK1  = EGENV (LA1)                                                FLOW0017
      IF (ABS(YK1- 1,),LE,EPS1) 22, 9                                   FLOW
9     CONTINUE                                                          FLOW
      XYZ = (1, - YK1) / (YK1 -YK0)                                     FLOW0019
      IF (ABS(XYZ),GE,2,) 12, 16                                        FLOW
16    CONTINUE                                                          FLOW
C                                                                       FLOW0021
C   EXTRAPOLATION TO KAPPA (YK1) = 1.0
C                                                                       FLOW0023
      LA2 = LA1 + XYZ * (LA1 - LA0 )                                    FLOW0024
      GOTO 14                                                           FLOW0025
C                                                                       FLOW0026
C   THE EXTRAPOLATION IS MAXIMIZED TO THE DOUBLE DISTANCE BETWEEN THE POINTS
C                                                                       FLOW0028
12    LA2 = LA1 + SIGN (2,0 , XYZ) * (LA1 - LA0) .                      FLOW0029
14    IF (NTRY2,EQ,1) 30,31                                             FLOW
30    WRITE(6,100) I1,LA2, LA1, LA0, YK1, YK0                           FLOW
31    CONTINUE                                                          FLOW
      YK0 = YK1                                                         FLOW0033
      LA0 = LA1                                                         FLOW0034
      LA1 =LA2                                                          FLOW0035
    1 CONTINUE                                                          FLOW0036
      WRITE  (6, 102)                                                   FLOW0037
      STOP                                                              FLOW
   20 LA1 = LA0                                                         FLOW0040
   22 LAMBDA = LA1                                                      FLOW0041
      IF (NTRY9,EQ,1) 34,35                                             FLOW
34    WRITE (6,110) J1                                                  FLOW
35    CONTINUE                                                          FLOW
C                                                                       FLOW0044
C   AFTER THE ROD POSITION LAMBDA IS FOUND THE VALUE OF EPS2 AND EPS3 IS MADE
C         SMALLER IN ORDER TO GET A BETTER ACCURACY
C                                                                       FLOW0046
      EPS03= EPS3                                                       FLOW0047
      EPS02 =EPS2                                                       FLOW0048
      EPS3 = EPS31                                                      FLOW0049
      EPS2 = EPS21                                                      FLOW0050
      YK1 = EGENV (LA1)                                                 FLOW0051
      EPS3 = EPS03                                                      FLOW0052
      EPS2 = EPS02                                                      FLOW0053
      RETURN
  100  FORMAT (10X,  6H(FLOW), 2X,I3, 2X, 3HLA2, F12,4, 3HLA1, F12,4,   FLOW0031
     C 3HLA0, F12,4, 3HYK1, F12,4,  3HYK0,  F12,4)                      FLOW0032
  102 FORMAT  ( 10X,  32HNY KONVERGERAR EJ MOT 1 I  FLOW )              FLOW0038
110   FORMAT  (/  8X, I4, 1X, 18HITERATIONER I FLOW  )                  FLOW0043
      END                                                               FLOW0055
```

```
      SUBROUTINE  MATR  (X)
C
C    REF.   GUSTAF OLSSON.  DIGITAL SIMULATION OF SPATIAL XENON OSCILLATIONS
C    DIV OF AUTOMATIC CONTROL LUND  REPORT 6911
C         APPENDIX 2  POINT  3 , 4
C
C    THE SUBROUTINE IS CALLED BY SUBROUTINE FLOW AND FUNCTION EGENV
C
C    MATR CALLS MATINV WHICH IS A MATRIX INVERSION SUBROUTINE
C
C    THE ROUTINE INSERTS THE LAST ITERATION VALUES OF XE AND FLUX
C         INTO THE BUCKLING
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),          CO       1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),     CO       2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP, CO      3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,  CO       4
     4 TIDO, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),    CO       5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,     CO       6
     6 AX(20), TETA                                                    CO       7
      COMMON  NRIT, KURV, NSTANS,                                      CO       8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,           CO       9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                              CO      10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,          CO      11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                CO      12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                             CO      13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                           CO      14
      INTEGER AX
      REAL JOD,  JOD0,     LAMDAX,  LAMDAI,  LAMBDA,  K1               XECO  11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                     XECO  12
      DO  2  I2 = 1, N                                                 MATR0003
      DO  4  I4 = 1, N                                                 MATR0004
      D (I2,I4) = 0.                                                   MATR0005
    4 CONTINUE                                                         MATR0006
    2 CONTINUE                                                         MATR0007
      L = STL5 - 1                                                     MATR
      DO 5 I5 = 1, N                                                   MATR   8
      IF(L) 10,12,10                                                   MATR
10    STAB (I5) = 0.                                                   MATR
12    CONTINUE                                                         MATR
5     D(I5, I5) = E(I5) + E(I5 +1) - HZ * HZ*(CONT(I5, X) + R(I5)*RR   MATR   9
     1 +BETA(I5) * XE (I5) + STAB(I5))                                 MATR
      NM1 =  N-1                                                       MATR0011
      DO 8  I8 = 1, NM1                                                MATR0012
      D(I8,I8+1) =  - E(I8+1)                                          MATR0013
      D (I8+1, I8) = -E(I8+1)                                          MATR0014
    8 CONTINUE                                                         MATR0015
      IF (NTRY3.EQ.1) 14,16                                            MATR
14    WRITE(6,200) ((D(I,J), J=1,N), I=1,N )                           MATR
16    CONTINUE                                                         MATR
      CALL MATINV (D,N)                                                MATR
      IF (NTRY3.EQ.1)  18 , 20                                         MATR
18    WRITE (6,100) ((D(I,J), J=1,N), I=1,N )                          MATR
20    CONTINUE                                                         MATR
      RETURN                                                           MATR0019


200   FORMAT (10X,  17HD BEFORE INVERTED , 6H(MATR) /  (20X,10F10.5))
  100 FORMAT (10X, 6HD(I,J), 6H(MATR)  / (20X, 10F10.5))               MATR0018
      END                                                              MATR0020
```

```
      SUBROUTINE RRPP
C
C    THE SUBROUTINE IS CALLED BY RAND
C
C    RRPP INTERPOLATES IN POLYGONE CHAINS FROM SUBROUTINE TIDF TO GET
C    MOMENTARY VALUES OF POWER P(T) AND DISTURBANCE U(Z,T)
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),              CO       1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),         CO       2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP,   CO       3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,     CO       4
     4 TID0, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),       CO       5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,         CO       6
     6 AX(20), TETA                                                        CO       7
      COMMON  NRIT, KURV, NSTANS,                                          CO       8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,              CO       9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                                 CO      10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,             CO      11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                   CO      12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                                 CO      13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                               CO      14
      INTEGER AX
      REAL JOD,  JOD0,      LAMDAX,  LAMDAI,  LAMBDA,  K1                  XECO    11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                         XECO    12
      IF (TI(ITID,2).EQ.TI(ITID,1))  1,2                                   RRPP
2     CONTINUE                                                             RRPP
      RR = RI(ITID,1)+(RI(ITID,2) - RI(ITID,1))*( TID - TI(ITID,1)) /      RRPP0003
     C ( TI(ITID,2) - TI(ITID,1))                                          RRPP0004
      PP = PI(ITID,1)+(PI(ITID,2) - PI(ITID,1))*( TID - TI(ITID,1)) /      RRPP0005
     C ( TI(ITID,2) - TI(ITID,1))                                          RRPP0006
      RETURN                                                               RRPP0007
1     RR=  RI (ITID , 1)                                                   RRPP0008
      PP =  PI (ITID,  1)                                                  RRPP0009
      RETURN                                                               RRPP0010
      END                                                                  RRPP0011
```

```
      FUNCTION EGENV (X)
C
C    REF.   GUSTAF OLSSON,  DIGITAL SIMULATION OF SPATIAL XENON OSCILLATIONS
C    DIV OF AUTOMATIC CONTROL LUND  REPORT 6911
C        APPENDIX 2  POINT 1, 6
C
C    THE FUNCTION IS CALLED BY FLOW
C    EGENV CALCULATES THE INITIAL GUESS OF FLUX
C    IF FLUX IN THE FIRST SPACE POINT IS ZERO (WHICH IS THE CASE WHEN
C    THE CALCULATION STARTS FROM EQUILIBRIUM) A SINE DISTRIBUTION IS ASSUMED
C
C    IF FLUX IS NOT ZERO THE LAST FLUX DISTRIBUTION IS TAKEN AS THE FIRST
C    ITERATION VALUE
C    SUBROUTINE XNORM IS CALLED IN ORDER TO NORMALIZE TO THE PRESCRIBED POWER
C
C    MATR IS CALLED TO INSERT THE VALUES IN THE BUCKLING
C
C    FUNCTION XKAPPA IS CALLED TO CALCULATE THE EIGENVALUE FOR THE FLUX
C        FOR THE PRESENT ITERATION OF THE BUCKLING
C
C
C    THE SUBROUTINE CALLS FUNCTION TEST
C    IN TEST IS MADE ITERATION ON FLUX IN THE BUCKLING TERM
C    THE ITERATION IN THE FLUX CONTINUES UNTIL THE FLUX IN THE BUCKLING
C        DIFFERS FROM THE FLUX PREVIOUSLY CALCULATED BY EPS2
C
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),             CO      1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),        CO      2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP,   CO      3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,    CO      4
     4 TID0, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),      CO      5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,        CO      6
     6 AX(20), TETA                                                       CO      7
      COMMON  NRIT, KURV, NSTANS,                                         CO      8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,             CO      9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                                CO     10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,            CO     11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                  CO     12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                                CO     13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                              CO     14
      INTEGER AX
      REAL JOD,  JOD0,      LAMDAX,  LAMDAI,  LAMBDA,  K1                 XECO   11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                        XECO   12
      IF (FI(1).GT.0.) 12, 1                                              EGEN
1     CONTINUE                                                            EGEN
      DO 2  I2 = 1,N                                                      EGEN0003
      FI  (I2) = SIN ((3.1416 * FLOAT(I2))/FLOAT (N+1))                   EGEN0004
    2 CONTINUE                                                            EGEN0005
   12 CONTINUE                                                            EGEN0006
      CALL XNORM (FI ,FI )                                                EGEN0007
      DO 4  I4= 1,N                                                       EGEN0008
      FI2 (I4) = FI (I4)                                                  EGEN0009
    4 CONTINUE                                                            EGEN0010
      SIG = SIGMAX / LAMDAX                                               EGEN 110
```

```
      DO  10 I10 = 1, ITE2                                          EGEN0011
      IF (STL4.EQ.2) 20,21                                          EGEN
21    CONTINUE                                                      EGEN
      DO 6 I6 = 1 , N                                               EGEN 211
      FIX = SIG * FI(I6)                                            EGEN 311
6     XE(I6) = FIX / (FIX+1.)                                       EGEN 411
      CALL MATR(X)                                                  EGEN 511
20    J1 = I10                                                      EGEN  12
      ZK1 = XKAPPA  (FI2, X)                                        EGEN0013
      IF (TEST(FI,FI2).LE.EPS2) 3, 30                               EGEN
30    IF (NTRY7.EQ.1)  31, 10                                       EGEN
31    WRITE (6,104)  I10, (FI2 (I2), I2=1,N)                        EGEN
   10 CONTINUE                                                      EGEN0019
      WRITE (6, 102)                                                EGEN0020
      STOP                                                          EGEN
    3 CONTINUE                                                      EGEN0023
      IF(NTRY9.EQ.1) 34,35                                          EGEN
34    WRITE (6,110) J1                                              EGEN
35    IF (NTRY2.EQ.1) 36,37                                         EGEN
36    WRITE (6,100) (FI(I2), I2=1,N)                                EGEN
37    CONTINUE                                                      EGEN
      EGENV  = ZK1                                                  EGEN0033
      RETURN                                                        EGEN0034
  100 FORMAT (10X, 2HFI /  (10X, 10F10.6))                          EGEN0032
  102 FORMAT  (10X,  32HINGEN KONVERGENS AV FI I  EGENV  )          EGEN0021
  104 FORMAT  (10X, 9HITERATION, I3, 2X, 13HAV FI (EGENV)  /        EGEN0017
     C  (10X, 10F10.6))                                             EGEN0018
110   FORMAT  ( / 8X, I4, 1X, 10HITER EGENV )                       EGEN0030
      END                                                           EGEN0035
```

```
      FUNCTION CONT ( I, X)
C
C     REF.   GUSTAF OLSSON.  DIGITAL SIMULATION OF SPATIAL XENON OSCILLATIONS
C     DIV OF AUTOMATIC CONTROL LUND  REPORT 6911
C         APPENDIX 2  POINT 1
C     SEE EQ.  2,11-14
C
C     THE FUNCTION IS CALLED BY SUBROUTINE MATR AND FFGG
C
C     THE REACTIVITY FOR ROD CONTROL OR HOMOGENEOUS CONTROL IS CALCULATED
C     IF STL3 = 1   ROD CONTROL IS USED,  LAMBDA(X) DETERMINES THE REACTIVITY
C     STL3 = 2      HOMOGENEOUS CONTROL OR A FIX INSERTION LENGTH IS USED
C
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),            CO       1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),       CO       2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PR,  CO       3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,   CO       4
     4 TID0, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),     CO       5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,       CO       6
     6 AX(20), TETA                                                      CO       7
      COMMON  NRIT, KURV, NSTANS,                                        CO       8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,            CO       9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                               CO      10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,           CO      11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                 CO      12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                               CO      13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                             CO      14
      INTEGER AX
      REAL JOD,  JOD0,      LAMDAX,  LAMDAI,  LAMBDA,  K1                XECO    11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                       XECO    12
      GOTO ( 1, 2), STL3                                                 CONT0002
1     IF (X.LE.0..OR.X.GE.1.)  3, 10                                     CONT
10    CONTINUE                                                           CONT
      II = FLOAT(N) * X + 1.                                             CONT0004
      IF ( I-II ) 4, 5, 6                                                CONT0005
    4 CONT = C                                                           CONT0006
      RETURN                                                             CONT0007
    5 CONT = ( FLOAT(N)*X -FLOAT(II) + 1. ) * C                          CONT0008
      RETURN                                                             CONT0009
    6 CONT = 0.                                                          CONT0010
      RETURN                                                             CONT0011
    3 CONT = X * C                                                       CONT0012
      RETURN                                                             CONT0013
2     IF (I.LE.K) 7, 11                                                  CONT
11    CONTINUE                                                           CONT
      CONT = 0                                                           CONT0015
      RETURN                                                             CONT0016
    7 CONT = X * C                                                       CONT0017
      RETURN                                                             CONT0018
      END                                                                CONT0019
```

```
      FUNCTION XKAPPA(FIZ, X)
C
C    REF,   GUSTAF OLSSON, DIGITAL SIMULATION OF SPATIAL XENON OSCILLATIONS
C    DIV OF AUTOMATIC CONTROL LUND  REPORT 6911
C        APPENDIX 2  POINT 5
C
C    THE FUNCTION IS CALLED BY FUNCTION EGENV
C
C    XKAPPA IS THE EIGENVALUE FOR THE EXPRESSION   KAPPA*FI = H * FI
C    XKAPPA IS CALCULATED WITH A POTENSE METHOD
C
C    THE FUNCTION CALLS THE SUBROUTINE FFGG AND FUNCTION GFI
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),          CO      1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),      CO      2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP, CO      3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,  CO      4
     4 TID0, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),    CO      5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,      CO      6
     6 AX(20), TETA                                                     CO      7
      COMMON  NRIT, KURV, NSTANS,                                       CO      8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,           CO      9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                              CO     10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,          CO     11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                CO     12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                              CO     13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                            CO     14
      INTEGER AX
      DIMENSION  FIZ (50)                                               XKAP
      REAL JOD,  JOD0,     LAMDAX,  LAMDAI,  LAMBDA,  K1                XECO   11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                      XECO   12
      CALL FFGG (FIZ, X)
      ZK1 = GFI(FIZ)
      DO  1  I1 =  1, ITE3                                              XKAP0004
      J1 =I1                                                            XKAP0005
2     ZK2 = GFI(FIZ)
      IF (NTRY1.EQ.1) 10,11                                             XKAP
10    WRITE (6,100) I1, ZK2                                             XKAP
11    CONTINUE                                                          XKAP
      ZZK = (ZK2/ZK1) - 1,                                              XKAP0009
      IF (ABS(ZZK).LE.EPS3) 3, 12                                       XKAP
12    CONTINUE                                                          XKAP
      ZK1 = ZK2                                                         XKAP0011
    1 CONTINUE                                                          XKAP0012
      WRITE  (6, 102)                                                   XKAP0013
      STOP                                                              XKAP
    3 XKAPPA = ZK2                                                      XKAP0017
      IF (NTRY9.EQ.1) 14,15                                             XKAP
14    WRITE (6,110) J1                                                  XKAP
15    RETURN                                                            XKAP
  100 FORMAT (10X, 8H(XKAPPA), 4X, 2HT=, I3, 5X, 8HKAPPA(T), F12.6)     XKAP0008
  102 FORMAT  (10X, 48HINGEN KONVERGENS MOT STORSTA EGENV=RDET I XKAPPAXKAP0014
     C)                                                                 XKAP0015
110   FORMAT ( /8X, I4,11HITER XKAPPA  )                                XKAP0019
      END                                                               XKAP0021
```

```
      FUNCTION XNORM( YYY, ZZZ )
C
C     REF.   GUSTAF OLSSON,  DIGITAL SIMULATION OF SPATIAL XENON OSCILLATIONS
C     DIV OF AUTOMATIC CONTROL LUND  REPORT 6911
C         APPENDIX 2  POINT 2, 5
C
C     THE FUNCTION IS CALLED BY THE FUNCTIONS EGENV AND GFI
C
C     THE FUNCTION CALCULATES TWO DIFFERENT NORMS
C     IF STL 2 = 1 IT NORMALIZE THE FLUX TO THE POWER CONDITION
C     IF STL 2 = 2 THE FLUX IN ONE SPACE POINT IS NORMALIZED TO A VALUE FIREF
C
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),           CO      1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),      CO      2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP, CO      3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,   CO      4
     4 TID0, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),    CO      5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,      CO      6
     6 AX(20), TETA                                                     CO      7
      COMMON  NRIT, KURV, NSTANS,                                       CO      8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,           CO      9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                              CO     10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,          CO     11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                CO     12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                              CO     13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                            CO     14
      INTEGER AX
      REAL JOD,  JOD0,     LAMDAX,  LAMDAI,  LAMBDA,  K1                XECO   11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                      XECO   12
      DIMENSION  YYY(50), ZZZ(50)                                       XNOR
      GOTO (1, 4), STL2                                                 XNOR0003
    1 SUM = 0                                                           XNOR0004
      DO 2 I2 = 1, N                                                    XNOR0005
      SUM = SUM + B(I2) * YYY(I2)                                       XNOR0006
    2 CONTINUE                                                          XNOR0007
      XXX = PP*FLOAT( N+1 ) / ( SUM*K1 )                                XNOR0008
    5 DO 3 I3 = 1, N                                                    XNOR0009
      ZZZ(I3) = XXX*YYY(I3)                                             XNOR0010
    3 CONTINUE                                                          XNOR0011
      XNORM = 1./XXX                                                    XNOR0012
      RETURN                                                            XNOR0013
    4 I4 = HN * FLOAT (N+1)  + 0.5                                      XNOR0014
      XXX =  FIREF/ YYY(I4)                                             XNOR0015
      GOTO 5                                                            XNOR0016
      END                                                               XNOR0017
```

```
      FUNCTION TEST (FIXX, FIYY)
C
C     REF.   GUSTAF OLSSON.  DIGITAL SIMULATION OF SPATIAL XENON OSCILLATIONS
C     DIV OF AUTOMATIC CONTROL LUND  REPORT 6911
C         APPENDIX 2  POINT 6 - 8
C
C     THE FUNCTION IS CALLED BY FUNCTION EGENV
C
C     THE FUNCTION USES A RELAXATION METHOD IN ORDER TO REPLACE AN OLD
C     VALUE OF THE FLUX IN THE BUCKLING TERM WITH A NEW ONE
C
C     TETA = RELAXATION CONSTANT
C
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),            CO       1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),       CO       2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP,  CO       3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,   CO       4
     4 TID0, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),     CO       5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,       CO       6
     6 AX(20), TETA                                                      CO       7
      COMMON  NRIT, KURV, NSTANS,                                        CO       8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,            CO       9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                               CO      10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,           CO      11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                 CO      12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                               CO      13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                             CO      14
      INTEGER AX
      DIMENSION FIXX(50), FIYY(50)                                       TEST
      REAL JOD,  JOD0,      LAMDAX,  LAMDAI,  LAMBDA,  K1                XECO    11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                       XECO    12
      DIMENSION  XFI(50)                                                 TEST
      S = 0                                                              TEST0003
      SUM =0                                                             TEST0004
      SUM1 = 0.                                                          TEST0005
      DO 2 I2 =1,N                                                       TEST0006
      XFI(I2) = FIYY(I2) - FIXX(I2)
      SUM = SUM + ABS(XFI(I2))                                           TEST0008
      SUM1 = SUM1 + ABS(FIYY(I2))
      FIYY(I2) = FIYY(I2) + TETA * (FIXX(I2) - FIYY(I2))                 TEST
      FIXX(I2) = FIYY(I2)
    2 CONTINUE                                                           TEST0011
      SUM = SUM / SUM1                                                   TEST0012
      IF (NTRY1.EQ.1) 3,4                                                TEST
3     WRITE (6,100)  (XFI (I3), I3=1,N)                                  TEST
4     CONTINUE                                                           TEST
  100 FORMAT  (10X, 6H(TEST), 2X, 3HXFI /  (10X, 10F10.6))               TEST0014
      S  = AMAX1( S, SUM)                                                TEST0015
      TEST  =  S                                                         TEST0016
      RETURN                                                             TEST0017
      END                                                                TEST0018
```

```
      SUBROUTINE FFGG(FIXY, X)
C
C     REF.   GUSTAF OLSSON,  DIGITAL SIMULATION OF SPATIAL XENON OSCILLATIONS
C     DIV OF AUTOMATIC CONTROL LUND  REPORT 6911
C        APPENDIX 2  POINT 4
C
C     THE SUBROUTINE IS CALLED BY FUNCTION XKAPPA
C
C     THE SUBROUTINE CALCULATES THE MATRIX G WHICH IS A PART OF THE MATRIX
C     H. THIS MATRIX IS USED IN THE ROUTINES XKAPPA AND MATR.
C     W = 0 ASSIGNS THAT EQUILIBRIUM SHALL BE CALCULATED
C     W = 1.0 ASSIGNS THAT TRANSIENT SHALL BE CALCULATED
C
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),           CO       1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),      CO       2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP, CO       3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,  CO       4
     4 TID0, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),    CO       5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,      CO       6
     6 AX(20), TETA                                                     CO       7
      COMMON  NRIT, KURV, NSTANS,                                       CO       8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,           CO       9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                              CO      10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,          CO      11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                CO      12
     5 DEDA(20),DRDA(19), XALF1 (20),MVOID                              CO      13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                            CO      14
      INTEGER AX
      DIMENSION FIXY (50)                                               FFGG
      REAL JOD,  JOD0,      LAMDAX,  LAMDAI,  LAMBDA,  K1               XECO    11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                      XECO    12
      DIMENSION F(50)                                                   FFGG
      DO 6 I6 =1,N                                                      FFGG0003
      F(I6) =0                                                          FFGG0004
      DO 8 I8 = 1, N                                                    FFGG   5
      G(I6,I8) = 0.0                                                    FFGG   6
8     CONTINUE
    6 CONTINUE                                                          FFGG0008
      DO 10 I10 =  1,N                                                  FFGG0009
      F(I10) = HZ * HZ * (Q(I10) + W*ALFA(I10)*(FIXY(I10)-FI1(I10)))    FFGG0010
   10 CONTINUE                                                          FFGG0012
      IF (NTRY4.EQ.1) 16,17                                             FFGG
16    WRITE (6,100) (I2,  FIXY(I2), F(I2), I2 = 1, N)
17    CONTINUE                                                          FFGG
  100 FORMAT (/10X, 6H(FFGG), 2X, 1HI, 7X, 2HFI, 13X, 1HF /             FFGG0014
     C(10X, I9, F15.6, F15.6))                                          FFGG0015
      DO 12 I12 = 1,N                                                   FFGG0016
      DO 14 I14 = 1,N                                                   FFGG0017
      G(I12,I14) = D(I12, I14) * F(I14)
   14 CONTINUE                                                          FFGG0019
   12 CONTINUE                                                          FFGG0020
      IF (NTRY5.EQ.1) 18,19                                             FFGG
18    WRITE (6,102)  ((G(I,J), J=1,N), I=1,N )                          FFGG
19    RETURN                                                            FFGG
```

```
102   FORMAT (10X,  5HD * F   , 6H(FFGG) /  (20X, 10F10,5))             FFGG  22
      END                                                              FFGG0024
```

```
        FUNCTION  GFI(YYY)
C
C     THE FUNCTION IS CALLED BY THE FUNCTION XKAPPA
C     REF,   GUSTAF OLSSON, DIGITAL SIMULATION OF SPATIAL XENON OSCILLATIONS
C     DIV OF AUTOMATIC CONTROL LUND  REPORT 6911
C         APPENDIX 2  POINT 5
C
C     CALCULATES THE MATRIX=VECTOR PRODUCT G*FI
C     THE PRODUCT IS USED BY THE FUNCTION XKAPPA IN THE EIGENVALUE
C     CALCULATION WITH THE POTENSE METHOD
C     XNORM IS CALLED IN ORDER TO NORMALIZE THE VECTOR
C
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),           CO      1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),      CO      2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP, CO      3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,  CO      4
     4 TID0, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),    CO      5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,      CO      6
     6 AX(20), TETA                                                     CO      7
      COMMON  NRIT, KURV, NSTANS,                                       CO      8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,           CO      9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                              CO     10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,          CO     11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                CO     12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                              CO     13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                            CO     14
      INTEGER AX
      REAL JOD,  JOD0,     LAMDAX,  LAMDAI,  LAMBDA,  K1                XECO   11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                      XECO   12
      DIMENSION  YYY(50), PSI(50)                                       GFI
      DO 2 I2=  1,N                                                     GFI 0003
      SUM = 0                                                           GFI 0004
      DO 4 I4 = 1,N                                                     GFI 0005
      SUM = SUM  + G (I2,I4) * YYY(I4)                                  GFI    6
    4 CONTINUE                                                          GFI 0007
      PSI (I2) = SUM                                                    GFI 0008
    2 CONTINUE                                                          GFI 0009
      IF (NTRY6,EQ,1) 6,8                                               GFI
6     WRITE (6,100) (PSI(I2), I2=1,N)                                   GFI
8     CONTINUE                                                          GFI
 100   FORMAT (10X,  4HG*FI  / (15X,  10F10,6 ))                        GFI 0011
      GFI = XNORM  (PSI, YYY)                                           GFI 0012
      RETURN                                                            GFI 0013
      END                                                               GFI 0014
```

```
      SUBROUTINE  MATINV  (A, N )
C
C    THE SUBROUTINE IS CALLED BY MATR
C
C    MATINV IS A MATRIX INVERSION ROUTINE FOR A TRIDIAGONAL MATRIX
C
C
      DIMENSION  A(50,50), B(50), C(50)                              MATI
      IF (A(1,1).EQ.0.) 200, 1                                       MATI
1     CONTINUE                                                       MATI
      A(1,1) = 1./ A(1,1)                                            MATI   4
      B(1) = A(1,2) * A(1,1)                                         MATI   5
      C(1) = A(2,1)                                                  MATI 105
      NM1  = N- 1                                                    MATI   6
      DO 2 I2 = 2, NM1                                               MATI   7
      C(I2) = A(I2 + 1, I2 )                                         MATI 107
      A(I2,I2)  =  A(I2,I2)  -     C(I2-1)* B(I2-1)                  MATI   8
      IF (A(I2,I2).EQ.0.) 200,3                                      MATI
3     CONTINUE                                                       MATI
      A (I2,I2) = 1./  A(I2,I2)                                      MATI  10
2     B(I2) =  A(I2, I2+1) *  A(I2,I2)                               MATI  11
      A(N,N) = A(N,N) -    C(N-1) * B(N-1)                           MATI  12
      IF (A(N,N).EQ.0.)  200,4                                       MATI
4     CONTINUE                                                       MATI
      A(N,N)=1./ A(N,N)                                              MATI  14
      I= N                                                           MATI  15
      J = N-1                                                        MATI  16
10    A (I,J)  =  -A (I,J+1) * C(J) *A(J,J)                          MATI  17
      J =J- 1                                                        MATI  18
      IF (J.EQ.0) 20,10                                              MATI
20    J= I                                                           MATI  21
      I=J-1                                                          MATI  22
25    A(I,J) = - B(I) * A(I+1,J)                                     MATI  23
      I= I-1                                                         MATI  24
      IF (I.EQ.0) 30,25                                              MATI
30    J=J-1                                                          MATI  27
      I=  J                                                          MATI  28
      A (I,J) =  - A(J,J) * (A(I,J+1) * C(J) - 1.)                   MATI  29
      J= I-1                                                         MATI  30
      IF (J.EQ.0) 40,10                                              MATI
40    RETURN                                                         MATI  33
200   WRITE (6,100)                                                  MATI  34
100   FORMAT (// 10X, 15HSINGULAR MATRIS )                           MATI  35
      STOP                                                           MATI  36
      END                                                            MATI  37
```

```
      SUBROUTINE  TIDF  (ZI, Z)
C
C    THE SUBROUTINE IS CALLED BY MAIN
C
C    TIDF CALCULATES TIME FUNCTIONS DOR THE DISTURBANCE
C    U(Z,T) = R(T) * R(Z) AND THE TOTAL POWER  P(T) OUT OF A
C    POLYGONE CHAIN.
C
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),          CO      1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),     CO      2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP, CO      3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,  CO      4
     4 TID0, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),   CO      5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,     CO      6
     6 AX(20), TETA                                                    CO      7
      COMMON  NRIT, KURV, NSTANS,                                      CO      8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,          CO      9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                             CO     10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,         CO     11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,               CO     12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                             CO     13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                           CO     14
      INTEGER AX
      REAL JOD,  JOD0,     LAMDAX,  LAMDAI,  LAMBDA,  K1               XECO   11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                     XECO   12
      DIMENSION  ZI(30,2),  Z(30)
      K= 0                                                             TIDF0004
      M = 0                                                            TIDF0005
      DO 1  I =1, 100                                                  TIDF0006
      IPK = I+K                                                        TIDF0007
      IF (T(IPK +1).LT.0.)  10,5                                       TIDF
5     M = M+1                                                          TIDF
      IF ( T(IPK).EQ.T(IPK+1).AND. Z(IPK).NE.Z(IPK+1))  2, 6           TIDF
6     CONTINUE                                                         TIDF
      TI(I,1)  = T(IPK)                                                TIDF0011
      TI(I,2) = T (IPK+1)                                              TIDF0012
      ZI(I, 1) = Z (IPK)                                               TIDF0013
      ZI (I,2) = Z (IPK+1)                                             TIDF0014
      GOTO 1                                                           TIDF0015
    2 TI(I,1) = T(IPK+1)                                               TIDF0016
      ZI(I,1) = Z (IPK +1)                                             TIDF0017
      TI (I,2) = T(IPK +2)                                             TIDF0018
      ZI (I,2) = Z (IPK+2)                                             TIDF0019
      K= K+ 1                                                          TIDF0020
    1 CONTINUE                                                         TIDF0021
10    RETURN                                                           TIDF
      END                                                              TIDF0023
```

```
      SUBROUTINE RUBR ( I )
C
C     THE SUBROUTINE IS CALLED BY MAIN
C     RUBR IS A EDITING PROGRAM FOR WRITING ALL INPUT DATAS.
C
      DIMENSION A(20)                                                   RUBR
      COMMON /DATA/ B(15)                                               RUBR
      INTEGER A,B
      DATA ((B(I1), I1=1,14) =  3HGEO, 3HNUK, 3HKON, 3HTID,3HDIF,       RUBR
     1 3HBUK,3HEFF, 3HTRY,3HTOL, 3HREG, 3HSTA, 3HTRA, 3HEXE, 3HSLU )    RUBR
      READ ( 5, 1) A                                                    RUBR0005
1     FORMAT (A3, 19A4)                                                 RUBR
      WRITE (6,2) A
2     FORMAT(/10X, 1X, A3, 19A4)
      DO 3 I3 = 1, 14                                                   RUBR0010
      IF (A(1),NE, B(I3))  3, 4                                         RUBR
4     I = I3                                                            RUBR
      RETURN                                                            RUBR0013
    3 CONTINUE                                                          RUBR0014
      WRITE (6,10)                                                      RUBR
10    FORMAT (10X, 20HFELAKTIGT RUBRIKKORT   )                          RUBR
      STOP                                                              RUBR0015
      END                                                               RUBR0016
```

```
      SUBROUTINE RITA
      J = 1
      RETURN
      END
```

```
      SUBROUTINE HYDRO
      J = 1
      RETURN
      END
```

```
      SUBROUTINE VOID
      J = 1
      RETURN
      END
```

```
      SUBROUTINE  ROD
C
C   REF  G. OLSSON,  DIGITALT PROGRAM TRAXEN FOR TRANSIENTBERAKNINGAR
C                    AV XENONSVANGNINGAR I EN AXIELL REAKTORMODELL
C        SWEDISH STATE POWER BOARD, STOCKHOLM 1966  REPORT E-53/66
C    THE SUBROUTINE IS CALLED BY RAND
C    IT CALCULATES THE REACTIVITY VALUES FOR A XENON CONTROL ROD
C    THE POSITION OF THE ROD IS DETERMINED BY
C                  IC = ROD CENTRE   ICU = UPPER BOUNDARY
C                  ICL = LOWER BOUNDARY
C    PART = ROD LENGTH (LT 0.5)
      COMMON FI(50),FI1(50),FI2(50),XE(50), XE0(50), JOD(50),           CO    1
     1 JOD0(50), XK1(100), XK2(100), XK4(100), XK(100), XEPS(100),      CO    2
     2 TI(30,2),  PI(30,2), RI(30,2), B(50), ALFA(50), BETA (50),RR,PP, CO    3
     3 N,C,K, LAMDAX, LAMDAI, LAMBDA,SIGMAX, K1, FIREF, IT,HT, GAMMAI,  CO    4
     4 TID0, TID, ITID, TMAX, M, DELTA, HN, YK1, D(50,50), G(50,50),    CO    5
     5 Q(50), E(51),R(50),DXDT(50),DIDT (50), HZ,S(150), T(30), W,      CO    6
     6 AX(20), TETA                                                     CO    7
      COMMON  NRIT, KURV, NSTANS,                                       CO    8
     1 IC, ICU, ICL, PROD, DROD, PART , ABSO, STAB(50),KSTYR,           CO    9
     2 STL1, STL2, STL3, STL4, STL5, STL6,                              CO   10
     3 NTRY1,NTRY2, NTRY3,NTRY4,NTRY5,NTRY6,NTRY7,NTRY8,NTRY9,          CO   11
     4 EPS1, EPS2, EPS21, EPS3, EPS31, ITE1, ITE2, ITE3,                CO   12
     5 DEDA(20),DBDA(19), XALF1 (20),MVOID                              CO   13
      COMMON  XSS(50),XD, EPSX, EPSDX, ICONT                            CO   14
      INTEGER AX
      REAL JOD,  JOD0,     LAMDAX,  LAMDAI,  LAMBDA,  K1                XECO 11
      INTEGER  STL1, STL2, STL3, STL4,  STL5, STL6                      XECO 12
      IF (STL4.EQ.1) 420,1                                              ROD
1     GOTO (100,200,400,500)KSTYR
C    KSTYR = 1
C        IFI AND IX ARE CALCULATED   (EQ. 2,1 - 2,2)
C        XFI1 = THE INTEGRATED VALUE OF FLUX DEVIATION OVER THE FIRST HALF OF
C                  THE CORE  (2,1)
C        XDX1 = INTEGRATED VALUE OF DXDT  (2,2)
C        XFI2 = THE SAME AS XFI1 BUT OVER SECOND CORE HALF
C        XDX2 = THE SAME AS XDX1 BUT OVER SECOND CORE HALF
C        IC IS DEFINED OF XDX AND XFI    (2,5)
C    DROD = CONSTANT          PROD = CONSTANT
C
100   CONTINUE
      N1 = 0.5 * FLOAT(N) + 0.6
      SUM1 = 0
      SUM11 = 0.0
      DO 2 I2 = 1,N1
      SUM11 = SUM11 + FI(I2) - FI1(I2)
2     SUM1 = SUM1  + DXDT (I2)
      XDX1 = SUM1/  FLOAT (N1)
      XFI1=  SUM11/  FLOAT (N1)
      K = 2* N1 - N
      IF (K.EQ.0) 11,12
11    N1 = N1 +1
12    CONTINUE
      SUM2 = 0.
      SUM21 = 0.0
```

```
      DO 4 I4 = N1, N
      SUM21 = SUM21 + FI(I4) - FI1 (I4)
4     SUM2  = SUM2 + DXDT (I4)
      XX =  1./ FLOAT (N-N1+1)
      XDX2 =  SUM2 * XX
      XFI2 =  SUM21* XX
      XDX = 0.5 * (ABS(XDX1)+ ABS(XDX2))
      XFI =  0.5 * (ABS(XFI1) + ABS(XFI2))
      IC = 0.5 * FLOAT (N+1) - FLOAT(N)*(XDX * SIGN (DROD, XDX2) -
     1 XFI * SIGN(PROD, XFI2) ) + 0.5
      GOTO 300
C
C     KSTYR = 2
C         IFI = THE COORDINATE WHERE FLUX DEVIATION IS BIGGEST
C         IFI = THE COORDINATE WHERE DXDT IS BIGGEST
C         IC IS CALCULATED OUT OF IFI AND IDX    (2,9)
C
200   CONTINUE
      XDX = DXDT(1)                                                   ROD
      IDX= 1                                                          ROD
      DO 24 I24= 2,N
      IF (DXDT(I24) - XDX) 22,22,24
22    XDX = DXDT(I24)
      IDX = I24
24    CONTINUE
      XFI= FI(1) - FI1(1)                                             ROD
      IFI = 1                                                         ROD
      DO 28 I28 = 2,N
      DELT = FI(I28) - FI1(I28)                                       ROD
      IF (XFI - DELT ) 26, 26, 28
26    XFI = DELT
      IFI = I28
28    CONTINUE
      PROD = 1.0 -DROD                                                ROD
      IC = PROD* FLOAT(IFI) +  DROD* FLOAT (IDX)                      ROD
      GOTO 300
C
C     KSTYR = 3
C         IC IS DETERMINED IN DATA INPUT
C         XDELX = XE DEVIATION IN POINT IC
C         DXDT(IC) = DXDT IN POINT IC
C         THE ROD SWITCHES BETWEEN IC AND N + 1 - IC DEPENDING ON DELX AND DXDT
C         (2,10-11)
C         WHEN XDELX LT EPSX AND DXDT(IC) LT EPSDX THE ROD IS MOVED TO CORE
C                   CENTRE  (2,12-13)
C
400   XDELX = XE(1) - XSS(1)                                          ROD
      DO 402 I402 =2,N
      IF (XE(I402) - XSS(I402) - XDELX ) 402,402, 406                 ROD
406   XDELX  = XE(I402) - XSS(I402)                                   ROD
      IXE= I402
402   CONTINUE
404   IF (ABS(XDELX).LT.EPSX.AND.ABS(DXDT(IXE)).LT.EPSDX) 420,403
403   IF (XDELX.GT.(-XD * DXDT(IXE)*ABS(DXDT(IXE)))) 408,410
408   IC =  N+1- IXE
      GOTO  412
```

```
410     IC =IXE
412     CONTINUE
        GOTO  300
C
C     KSTYR = 4
C        THE COORDINATE WHERE FLUX DEVIATION IS BIGGEST IS CALCULATED
C        IF XDELX IN THIS POINT IS BIG ENOUGH (2,15) THE ROD IS MOVED TO DAMP
C                THE FLUX
C     IIC = NUMBER OF SPACE POINTS CORRESPONDING TO THE HALF ROD
C     THE ROD MUST ALL THE TIME BE INSIDE THE BOUNDARIES OF THE CORE
C     STAB = ABSORBTION VALUE ALONG THE ROD
C     ABSO = ABSORBTION CONSTANT
C
500     IXE = ICONT
        XDELX = XE(IXE) - XSS(IXE)
        GOTO 404
420     IC = 0,5 * FLOAT (N+1) + 0,5
300     CONTINUE                                                        ROD
        IIC  =      0,5*PART * FLOAT (N+1)
        N2 = N/2                                                        ROD
        IIC1 = 1 + IIC                                                  ROD
        IF (IC,LT,N2) 6, 14
14      CONTINUE
        IIC2 = N - IIC                                                  ROD
        IC = MIN0 (IIC2 , IC)                                           ROD
        GOTO 60                                                         ROD
6       IC = MAX0(IIC1,IC)                                              ROD
60       CONTINUE
        ICU =  IC + IIC
        ICL = IC- IIC
7       DO  8 I8 = 1,N
8       STAB (I8) = 0,0
        DO  10  I10 = ICL, ICU
10      STAB(I10) = ABSO
        STAB (ICU +1) =  ABSO * (0,5 * PART *FLOAT (N+1) - FLOAT (IIC))
        STAB (ICL -1) = STAB (ICU +1)
        IF(NTRY9,EQ,1)302,304
302     WRITE(6,306)
304     CONTINUE
306     FORMAT (10X, 8HROD KLAR)
        RETURN
        END
```

Appendix 4

DERIVATION OF A TRANSFER FUNCTION FOR A TWO POINT XENON MODEL

In order to get a simple estimation of the amplitude of the transients we will derive a transfer function for a two point linear xenon model, defined in [5]. From the transfer function it is possible to analytically derive the maximum amplitude of the output variable.

In [5] eq. (2:24 - 25) is derived an expression for the diffussion equation in two space points:

$$\alpha_1 \varphi_1^2 + \varphi_1(\beta\xi_1 + c_1 - g_1) + \phi_1^o(c_1 + \beta\xi_1) = 0 \qquad (1)$$

$$\alpha_2 \varphi_1^2 + \varphi_1(-\beta\xi_2 - c_2 + g_2) + \phi_2^o(c_2 + \beta\xi_2) = 0 \qquad (2)$$

$$\text{where } g_i = \frac{3}{h^2} - (B_i^{2*} + \alpha_i \phi_i^o) \quad i = 1, 2 \qquad (3)$$

We add one more control term u which will make it possible to disturb the flux externally.

$$\alpha_1 \varphi_1^2 + \varphi_1(\beta\xi_1 + c_1 + u_1 - g_1) + \phi_1^o(c_1 + u_1 + \beta\xi_1) = 0 \qquad (4)$$

$$\alpha_2 \varphi_1^2 + \varphi_1(-\beta\xi_2 - c_2 - u_2 + g_2) + \phi_2^o(c_2 + u_2 + \beta\xi_2) = 0 \qquad (5)$$

where g_i is defined in (3).

We linearize eq. (4) and (5) and assume:

$$u_1 = -u_2 = u \qquad (6)$$

Further we assume a symmetric equilibrium flux and homogeneous control:

$$g_1 = g_2 = g$$

$$\phi_1^o = \phi_2^o = \phi^o$$

$$X_1^o = X_2^o = X^o$$

$$c_1 = c_2 = c$$

Then (4) and (5) are simplified to:

$$-\varphi_1 g + \phi^o(c + u + \beta\xi_1) = 0 \tag{7}$$

$$\varphi_1 g + \phi^o(c - u + \beta\xi_2) = 0 \tag{8}$$

For the symmetric flux we have derived in [5], eq. (2:44):

$$g = \frac{2}{h^2} - \alpha\ \phi^o \tag{9}$$

We subtract (8) from (7) and eliminate c:

$$- 2\varphi_1 g + 2\phi^o u + \beta\phi^o(\xi_1 - \xi_2) = 0 \tag{10}$$

or

$$\varphi_1 = \frac{\beta\phi^o}{2g} (\xi_1 - \xi_2) + \frac{\phi^o}{g} \cdot u \tag{11}$$

where g is defined in (9).

We introduce the state variables:

$$x_1 = \xi_1 \quad x_2 = \eta_1 \quad x_3 = \xi_1 + \xi_2 \quad x_4 = \eta_1 + \eta_2 \tag{12}$$

and rewrite (11):

$$\boxed{\varphi_1 = \frac{\beta\phi^o}{2g} (2x_1 - x_3) + \frac{\phi^o}{g} u} \tag{13}$$

The xenon and iodine equations are rewritten directly from [5], eqs. (2:37) - (2:40) and we linearize the equations:

$$\frac{dx_1}{dt} = (- \lambda_x - \sigma_x\ \phi^o)x_1 + \lambda_i\ x_2 + \sigma_x(\gamma_x - X^o)\varphi_1 \tag{14}$$

$$\frac{dx_2}{dt} = - \lambda_i x_2 + \gamma_i \sigma_x \varphi_1 \tag{15}$$

$$\frac{dx_3}{dt} = (- \lambda_x - \sigma_x \phi^o)x_3 + \lambda_i x_4 \tag{16}$$

$$\frac{dx_4}{dt} = - \lambda_i x_4 \tag{17}$$

where the state variables are defined in (12). Eq. (13) is inserted in the system equations (14) - (17).

We get directly:

$$\begin{aligned} \frac{dx}{dt} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{18}$$

where

$$A = \begin{bmatrix} - \lambda_x - \sigma_x \phi^o \left[1 + (X^o - \gamma_x)\frac{\beta}{g}\right] & \lambda_i & \frac{\sigma_x \beta\phi^o}{2g} (X^o - \gamma_x) & 0 \\ \gamma_i \sigma_x \frac{\beta\phi^o}{g} & -\lambda_i & - \gamma_i \sigma_x \frac{\beta\phi^o}{2g} & 0 \\ 0 & 0 & - \lambda_x - \sigma_x \phi^o & \lambda_i \\ 0 & 0 & 0 & -\lambda_i \end{bmatrix}$$

$$B = \frac{\phi^o \sigma_x}{g} \begin{bmatrix} \gamma_x - X^o \\ \gamma_i \\ 0 \\ 0 \end{bmatrix}$$

$$C = \frac{\beta\phi^{o}}{2g}\,(2 \quad 0 \quad -1 \quad 0)$$

$$D = \frac{\phi^{o}}{g}$$

The transfer function

$$G(s) = \frac{Y(s)}{u(s)}$$

is easily derived from (18). As only two states are both controllable and observable, the transfer function is of second order

$$G(s) = C(sI - A)^{-1}\, B + D \tag{19}$$

Simple calculus of (18) inserted into (19) gives:

$$G(s) = \beta\sigma_x \left(\frac{\phi^{o}}{g}\right)^2 \left\{\frac{(\gamma_x - X^{o})(s-a_{22}) + \gamma_i\, a_{12}}{(s-a_{11})(s-a_{22}) - a_{12}\, a_{21}}\right\} + \frac{\phi^{o}}{g} =$$

$$= \frac{\frac{\phi}{g} s^2 + \left[\frac{\beta\sigma_x \phi^2}{g^2}(\gamma_x - X^{o}) - \frac{\phi}{g}(a_{11}+a_{22})\right]s + \frac{\beta\sigma_x \phi^2 \lambda_i}{g^2}(1-X^{o}) + \frac{\phi}{g}(a_{11}\cdot a_{22} - a_{12}\cdot a_{21})}{s^2 - (a_{11}+a_{22})s + a_{11}\, a_{22} - a_{12}\, a_{21}} \tag{20}$$

where the parameters a_{ij} are elements in A (18).

Appendix 5

DESCRIPTION AND PROOF OF THE ROD MOVEMENT FOR A SIMPLIFIED FLUX MODEL

The rod movement or absorbtion variation is included in the buckling term. Here we will show the variation of the buckling of a very simplified core model, when the flux distribution is disturbed. The statement is shown in eg.(4:1). Out of the proof we can conclude how rod movement will take place.
Let us assume a neutron flux which is stationary. Further we assume one group diffussion theory. The core is divided into two parts of equal length, normalized to 1. Buckling is space independent in the two halves of the core in fig. 1. Then the diffussion equation reads:

$$\frac{d^2 \phi_1}{dz^2} + B_1^2 \phi_1 = 0 \qquad (1)$$

$$\frac{d^2 \phi_2}{dz^2} + B_2^2 \phi_2 = 0 \qquad (2)$$

with solutions:

$$\phi_1 = A_1 \sin (B_1 z + \delta_1) \qquad (3)$$

$$\phi_2 = A_2 \sin (B_2 z + \delta_2) \qquad (4)$$

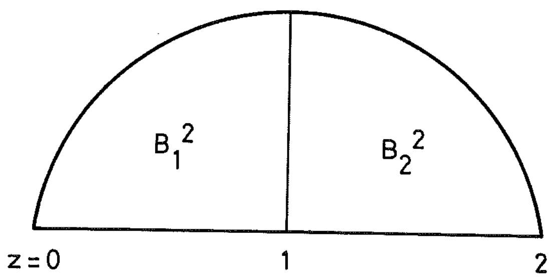


Figure 1: Flux distribution

The unknown parameters A_1, A_2, δ_1, δ_2 and one of the terms B_1, B_2 are determined out of the boundary conditions (5) - (9).

$$\phi_1(0) = 0 \qquad (5)$$

$$\phi_2(2) = 0 \qquad (6)$$

$$\left(\frac{d\phi_1}{dz}\right)_{z=1} = \left(\frac{d\phi_2}{dz}\right)_{z=1} \qquad (7)$$

$$\phi_1(1) = \phi_2(1) \tag{8}$$

$$\int_0^1 \phi_1(z)\, dz + \int_1^2 \phi_2(z)\, dz = C = \text{constant} \tag{9}$$

The last condition means that mean flux is unaltered.

Now, let us fix B_1. We solve (5) - (9) in order to get B_2 and find:

$$\boxed{\frac{B_2}{B_1} = -\frac{\operatorname{tg} B_2}{\operatorname{tg} B_1}} \tag{10}$$

We assume all the time, that:

$$0 \leq B_1 \leq \Pi/2$$
$$\Pi/2 \leq B_2 \leq \Pi \tag{11}$$

For the special case:

$$B_1 = \Pi/2$$

we have

$$B_2 = \Pi/2$$

This is the symmetric sine curve.

We will prove the following statements, and assume all the time (10) is satisfied.

If B_1 decreases (increases) from B_{11} to B_{12} we have an increase (decrease) in B_2 from B_{21} to B_{22} such as:

$$\left|(B_{11})^2 - (B_{12})^2\right| > \left|(B_{21})^2 - (B_{22})^2\right| \quad \text{or}$$

$$\left|\Delta B_1^2\right| > \left|\Delta B_2^2\right|$$

Thus if B_1 decreases we have

$$(B_{11})^2 + (B_{21})^2 > (B_{12})^2 + (B_{22})^2$$

The inequality is opposite if B_1 increases.

In order to prove (12) we must take the constraint (10) into account. Due to the cumbersome calculations we prefer to show the statement by numerical calculations.

The function $f = B_1^2 + B_2^2$ is monotonicly increasing with B_1, when B_1 varies between 0 and $\Pi/2$. The function is biggest for $B_1 = \Pi/2$ (see figure 2), i.e. the symmetric flux shape, where

$$B_1^2 + B_2^2 = \Pi^2/2$$

Out of this discussion we realize that (12) is valid all the time when B_1 varies between two values in the domain

$$0 \leq B_1 \leq \Pi/2.$$

To sum up, the variation in the buckling B_1^2 is always bigger than the variation of the buckling B_2^2, when the bucklings are within the boundaries (11).

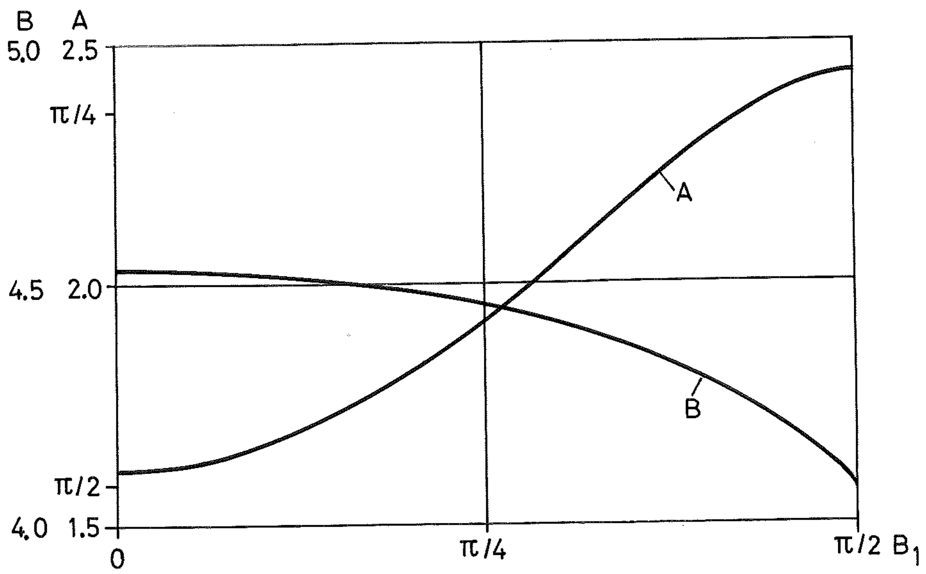


Fig. 2 - The variation of the sum of bucklings $B_1^2 + B_2^2$ (curve A) and of B_2 (curve B) as function of B_1