FLUX DISTRIBUTION AND CRITICALITY OF FAST REACTORS USING MULTIGROUP METHODS

by

Nicholas Kenneth Simon

A Thesis Submitted to the
Graduate Faculty in Partial Fulfillment of
The Requirements for the Degree of
MASTER OF SCIENCE

Major Subject: Nuclear Engineering

rsity
Of Science and Technology
Ames, Iowa

1963

TABLE OF CONTENTS

INTRODUCTION

Various calculational methods are utilized in the determination of neutron flux distributions in nuclear reactors. Neutron behavior has been described by diffusion theory, transport theory and stochastic processes. For complex systems it is necessary to utilize digital computers to obtain accurate results.

Diffusion theory has been used in many computer codes for reactor flux calculations. These include CURE (1) and ANGIE (2). Both of these codes were designed for use in an IBM computer with a large memory capacity. At the beginning of the present work, no machine of this type was available for use on the Iowa State University Campus.

The original objective of the author's work was to investigate the effect of certain calculational techniques on the results from a diffusion code. Examples of techniques that were to be studied included the calculation of initial flux distribution and of convergence to the critical mass. The techniques used affect the accuracy of obtaining results. The purpose of the original investigation was to improve computer programming techniques in their application to multigroup diffusion codes. In order to investigate these techniques at Iowa State it was necessary to develop a multigroup diffusion code for the Cyclone Computer and demonstrate its ability to solve problems. The difficulty and length of time involved in developing the code allowed for only a minor investigation to determine methods of improving neutron diffusion coding techniques.

In conjunction with the work a code was developed to study these techniques. The code equations were developed to be used in fast reactor

studies. The primary concern in this thesis will be to present the code developed and demonstrate its ability to solve problems.

The Computer Code

A multigroup, two-dimensional code called FABRA* was developed to investigate neutron diffusion approximations for reactor studies. The important features of FABRA are:

1. two dimensional (cylindrical geometry),
2. the ability to handle up to six reactor materials, one region and up to sixteen energy groups,
3. cosine-Bessel initial source guess,
4. up to 324 mesh points,
5. arbitrary external geometric boundaries,
6. scaled microscopic data used for input,
7. fractional fixed-point arithmetic for calculating coefficients,
8. floating point output.

It should be emphasized that any of these features can be changed, limited only by the memory capacity of the computer utilized. FABRA is assembled in machine language by SAR (Symbolic Assembly Routine). The assembly routine was developed for the Cyclone Computer and requires a 16,000 word memory in its present form.

Code Assumptions

The code was developed assuming that the neutron diffusion equations were adequate to describe the behavior of neutrons in the reactor system.

*Fast assembly breeder reactor approximation.

Finite difference equations are then developed assuming that the nuclear parameters are constant in the region about a point and that the neutron flux is symmetric about the coordinate axis. To determine convergence to the critical mass it was assumed that the "life cycle" process described in Weinberg and Wigner (7) applies to core neutrons.

Succeeding Material

In Section II the diffusion equation and its finite difference approximations are presented. A description and discussion of the FABRA Code is presented in Section III and the results of a sample problem in Section IV. The FABRA Code is included in the Appendices.

THE DIFFUSION EQUATION

The multigroup diffusion equation representing a steady-state critical flux distribution is

(1) $D^{(j)} \nabla^2 \phi^{(j)} - \Sigma_{re}^{(j)} \phi^{(j)} + S^{(j)} = 0$ (Reference 4)

Where: ϕ = neutron flux $\frac{\text{on}^1}{\text{cm}^2 \text{ sec}}$

D = diffusion length (cm)

Σ_{re} = total removal cross section (cm^{-1})

$S^{(j)}$ = source term (n/cm^3 sec)

(j) = group averaged value for the j group

The values for these terms are developed by the code from microscopic cross sections and spectral data according to the following relations:

(a) $D^{(j)} = \frac{1}{3\Sigma_{tr}^{(j)}}$ Where: $\Sigma_{tr}^{(j)}$ = Microscopic transport for the j^{th} group (cm^{-1})

(b) $\Sigma_{re}^{(j)} = \Sigma_{c}^{(j)} + \Sigma_{in}^{(j)} + \Sigma_{er}^{(j)} + \Sigma_{f}^{(j)}$

Where: $\Sigma_{c}^{(j)}$ = Microscopic capture cross section for the j^{th} group (cm^{-1})

$\Sigma_{in}^{(j)}$ = Microscopic inelastic scattering cross section for j^{th} group (cm^{-1})

$\Sigma_{er}^{(j)}$ = Macroscopic elastic removal cross section for j^{th} group (cm^{-1})

$f^{(j)}$ = Macroscopic fission cross section for j^{th} group (cm^{-1})

(c) $$S^{(j)} = X^{(j)} \sum_{k=1}^{g} \nu^{(k)} \Sigma_f^{(k)} \phi^{(k)} + \sum_{\ell=1}^{(j-1)} \Sigma_{in(\ell \to j)} \phi^{(\ell)} + \sum_{er}^{(j-1)} \phi^{(j-1)}$$

$X^{(j)}$ = Fraction of neutrons from fusion that is born into j^{th} group

$\nu^{(k)}$ = Number of neutrons per fission

(k) = Group averaged value for the k^{th} group

Microscopic data for the demonstration problem was taken from Yiftah (4).

Difference Equations*

To solve equation (1), $\nabla^2 \phi(r,z)$ is approximated by

(2) $$\nabla^2 \phi_{n,m} \to \frac{1}{h^2}\left\{\left(1 + \frac{1}{2n}\right)\phi_{n+1,m} + \left(1 - \frac{1}{2n}\right)\phi_{n-1,m} + \frac{1}{\theta^2}\phi_{n,m+1} + \frac{1}{\theta^2}\phi_{n,m-1} - \left(2 + \frac{2}{\theta^2}\right)\phi_{n,m}\right\}$$

where the (r,z) plane is replaced by a mesh and

$r = nh \qquad 1 \leq n \leq 18$

$z = m\theta h \qquad 1 \leq m \leq 18$

*These equations are developed in reference (5).

Now substituting (2) into (1)

$$(3)\quad \left[D\left(2 + \frac{2}{\theta^2}\right) + h^2 \Sigma_{re}\right] \phi_{n,m} = h^2 S_{n,m} + \frac{D}{\theta^2} \phi_{n,m+1}$$

$$\frac{D}{\theta^2} \phi_{n,m-1} + \left(D\left(1 + \frac{1}{2n}\right) \phi_{n+1,m} + D\left(1 - \frac{1}{2n}\right) \phi_{n-1,m}\right.$$

This relation is based on the assumption that D and Σ_{re} are constant in the region about the point (n,m). It is used in the code to compute flux at the point (n,m) when (n,m) is a distance h(θ) away from the boundary.

For the z axis $n = 1,\ m > 1$

$$(4)\quad \left[2D\left(2 + \frac{1}{\theta^2}\right) + h^2 \Sigma_{re}\right] \phi_{1,m} =$$

$$h^2 S_{1,m} + \frac{D}{\theta^2} \phi_{1,m} + 1\,\frac{D}{\theta^2} \phi_{1,m-1} + 4 D_{2,m}$$

For the axis $n > 1,\ m = 1$

$$(5)\quad \left[2D\left(1 + \frac{1}{\theta^2}\right) + h^2 \Sigma_{re}\right] \phi_{n,1}^{1} =$$

$$h^2 S_{n,1} + D\left(1 + \frac{1}{2n}\right) \phi_{n+1,1} +$$

$$D\left(1 - \frac{1}{2n}\right) \phi_{n-1,1} + 2\,\frac{D}{\theta^2} \phi_{n,2}$$

For the origin

$$(6)\quad \left[2D\left(2 + \frac{1}{\theta^2}\right) + h^2 \Sigma_{re}\right] \phi_{1,1} =$$

$$h^2 S_{1,1} + \frac{2D}{\theta^2} \phi_{1,2} + 4D \phi_{2,1}$$

Boundary Considerations

The boundary conditions used in developing the code are

(i) The neutron flux must be everywhere finite and non-negative

(ii) At an interface between two regions (A and B),

$$D_A \quad \phi_A^{(j)} = D_B \quad \phi_B^{(j)}$$

$$\phi_A^{(j)} = \phi_B^{(j)}$$

(iii) $\phi^{(j)} = 0$ at extrapolated boundary

(iv) Flux is symetric about the axis.

These conditions are justified in Glasstone (6).

Condition (i) is satisfied by generating a positive, finite initial source distribution and checking flux after each source iteration. Condition (ii) is automatically satisfied by the finite difference equations developed for boundary points. Condition (iii) is approximated by specifying a zero geometric boundary in the code input. No extrapolation is given because the extrapolation distance (d) is only about 1/4h for most problems, where (n,M) $\leq$18 and (d) varies with neutron energy. Condition (iv) is used to develop the flux equations on the axis.

COMPUTER CODE DESCRIPTION

The FABRA Code utilizes the second order difference equations shown in Section II to calculate an improved flux distribution over a two-dimensional (r,z) mesh. The initial flux distribution is calculated by assuming the classical Cosine-Bessel function distribution (6) of flux in the z and r directions, respectively, for a cylindrical reactor. The value of flux in the first energy group at the origin is assured to be one.

The source distribution is then calculated according to equation (c) on Page 4, Section II. After calculation of the source distribution, the flux mesh is swept until the average difference between the flux value and its previously calculated value is less than some prescribed epsilon. This is expressed as:

(7) $\left[\Sigma|\phi_n - \phi_o|\right]$ /TMNG -EPS where

ϕ_{nm} = the p^{th} flux value at a particular mesh point

ϕ_o = the (p-1) value at the same mesh point

TMNG = the total active or not zero mesh points along the z axis (m), the r axis, and through the energy groups (g).

EPS = the convergence criterion (part of the code input data).

The source is now recalculated to reflect the improved flux distribution. Iterations are continued until convergence is reached on both the source and the flux. The source is iterated by adjusting the size of the reactor. A flow diagram of the program operation is presented in Figure 1. Flag numbers in the figure refer to steps for solution in the following:

Steps for Solution

1. Input
2. (a) Compute total macroscopic cross sections in fixed point,

   (b) convert to floating point,

   (c) compact storage and clear remaining locations.
3. Generate initial flux distribution

   (a) Cosine distribution along z-axis

   (b) Bessel ($J_o(x)$) along r-axis

   (c) Sine-squared distribution along energy axis

   (d) Zero along geometric boundary
4. Calculate initial source based on initial flux distribution according to equation (c), page 4, section II.
5. Begin flux iteration according to equations listed in section II
6. After each flux iteration calculate the following to determine convergence:

$$\frac{\Sigma|\phi_n - \phi_o|}{\text{total mesh number}} - \epsilon$$

   where

   ϕ_n - new value of flux

   ϕ_o - previous flux value

   ϵ - convergence criterion.

Branch to source calculation on zero or negative value of above.

7. Calculate new source distribution.
8. Compute:

$$\frac{\Sigma|S_o - S_N|}{\text{total mesh number}} - \xi$$

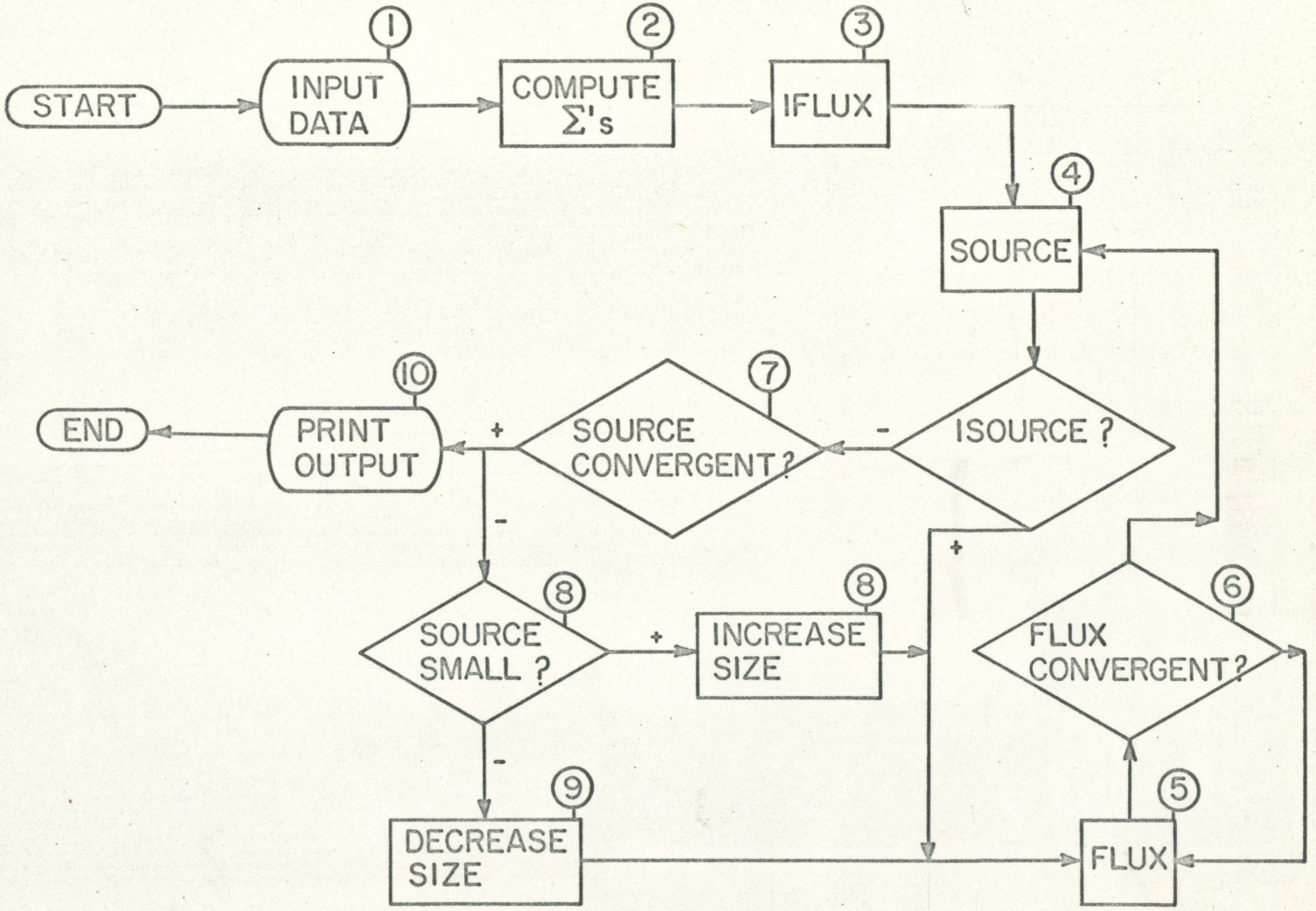

Figure 1. Flow diagram of FABRA program

where:

S_o - previous source value

S_N - new source value

ε - source convergence criterion

(a) Branch to print if quantity is zero or negative

(b) Branch to size increment if quantity is positive.

9. Compute: (after third iteration)

$$\Sigma (S_o - S_N)$$

(a) Decrease reactor mesh size if sum is negative (return to flux iteration)

(b) Increase reactor mesh size if sum is positive (return to flux iteration)

(c) After one sign change use binary chop to determine correct reactor size.

10. Print critical size and flux distribution on completion of the calculations.

Cross section calculations

The macroscopic cross sections are calculated in fixed point arithmetic and converted to floating point numbers for succeeding calculations. Macroscopic cross sections are put in for each energy group and material. The code will allow up to sixteen energy groups and six materials.

Macroscopic cross sections are computed by the formula

(8) $\Sigma = N\sigma$ where

Σ = Macroscopic cross section (cm^{-1})

N = Atom concentration (atoms/cm^3)

σ = Microscopic cross section (cm^2)

The total cross section is then computed

(9) $$\Sigma_T = \sum_1^h \Sigma(k) \quad \text{where}$$

$\Sigma_{(k)}$ = Cross section of the k th material (cm^{-1}).

The diffusion coefficients and the removal cross sections are then computed according to the relations on P. 4, eqs. (a) and (b).

The code was originally to be developed for a multi-region calculation. The multi-region flux calculation was coded and found to be too large for the memory available for coding purposes, and was not included in the final code. The cross-section calculations however, were set up to handle up to six regions. If the code is modified to a multi-region code the cross-section code will still be appropriate.

Flux Initialization

The one group solution for flux in a bare cylindrical reactor is:*

(10) $$\phi(r,z) = A J_o \left(\frac{2.405r}{R} \right) \cos \left(\frac{\pi Z}{H} \right)$$ (Reference 6)

where: ϕ = neutron flux (o^n/cm^2 - sec)

A = constant (amplitude)

J_o = bessel function of the first kind (zeroth order)

r = radial distance (cm)

R = critical radius (cm)

Z = axial distance (cm)

H = critical height (cm)

The user specifies an assumed critical height and radius. The code then calculates an initial flux distribution for the highest energy group based on the above equation. The flux for each succeeding group is then calculated by:

$$(11)\quad \phi^{(j)}(r,z) = \phi^{(1)}(r,z)\ \sin^2\left(\frac{\pi\, g}{G}\right)$$

where: $\phi^{(j)}(r,z)$ = flux of the j^{th} group (o^{n^1}/cm^2-sec)

$\phi^{(j)}(r,z)$ = flux of the first group (o^{n^1}/cm^2-sec)

g = g^{th} group

G = total number of groups

This sine-squared approximation to the energy dependent flux is used to simulate the ordinary peaking of the flux in about the fifth energy group.

Source Calculation

The source is calculated over the (r,z) mesh according to equation (c) on page 4. The first term in this equation represents that portion of the fissions which enter the j^{th} group. The second term represents inelastic scattered from other higher groups. No upscattering is assumed. The third term represents elastic removal from the proceeding group.

Flux Calculation

An improved flux distribution is calculated from equations (3) and (6) over the (r,z) mesh. The mesh is swept along the r axis from left to right. The external geometric boundary which is specified on input is delineated by a zero flux. The program is designed to treat a zero

flux as a geometric boundary. Any zero flux remains as zero throughout the entire calculation.

Iteration Procedures and Convergence

A reactor is critical when it produces as many neutrons as it loses. If the source neutrons produced in one generation are equal to the source neutron produced from the last generation the reactor will be critical. The equivalence of the source from generation to generation is the criterion used to determine criticality of a reactor by the code. The criticality criterion is

$$(12)\quad \sum_{1}^{MNG} \frac{\left|S^{(i)}(r,z,g) - S^{(i+1)}(r,z,g)\right|}{MNG} < \xi$$

where: $S^{(i)}(r,z,g)$ = source neutrons of the i^{th} generation at mesh point (r,z) in the g^{th} energy group (o^{n^1}/cm^3 - sec)

$S^{(i+1)}(r,z,g)$ = source neutrons at the same point of the $(i+1)^{th}$ generation (o^{n^1}/cm^3-sec)

ξ = input specified convergence criterion

MNG = Total non-zero flux points in the (M,N,G) mesh.

If the reactor is not critical it is either sub-critical or super-critical. A subcritical reactor can be made critical by increasing its size; a supercritical reactor can be made critical by decreasing its size. The code determines sub or super-criticality by the equation:

$$(13)\quad X = \sum_{1}^{MNG} (S_{old} - S_{new})$$

If X is negative the reactor is supercritical and the size must be decreased. If X is positive the size must be increased. The size is increased or decreased by 50% until X changes sign. A binary chop is used on succeeding iterations until the critical convergence criterion is met.

After a new source is calculated, the flux is modified by equations (3) → (6) until the distribution is correct for the given source. The convergence of the flux after succeeding mesh sweeps is given by

$$(14)\quad \sum_{1}^{MNG} \frac{\left| \phi^{(p)}(r,z,g) - \phi^{(p-1)}(r,z,g) \right|}{MNG} < \epsilon$$

where: $\phi^{(p)}(r,z,g)$ = neutron flux at the (r,z) mesh point in the g energy group on the p^{th} iteration

(o^{n1}/cm^2-sec)

$\phi^{(p-1)}(r,z,g)$ = neutron flux at the same point on the previous iteration

(o^{n1}/cm^2-sec)

ϵ = input specified convergence criterion

MNG = total active flux points in the (M,N,G) mesh.

RESULTS OF DEMONSTRATION PROBLEM

To demonstrate the ability of the code to solve reactor criticality problems a sample problem was run using the date from Yiftah (4) for ZPR-III, Assembly No. 10. The energy spectrum and sixteen group cross sections were taken from the same text.

Flux Distribution

The code was designed to handle arbitrary external geometric boundaries. This feature was checked by selecting a mesh configuration that would approximate a spherical reactor (see figure 2). With this spherical boundary the flux should, after sufficient flux mesh sweeps, have the same radial distribution in all directions. Figure 3 shows a flux plot along the r, z and 45° axes. This nearly identical flux distribution is considered an adequate demonstration of the ability of the code to handle arbitrary geometric boundaries.

In a hard spectrum reactor such as ZPR-III, flux peaking should occur in the fifth energy group (0.5 → 0.825 Mev). A plot of group flux distribution (Figure 4) shows this peak in the output from the code. The FAIM-6 distribution is shown in Figure 6. The sixteen group calculation with a spherical reactor approximation proved too time consuming for testing the code. Consequently, a nine-group set was developed by adding the energy weighted cross sections together for the last eight of the original sixteen groups.

The reactor mesh was changed from the spherical approximation to a thirteen by thirteen mesh cylinder (Figure 5). Using this procedure

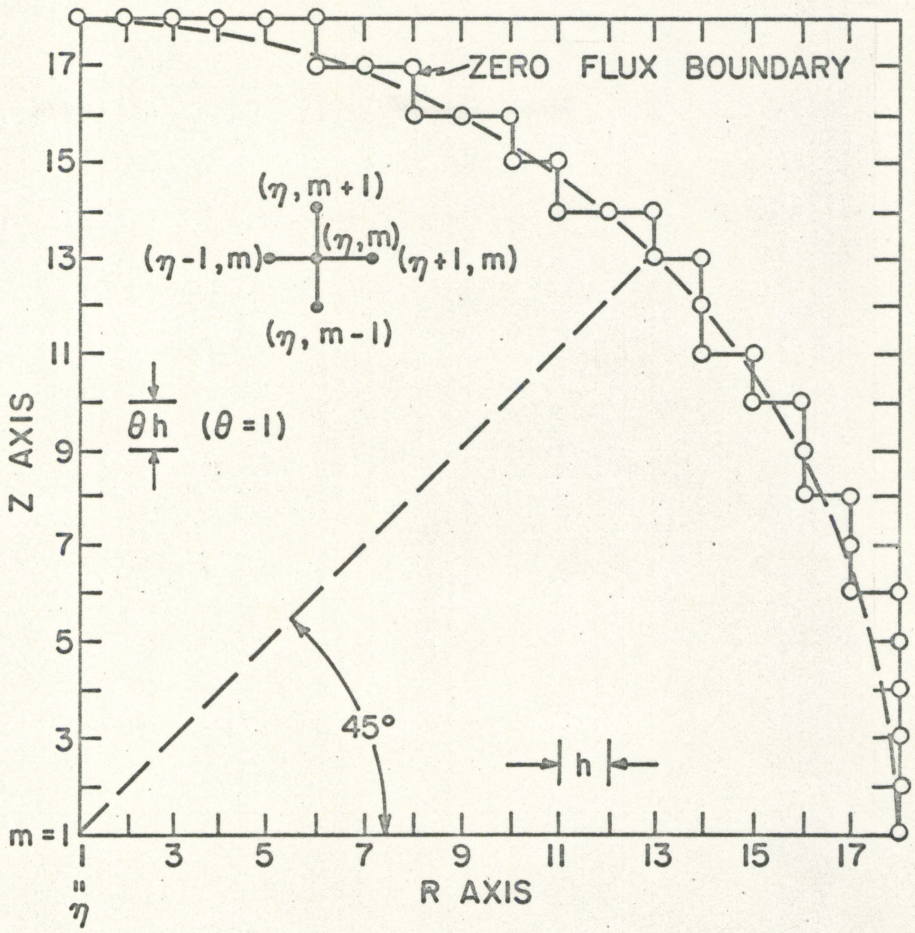

Figure 2. Mesh layout for approximation of spherical reactor

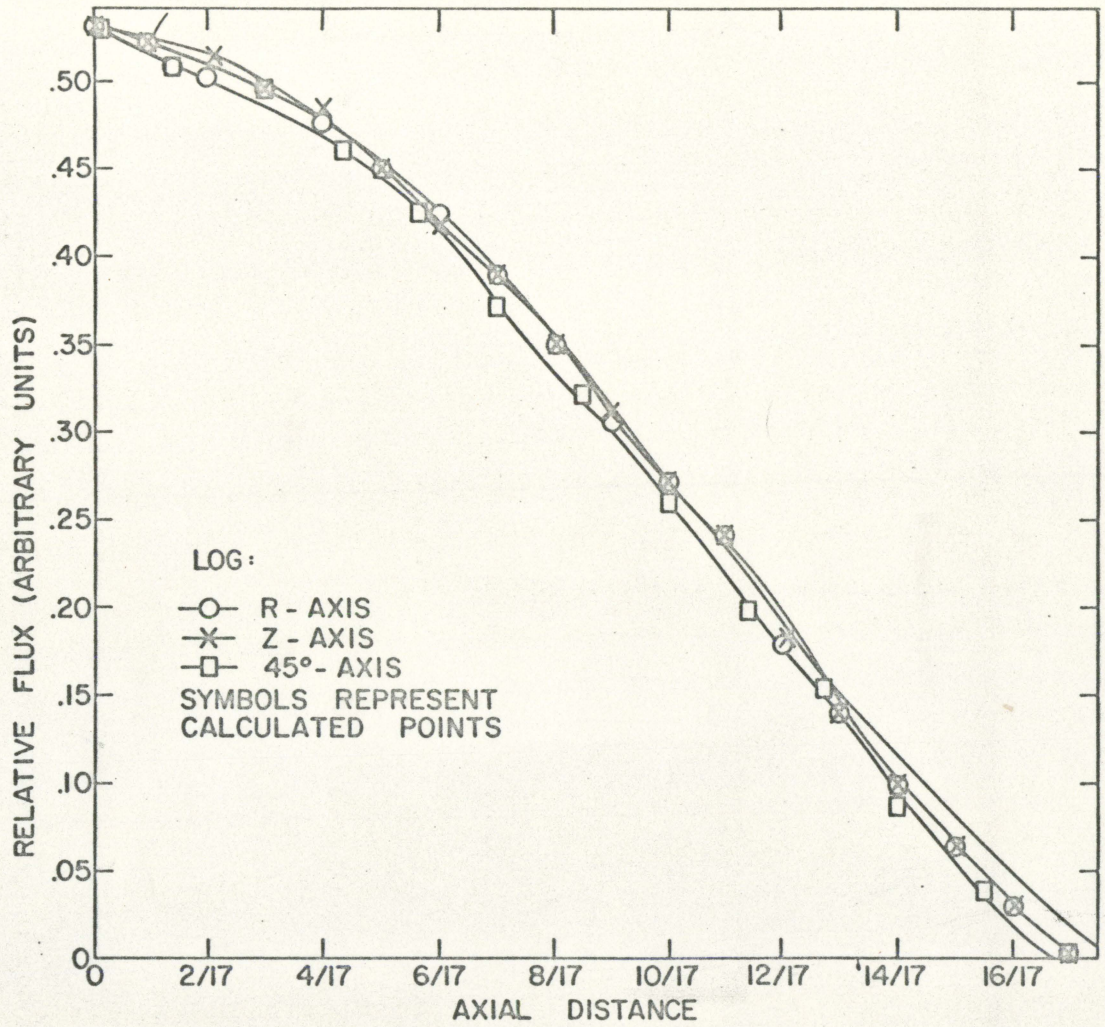

Figure 3. Comparison of flux distribution for sphere (SPR-111 Assembly 18)

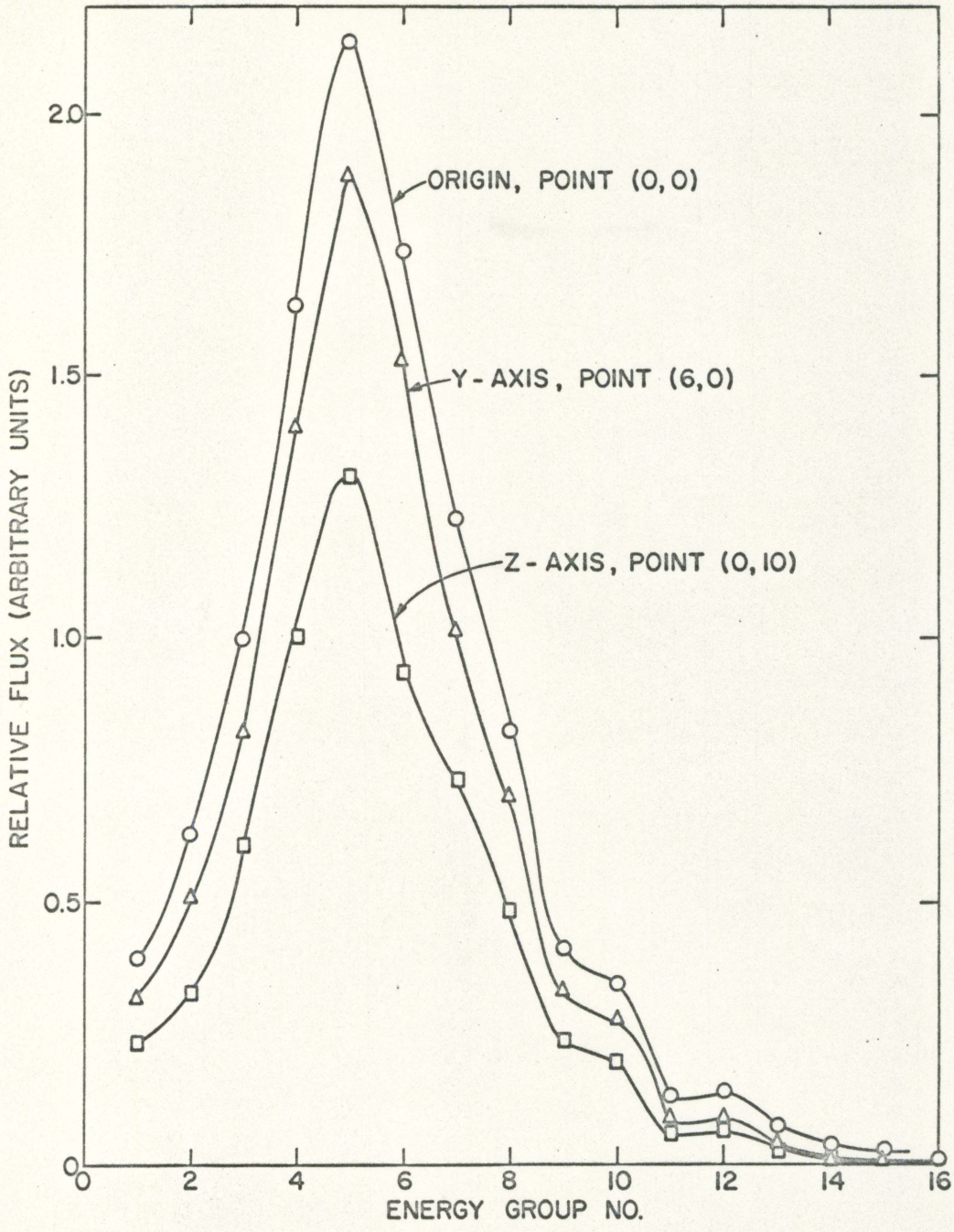

Figure 4. FABRA flux distribution for ZPR-III, Assembly No. 10

The total number of mesh points, and hence the time for one iteration, was cut by a factor of two. The energy flux distribution for this configuration is shown in Figure 7.

Convergence

To complete a problem two convergence criterion must be satisfied. These are expressed in equation (14), p. 15, for flux convergence and equation (12), p. 14, for the source convergence. The source convergence implies criticality and terminates the problem. The source iteration process used to arrive at convergence is known as the outer iteration. The flux convergence process using the finite difference equations is known as the inner iteration. The flux must converge on every iteration while the source convergence need be attained only once. Flux convergence with the finite difference equations is illustrated in Figure 8. Each iteration or mesh sweep with 16 groups and 324 mesh points, takes approximately 4 minutes. Convergence of the flux from the initial flux distribution was attained after twenty-seven iterations. After the source was corrected to the new distribution, convergence was attained in twenty iterations.

The source convergence depends upon finding the critical size for the reactor. A bindary drop is used on the mesh size (h) of the reactor to find the critical size. Criticality is determined by equation (13), p. 14.

Criticality

The FABRA Code, using the data from ZPR-III, Assembly No. 10, predicts a critical size of at least 36 cm radius and 454 kg of U-235.

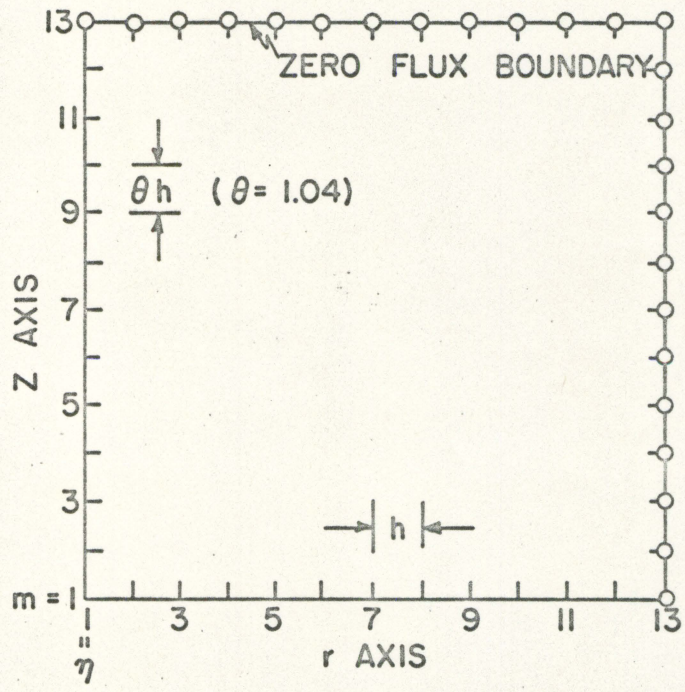

Figure 5. Cylindrical mesh layout for nine group calculation

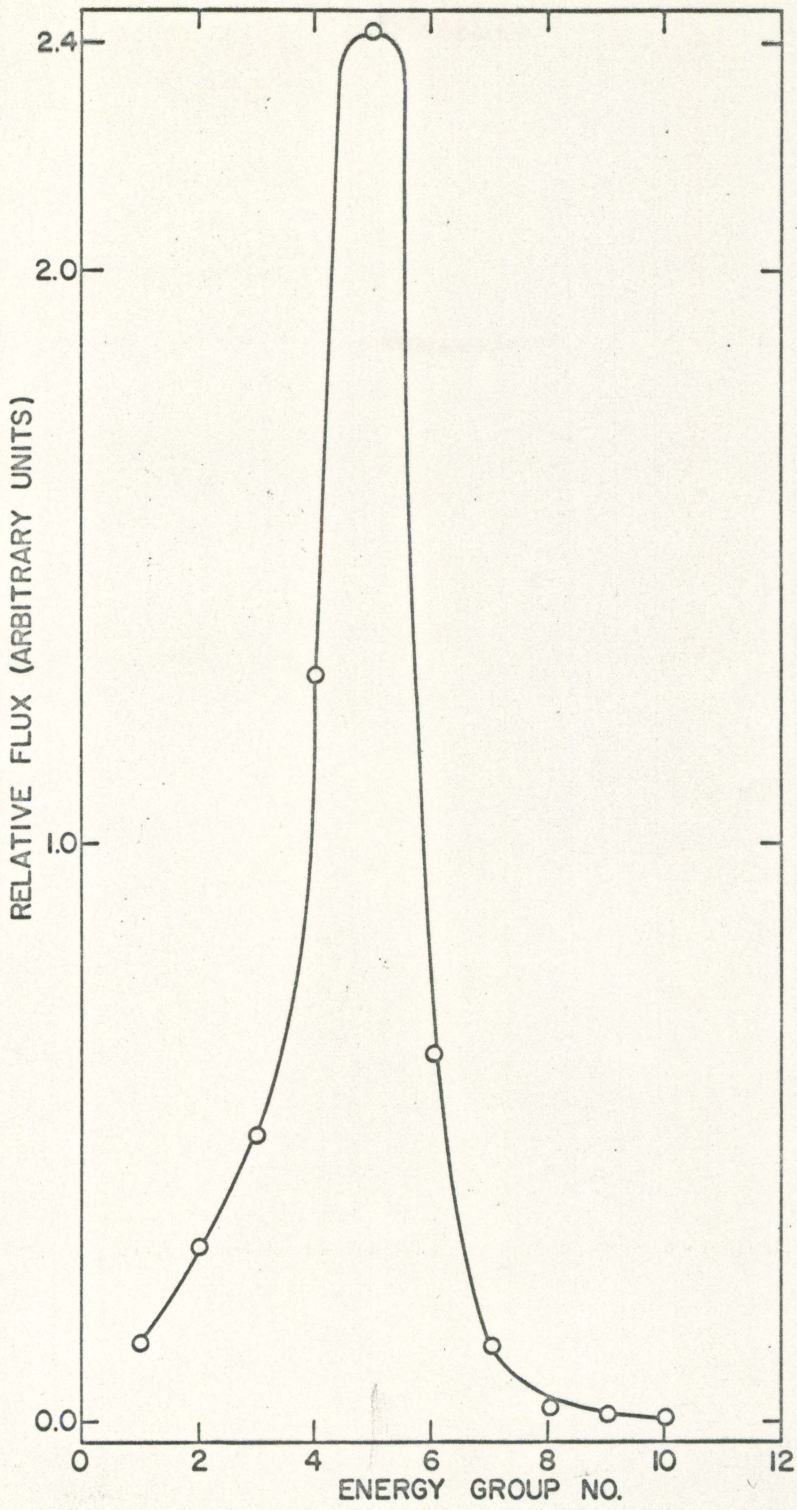

Figure 6. FAIM 6 Flux distribution for ZPR-III, Assembly No. 10

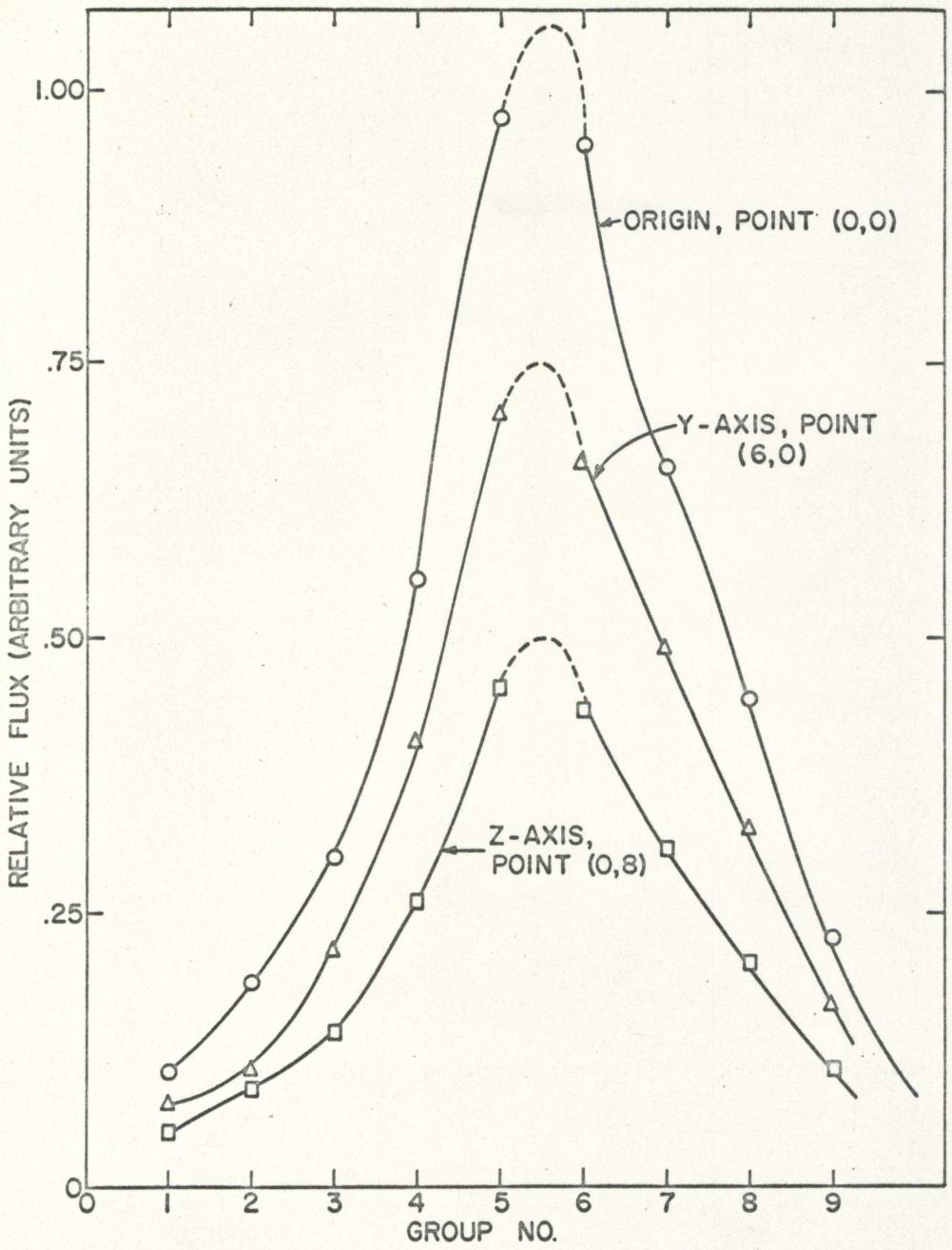

Figure 7. FABRA nine-group flux distribution for ZPR-III; Assembly No. 10

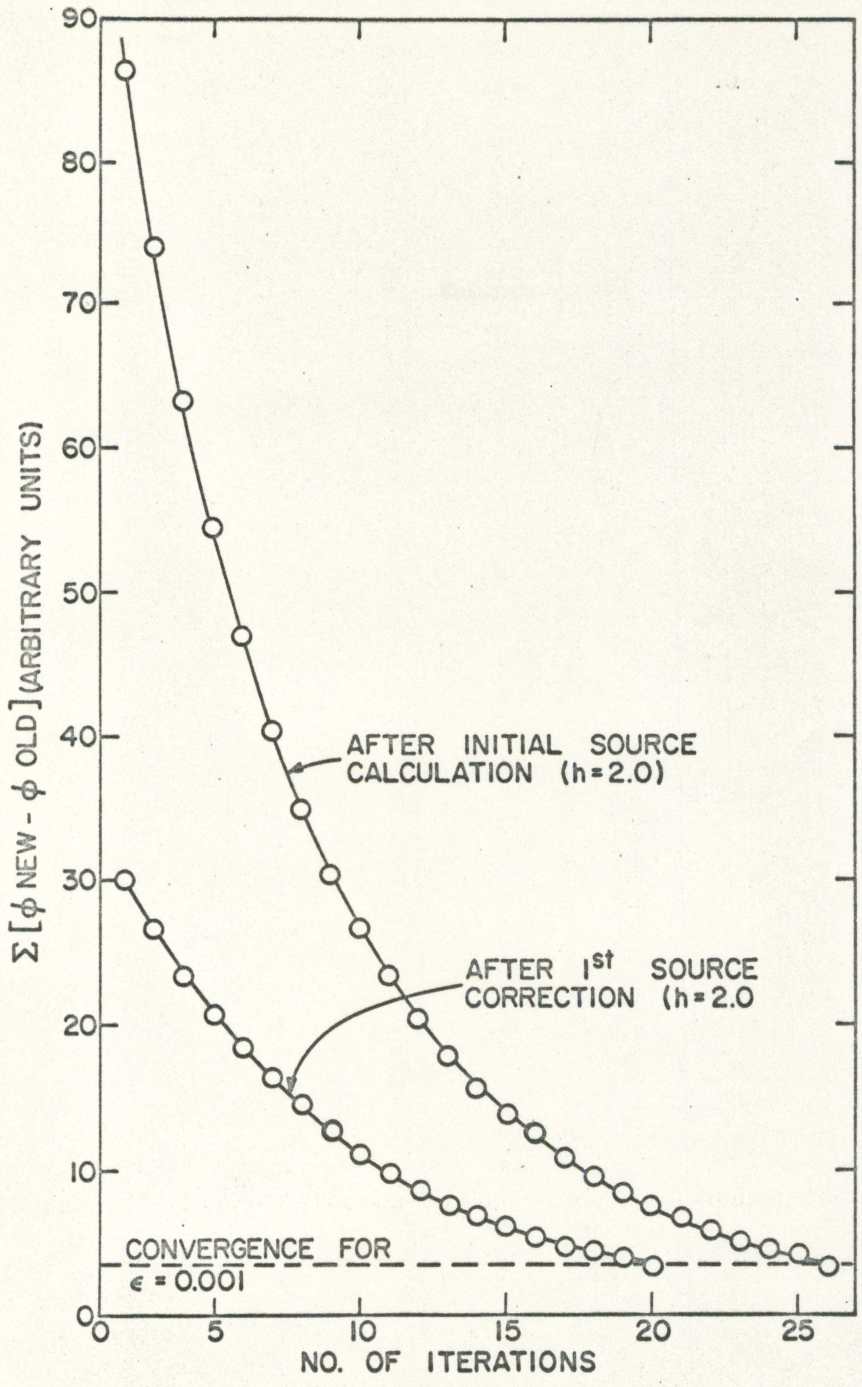

Figure 8. FABRA flux convergence for EFR-III, Assembly No. 10

The calculation on the Cyclone was terminated after 3 hrs. because the experimental critical size was 25.6 cm radius and 156 kg U-235. However, a later check using the FAIM* neutron diffusion code predicted a critical size of 39.5 cm radius and 574 kg of U-235. The difference between the experimental and calculated results is due to the 50 cm blanket of natural uranium surrounding the actual core (9). When the calculation is made with the blanket using the FAIM code the calculated radius is in excellent agreement with the experimental critical radius (Reference 10, Table II). The calculation with the blanket was made with an earlier version of FAIM, but the changes in the program to the present time have been superficial.

*(8) The FAIM Code, a multigroup, one-dimensional diffusion equation code by D. C. Baller, AMTD-118, Atomics International. January 1962.

CONCLUSIONS

The FABRA Code will predict the flux distribution and criticality of a fast reactor. The flux distribution compares favorably with that calculated by the FAIM Code (Figures 4 (Origin) and Figure 5). Differences in the flux distribution after the fifth energy group can be explained by the fact that the outer iteration was not converged at the time the flux distribution was printed out. In the example shown, the flux distribution was assumed to be linearly degenerated through the energy groups. This required a greater readjustment in the flux distribution than if the assumption is made of sine-squared distribution as in the nine group case shown in Figure 7.

The ZPR-III Assembly No. 10 reactor was predicted slightly subcritical at 36 cm with no blanket by the FABRA Code. This was confirmed by the FAIM Code calculation which predicted a critical size of 39.5 cm for the bare reactor.

The ability of the code to handle various geometric boundaries is demonstrated in Figure 3. The flux distribution for a spherical approximation was very similar in all directions. Hence, the code was able to predict spherical symmetry using cylindrical finite difference equations and a specially defined boundary.

RECOMMENDATIONS

The FABRA Code was designed to be a flexible entity. The major sections of the code can be removed, modified in any way, and replaced without affecting the other sections. In this light the following recommendations are made for using and improving the code over the next several years to obtain maximum benefit from its development.

Code Utilization

Due to the inherent flexibility of the FABRA Code built into it by the SAR programming technique this code can be used both for reactor studies and for improving coding techniques. In its present form the code can be utilized to make singular or parametric studies of flux distribution and criticality of fast reactors. Various fuels, poisons and geometric shapes may be chosen. In addition to this however, coding techniques may be studied that can lead to improvements in reactor simulation. Computer studies are limited by the time and memory space necessary to solve a problem. The Cyclone Computer, for example, has 16K memory units available for computation. This limits the number of mesh points that can be calculated since two memory locations are required for each mesh point. It would be useful to determine the optimum way to use these available memory units.

The following questions may be asked by way of example to indicate possible studies that might be made with the code. Would it be more valuable to use sixteen neutron energy groups and three hundred space mesh points instead of eight neutron energy groups and six hundred

space mesh points? It is better to sweep the mesh from top to bottom or in random suit rather than bottom to top? These questions can be answered by small modifications in one section of the present program while holding other sections the same.

Code Improvements

The following things could be done to the present version of the FABRA Code to contemporize it with other similar codes.

1. Use a floating rather than fixed point input routine.
2. Incorporate some form of source extrapolation technique to speed convergence in the outer iteration.
3. Add options to search on other parameters than critical radius.
4. Incorporate a subroutine in the program to allow more than one region in the reactor.

ACKNOWLEDGEMENTS

I wish to thank Dr. Glenn Murphy for his cooperation and assistance in the work that was done for this thesis. I would also like to thank Robert Sharpe of the Cyclone Computer Laboratory for his help in coding the multigroup problem.

Special thanks to Frederick Graham who in my absence from campus, supervised the preparation of this thesis in its final form. I would also like to thank Linda Harrington who struggled to type this text with no author available to discuss the many problems that arise in the preparation of a technical document.

BIBLIOGRAPHY

1. Wachspress, E. L. CURE: a generalized two-space-dimension multigroup coding for IBM-704. U.S. Atomic Energy Commission Report KAPL-1724 [Knolls Atomic Power Lab., Schenectady, N.Y.]. 1957.

2. Stone, Stuart P. A two dimensional multigroup neutron diffusion theory reactor code for the IBM-709 or 7090 (9-ANGIE). U.S. Atomic Energy Commission Report UCRL-6076 [University of California Radiation Lab., Berkeley]. 1960.

3. Okrent, David. Multigroup calculations. In Etherington, Harold, ed. Nuclear engineering handbook. pp. 6-88 to 6-89. McGraw-Hill, New York. 1958.

4. Yiftah, S., Okrent, D., and Moldauer, P.A. Fast reactor cross sections. Pergamon Press, London. 1960.

5. Roe, G. M. Adaptation of multigroup methods to cylindrical geometries. U.S. Atomic Energy Commission Report KAPL-950 [Knolls Atomic Power Lab., N. Y.] 1954.

6. Glasstone, Samuel and Edlund, Milton C. The elements of nuclear reactor theory. Van Nostrand, N. Y. 1952.

7. Weinberg, Alvin M. and Wigner, Eugene P. The physical theory of neutron chain reactions. University of Chicago Press, Chicago. 1958.

8. Baller, C. D. The FAIM Code: a multigroup one-dimensional diffusion equation code. North American Aviation, Inc. Atomics International Division Applied Mathematics Technical Document No. 118. 1962.

9. Long, J. K. and Lowenstein, W. B. Fast neutron power reactor studies with SPR-III. In Internation Conference on the Peaceful Uses of Atomic Energy, 2nd, Geneva, 1958. Proceedings 12: 119-121. 1958.

10. Campise, A. V. Analytical techniques for interpreting AETR critical experiments. In U.S. Atomic Energy Commission Report AI-6998 [Atomics International, Canoga Park, California] pp. 62-66. 1962.

APPENDIX A

Code Input

The input required for the code is shown in Table A 1. Since the code input routine requires non-integer numbers to be less than 1.0 some numbers must be scaled before they can be input to the machine. The required scaling is listed in Table A 2. A sample data input sheet is shown in Figure A 1 with zeros input as plus signs (+).

Table A 1. Order of input (second correction)

Order of Input	Symbol	Memory Locations	Max. No Used	Description
1	G_1	1991	1	No. of Groups to 16
2	SG	1992	1	No. of Subgroups to 10
3	MAT	1993	1	No. of Materials to 6
4	REG	1994	1	No. of Regions to 6
5	h	1995	1	Basic Mesh Spacing (r) for
6	θ	1996		variable z spacing
7	M	1997	1	Where M = Reactor Height/2
8	N	1998	1	Where N = Reactor Radius
9	ϵ	1999	1	Flux Iteration Parameter
10	ξ	2000	1	Source Iteration Parameter
11	1/3	2001	1	
12	30/48	2002	1	
13		2003	1	
14		2004	1	(2003 through 2010 are zero)
15		2005	1	
16		2006	1	
17		2007	1	
18		2008	1	
19		2009	1	
20		2010	1	
21	Temp	2011-2026	16	
22	χ	2027-2042	16	Fission Spectrum

Table A 1. Order of input (second correction) (continued)

Order of Input	Symbol	Memory Locations	Max. No. Used	Description
23	N	2043-2048	6	No. Atoms/CM^3 in each material
24	C	2049-2084	36	Fraction of (N) in Region (R) Order N_1, R_1, N_2, R_2 ----------
25	ν	2085-2180	96	Neutrons/Fission for each Group Stored One Material at a Time
26	a_f	2181-2196	16	Fission C.S. for Material 1
	$\sigma_{N\gamma}$	2197-2212	16	Capture C.S. for Material 1
	σ_{er}	2213-2228	16	Elastic C.S. for Material 1
	σ_{er}	2229-2244	16	Transport C.S. for Material 1
	σ_{n,n^1}	2245-2254	10	Total Inelastic C.S. for Material 1
	$\sigma_{n,(o)}$	2255-2264	10	Partial Inelastic C.S. for Material
	$\sigma_{n(1)}$	2265-2274	10	From j^{th} froup to group j+k
	$\sigma_n(2)$	2275-2284	10	k = 1, 2 -------- 6
	$\sigma_n(3)$	2285-2294	10	
	$\sigma_n(4)$	2295-2304	10	
	$\sigma_n(5)$	2305-2314	10	
	$\sigma_n(6)$	2315-2324	10	
		TOTAL	144	144 C.S. for Each Material
27	σ_2	2325-2468	144	C.S. (Same as 26) for Material 2
28	σ_3	2469-2612	144	C.S. (Same as 26) for Material 3
29	σ_4	2613-2756	144	C.S. (Same as 26) for Material 4

Table A 1. Order of input (second correction) (continued)

Order of Input	Symbol	Memory Locations	Max. No. Used	Description
30	σ_5	2657-2900	144	C.S. (Same as 26) for Material 5
31	σ_6	2901-3044	144	C.S. (Same as 26) for Material 6
32	BC	(Not used)	48	Flags to indicate Boundaries Between Two or More Regions
33	BML	3045-3144	299	Boundary Memory Locations
33a	40	(Last BML)	1	(i.e. Flag Storage)
34	ZBML	(Not used)	99	Zero Boundary Locations
35				
36	n_1 n_{25}	3145-3162	25	n values to zero boundary m = 1 through m = 25

Table A 2. Input scaling

1. G, SG, MAT., REG, M, N, BML, n_1 n_m to be input as integers. No scaling required.
2. 1/3, 30/36, X, C, N ($x10^{24}$), BC; are already in fractional form and may be input as is. No scaling required.
3. $\Delta\mu$, if used, must be input as $\Delta\mu/2$.
4. The largest No. ν(NU) that will ever be input is < 5.0. $\therefore$ ν may be entered as $\nu/8$ or $\nu/2^3$.
5. The largest No. for any cross section ($\sigma x10^{-24}$) that will be allowed for this program is < 256. $\therefore \sigma$ may be entered as $\sigma/256$ which is $\sigma/2^8$.
6. For the present program it will be adequate to assume that $\sigma < 25$. $\therefore$ σ will be entered as $\sigma/32$ which is $\sigma/2^5$. The smallest value than that could be found for σ was 0.0001 ($x10^{-24}$).

$$\sigma \text{ min} = \frac{.0001}{32} = 0.00003125 \text{ or } 3.125\ (10)^{-6}$$

which is within the range of the computer.

7. h should never exceed 16 cm (for 9 ft radius reactor). $\therefore$ h should be entered as h/16 or $h/2^4$ (18 mesh point in r direction).
8. θ cannot exceed 4. Therefore, $\theta/4$ is entered.

9 GROUP ZPR-III ASSEMBLY NO. 10 DATA

h = 2.0 cm

θ = 1.04

ϵ = .0004

= .0001

MESH: 13X13

CONFIGURATION: CYLINDER

SIMON FABRA SEPTEMBER 13, 1962

+9+8+5+1;+125+0325;+13+13;+0004+0001+333333333+625+++++++++;+;
+132+213+232+179+116+067+034+027;

+0193125+0191625+0134375+00075+++++
+0003125+0006875+0017812 5+0040312 5+0043437 5+0039687 5+0050312 5+0085451812+
0189130907

+0001812 5+0000312 5+0000531 25+0000937 5+0001312 5+0001468 75+000187 5+00028084
7738+000475017
+001625+0023125+002375+0031875+0047187 5+0059375+004625+007370087+

+0000468 75+0000781 25+0001406 25+0002062 5+0002312 5+0002812 5+000353887042+
0010639
+0019062 5+0025937 5+004187 5+0055+0060625+0073437 5+009437 5+0116749596+

+000021875+0000406 25+000037 5+000012 5+0000218 75+00002 5+000318467336+
000767028
+0017812 5+003+0032187 5+0050312 5+0050937 5+005937 5+00+009359947+

APPENDIX B

Code Output

The code prints out all the calculated macroscopic cross-sections and certain input parameters to assure correct data input. After the core calculation, the critical dimensions and the flux distribution are printed out. A sample output sheet is included for illustration.

Sample Output Sheet

D

$.120469_{10}$ + 01

$.173031_{10}$ + 01

Σa

$.129167_{10}$ + 00

$.126675_{10}$ + 00

Σer

$.145538_{10}$ - 02

$.212981_{10}$ - 02

Σin

$.103440_{10}$ + 00

$.100230_{10}$ + 00

xvΣf

$.100110_{10}$ - 01

$.886788_{10}$ - 02

Final Mesh Spacing = 1.35 cm

Critical Radius = 25.8 cm

Critical Height = 25.8 cm

Final Flux Values

Group 01

$.9918_{10}$ + 00 $.9785_{10}$ + 00

$.7900_{10}$ + 00 $.7313_{10}$ + 00

Group 02

$.9298_{10} + 00$ $.9173_{10} + 00$

$.7406_{10} + 00$ $.6856_{10} + 00$

Group 03

$.8717_{10} + 00$ $.8600_{10} + 00$

$.6943_{10} + 00$ $.6428_{10} + 00$

Group 04

$.8172 + 00$ $.8063_{10} + 00$

$.6509_{10} + 00$ $.6026_{10} + 00$

Group 05

$.7661_{10} + 00$ $.7559_{10} + 00$

$.6102_{10} + 00$ $.5649_{10} + 00$

Group 06

$.7182_{10} + 00$ $.7086_{10} + 00$

$.5721_{10} + 00$ $.5296_{10} + 00$

Group 07

$.6733_{10} + 00$ $.6643_{10} + 00$

$.5363_{10} + 00$ $.4965_{10} + 00$

Group 08

$.6313_{10} + 00$ $.6228_{10} + 00$

$.5028_{10} + 00$ $.4655_{10} + 00$

Group 09

$.5918_{10}$ + 00 $.5839_{10}$ + 00

$.4714_{10}$ + 00 $.4364_{10}$ + 00

Group 0+

$.5548_{10}$ + 00 $.5474_{10}$ + 00

$.4419_{10}$ + 00 $.4091_{10}$ + 00

Group 0-

$.5201_{10}$ + 00 $.5132_{10}$ + 00

$.4143_{10}$ + 00 $.3836_{10}$ + 00

Group 0;

$.4876_{10}$ + 00 $.4811_{10}$ + 00

$.3884_{10}$ + 00 $.3596_{10}$ + 00

Group 0,

$.4572_{10}$ + 00 $.4510_{10}$ + 00

$.3641_{10}$ + 00 $.3371_{10}$ + 00

Group 0.

$.4286_{10}$ + 00 $.4228_{10}$ + 00

$.3414_{10}$ + 00 $.3160_{10}$ + 00

Group 0_{10}

$.4018_{10}$ + 00 $.3964_{10}$ + 00

$.3200_{10}$ + 00 $.2963_{10}$ + 00

Group 10

$.3767_{10}$ + 00 $.3716_{10}$ + 00

$.3000_{10}$ + 00 $.2778_{10}$ + 00

APPENDIX C

Code Instructions

Sample sections of the code it its most recent form are presented on the following pages. The steps listed in Table I are related to the code in the following list.

*STEP FOR SOLUTION	SECTION OF THE CODE IN WHICH THE CALCULATION IS MADE
1	FABRA PRELIM
2	FABRA PRELIM
3	IFLUX AND FLOAT
4	SOURCE FABRA
5	FLUX FABRA
6	FLUX ITERATION DECISION
7	SOURCE FABRA
8	INCH
9	INCH
10	FINI FABRA

*See Table I in the text

2nd CORR. PRELIMINARY PACKAGE

FABRA SIMON

ASSEMBLED: AUGUST 27, 1962

CONTAINS

1. FABRA PRELIM (11th Tape V) 7-3-62
   Add. Corrections:

   OUT 1: 106
   LDQ 3045;

2. FLOAT & PRINT INPUT (2nd Corr.) 7-10-62
   Add. Corr.

   DDO + 9: LDQ EXPFR

3. INFLUX & FLOAT (6th Corr.) 8-20-62

4. N12a (1st Corr.) 7-3-62

5. COS & JOX (4th Corr.) 7-9-62

6. FLOATING ARITH & OUT (Copy) 7-5-62

```
35:     jldq    1991;
        uldq    *;

        ubl     n12a;
        nop     ;

        ldq     1995;
        uldq    *;

        ubl     n12a;
        nop     ;

        jldq    1997;
        uldq    *;

        ubl     n12a;
        nop     ;

        ldq     1999;
        uldq    *;
```

```
ubl     n12a;
nop     ;

ldq     2011;
uldq    *;

ulb     n12a;
nop     ;

ldq     2027;
uldq    *;

ubl     n12a;
nop     ;

ldq     2043;
uldq    *;

ubl     n12a;
nop     ;

ldq     2049;
uldq    *;

ubl     n12a;
nop     ;

ldq     2085;
uldq    *;

ubl     n12a;
nop     ;

ldq     2181;
uldq    *;

ubl     n12a;
nop     ;

jldq    3045;
uldq    *;

ulb     n12a;
nop     ;

jldq    3145;
uldq    *;
```

```
        ubl     n12a;
        nop     ;

        za      1991;
        sli     2;

        sto     2005;
        za      1992;

        sli     3;
        sto     2006;

        a       2005;
        sto     2007;

ca:     begin   *;

        nop     ;
        sli     1;

        nop     ;
        sli     1;

        nop     ;
        ldq     2181;

        nop     ;
        rmp     2043;

        nop     ;
        usto    3481;

        xa      ?+1;
        s       2007;
        blp     *+4;
        zia     ?+1;

        sto     ?+1;
        zia     ?+2;

        sto     ?+2;
        zia     ?2+;

        sto     ?+4;
        br      ?+2;

        za      ?;
        s       1993;
```

```
brp     di+3;
zia     ?+1;

s       2007;
sto     ?+1;

zia     ?;
sto     ?;

zia     ?2+;
sto     ?+2;

zia     ?+3;
sto     ?+3;

zia     ?+4;
sto     ?+4;

bl      ?+2;
nop     ;
```

12th CORR. SOURCE FABRA

SIMON AUGUST 25, 1962

```
ISRCZ:  begin   *;

        +1;             zero the source

PISR:   ldq     zroe;   compute total active mesh points (tmng)
        stoq    tmng;

PISRI:  z       ;
        a       3145;

        a       mone;
        a       tmng;

        sto     tmng;
        za      c13;

        s       1997;
        blp     pisr2;

        zia     c13;
        sto     c13;

        zia     pisr1;
        str     pisr1;

        bl      pisr1;
        nop     ;

PISR2:  zia     c13;
        s       1997;

        sto     c13;
        zia     pisr1;

        s       1997;
        str     pisr1;

        ldq     tmng;
        zmp     1991;

        stoq    Σman;
        ldq     expin;

        stoq    Σexp;
        zia     *;
```

```
        bl      fnorz;
        sto     tmng;

        nop     ;
        uldq    5878;

        za      *-1;
        a       mng;

        stu     *+1;
        nop     ;

A1:     ldq     zroe;
        ustoq   ;

        za      isrcz;
        s       mng;

        blp     sourc;
        zia     isrcz;

        stu     isrcz;
        zia     a1;

        stu     a1;
        br      a1;
```

4th CORR. FLUX FABRA

SIMON AUGUST, 21, 1962

```
FLUX:   phcr    1;
        phcr    1;

        phuc    1;
        phsp    1;'

        phsp    1;
        phf     1;

        ph1     1;
        phu     1;

        phx     1;
        ph1c    1;

        phcr    1;
        phcr    1;

        zia     t133;
        str     t133;

        za      t133;
        ph4a    10;

        phcr    1;
        za      t121;

        ph4a    10;
        phcr    1;

        zia     zroe;
        str     cm70;

        str     cn71;
        str     cgp73;

        str     cg75
        str     cdi76;

        za      zroe;
        sto     t132;

        sto     crg74;
        nop     ;
```

```
        ldq     2015;       initial calculation
        zia     *;

        bl      fcla;
        mop     ;

        ldq     2015;
        zia     *;

        bl      fmul;       hsqd
        sto     hsqd

        ldq     quar4;
        zia     *;

        bl      fcla;
        nop     ;

        ldq     2016;
        zia     *;

        bl      fdiv;
        nop     ;

        ldq     2016;
        zia     *;

        bl      fdiv;
        sto     ithsq;      ithsq

P:      nop     ;           check type flux
        uldq    5878;

        stoq    t11;
        zmsq    ;

        blp     incp;
        za      p;

        a       mng;
        stu     ts;

        nop     ;
        zs      cn71;

        sto     2008;
        zism    2008;
```

```
        blp     p1;
        zs      cm70;

        sto     2008;
        zism    2008;

        blp     aw;
        bl      u;

P1:     zs      cm70;
        sto     2008;

        zism    2008;
        blp     x;

        bl      av;
        nop     ;
```

8th CORR. FLUX ITERATION DECISION

FABRA SIMON SEPTEMBER 5, 1962

```
N:      ldq     t132;
        zia     *;

        bl      fcla;       Σ ∅ - ∅o
        nop     ;

        ldq     tmng;
        zia     *;

        bl      fdiv;
        nop     ;

        ldq     2019;
        zia     *;

        bl      fsub;       Σ ∅n - ∅o  / TMNG - EPSILON
        blp     *+2;

        bl      nl;
        hbl     nl;

        phcr    1;
        phur    1;

        phn     1;
        pho     1;

        phsp    1;
        phc     1;
        pho     1;
        phn     1;
        phv     1;
        phe     1;
        phr     1;
        phg     1;
        phe     1;
        phn     1;
        phc     1;
        phe     1;

        phlc    1;
        phcr    1;

        uldq    5878;
        zia     *;
```

```
bl      fout;
nop     ;

uldq    5879
zia     *;

bl      fout;
phsp    1;

ph8r    1025;
phbks   1;

phuc    1;
ph8r    673;

ph1c    1;
ph8r    17;

ph8r    17;
phuc    1;

ph8r    1265;
ph1c    1;

ph8r    17;
phsp    1;

pha     1;
phn     1;
phd     1;
phsp    1;

ph8r    1025;
phbks   1;

phuc    1;
ph8r    673;

ph1c    1;
ph8r    33;

ph8r    17;
phuc    1;

ph8r    1265;
ph1c    1;

ph8r    17;
phcr    1;
```

```
phcr    1;
phuc    1;

ph8r    129;
ph8r    33;

pho     1;
phbks   1;

ph8r    673;
phlc    1;

phn     1;
phsp    1;

ph8r    177;
phsp    1;

phuc    1;
pho     1;

phbks   1;
ph8r    673;

phlc    1;
pho     1;

phuc    1;
ph8r    33;

phsp    1;
ph8r    177;

phlc    1;
nop     ;

ldq     zroe;
stoq    formt;

ldq     t132;
zia     *;

bl      fout;
phcr    1;

bl      flux;
hbl     flux;
```

```
N1:     za      zroe;
        sto     t133;

        bl      sourc;
        nbl     sourc;

NINE:   +12;

Δ h:
```

4th. CORR. FINI FABRA

SIMON AUGUST 23, 1962

```
FINI:   phcr    2;
        phuc    1;

        phf     1;
        phi     1;
        phn     1;
        pha     1;
        phl     1;
        phsp    1;

        phm     1;
        phe     1;
        phs     1;
        phn     1;

        phsp    1;
        phs     1;
        php     1;
        pha     1;
        phc     1;
        phi     1;
        phn     1;
        phg     1;

        phsp    7;
        phc     1;
        phr     1;
        phi     1;
        pht     1;
        phi     1;
        phc     1;
        pha     1;
        phl     1;
        phsp    1;

        phr     1;
        pha     1;
        phd     1;
        phi     1;
        phu     1;
        phs     1;

        phsp    9;
        phc     1;
        phr     1;
```

```
phi     1;
pht     1;
phi     1;
phc     1;
pha     1;
phl     1;
phsp    1;

phh     1;
phe     1;
phi     1;
phg     1;
phh     1;
pht     1;

phcr    1;
phsp    1;

phh     1;
ph8r    177;

phlc    1;
ldq     out14;

stoq    formt;
nop     ;

ldq     2105;
zia     *;

bl      fout;
phsp    11;

phuc    1;
phr     1;

ph8r    177;
phlc    1;

ldq     2018;
zia     *;

bl      fcla;
nop     ;

ldq     2uar4;
zia     *;

bl      fsub;
nop     ;
```

```
ldq     2015;
zia     *;

bl      fmul;
sto     2008;

ldq     2008;
zia     *;

bl      fout;
phsp    10;

phuc    1;
phz     1;

ph8r    673;
phlc    1;

ph8r    33;
phuc    1;

ph8r    177;
phlc    1;

;dq     2017;
zia     *;

bl      fcla;
nop     ;

ldq     quar4;;
zia     *;

bl      fsub;
nop     ;

ldq     2016;
zia     *;

bl      fmul;
nop     ;

ldq     2015;
zia     *;

bl      fmul;
sto     2008;

ldq     2008;
zia     *;
```

```
        bl      fout;
        phcr    3;

        phtab   3;
        phuc    1;

        phf     1;
        phi     1;
        phn     1;
        pha     1;
        phl     1;
        pjsp    1;

        phf     1;
        phl     1;
        phu     1;
        phx     1;

        phsp    1;
        phv     1;
        pha     1;
        phl     1;
        phu     1;
        phe     1;
FINI1:  phs     1;
        phcr    1;

        phsp    3;
        phuc    1;

        phg     1;
        phr     1;
        pho     1;
        phu     1;
php     php     1;
        phlc    1;

        phsp    1;
        ldq     cg75;

        src     8;
        ph4a    2;

        nop     ;
        phcr    1;

FINIS2: phcr    1;
        uldq    5878;
```

```
        zsmq    ;
        brp     *+1;

        br      fini5;
        phsp    2;

        ph8r    1025;
        br      fini6;

FINIS5: nop     ;
        zia     *;

FINIS6: bl      fout;
        za      cdi76;

        s       mn
        blp     fini3;

        zia     cdi76;
        str     cdi76;

        zia     fini2;
        stu     fini2;

        za      c15;
        s       six;

        blp     fini7;
        zia     c15;

        sto     c15;
        br      fini2;

FINI7:  zia     aroe;
        sto     c15;

        bl      fini2;
        nop     ;

FINI3;  zia     cdi76;
        s       mn;

        str     cdi76;
        nop     ;

        za      cg75
        s       1991;

        blp     fini4;
        zia     cg75;
```

```
        str     cg75;
        zia     fini2;

        stu     fini2;
        br      fini1;

FINI4:  zia     cg75;
        s       1991;

        str     cg75;
        zia     fini2;

        s       mng;
        stu     fini2;

        phcr    1;
        phe     1;

        phn     1;
        phd     1;

        phcr    1;
        phsc    1;

        phsc    1;
        hbl     35;

OUT14:  +104;

c15:    +1;
```