AN APPLICATION OF CHEBYSHEV POLYNOMIALS TO THE SOLUTION OF A TWO-DIMENSIONAL ELLIPTIC BOUNDARY-VALUE PROBLEM

by

Stephen V. Prewett

Dissertation submitted to the Graduate Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Nuclear Science and Engineering

APPROVED:

Milton C. Edlund, Chairman

Harold A. Kurstedt

R. Dean Riess

Ronald J. Onega

James R. Thomas

May 1978

Blacksburg, Virginia

ACKNOWLEDGMENTS

I would like to extend my thanks to the members of my committee for their guidance and help. Special thanks also go to my major professor, Dr. M. C. Edlund, for his kind and generous support on this thesis and throughout my graduate studies. Mrs. Lucy Terry is to be commended for her skill and patience in the typing of this thesis. Finally, I would like to thank my wife, Rebecca, without whose encouragement this effort would not have been possible.

MER/MRS 7/27/78

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF GRAPHS

CHAPTER I

INTRODUCTION

1.1 Goal of This Thesis

The goal of this thesis is to investigate the feasibility of using a bivarate Chebyshev polynomial to approximate solutions to the two-dimensional neutron diffusion equation. This is the first application of Chebyshev polynomials to obtain approximate solutions to the neutron diffusion equations by direct expansion of neutron fluxes in Chebyshev polynomials. They are used because of their simple form, economy of expansion, ease of manipulation, and earlier successful experience in the approximation of functions. Recently they have been used successfully in the solution of heat transfer problems.[1]

A fission reactor problem can be represented mathematically as an elliptic boundary value problem. The diffusion equations are $\nabla \cdot D_{ip}(\underline{r})\nabla\phi_{ip}(\underline{r}) - \sigma_{ipr}\phi_{ip}(\underline{r}) + S_{ip}(\underline{r}) = 0$ for $i = 1, \ldots, n$, where n is equal to the number of neutron groups used to span the energy range from 0 to 10 Mev. The boundary conditions are $\phi_{ip}(\underline{b}) = 0$ for all i, where $\underline{b}$ describes the surface of the reactor. Appendix I has a listing of the definitions of symbols and their subscripts. Of course, the neutron flux and current are continuous and finite within the reactor.

Finite difference methods have been used in the solution of these boundary value problems with success.[2] However, for two and three dimensional problems, computer time can become very large for the small mesh lengths usually required to obtain a reasonably accurate approximation to the solution of the continuous boundary value problem. Although finite difference methods are used in detailed design of reactor cores, they generally prove to be too expensive to use in conceptual design, where many parameters are varied to explore new concepts. Thus, the objective of this work is to develop a method which can be used for conceptual design of reactor cores.

1.2 Some Properties of Chebyshev Polynomials

Chebyshev polynomials were formulated in 1859 by the Russian mathematician, P. Chebyshev (1821-1894).[3] The Chebyshev polynomials are simply related to the trigonometric functions by the formula,

$$T_n(x) = \cos(n\theta), \quad x = \cos\theta \text{ and } n = 0, 1, 2, \ldots .$$

The recurrence relation for Chebyshev polynomials is

$$T_{n+2}(x) = 2xT_{n+1}(x) - T_n(x), \qquad -1 \le x \le 1.$$

The first few polynomials are:

$$T_0(x) = 1;$$

$$T_1(x) = x;$$

$$T_2(x) = 2x^2 - 1.$$

They are orthogonal with respect to the weight function $\frac{1}{\sqrt{1-x^2}}$ over the interval -1 to 1 and are normalized as follows:

$$\int_{-1}^{1} \frac{T_m(x)T_n(x)\,dx}{\sqrt{1 - x^2}} = \begin{cases} 0 & m \neq n; \\ \pi & m = n = 0; \\ \pi/2 & m = n \neq 0. \end{cases}$$

Mason used Chebyshev polynomials in a generalized "selected points" method of Lanczos' to solve the eigenvalue problem for a vibrating L-shaped membrane.[4] For this problem the work involved in computation was much the same as for finite difference methods of solving the problem. However, the method is easy to apply and eigenfunctions are obtained in a convenient form. The polynomials produced by this method are found to be partial sums of a double Chebyshev expansion.

In another application, Dew and Scraton (1973) used Chebyshev series to solve the heat equation in two-space variables.[5] This work served as the basis for a more general paper by the same authors (1975) using Chebyshev polynomials for the numerical solution of parabolic partial differential equations in two- and three-space variables.[6] They showed that approximate solutions to problems with more than one space variable can only be obtained in the form of Chebyshev polynomials by assuming separation of variables. The latter work of Dew and Scraton presents a more efficient algorithm which results in large savings in computer time compared to their 1973 method.

CHAPTER II

PROBLEM FORMULATION

2.1 Introduction

The first sections of this chapter present the algebraic aspects of the formulation of the two-group, two-dimensional (x,y) neutron diffusion boundary value problem in bivariate Chebyshev polynomials. The boundary conditions are then derived for an illustrative region of the test problem. Finally, a numerical method for solving the system of equations is presented, including a flow chart for the algorithm.

2.2 Two-Group Neutron Diffusion Problem

In two dimensions, the two-group neutron diffusion equations having constant coefficients within each physical region of the reactor, denoted by p, are:

Fast Group; (2.1)

$$[D_{1p}\nabla^2 - \sigma_{1pr}] \sum_{j=0}^{M} \sum_{k=0}^{M} C_{1pjk}T_{jk}(x,y) = \frac{\nu_{2p}\sigma_{2pf}}{\lambda} \sum_{j=0}^{M} \sum_{k=0}^{M} C_{2pjk}T_{jk}(x,y),$$

"Thermal" Group; (2.2)

$$[D_{2p}\nabla^2 - \sigma_{2pr}] \sum_{j=0}^{M} \sum_{k=0}^{M} C_{2pjk}T_{jk}(x,y) = \sigma_{1ps} \sum_{j=0}^{M} \sum_{k=0}^{M} C_{1pjk}T_{jk}(x,y),$$

where $$\phi_{ip}(x,y) = \sum_{j=0}^{M} \sum_{k=0}^{M} C_{ipjk}T_{jk}(x,y).$$

The reactor is assumed to be uniform within each region in the z direction. The model assumes that all fissions occur in the "thermal" group and all fission neutrons are produced as fast neutrons. The following boundary conditions are used: a) continuity of neutron fluxes and currents; b) neutron fluxes vanish at the extrapolated boundary of the reactor; and c) finite fluxes throughout the reactor.

Figure 2.1 shows the x,y problem geometry used in this feasibility study. Within each of the regions the material properties are assumed to be constant. An expansion is made in a finite Chebyshev polynomial series for the fast and thermal fluxes and sources in each region. Figure 2.2 shows the coordinate system within each region. Obviously, the coordinate system in each region of the reactor must be transformed into the range of the Chebyshev polynomials (-1 to 1).

For M = 2 in equations 2.1 and 2.2, which is the maximum number of terms used in this study,

$$\phi_{ip}(x,y) = C_{ip0}T_{00} + C_{ip1}T_{01} + C_{ip2}T_{02} + C_{ip3}T_{10} + C_{ip4}T_{11} + C_{ip5}T_{12} + C_{ip6}T_{20} + C_{ip7}T_{21} + C_{ip8}T_{22}, \qquad (2.3)$$

where: $C_{ip0} \equiv C_{ip00}$; $C_{ip1} \equiv C_{ip01}$; $C_{ip2} \equiv C_{ip02}$; $C_{ip3} \equiv C_{ip10}$; $C_{ip4} \equiv C_{ip11}$; $C_{ip5} \equiv C_{ip12}$; $C_{ip6} \equiv C_{ip20}$; $C_{ip7} \equiv C_{ip21}$; $C_{ip8} \equiv C_{ip22}$; and $T_{jk} \equiv T_{jk}(x,y)$.

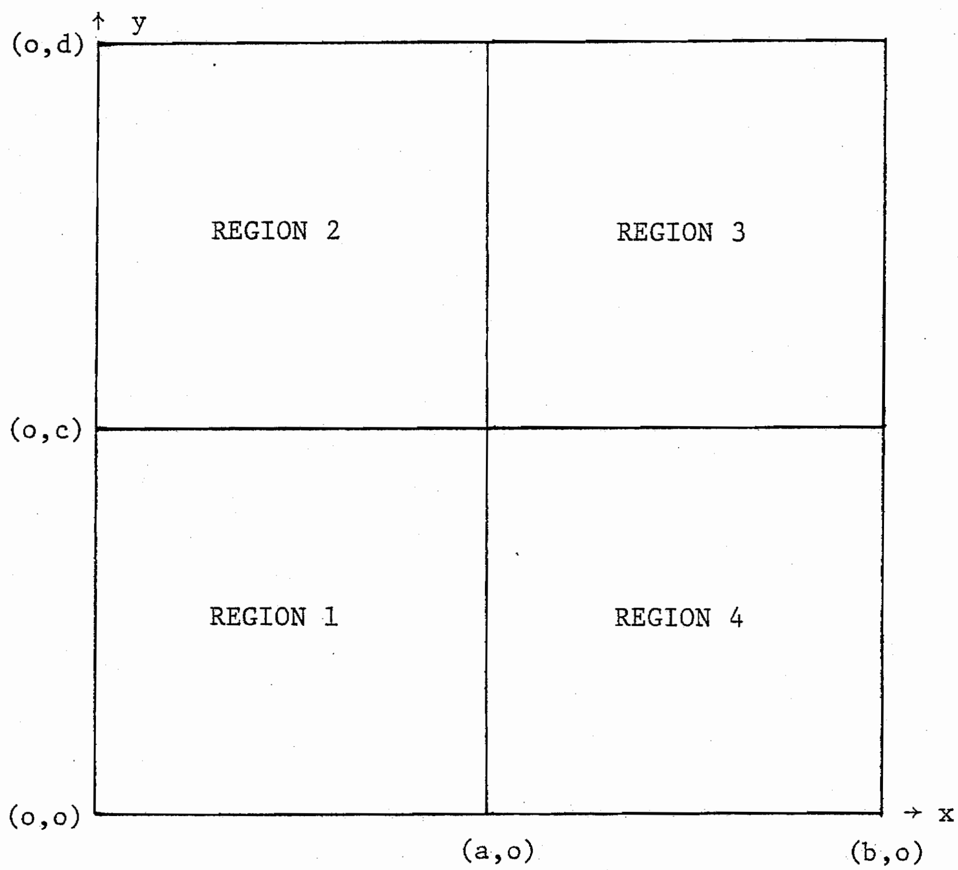


Figure 2.1 Reactor Geometry

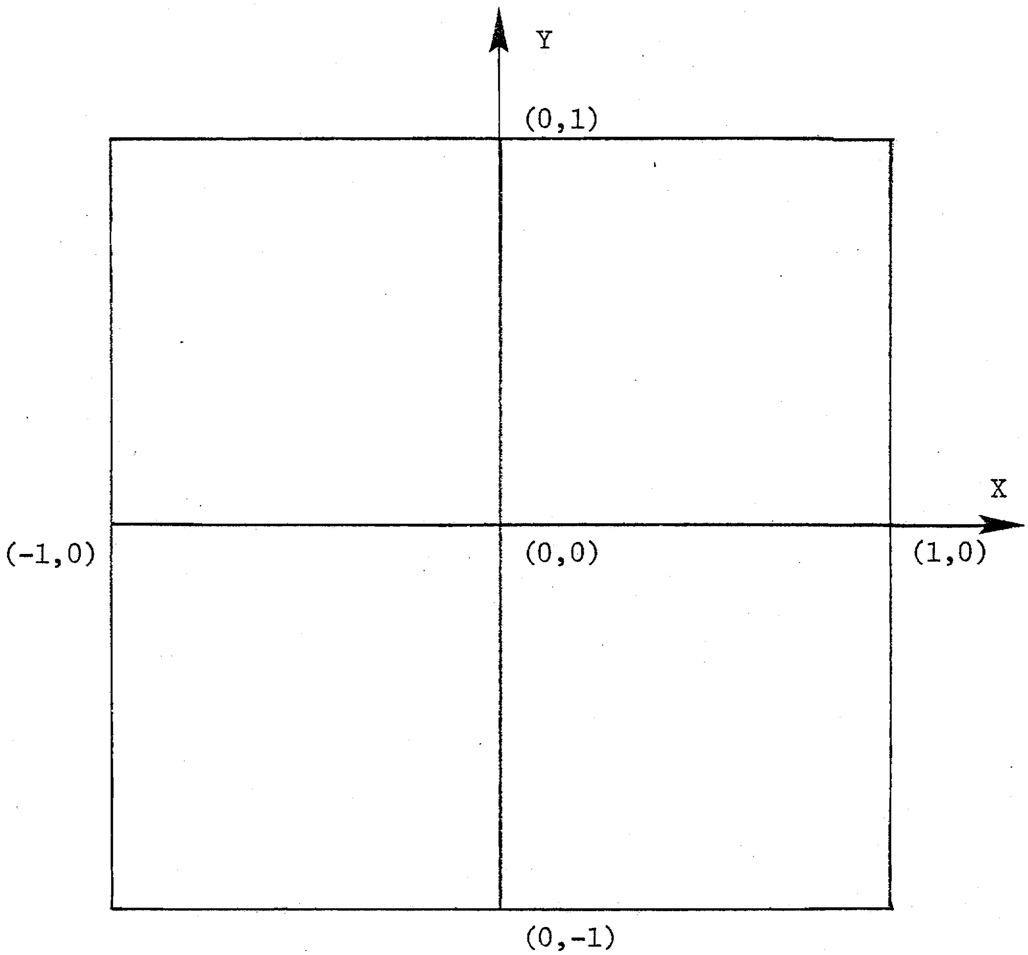


Figure 2.2 Coordinate System for One Region.

The neutron sources are:

$$\frac{\nu_{2p}}{\lambda}\,\sigma_{2pf}\phi_{2p}(x,y)$$

for the fast group, and

$$\sigma_{1ps}\phi_{1p}(x,y)$$

for the thermal group. A set of algebraic equations in the expansion coefficients, C_{ipjk}, can be obtained by taking the inner product of $T_{m,n}(x,y)$ and equations 2.1 and 2.2. The Laplacian operator is obtained from the material in Appendix II. This yields a set of eighteen algebraic equations for each region. The following nine equations are obtained from the fast group equation 2.1, where $f_p \equiv \frac{\nu_{2p}}{\lambda}\sigma_{2pf}$:

$$-D_{1p}(C_{1p2}+C_{1p6})+\sigma_{1pr}C_{1p0} = f_pC_{2p0};$$

$$-D_{1p}C_{1p7}+\sigma_{1pr}C_{1p1} = f_pC_{2p1};$$

$$-D_{1p}C_{1p8}+\sigma_{1pr}C_{1p2} = f_pC_{2p2};$$

$$-D_{1p}C_{1p5}+\sigma_{1pr}C_{1p3} = f_pC_{2p3};$$

$$\sigma_{1pr}C_{1p4} = f_pC_{2p4};$$

$$\sigma_{1pr}C_{1p5} = f_pC_{2p5};$$

$$-D_{1p}C_{1p8}+\sigma_{1pr}C_{1p6} = f_pC_{2p6};$$

$$\sigma_{1pr}C_{1p7} = f_pC_{2p7};$$

$$\sigma_{1pr}C_{1p8} = f_pC_{2p8}.$$

Similarly, nine equations are obtained from the thermal group equation 2.2:

$$-D_{2p}(C_{2p2}+C_{2p6}) + \sigma_{2pr}C_{2p0} = \sigma_{1pr}C_{1p0};$$

$$-D_{2p}C_{2p7} + \sigma_{2pr}C_{2p1} = \sigma_{1pr}C_{1p1};$$

$$-D_{2p}C_{2p8} + \sigma_{2pr}C_{2p2} = \sigma_{1pr}C_{1p2};$$

$$-D_{2p}C_{2p5} + \sigma_{2pr}C_{2p3} = \sigma_{1pr}C_{1p3};$$

$$\sigma_{2pr}C_{2p4} = \sigma_{1pr}C_{1p4};$$

$$\sigma_{2pr}C_{2p5} = \sigma_{1pr}C_{1p5};$$

$$-D_{2p}C_{2p8} + \sigma_{2pr}C_{2p6} = \sigma_{1pr}C_{1p6};$$

$$\sigma_{2pr}C_{2p7} = \sigma_{1pr}C_{1p7};$$

$$\sigma_{2pr}C_{2p8} = \sigma_{1pr}C_{1p8}.$$

Therefore, 18 equations with 18 unknowns are used to represent the fast and thermal fluxes for each region of the reactor.

2.3 Evaluation of Boundary Conditions

Region 2 is selected to illustrate all types of boundary conditions encountered in the reactor. The three types of boundary conditions as shown in Figure 2.1 are: A) plane of symmetry--from point (o,c) to (o,d); B) exterior surface--from point (o,d) to (a,d); and C) interior surface--from points (o,c) to (a,c) and (a,c) to (a,d).

Since all functions will be evaluated at ±1, the values of the first three Chebyshev polynomials are (see reference 7):

$$T_n(1) = 1 \quad \text{and} \quad T_n(-1) = (-1)^n \quad \text{for} \quad n = 0, 1, 2.$$

First, consider the boundary (o,c) to (o,d) at which the normal component of the current is zero. Using the material from Appendix II, the derivative of the flux evaluated at $x = -1$ is:

$$\frac{\partial \phi_{i2}(-1,y)}{\partial x} \simeq T_0(y)[C_{i23} - 4C_{i26}] + T_1(y)[C_{i24} - 4C_{i27}] + T_2(y)[C_{i25} - 4C_{i28}] = 0 \qquad (2.4)$$

Again, the inner product of $T_m(y)$ and equation 2.4 yield the following set of three equations:

$$C_{i23} - 4C_{i26} = 0;$$

$$C_{i24} - 4C_{i27} = 0;$$

$$C_{i25} - 4C_{i28} = 0.$$

Thus, a plane of symmetry for a region requires three equations for each group.

Next, consider the boundary (o,d) to (a,d) at which $\phi_{i2}(x,1) = 0$. Then,

$$\phi_{i2}(x,1) \simeq T_0(x)[C_{i20} + C_{i21} + C_{i22}] + T_1(x)[C_{i23} + C_{i24} + C_{i25}] + T_2(x)[C_{i26} + C_{i27} + C_{i28}]. \qquad (2.5)$$

Again, the inner product of $T_m(x)$ and equation 2.5 yields the following three equations:

$$C_{i20} + C_{i21} + C_{i22} = 0;$$

$$C_{i23} + C_{i24} + C_{i25} = 0;$$

$$C_{i26} + C_{i27} + C_{i28} = 0.$$

Thus, three equations for each group are needed to describe an exterior surface at which the flux is zero.

Finally, consider the interior boundary (a,c) to (a,d). Since it is an interior boundary, both continuity of flux and the normal component of current must be satisfied. Continuity of flux gives:

$$\phi_{i2}(1,y) = \phi_{i3}(-1,y). \tag{2.6}$$

Equation 2.6 evaluated at the appropriate points is:

$$\begin{aligned} &T_0(y)[C_{i20} + C_{i23} + C_{i26}] + T_1(y)[C_{i21} + C_{i24} + C_{i27}] \\ &+ T_2(y)[C_{i22} + C_{i25} + C_{i28}] = T_0(y)[C_{i30} - C_{i33} + C_{i36}] \\ &+ T_1(y)[C_{i31} - C_{i34} + C_{i37}] + T_2(y)[C_{i32} - C_{i35} + C_{i38}]. \end{aligned} \tag{2.7}$$

The inner product of $T_m(y)$ with equation 2.7 yields the following three equations:

$$C_{i20} + C_{i23} + C_{i26} - C_{i30} + C_{i33} - C_{i36} = 0;$$

$$C_{i21} + C_{i24} + C_{i27} - C_{i31} + C_{i34} - C_{i37} = 0;$$

$$C_{i22} + C_{i25} + C_{i28} - C_{i32} + C_{i35} - C_{i38} = 0.$$

To satisfy continuity across the boundary (a,c) to (a,d) requires:

$$D_{i2} \frac{\partial \phi_{i2}(1,y)}{\partial x} = D_{i3} \frac{\partial \phi_{i3}(-1,y)}{\partial x}. \tag{2.8}$$

Using the material in Appendix II, Equation 2.8 is:

$$D_{i2}\{T_0(y)[C_{i23} + 4C_{i26}] + T_1(y)[C_{i24} + 4C_{i27}]$$

$$+ T_2(y)[C_{i25} + 4C_{i28}]\} = D_{i3}\ \{T_0(y)[C_{i33} - 4C_{i36}]$$

$$+ T_1(y)[C_{i34} - 4C_{i37}] + T_2(y)[C_{i35} - 4C_{i38}]\}. \qquad (2.9)$$

The inner product of $T_m(y)$ with equation 2.9 yields the following three equations:

$$D_{i2}(C_{i23} + 4C_{i26}) - D_{i3}(C_{i33} - 4C_{i36}) = 0;$$

$$D_{i2}(C_{i24} + 4C_{i27}) - D_{i3}(C_{i34} - 4C_{i37}) = 0;$$

$$D_{i2}(C_{i25} + 4C_{i28}) - D_{i3}(C_{i35} - 4C_{i38}) = 0.$$

Thus, for each interior surface six equations are required for each group to satisfy the two boundary conditions.

2.4 A Numerical Method of Solution

From sections 2.3 and 2.4, the number of unknowns and equations required to completely specify the four region-two group neutron diffusion problem shown in Figure 2.1 can be determined. The number of equations for each energy group are:

Number of Equations

1) Inner product of T_{jk} with the diffusion equation within each region:

    (9 equations) (4 regions) = 36

2) Planes of symmetry:

    (o,o) to (o,c); (o,c) to (o,d);

(o,o) to (a,o); and (a,o) to (b,o).

(3 equations) (4 planes) = 12

3) Exterior surface:

(o,d) to (a,d); (a,d) to (b,d);

(b,d) to (b,c); and (b,c) to (o,b).

(3 equations) (4 surfaces) = 12

4) Interior surface:

(a,o) to (a,c); (o,c) to (a,c);

(a,c) to (a,d); and (a,c) to (a,b).

(6 equations) (4 surfaces) = 24

The total number of equations for one group is: 84.

Since each flux in each region has nine unknown expansion coefficients, the total number of independent variables is 36. Therefore, the system for two energy groups has 168 equations with 72 unknowns. Appendix III gives a complete listing of these equations.

Two possible approaches to solving this overdetermined system of algebraic equations are: a) to form and solve the normal equations, and b) to use an orthogonalization method. Generally, the normal equations have a very ill-conditioned matrix. To avoid this problem, an orthogonalization method is used here.[8]

Algorithms which avoid forming the normal equations are based on the decomposition of the rectangular coefficient matrix into the product of a matrix $\underline{Q}$ with orthogonal columns and an upper-triangular matrix $\underline{R}$. In practice, the $\underline{Q}$ matrix need be computed only once at the start of the problem. The solution to the system of equations can be obtained by interation in which only the R matrix is recalculated at each iteration.

The problems are solved using a power iteration method. An initial source distribution for the fast group equations is assumed for the reactor. These equations are then solved by a Gram-Schmidt method[9] to give the zeroth iterate for the fast flux. The thermal group source is then calculated from the fast flux and the zeroth iterate of the thermal flux is obtained in a similiar manner. Next, the fast group source is recalculated and the first iterate of the fast flux is obtained. The iterative cycle is then repeated. For each iteration, the eigenvalue is calculated. The successive eigenvalues do converge for the cases studied and the calculation is stopped when $|\lambda^{(k)} - \lambda^{(k-1)}| \leq 5 \times 10^{-4}$. The Chebyshev Diffusion Program code, called CDP, is written in IBM[10] Fortran IV. A flow chart of CDP is given in Figure 2.3. The computer code is written using double precision arithmetic to reduce round-off error.

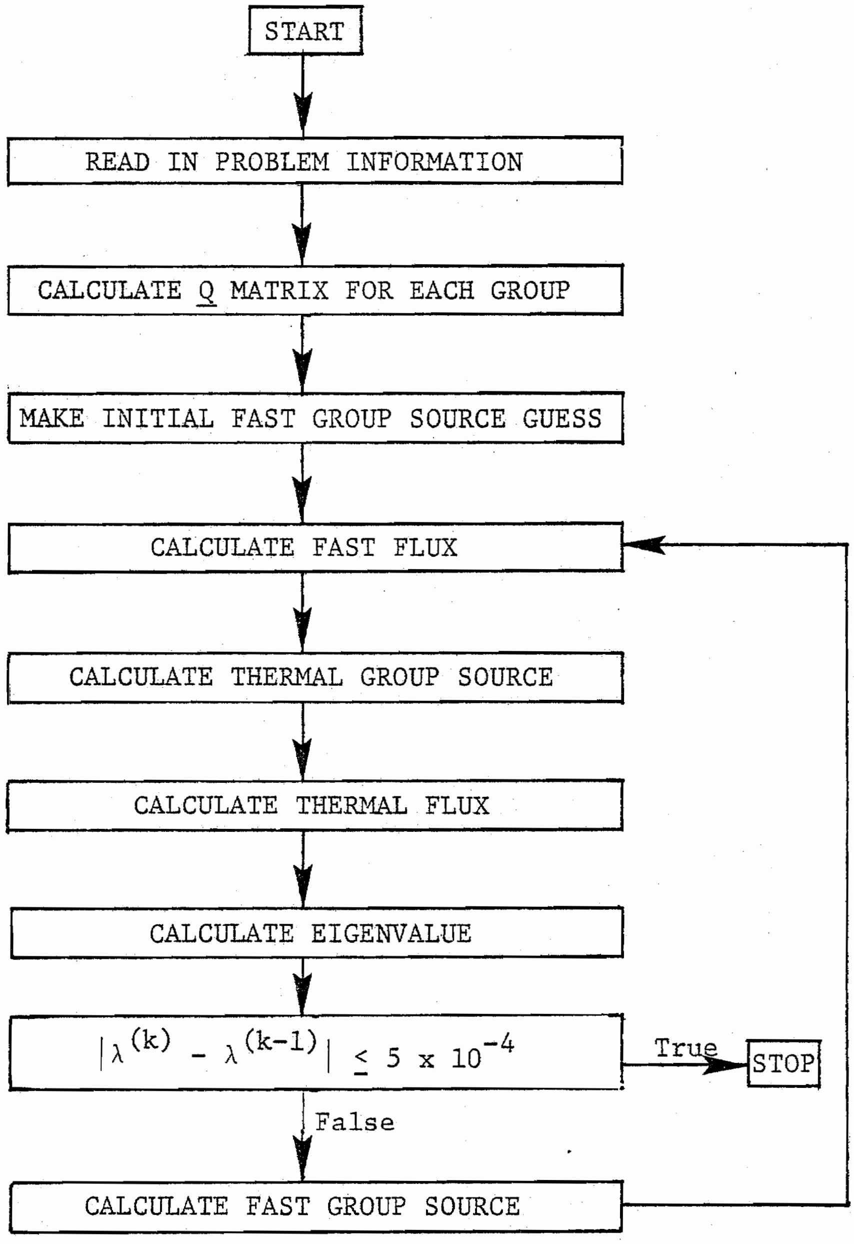


Figure 2.3 Flow Chart of CDP

CHAPTER III

RESULTS AND CONCLUSIONS

3.1 Numerical Results and Conclusions

The CDP method developed herein is tested by comparing eigenvalues and flux distributions of the CDP approximation to those obtained using a well known finite difference code, PDQ-7,[11] for two different problems. The mesh spacing for the PDQ-7 "solutions" is 1 cm, which is somewhat less than the smallest mean free path. Thus, the PDQ-7 solutions with point by point flux convergence give the "best" answers that can be obtained with a first order finite difference method, which will therefore be used as a standard of comparison. The first problem is a water-reflected rectangular core. The second problem is an actual problem solved by a utility for on-line fuel management in a Pressurized Water Reactor.[12] To remove any computer induced bias from the results, all problems are executed on an IBM 370/158 series computer.

Problem I is a water-reflected core which presents a rather severe test of the CDP method because of the large thermal flux peaking at the water-core interface. The geometry shown in Figure 2.1 represents one-quarter section of the reactor in which region 1 represents the core and regions 2, 3, and 4 represent a water reflector. The core is a 40 x 40 cm square surrounded by a 20 cm reflector. The material properties are given in Table 3.1.

TABLE 3.1

MATERIAL PROPERTIES FOR PROBLEM I

REFLECTOR REGIONS (p = 2, 3, 4)

	FAST (i=1)	THERMAL (i=2)
σ_{ipr} (cm^{-1})	0.101	0.020
σ_{ips} (cm^{-1})	0.100	0.000
D_{ip} (cm)	1.200	0.150

CORE REGION (p = 1)

	FAST (i=1)	THERMAL (i=2)
σ_{ipr} (cm^{-1})	0.0623	0.020
σ_{ips} (cm^{-1})	0.0600	0.000
D_{ip} (cm)	1.5000	0.400
σ_{ipf} (cm^{-1})	0.0000	0.218

Problem II represents a Pressurized Water Reactor at the time of a refueling. Four categories of fuel assemblies can be identified, each of which have different material properties. They are arranged in a checkerboard pattern in the interior of the core as shown in Figure 3.1. Higher enrichment fuel is placed at the exterior of the core. The material properties are given in Tables 3.2 and 3.3.

Table 3.4 summarizes the computer time requirements and eigenvalues found for both problems. Graphs 3.1 to 3.6 show the flux distributions along the X axis at Y = 0, 10, and 20 cm for Problem I. The fast flux is normalized to unity at (0,0). Both PDQ-7 and CDP give about the same distribution for the fast flux. The eigenvalues calculated by both methods agree very well, being identical to one part in 4806. Except for a small region near the core-reflector interface, the thermal neutron fluxes within the core differ by less than about 4%. The maximum difference in the core thermal fluxes occurs at the core-reflector interface, and is about 20%. Within the reflector, the difference in thermal fluxes becomes quite large. The goal of significantly reducing the computer time required is met for Problem I (0.45 sec for CDP vs 32.6 sec for PDQ-7). It is important to note, however, that the time required to obtain the "standard" solution for comparison, using PDQ-7 is larger than would be needed for a large mesh spacing. Thus, the savings in computer time required to achieve a solution using a PDQ-7 type code giving results comparable to those obtained with the CDP code cannot be inferred directly from these results. In any event, note that the CDP method is indeed economical in both preparation of input data and computer time.

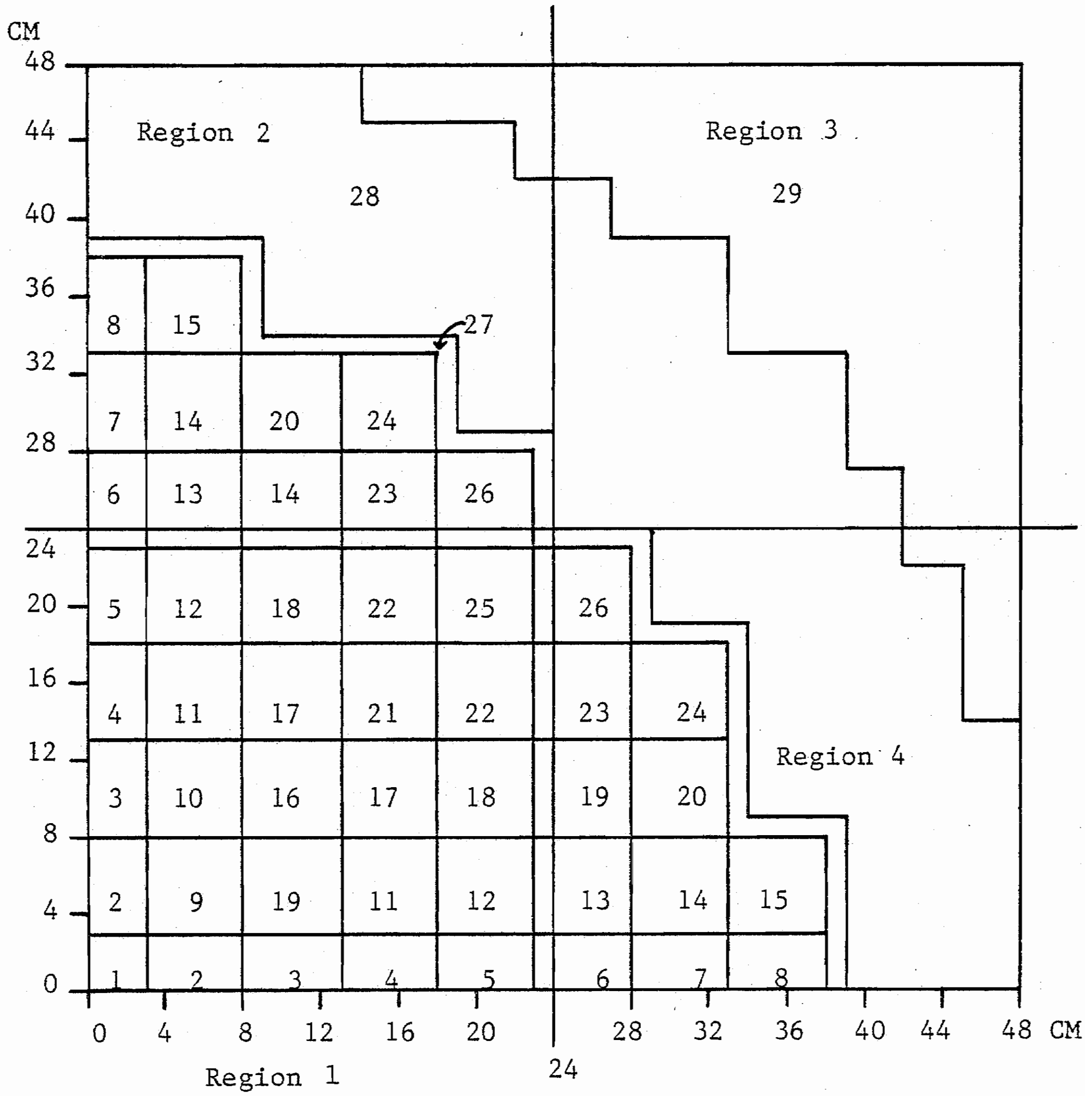


Figure 3.1 PWR Core Layout for Problem II

The number in each region are the composition number for Tables 3.2 and 3.3.

TABLE 3.2

FAST GROUP CONSTANTS FOR PROBLEM II

COMPOSITION NUMBER	D_{1p}	σ_{1pa}	σ_{1pr}
1	1.412	.00883	.0195
2	1.420	.00966	.0180
3	1.412	.00883	.0195
4	1.420	.00966	.0180
5	1.412	.00883	.0195
6	1.420	.00966	.0180
7	1.412	.00883	.0195
8	1.431	.00962	.0188
9	1.412	.00883	.0195
10	1.420	.00966	.0180
11	1.412	.00883	.0195
12	1.420	.00966	.0180
13	1.412	.00883	.0195
14	1.431	.00997	.0177
15	1.431	.00962	.0188
16	1.412	.00883	.0195
17	1.420	.00966	.0180
18	1.412	.00883	.0195
19	1.420	.00966	.0180
20	1.431	.00962	.0188
21	1.412	.00883	.0195
22	1.420	.00966	.0100
23	1.431	.00997	.0177
24	1.431	.00962	.0188
25	1.412	.00883	.0195
26	1.431	.00962	.0188
27	1.084	.00335	.0117
28	1.446	.00121	.0339
29	0.001	10.0	0.1

TABLE 3.3

THERMAL GROUP CONSTANTS FOR PROBLEM II

COMPOSITION NUMBER	D_{2p}	σ_{2pa}	σ_{2pr}	σ_{2pf}
1	0.4317	0.0615	0	0.0755
2	0.4538	0.0806		0.0976
3	0.4317	0.0615		0.0755
4	0.4538	0.0806		0.0976
5	0.4317	0.0615		0.0755
6	0.4538	0.0806		0.0976
7	0.4317	0.0615		0.0755
8	0.4485	0.0775		0.1118
9	0.4317	0.0615		0.0755
10	0.4538	0.0806		0.0976
11	0.4317	0.0615		0.0755
12	0.4538	0.0806		0.0976
13	0.4317	0.0615		0.0755
14	0.4608	0.0870		0.1123
15	0.4485	0.0775		0.1118
16	0.4317	0.0615		0.0755
17	0.4538	0.0806		0.0976
18	0.4317	0.0615		0.0755
19	0.4538	0.0806		0.0976
20	0.4485	0.0775		0.1118
21	0.4317	0.0615		0.0755
22	0.4538	0.0806		0.0976
23	0.4608	0.0870		0.1123
24	0.4485	0.0775		0.1118
25	0.4317	0.0615		0.0755
26	0.4485	0.0775		0.1118
27	0.3332	0.1174		0
28	0.2904	0.04858		0
29	0.001	10.0	0	0

TABLE 3.4

RESULTS OF TEST PROBLEMS

		PDQ-7	CDP
Execution Time (Sec)	Problem I	32.6	0.45
	Problem II	101	0.69
Eigenvalue	Problem I	0.48634	0.48628
	Problem II	1.00644	1.00597

Problem I: Water Reflected Core

Problem II: Fuel Management Problem

In Problem II, note that only four very large regions are used in the CDP method for the entire reactor. This is much larger than a typical finite difference mesh which might be used for such a problem. This large region size might seem inappropriate. However, Chebyshev polynomials are very efficient for the approximate representation of well-behaved functions. In this particular case, the reactor is partitioned into four large regions, within each of which the properties vary markedly from fuel assembly to fuel assembly. This partitioning into large regions gives a rather severe test of the use of a minimal matrix size for the CDP method. Within each of these large regions, the average material properties are used in the calculation.

The flux distributions for both methods at Y = 0, 12, and 24 cm for the fast and thermal groups are shown in graphs 3.7 to 3.12 with the fast fluxes for both solutions normalized to $\phi(0,0) = 1$. Both methods give nearly the same fast flux distributions. The PDQ-7 results, however, show the variation in thermal flux due to the different enrichments of the individual fuel assemblies. The CDP results, on the other hand, yield the correct average flux distribution to within 2% within the core, except near the core boundary, and the eigenvalue to within one part in 2141. The very large (0.69 sec vs. 101 sec) savings in computer time should make this method very useful for on-line fuel management, fuel cycle survey work, or conceptual design calculations.

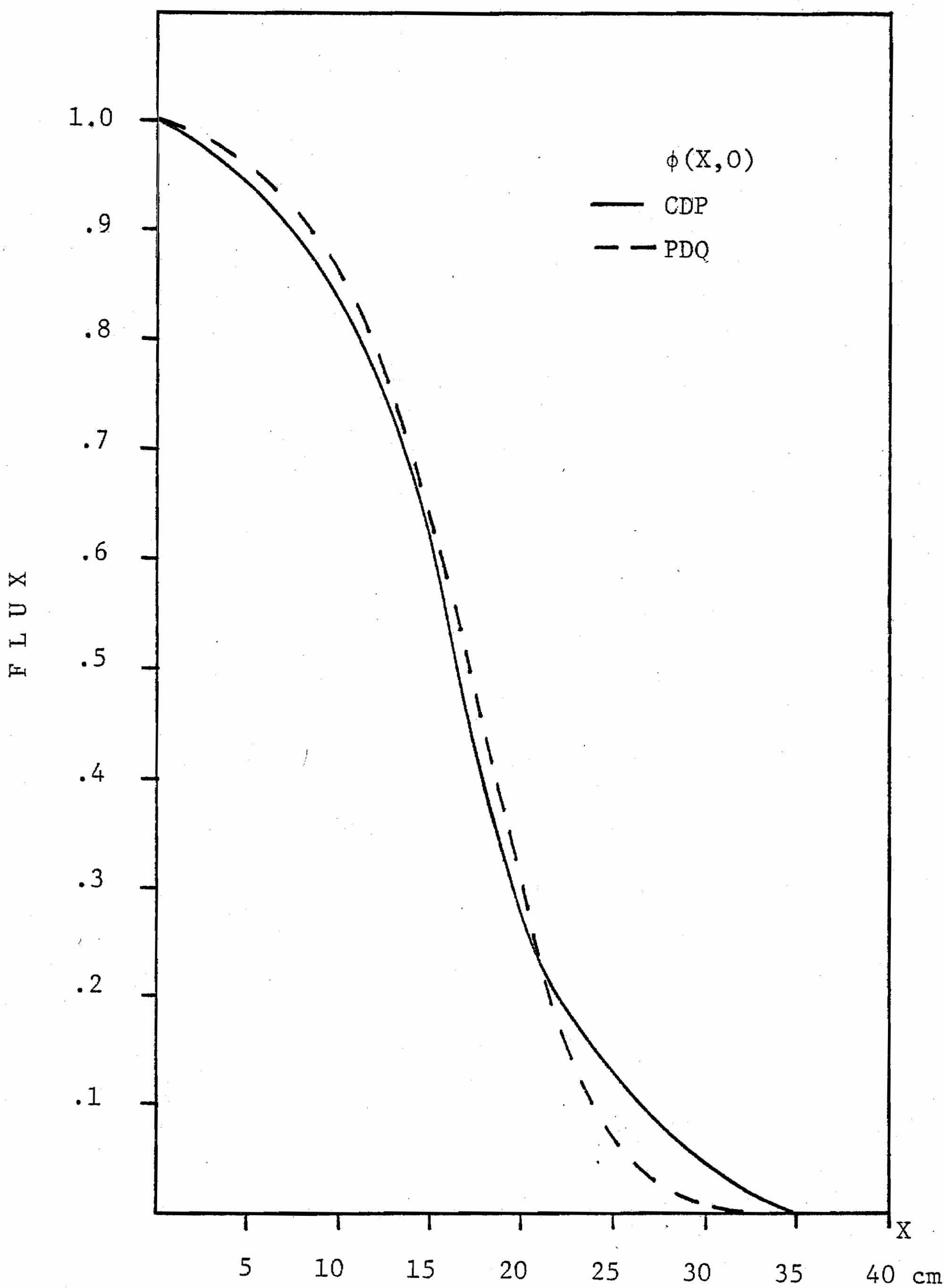
The small matrices make implementation very easy because the fast computer core memory required is only 100K bytes. The PDQ-7 program contains about 20,000 statements, which requires a large amount of disk space for storage. Also, the man-hours needed to prepare the input data for PDQ-7 are high because of the need to specify the material for each mesh point. In contrast, the input of the CDP program is relatively simple because only a few regions need to be specified.

These two diverse examples indicate the power of using Chebyshev polynomial expansions over large regions for the solution of elliptic boundary value problems. The time required for solution of a two group-two dimensional problem is about 0.01 n^3 sec, where n is the number of regions in the problem. Thus, more regions of smaller size to obtain greater detail in flux calculations can be used in practical applications. For example, a six-region problem would require about 2 seconds. The key to the small computer time required is, of course, the very small matrices required for adequate representation of a reactor core.

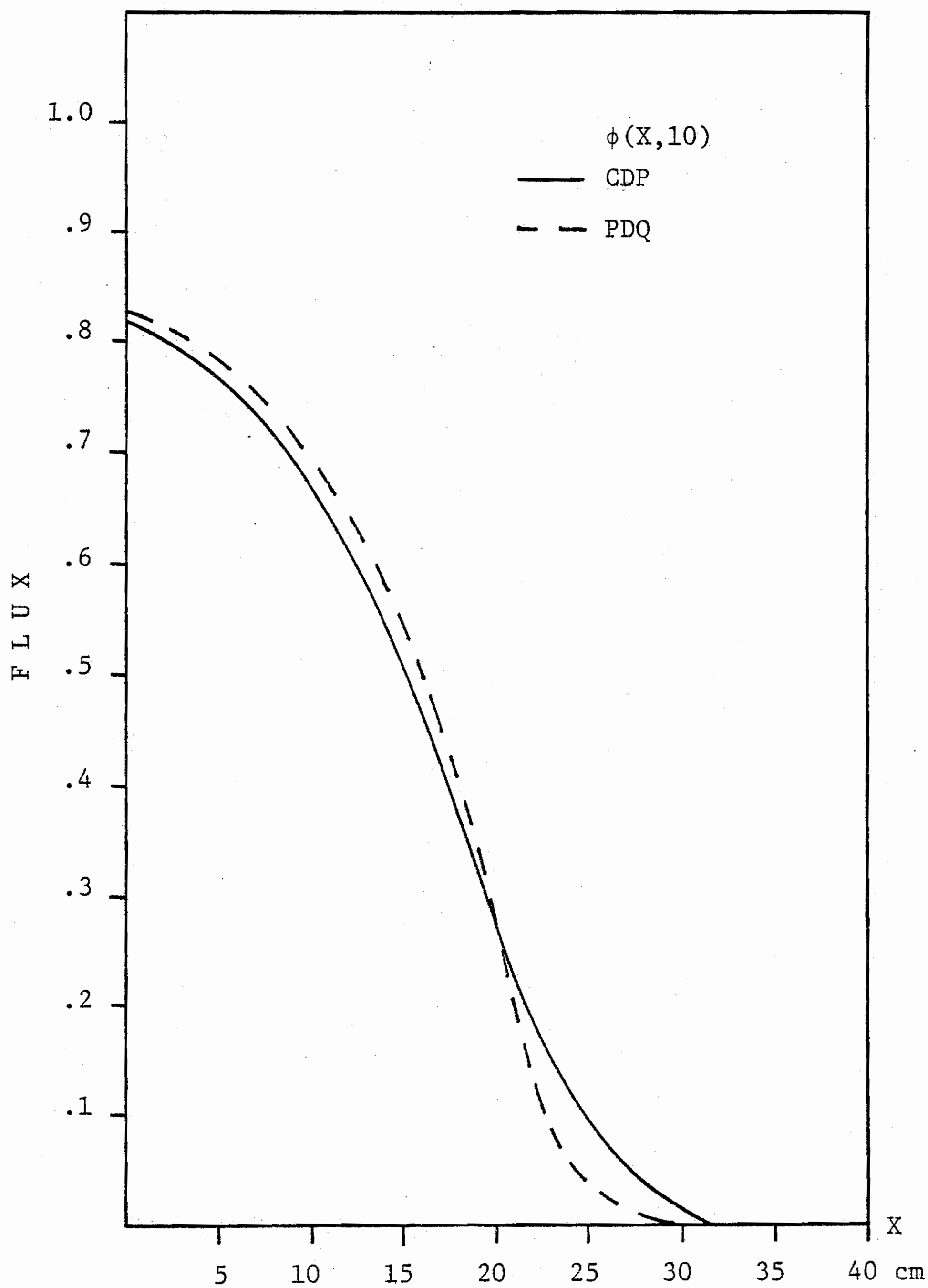
The feasibility of using Chebyshev polynomial expansions for two-dimensional multi-group diffusion calculations is demonstrated for these two diverse problems. This method compared with PDQ-7 gives not only very accurate eigenvalues, but also reasonably accurate flux distributions within fuel regions. The first result is not surprising, since eigenvalues are relatively easy to calculate to good accuracy by a variety of methods, especially those using a variational principle. On the other hand, it is much more difficult to calculate the spatial distribution of fluxes with approximate methods using very little computer time.

3.2 Recommendations

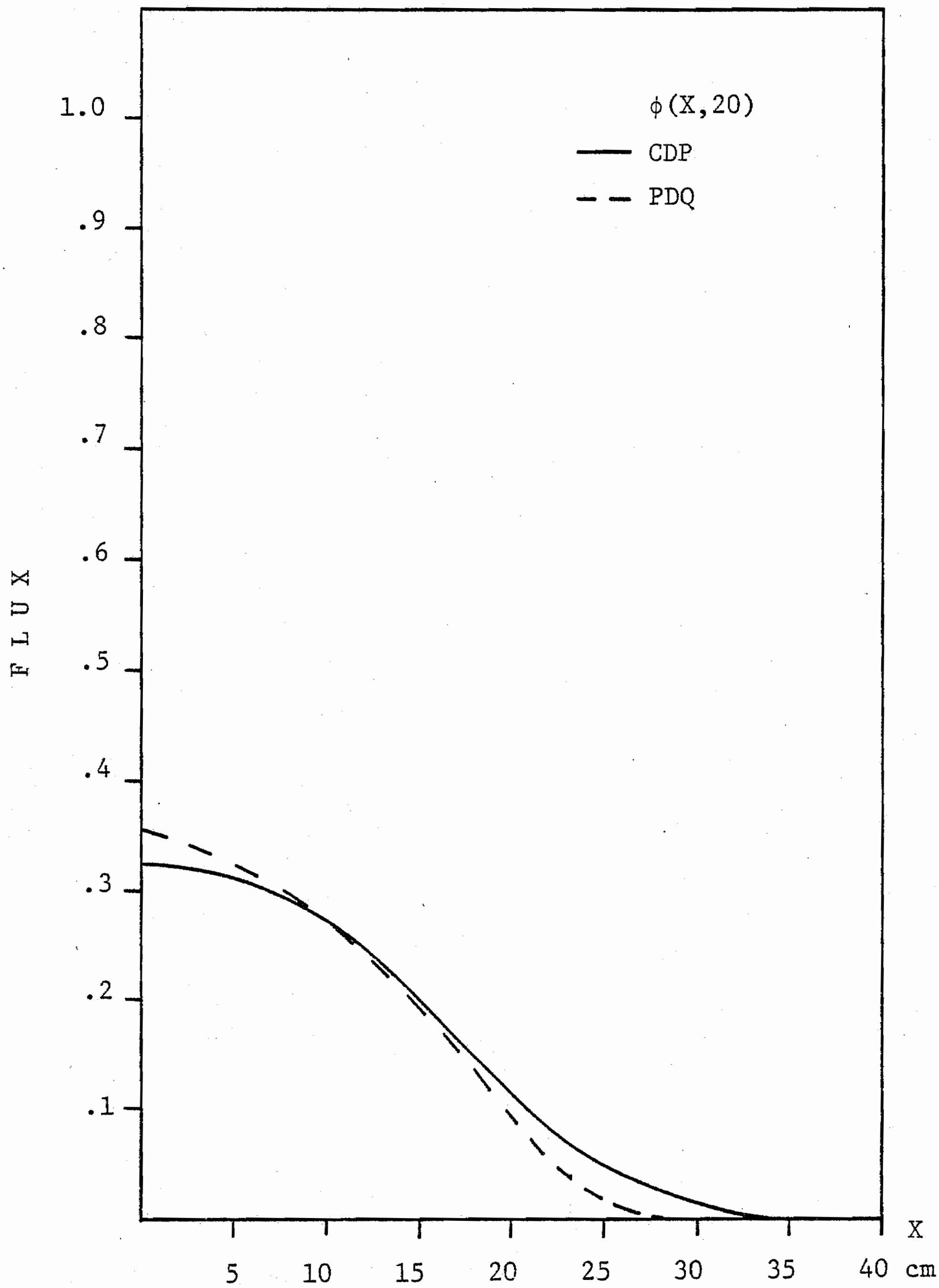
The results obtained in this dissertation lead to several promising areas for further investigation. Because of the very fast solution time, this method might be very useful in the solution of three dimensional neutron diffusion problems. Also, since the algorithm used in these calculations may not necessaryily be optimal for all applications of interest, other algorithms can be developed. In particular, a bivariate Chebyshev polynomial surface fitting procedure can be used to fit typical flux distributions for specific applications. This would show which terms of an expansion are the largest for the specific application and the algorithm could be modified to reduce the computer time required. An analysis of the error in the magnitude of the flux distribution in the neighborhood of singular points, such as point (a,c) shown in Figure 2.1, might also lead to algorithms giving better accuracy for this method.



Graph 3.1 Fast Flux at Y=0
FOR PROBLEM I

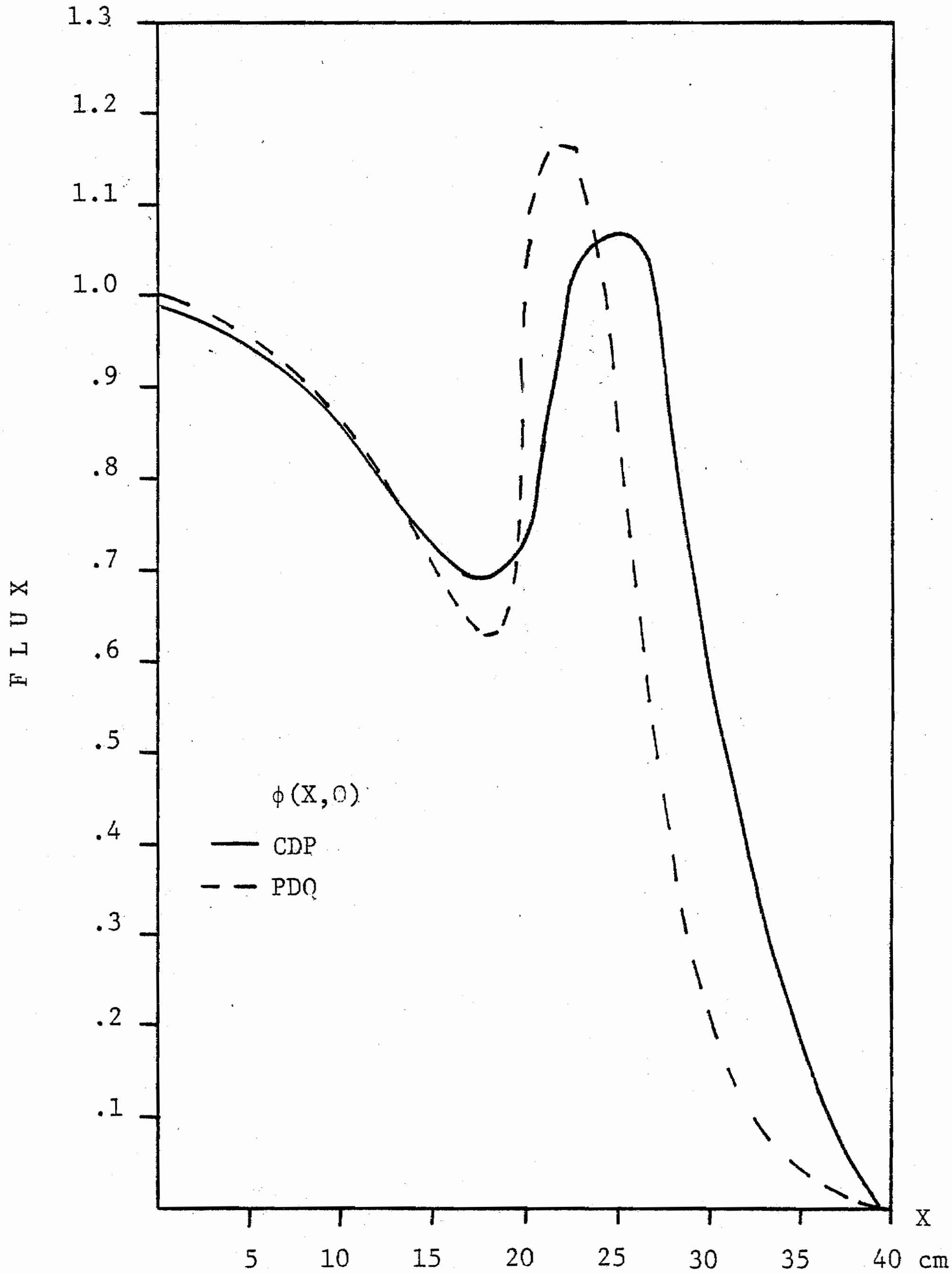


Graph 3.2 Fast Flux at Y=10
FOR PROBLEM I

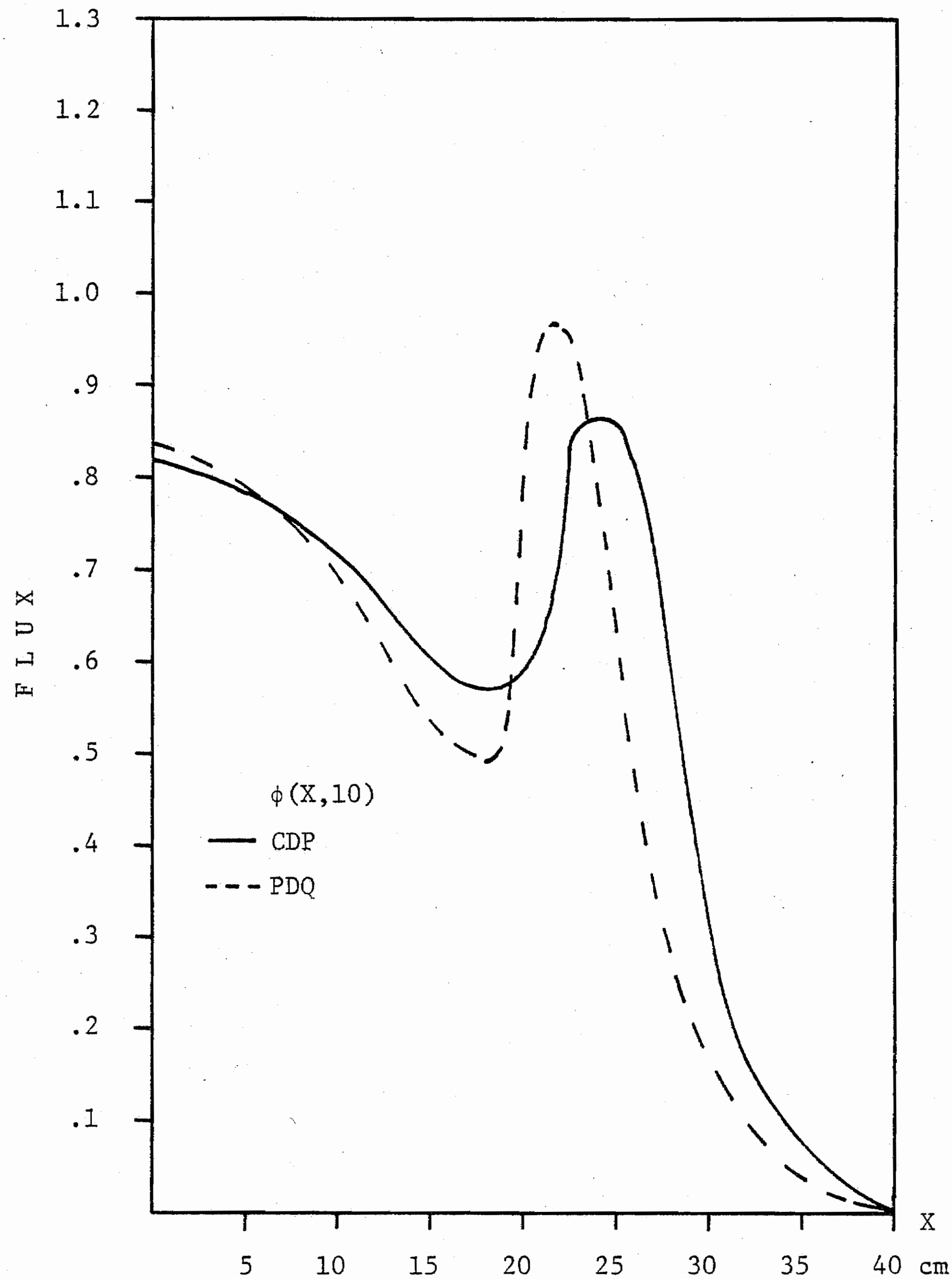


Graph 3.3 Fast Flux at Y=20

FOR PROBLEM I

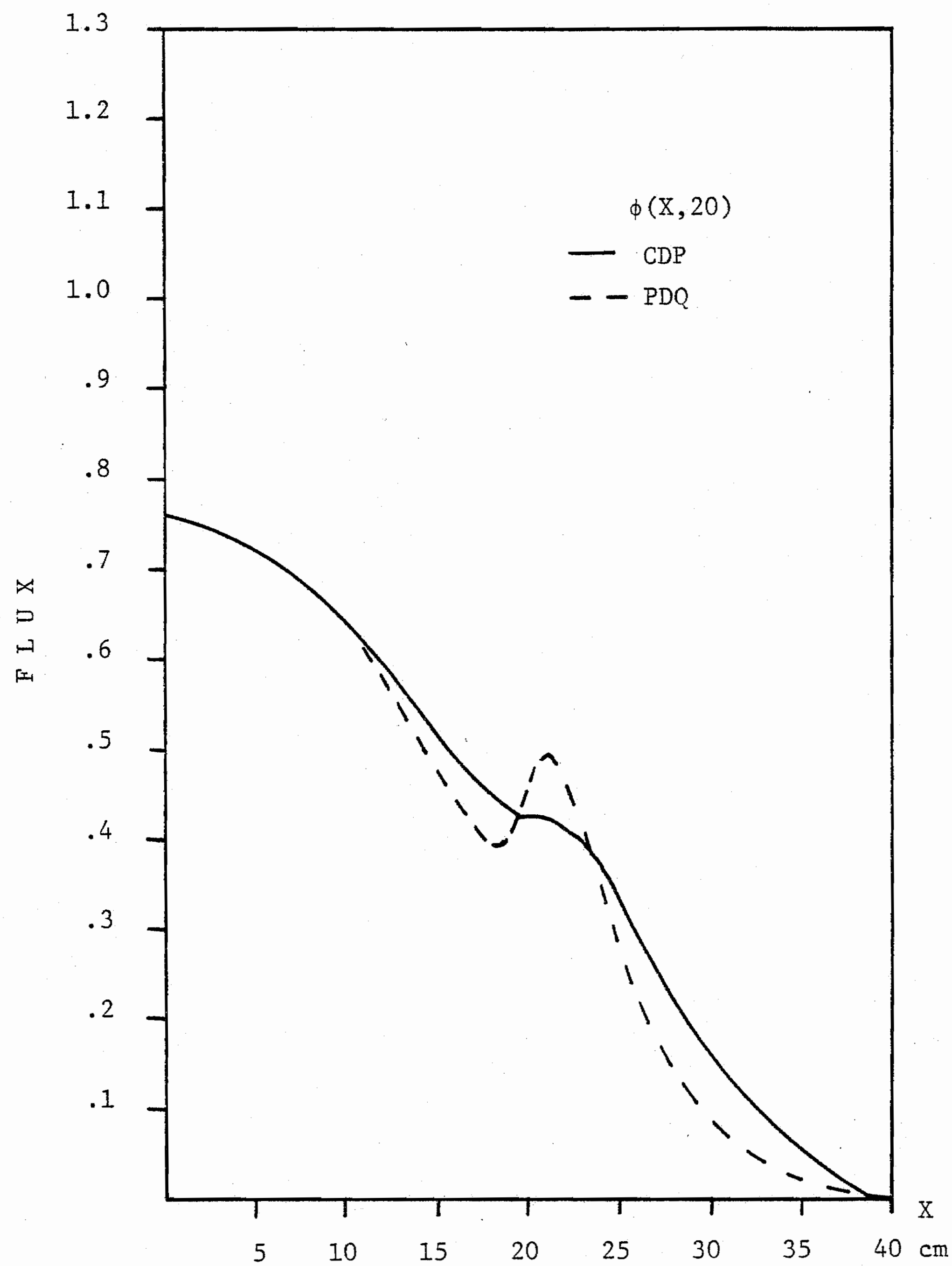


Graph 3.4 Thermal Flux at Y=0
FOR PROBLEM I

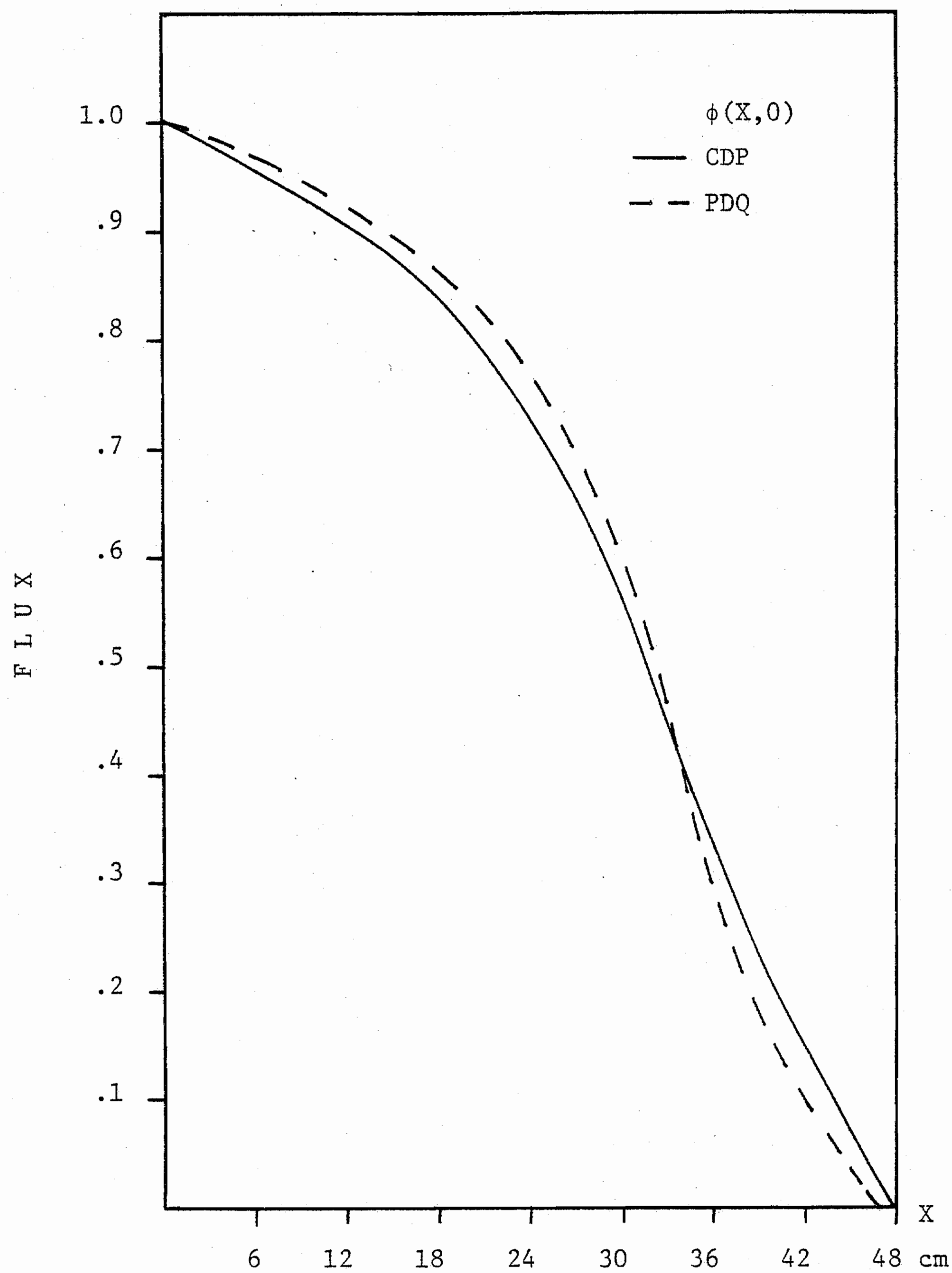


Graph 3.5 Thermal Flux at Y=10

FOR PROBLEM I

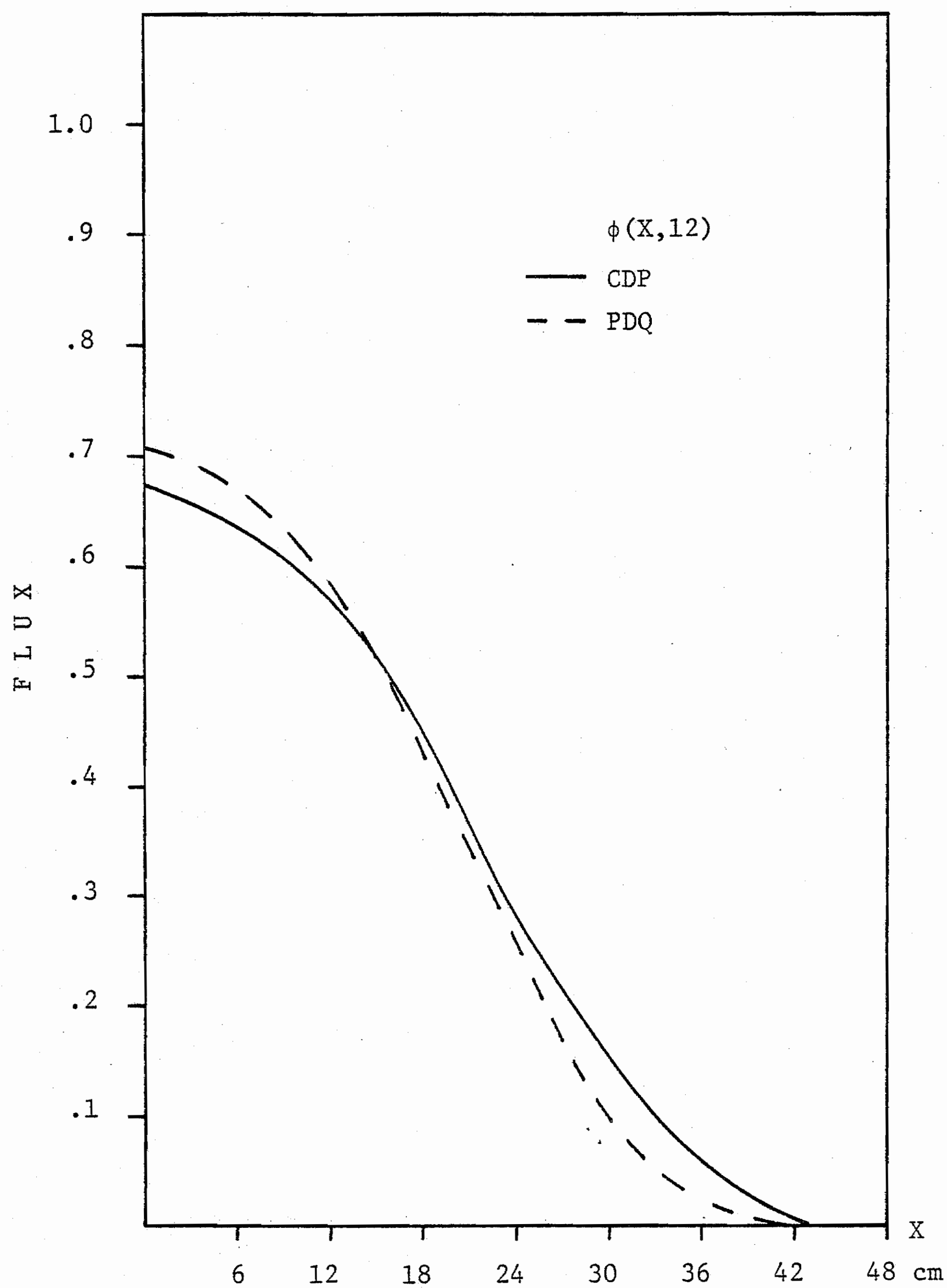


Graph 3.6 Thermal Flux at Y=20
FOR PROBLEM I

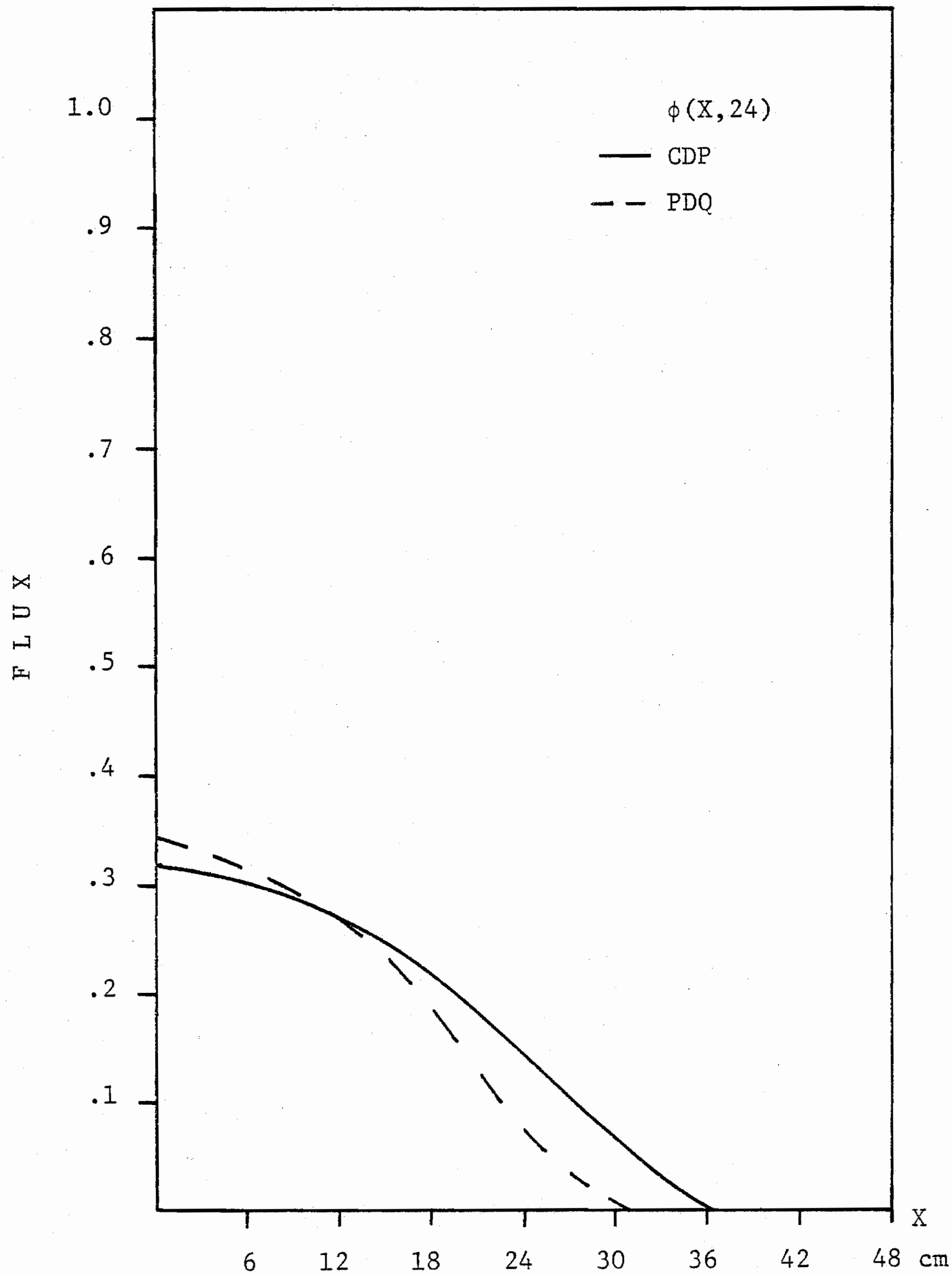


Graph 3.7 Fast Flux at Y=0

FOR PROBLEM II

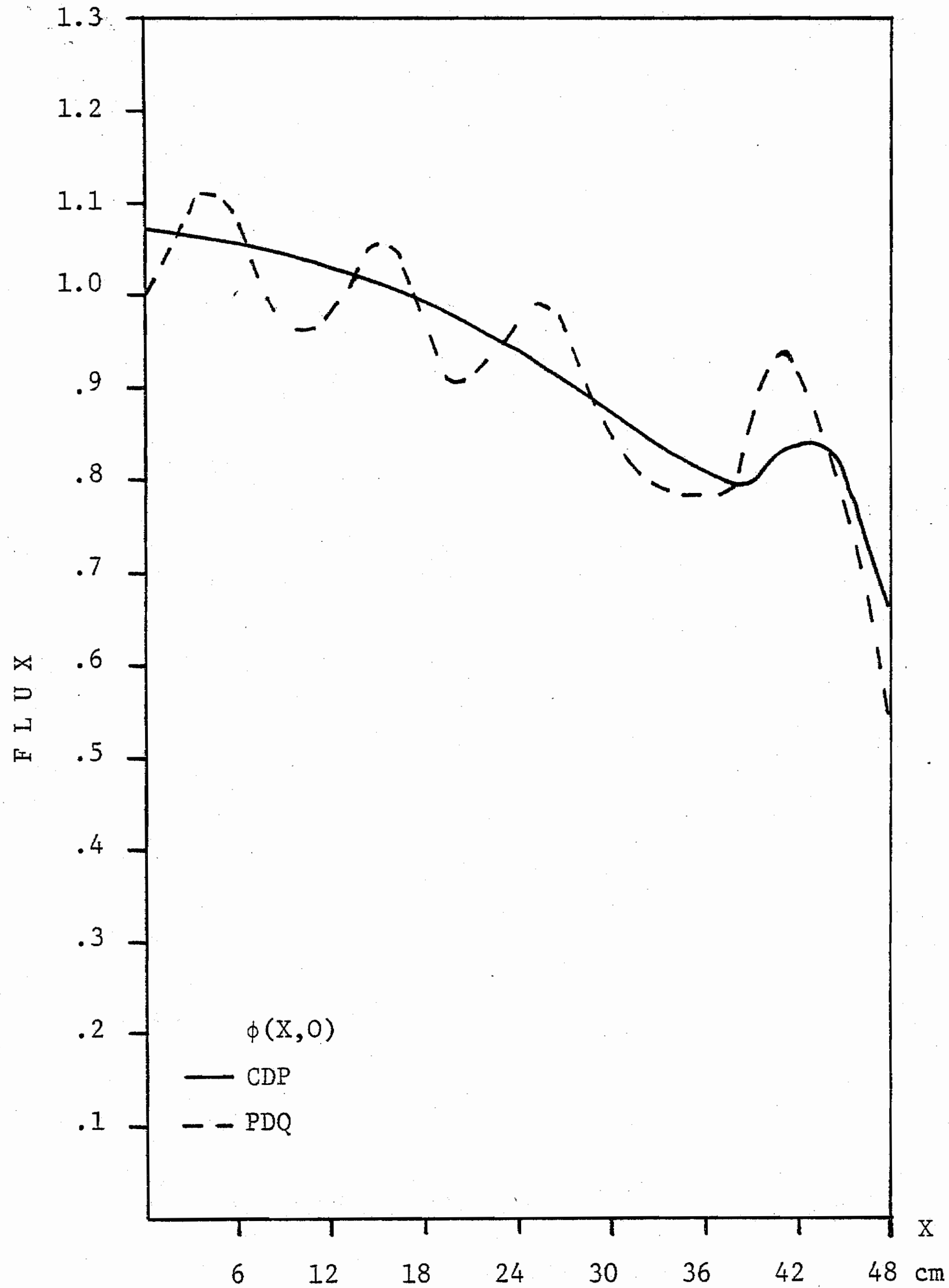


Graph 3.8 Fast Flux at Y=12
FOR PROBLEM II

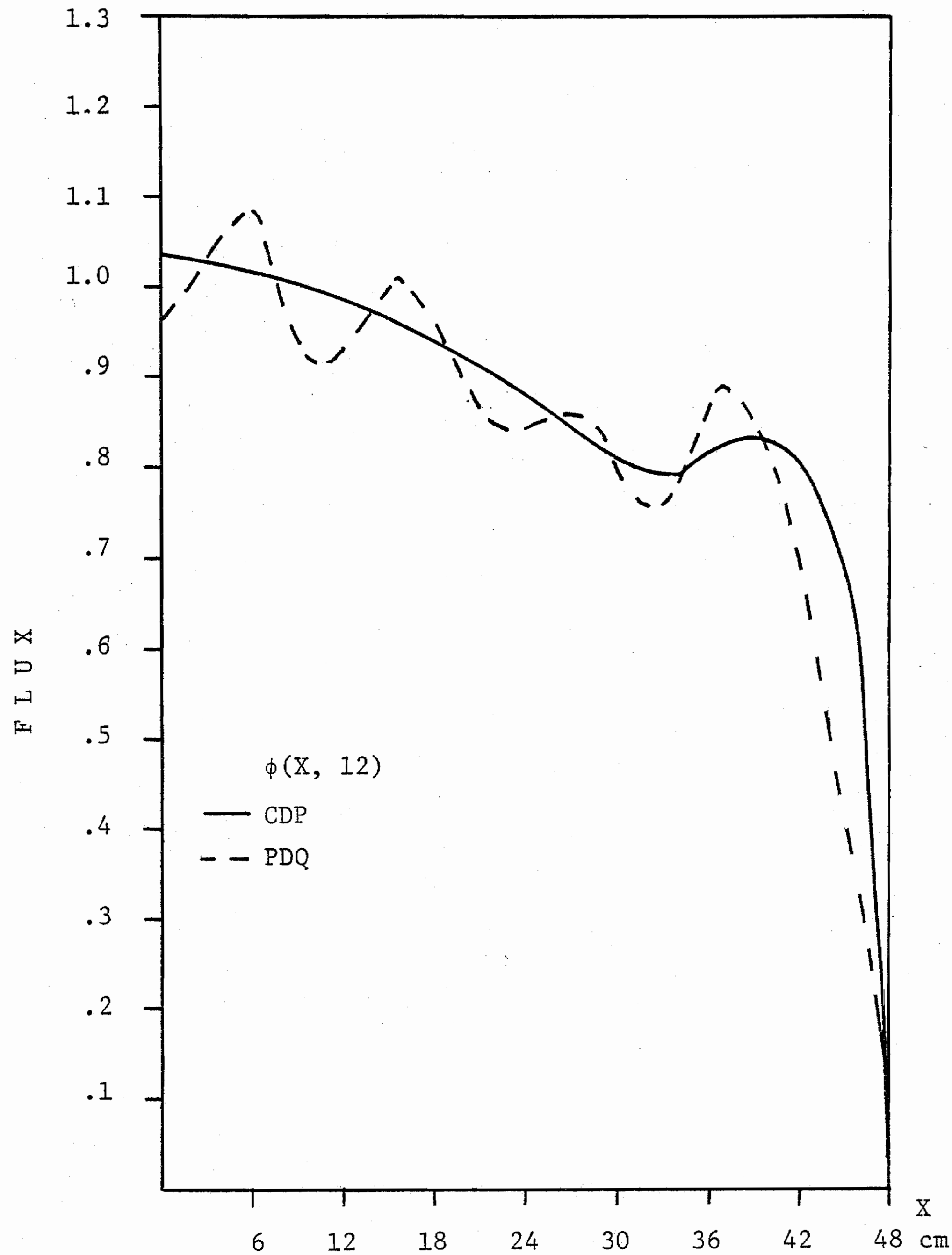


Graph 3.9 Fast Flux at Y=24

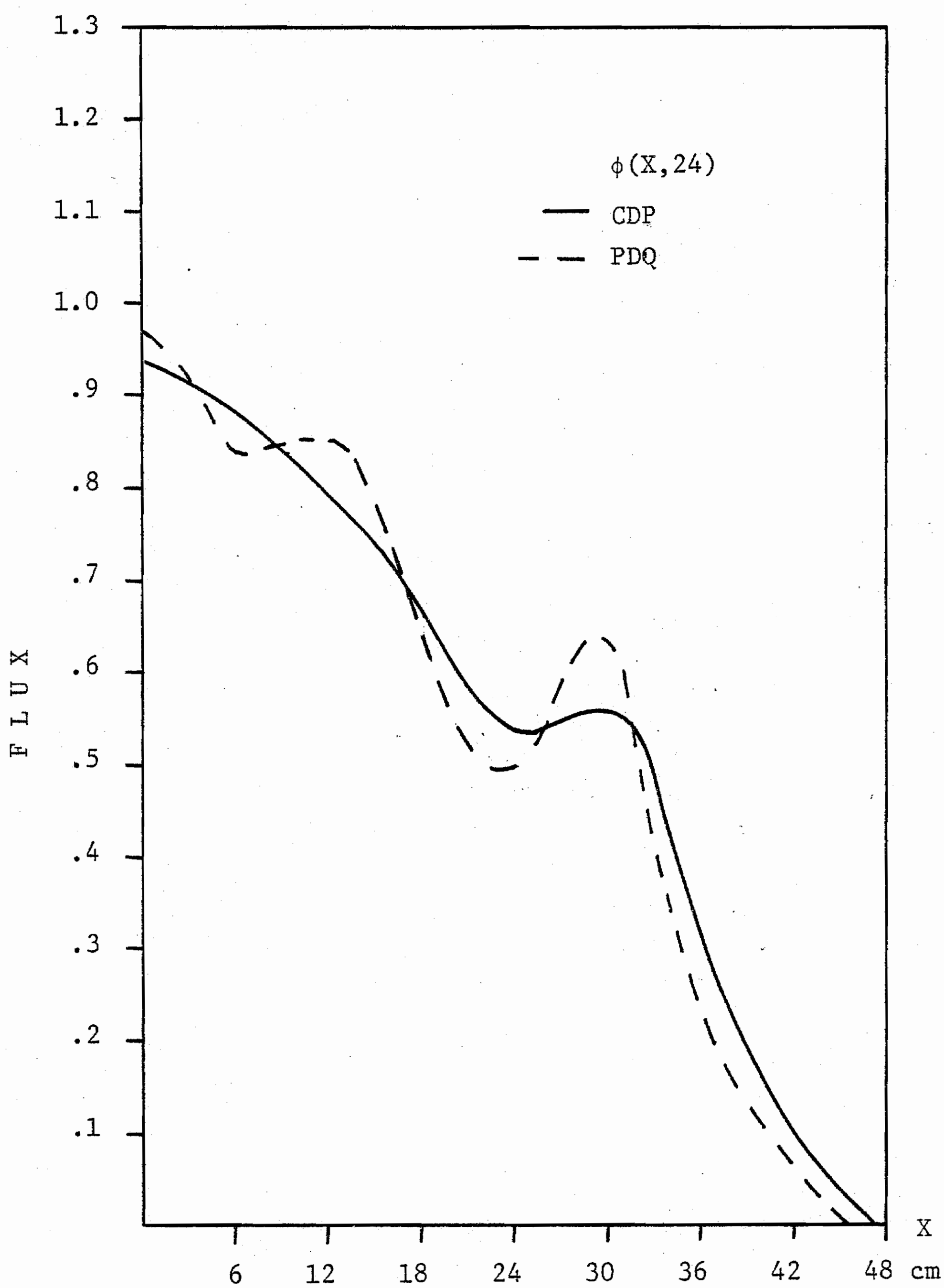
FOR PROBLEM II



Graph 3.10 Thermal Flux at Y=0
FOR PROBLEM II



Graph 3.11 Thermal Flux at Y=12
FOR PROBLEM II



Graph 3.12 Thermal Flux at Y=24
FOR PROBLEM II

REFERENCES

1. P. M. Dew and R. E. Scraton, "The Solution of the Heat Equation in Two Space Variables Using Chebyshev Series," Institute of Mathematics and Its Applications, Journal, VII, 1973, p. 231-240.

2. Melville Clark, Jr. and Kent F. Hansen, Numerical Methods of Reactor Analysis, Academic Press, 1964.

3. Cornelius Lanczos, Applied Analysis, Prentice-Hall, 1956.

4. J. C. Mason, "Chebyshev Polynomial Approximation for the L-Membrance Eigenvalue Problem," SIAM Journal Applied Mathematics, V15, 1967, p. 172-186.

5. P. M. Dew and R. E. Scraton, op. cit.

6. P. M. Dew and R. E. Scraton, "Chebyshev Methods for the Numerical Solution of Parabolic Partial Differential Equations in Two and Three Space Variables," Institute of Mathematics and Its Applications, Journal, V16, 1975, p. 121-131.

7. U. S. Department of Commerce. Tables of Chebyshev Polynomials $S_n(x)$ and $C_n(x)$, 1951.

8. G. Dahlquist and A. Bjorck, Numerical Methods translated by N. Anderson, Prentice-Hall, 1974.

9. Ibid.

10. International Business Machine, IBM System/360 and System/370, Fortran IV Language, 1972.

11. C. J. Pfeifer, PDQ-7 Reference Manual II, WAPC-TM-947(L), February 1971.

12. Personal Communication from Marvin Smith, Virginia Power and Electric Company, Richmond, Virginia, 15 August 1976.

APPENDICES

APPENDIX I

GLOSSARY OF SYMBOLS

D_{ip} -- diffusion coefficient for group i in region p(cm).

$\phi_{ip}(x,y)$ -- neutron flux for group i in region p at the point x,y (n/cm^2 - sec).

σ_{ipr} -- macroscopic removal cross section for group i in region p (cm^{-1}).

$\sigma_{ipr} \equiv \sigma_{ips} + \sigma_{ipa} + D_{ip}B_z^{\ 2}$

$B_z^{\ 2} = (\pi/H)^2$ where H is the extropolated height of the reactor.

σ_{ips} -- macroscopic scattering cross section out of group i in region p(cm^{-1}).

σ_{ipa} -- macroscopic absorption cross section for group i in region p(cm^{-1}).

σ_{ipf} -- macroscopic fission cross section for group i in region p (cm^{-1}).

ν_{ip} -- average number of neutrons produced per fission for group i in region p.

$\lambda^{-1} = K_{eff}$ -- multiplication factor for the reactor.

$S_{ip}(x,y)$ -- the neutron source density for group i in region p at the point x,y(n/cm^3 - sec).

i -- group number.

1 = fast group.

2 = thermal group.

p -- region number.

$$fp \equiv \frac{\nu_{2p}}{\lambda} \sigma_{2pf}.$$

$$T_{jk}(x,y) \equiv T_j(x)T_k(y).$$

$$C_{ipjk} = \langle \phi_{ip}, T_{jk} \rangle \quad \text{where,}$$

$$\langle T_{ij}, T_{k\ell} \rangle \equiv \int_{-1}^{1} \int_{-1}^{1} \frac{T_{i,j}(x,y)T_{k,\ell}(x,y)\,dxdy}{\sqrt{1-x^2}\quad\sqrt{1-y^2}}$$

$$= \pi^2 \qquad i = j = k = \ell = 0;$$

$$= \pi^2/4 \qquad i = k \neq 0; \quad j = \ell \neq 0;$$

$$= \pi^2/2 \qquad j = \ell \neq 0; \; i = k = 0; \quad \text{or} \quad j = \ell = 0; \; i = k \neq 0;$$

$$= 0 \text{ for all other values of } i,\ j,\ k,\ \ell.$$

APPENDIX II

DEVELOPMENT OF TABLES FOR THE FIRST AND SECOND DERIVATIVES OF CHEBYSHEV POLYNOMIALS OF THE FIRST KIND

When solving differential equations by expansion in orthogonal Chebyshev polynomials, it is necessary to take derivatives of the expansion. While the derivatives can be taken with the aid of some simple relations, it is a repetitive and time-consuming operation. A brief description of the process and tabulated results for the first and second derivatives of polynomials of the first kind are given here.

The Chebyshev polynomials of the first kind $T_n(x)$ are defined as

$$T_n(x) = \cos(n \cos^{-1} x) \qquad -1 \quad x \quad 1 \quad \text{and } m = 0, 1, 2 \ldots ,$$

with the following recurrence relation:

$$T_{n+2}(x) = 2xT_{n+1}(x) - T_n(x)$$

and orthogonality relation:

$$\int_{-1}^{1} \frac{T_m(x)T_n(x)\,dx}{\sqrt{1-x^2}} = \begin{array}{ll} 0 & m \neq n, \\ \pi & m = n = 0, \\ \pi/2 & m = n \neq 0. \end{array}$$

Chebyshev polynomials of the second kind, $U_n(x)$ are defined as

$$U_n(z) = \frac{\sin[(n+1)\cos^{-1} z]}{\sqrt{1-z^2}}$$

with the following recurrence and orthogonality relations:

$$U_{n+2}(z) = 2z\,U_{n+1}(z) - U_n(z)$$

$$\int_{-1}^{1} U_m(z)U_n(z)\sqrt{1-z^2}\,dz = \begin{cases} 0 & m \neq n \\ \pi & m = n = 0 \\ \pi/2 & m = n \neq 0 \end{cases}$$

From these definitions and recurrence relations, one can develop the explicit expressions for the $T_n(x)$ and $U_n(x)$ polynomials.

$$T_0(x) = 1, \qquad T_2(x) = 2x^2-1,$$

$$T_1(x) = x, \qquad T_3(x) = 4x^3-3x;$$

and

$$U_0(x) = 1, \qquad U_{\cancel{3}\,2}(x) = 4x^2-1,$$

$$U_1(x) = 2x, \qquad U_3(x) = 8x^{\cancel{2}\,3}-4x.$$

One can also develop some other necessary relationships:

$$T'_{n+1}(x) = \overset{n+1}{\cancel{n}}U_n(x), \tag{A-1}$$

$$U_{n+1}(x) = xU_n(x) + T_{n+1}(x), \tag{A-2}$$

$$xU_{n+1}(x) = U_{\cancel{n}\,n+2}(x) \cancel{+} - T_{n+2}(x). \tag{A-3}$$

Note that $T_n'(x)$ can be simply expanded in terms of $U_n(x)$. This leads to a mixed set of polynomials in solving differential equations. One can develop a more complicated relation that expresses $T_n'(x)$ in terms of the same set of polynomials from equations A-1, A-2, and A-3:

$$T'_{n+1}(x) = \overset{n+1}{\cancel{n}}U_n(x)$$

$$U_{n+1}(x) = xU_n(x) + T_{n+1}(x), \qquad \text{From A-2}$$

$$xU_{n+1}(x) = U_n(x) + T_{n+2}(x), \qquad \text{From A-3}$$

$$U_{n+2}(x) = U_n(x) + 2T_{n+2}(x).$$

The above recursion relations give the following:

$$U_n(x) = 2[T_n(x) + T_{n-2}(x) + \ldots + T_1(x)], \text{ n odd,}$$

$$U_n(x) = 2[T_n(x) + T_{n-2}(x) + \ldots + T_2(x)] + U_0(x), \text{ n even.}$$

The first and second derivatives of the Chebyshev polynomials are presented in Table A-1. The numbers in this Table are the coefficients of the polynomials given in the top row.

TABLE A-1

FIRST AND SECOND DERIVATIVES FOR CHEBYSHEV POLYNOMIALS

	T_4	T_3	T_2	T_1	T_0
T_5'	10		10		5
T_4'		8		8	
T_3'			6		3
T_2'				4	
T_1'					1
T_5''		80		120	
T_4''			48		32
T_3''				24	
T_2''					4
T_1''					0

APPENDIX III

COEFFICIENT MATRIX FOR EACH GROUP OF EIGHTY FOUR EQUATIONS

The elements in the block matrices shown on pages 47 and 48, are presented on pages 49-57. The blocks having all elements equal to zero are indicated by - .

COEFFICIENT MATRIX

(Rows 1-54)

	1 — 9	10 — 18	19 — 27	28 — 36
1 — 9	A	–	–	–
10 — 18	–	B	–	–
19 — 27	–	–	C	–
28 — 36	–	–	–	D
37 — 42	E	–	–	–
43 — 48	F	G	–	H
49 — 54	I	–	–	J

COEFFICIENT MATRIX

(Rows 55-84)

	1 — 9	10 — 18	19 — 27	28 — 36
55–60	-	K	-	-
61–66	-	L	M	-
67–72	-	-	N	-
73–78	-	-	P	Q
79–84	-	-	-	R

	1	2	3	4	5	6	7	8	9
1	$\sigma_{i,1,r}$		$-4D_{i1}$				$-4D_{i1}$		
2		$\sigma_{i,1,r}$						$-4D_{i1}$	
3			$\sigma_{i,1,r}$						$-4D_{i1}$
4				$\sigma_{i,1,r}$					
5					$\sigma_{i,1,r}$				
6						$\sigma_{i,1,r}$			
7							$\sigma_{i,1,r}$		$-4D_{i1}$
8								$\sigma_{i,1,r}$	
9									$\sigma_{i,1,r}$

Region A

	10	11	12	13	14	15	16	17	18
10	$\sigma_{i,2,r}$		$-4D_{i2}$				$-4D_{i2}$		
11		$\sigma_{i,2,r}$						$-4D_{i2}$	
12			$\sigma_{i,2,r}$						$-4D_{i2}$
13				$\sigma_{i,2,r}$					
14					$\sigma_{i,2,r}$				
15						$\sigma_{i,2,r}$			
16							$\sigma_{i,2,r}$		$-4D_{i2}$
17								$\sigma_{i,2,r}$	
18									$\sigma_{i,2,r}$

Region B

	19	20	21	22	23	24	25	26	27
19	$\sigma_{i,3,r}$		$-4D_{i3}$				$-4D_{i3}$		
20		$\sigma_{i,3,r}$						$-4D_{i3}$	
21			$\sigma_{i,3,r}$						$-4D_{i3}$
22				$\sigma_{i,3,r}$					
23					$\sigma_{i,3,r}$				
24						$\sigma_{i,3,r}$			
25							$\sigma_{i,3,r}$		$-4D_{i3}$
26								$\sigma_{i,3,r}$	
27									$\sigma_{i,3,r}$

Region C

	28	29	30	31	32	33	34	35	36
28	$\sigma_{i,4,r}$		$-4D_{i4}$				$-4D_{i4}$		
29		$\sigma_{i,4,r}$						$-4D_{i4}$	
30			$\sigma_{i,4,r}$						$-4D_{i4}$
31				$\sigma_{i,4,r}$					
32					$\sigma_{i,4,r}$				
33						$\sigma_{i,4,r}$			
34							$\sigma_{i,4,r}$		$-4D_{i4}$
35								$\sigma_{i,4,r}$	
36									$\sigma_{i,4,r}$

Region D

	1	2	3	4	5	6	7	8	9
37		1	-4						
38					1	-4			
39								1	-4
40				1			-4		
41					1			-4	
42						1			-4

Region E

	1	2	3	4	5	6	7	8	9
43	1	1	1						
44				1	1	1			
45							1	1	1
46	1			1			1		
47		1			1			1	
48			1			1			1

Region F

	10	11	12	13	14	15	16	17	18
43	-1	1	-1						
44				-1	1	-1			
45							-1	1	-1
46									
47									
48									

Region G

	28	29	30	31	32	33	34	35	36
43									
44									
45									
46	-1			1			-1		
47		-1			1			-1	
48			-1			1			-1

Region H

	1	2	3	4	5	6	7	8	9
49		D_{i1}	$4D_{i1}$						
50					D_{i1}	$4D_{i1}$			
51								D_{i1}	$4D_{i1}$
52				D_{i1}			$4D_{i1}$		
53					D_{i1}			$4D_{i1}$	
54						D_{i1}			$4D_{i1}$

Region I

	28	29	30	31	32	33	34	35	36
49									
50									
51									
52				$-D_{i4}$			$4D_{i4}$		
53					$-D_{i4}$			$4D_{i4}$	
54						$-D_{i4}$			$4D_{i4}$

Region J

	10	11	12	13	14	15	16	17	18
55				1			-4		
56					1			-4	
57						1			-4
58	1	1	1						
59				1	1	1			
60							1	1	1

Region K

	10	11	12	13	14	15	16	17	18
61	1			1			1		
61		1			1			1	
62			1			1			1
63				D_{i2}			$4D_{i2}$		
64					D_{i2}			$4D_{i2}$	
65						D_{i2}			$4D_{i2}$

Region L

	19	20	21	22	23	24	25	26	27
61	-1			1			-1		
62		-1			1			-1	
63			-1			1			-1
64				$-D_{i3}$			$4D_{i3}$		
65					$-D_{i3}$			$4D_{i3}$	
66						$-D_{i3}$			$4D_{i3}$

Region M

	19	20	21	22	23	24	25	26	27
67	1	1	1						
68				1	1	1			
69							1	1	1
70	1			1			1		
71		1			1			1	
72			1			1			1

Region N

	19	20	21	22	23	24	25	26	27
73	-1	1	-1						
74				-1	1	-1			
75							-1	1	1
76		$-D_{i3}$	$4D_{i3}$						
77					$-D_{i3}$	$4D_{i3}$			
78								$-D_{i3}$	$4D_{i3}$

Region P

	28	29	30	31	32	33	34	35	36
73	1	1	1						
74				1	1	1			
75							1	1	1
76		D_{i4}	$4D_{i4}$						
77					D_{i4}	$4D_{i4}$			
78								D_{i4}	$4D_{i4}$

Region Q

	28	29	30	31	32	33	34	35	36
79	1			1			1		
80		1			1			1	
81			1			1			1
82		1	-4						
83					1	-4			
84								1	-4

Region R

The letter O is not used for a region to avoid confusion. All blank elements are zero.

APPENDIX IV

GROUPING FOR REDUCED SYSTEM MATRIX

There is another plane of symmetry in each problem along the line $x = y$. Therefore, the C_{ipjk} in Regions 2 and 4 are identical. Also, within Regions 1 and 3, $C_{ipjk}T_j(x)T_k(y) \equiv C_{ipkj}T_k(x)T_j(y)$ for each group. Thus, the algebraic problem is reduced to a system of 45 equations in 21 unknowns.

The elements in the block matrices for the reduced system of equations shown on page 59 are presented on pages 60-63.

COEFFICIENT MATRIX FOR REDUCED SYSTEM

	1 … 11	12 … 21
1 … 23	A	C
24 … 45	B	D

A

	1	2	3	4	5	6	7	8	9	10	11
1	$\sigma_{i,1,r}$		$-8D_{i1}$								
2		$\sigma_{i,1,r}$			$-4D_{i1}$						
3			$\sigma_{i,1,r}$			$-4D_{i1}$					
4				$\sigma_{i,1,r}$							
5					$\sigma_{i,1,r}$						
6						$\sigma_{i,1,r}$					
7							$\sigma_{i,2,r}$		$-4D_{i2}$		
8								$\sigma_{i,2,r}$			
9									$\sigma_{i,2,r}$		
10										$\sigma_{i,2,r}$	
11											$\sigma_{i,2,r}$
12											
13											
14											
15											
16											
17											
18											
19											
20											
21											
22		1	-4								
23				1	-4						

All Blank Terms Are Zero

B

	1	2	3	4	5	6	7	8	9	10	11
24					1	-4					
25	1	1	1				-1	1	-1		
26		1		1	1					-1	1
27		1		1	1						
28		D_{i1}	$4D_{i1}$					$-D_{i2}$	$4D_{i2}$		
29				D_{i1}	$4D_{i1}$						$-D_{i2}$
30					D_{i1}	$4D_{i1}$					
31											
32											
33											
34							1	1	1		
35										1	1
36											
37							1			1	
38								1			1
39									1		
40										D_{i2}	
41											D_{i2}
42											
43											
44											
45											

All Blank Terms Are Zero

C

	12	13	14	15	16	17	18	19	20	21
1										
2										
3										
4										
5										
6										
7		$-4D_{i2}$								
8			$-4D_{i2}$							
9				$-4D_{i2}$						
10	$-4D_{i2}$									
11										
12	$\sigma_{i,2,r}$									
13		$\sigma_{i,2,r}$		$-4D_{i2}$						
14			$\sigma_{i,2,r}$							
15				$\sigma_{i,2,r}$						
16					$\sigma_{i,3,r}$		$-8D_{i3}$			
17						$\sigma_{i,3,r}$			$-4D_{i3}$	
18							$\sigma_{i,3,r}$			
19								$\sigma_{i,3,r}$		
20									$\sigma_{i,3,r}$	
21										$\sigma_{i,3,r}$
22		1	-4							
23				1	-4					

All Blank Terms Are Zero

D

	12	13	14	15	16	17	18	19	20	21
24										
25										
26	-1									
27		-1	1	-1						
28										
29	$4D_{i2}$									
30			$-D_{i2}$	$4D_{i2}$						
31		-4								
32			-4							
33	1			-4						
34										
35	1									
36		1	1	1						
37		1			-1	1	-1			
38			1			-1		1	1	
39	1			1			-1		1	1
40		$4D_{i2}$				$-D_{i3}$	$4D_{i3}$			
41			$4D_{i2}$					$-D_{i3}$	$4D_{i3}$	
42	D_{i2}			$4D_{i2}$					$-D_{i3}$	$4D_{i3}$
43					1	1	1			
44						1		1	1	
45							1		1	1

All Blank Terms Are Zero

APPENDIX V

INPUT DATA FOR PROGRAM

Card 1 Format (20A4)
Title Card

Card 2 Format (F10.3)
Length of side of 1 region in cm

Cards 3-10 Format (4 F10.5, 2I4)

σ_{ipr}	Total Removal Cross Section	cm^{-1}
D_{ip}	Diffusion Coefficient	cm
σ_{ipf}	Fission Cross Section	cm^{-1}
σ_{ips}	Scattering Cross Section	cm^{-1}
i	Group Number	
p	Region Number	

Cards 3-10 must be present and may be in any order.

APPENDIX VI

COMPUTER PROGRAM SOURCE LISTING

```
      COMMON /BLOCK1/ A(2,45,22)
      COMMON /BLOCK2/ RS(2,21)
      COMMON /BLOCK3/ S(2,45)
      COMMON /BLOCK6/ SF(2,4),SS(2,4)
      COMMON /BLOCK7/ PHI(3,2,5,10)
      DIMENSION X(21), EV(2), ET(2), EH1(2,21), IGC(4), EH2(2,21)
      DATA EH1,EH2/84*0./
      DATA ET,EV/4*1./
      DATA IGC/1,2,3,4/
C
      EIGEN(A,B,C)=4.*A-1.333333*(B)+.444444*C
C     EIGEN IS A FUNCTION THAT INTEGRATES THE FLUX OVER A REGION
C
C     EH1  STORAGE VECTOR FOR THE PRESENT AND PAST SOLN FOR GROUP 1
C
C     EH2  STORAGE VECTOR FOR THE PRESENT AND PAST SOLN FOR GROUP 2
C
C      PHI(IR,IG,X,Y)  THE FLUX AT POINT X,Y IN REGION IR FOR GROUP IG
C
C     IM  THE MAXIMUN NUMBER OF ITERATIONS
      IM=20
C
C
C     IG THE GROUP NUMBER
C     1   FAST GROUP
C     2  THERMAL GROUP
C
C
C     PCE  MAXIMUN PERCENTAGE CHANGE IN EIGENVALUE
      PCE=0.001
C
C     NP = # OF TERMS IN EXPANSION
```

```
C
C     NU = # OF UNKNOWNS
C
C     S(K,J) = J TH SOURCE TERM FOR GROUP K
C
C     IP  ITERATION COUNTER
      IP=0
C
C     RS(IG,J) J TH FLUX TERM FOR GROUP IG
C
C     RS(N,7)=MU0
C
C     RS(N,8)=MU1
C
C     RS(N,9)=MU2
C
C     RS(N,10)=MU3
C
C     RS(N,11)=MU4
C
C     RS(N,12)=MU5
C
C     RS(N,13)=MU6
C
C     RS(N,14)=MU7
C
C     RS(N,15)=MU8
C
C
C     IFC IS A PRINT FLAG FOR THE FLUX DISTRIBUTIONS
      IFC=0
C         0 = SKIP PRINT FOR FLUX DISTRIBUTIONS
```

```
C          1 = PRINT FLUX DISTRIBUTIONS
C
C      ACC IS THE ACCELERATION FACTOR
       ACC=0.3
       ACC=0.4
C
C      EH2  VECTOR TO HOLD THE PAST AND PRESENT SOLUTIONS FOR GROUP 2
C
C      EH1  VECTOR TO HOLD THE PAST AND PRESENT SOLUTIONS FOR GROUP 1
C
C      IGP IS A PRINT FLAG FOR THE COEFFICIENTS
       IGP=0
C          0 = PRINT COEFFICIENTS EACH ITERATION
C          1 = SKIP PRINT FOR EACH ITERATION
C
       NU=21
       IPT=0
C      SET INITIAL GUESS TO 2. FOR FIRST TERM IN EACH REGION
       RS(1,1)=2.
       RS(1,7)=2.
       RS(1,15)=2.
C
       CALL INIT
C      CALL INIT TO SET UP BOTH GROUPS A MATRIX
       WRITE (6,19) ACC
1      IF (IP.LT.1) GO TO 3
       IF (S(1,1).GT.0.1.AND.S(1,1).LT.1.) GO TO 3
C       SOURCE NORMALATION CHECK
C
       WRITE (6,20)
       SAVE=RS(1,1)
       DO 2 IG=1,2
```

```
      DO 2 J=1,21
2     RS(IG,J)=RS(IG,J)/SAVE
3     EGV=EV(1)
      CALL UP1 (EGV)
C     UPDATE GROUP ONE SOURCE
C
C     SOLVE FOR GROUP ONE FLUX
      CALL HAM (1,IP)
C
      CALL UP2
C     UPDATE GROUP TWO SOURCE
C
C     SOLVE FOR GROUP TWO FLUX
      CALL HAM (2,IP)
C
      IP=IP+1
      WRITE (6,23) IP
      IF (IGP.GT.0) GO TO 5
      DO 4 IG=1,2
      WRITE (6,24)
      WRITE (6,18) (RS(IG,I),I=1,6),IG,IGC(1)
      WRITE (6,29) (RS(IG,I),I=7,15),IG,IGC(2)
4     WRITE (6,18) (RS(IG,I),I=16,21),IG,IGC(3)
5     CONTINUE
      IF (IFC.GT.0) CALL FLUX
C
C     EIGENVALUE SECTION
C
      ET(2)=ET(1)
C     MOVE THE OLD FLUX INTERGRAL INTO ET(2)
C
      EG=SF(2,1)*EIGEN(RS(2,1),RS(2,3),RS(2,6))
```

```
C     CALCULATE THE NEW FLUX INTERGRAL
C
      ET(1)=EG
      EV(2)=EV(1)
C     MOVE THE OLD EIGENVALUE INTO EV(2)
C
      EV(1)=ET(1)/ET(2)
C     CALCULATE THE NEW EIGENVALUE
C
      PC=100.*(EV(1)-EV(2))/EV(2)
C     CALCULATE THE CHANGE IN THE EIGENVALUE
C
      IF (IP.LT.2) GO TO 6
C     SKIP THE TEST FOR CONVERGENCE UNTIL 2 ITERATIONS
C
      WRITE (6,26) PC
      WRITE (6,27) EV(2),EV(1)
      IF (IP.GT.IM) GO TO 10
C     TEST FOR CONVERGENCE OF EIGENVALUE
C
      IF (ABS(PC).LE.PCE) GO TO 9
C
C     ACCELERATION SECTION
C
6     DO 7 J=1,21
C
C     MOVE THE OLD SOLUTION BACK 1
      EH1(1,J)=EH1(2,J)
      EH2(1,J)=EH2(2,J)
C
C     MOVE THE NEW SOLUTION IN
      EH1(2,J)=RS(1,J)
```

```
7       EH2(2,J)=RS(2,J)
C
        IF (IP.LE.1) GO TO 1
C       DO NOT ACCELERATE UNTIL 2 ITERATIONS
C
        DO 8 J=1,21
        RS(1,J)=EH1(2,J)+ACC*(EH1(2,J)-EH1(1,J))
        RS(2,J)=EH2(2,J)+ACC*(EH2(2,J)-EH2(1,J))
C       CALCILATE THE NEW SOLUTION FROM THE PAST 2 SOLUTIONS
C
8       CONTINUE
C
C       END ACCLERATION SECTION
C
        GO TO 1
9       WRITE (6,28)
10      CONTINUE
C
C       OUTPUT FLUX
C
        WRITE (6,21)
        CALL FLUX
        DO 13 IG=1,2
        PN=0.
        DO 11 L=1,3
        DO 11 J=1,5
        DO 11 K=1,5
        IF (PHI(L,IG,J,K).LE.PN) PN=PHI(L,IG,J,K)
C        FIND MIN FLUX
C
11      CONTINUE
        DO 12 L=1,3
```

```
      DO 12 J=1,5
      DO 12 K=1,5
12    PHI(L,IG,J,K)=PHI(L,IG,J,K)-PN
13    CONTINUE
      DO 14 IG=1,2
      DO 14 K=1,5
      PHI(2,IG,K,5)=0.
      PHI(3,IG,K,5)=0.
14    PHI(3,IG,5,K)=0.
      DO 17 IG=1,2
      WRITE (6,22) IG
      DO 15 J=1,5
      K=6-J
15    WRITE (6,25) (PHI(2,IG,L,K),L=1,5),(PHI(3,IG,L,K),L=1,5)
      WRITE (6,30)
      DO 16 J=1,5
      K=6-J
16    WRITE (6,25) (PHI(1,IG,L,K),L=1,5),(PHI(2,IG,K,L),L=1,5)
17    WRITE (6,30)
      STOP
C
C
18    FORMAT (1H ,6F10.4,30X,2I10)
19    FORMAT (1H ,5HACC =,F10.5)
20    FORMAT (1H ,17HSOURCE NORMALIZED)
21    FORMAT (1H ,///,46X,23HFINAL FLUX DISTRIBUTION,/////)
22    FORMAT (1H ,53X,5HGROUP,I3)
23    FORMAT (1H ,///,20X,28H ******* ITERATION # *******,I6)
24    FORMAT (1H ,/,33H THIS IS THE SOLUTION VECTOR FOR ,65X,5HGROUP,4X,
     16HREGION,/)
25    FORMAT (1H ,5F10.5,10X,5F10.5)
26    FORMAT (1H ,33HTHE % CHANGE IN THE EIGENVALUE IS,F10.5)
```

```
27    FORMAT (1H ,22HTHE OLD EIGENVALUE WAS,F10.6,/,23H THE NEW EIGENVAL
     1UE  IS,F10.6)
28    FORMAT (1H ,42H******** EIGENVALUE CONVERGENCE***********)
29    FORMAT (1H ,9F10.4,2I10)
30    FORMAT (1H ,1H )
      END
      FUNCTION PRDR (J,M,RS)
      DIMENSION X(5), Y(10), RS(6)
      DATA X/-1.,-.5,0.,.5,1./
      DATA Y/-1.,-.9,-.8,-.7,-.5,-.2,0.,.2,.5,1.0/
C
C      THIS FUNCTION EVALUATES THE FLUX AT POINT X(J) AND Y(M) FOR THE
C      FORM PHI(X,Y)=SUM I=1,9 OF A(I)T(X)H(Y) WHERE T(X) IS THE CHEBY-
C      SHEV POLYNOMIAL OF DEGREE N AND H(Y) IS THE POLYNOMIAL FOR Y
C
C     NOTE THAT:
C
C     T0(X)=1
C     T1(X)=X
C
C      USED FOR REGIONS I AND III
C
      S(U)=2.*U*U-1.
C     S(U) IS A STATEMENT FUNCTION THAT EVALUATES T2(X)
      RS(1)=RS(1)*0.25
      RS(2)=RS(2)*0.5
      RS(3)=RS(3)*0.5
C
      PRDR=RS(1)+RS(2)*(Y(M)+X(J))+RS(3)*(S(Y(M))+S(X(J)))+RS(4)*X(J)*Y(
     1M)+RS(5)*X(J)*S(Y(M))+RS(6)*(S(X(J))*S(Y(M)))+RS(5)*Y(M)*S(X(J))
      RETURN
      END
```

```
      SUBROUTINE HAM (IG,IP)
      REAL*8 R,DKR,RKJ,Y,DK
      DIMENSION R(2,21,22), B(45), Y(21), DKR(2,21)
      COMMON /BLOCK1/ A(2,45,22)
      COMMON /BLOCK2/ RS(2,21)
      COMMON /BLOCK3/ S(2,45)
      DATA IGP/0/
      DATA R/924*0.0D0/
C
C     THIS SUBROUTINE SOLVES THE OVERDETERMINED SYSTEM OF EQUATIONS
C
C           AX = B
C
C      OF N UNKNOWNS WITH M EQUATIONS BY MODIFIED GRAM-SCHMIDT METHOD
C
C
C              DIMENSIONS
C
C     A DIMENSION (M,N+1)  AUGMENTED COEFFICIENT MATRIX
C
C     R(IG,N,N+1)
C
C     A(IG,M,N+1)
C
C     B(M)
C
C     Y(N)
C
C     DKR(IG,N)
C
C     IG SELECTS THE GROUP TO SOLVE FOR
C
```

```
C
      N=21
      M=45
      N1=N-1
      N2=N+1
C
      DO 1 K=1,M
1     B(K)=S(IG,K)
C     LOAD B WITH THE SOURCE FOR THAR GROUP
C
      IF (IGP.GE.2) GO TO 8
      IGP=IGP+1
C     SKIP THE FORMATION OF R MATRIX AFTER THE FIRST ITERATION
C
      DO 2 I=1,M
2     A(IG,I,N+1)=B(I)
C     FORM THE AUGMENTED A MATRIX
C
C
      DO 3 K=1,N
3     R(IG,K,K)=1.0D0
      DO 6 K=1,N
      DK=0.0D0
      DO 4 I=1,M
4     DK=DK+A(IG,I,K)*A(IG,I,K)
      DKR(IG,K)=DK
C     FORM THE SUM OF THE SQUARES OF COLUMN I
C
C
      DO 6 JM1=K,N1
      J=JM1+1
      RKJ=0.0D0
```

```
      DO 5 I=1,M
5     RKJ=RKJ+A(IG,I,K)*A(IG,I,J)
C     FORM THE SUM OF THE TERMS FROM THE J TH AND K TH COLUMN
C
      RKJ=RKJ/DK
      R(IG,K,J)=RKJ
      DO 6 I=1,M
6     A(IG,I,J)=A(IG,I,J)-RKJ*A(IG,I,K)
      WRITE (6,15)
      DO 7 K=1,N
7     WRITE (6,14) (R(IG,K,J),J=1,N)
C
C     THIS IS THE ENTRY POINT AFTER THE FIRST ITERATION
8     CONTINUE
C
      DO 10 K=1,N
      RKJ=0.0D0
      DO 9 I=1,M
9     RKJ=RKJ+A(IG,I,K)*B(I)
10    R(IG,K,N2)=RKJ/DKR(IG,K)
C
C     SOLVE THE TRIANGULAR SYSTEM OF EQUATIONS FOR Y AT THIS POINT
      DO 11 I=1,N
11    Y(I)=R(IG,I,N+1)
      DO 12 IB=2,N
      I=N-IB+1
      IP1=I+1
      DO 12 K=IP1,N
12    Y(I)=Y(I)-R(IG,I,K)*Y(K)
C
C
      DO 13 K=1,N
```

```
13    RS(IG,K)=Y(K)
C     MOVE THE ANSWERS BACK INTO RS FOR GROUP IG
C
      RETURN
C
14    FORMAT (1H ,11F11.5)
15    FORMAT (1H ,//,10X,10H   R MATRIX)
      END
      SUBROUTINE FLUX
      COMMON /BLOCK7/ PHI(3,2,5,10)
      COMMON /BLOCK2/ RS(2,21)
      DIMENSION TS(9), TR(6)
C
C     THIS SUBROUTINE EVALUATES THE FLUX IN EACH REGION FOR THE MESH
C
C     TS  WORK VECTOR TO STORE THE COEFFICIENTS FOR REGION II
C
C     TR  WORK VECTOR TO STORE THE COEFFICIENTS FOR REGIONS I OR III
C
C     NX IS THE NUMBER OF POINTS THE FUNCTION IS EVALUATED AT
      NX=5
      NY=10
C
      DO 7 IG=1,2
C     L IS THE NUMBER OF REGIONS
      DO 1 J=1,6
1     TR(J)=RS(IG,J)
      DO 2 J=1,NX
      DO 2 K=1,NY
2     PHI(1,IG,J,K)=PRDR(J,K,TR)
C     CALCULATE THE FLUX FOR REGION I
C
```

```
      DO 3 J=1,6
3     TR(J)=RS(IG,J+15)
      DO 4 J=1,NX
      DO 4 K=1,NY
4     PHI(3,IG,J,K)=PRDR(J,K,TR)
C     CALCULATE THE FLUX FOR REGION III
C
      DO 5 N=1,9
5     TS(N)=RS(IG,N+6)
      DO 6 J=1,NX
      DO 6 K=1,NY
6     PHI(2,IG,J,K)=PRD(J,K,TS)
C     CALCULATE THE FLUX FOR REGION II
C
7     CONTINUE
      DO 10 IG=1,2
      WRITE (6,11) IG
      DO 8 J=1,NY
      K=NY+1-J
8     WRITE (6,12) (PHI(2,IG,L,K),L=1,5),(PHI(3,IG,L,K),L=1,5)
      WRITE (6,13)
      K=NY+1-J
      DO 9 J=1,NY
9     WRITE (6,12) (PHI(1,IG,L,K),L=1,5)
10    WRITE (6,13)
      RETURN
C
C
11    FORMAT (1H ,8HMAP FLUX,I6)
12    FORMAT (1H ,5F10.3,10X,5F10.3)
13    FORMAT (1H ,1H )
      END
```

```
      SUBROUTINE INIT
      COMMON /BLOCK1/ A(2,45,22)
      COMMON /BLOCK6/ SF(2,4),SR(2,4)
      DIMENSION D(2,3), SA(2,3), SS(2,3)
C
C     THIS SUBROUTINE SETS UP THE A MATRIX FOR GROUP 1 AND 2
C
C     FIRST THE MATERIAL AND REACTOR SPECIFICATIONS ARE READ IN
C     THEN THE DIMENSIONS ARE NORMALIZED FROM CM TO -1 TO +1
C
      CALL HEADER
C     HEADER WRITES OUT THE COORDINATE SYSTEM AND REGION LAYOUT
C
C
C     H  LENGTH OF REGION IN CM
C
C     SR(IG,N)  MACROSCOPIC REMAVAL CROSS SECTION FOR GROUP IG IN REGION
C
C     SF(IG,N) MACROSCOPIC FISSION CROSS SECTION  FOR GROUP IG IN REGION
C
C     SS(IG,N)  MACROSCOPIC SCATTERING CROSS SECTION FOR GROPP IF IN REG
C
C     SA(IG,N)  MACROSCOPIC ABSORPTION CROSS SECTION FOR GROUP IG IN REG
C
C     D(IG,N)  DIFFUSION COEFFICIENT FOR GROUP IG IN REGION N
C
      WRITE (6,12)
      READ (8,10) H
      DO 1 J=1,6
C     READ IN THE CROSS SECTIONS AND DIFFUSION COEFFICIENT FOR
C     GROUP IG AND REGION N
      READ (8,13) SRI,DI,SFI,SSI,IG,N
```

```
C
      SR(IG,N)=SRI
      D(IG,N)=DI
      SF(IG,N)=SFI
      SS(IG,N)=SSI
C
C     CALCULATE THE ABSORBTION CROSS SECTION
1     SA(IG,N)=SRI-SSI
C
      DO 2 IG=1,2
      DO 2 N=1,3
2     WRITE (6,14) SR(IG,N),D(IG,N),SF(IG,N),SS(IG,N),SA(IG,N),IG,N
C     WRITE OUT THE CROSS SECTIONS FOR THE REGIONS AND GROUPS
C
      WRITE (6,11) H
C
C     NORMALIZE FROM CM TO -1 TO +1
C
      DO 3 IG=1,2
      DO 3 N=1,3
      D(IG,N)=D(IG,N)/H
      SR(IG,N)=SR(IG,N)*H
      SF(IG,N)=SF(IG,N)*H
      SS(IG,N)=SS(IG,N)*H
3     SA(IG,N)=SR(IG,N)-SS(IG,N)
      DO 8 I=1,2
      DO 4 J=1,6
C
C     INPUT REMOVAL CROSS SECTIONS FOR REGION I
      A(I,J,J)=SR(I,1)
C
C     INPUT REMOVAL CROSS SECTIONS FOR REGION III
```

```
4     A(I,J+15,J+15)=SR(I,3)
C
      DO 5 J=1,9
C
C     INPUT REMOVAL CROSS SECTIONS FOR REGION II
5     A(I,J+6,J+6)=SR(I,2)
C
      D1=-4.*D(I,1)
C
C     INPUT LINES 1 - 3
      A(I,1,3)=2.*D1
      A(I,2,5)=D1
      A(I,3,6)=D1
C
C     INPUT LINES 7 - 13
      D1=-4.*D(I,2)
      A(I,7,9)=D1
      A(I,7,13)=D1
      A(I,8,14)=D1
      A(I,9,15)=D1
      A(I,10,12)=D1
      A(I,13,15)=D1
C
C     INPUT LINES 16 - 18
      D1=-4.*D(I,3)
      A(I,16,18)=2.*D1
      A(I,17,20)=D1
      A(I,18,21)=D1
C
C     INPUT LINES 22 - 24
      A(I,22,2)=1.
      A(I,22,3)=-4.
```

```
      A(I,23,4)=1.
      A(I,23,5)=-4.
      A(I,24,5)=1.
      A(I,24,6)=-4.
C
C     INPUT LINES 25 - 27
      A(I,25,1)=1.
      A(I,25,2)=1.
      A(I,25,3)=1.
      A(I,26,2)=1.
      A(I,26,4)=1.
      A(I,26,5)=1.
      A(I,27,3)=1.
      A(I,27,5)=1.
      A(I,27,6)=1.
C
      DO 6 J=1,3
      AB=(-1)**J
C
C     INPUT LINES 25 - 27
      A(I,25,J+6)=AB
      A(I,26,J+9)=AB
      A(I,27,J+12)=AB
C
C     INPUT LINES 34 - 36
      A(I,34,J+6)=1.
      A(I,35,J+9)=1.
      A(I,36,J+12)=1.
C
C     INPUT LINE 37
      A(I,37,J+15)=AB
C
```

```
C      INPUT LINE 43                                          E 129
6      A(I,43,J+15)=1.                                        E 130
       DO 7 J=1,3                                             E 131
C                                                             E 132
C      INPUT LINES 28- 30                                     E 133
       A(I,J+27,8+(J-1)*3)=-D(I,2)                            E 134
       A(I,J+27,9+(J-1)*3)=4.*D(I,2)                          E 135
C                                                             E 136
C      INPUT LINES 31 - 33                                    E 137
       A(I,J+30,J+9)=1.                                       E 138
       A(I,J+30,J+12)=-4.                                     E 139
C                                                             E 140
C      INPUT LINES 40 - 42                                    E 141
       A(I,J+39,J+9)=D(I,2)                                   E 142
       A(I,J+39,J+12)=4.*D(I,2)                               E 143
C                                                             E 144
C      INPUT LINES 37 39                                      E 145
       A(I,J+36,J+6)=1.                                       E 146
       A(I,J+36,J+12)=1.                                      E 147
7      A(I,J+36,J+9)=1.                                       E 148
C                                                             E 149
       D1=D(I,1)                                              E 150
       A(I,28,2)=D1                                           E 151
       A(I,28,3)=4.*D1                                        E 152
C                                                             E 153
C      INPUT LINE 29                                          E 154
       A(I,29,4)=D1                                           E 155
       A(I,29,5)=4.*D1                                        E 156
C                                                             E 157

       A(I,30,5)=D1                                           E 159
       A(I,30,6)=4.*D1                                        E 160
```

```
C
C     INPUT LINE 38
      A(I,38,17)=-1.
      A(I,38,19)=1.
      A(I,38,20)=-1.
C
C     INPUT LINE 39
      A(I,39,18)=-1.
      A(I,39,20)=1.
      A(I,39,21)=-1.
C
      D1=D(I,3)
C
C     INPUT LINE 40
      A(I,40,17)=-D1
      A(I,40,18)=4.*D1
C
      A(I,41,19)=-D1
      A(I,41,20)=4.*D1
C
C     INPUT LINE 42
      A(I,42,20)=-D1
      A(I,42,21)=4.*D1
C
C     INPUT LINE 44
      A(I,44,17)=1.
      A(I,44,19)=1.
      A(I,44,20)=1.
C
C     INPUT LINE 45
      A(I,45,18)=1.
      A(I,45,20)=1.
```

```
      A(I,45,21)=1.
C
8     CONTINUE
      X=1.
      Y=1.
      X=0.25
      Y=0.5
      DO 9 J=1,45
      A(I,J,1)=A(I,J,1)*X
      A(I,J,2)=A(I,J,2)*Y
      A(I,J,3)=A(I,J,3)*Y
      A(I,J,7)=A(I,J,7)*X
      A(I,J,8)=A(I,J,8)*Y
      A(I,J,9)=A(I,J,9)*Y
      A(I,J,10)=A(I,J,10)*Y
      A(I,J,13)=A(I,J,13)*Y
      A(I,J,16)=A(I,J,16)*X
      A(I,J,17)=A(I,J,17)*Y
      A(I,J,18)=A(I,J,18)*Y
9     CONTINUE
      CALL PT
      RETURN
C
10    FORMAT (F10.3)
11    FORMAT (1H ,///,19H THE REGION SIZE IS,F6.2,3H CM)
12    FORMAT (1H ,7H     SR,9X,1HD,8X,2HSF,8X,2HSS,8X,2HSA,9X,5HGROUP,5X
     1,6HREGION)
13    FORMAT (4F10.5,2I4)
14    FORMAT (1H ,5F10.5,2I10)
      END
      BLOCK DATA
      COMMON /BLOCK1/ A(2,45,22)
```

```
      COMMON /BLOCK2/ RS(2,21)
      COMMON /BLOCK3/ S(2,45)
      DATA A,S/2070*0./
      DATA RS/2*1.,40*1./
      END
      SUBROUTINE PT
      COMMON /BLOCK1/ A(2,45,22)
C
C     THIS SUBROUTINE PRINTS OUT THE INPUT MATRIX FOR EACH GROUP FOR
C     ALL REGIONS
C
      IP=0
      IP=1
C     IP IS A PRINT FLAG FOR BOTH A MATRICES OUTPUT
C        0 = OUTPUT A MATRIX
C        1 = DO NOT OUTPUT A MATRIX
C
      IF (IP.GE.1) GO TO 2
      DO 1 IG=1,2
      WRITE (6,3) IG
      DO 1 M=1,45
      WRITE (6,4) (A(IG,M,K),K=1,21),M
C
1     CONTINUE
2     CONTINUE
      RETURN
C
C
3     FORMAT (1H ,6HGROUP ,I6,15H   COEFFICIENTS)
4     FORMAT (1H ,21F6.2,I4)
      END
      SUBROUTINE UP2
```

```
      COMMON /BLOCK2/ RS(2,21)
      COMMON /BLOCK6/ SF(2,4),SR(2,4)
      COMMON /BLOCK3/ S(2,45)
C
C     THIS SUBROUTINE CALCULATES THE SOURCE IN GROUP 2 DUE TO DOWN
C     SCATTER FROM GROUP 1
C
      WRITE (6,4)
      DO 1 J=1,6
C     CALCULTE FOR REGION 1
      S(2,J)=RS(1,J)*SR(1,1)
C
C     CALCULATE FOR REGION 2
1     S(2,J+15)=RS(1,J+15)*SR(1,3)
C
      DO 2 J=1,9
C     CALCULATE FOR REGION 3
2     S(2,J+6)=RS(1,J+6)*SR(1,2)
C
      NP=1
      IF (NP.GT.0) GO TO 3
      WRITE (6,5) (S(2,J),J=1,6)
      WRITE (6,5) (S(2,J),J=7,15)
      WRITE (6,5) (S(2,J),J=16,21)
3     CONTINUE
      RETURN
C
C
C
4     FORMAT (1H ,18HSOURCE FOR GROUP 2)
5     FORMAT (1H ,9F12.4)
      END
```

```
      SUBROUTINE UP1 (EGV)
      COMMON /BLOCK2/ RS(2,21)
      COMMON /BLOCK3/ S(2,45)
      COMMON /BLOCK6/ SF(2,4),SR(2,4)
      WRITE (6,4)
      DO 1 J=1,6
C
      S(1,J+15)=RS(2,J+15)*SF(2,3)
1     S(1,J)=RS(2,J)*SF(2,1)
C
C
      DO 2 J=1,9
2     S(1,J+6)=RS(2,J+6)*SF(2,2)
C
      IF (NP.GT.0) GO TO 3
      NP=1
      WRITE (6,5) (S(1,J),J=1,6)
      WRITE (6,5) (S(1,J),J=7,15)
      WRITE (6,5) (S(1,J),J=16,21)
3     CONTINUE
      RETURN
C
C
C
4     FORMAT (1H ,14HSOURCE GROUP 1)
5     FORMAT (1H ,9F12.4)
      END
      SUBROUTINE HEADER
      DIMENSION A(20)
C
C      THIS SUBROUTINE WRITES THE HEADER AT THE TOP OF THE PROGRAM FOR T
C      THE REGION LAYOUT AND COORDINATE SYSTEM FOR EACH REGION
```

```
C
C      READ IN TITLE CARD
       READ (8,6) A
C
C
C      START OF LAYOUT SECTION
C
       WRITE (6,12) A
       WRITE (6,5)
       DO 1 J=1,11
       IF (J.EQ.5) WRITE (6,8)
       IF (J.EQ.6) WRITE (6,10)
1      WRITE (6,7)
       WRITE (6,5)
       DO 2 J=1,11
       IF (J.EQ.5) WRITE (6,8)
       IF (J.EQ.6) WRITE (6,11)
2      WRITE (6,7)
       WRITE (6,5)
C
C      END OF LAYOUT SECTION
C
C      START OF COORDINATE SECTION
C
       WRITE (6,9)
       WRITE (6,13)
       DO 4 J=1,4
       K=6-J
       WRITE (6,15) K,K,K,K,K
       DO 3 L=1,4
3      WRITE (6,14)
4      CONTINUE
```

```
      K=1
      WRITE (6,15) K,K,K,K,K
      WRITE (6,16)
C
C     END OF COORDINATE SECTION
      WRITE (6,17)
      RETURN
C
C
5     FORMAT (1H ,10X,2(21H+--------------------),1H+)
6     FORMAT (20A4)
7     FORMAT (1H ,10X,1H|,20X,1H|,20X,1H|)
8     FORMAT (1H+,20X,6HREGION,14X,6HREGION)
9     FORMAT (1H ,//,23X,18HLAYOUT FOR REGIONS)
10    FORMAT (1H+,22X,2HII,18X,3HIII)
11    FORMAT (1H+,22X,1HI,19X,2HIV)
12    FORMAT (1H1,20X,20A4,///)
13    FORMAT (1H ,//,1H )
14    FORMAT (14X,4(1H|,9X),1H|)
15    FORMAT (10X,I1,12H,1 +--------,I1,9H,2-------,I1,9H,3-------,I1,11
     1H,4--------+,I2,2H,5)
16    FORMAT (1H ,//,18X,33HCOORDINATE SYSTEM FOR EACH REGION)
17    FORMAT (1H ,1H )
      END
      FUNCTION PRD (J,M,RS)
      DIMENSION X(5), Y(10), RS(9)
      DATA Y/-1.,-.9,-.8,-.7,-.5,-.2,0.,.2,.5,1.0/
C
C      THIS FUNCTION EVALUATES THE FLUX AT POINT X(J) AND Y(M) FOR THE
C      FORM PHI(X,Y)=SUM I=1,9 OF A(I)T(X)H(Y) WHERE T(X) IS THE CHEBY-
C      SHEV POLYNOMIAL OF DEGREE N AND H(Y) IS THE POLYNOMIAL FOR Y
C
```

```
C     THIS IS USED FOR REGION II ONLY
C
C     NOTE THAT:
C
C     T0(X)=1
C     T1(X)=X
C
      DATA X/-1.,-.5,0.,.5,1./
      S(U)=2.*U*U-1.
      RS(1)=RS(1)*0.25
      RS(2)=RS(2)*0.5
      RS(3)=RS(3)*0.5
      RS(4)=RS(4)*0.5
      RS(7)=RS(7)*0.5
C     S(U) IS A STATEMENT FUNCTION THAT EVALUATES T2(X)
C
      PRD=RS(1)+RS(2)*Y(M)+RS(3)*S(Y(M))+RS(4)*X(J)+RS(5)*X(J)*Y(M)+RS(6
     1)*X(J)*S(Y(M))+RS(7)*S(X(J))+RS(8)*S(X(J))*Y(M)+RS(9)*S(X(J))*S(Y(
     2M))
      RETURN
      END
```

VITA

The author was born June 17, 1945, in Atlanta, Georga. He grew up in Greenville, North Carolina where he finished his secondary education. After having attended North Carolina State and East Carolina Universities, he obtained his B.S. degree in Applied Physics from East Carolina University. In June 1972, he enrolled in the Nuclear Engineering Graduate Program at Virginia Polytechnic Institute and State University to study for his Ph.D. in Nuclear Engineering. He received his M.S. in 1974. He is married to the former Rebecca Ann Johnson, and they have no children.

Stephen V Pruett

AN APPLICATION OF CHEBYSHEV POLYNOMIALS TO THE SOLUTION

OF A TWO-DIMENSIONAL ELLIPTIC BOUNDARY-VALUE PROBLEM

by

Stephen V. Prewett

(ABSTRACT)

The goal of this dissertation is to investigate the feasibility of using a bivariate Chebyshev polynomial to approximate solutions to the two-dimensional neutron diffusion equation.

The two-dimensional two-group neutron diffusion equations are solved by expanding neutron fluxes in a finite series of Chebyshev polynomials over large regions of a fission reactor. All the equations for the expansion coefficients necessary to satisfy the appropriate boundary conditions for the flux and current for a typical region are developed. The resulting system of algebraic equations is solved, using the power iteration method. Since the system of equations is overdetermined, the Gram-Schmidt method of orthogonalization is used. Calculations are done with the aid of a computer code, CDP, developed as part of this dissertation.

Two different test problems are solved using a first order finite difference computer code, PDQ-7 as a standard for comparison, and CDP. The first problem is a water-reflected square core with homogeneous material properties in each region. This problem is selected to provide a rather severe test of the CDP method in the calculation of large thermal neutron flux peaking at the core-

reflector interface. The second problem is an actual problem solved by a utility for on-line-fuel management in a Pressurized Water Reactor. The reactor core consists of four different fuel assemblies arranged in a checkerboard pattern in the interior of the core.

For the first problem, both PDQ-7 and CDP give about the same fast neutron flux distributions. The eigenvalues calculated by both methods are identical to one part in 4800. Except for a small region near the core-reflector interface, the thermal neutron fluxes within the core differ by less than about 4%. At the core-reflector interface, the difference in thermal fluxes is about 20%. A significant reduction in computer time required for the solution is achieved (0.45 sec for CDP vs 32.6 sec for PDQ-7). It is important to note, however, that the time required to obtain the "standard" solution for comparison, using PDQ-7 is larger than would be needed for a large mesh spacing. Thus, the savings in computer time required to achieve a solution using a PDQ-7 type code giving results comparable to those obtained with the CDP code cannot be inferred directly from these results. In any event, note that the CDP method is indeed economical in both preparation of input data and computer time.

In the second problem, four very large regions are used in the CDP method for the entire reactor. These regions are much larger than those used in a typical finite difference solution for the same problem. Both methods give about the same fast neutron flux distribution. The eigenvalues calculated by the two methods are identical to one part in 2140. Although the CDP method does not show the assembly-to-assembly variation in thermal neutron flux, it gives the same average thermal neutron flux as calculated with PDQ-7 to within 2%, except near the core boundary. Again, a large reduction in computer time is achieved (0.69 sec vs 101 sec).

The feasibility of using Chebyshev polynomial expansions for two-dimensional multi-group diffusion calculations has been demonstrated for these two problems. The method gives, not only very accurate eigenvalues, but also reasonably accurate neutron flux distributions within the reactor core.