

**TÜRKİYE ATOM ENERJİSİ KURUMU**  
**ÇEKMECE NÜKLEER ARAŞTIRMA VE EĞİTİM MERKEZİ**

**Ç.N.A.E.M. A.R- 263**

**IBD-1L : ONE LINE OVERRELAXATION CODE FOR DIFFUSION THEORY  
CALCULATIONS (AN IMPROVED VERSION OF IBD-1 CODE)**

**Ulvi ADALIOĞLU, Raşit TUNCEL**

**Nuclear Engineering Department**

**July 1989**

**P. K. 1, Hava Alanı, İstanbul**



**TÜRKİYE ATOM ENERJİSİ KURUMU**  
**ÇEKMECE NÜKLEER ARAŞTIRMA VE EĞİTİM MERKEZİ**

**Ç.N.A.E.M. A.R- 263**

**IBD-1L : ONE LINE OVERRELAXATION CODE FOR DIFFUSION THEORY  
CALCULATIONS (AN IMPROVED VERSION OF IBD-1 CODE)**

**Ulvi ADALIOĞLU, Raşit TUNCEL**

**Nuclear Engineering Department**

**July 1989**

**P. K. 1, Hava Alanı, İstanbul**

## SUMMARY

### IBD-1L : ONE LINE OVERRELAXATION CODE FOR DIFFUSION THEORY CALCULATIONS (AN IMPROVED VERSION OF IBD-1 CODE)

The block overrelaxation method was applied to IBD code to accelerate the solution of few group diffusion equations in two dimensions some years ago. It was demonstrated that the one line overrelaxation has improved the convergence in inner iteration process greatly. But the CPU time was larger than that of point relaxation case due to the algorithm used at that time.

This new version of the IBD-1 code, tested on VAX/11-750 machine available at Çekmece Nuclear Research Center, has a fast algorithm which reduces CPU time drastically. In fact, it is found that IBD-1L is about 1.41 times faster than IBD code.

## ÖZET

### IBD-1L : DİFUZYON HESAPLARI İÇİN BİR SATIR OVERRELAKSASYON KODU (IBD-1 KODUNUN GELİSTİRİLMİŞ SEKLİ)

Bir kaç yıl önce iki boyutlu, birkaç gruplu difüzyon denklemlerinin çözümünü hızlandırmak için blok overrelaksasyon metodu IBD koduna tatbik edilmişti. Tek satır overrelaksasyonunun iç iterasyonun yakınsamasını büyük ölçüde iyileştirdiği, fakat kullanılan algoritma dolayısıyla harcanan CPU zamanının eskisine göre arttığı görülmüştü.

Çekmece Nükleer Araştırma Merkezi'ndeki VAX/11-750 makinesinde test edilen IBD-1 kodunun yeni versiyonu CPU zamanını büyük miktarda azaltan bir algoritmaya sahiptir. Gerçekten de IBD-1L kodunun, IBD koduna nazaran yaklaşık 1.41 kere daha hızlı olduğu görülmüştür.

## TABLE OF CONTENTS

|                                    | <u>Pages</u> |
|------------------------------------|--------------|
| 1. INTRODUCTION                    | 1            |
| 2. THEORY AND ALGORITHM            | 1            |
| 2.1- Matrix formulation            | 1            |
| 2.2- Algorithm                     | 4            |
| 3. SUMMARY DESCRIPTION OF THE CODE | 5            |
| 3.1- Subroutines                   | 5            |
| 3.2- Input data                    | 6            |
| 4. APPLICATIONS AND RESULTS        | 7            |
| REFERENCES                         | 10           |
| Appendix 1- Input Data Deck        |              |

## TABLES

|                                                               |    |
|---------------------------------------------------------------|----|
| Table 1- Mesh Structure of Test Problem No. 2                 | 8  |
| Table 2- Some Characteristic Results obtained<br>by Two Codes | 11 |

## FIGURES

|                                                |   |
|------------------------------------------------|---|
| Figure 1- Axially Symmetric Test Problem No. 1 | 7 |
| Figure 2- TR-2 Small Core Configuration        | 9 |

## 1. INTRODUCTION

The theory of one line overrelaxation technique was applied to the solution of neutron diffusion equation some years ago (1). The calculations done for several problems proved that the number of inner iterations was drastically reduced. Infact, for a 2D diffusion problem of size 24x29 mesh points the total number of inner iteration was decreased more than 50 % as compared with the results of point relaxation method used in IBD code (2).

The first version of IBD-1L code has rather slow algorithm which resulted in longer CPU times than that of IBD. Hence, it was necessary to improve its algorithm in order to have a practically useful and fast diffusion code. This report summarizes the features of new fast algorithm and corrections of some misprints in the first report. This new version of the code, tested on VAX/11-750 machine, is about 1.41 times faster than IBD code as it was reported in literature (3).

## 2. THEORY AND ALGORITHM

### 2.1- Matrix Formulation

The finite difference form of group diffusion equations in two dimensions is

$$\begin{aligned}
 & -b_{i,j}^g \phi_{i-1,j}^g + a_{i,j}^g \phi_{i,j}^g - c_{i,j}^g \phi_{i+1,j}^g - d_{i,j}^g \phi_{i,j-1}^g \\
 & - e_{i,j}^g \phi_{i,j+1}^g = \frac{X_{ef}^g}{k_{ef}} S_{fi,j}^g + S_{si,j}^g = S_{i,j}^g
 \end{aligned} \tag{2.1}$$

$$g = 1, 2, \dots, G$$

which can be converted into a matrix equation

$$A \underset{g}{\phi}^g = S^g \quad (2.2)$$

This matrix equation is converted into a partitioned matrix equation form:

$$\begin{bmatrix} B_1 & E_1 & & & & \\ & D_2 & B_2 & E_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & E_{M-1} \\ & & & & & D_M & B_M \\ & & & & & & O \end{bmatrix} \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \vdots \\ \phi_{M-1} \\ \phi_M \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ \vdots \\ S_{M-1} \\ S_M \end{bmatrix} \quad (2.3)$$

The definitions of  $B_i$ ,  $D_i$ ,  $E_i$ ,  $\phi_i$ , and  $S_i$  are given in Ref.(1). The matrices  $B_i$  are three diagonal, but  $D_i$  and  $E_i$  matrices are diagonal. The subscript  $M$  shows the number of rows in mesh structure for one line partitioning.

The successive overrelaxation technique applied to the Eq. 2.3 gives the following set of equations:

$$B_i \hat{\phi}_i^{g(m)} = -D_i \phi_{i-1}^{g(m)} - E_i \phi_i^{g(m-1)} + S_i^g \quad (2.4)$$

$$\phi_i^{g(m)} = \omega_b \left[ \hat{\phi}_i^{g(m)} - \phi_i^{g(m-1)} \right] + \phi_i^{g(m-1)}$$

$$i=1,2,\dots,M$$

where  $m$  shows the iteration number.

Eqs. (2.4) can be converted into a more convenient form in order to apply relaxation technique:

$$N_{i,i} \hat{Y}_i^{(m)} = - N_{i,i-1} Y_{i-1}^{(m)} - N_{i,i+1} Y_{i+1}^{(m-1)} + S_i \quad (2.5)$$

$$Y_i^{(m)} = \omega_b \left[ \hat{Y}_i^{(m)} - Y_i^{(m-1)} \right] + Y_i^{(m-1)}$$

for each energy group. Here

$$N_{1,0} = N_{M,M+1} = 0, \text{ and}$$

$B_i$  is factored such that

$$B_i = S_{i,i}^T \cdot S_{i,i}$$

and  $R_{i,i}$  is the diagonal matrix whose diagonal elements are the diagonal entries of  $S_{i,i}$ . And also

$$N_{i,i} = R_{i,i}^{-1} B_i R_{i,i}^{-1}$$

$$N_{i,i-1} = R_{i,i}^{-1} D_i R_{i-1,i-1}^{-1} \quad (2.6)$$

$$N_{i,i+1} = R_{i,i}^{-1} E_i R_{i+1,i+1}^{-1}$$

The new solution vector  $Y_i^{(m)}$  is defined as

$$Y_i^{(m)} = R_{i,i} \phi_i^{(m)} \quad (2.7)$$

## 2.2- Algorithm

The solution of Eq. (2.5) still follows the methodology given previously in Ref. (1), except major changes in programme structure. The new algorithm uses the following properties of matrices:

- i-  $N_{i,i-1}$  and  $N_{i,i+1}$  are diagonal matrices and can be constructed easily. The products of  $N_{i,j}$  and  $Y_j$  are also calculated readily.
- ii- The factorization of  $N_{i,i}^{-1}$  is very simple due to special structure of  $S_{i,i}$  and  $R_{i,i}$  matrices (refer to Ref. (1) for the description of factorization process).
- iii- The application of Square Root Method becomes very simple due to simple matrix structure.

The inner iteration process utilizes the following convergence criteria ( see Ref. (1) or Ref. (3) for the notation used here. ).

1) When

$$\left| \frac{-^{(m)}}{\lambda} - \frac{\lambda^{(m)}}{-} \right| \geq 10^{-5}$$

then a new  $\lambda$  value calculated for a new  $\omega_b$  ; otherwise  $\lambda$  and therefore  $\omega_b$  is not changed.

2) If

$$\left| \bar{\omega}^{(m)} - \omega^{(m)} \right| \geq \frac{2 - \omega^{(m)}}{5}$$

then iteration is continued with same omega value. Otherwise a new optimum omega is calculated.

3) If either  $m = 40$  , or the difference between the point values of fluxes in successive iterations becomes less than EPS4 , then inner iteration stops. The parameter EPS4 is made smaller in first few (here,2) iterations, then it is kept constant.

### 3. SUMMARY DESCRIPTION OF CHANGES IN PROGRAM

A small routine is added to the program in order to obtain average group fluxes over the core region.

#### 3.1- Subroutines

The new form of code has fewer subroutines in order to improve the speed of program. Most of the subprograms, not common in both IBD and IBD-1L is combined in two subroutines.

The following subroutines do whole block acceleration of inner iteration:

COZ2 : finds flux distributions by applying Square Root Method and does one line acceleration by calculating optimum overrelaxation factor  $\omega_b$  .

KATSA : sets up the matrices  $N_{i,i}$ ,  $N_{i,j}$ ,  $R_{i,j}^{-1}$  s, and factorized matrix  $S_{i,i}$ .

### 3.2- Input Data

The first card in input data deck contains four parameters which define the core region for average flux calculation.

1. card : IRRX1, IRRX2, IRHZ1, IRHZ2 4I5

The second card is changed to have two more input data, that is

2. card : EPS2, EPS3, EPS4, NRR 3F10.7, I2

The integer NRR defines options to have full output which includes normalized flux and fission source distributions over the mesh structure. The EPS4 is the convergence factor used for flux approximation and it should be smaller than EPS3 after first few iterations. The initial value of EPS4 is chosen according to the accuracy required for flux calculations. Its initial value can be taken as 0.02, or 0.01 and final value becomes as 0.0002 or 0.0001.

If NRR =  $\begin{cases} 1 & \text{for full output of distributions} \\ 2 & \text{for only relative flux distribution} \end{cases}$

The rest of the data deck is not changed as given in Ref. (1).

Appendix 1 gives whole input data and information on parameters.

#### 4. APPLICATIONS AND RESULTS

The new version of IBD-1 code, that is IBD-1L, has been tested on several problems and compared with the performances of IBD under the same convergence conditions.

i- The first test problem was a fictive reactor shown in Fig. 1. A mesh structure of 10 x 10 points in R - Z geometry and two energy groups have been used in the criticality calculations.

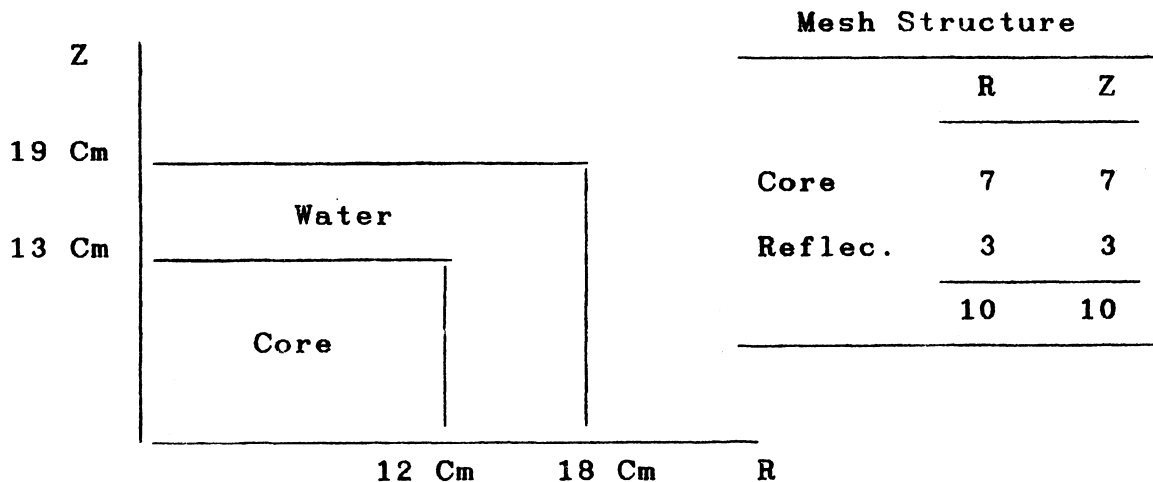


Figure 1- Axially Symmetric Test Problem No. 1

ii- The second test problem is same as the test problem no. 1, except the size of the fictive reactor and mesh structure. The following Table 1 shows the relevant information for reactor geometry.

Table 1 - Geometry and mesh structure of test problem No. 2

| Region    | R axis         |                | Z axis         |                |
|-----------|----------------|----------------|----------------|----------------|
|           | Length<br>(Cm) | Mesh<br>points | Length<br>(Cm) | Mesh<br>points |
| Core      | 38             | 20             | 39             | 20             |
| Reflector | 8              | 4              | 18             | 9              |
|           | 46             | 24             | 57             | 29             |

Two group criticality calculations have been performed for this axially symmetric problem.

iii- The third problem is the TR-2 small core configuration shown in Figure 2. The core is reflected from one side by Berillium blocks, and the other side has two Aluminium blocks located at each corner. The control rod BC-2 is fully inserted while three other rods are out of the core for this test run.

A four group, 24 x 29 mesh point structure in X-Y geometry modelling has been used for criticality calculations.

The results obtained by both codes for test problems have showed that IBD-1L is almost 1.41 times faster than IBD.

Table 2 summarizes the information about the runs of the codes on the test cases. Total numbers of inner iterations are largely reduced by overrelaxation method. The outer iteration numbers remain same for both codes. The values of effective multiplication factors changes in fifth digit. The flux values at

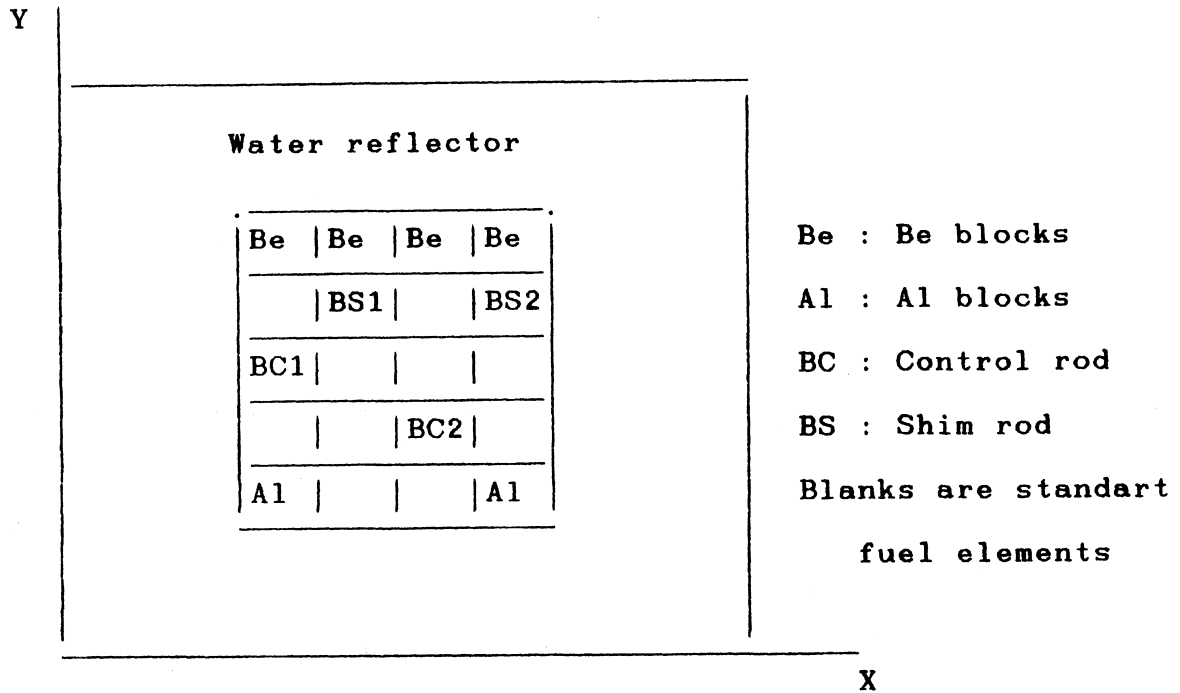


Figure 2- TR-2 Small Core Configuration

mesh points obtained by two programs diverges from each other at different digits depending on the sensitivity factor, EPS4 used for flux convergence. The changes in average group fluxes are at fourth digits. The reduction of EPS4 reduces the gain in CPU time for small size problems, but on the contrary it improves the point and integral values without increasing computation time too much for more complex problems.

Hence, it can be used as an all purpose code in any criticality calculations. Point overrelaxation codes would give more sensitive results in shorter times if they uses flux distributions obtained by IBD-1L code as input flux for large and complex problems.

## REFERENCES

1. U. Adalıoğlu, R. Tuncel, "IBD-1: Two Dimensional, Multigroup Neutron Diffusion Code", ÇNAEM A.R. No. 233, Oct. 1985.
2. R. Tuncel, U. Adalıoğlu, "Code IBD - A Two Dimensional, Multigroup Neutron Diffusion Code", ÇNAEM-R-190, 1978. (in Turkish)
3. L. A. Hageman, "Numerical Methods and Techniques used in the Two-Dimensional Neutron Diffusion Program, PDQ-5", WAPD-TM-364, Feb. 1963.

Table 2- Some Characteristic Results obtained by two codes

| Some values                         | Problem no. 1 |          | Problem no. 2 |          | Problem no. 3 |          |
|-------------------------------------|---------------|----------|---------------|----------|---------------|----------|
|                                     | IBD           | IBD-1L   | IBD           | IBD-1L   | IBD           | IBD-1L   |
| EPS4 (finals are 0.0002 and 0.0001) | -             | 0.02     | 0.01          | -        | -             | 0.01     |
| CPU times                           | 6.75 s        | 6.21 s   | 6.69 s        | 1 m 22 s | 3 m 11 s      | 2 m 29 s |
| No. of outer iterations             | 12            | 11       | 11            | 23       | 11            | 11       |
| Total no. of inner iterations       | 261           | 129      | 152           | 321      | 1819          | 664      |
| $k_{ef}$                            | 1.540839      | 1.540858 | 1.540850      | 1.750985 | 1.050307      | 1.050349 |
| $\bar{\phi}_1$                      | 1.876149      | 1.876805 | 1.876807      | 2.028130 | 1.120786      | 1.123091 |
| $\bar{\phi}_2$                      | -             | -        | -             | -        | 1.207063      | 1.209555 |
| $\bar{\phi}_3$                      | -             | -        | -             | -        | 1.049276      | 1.051458 |
| $\bar{\phi}_{1,2}$                  | 1.520445      | 1.520987 | 1.520982      | 1.602352 | 1.049050      | 1.051233 |
| Ratio of CPU times                  | 1.            | 1/1.09   | 1/1.008       | 1/1.35   | 1.            | 1/1.28   |

# APPENDIX 1- INPUT DATA DECK

|                    |                                                                                               |              |
|--------------------|-----------------------------------------------------------------------------------------------|--------------|
| 1. card :          | IRRX1, IRRX2, IRHZ1, IRHZ2                                                                    | 415          |
| 2. card :          | EPS2, EPS3, EPS4, NRR                                                                         | 3F10.7, 12   |
| 3. card :          | IB, ICAP, IGEO, IAD, IGUC, IBAK, ICEY,<br>KMAX, IMAX, JMAX, KOMSA, IBIS                       | 1213         |
| 4. card :          | (IS1(I), IS2(I), JS1(I), JS2(I), I=1, IB<br>(4+IB).<br>card : KK(K), IMAT(K), K=1, IB         | 1814<br>1814 |
| (4+2IB).<br>card : | DELRI(I), I=1, IMAX                                                                           | 8F10.5       |
| (...)<br>card :    | DELZJ(I), I=1, JMAX                                                                           | 8F10.5       |
| (...)<br>card :    | X(K), K=1, KMAX                                                                               | 8F10.5       |
| (...)<br>card :    | XNU(K, N), DF(K, N), SA(K, N), SF(K, N),<br>(SS(K, M, N), M=K, KMAX)<br>N=1, KOMSA, K=1, KMAX | 8F10.5       |
| (...)<br>card :    | P                                                                                             | E14.7        |
| (...)<br>card :    | HYUK                                                                                          | E14.7        |

## Information about some of the input parameters:

IB : number of regions

IGEO : 1 for cylindrical geometry  
0 for cartesian geometry

ICAP  $\neq$  1 + ICEY  $\neq$  1 : whole core in cartesian geometry

ICAP = 1 + ICEY  $\neq$  1 : half core (Y axis sym.) in cartesian geometry, or whole core in cylindrical geometry

ICAP  $\neq$  1 + ICEY = 1 : half core (X axis sym.) in cartesian geometry

ICAP = 1 + ICEY = 1 : quarter core in cartesian geometry, or half core in cylindrical geometry

IAD : 0 for flux and adjoint flux calculation  
           $\geq 2$  for flux calculation only

IGUC : 1 for power and source distribution calculation

IBAK : 1  $B_1^2$  is calculated by HYUK for X-Y geometry  
          0 for cylindrical geometry case

KMAX : number of groups

IMAX : number of mesh points in X or r axis

JMAX : number of mesh points in Y or Z axis

KOMSA : number of compositions

IBIS : 0 for relative flux calculation  
          1 for relative and normalised (power) flux

IS1(I) : left side boundary of region I

IS2(I) : right side boundary of region I

JS1(I) : bottom side boundary of region I

JS2(I) : top side boundary of region I

IRRX1 : left boundary of core region

IRRX2 : right boundary of core region

IRHZ1 : bottom boundary of core region

IRHZ2 : top boundary of core region

KK(K) : mesh number of region K

IMAT(K) : composition number in region K

DELRI(I): mesh intervals in region I on X or r axis

DELZJ(I): mesh intervals in region I on Y or Z axis

X(K) : fraction of fission neutrons in group K

XNU(K,N):  $\nu$  for group K and composition N

-A3-

DF(K,N) : diffusion coefficient for group K and composition N

SA(K,N) :  $\Sigma_a$  for group K and composition N

SF(K,N) :  $\Sigma_f$  for group K and composition N

SS(K,M,N):  $\Sigma_s$  from group K to group M for composition N

P : reactor power in watts for IGUC =1, or IBIS=1 cases  
 , otherwise no need for P

HYUK : height of reactor core in Cm for X-Y geometry