



**AUSTRALIAN ATOMIC ENERGY COMMISSION
RESEARCH ESTABLISHMENT
LUCAS HEIGHTS**

CONVERSION OF THE NEUTRON CODE TDC TO FORTRAN IV

by

B. MCGREGOR



February 1967

AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT

LUCAS HEIGHTS

CONVERSION OF THE NEUTRON CODE TDC TO FORTRAN IV

by

B. MCGREGOR

ABSTRACT

The conversion of the code TDC from CDC 1604 FORTRAN to FORTRAN IV is discussed with some notes on additional options provided. A series of test problems was devised and run and some points on code usage are reported. The preparation of input data is not described. This is given in the TDC User's Guide by Rodgers and Check (1965).

CONTENTS

	<u>Page No:</u>
1. INTRODUCTION	1
2. CODE CONVERSION	1
3. CODE TESTING AND TIMING	1
4. FLOCO INPUT CONVENTIONS	4
5. CODE ALTERATIONS	5
5.1 Input Data Checking	
5.2 Dump and Restart Facilities	
6. USE OF THE CODE	5
7. TDC FLUX GUESSES FROM CRAM	6
8. REFERENCES	6

1. INTRODUCTION

The TDC code solves either the regular or adjoint Boltzmann equation in finite (R,Z) cylindrical geometry by the discrete SN method. The code was first written in FLOCO (Carlson et al. 1960) and was translated to FORTRAN 63 for the CDC 1604 by Fader et al. (1964) and Rodgers and Check (1965). It was this version of the code distributed by the Argonne Code Centre that was chosen for conversion at the A.A.E.C. Research Establishment, Lucas Heights. There are no specific limits on the parameters in a problem, except that the whole problem fits into core.

2. CODE CONVERSION

The main conversion problems were associated with the input-output routines and other special facilities provided in FORTRAN 63. Such special features are multiple statements on one card, and the data handling routines ENCODE, DECODE, BUFFER IN, and BUFFER OUT. The CDC machine has 48 bits so that A8 is the standard word length and the change to the 7040 required changed input formats. One word of length 8 BCD characters was replaced by two 6 character words. The FLOCO free input reading requirements were handled by using the FREE routine (Pollard and Robinson 1966) modified to read from core. Further input data checking and additional termination facilities were added to the code.

3. CODE TESTING AND TIMING

Nine test problems were run to test the accuracy of the converted code.

- (1) Standard test problem from the FORTRAN TDC report (Fader et al. 1964).
- (2) A problem run with the FLOCO version of TDC.
- (3) A k-calculation with 6 groups and a 25 x 25 mesh.
- (4) An adjoint problem.
- (5) A size search for criticality.
- (6) A concentration search for criticality.
- (7) An inhomogeneous problem with an external source.
- (8) An inhomogeneous problem with a distributed source.
- (9) Problem 1 with different SN constants.

The standard TDC test problem converged in about 1 hour. All numbers agreed with the previous FLOCO TDC calculation to at least 5 significant figures. The restart facility was used with the converged fluxes and the code converged in about 10 minutes.

The second test problem was an attempt to duplicate results obtained from the FLOCO version of the code in April 1962 (Gemell, A.A.E.C. Private Communication). The problem was a k-calculation with 16 energy groups and a 12 x 12 mesh in S4. The FLOCO version gave a value of k of 1.041. This problem could not be repeated exactly as the convergence schemes of FORTRAN TDC and FLOCO DSN are different. Because a complete flux guess was used in the original problem while the input flux to FORTRAN TDC had the same spatial variation in each group, a difference between the performance of the two codes could be expected but the two problems should give the same converged answers. The converged value of k after 6 hours was 1.049. The fast fluxes agreed well with the original results but the thermal fluxes differed by up to 50 per cent. This difference did not alter the multiplication factor greatly as the spectrum in this problem was mainly fast. Because the second test problem failed to prove conclusively that the code was functioning correctly, another problem was required. As no further FLOCO TDC runs were available for comparison a problem was devised for which TDC and the diffusion theory code CRAM were applicable. The reactor model considered was a cylindrical core of BeO fuelled with U235 (Mod./fuel = 1500/1) and surrounded on all sides by a graphite reflector. Six energy groups were used with a 25 x 25 mesh in the S4 approximation. The CRAM problem required 2 hours to give a converged k of 1.10.

The table below shows the unusual convergence pattern with this problem. With most nuclear codes the final solution is approached in a steady manner but this is not necessarily true for TDC. It is important to realise this as long problems would normally be run in stages. It could be wrongly concluded from the table that the code had converged after 8 hours as the multiplication factor k and the fluxes varied only slightly during the previous 3 hours. Tabulated for the centre of the system are ratios of the TDC fluxes at various stages of the calculation to the final converged CRAM fluxes. Group 1 fluxes are normalised to unity at the centre for both codes. The table shows that both k and the fluxes do not converge steadily to their final answer. The code is closer to its final answers after 4 hours than after 8 hours and care should be exercised in predicting the final answer to a problem on the basis of an uncompleted problem.

TDC CONVERGENCE PATTERN FOR PROBLEM 3

<u>Time (hr)</u>	k	<u>Flux Ratio in Groups</u>		
		2	4	6
4	1.10	0.998	1.006	1.013
5	1.14	1.001	1.035	1.048
8	1.14	1.002	1.040	1.050
12	1.110	1.001	0.999	0.999

The converged CRAM and TDC agreed closely except near the core/reflector interface where the differences were of the order of 5 per cent. The results of this problem showed that the main sections of the code were working correctly.

The following problems test the various options:

Problem 4 was the adjoint solution to problem 1. The code converged in 50 minutes to a k of 1.026 which agrees with the result obtained in the regular calculation under the convergence criteria used.

Problem 5 tested 2 options; searching on mesh size (A02=4) and searching for a k not equal to unity. Using the same data as in problem 1, S05 was set equal to unity and S06 equal to 1.1 to force a search for the reactor size for which k would be 1.1. A further problem found the k of the converged system to be 1.099.

Problem 6 tested the concentration search (A02=3) option. A k calculation was run with the converged concentrations giving k = 0.9994.

Problem 7 is the type of problem encountered when analysing exponential experiments where there is an external source as well as an internal fission source. The problem with 2 energy groups simulated the geometry used in Physics Division experiments (McCulloch et al. 1965). The problem failed to converge. After 10 hours the thermal flux showed steady spatial changes while the convergence parameter λ varied rapidly between 0.9 and 1.1. When there are no fissions in the stack the code behaves less wildly but still fails to converge in a reasonable time. This difficulty with an external source occurs in DSN as well as TDC. It seems there is a fault in the basic formulation of these problems. It is recommended that this type of problem should not be attempted until some method of obtaining a reasonable answer is devised. The next section provides an alternative procedure.

Problem 8 is the same as 7 except that the source is not external but is distributed in the first mesh interval in the stack (A02=0, S01=2, S02=1 with a Q0 vector). This problem converged to a reasonable answer in 2 hours and it seems that source problems should be handled in this way.

Problem 9 tested various values of N in the SN approximation. The SN constants (vectors M4, M5, M6, M7) are entered as input by the user. Sets (with duplicates) were prepared and tested with a similar problem to problem 1 but with the required accuracy in k set to about 0.0005.

RESULTS

SN number	k
4	1.0304
6	1.0268
8	1.0258

4. FLOCO INPUT CONVENTIONS

The main input conventions used in FLOCO codes allow interpolation (I), repeat (R) and zero (Z) data. These symbols appear in the last column of an input field (6 fields of length 12 on a card) preceded by a fixed point number. With one minor proviso, all these conventions have been retained in the 7040 FORTRAN version of the code.

Examples

Column	12	24	36
A	3Z	2.0	3.0
B	1.0	2R	2.0
C	1.0	3I	0.

nZ Option Z - the next n positions in the specified input vector are set to zero - in example A the input data for the first 5 positions are:

0. 0. 0. 2.0 3.0

nR Option R - the next n positions in the specified input vector are set to the preceding number read - in example B the input data for the first 4 positions are:

1.0 1.0 1.0 2.0

I Option x nI y - The n positions (2 to (n + 1)) are set to values interpolated at equal intervals between x and y - in example C the data for the first 5 positions are:

1.0 0.75 0.5 0.25 0.

Restriction I cannot be used in the last column (72) of a card since the interpolate routine requires that both x and y appear on the one card.

5. CODE ALTERATIONS

5.1 Input Data Checking

The code previously did not check whether an input vector was missing from any particular data set. A message now gives the number of missing vectors. This system does have the disadvantage that, in some problems, vectors such as K6 and K7 must be entered when they are not used but the advantage is that missing data vectors can be easily located.

5.2 Dump and Restart Facilities

Because of the long time involved in running most problems (5 hours would be average) adequate termination and dump facilities are required. Although the code already had facilities for problem termination with a flux dump and print out on sense switch 4 and a periodic flux dump and print on sense switch 2, a time termination was added with the time in minutes being entered in columns 70-72 of the title card. However, this termination, as well as the sense switch 4 termination, required the completion of 2 outer iterations before a flux dump could be given and a satisfactory exit made. As an outer iteration can take several hours, a quick dump procedure was incorporated. Seven minutes before an interval timer overflow a termination with a flux dump and print is executed.

6. USE OF THE CODE

As use of the code and the preparation of input are well described in the TDC User's Guide (Rodgers and Check 1965) this section is designed to complement that report.

A difficulty in most neutron codes is that the code does not converge on the required eigenvalue (say k) but on a convergence parameter (λ in TDC) which measures the changes in flux from one iteration to the next. The accuracy required in λ is specified in I07(= ϵ). The maximum error in λ for a specified value of ϵ varies with the problem but is usually about 2ϵ . The maximum error in k is the product of the error in λ and $\frac{dk}{d\lambda}$, the ratio of the error in k to the error in λ which varies with the problem, values in the range 1 to 10 being usual.

For example in problem 1, $\epsilon = 0.01$, $\frac{dk}{d\lambda} = 1.05$,

therefore the error in $k = 0.02$,

while in test problem 3, $\epsilon = 0.002$, $\frac{dk}{d\lambda} = 8.1$

and the error in $k = 0.032$.

The use of the double option (A06 = -2) requires some care. The number of points IO3, IO4 and the vectors FO and WO are left unchanged but the vectors R1, Z1, and Q0 are expanded to the new size.

7. TDC FLUX GUESSES FROM CRAM

Significant computer time can be saved by providing a good initial flux guess which could come from a converged CRAM problem.

Routine 6 in CRAM LINK 5 produces punched binary flux cards (with a header card) for use in TDC. The fluxes are taken from the current CRAM dump tape. The CRAM and TDC problems must agree in the number of energy groups, number of points in the R direction, and the number of points in the Z direction. To use the routine, the input cards to CRAM would be DUMP COMPILE ROUTINE 6 FINISH.

8. REFERENCES

- Carlson, B., Lee, C., Worlton, T. (1960). - The DSN and TDC neutron transport codes LAMS 2346, the TDC code LAMS 2346 Appendix II.
- Fader, W. J., Rodgers, R. J., Mynatt, F. R., Lawrence, R. E. (1964). - The TDC Code, Pratt and Whitney Aircraft - Canel, TIM 847.
- McCulloch, D. B., Duerden, P., Brittliff, E. (1965). - Buckling and integral spectrum measurements in U235 fuelled sub-critical assemblies moderated by BeO/fertile material mixtures. AAEC/E146.
- Pollard, J. P., Robinson, G. S. (1966). - GYMEA. A nuclide depletion space independent, multigroup neutron diffusion, data preparation code, AAEC/E147.
- Rodgers, R. J., Check, P. S. (1965). - User's Guide to FORTRAN TDC, Pratt and Whitney Aircraft, Canel, TIM 890.