Neutron Kinetics
via
Numerical Laplace Transform Inversion

by
B.D. Ganapol[+] and R. Furfaro
Department of Aerospace and Mechanical Engineering
University of Arizona

ABSTRACT

A newly developed numerical Laplace transform inversion (NLTI) will be presented to determine the transient flux distribution within a subcritical fissile sample. The NLTI considered in this presentation has evolved to its present state over the past 10 years of application. The methodology adopted relies on the evaluation of the Bromwich contour through acceleration of the convergence of an infinite series. The new inversion will be applied to the prediction of the transient flux distribution within fissile material. The neutron flux is determined through a novel formulation of the solution to the transformed one group neutron diffusion equation in a heterogeneous medium. The methodology can be extended to the multigroup approximation.

I. INTRODUCTION

Laplace transforms and their inversions provide a valuable learning tool with regard to solving linear ordinary differential equations. Generally, inversion by table lookup is the preferred method of inversion in the beginning undergraduate engineering and mathematics courses. Then, as the student matures in mathematical understanding, inversion using the Bromwich contour and analytical continuation is presented. Often, however, complex analysis and table lookup do not provide the required inversions sought. Therefore, it may become necessary to apply a numerical Laplace transform inversion (NLTI), usually requiring some artful manipulation, to enable an evaluation. There are many numerical inversions from which to choose all with at least one parameter that must be pre-determined in order to generate reliable results. Numerical algorithms based on inverting an integral equation, taking limits of higher order derivatives, various expansions of the integrand on the Bromwich contour and numerical quadrature of different flavors, are some of the myriad of methods that have previously been proposed. Davies (Davies, 1979) provides an excellent review of previous inversion methods. In this presentation, a new inversion based on direct numerical integration on the Bromwich contour will be presented and applied to determine the transient neutron flux in a scattering and absorbing medium.

[+] The authors gratefully acknowledge the support of the Department of Energy (subcontract #0757400199) through the ASCI program at Los Alamos National Laboratory.

A straightforward (as straightforward as possible when dealing with Laplace transform inversion) quadrature approach is proposed and demonstrated. The numerical integration is based on the Bromwich inversion integral defined as follows for $0 < t < \infty$:

$$f(t) = \frac{1}{2pi} \int_{g-i\infty}^{g+i\infty} ds e^{st} \overline{f}(s)$$

where *g* is to the right of the rightmost singularity of the image function $\overline{f}(s)$ given by

$$\overline{f}(s) \equiv \int_0^\infty dt e^{-st} f(t) \ .$$

In this way, analytical continuation into the complex *s*-plane is avoided at the expense of treating highly oscillatory integrands. Our analysis will demonstrate the applicability of the proposed algorithm applied to two classes of image functions about which differing amounts of information concerning the distribution of singularities is known.

The existence of so many numerical Laplace transform inversion algorithms, is an indication of the highly ill-conditioned nature of the inversion process. As shown by Davies (Davies, 1979), there are particular classes of functions for which certain algorithms will provide reliable results. Most of the algorithms developed to date have a tunable parameter that must be correctly set in order to achieve the "best" results. Here the term "best" may refer to the fewest function evaluations or the minimum error for a fixed number of function evaluations or just getting a numerical result at all. In many cases, the numerical procedure comes in two parts, i.e., a parameter determination based on a desired error and maximum machine precision and then the actual application of the numerical algorithm itself. The present investigation encompasses a new line of reasoning that is based on the following observations:

1) No single NLTI algorithm is suitable for all classes of invertible functions for all values of *t*.

Such a large collection of NLTI algorithms is evidence that no single algorithm will ever be adequate for all applications. Indeed, some algorithms are reliable for small *t* and others for large *t*. Most often, machine accuracy is the limiting factor that can, of course, be temporarily overcome by performing high precision arithmetic at the expense of CPU time. This is a temporary fix at best however.

2) Results are usually considered adequate when the relative error is in the range 10^{-7} to 10^{-4}.

Most useful results in engineering required single precision and no more. It is for this reason, we will consider 7-digit accuracy, at most, in this presentation. While higher precision can be and is actually obtained, this is not true in general. In some instances, as will also be demonstrated, even 7 reliable digits cannot be found. We focus here on results useful for nuclear engineering applications and not on the extremely high accuracy that usually only appeals to the computer scientist.

3) Computational effort is relatively unimportant in lieu of obtaining a reliable numerical result.

Because of the often highly ill-conditioned nature of the Laplace transform inversion, where a small change in the integrand, produces a large fluctuation in the inversion, a poor choice of contour or any other required parameter can produce erroneous results. In such instances, it is therefore of paramount importance to ensure that the proper settings are chosen in order to obtain even a minimally acceptable numerical estimate, thus the reason for a proposed parameter determination phase which can be the most costly part of the NTLI algorithm.

4) Computational effort is relatively inexpensive with today's computational platforms.

With faster and faster CPUs and massively parallel computational architectures, computing times have plummeted over the past 10 years, even for PCs. Thus, more computations can be accommodated in a second than this time last year making feasible more computationally intensive and, therefore, more reliable NLTIs. Thus, in today's computational environment, the choice of LTI options has broadened. Indeed, one can argue that with massively parallel processing even the least efficient NLTI now becomes viable.

5) A reliable NLTI algorithm should be able to generate results with minimum information known about the image function.

To be considered a successful NLTI, an algorithm should perform reliably. That is, the user must have confidence that the result obtained is the true result to the desired number of digits requested. Also, failure to achieve a reliable result should always be indicated in any NTLI procedure. When the singularities of a given image function are known apriori, this information can often be used to specify the proper choice of the required parameter(s). But, what if nothing is known about the singularities--can the algorithm still produce reliable results? In our estimation, this is the most stringent test of an NLTI. This situation occurs when an NLTI is included within an overarching numerical procedure. In the case to be considered here, estimating the flux within a scattering and absorbing medium through diffusion theory, the image function cannot in general be written down explicitly and therefore very little is known about its singularities.

The NLTI to be presented in the following section has evolved to its present state over the past 10 years of application. The methodology adopted is one that relies on acceleration of the convergence of an infinite series towards its limit. This is a necessary feature of the algorithm, called **LTI:R** for Laplace transform inversion with Romberg integration, in order to resolve integrals with highly oscillatory integrands. As the name suggests, the usual trapezoidal integration is replaced by a Romberg rule with a modest increase in computational effort. Finally, the choice of the Bromwich contour, crucial to

relaible operation, is determined by a minimization of the number of function evaluations. The **LTI:R** algorithm will also be applied to the one group time-dependent neutron diffusion equation in a highly heterogeneous medium.

2. LTI:R ALGORITHM

a. The Inversion Formula

The algorithm to be developed takes advantage of the following cosine integral form of the Laplace transform:

$$f(t) = \frac{2e^{gt}}{pt}\int_0^{\infty} dw\ \mathrm{Re}\,\overline{f}\left(g + i\frac{w}{t}\right)\cos(w)\ . \qquad (1)$$

As suggested in many references and articles, the standard approach of decomposing the cosine integral into integrals over the cosine periods will be followed to give

$$f(t) = \frac{2e^{gt}}{pt}\sum_{k=0}^{\infty}(-1)^k\int_0^{p} dw\ \mathrm{Re}\,\overline{f}\left(g + i\frac{w+kp}{t}\right)\cos(w)\ . \qquad (2)$$

Each integration is performed using a Romberg rule featuring the re-use of function values. If

$$g(w) \equiv \mathrm{Re}\,\overline{f}\left(g + i\frac{w+kp}{t}\right),$$

the Romberg rule is

$$T_l^{m+1} = T_l^m + \frac{T_l^m - T_{l-1}^m}{4^{m+1}-1}\ , \qquad (3a)$$

initiated by the trapezoidal rule

$$T_l^0 = \frac{1}{2}T_{l-1}^0 + \frac{p}{2^l}\sum_{j=1,odd}^{2^l-1} g\left(\frac{jp}{2^l}\right), \qquad T_0^0 = \frac{p}{2}\left[g(0)+g(p)\right]\ . \qquad (3b)$$

The central feature, which is the essence of the algorithm, is the acceleration of the convergence of the series in eq(2) using the nonlinear Wynn-epsilon acceleration (Baker, 1996). This algorithm seems to outperform the ever-popular (linear) Euler-Knopp acceleration. The proposed NTLI algorithm is only made possible through the efficiency of convergence provided by the epsilon convergence accelerator.

b. Error Control

As with any numerical algorithm, its truth lies in the details of the implementation. For the **LTI:R** algorithm, four potential sources of error have been identified, including error in the numerical quadrature, error in the convergence accelerator, loss of significance and series truncation error.

b.1 Numerical Quadrature Error

The diagonal elements of the tableau formed by the approximations T_l^m are monitored for convergence using the relative error between two Romberg approximations

$$e_{R1} \equiv \left|\frac{T_l^{m+1} - T_l^m}{T_l^{m+1}}\right|\ . \qquad (4)$$

In addition, the relative error between trapezoidal quadratures is also monitored

$$e_{R2} \equiv \left| \frac{T_{l+1}^{0} - T_{l}^{0}}{T_{l+1}^{m+1}} \right| \tag{5}$$

for convergence. The integration is considered converged when $\min(e_{R1}, e_{R2}) \le e$ where e is the desired relative error for the inversion under consideration. In parallel, the trapezoidal integrations and the diagonal elements are considered sequences in l and m respectively. The epsilon algorithm is applied to each sequence with e_{R2} and e_{R1} replaced by the errors associated with the epsilon algorithm if smaller than the Romberg and trapezoidal integration errors respectively.

b.2 Use of Epsilon algorithm

The epsilon convergence algorithm can be written as

$$\begin{aligned} e_{-1}^{(n)} &= 0,\ e_{0}^{(n)} = S_n \\ e_{k+1}^{(n)} &= e_{k}^{(n+1)} + \left[e_{k}^{(n+1)} - e_{k}^{(n)} \right]^{-1} \end{aligned} \tag{6}$$

where S_n is the sequence to be accelerated toward convergence. A tableau is constructed with sequential diagonal elements interrogated for convergence. It could happen that the denominator of the second term in eq(6) vanishes leading to failure. In this case, either $e_{k}^{(n+1)}$ or $e_{k}^{(n)}$ is returned since convergence has been achieved. This situation is flagged and reported. In normal operation, the final diagonal element of the tableau is returned as the limit of the sequence.

b.3 Loss of Significance

When summing a series of numbers, there is always the potential for contamination by roundoff. In an attempt to minimize this contamination, the series in eq(2) is summed using compensated summation. When the sum is predicted to have converged after *n* terms and if the condition number

$$r \equiv \sum_{k=1}^{n} |a_k| \Big/ \left| \sum_{k=1}^{n} a_k \right|$$

is larger than $1/e$, then potential roundoff error has occurred. In this case, the relative integration error is reduced by a factor of 10 and the summation repeated until no roundoff error is predicted or the integration tolerance is below machine accuracy. If the sum cannot be computed without roundoff, an error is recorded.

b.4 Series Truncation Error

The final source of error concerns the truncation of the series in eq(2). Here the error is estimated by

$$e_s \equiv \left| \frac{S_n - S_{n-1}}{S_n} \right| . \tag{7}$$

where S_n is either the epsilon algorithm result or original partial sum after *n* terms. The epsilon algorithm tableau is created using all partial sums up to and including S_n. The

series is considered converged when $e_s \le e$ for two consecutive partial sums or accelerated partial sums.

c. Numerical Implementation

c.1 Positioning of the Bromwich Contour

Not unlike essentially all existing numerical Laplace transform algorithms, there is at least one parameter that needs to be predetermined in order for the **LTI:R** algorithm to operate properly. In particular, the position g of the Bromwich contour on the real line needs to be specified. While, in theory, g need only be larger than the largest singularity, in practice, the numerical evaluation can be extremely sensitive to its value. For this reason, special attention must be given to the choice of g to ensure reliable results.

Based on recent work by Weideman (Weideman, 1999), an optimization approach will be followed. Unlike Weideman, however, the optimization will not be based on the minimization of an error estimate. An explicit error estimate is difficult to obtain for the **LTI:R** algorithm because of the inclusion of the epsilon algorithm for convergence of the series in eq(2). Instead, an optimization is based on the minimization of the number of function evaluations. Using the **BRENT** minimization routine (Press, 1992) and assuming an initial g and a computational error of a factor of 10 times the desired error e, a local minimum number of function evaluations is obtained for a given t. Special attention must be given to avoid a prediction of g that is less than the real parts of the rightmost singularity. While what is found may not be the global minimum, it is hoped that the g found will allow the inversion to proceed. However, at this point there is no guarantee that the inversion obtained is indeed accurate. To provide a sense of the inversion accuracy, the error is decreased by a factor of 10 (to e) and the inversion repeated until the relative error between two successive approximations is less than the desired error e. The implementation of this error check is accomplished by re-using all previous function evaluations for each new determination with a reduced desired error. In this way, the overall number of function evaluations and therefore computational time is minimized.

c.2 A Useful Transformation

While every attempt has been made to avoid developing this algorithm by trial and error; nevertheless, a very useful transformation has been found to reduce the computational effort. The equivalent relation

$$f(t) = \frac{be^{-at}}{2pi} \int_{g-i\infty}^{g+i\infty} dse^{sbt}\, \overline{f}(bs-a) \qquad (8)$$

obtained from dilation and translation of the transformed variable, has been found to provide an exceptionally reliable inversion in many cases. For all calculations to date using **LTI:R**, the value of b giving the best results has been 0.1. It seems that if nothing is known about the singularities, a = 0 is the best choice. If the position of the rightmost singularity is known, then a is chosen as its real part (at least for large t).

3. DEMONSTRATION: Type 1

In this section, the reliability and usefulness of **LTI:R** algorithm will be demonstrated on selected examples of type 1. The examples are chosen from the

inversion literature and which can generally be of three types of image functions. The first type describes relatively simple inversions that are most often used for demonstration purposes leading to continuous functions of t. The second type concerns transforms that lead to discontinuous functions or their derivatives. The last type depicts transforms that are not explicitly specified since they are a part of a numerical procedure -- here, the solution to the one group neutron diffusion equation in a heterogeneous medium. In this presentation, only types 1 and 3 will be considered. All calculations in this section were performed on a DEC Alpha Model 3000 workstation for a desired relative error of 10^{-7}. The initial contour guess for all optimizations is $g = 10$. $a = 0$ and $b = 0.1$ are used almost exclusively except for large t. When the number of function evaluations is reported, that number is the total number of evaluations required for the error check only and does not include the number of evaluations required for the optimization procedure.

In order to demonstrate the **LTI:R** algorithm, 21 transforms, from Table I (Abramowitz, 1964), are to be numerically inverted. These inversions are simple enough to be reliably inverted using most numerical inverters. The image functions are taken "as is" without any modification to achieve improved accuracy. As shown in Table 1.1 for 20 uniform points in the time interval [0.1,10.0] and for optimization at each point, all actual (L_∞) errors, shown in column 2, are well within the desired relative error of 10^{-7}. The next column shows the average number of function evaluations of the last two determinations ($e = 10^{-6}, 10^{-7}$) of the inversion. While the number of function evaluations is generally somewhat higher than for other inversions, this increase is a small price to be paid for the reliability provided by the optimization procedure. The fourth column gives the time (in seconds) required by the optimization procedure for each transform which can be seen to be the majority of the computational effort. The fifth column gives the estimated time per edit point not counting the optimization time. The possible "local optimum" real part of the Bromwich contour is given in the next column. It is interesting to note that the chosen Bromwich contour is invariably relatively far from singularities. The average number of terms to converge the series of eq(2) is given in the second to last column. In the last column, the number of errors (as indicated above) encountered in the numerical evaluation of the final inversion is presented. A negative number indicates that the final error check failed. The total time required for construction of Table 1.1 was 72s. Table 1.2 depicts the same cases when optimization is applied to the first t edit point only. It is observed that the actual errors for 5 of the 21 cases are now not within the desired error and, in addition, the number of function evaluations and calculation time per point has increased. However, because only one optimization was performed per case, the overall time of computation for this table was reduced by a factor of about 4 to 19s. Possibly, an optimum strategy would be to perform an optimization after the second edit. Tables 1.3 and 1.4 extend the results to small t with 20 points in the interval $[10^{-3},10^{-1}]$. Except for case 15, all inversions give the expected accuracy. Note that the optimized position of the Bromwich contour has significantly increased over the previous case. This is just a conformation of the initial value theorem that states that small time behavior is governed by large s. The difficulty with case 15 can immediately be recognized from $F_{15}(t)$. Since the inversion algorithm cannot be expected to give better than machine accuracy, the inversion, which produces an extremely small number for small values of t,

will never be achieved. It is comforting to note that six errors in the computation of case 15 are encountered indicating that the result fo this case should not be trusted. If the optimization is required for every 4th point then Table 1.4 results. For some of the inversions there is little difference. For several, however, the inversion is not acceptable without the optimization procedure. The time to produce the last two tables was 101s and 46s respectively.

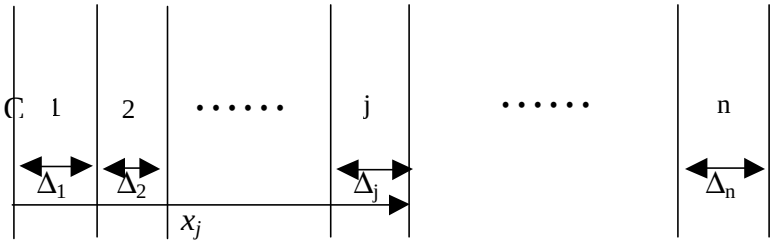
4. DEMONSTRATION: Type 3

While it is necessary for any numerical Laplace transform inversion to treat explicitly given image functions, a more useful demonstration is reliable when the image function is not specified in a closed form. Such a demonstration can be conceived of when the inversion is part of an application, in this case for the neutron flux. The demonstration to follow is intended to show that the **LTI:R** algorithm can be used confidently to analyze time-dependent neutron behavior in a subcritical heterogeneous medium of many materials.

We begin with the space-time reactor kinetics equation (without delayed neutrons) in a 1-D heterogeneous medium of *n* slabs.

$$\frac{1}{v}\frac{\partial f_j(x,t)}{\partial t} = D_j \frac{\partial^2 f_j(x,t)}{\partial x^2} + \left(n\Sigma_{fj} - \Sigma_{aj}\right) f_j(x,t) + S_{0j} \qquad (9a)$$

Any configuration of fissioning and non-fissioning materials can be represented. All symbols have their usual meanings. A spatially and temporally uniform source can be



located in any region and the surface boundary conditions are

$$\begin{aligned} f_n(x_0,t) &= 0 \\ f_n(x_n,t) &= 0\ . \end{aligned} \qquad (9b)$$

The interfacial and initial conditions are

$$\begin{aligned} f_{j-1}(x_{j-1},t) &= f_j(x_{j-1},t) \\ -D_{j-1}\frac{\partial f_{j-1}(x,t)}{\partial x}\bigg|_{x=x_{j-1}} &= -D_j \frac{\partial f_j(x,t)}{\partial x}\bigg|_{x=x_{j-1}} \\ f(x,0) &= f_0\ , \end{aligned} \qquad (9c)$$

After a Laplace transform and from the definition

$$\bar{f}_j(s) \equiv \bar{f}_j(x_j,s)\ ,$$

the following solution in transform space can be specified:

$$\bar{f}_j(x,s) = \frac{1}{S_j(s)} \left\{ \begin{array}{l} \sinh\left(b_j(s)[x - x_{j-1}]\right)\left[\bar{f}_j(s) - \bar{f}_{pj}(s)\right] + \\ \quad + \sinh\left(b_j(s)[x_j - x]\right)\left[\bar{f}_{j-1}(s) - \bar{f}_{pj}(s)\right] \end{array} \right\} + \bar{f}_{pj}(s) \qquad (10)$$

where

$$b_j(s) \equiv \frac{s}{vD_j} - \frac{n\Sigma_f - \Sigma_{aj}}{D_j}$$

$$L_j^2 \equiv \frac{D_j}{\Sigma_{aj}}$$

$$S_j(s) \equiv \sinh\left(b_j(s)\Delta_j\right)$$

and $\bar{f}_{pj}$ is the particular solution

$$\bar{f}_{pj}(s) \equiv S_{0j} / \left(sD_j b_j(s)\right) + f_0 / \left(vD_j b_j(s)\right) .$$

Note that the coefficients $\bar{f}_j$ are yet to be determined.

When eq(10) is introduced into the current continuity conditions at the j-1 interface, a 3-term recurrence relation for $\bar{f}_j$ results

$$\begin{aligned} &\bar{f}_j + b_j \bar{f}_{j-1} + g_j \bar{f}_{j-2} = \\ &\qquad = \left[1 - C_j\right] f_{pj}(s) + g_j \left[1 - C_{j-1}\right] f_{pj-1}(s) \end{aligned} \qquad (11)$$

where

$$b_j \equiv -\left(C_{j+1} + g_j C_j\right)$$

$$g_j \equiv \frac{b_{j-1} D_{j-1} S_j}{b_j D_j S_{j-1}}$$

$$C_j \equiv \cosh\left(b_j(s)\Delta_j\right) .$$

The recurrence relation can be solved most conveniently using a tri-diagonal solver (Press, 1992). For the homogeneous case, a two-point version of eq(11) can also be used.

At this point in the solution strategy, we have the Laplace transform (or image function) completely defined [eqs(10) and (11)] and are ready to apply the numerical inversion. However, we know nothing about the configuration of the singularities of the image function. The only fact that is known is that for the subcritical systems considered here, no poles exist in the right half plane.

The inversion is applied to the first Durkee 8-region problem (Durkee, 1990) whose nuclear parameters are given in Table 2. Figure 1 shows the flux evolution toward steady state for the nominal case with a source in each region. Steady state is reached near t = 10^{-4} and the results compare well (graphically) to Durkee's who uses an entirely different analytical formulation. Figure 2 shows the case for a source of 20×10^{12} in region 4 only. Again the graphical agreement is excellent. Figure 3a shows the flux for Durkee's second case where sources exist in all but the end slabs and region 5 has relatively large fission cross section and small diffusion coefficient. While the boundary conditions are not exactly the same as Durkee's, the observed trend is the same. A detailed "zoom in"

look at region 5 is given in fig. 3b. This figure is included to feature a distinct advantage of an analytical evaluation over a purely numerical one. The "zoom in" was calculated just for regions 4, 5 and 6. Since eq(10) gives an analytic representation of the flux image function in any region, the inversions in the other regions need be performed. Finally, figure 4 shows the flux in for a 32-slab problem created by considering 4 of Durkee's case 2 8-slab configurations. This result is included to indicate that the new NLTI can indeed handle a very highly heterogeneous media. All calculations were performed for an error of 10^{-4} and took under 1 hr to perform on a SUN Ultra5™.

5. FUTURE EFFORT

Several future directions of this work naturally present themselves. First, delayed neutrons could easily be included in the analytical solution given by eq(10). Difficulties may arise in the inversion however, because of the stiffness introduced. A second extension is to the multigroup diffusion equation. This would first be done for down scatter only and then iteratively for the full scattering matrix. Finally, the critical and supercritical cases could be considered. Here, care must be taken in specifying the contour and the dilation and translation parameters in eq(8). Criticality parameters can be most remarkably be specified by the recurrence solution to eq(11) virtually for any number of slabs. Multigroup criticality will most likely be the first extension treated in a future article. Finally, the transient flux in cylindrical geometry will be found using the methodology presented in this work.

REFERENCES

Baker, G.A. and Graves-Morris, P. R, *Padé Approximants*, Cambridge University Press, 1996.

Davies, B. and Martin, B., *J. Comp. Phys.*, **33**(1979),1-32.

Durkee, J.W., Jr, *Prog. In Nucl. Ener.*, **21**,191-222 1990.

Press, W.H.., Vettering, W.T., Teulolsky, S.A., Flannery, B.P., *Numerical Recipes*, Cambridge University Press, 1992.

Abramowitz, M., and Stegun, I.A., *Handbook of Mathematical Functions*, Dover, 1964.

Weideman, J.A.C.,*SIAM J. Sci. Comput.*,21, 1999,111-128.

Table 1. Trial Laplace Transforms and Their Inverses	
Laplace Transform, $F_i(s)$	Inverse Function, $f_i(t)$
$F_1(s) = (s^2+1)^{-1/2}$	$f_1(t) = J_0(t)$
$F_2(s) = s^{-1/2} e^{-1/s}$	$f_2(t) = (pt)^{-1/2} \cos(2t^{1/2})$
$F_3(s) = (s+1/2)^{-1}$	$f_3(t) = e^{-t/2}$
$F_4(s) = ((s+0.2)+1)^{-1}$	$f_4(t) = e^{-0.2t} \sin(t)$
$F_5(s) = s^{-1}$	$f_5(t) = 1$
$F_6(s) = s^{-2}$	$f_6(t) = t$
$F_7(s) = (s^{72}+1)^{-2}$	$f_7(t) = te^{-t}$
$F_8(s) = (s^2+1)^{-1}$	$f_8(t) = \sin(t)$
$F_9(s) = s^{-1/2}$	$f_9(t) = (pt)^{-1/2}$
$F_{10}(s) = s^{-1} e^{-5s}$	$f_{10}(t) = H(t-5)$
$F_{11}(s) = s^{-1} \ln(s)$	$f_{11}(t) = -g - \ln(t)$
$F_{12}(s) = (s(1+e^{-s}))^{-1}$	$f_{12}(t) =$ the unit square wave
$F_{13}(s) = (s^2-1)(s^2+1)^{-2}$	$f_{13}(t) = t\cos(t)$
$F_{14}(s) = (s-1/2)^{1/2} - (s+1/4)^{1/2}$	$f_{14}(t) = (e^{-t/4} - e^{-t/2})/(4pt^3)^{1/2}$
$F_{15}(s) = \exp(-4s^{1/2})$	$f_{15}(t) = 2e^{-4/t}/(pt^3)^{1/2}$
$F_{16}(s) = \arctan(s^{-1})$	$f_{16}(t) = t^{-1}\sin(t)$
$F_{17}(s) = (s+1)/((s+1)^2+1)^{3/2}$	$f_{17}(t) = te^{-t} J_0(t)$
$F_{18}(s) = (s+1/2)^{-\frac{3}{2}} (p^{1/2}) e^{-1/(s+1/2)}$	$f_{18}(t) = e^{-t/2} \sin(2t^{1/2})$
$F_{19}(s) = \arctan(1/(s+1))$	$f_{19}(t) = e^{-t}(t^{-1}\sin(t))$
$F_{20}(s) = 1/(s+1/2)^2$	$f_{20}(t) = te^{-t/2}$
$F_{21}(s) = (s+2a)^{1/2}/s^{1/2} - 1$	$f_{21}(t) = ae^{-at}(I_1(at) + I_0(at))$
$F_{22}(s) = \ln(s)/(s-a)$	$f_{22}(t) = -e^{at}(\ln(a) + E_1(at))$
$F_{23}(s) = ((s^2+a^2)^{1/2} - s)^k ; k > 0$	$f_{23}(t) = (ka^k/t) J_k(at)$
$F_{24}(s) = (k/(s^2+k^2))\coth(ps/2k)$	$f_{24}(t) = \lvert\sin(kt)\rvert$

Table 1.1

lt#	error	#fcn eval	opt time	time/pt	gam	#terms	diag
1	5.7975E-08	5.3733E+02	2.9E+00	1.2E-02	2.65E+01	1.46E+01	0
2	5.9893E-08	5.6638E+02	3.9E+00	1.8E-02	2.25E+01	1.42E+01	0
3	4.6579E-09	5.1967E+02	1.6E+00	7.6E-03	2.15E+01	1.40E+01	0
4	3.5673E-09	5.5205E+02	1.8E+00	9.1E-03	1.94E+01	1.41E+01	0
5	4.8181E-10	4.3900E+02	1.5E+00	6.1E-03	2.02E+01	1.20E+01	0
6	5.1611E-10	4.3800E+02	1.6E+00	6.1E-03	1.76E+01	1.10E+01	0
7	3.3259E-08	5.6162E+02	1.9E+00	8.6E-03	2.27E+01	1.47E+01	0
8	7.1376E-08	5.8800E+02	1.8E+00	9.1E-03	2.43E+01	1.40E+01	0
9	3.9780E-09	4.4200E+02	2.7E+00	1.0E-02	2.71E+01	1.30E+01	0
11	6.3967E-09	4.7276E+02	3.0E+00	1.4E-02	2.34E+01	1.27E+01	0
13	1.5961E-08	5.5143E+02	1.9E+00	9.5E-03	2.41E+01	1.36E+01	0
14	5.5692E-09	5.0748E+02	3.2E+00	1.4E-02	2.48E+01	1.41E+01	0
15	3.4654E-08	5.9976E+02	5.0E+00	1.7E-02	2.55E+01	1.38E+01	0
16	2.1933E-08	5.2448E+02	3.6E+00	1.4E-02	2.69E+01	1.46E+01	0
17	5.0672E-08	8.0057E+02	3.3E+00	1.8E-02	2.67E+01	1.74E+01	0
18	8.7397E-08	5.6933E+02	4.7E+00	1.9E-02	3.24E+01	1.65E+01	0
19	3.4519E-08	6.1514E+02	3.9E+00	1.8E-02	2.56E+01	1.73E+01	0
20	4.6151E-08	4.9100E+02	1.7E+00	7.3E-03	2.39E+01	1.31E+01	0
21	1.0372E-09	4.6476E+02	2.2E+00	1.0E-02	1.93E+01	1.21E+01	0
22	4.1381E-09	6.9010E+02	3.1E+00	1.7E-02	1.65E+01	1.36E+01	0
23	5.4477E-08	6.2419E+02	2.5E+00	1.4E-02	2.48E+01	1.50E+01	0

Table 1.2

lt#	error	#fcn eval	opt time	time/pt	gam	#terms	diag
1	2.6083E-08	6.1986E+02	1.9E-01	1.3E-02	1.10E+01	1.38E+01	0
2	3.9636E-09	6.4438E+02	1.9E-01	2.0E-02	1.19E+01	1.44E+01	0
3	6.5878E-10	4.8267E+02	7.0E-02	7.1E-03	1.09E+01	1.33E+01	0
4	9.8000E-07	5.8071E+02	7.8E-02	9.1E-03	1.08E+01	1.43E+01	0
5	4.8180E-10	4.9600E+02	7.0E-02	6.5E-03	1.10E+01	1.20E+01	0
6	5.1611E-10	4.9590E+02	6.6E-02	6.7E-03	1.07E+01	1.19E+01	0
7	4.7328E-09	5.4095E+02	7.8E-02	7.6E-03	1.07E+01	1.44E+01	0
8	1.0799E-08	7.1343E+02	8.2E-02	1.0E-02	1.07E+01	1.42E+01	0
9	9.0929E-09	5.2795E+02	1.3E-01	1.2E-02	1.16E+01	1.30E+01	0
11	1.2143E-09	5.5200E+02	1.4E-01	1.5E-02	1.10E+01	1.27E+01	0
13	1.5736E-09	6.5533E+02	7.8E-02	9.7E-03	1.07E+01	1.36E+01	0
14	5.0529E-09	4.8657E+02	1.4E-01	1.4E-02	1.15E+01	1.35E+01	0
15	3.5665E-06	6.6686E+02	1.5E+00	1.8E-02	1.84E+01	1.36E+01	0
16	3.1741E-06	5.5233E+02	1.4E-01	1.7E-02	1.10E+01	1.39E+01	0
17	1.4671E-07	7.2233E+02	1.1E-01	1.7E-02	1.10E+01	1.69E+01	0
18	9.2239E-08	6.1767E+02	1.5E-01	2.2E-02	1.15E+01	1.57E+01	0
19	4.3500E-09	6.0652E+02	1.4E-01	1.9E-02	1.10E+01	1.70E+01	0
20	7.2043E-09	5.5476E+02	5.9E-02	7.6E-03	1.06E+01	1.34E+01	0
21	8.1737E-09	5.2986E+02	1.1E-01	1.1E-02	1.09E+01	1.24E+01	0
22	9.9944E-09	6.8719E+02	1.3E-01	1.9E-02	1.16E+01	1.34E+01	0
23	6.3002E-07	5.9414E+02	9.8E-02	1.3E-02	1.10E+01	1.42E+01	0

Table 1.3

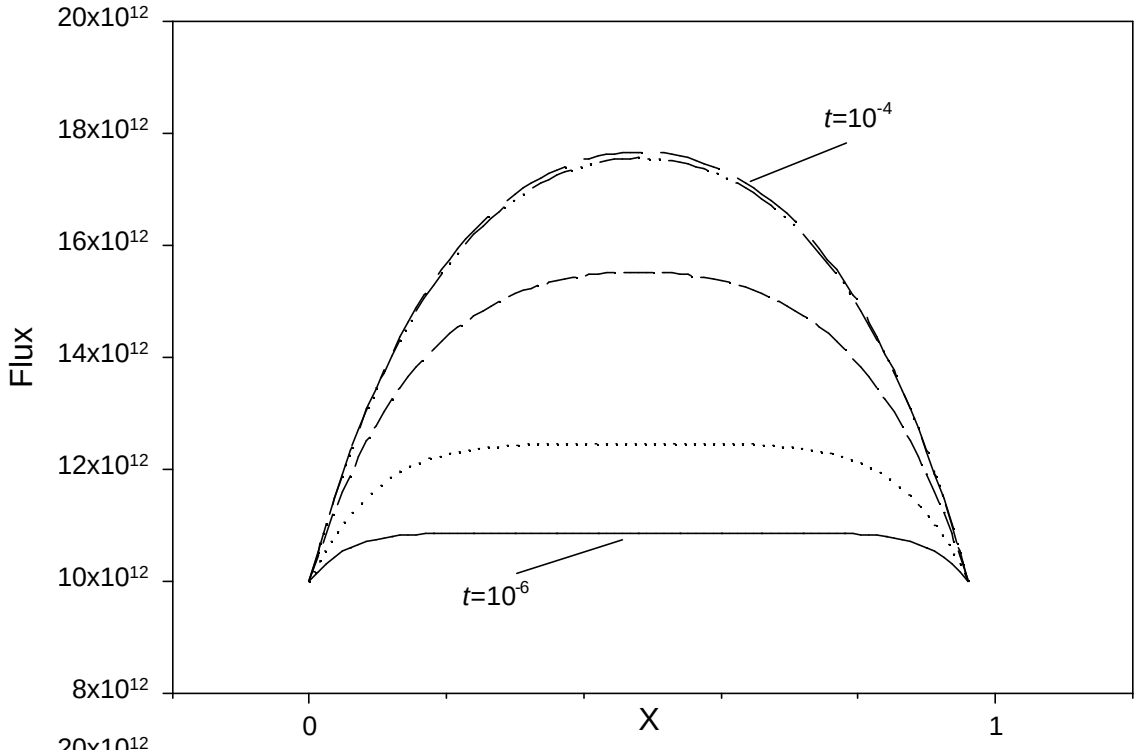
lt#	error	#fcn eval	opt time	time/pt	gam	#terms	diag
1	2.5661E-08	6.8824E+02	3.8E+00	1.5E-02	9.36E+01	9.71E+00	0
2	1.5976E-08	6.5676E+02	4.9E+00	2.0E-02	6.88E+01	1.23E+01	0
3	2.5365E-08	5.7481E+02	2.4E+00	8.0E-03	1.17E+02	9.71E+00	0
4	3.7986E-08	7.8043E+02	2.5E+00	1.2E-02	7.67E+01	1.13E+01	0
5	2.5665E-08	5.3100E+02	2.7E+00	7.6E-03	2.10E+02	1.00E+01	0
6	3.1127E-10	6.9100E+02	2.6E+00	1.0E-02	5.30E+01	1.10E+01	0
7	8.4334E-08	7.4519E+02	2.3E+00	1.1E-02	7.56E+01	1.14E+01	0
8	3.9720E-08	8.4786E+02	2.7E+00	1.2E-02	6.57E+01	1.14E+01	0
9	2.1618E-08	5.1100E+02	3.4E+00	1.1E-02	2.52E+02	1.20E+01	0
11	8.5635E-09	6.3124E+02	4.4E+00	1.6E-02	1.05E+02	1.20E+01	0
13	6.5825E-08	9.2252E+02	2.7E+00	1.3E-02	5.90E+01	1.11E+01	0
14	2.1582E-08	5.7143E+02	4.1E+00	1.6E-02	1.02E+02	1.21E+01	0
15	1.043E+272	8.1215E+03	1.1E+01	2.5E-01	1.90E+02	1.22E+01	-6
16	3.1016E-08	6.7295E+02	4.8E+00	1.8E-02	6.46E+01	9.33E+00	0
17	6.4896E-09	8.3152E+02	3.7E+00	1.8E-02	5.99E+01	1.16E+01	0
18	7.5480E-09	6.3471E+02	5.0E+00	2.0E-02	4.67E+01	1.11E+01	0
19	6.1910E-08	5.6614E+02	4.8E+00	1.5E-02	1.20E+02	1.00E+01	0
20	2.4399E-08	7.8719E+02	2.3E+00	1.2E-02	7.32E+01	1.16E+01	0
21	2.6285E-08	6.4619E+02	3.5E+00	1.3E-02	5.58E+01	9.81E+00	0
22	3.2207E-08	5.5000E+02	4.3E+00	1.5E-02	1.28E+02	1.19E+01	0
23	5.1039E-08	6.7405E+02	3.6E+00	1.3E-02	5.91E+01	9.38E+00	0

Table 1.4

lt#	error	#fcn eval	opt time	time/pt	gam	#terms	diag
1	1.3143E-07	8.1024E+02	1.2E+00	1.6E-02	6.91E+01	9.48E+00	0
2	1.3791E-07	7.3495E+02	1.4E+00	2.3E-02	4.24E+01	1.22E+01	0
3	3.4984E-08	6.4557E+02	7.5E-01	9.5E-03	7.16E+01	9.67E+00	0
4	1.1680E-06	9.2881E+02	7.0E-01	1.3E-02	5.21E+01	1.14E+01	0
5	2.5665E-08	6.4200E+02	8.0E-01	1.5E-02	1.24E+02	9.57E+00	0
6	3.1126E-10	1.2066E+03	1.2E+00	2.6E-02	3.46E+01	1.23E+01	0
7	2.1138E-08	7.4724E+02	6.6E-01	1.1E-02	6.36E+01	1.12E+01	0
8	2.6901E-03	1.1503E+03	7.6E-01	1.6E-02	6.16E+01	1.14E+01	1
9	2.1618E-08	6.0271E+02	1.0E+00	1.3E-02	1.48E+02	1.20E+01	0
11	5.4647E-08	7.3610E+02	1.2E+00	1.9E-02	6.47E+01	1.20E+01	0
13	5.0474E+00	1.4028E+03	7.2E-01	2.0E-02	3.40E+01	1.17E+01	1
14	2.1582E-08	5.9238E+02	1.3E+00	1.7E-02	9.86E+01	1.21E+01	0
15	1.043E+272	7.4289E+03	4.4E+00	2.2E-01	1.17E+02	1.18E+01	-6
16	2.5663E-08	7.8105E+02	1.3E+00	2.1E-02	3.92E+01	9.10E+00	0
17	4.3047E-09	8.8376E+02	1.0E+00	2.0E-02	3.95E+01	1.16E+01	0
18	7.5480E-09	6.9305E+02	1.6E+00	2.1E-02	3.01E+01	1.11E+01	0
19	1.4341E-07	6.1414E+02	1.4E+00	1.7E-02	7.24E+01	9.90E+00	0
20	1.3667E-07	9.5652E+02	6.6E-01	1.3E-02	6.24E+01	1.20E+01	0
21	2.6285E-08	6.7495E+02	1.0E+00	1.5E-02	4.76E+01	9.71E+00	0
22	2.3715E-08	6.2200E+02	1.3E+00	1.6E-02	8.93E+01	1.20E+01	0
23	5.1039E-08	8.0619E+02	1.0E+00	1.7E-02	2.26E+01	9.10E+00	0

Table 2 Nominal Neutronic Parameters for Durkee Case 1	
v.(m/s)	$3x10^5$
D(m)	0.00135
S_f(m^{-1})	0.0240
S_a(m^{-1})	0.03682
ν	2.41
f_0 (n/m^2-s)	$10x10^{12}$
S_0 (n/m^3-s)	$30x10^{12}$

Fig. 1 Approach of nominal case to steady state



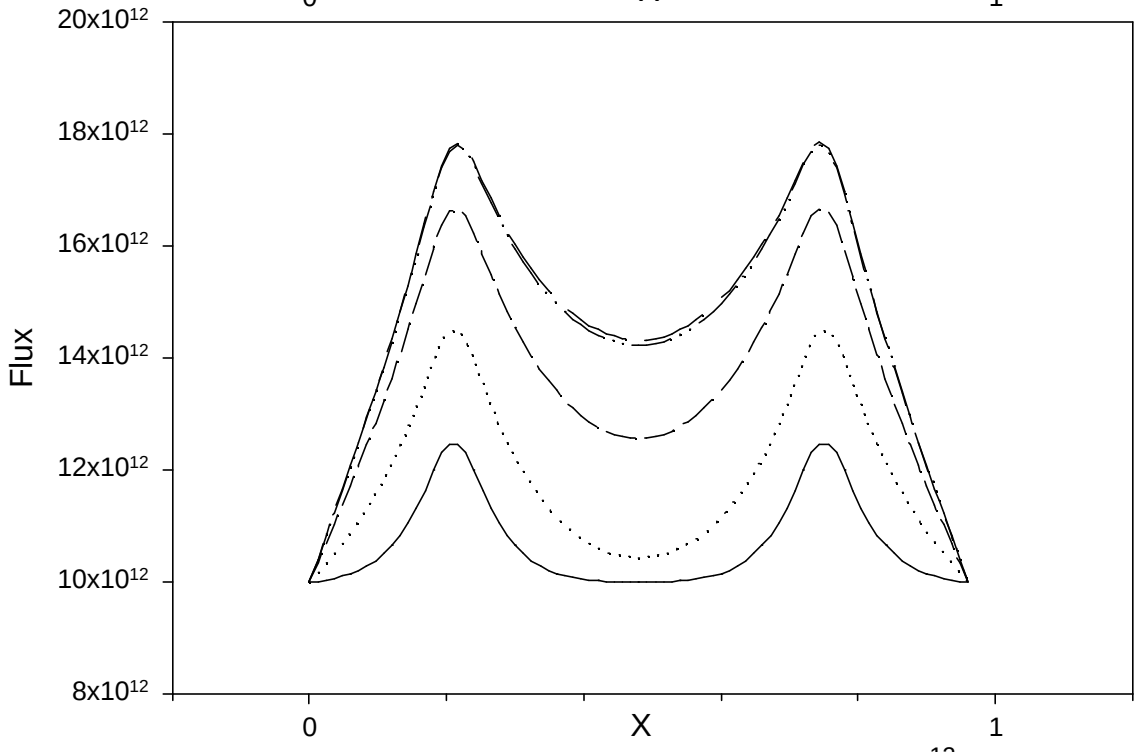
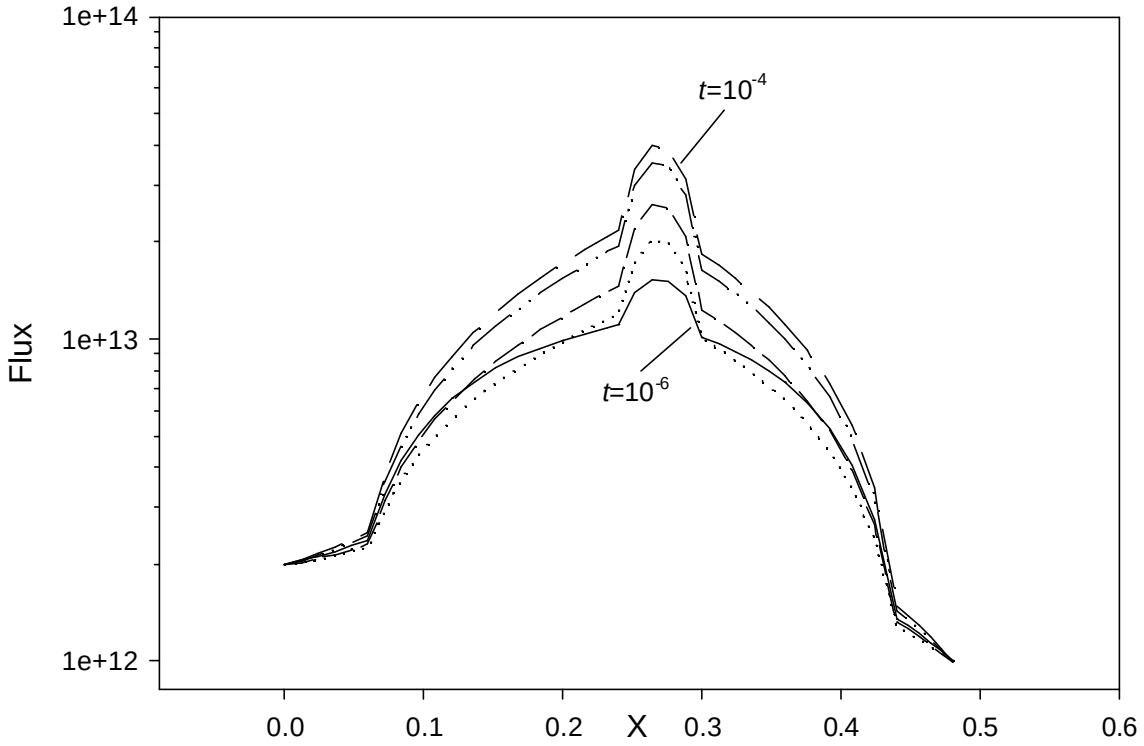


Fig. 2 Source only in regions 4 and 8 of 20×10^{12}

Fig. 3a Durkee case 2 with incresed fission in region 5



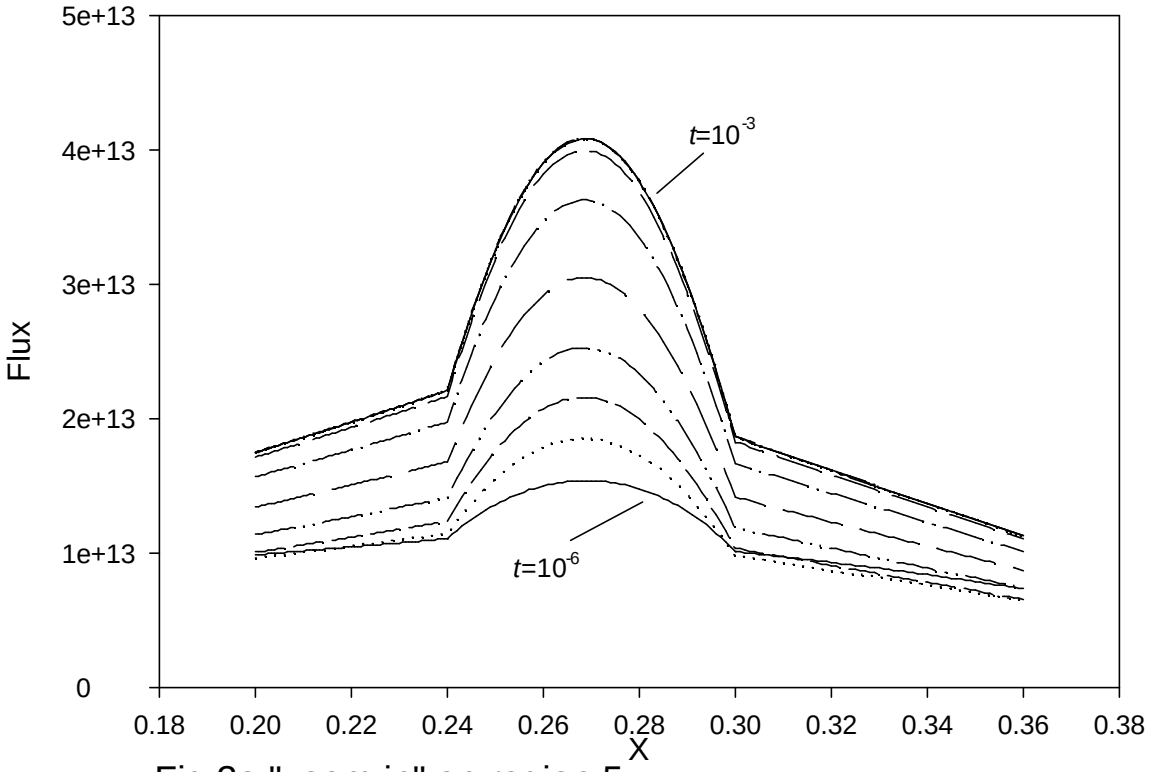


Fig 3a "zoom in" on region 5

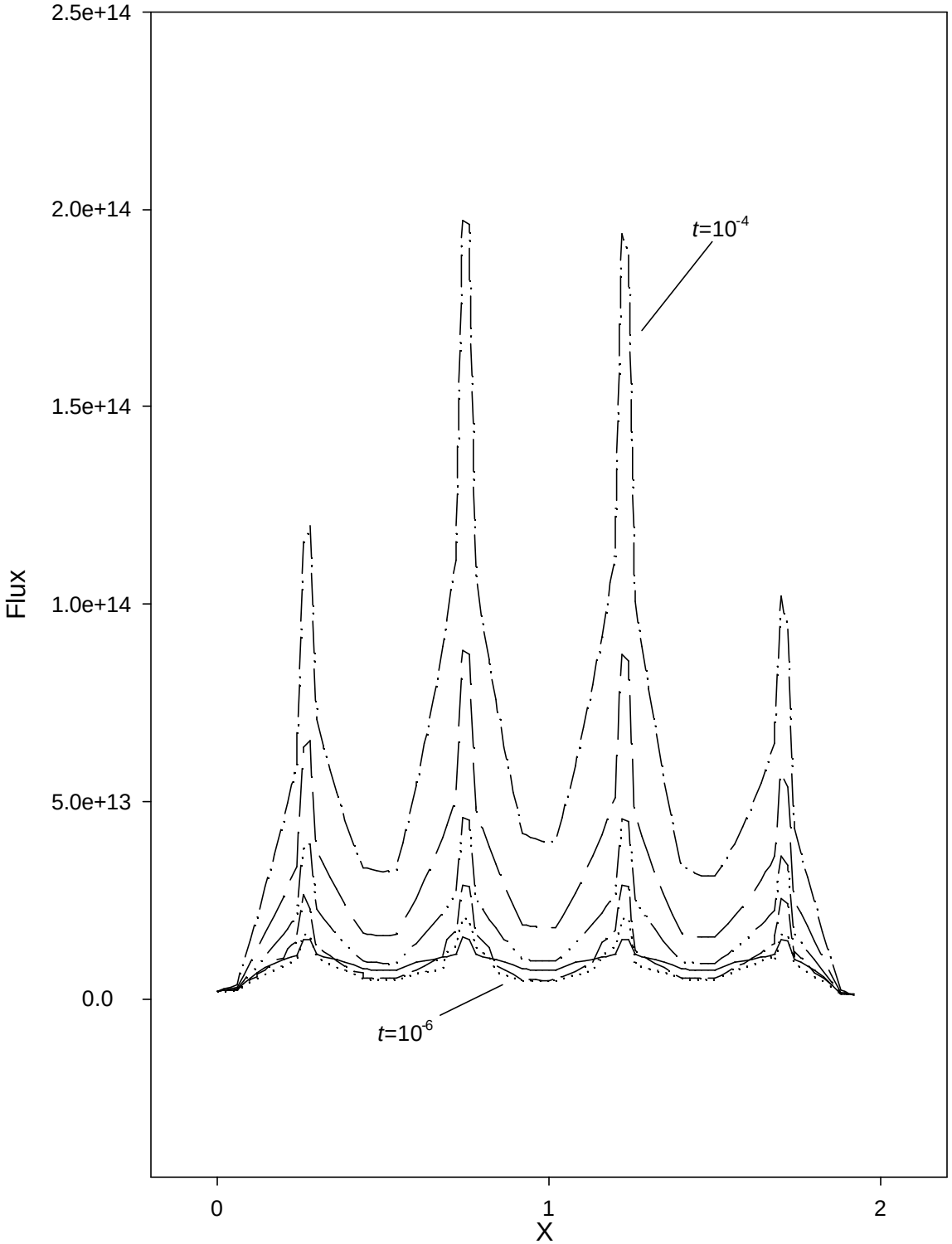


Fig. 4 32-slab configuration