# KERNFORSCHUNGSZENTRUM

# KARLSRUHE

Oktober 1973 KFK 1856

Institut für Neutronenphysik und Reaktortechnik
Projekt Schneller Brüter

**Alternative Numerical Methods for One-Dimensional Multigroup Diffusion Problems**

H.B. Stewart

GESELLSCHAFT FÜR KERNFORSCHUNG M.B.H.

KARLSRUHE

GESELLSCHAFT FÜR KERNFORSCHUNG M.B.H.
KARLSRUHE

KERNFORSCHUNGSZENTRUM KARLSRUHE

KFK 1856

Institut für Neutronenphysik und Reaktortechnik

Projekt Schneller Brüter

# Alternative Numerical Methods for One-Dimensional Multigroup Diffusion Problems

by

H.B. Stewart

Gesellschaft für Kernforschung m.b.H., Karlsruhe

Abstract

The most popular numerical method of solving one-dimensional diffusion equations is Gaussian elimination of three-point difference formulas. However, other methods have also been used, based on the factorization of the differential equations. We attempt to clarify the theoretical relationship of these alternative methods. Then some simple numerical comparisons are made to find the most efficient method. The Gaussian elimination procedure is found to be more accurate, but to achieve this accuracy on the IBM 370/175 computer, it is shown one must use double precision arithmetic.

The implementation of these results in the one-dimensional diffusion program 06731 of the NUSYS program system is documented in the Appendices.


Vergleich numerischer Methoden für das eindimensionale Multigruppen-Diffusionsproblem


Kurzfassung

Das meist benutzte numerische Verfahren zur Lösung eindimensionaler Diffusionsgleichungen ist die Gauß'sche Elimination für Dreipunkt-Differenzenformeln. Manchmal werden aber andere Methoden benutzt, die aus der Faktorisierung der Differentialgleichungen folgen. Es wird versucht, das theoretische Verhältnis zwischen den beiden Methoden zu klären. Zur Beurteilung der Leistungsfähigkeit werden einige einfache numerische Vergleiche durchgeführt. Es wird gezeigt, daß die Gauß'sche Elimination genauer ist, daß jedoch dabei auf der IBM 370/175 in doppelter Genauigkeit gerechnet werden muß.

Die Anwendung dieser Ergebnisse auf das eindimensionale Diffusionsprogramm 06731 im NUSYS Programmsystem wird in den Anhängen dokumentiert.

Table of Contents



Zum Druck eingereicht am: 27.9.1973

## Introduction

The investigation in this paper began with two problems:

1. Experience with existing one-dimensional diffusion programs at the Karlsruhe Nuclear Research Center (programs in NUSYS, /2/, and KARCOS,/11/) had shown their accuracy to be unsatisfactory.

2. The technique of factorization of second-order ordinary differential equations, although often mentioned in the literature of numerical methods for boundary value problems, holds an unclear position;in particular, there seems to be no complete comparison, both theoretical and practical, of factorization with the common method of solving three-point difference equations by Gaussian elimination.

The second point took on practical importance because the existing one-dimensional programs at Karlsruhe do in fact use variants of factorization. This suggested the need to compare factorization with the three-point difference equations, with the aim of finding the most efficient method.

We consider homogeneous multigroup eigenvalue problems in which no up-scattering is allowed. Using the common fission source iteration (see e.g./3/), the problem reduces to solving a two-point boundary value problem for each energy group:

$$(1) \qquad \frac{1}{r^{n-1}} (Dr^{n-1}\phi')' - \Sigma\phi = -f, \qquad R_{i-1} < r < R_i,\ i = 1,\dots,p$$

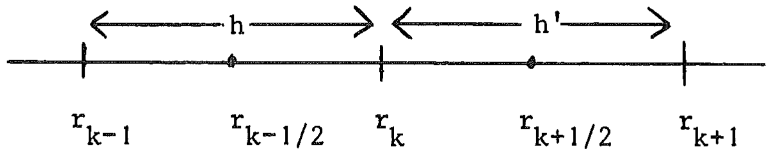
$$\alpha_o D R_o^{n-1} \phi'(R_o) - \beta_o \phi(R_o) = 0$$

$$\alpha_p D R_p^{n-1} \phi'(R_p) + \beta_p \phi(R_p) = 0$$

where f is a piecewise continuous and non-negative function (comprising scattering and fission sources), and D and $\Sigma$ are piecewise constant and positive. At points $R_i$ where D and $\Sigma$ are discontinuous, $\phi$ and $D\phi'$ are required to be continuous. The value of n in the differential equation is 3 for spherical, 2 for cylindrical, and 1 for slab geometry. For this boundary value problem we look at some alternative numerical methods of solution.

## Three-Point Difference Equations

A common numerical approach to solving (1) is to choose a set of mesh points and approximate the differential equation at each point by a difference equation involving the two neighboring points. To obtain difference formulas one can integrate the D.E. between mesh points. Let $r_k$ be a mesh point with neighbors $r_{k-1}$ and $r_{k+1}$. Call the midpoints of the respective mesh intervals $r_{k-1/2}$ , $r_{k+1/2}$. Allowing for non-uniform mesh at $r_k$, we suppose the interval widths are h, h' to the left and right of $r_k$ respectively.




First we integrate (1) from $r_{k-1/2}$ to $r_{k+1/2}$(after multiplying the D.E. by $r^{n-1}$); over a discontinuity this is done in two steps:

$$\int_{r_{k-1/2}}^{r_{k-o}} + \int_{r_{k+o}}^{r_{k+1/2}} (Dr^{n-1}\phi')'\,dr - \int_{r_{k-1/2}}^{r_{k-o}} - \int_{r_{k+o}}^{r_{k+1/2}} (\Sigma\phi-f)r^{n-1}dr = 0 \tag{2}$$

where $r_k \pm o$ indicates $r_k$ approached from the right or left. Now denoting $\phi(r_k)$ by $\phi_k$, etc., and letting $D, D'$ and $\Sigma, \Sigma'$ be the values on the left and right intervals respectively, the first part of (2) becomes

$$D\left[r_{k-o}^{n-1}\,\phi'_{k-o} - r_{k-1/2}^{n-1}\,\phi'_{k-1/2}\right] + D'\left[r_{k+1/2}^{n-1}\,\phi'_{k+1/2} - r_{k+o}^{n-1}\,\phi'_{k+o}\right]$$

By the continuity conditions,

$$D'\phi'_{k+o} = D\phi'_{k-o}$$

so the above reduces to

$$D'r_{k+1/2}^{n-1}\,\phi'_{k+1/2} - Dr_{k-1/2}^{n-1}\,\phi'_{k-1/2}$$

We now approximate the derivatives by, for example,

$$\phi'_{k-1/2} = h^{-1}(\phi_{k+o} - \phi_{k-1})$$

and since again continuity of $\phi$ means $\phi_{k+o} = \phi_{k-o} = \phi_k$, the result from above is

$$D'r_{k+1/2}^{n-1}(\phi_{k+1} - \phi_k)/h' - Dr_{k-1/2}^{n-1}(\phi_k - \phi_{k-1})/h$$

To integrate the second part of (2), $\phi$ and f are approximated by their values at $r_{k\pm o}$, giving

$$-(\Sigma'\phi_k - f_{k+o})\cdot \frac{1}{n}\cdot(r^n_{k+1/2} - r^n_k) - (\Sigma\phi_k - f_{k-o})\cdot \frac{1}{n}\cdot(r^n_k - r^n_{k-1/2})$$

Defining

$$V_{k+} = \frac{1}{n}(r^n_{k+1/2} - r^n_k)$$

which is (within a multiple of $\pi$) the volume of the n-dimensional shell with inner radius $r_k$ and outer radius $r_{k+1/2}$, and defining $V_{k-}$ similarly, the complete difference equation becomes

$$(3)\qquad D' r^{n-1}_{k+1/2}\left[\frac{\phi_{k+1} - \phi_k}{h'}\right] - D r^{n-1}_{k-1/2}\left[\frac{\phi_k - \phi_{k-1}}{h}\right]$$

$$- (\Sigma' V_{k+} + \Sigma V_{k-})\,\phi_k = -(V_{k+} f_{k+o} + V_{k-} f_{k-o})$$

At the boundary points $R_o = r_o$ and $R_p = r_N$, a pair of difference equations are found by integrating over a half-interval and using the boundary conditions. For example, at the left boundary we integrate

$$\int_{r_o}^{r_{1/2}} (D r^{n-1}\,\phi')'\,dr - \int_{r_o}^{r_{1/2}} (\Sigma\phi - f)\,r^{n-1}dr = 0$$

to get

$$Dr^{n-1}_{1/2}\,\phi'_{1/2} \;-\; Dr^{n-1}_{o}\,\phi'_{o} \;-\; (\Sigma\phi_o - f_o)\cdot\frac{1}{n}\cdot(r^n_{1/2} - r^n_o) = 0$$

We now suppose the boundary condition has $\alpha_o \neq 0$ (otherwise the difference equation at $r_o = R_o$ is trivial); further we suppose that if $r_o^{n-1} = 0$ (i.e. $n > 1$ and $R_o = 0$), then $\beta_o = 0$ (i.e. we have the boundary condition of symmetry). Then substituting the boundary condition to remove $\phi'_o$ yields

$$Dr^{n-1}_{1/2}\left[\frac{\phi_1 - \phi_o}{h}\right] - \frac{\beta_o}{\alpha_o}\,\phi_o - \Sigma V_{o+}\phi_o = -V_{o+}f_o$$

To include the case $\alpha_o = 0$ we may write

(4a) $$\alpha_o\,Dr^{n-1}_{1/2}\left[\frac{\phi_1 - \phi_o}{h}\right] - \beta_o\phi_o - \alpha_o\Sigma V_{o+}\phi_o = -\alpha_o V_{o+}f_o$$

and similarly at the right boundary

(4b) $$\alpha_p Dr^{n-1}_{N-1/2}\left[\frac{\phi_{N-1} - \phi_N}{h}\right] - \beta_p\phi_N - \alpha_p\Sigma V_{N-}\phi_N = -\alpha_p V_{N-}f_N$$

Taken together, the difference equations (3),(4a),(4b) form a system of equations of the form

$$A_k\,\phi_{k-1} - B_k\,\phi_k + C_k\,\phi_{k+1} = -F_k$$

for the $\phi_k$ in terms of the $F_k$. Here $A_o = C_N = 0$. This system has a tri-diagonal coefficient matrix which can be inverted by the Gauss method of forward elimination and backward substitution. It is well known that this procedure is numerically stable for these difference equations (see e.g. /10/). The approximation error goes to zero as $h^2$ in the limit, for continuous coefficients and constant h; this is proved in /1/. For piecewise constant coefficients the error is studied in /12/.

The elimination procedure can be described by the following equations:

$$(5)\qquad \begin{aligned} E_k &= -B_k - A_k C_{k-1}/E_{k-1} \qquad &, E_1 &= -B_1 \\ G_k &= -F_k - A_k G_{k-1}/E_{k-1} \qquad &, G_1 &= -F_1 \\ \phi_{k-1} &= (G_{k-1} - C_{k-1}\phi_k)/E_{k-1} \qquad &, \phi_N &= G_N/E_N \end{aligned}$$

Here $E_k$ is the diagonal entry and $G_k$ the right-hand side found by forward elimination; $\phi_k$ is found from right to left by the backward substitution.

We note that if points of discontinuity of f are relatively few, it is convenient to normalize (3) by dividing by $V_{k+} + V_{k-}$; with this normalization

$$F_k = (V_{k+} f_{k+o} + V_{k-} f_{k-o})/(V_{k+} - V_{k-})$$

which reduces to $F_k = f_k$ if f is continuous at $r_k$. Then we have

$$A_k = D_{k-1/2} r_{k-1/2}^{n-1} / (\Delta r_{k-1/2} (V_{k+} + V_{k-}))$$

$$B_k = (\Sigma_{k-1/2}V_{k-} + \Sigma_{k+1/2}V_{k+})/(V_{k+} + V_{k-}) + A_k + C_k$$

$$C_k = D_{k+1/2}r^{n-1}_{k+1/2} \;/\; (\Delta r_{k+1/2}(V_{k+} + V_{k-}))$$

where $\Delta r_{k+1/2} = r_{k+1} - r_k$. Finally we remark that before normalization the matrix of coefficients was symmetric; after normalization this is only true if $\Delta r_{k+1/2}$ is constant and $n = 1$.

## Continuous Factorization

Discussions of numerical solution for (1) often include a technique variously called "factorization", "simple factorization", "method of sweeps", or "chasing". Since all of these names might also apply to the procedure (5), we shall use "continuous factorization" to indicate that the continuous equation (1) is factored. (Discrete factorization is discussed in the next section)

Continuous factorization transforms the second-order linear boundary value problem (1) into three first-order initial value problems, as follows. We assume the second-order operator can be factored into

$$(6) \quad \frac{1}{r^{n-1}}\,\frac{d}{dr}\,Dr^{n-1}\frac{d}{dr} - \Sigma = \frac{1}{r^{n-1}}\left[\frac{d}{dr} + \frac{\alpha}{Dr^{n-1}}\right]Dr^{n-1}\left[\frac{d}{dr} - \frac{\alpha}{Dr^{n-1}}\right]$$

Expanding the right side, we find that the function $\alpha$ must satisfy the condition

$$(7) \qquad \alpha' + \alpha^2/\,Dr^{n-1} = r^{n-1}\,\Sigma$$

This is a Ricatti equation for $\alpha$. (For the equivalence of Ricatti equations and second-order linear equations, see /9/.) Once $\alpha$ is found, we can invert the operator (6) by successively inverting the first-order operators on the right. If

$$(8) \qquad Dr^{n-1}\phi' - \alpha\phi = \beta$$

then

$$(9) \qquad \beta' + \alpha\beta/Dr^{n-1} = -r^{n-1}f$$

The appropriate boundary conditions are found to be

for $\alpha$: $\qquad DR_o^{n-1}\,\alpha(R_o) = \beta_o/\alpha_o$

for $\beta$: $\qquad \beta(R_o) = 0$

for $\phi$: $\qquad \alpha_p DR_p^{n-1}\phi'(R_p) + \beta_p\phi(R_p) = 0$

(If $\alpha_o = 0$, the factorization is slightly different.) The continuity of $\phi$ and $D\phi'$ are implicit in the continuity of $\alpha/Dr^{n-1}$, $\beta$ and $\phi$.

The resulting method is analogous to the Gaussian elimination in (5): one first determines the auxiliary function $\alpha$ by solving an initial value problem from left to right; then one integrates (9) from left to right and finally (8) from right to left, which gives the solution.

## Formal Comparison

The analogy between continuous factorization and the process of Gaussian elimination raises an interesting question: Is there a discretization of (7) - (9) which yields (5)? One might suspect there is, but the question is complicated by the fact that there are many possible discretizations of the continuously factored equations, no one of which is obviously preferable.

/1/ gives a result relating the two methods. Considering the simple case n = 1 and h constant, one can define quantities $\alpha_k$ and $\beta_k$ by

$$(10) \qquad -E_k = C_k + h\alpha_k$$

$$G_k = h\beta_k$$

such that $\alpha_k$ , $\beta_k$, $\phi_k$ converge to solutions of (7)-(9) as $h \to o$. In fact, $\alpha_k$, $\beta_k$, $\phi_k$ satisfy the difference equations:

$$(11) \qquad \alpha_k = \alpha_{k-1} + h \cdot \left\{ \frac{-\alpha_{k-1}^2}{D_{k-1/2} + h\alpha_{k-1}} + \Sigma_{k-1/2} \right\}$$

$$\beta_k = \beta_{k-1} + h \cdot \left\{ \frac{-\beta_{k-1}\alpha_{k-1}}{D_{k-1/2} + h\alpha_{k-1}} - F_k \right\}$$

$$\phi_{k-1} = \phi_k - h \cdot \left\{ \frac{\beta_{k-1} + \alpha_{k-1}\phi_k}{D_{k-1/2} + h\alpha_{k-1}} \right\}$$

Clearly these define approximate solutions of (7)-(9).

These equations can be rearranged to resemble those in (5); in fact, an efficient computation of $\beta_k$ and $\phi_k$ would proceed similarly to the process for $G_k$ and $\phi_k$ in (5).

On the other hand, (11) does not seem to arise from (7)-(9) in a completely obvious way. We remark that (11) would appear much more arbitrary (as a discretization of (7)-(9)) in the general case; the relative simplicity of (11) depends both on h being constant and the coefficient matrix of the $A_k$, $B_k$, $C_k$ being symmetric.

Another way of comparing the Gaussian elimination method with continuous factorization is to look for a discrete factorization of the difference equation (4). In /10/ this approach is used to derive the process of Gaussian elimination; since we already have the equations (5) at hand, we can easily recover the factorization they represent. For example, the recursion relation for $G_k$ can be rewritten

$$G_k - G_{k-1} + \left( \frac{A_k}{E_{k-1}} + 1 \right) G_{k-1} = -F_k$$

or

$$\left[ \Delta^- + \left( \frac{A_k}{E_{k-1}} + 1 \right) \right] G_k = -F_k$$

where

$$\Delta^- G_k = G_k - G_{k-1}$$

Similarly the recursion relation for $\phi_k$ becomes

$$C_k \left[ \Delta^+ + \left( \frac{E_k}{C_k} + 1 \right) \right] \phi_k = G_k$$

where

$$\Delta^+ \phi_k = \phi_{k+1} - \phi_k$$

Substituting the second relation into the first yields

$$(12) \qquad \left[\Delta^- + \left(\frac{A_k}{E_{k-1}} + 1\right)\right] C_k \left[\Delta^+ + \left(\frac{E_k}{C_k} + 1\right)\right] \phi_k = -F_k$$

This is the desired discrete factorization of (4).

Again (12) is not an obvious discretization of the continuous factorization (6); in particular

$$(13) \qquad \left(\frac{A_k}{E_{k-1}} + 1\right) \neq -\left(\frac{E_k}{C_k} + 1\right)$$

The fact that (12) converges in a sense to the right side of (6) as $h \to o$ happens because the two unequal quantities in (13) approach a common value in the limit.

To summarize: Although (7)-(9) are quite analogous to (5), the discretization of (7)-(9) which yields a method equivalent to (5) is not one which is obvious from the equations (7)-(9) alone.

This unusual discretization appears as the dashed arrow in the following commutative diagram:

$$\begin{tikzcd}
2^{nd} \text{ order ODE} \arrow[r, "\text{factor}"] \arrow[d, "\text{discretize}"'] & 3 \ 1^{st} \text{ order DEs} \arrow[d, dashed, "\text{discretize}"] \\
\text{3-point difference equation} \arrow[r, "\text{factor}"] & \text{Gauss elimination formulas}
\end{tikzcd}$$

## Other Difference Formulas for the Continuously Factored System.

Some authors (/11/, /5/, /1/, /4/) offer, or seem to offer the continuously factored system (7)-(9) as a practical approach to solving (1). Since the discretization equivalent to (5) is somewhat unusual, we expect that starting with (7)-(9) and attempting some prima facie reasonable discretization, we would end with a method not equivalent to (5).

One then has a practical decision to make: whether to use (5), or some non-equivalent discretization of (7)-(9). The criteria should be low approximation error and computation time. To this end, a set of numerical tests was undertaken, involving the Gaussian elimination of three-point difference equations and different discretizations of the continuously factored equations. The following discretizations were tested:

(A) The method of the one-dimensional diffusion program in NUSYS, Program 06731. Unfortunately this program is not well documented, so the precise difference formulas are not understood. However, a numerical test does have practical significance for NUSYS users.

(B) The method of the KARCOS one-dimensional diffusion program for a large number of energy groups. These difference formulas are derived in /11/; we here indicate the derivation for the case $D, \Sigma$, $h$ constant and $n = 1$. The basic idea is to integrate each of the equations (7)-(9) over single mesh steps, using the trapezoid rule where necessary.
For example, the $\beta$ equation

$$\beta' + \alpha\beta/D = -f$$

yields

$$\beta_{i+1} - \beta_i + \frac{h}{2D}(\alpha_{i+1}\beta_{i+1} + \alpha_i\beta_i) = -\frac{h}{2}(f_{i+1} + f_i)$$

which is an implicit equation for $\beta_{i+1}$; being linear it is easily solved for $\beta_{i+1}$ explicitly. The same applies to (8). Equation (7) is also integrated, and the result is a quadratic implicit equation for $\alpha_{i+1}$. One could use the quadratic formula to find $\alpha_{i+1}$, or one could use a Newtonian iteration. The latter method might be advantageous since $\alpha_i$ is available as a good initial guess for the Newtonian iteration for $\alpha_{i+1}$. The three difference equations are finally:

(14) $$\alpha_{i+1}^{(n+1)} = \left(1 + \frac{h}{2D}\alpha_{i+1}^{(n)}\right)^{-1}\left(\alpha_i - \frac{h}{2D}\alpha_i^2 + \frac{h}{2}\Sigma\right),$$

$$\alpha_{i+1}^{(o)} = \alpha_i$$

$$\beta_{i+1} = \left(1 + \frac{h}{2D}\alpha_{i+1}\right)^{-1}\left(\beta_i - \frac{h}{2D}\alpha_i\beta_i - \frac{h}{2}(f_i + f_{i+1})\right)$$

$$\phi_{i-1} = \left(1 + \frac{h}{2D}\alpha_{i-1}\right)^{-1}\left(\phi_i - \frac{h}{2D}\alpha_i\phi_i - \frac{h}{2}(\beta_{i-1} + \beta_i)\right)$$

We note that this process uses two values of f for every mesh interval, making it possibly more costly than (5) in calculation time.

(C) Difference equations using only one value of f per interval. One way to achieve this is to follow an analogy with (2) and integrate the $\beta$ equation between midpoints of successive intervals. The resulting difference equation for $\beta$ is

$$(15) \qquad \beta_{i+1/2} = \left(1 + \frac{h}{2D}\alpha_i\right)^{-1} \left(\beta_{i-1/2} - \frac{h}{2D}\alpha_i\beta_{i-1/2} - hf_i\right)$$

This could be used with equations (14) for $\alpha$ and $\phi$, replacing $(\beta_{i-1}+\beta_i)/2$ by $\beta_{i-1/2}$ in the latter.

(D) An analytic expression for $\alpha$. Since (7) is an initial value problem (instead of a two-point boundary value problem), a problem with piecewise constant coefficients is equivalent to a sequence of initial value problems with constant coefficients. Furthermore (7) does not involve the source function f, so we might well look for an analytic solution of (7). According to /9/, one can make the transformation

$$\alpha = \frac{u'}{u} Dr^{n-1}$$

where u must then satisfy

$$u'' + \frac{n-1}{r} u' - \frac{\Sigma}{D} u = 0$$

This is a transformation of Bessel's equation (/6/); its solutions are

$$(16) \qquad u = \begin{cases} e^{\pm r/L} & n = 1 \\ I_o(r/L), K_o(r/L) & n = 2 \\ (1/r)\, e^{\pm r/L} & n = 3 \end{cases}$$

where $L = \sqrt{D/\Sigma}$. Choosing the appropriate linear combination to satisfy the initial condition, one can use the analytic expression in place of the difference equation for $\alpha$ in (14).

Numerical Comparison

To compare the accuracy of (A), (B), (C), and (D) with the method of Gaussian elimination, a simple but not unrealistic problem which has been used in /13/ was chosen. The problem represents a bare homogeneous core modeled on the ZPR-III-10 critical assembly; the number of energy groups is 26. The authors of /13/ used a zero-dimensional calculation to find a buckling which would give $k_{eff} = 1 \pm 1 \cdot 10^{-6}$ From this buckling they determined the half-thickness of a slab with $k_{eff} = 1$. The resulting homogeneous problem, although quite simple, illustrates the performance of the various numerical methods well enough to warrant a practical decision. (More complicated problems were checked for methods (5) and (A); see Appendix C.)

We do not compare the calculation times for the various methods in a precise manner. Appendix A gives a programming strategy for the Gaussian elimination method. We merely remark that similar strategies and hence similar calculation costs apply to the other methods, with one exception: as noted above, the difference equation for ß in method (B) uses two values of f for each interval, which might make it slightly more time-consuming.

The numerical results below were calculated by NUSYS Program 06731 running on the IBM 370/165; Program 06731 was modified to use the various difference equations above. For all methods, a series of mesh interval lengths h was chosen such that each is about half of the preceding one. In all cases , $k_{eff}$ converged to within $\pm\ 1.10^{-5}$ of the true value for the discrete problem.

Table I shows results for this problem using the methods (5) and (A), the old NUSYS method. One immediately sees that as $h \to o$, $k_{eff}$ does not seem to converge to the correct answer 1; in fact, there is no apparent convergence at all. For method (A) this fact was already discovered in /13/.

The especially erratic behavior of (5) casts doubt on the sufficiency of single precision arithmetic for these calculations. One should recall that the IBM 370 carries only about 7 decimal digits for single precision arithmetic.

Table II shows the same problems calculated with double precision arithmetic; more precisely, the boundary value problems (1) for the individual energy groups are solved in double precision, but the fluxes $\phi$, once found, are stored in single precision. Using this partial strategy of double precision, both (A) and (5) converge to the correct value $k_{eff} = 1$, with error falling off roughly as $h^2$ (as one would expect from the fact that approximation error for (1) decreases like $h^2$ in the limit.)

Table I.

Methods (5) and (A) in Single Precision

| Intervals | $k_{eff}$-(A) | $k_{eff}$-(5) |
|---|---|---|
| 9 | 1.00298 | 1.00088 |
| 17 | 1.00086 | 1.00024 |
| 34 | 1.00027 | 1.00002 |
| 68 | 1.00054 | 1.00010 |
| 134 | 1.00124 | .99947 |

Table II.

Methods (5) and (A) in Double Precision

| Intervals | $k_{eff}$-(A) | $k_{eff}$-(5) |
|---|---|---|
| 9 | 1.00297 | 1.00088 |
| 17 | 1.00083 | 1.00024 |
| 34 | 1.00020 | 1.00006 |
| 68 | 1.00004 | 1.00000 |
| 134 | 1.00000 | .99999 |

Table III.

Error in $k_{eff}$ for Methods (5) and (A)-(D) in Double Precision

| Intervals | Method (5) | (A) | (B) | (C) | (D) |
|---|---|---|---|---|---|
| 9 | .00088 | .00297 | -.00276 | .04470 | -.00605 |
| 17 | .00024 | .00083 | -.00077 | .01216 | -.00172 |
| 34 | .00006 | .00020 | -.00018 | .00301 | -.00042 |
| 68 | .00000 | .00004 | -.00003 | .00074 | -.00008 |
| 134 | -.00001 | .00000 | +.00004 | .00017 | -.00002 |

Table III shows the results for methods (A)-(D) using the partial strategy of double precision. The practical conclusion is clear:
three-point difference formulas solved by Gaussian elimination are substantially more accurate than any other difference formulas tested.

## Conclusion

The theoretical and practical conclusions of this investigations are:

(1) Single precision arithmetic on the IBM 370 (about 7 decimal digits) is insufficient for solving one-dimensional multigroup diffusion problems; one must solve the individual energy groups in double precision, although fluxes may be stored in single precision.

(2) The continuously factored differential equations (7)-(9) are analogous to the Gaussian elimination procedure, and there is a discretization of (7)-(9) which makes the procedures equivalent. But proceeding directly from (7)-(9) would probably yield a method not equivalent to (5).

(3) Gaussian elimination of three-point difference equations was in practice clearly more accurate than several different discretizations of (7) -(9) which were tested.

These conclusions have been implemented by rewriting NUSYS Program 06731 to use three-point difference equations and Gaussian elimination in double precision; the new version is documented in the Appendices. A replacement for the KARCOS one-dimensional program is also planned.

## Acknowledgements

The author wishes to express his thanks to many co-workers in the Institute for Neutron Physics and Reactor Technology for their generous advice and assistance, particularly with the use of computing facilities; and especially to Dr. Edgar Kiefhaber for help with realistic reactor calculations and their interpretation, and to Dr. Reimar Fröhlich for stimulating discussions and valuable encouragement.

## References

1. BABUSKA, I., M.PRAGER, and E.VITASEK: Numerical Processes in Differential Equations. London: Interscience, 1966.
   Continuous factorization appears in Section 4.2; equivalence to Gaussian elimination is shown in Section 4.5, and is used to study numerical stability of Gaussian elimination.

2. BACHMANN,H., W.HÖBEL, et al.: NUSYS program descriptions, unpublished
   A system largely for diffusion theory calculations in zero and one dimension.

3. BELL, G.I., and S.GLASSTONE: Nuclear Reactor Theory. New York: Van Nostrand, 1970
   The volume integration method of deriving three-point difference equations is described on p.146; overall multigroup computation strategy is outlined on pp. 190 ff.

4. BEREZIN,I.S., and N.P. ZHIDKOV: Computing Methods, Volume II. Oxford: Pergamon, 1965
   An extensive text which mentions continuous factorization at pp. 545 ff.

5. BUTLER, M.K., and J.M. COOK : "One Dimensional Diffusion Theory," in Greenspan, H., C.N. Kelber, and D. Okrent, Computing Methods in Reactor Physics, New York: Gordon and Breach,1968
   Both Gaussian elimination and continuous factorization are discussed in detail; extensive references.

6. COURANT, R., and D. HILBERT: Methods of Mathematical Physics, Volume II. New York, Interscience, 1962.
Fundamental solutions of $\Delta u + cu = 0$ are found in terms of Bessel functions at p. 244.

7. EHRLICH,R., and H. HURWITZ, JR.: "Multigroup Methods for Neutron Diffusion Problems," Nucleonics 12 , No.2 (1954) 23-30.
The method of three-point formulas is described, but without using the words "Gaussian elimination"; in fact, the formalism is more like solution of continuously factored equations.

8. GODUNOV,S.K., and V.S. RYABENKI: Theory of Difference Schemes. Amsterdam: North-Holland, 1964.
Contains an early paper by I.M. Gelfand and O.V. Lokutsievski on continuously factored equations and analogy with Gaussian elimination.

9. KAMKE, E.: Differentialgleichungen, Lösungsmethoden und Lösungen, I. Gewöhnliche Differentialgleichungen. Leipzig: Akademische, 1956.
At p. 22 the equivalence of Ricatti's equation and the general second-order linear equation is shown.

10. MARCHUK, G.I.: Numerical Methods for Nuclear Reactor Calculations. New York: Consultants Bureau,1959 (Translation of Supplement Nos. 3-4 to Soviet Journal of Atomic Energy, 1958).
Chapter 8 thoroughly discusses three-point difference equations, and Chapter 9 considers their solution by Gaussian elimination.

11. SANITZ,D.: "Ein Programm zur Berechnung des Neutronenflusses in eindimensionaler konsistenter P1-Approximation für viele Energiegruppen". Report KFK 1258 (1972).

A multigroup program is outlined, using method (B) above.

12. STEWART, H.B.: "Choosing a Spatial Mesh for a One-Dimensional Multigroup Diffusion Calculation," unpublished

Estimates of the approximation error for the three-point difference equations are obtained.

13. THIEM,D., and E. KIEFHABER: "Untersuchungen zur Genauigkeit der in Karlsruhe häufig benutzten ein- und zwei-dimensionalen Diffusionsprogramme," unpublished

A careful numerical study of the accuracy of the previous version of NUSYS Program 06731.

14. VARGA, R.S.: Matrix Iterative Analysis. Englewood Cliffs: Prentice Hall,1962.

Polynomial acceleration of iterative procedures is discussed in Chapter 5.

# APPENDIX A

## Description of Changes in Program 06731

Following the conclusions above, Program 06731 in NUSYS has been reprogrammed to use the three-point difference formulas and to solve them by Gaussian elimination. At the same time some other improvements to 06731 have also been made. The purpose of this Appendix is to document those changes.

For reference, the multigroup eigenvalue problem equations can be written

$$\text{(A 1)} \quad -\nabla\cdot(D_g\nabla\phi_g) + (\sigma_{rem,g} + D_g B^2)\,\phi_g = \sum_{h<g} \sigma_{scat,h\to g}\,\phi_h + \frac{1}{k_{eff}}\,\chi_g \sum_h \nu\sigma_{fis,h}\phi_h$$

for g = 1,2, ... G (g = 1 is the group of highest energy). The coefficients are all non-negative ($D_g$ are positive) and assumed constant for each material region. The adjoint problem is the same but with h and g interchanged in scattering and fission cross sections. The external source problem is

$$\text{(A 2)} \quad -\nabla\cdot(D_g\nabla\phi_g) + (\sigma_{rem,g} + D_g B^2)\,\phi_g = \sum_{h<g} \sigma_{scat,h\to g}\,\phi_h + \chi_g \sum_h \nu\sigma_{fis,h}\,\phi_h + S_g$$

where the $S_g$ are the external source. To these equations one applies fission source iterations, as described in /3/.

Difference Equations and Computing Strategy

The process of one fission source iteration can be summarized as receiving an n-th approximation of the fission source

$$f^{(n)} = \frac{1}{k_{eff}^{(n)}} \sum_h \nu\phi_{fis,h} \, \phi_h^{(n)}$$

and using this to find new approximate fluxes $\phi_g^{(n+1)}$ and the new approximate fission source

$$F^{(n+1)} = \sum_h \nu\sigma_{fis,h} \, \phi_h^{(n+1)}$$

We denote this single iteration by the operator L:

$$\text{(A 3)} \qquad L : f^{(n)} \longmapsto F^{(n+1)}$$

(Although not explicitly indicated, we are referring to the discrete problem for a certain spatial mesh.)
In NUSYS Program 06731, each application of L is calculated by subroutine CORK1.

CORK1 has been completely reprogrammed to use the difference equations (3) (with the renormalization as explained following (5)), and (4a) and (4b), solved by Gaussian elimination (5) in double precision arithmetic.

Of course the difference equations need not be set up during each iteration. So long as the number of groups is not too large (that is, provided the time spent summing scattering terms is not overwhelming), one should try to minimize the time spent solving difference equations during each outer iteration. By using three words of storage per mesh point per energy group, one could calculate and store all the difference equation coefficients $A_k$, $B_k$, $C_k$ before iterations begin.

Still more time during the iterations can be saved by observing that the first equation in (5) does not include the source $F_k$. Hence it can be solved beforehand. To solve the $G_k$ and $\phi_k$ equations requires, for example, that $A_k/E_{k-1}$, $1/E_{k-1}$, and $C_{k-1}/E_{k-1}$ be stored and available during the iterations - again three quantities per space-energy point. This is the strategy employed in the subroutine CORK1. During the outer iterations, inverting the difference equations involves just three multiplications and two additions per space-energy point.

## Handling Discontinuities

Because discontinuities are allowed in the cross-sections, and since we choose certain mesh points coinciding with the points of discontinuity, the fission sources $f^{(n)}$ and $F^{(n+1)}$ must in general be stored with two values for such points. For example, let us consider a point of discontinuity, and denote limit values from the left by $[\ ]_-$, from the right by $[\ ]_+$. According to the difference equations, iteration n + 1 will require

$$v_+ \left[ \chi_g f^{(n+1)} \right]_+ + v_- \left[ \chi_g f^{(n+1)} \right]_-$$

where $v_\pm = V_\pm / (V_+ + V_-)$, and the mesh point index has been suppressed. If $\chi_g$ is not material dependent, then this is, neglecting $k_{eff}^{(n+1)}$,

$$v_+ \left[ \chi_g \sum_h \nu\sigma_{fis,h} \, \phi_h^{(n+1)} \right]_+ \quad + \quad v_- \left[ \chi_g \sum_h \nu\sigma_{fis,h} \, \phi_h^{(n+1)} \right]_-$$

$$= \chi_g \left\{ v_+ \sum_h \left[ \nu\sigma_{fis,h} \right]_+ \phi_h^{(n+1)} + v_- \sum_h \left[ \nu\sigma_{fis,h} \right]_- \phi_h^{(n+1)} \right\}$$

In this case, even at points of discontuity it suffices to use only one value, the quantity in braces, in building $F^{(n+1)}$. If $\chi_g$ is material dependent, however, one must retain two separate values.

For this reason, Program 06731 was changed so that fission sources are always handled internally with two values at each material interface point.

## Convergence Acceleration

Another problem with Program 06731 mentioned in /13/ had been the sometimes slow convergence of outer iterations. Formerly, the acceleration was by over-relaxation; this has been replaced by Tchebyshev polynomial acceleration. We include here a sketch of this well-known method.

The process of fission power iterations without acceleration can be written as

(A 4)
$$F^{(n+1)} = L \, f^{(n)}$$
$$f^{(n+1)} = F^{(n+1)} / \| F^{(n+1)} \|_1$$

where

$$\| F \|_1 = \Sigma F_k \Delta V_k$$

and

$$\| F^{(n+1)} \|_1 = k_{eff}^{(n+1)} \rightarrow k_{eff} ,$$

which is the largest eigenvalue of the operator L of (A 3).

Now suppose $f^{(o)}$, the initial fission source guess, has an eigenvector expansion

$$f^{(o)} = b_1 e_1 + b_2 e_2 + \ldots$$

Here $e_1$ corresponds to the largest eigenvalue $\lambda_1 = k_{eff}$, while $e_2$ is associated with the next largest eigenvalue $\lambda_2$.

Then

(A 5) $$f^{(n)} = \Pi_n \; (b_1 \lambda_1^n e_1 + b_2 \lambda_2^n e_2 + \ldots )$$

where

$$\Pi_n = \frac{1}{k_{eff}^{(1)}} \cdot \frac{1}{k_{eff}^{(2)}} \cdots \frac{1}{k_{eff}^{(n)}}$$

The first term above is the desired eigenvector, the second term is the dominant error term.

The method of polynomial acceleration consists of choosing a polynomial with coefficients $a_j$ such that

(A 6) $$\tilde{f}^{(n)} = \sum_{j=o}^{n} a_j \, f^{(j)}$$

will have the same $e_1$ term as $f^{(n)}$ but smaller error terms. To achieve this, we first note that $\Pi_n \cdot \lambda_1^n$ approaches some finite limit as $n \to \infty$. We assume that before polynomial acceleration begins, enough iterations have been done that $k^{(n)} \simeq \lambda_1$ and hence $\Pi_n \cdot \lambda_1^n \simeq 1$ for all n. Then

$$\tilde{f}^{(n)} = b_1 \sum_{j=o}^{n} a_j e_j + b_2 \sum_{j=o}^{n} a_j (\lambda_2 / \lambda_1)^j e_2 + \ldots$$

$$= b_1 \, p_n(1) e_1 + b_2 p_n(\lambda_2 / \lambda_1) e_2 + \ldots$$

So we should choose a polynomial with $p_n(1) = 1$ and with the property of a minimized maximum value in $[0, \lambda_2 / \lambda_1]$ (and hence for $\lambda_i / \lambda_1$, $i > 1$). The choice is solved by Tchebyshev polynomials; following /14/ we choose

$$p_n(x) = T_n(2x/\rho - 1)/T_n(2/\rho - 1)$$

where $\rho = \lambda_2 / \lambda_1$ is the dominance ratio.

Rather than save all fission sources $f^{(j)}$ to compute $\tilde{f}^{(n)}$, one can exploit the recursion relation for Tchebyshev polynomials

$$T_{n+1}(Z) = 2Z \, T_n(Z) - T_{n-1}(Z)$$

to find $\tilde{f}^{(n+1)}$ in terms of $f^{(n+1)}$, $\tilde{f}^{(n)}$, and $\tilde{f}^{(n-1)}$. Letting

$$\text{(A 7)} \qquad \tilde{f}^{(n+1)} = \alpha_{n+1}\,(f^{(n+1)} - \tilde{f}^{(n)}) + \beta_{n+1}(\tilde{f}^{(n)} - \tilde{f}^{(n-1)})$$

one finds that

$$\text{(A 8)} \qquad \alpha_{n+1} = \frac{4}{\rho}\;\frac{T_n\,(2/\rho - 1)}{T_{n+1}(2/\rho - 1)} \quad , \; \alpha_1 = \frac{2}{2-\rho}$$

$$\beta_{n+1} = \frac{T_{n-1}(2/\rho - 1)}{T_{n+1}(2/\rho - 1)} \quad , \quad \beta_1 = 0$$

It remains only to show how $\rho = \lambda_2/\,\lambda_1$ is found. For this purpose one performs preliminary iterations. Referring back to (A 5), for unaccelerated iterations we have

$$f^{(n)} - f^{(n-1)} = \Pi_n\;(b_1\lambda_1^n e_1 + b_2\lambda_2^n e_2 + \ldots\;)$$

$$- \;\Pi_{n-1}\;(b_1\lambda_1^{n-1}e_1 + b_2\lambda_2^{n-1}e_2 + \ldots)$$

$$\simeq \Pi_{n-1}\;(b_1(\lambda_1^{n-1} - \lambda_1^{n-1})e_1 + b_2(\lambda_2^n\,\lambda_1^{-1} - \lambda_2^{n-1})e_2 + \ldots\;)$$

$$\simeq \Pi_{n-1}\;b_2\,\lambda_2^{n-1}\;(\lambda_2/\,\lambda_1 - 1)e_2 + \ldots$$

Dropping the terms with smaller eigenvalues, we take the inner product

$$(f^{(n)} - f^{(n-1)}, f^{(n)} - f^{(n-1)})$$

$$\simeq \Pi_{n-1}^2 \; b_2^2 \; \lambda_2^{2(n-1)} \; (\lambda_2 / \lambda_1 - 1)^2 \; (e_2, e_2)$$

and

$$\text{(A 9)} \quad \frac{(f^{(n+1)} - f^{(n)}, f^{(n+1} - f^{(n)})}{(f^{(n)} - f^{(n-1)}, f^{(n)} - f^{(n-1)})} \simeq \frac{\Pi_n^2}{\Pi_{n-1}^2} \cdot \lambda_2^2 \simeq \left( \frac{\lambda_2}{\lambda_1} \right)^2$$

If, as in Program 06731, both the problem for L and the problem for its adjoint $L^*$ are solved concurrently, we may use instead of (A 9) the expression

$$\text{(A 10)} \quad \frac{(f^{(n+1)} - f^{(n)}, f^{*(n+1)} - f^{*(n)})}{(f^{(n)} - f^{(n-1)}, f^{*(n)} - f^{*(n-1)})} \simeq$$

$$\left( \frac{\lambda_2}{\lambda_1} \right)^2 \simeq \rho^2$$

Finally we note that the preliminary iterations, used to reach an estimate of $\rho$, can also be accelerated, at least by over-relaxation. An over-relaxation parameter can be found from the Tchebyshev polynomial $T_1$ to be

$$\alpha = \frac{2}{2-\rho}$$

Then for preliminary iterations we replace (A 4) by

(A 11)
$$F^{(n+1)} = L\,\hat{f}^{(n)}$$
$$f^{(n+1)} = F^{(n+1)} / \left\| F^{(n+1)} \right\|_1$$
$$\hat{f}^{(n+1)} = \hat{f}^{(n)} + \alpha_{n+1}(f^{(n+1)} - \hat{f}^{(n)})$$

One finds that the dominance ratio of an over-relaxation step, $\rho(\alpha_n)$ , is estimated by

(A 12)
$$\rho(\alpha_n) \simeq \frac{(\alpha_n(f^{(n+1)} - \hat{f}^{(n)}),\ \alpha_n(f^{*(n+1)} - \hat{f}^{*(n)}))}{(\hat{f}^{(n)} - \hat{f}^{(n-1)},\ \hat{f}^{*(n)} - \hat{f}^{*(n-1)})}$$

and is related to $\rho$ by

(A 13)
$$\rho = \rho(\alpha_n)/\alpha_n + 1 - 1/\alpha_n$$

So for the eigenvalue problem we proceed as follows. Preliminary iterations are performed using over-relaxation. We begin with $\alpha_1 = 1$; after each step we get a new estimate of $\rho$ using (A 12) and (A 13). If this estimate is not close enough to the previous estimate, we continue with over-relaxation (using a new parameter based on the new estimate of $\rho$ ). Once the estimate of $\rho$ converges sufficiently, we use the last estimate to begin Tchebyshev acceleration using (A 8).

Convergence acceleration for external source problems is essentially the same. The fission source iterations without acceleration are

$$F^{(n+1)} = LF^{(n)} + K \quad , \; F^{(o)} = K \tag{A 14}$$

where K is the result of the external source with a fission source guess of zero. If F is the exact solution, then defining

$$\hat{F}^{(n)} = F^{(n)} - F$$

and

$$\hat{F}^{(o)} = b_1 e_1 + b_2 e_2 + \ldots$$

where $e_i$ are the same eigenvectors as before, we find

$$\hat{F}^{(n+1)} = L\hat{F}^{(n)}$$

and so

$$\hat{F}^{(n)} = b_1 \lambda_1^n e_1 + b_2 \lambda_2^n e_2 + \ldots$$

from which

$$F^{(n)} - F^{(n-1)} = \hat{F}^{(n)} - \hat{F}^{(n-1)} = b_1 \lambda_1^{(n-1)} (\lambda_1 - 1)\, e_1 + \ldots$$

and hence

$$\frac{(F^{(n+1)} - F^{(n)},\ F^{(n+1)} - F^{(n)})}{(F^{(n)} - F^{(n-1)},\ F^{(n)} - F^{(n-1)})} \simeq \lambda_1^2 = \rho^2 .$$

Again defining

$$\tilde{F}^{(n)} = \sum_{j=o}^{n} a_j F^{(j)}$$

we seek polynomials $p_n(x)$ with $p_n(1) = 1$ which minimize

$$\tilde{F}^{(n)} - F = \sum_{j=o}^{n} a_j F^{(j)} - F = \sum_{j=o}^{n} a_j (F^{(j)} - F)$$

$$= \sum_{j=o}^{n} a_j \hat{F}^{(j)} = b_1 p_n (\lambda_1) e_1 + \dots$$

So the appropriate polynomials are

$$p_n(x) = T_n(2x/\rho - 1)/T_n(2/\rho - 1)$$

and the acceleration method is just as for the eigenvalue problem.

## Bounds for the Eigenvalue

Using matrix properties of L it is possible to establish general bounds for the true value of $k_{eff}$ for the particular discrete problem (/14/, p.32). Applying these to our case we have

$$\text{(A 15)} \quad \lambda_m = \min \frac{F^{(n+1)}}{f^{(n)}} \leqslant k_{eff} \leqslant \max \frac{F^{(n+1)}}{f^{(n)}} = \lambda_M$$

where the minimum and maximum are over points in the space mesh. Program 06731 now prints the bounds $\lambda_m$ and $\lambda_M$ together with the final estimate of $k_{eff}$. Furthermore, for each iteration the value $(\lambda_M - \lambda_m)/\lambda_M$ is printed as a measure of convergence of the fission source.

## Further Program Options

Program 06731 has also been changed by adding some program options.

(1) The external source problem (A 2) can now be solved; convergence is guaranteed for subcritical problems.

(2) A radius criticality search can be performed in two ways. Formerly the size of a single material region was varied; that is, outer regions were displaced parallel. Now it is also possible to shift one material region into the neighboring region, so that only one material interface is moved.

(3) Time-eigenvalue calculations are now available. This involves augmenting the removal cross section by a term to read

$$\sigma_{rem,g} + D_g B^2 + \alpha/V_g$$

where $V_g$ is the mean neutron velocity for the group. The user may give values of $\alpha$ for which $k_{eff}$ is to be found; or he may request a criticality search by varying $\alpha$.

# APPENDIX B

## Program Listing

The following is a FORTRAN source statement listing of the subroutine CORK1 in NUSYS Program 06731, as reprogrammed for three-point difference equations and Gaussian elimination.

CORK1 performs a single fission source iteration, as denoted by the operator L in Appendix A.

```
      SUBROUTINE  CORK1I (XL,NXL,SIGMA,HF,EL,NPK,NGRU)
C  SUBROUTINE FOR A SINGLE INVERSION OF A SET OF MULTIGROUP EQUATIONS.
C    REPROGRAMMED FOR STANDARD 3-POINT DIFFERENCE FORMULAS AND
C    GAUSSIAN ELIMINATION IN DOUBLE PRECISION. (D.P. IS NECESSARY
C    WITH IBM 360 TO AVOID ROUNDOFF ERROR.)
C    REPROGRAMMED  6/73  BY H.B. STEWART
      DIMENSION  XL(1),NXL(1),SIGMA(1),HF(NPK,NGRU,2),EL(NPK,NGRU,3)
      COMMON /M06731/     SW(80), EPS(3),
     1          RADIUS(36), INTERV(35), DELTA(35),
     2          FM(35), FP(35), NGEOZ(35), LGEOZ(35),
     3          VELINV(60), RDBED(60,4), ALFARG(20), LF(12), LFN(12),
     4          DS(185),
     5          FHILF1(185), FHILF2(185),
     6          FADJ(185), FNEU(185), FNORM(185), FALT(185),
     7          DIAG1(150), DIAG2(150), DIAG3(150)
        DOUBLE PRECISION  DIAG1,DIAG2,DIAG3,QU(150), TS,TS1,TS2
        DIMENSION  RNM1(150)
        EQUIVALENCE (DIAG1(1), QU(1)), (RNM1(1), FHILF2(1))
C  PROBLEM SPECIFIERS...
      EQUIVALENCE  (SW(1),NGRUP), (SW(2),MZONE), (SW(3),NPKT),
     1             (SW(4),NGEO), (SW(5),NZONE), (SW(19),NPKTZ),
     2             (SW(36),MIX), (SW(38),MZ), (SW(39),NG)
C SWITCHES HAVE THE FOLLOWING VALUES AND MEANINGS
C        NADJ      =0  FOR NORMAL PROBLEM
C                  =1  FOR ADJOINT
C        NHOM      =0  FOR HOMOGENEOUS EIGENVALUE PROBLEM
C                  =1  FOR EXTERNAL SOURCE PROBLEM
C        NGAMMA    =0  ON FIRST CALL OF CORK1
C                  =1  ON SUBSEQUENT CALLS
      EQUIVALENCE  (SW(20),NADJ), (SW(21),NHOM), (SW(22),NGAMMA)
C  POINTERS TO SPECIFIC KINDS OF GROUP CONSTANTS WITHIN SIGMA...
      EQUIVALENCE  (SW(42),NH), (SW(43),NHB), (SW(44),NHF),
     1            (SW(45),NHC), (SW(46),NHC), (SW(47),NHR),
     2            (SW(48),NHD), (SW(49),NHT), (SW(50),NHS)
      RETURN
C
      ENTRY  CORK1
      IF (NGAMMA.EQ.0)  GO TO 60
   20 NG1 = NGRUP
   31 DO  32  K = 1, NPKTZ
   32   FNEU(K) = 0.
      IADJ = NADJ + 1
      NC1 = NHC
      NC2 = NHF
      IF (NADJ .EQ. 0)  GO TO 33
        NC1 = NHF
        NC2 = NHC
C
   33 NI = 0
  333 NI = NI + 1
      IF (NI .GT. NG1)  RETURN
C  BUILD THE SOURCE FUNCTION FOR THIS GROUP
   34 NI4 = NGRUP - NI + 1
      IF (NADJ)  38,35,38
   35   NI4 = NI
        IF (NHOM)  36,38,36
```

```
   36      DO  37   K = 1, NPKT
   37        QU(K) = HF(K,NI4,2)
           IF (NGAMMA)  40,41,40
   38 DO  39  K = 1, NPKT
   39   QU(K) = 0.D0
   40 CALL  QUCALC (NC1,NI4,QU(1),FALT(1) )
C  SUM SCATTERING CONTRIBUTIONS FROM HIGHER ENERGY GROUPS
   41 LZ2 = 0
   42   LZ2 = LZ2 + 1
        IF (LZ2 .EQ. NI)  GO TO 48
        NI1 = LZ2
        NZ2 = NI
        NII = NI1
        IF (NADJ)  44,45,44
   44     NI1 = NI4
          NZ2 = NGRUP - LZ2 + 1
          NII = NZ2
   45   CALL  QSCALC (NI1+(NZ2-1)*MIX*(NGRUP+7)+NHS,
     *                    QU(1), HF(1,NII,IADJ))
   46   GO TO 42
   48 CONTINUE
C  GAUSSIAN ELIMINATION TO SOLVE THE DIFFERENCE EQUATIONS, USING
C    PARTIALLY ELIMINATED COEFFICIENTS IN  EL
      QU(1) = - QU(1)*RDBED(NI4,2)
      QU(NPKT) = QU(NPKT)*RDBED(NI4,4)
   50 QU(1) = QU(1) * EL(1,NI4,1)
      DO  51  K = 2, NPKT
   51   QU(K) = EL(K,NI4,1) * (QU(K) - EL(K,NI4,3) * QU(K-1))
      TS = 0.D0
      K = NPKT
   52   TS = EL(K,NI4,2) * TS + QU(K)
        HF(K,NI4,IADJ) = TS
        K = K - 1
        IF (K .GT. 0)  GO TO 52
C  CONTRIBUTION TO THE NEXT FISSION SOURCE
   54 CALL  QNCALC (NC2,NI4, FNEU(1), HF(1,NI4,IADJ))
      GO TO 333
C
C  INITIALIZATION.   SET UP COEFFICIENT MATRIX FOR EACH GROUP AND
C    PERFORM FORWARD ELIMINATION, STORE THE PARTIAL RESULTS IN  EL.
C    ALSO OTHER CONSTANTS RELATED TO GEOMETRY.
   60 DO  61  NZ = 1, NZONE
   61   DELTA(NZ) = (RADIUS(NZ+1)-RADIUS(NZ)) /FLOAT(INTERV(NZ))
      K2 = 1
      DO  64  NZ = 1, NZONE
        K1 = K2
        K2 = K2 + INTERV(NZ)
        TS1 = RADIUS(NZ) - DELTA(NZ)
        TS2 = DELTA(NZ) /2
        DO  64  K = K1, K2
          TS1 = TS1 + DELTA(NZ)
          IF (NGEO .NE. 0)  GO TO 63
            RNM1(K) = 1.
            GO TO 64
   63     RNM1(K) = TS1 + TS2
          IF (NGEO .EQ. 1)  GO TO 64
            RNM1(K) = RNM1(K) * RNM1(K)
   64     CONTINUE
```

```
C   VOLUME ELEMENT ASSOCIATED WITH EACH FISSION SOURCE POINT
C     (I.E., ZONE BOUNDARY POINTS COUNTED TWICE)
      LZ = NGEO + 1
      K2 = -1
      DO  66    NZ = 1, NZONE
        K1 = K2 + 2
        K2 = K2 + INTERV(NZ) + 1
        TS1 = RADIUS(NZ)
        TS2 = DELTA(NZ) /2
        DO  66    K = K1, K2
          DS(K) = ((TS1+TS2)**LZ - TS1**LZ) /LZ
   66     TS1 = TS1 + DELTA(NZ)
      K2 = 0
      DO  67    NZ = 1, NZONE
        K1 = K2 + 2
        K2 = K2 + INTERV(NZ) + 1
        TS1 = RADIUS(NZ)
        TS2 = DELTA(NZ) /2
        DO  67    K = K1, K2
          TS1 = TS1 + DELTA(NZ)
          IF (K .EQ. K2)  DS(K) = 0.
   67     DS(K) = DS(K) + (TS1**LZ - (TS1-TS2)**LZ) /LZ
C   FM, FP  (VOLUME WEIGHTS FOR POINT PAIRS ON ZONE BOUNDARIES)
      FP(1) = 1.
      FM(NZONE) = 1.
      NZ1 = NZONE - 1
      IF (NZ1)  72,72,70
   70 K = 0
      DO  71    NZ = 1, NZ1
        K = K + INTERV(NZ) + 1
        K1 = K + 1
        TS1 = DS(K)
        TS2 = DS(K1)
        TS = TS1 + TS2
        FM(NZ) = TS1 / TS
   71   FP(NZ+1) = TS2 / TS
   72 CONTINUE
      NG1 = NGRUP
      DO  90   NI = 1, NG1
        NH = NI * MZ
        LZ = NGEOZ(1) + NH + NHD
        LZ1 = NGEOZ(1) + NH + NHT
C   CONSTRUCTION OF DIFFERENCE EQUATIONS
        TS1 = RADIUS(1)
        IF (NGEO .EQ. 0)  TS1 = 1.D0
        IF (NGEO .EQ. 2)  TS1 = TS1 * TS1
        TS2 = RADIUS(1) + DELTA(1) /2
        IF (NGEO .EQ. 0)  TS2 = 1.D0
        IF (NGEO .EQ. 2)  TS2 = TS2 * TS2
        TS = DS(1) * DELTA(1) / SIGMA(LZ) / TS2
        DIAG1(1) = 0.D0
        DIAG2(1) = - RDBED(NI,2) * (SIGMA(LZ1) + 1.D0/TS)
     *                   + RDBED(NI,1) * DELTA(1) * TS1 / TS2 / TS
        DIAG3(1) =    RDBED(NI,2) / TS
        K2 = 1
        DO  76   NZ = 1, NZONE
          K1 = K2 + 1
          K2 = K2 + INTERV(NZ)
```

```
          DO  75  K = K1, K2
            KK = K + NZ - 1
            TS2 = DS(KK)
            TS1 = RNM1(K-1)
            DIAG1(K) = - SIGMA(LZ) * TS1 / DELTA(NZ) / TS2
            TS1 = RNM1(K)
            DIAG3(K) = - SIGMA(LZ) * TS1 / DELTA(NZ) / TS2
            DIAG2(K) = SIGMA(LZ1) - DIAG1(K) - DIAG3(K)
   75       CONTINUE
          IF (NZ .EQ. NZONE)  GO TO 76
          K = K2
          DIAG1(K) = DIAG1(K) * TS2
          TS2 = TS2 + DS(KK+1)
          DIAG1(K) = DIAG1(K) / TS2
          DIAG2(K) = SIGMA(LZ1) * DS(KK)
          LZ = NGEOZ(NZ+1) + NH + NHD
          LZ1 = NGEOZ(NZ+1) + NH + NHT
          DIAG3(K) = -SIGMA(LZ) * TS1 / DELTA(NZ+1) / TS2
          DIAG2(K) = (DIAG2(K) + SIGMA(LZ1) * DS(KK+1)) / TS2
     *                 - DIAG1(K) - DIAG3(K)
   76     CONTINUE
        TS1 = RADIUS(NZONE+1)
        IF (NGEO .EQ. 0)  TS1 = 1.D0
        IF (NGEO .EQ. 2)  TS1 = TS1 * TS1
        TS2 = RADIUS(NZONE+1) - DELTA(NZONE) /2
        IF (NGEO .EQ. 0)  TS2 = 1.D0
        IF (NGEO .EQ. 2)  TS2 = TS2 * TS2
        TS = DS(NPKTZ) * DELTA(NZONE) / SIGMA(LZ) / TS2
        DIAG3(NPKT) = 0.D0
        DIAG2(NPKT) = RDBED(NI,4) * (SIGMA(LZ1) + 1.D0 / TS)
     *                 + RDBED(NI,3) * DELTA(NZONE) * TS1 / TS2 / TS
        DIAG1(NPKT) = - RDBED(NI,4) / TS
C  PERFORM FIRST ELIMINATION, STORE RESULTS IN  EL
        TS = 1.D0 / DIAG2(1)
        TS1 = - DIAG3(1) * TS
        EL(1,NI,1) = TS
        EL(1,NI,2) = TS1
        DO  77  K = 2, NPKT
          TS = 1.D0 / (DIAG2(K) + DIAG1(K) * TS1)
          TS1 = - TS * DIAG3(K)
          EL(K,NI,1) = TS
          EL(K,NI,2) = TS1
          EL(K,NI,3) = DIAG1(K)
   77     CONTINUE
   90   CONTINUE
      GO TO 20
      END
```

```
      SUBROUTINE  QUINIT (MZ,NZONE,NGRUP,SIGMA,XL)
C  FOR THE ADDITION OF VARIOUS CONTRIBUTIONS TO SOURCE TERMS
      COMMON /MO6731/ SW(80), EPS(3), RADIUS(36), INTERV(35),
     1  DELTA(35), FM(35), FP(35), NGEOZ(35), LGEOZ(35)
      DIMENSION  SIGMA(1), XL(1), F(1), FF(1), H(1)
      DOUBLE PRECISION  F
      RETURN
C
      ENTRY  QUCALC (NZ1, NI4, F, FF)
      NH = MZ * NI4 + NZ1
      K2 = 0
      DO  20  NZ = 1, NZONE
        LZ = NH + NGEOZ(NZ)
        TS = SIGMA(LZ)
        K1 = K2 + 2
        K2 = K2 + INTERV(NZ)
        IF (TS .EQ. 0)  GO TO 20
        F(K1-1) = F(K1-1) + TS * FP(NZ) * FF(K1+NZ-2)
        F(K2+1) = F(K2+1) + TS * FM(NZ) * FF(K2+NZ)
        IF (K2 .LT. K1)  GO TO 20
        DO  10  K = K1, K2
   10     F(K) = F(K) + TS * FF(K+NZ-1)
   20   CONTINUE
      RETURN
C
      ENTRY  QSCALC (NI2, F, H)
      K2 = 0
      DO  40  NZ = 1, NZONE
        K1 = K2 + 2
        K2 = K2 + INTERV(NZ)
        LZ = NI2 + (LGEOZ(NZ) -1) * (NGRUP + 7)
        TS = XL(LZ)
        IF (TS .EQ. 0)  GO TO 40
        F(K1-1) = F(K1-1) + TS * FP(NZ) * H(K1-1)
        F(K2+1) = F(K2+1) + TS * FM(NZ) * H(K2+1)
        IF (K2 .LT. K1)  GO TO 40
        DO  30  K = K1, K2
   30     F(K) = F(K) + TS * H(K)
   40   CONTINUE
      RETURN
C
      ENTRY  QNCALC (NZ1, NI4, FF, H)
      NH = MZ * NI4 + NZ1
      K2 = 0
      DO  60  NZ = 1, NZONE
        LZ = NH + NGEOZ(NZ)
        TS = SIGMA(LZ)
        K1 = K2 + 1
        K2 = K2 + INTERV(NZ) + 1
        IF (TS .EQ. 0)  GO TO 60
        DO  50  K = K1, K2
   50     FF(K) = FF(K) + TS * H(K-NZ+1)
   60   CONTINUE
      RETURN
      END
```

# APPENDIX C

## Program 06731 Checkout

To check the new version of Program 06731 a set of test problems were prepared and run.
These problems actually serve three purposes: to verify and document the performance of the new version; to assist the user in the transition by comparing the new with the old version; and to provide benchmarks for any future changes. The test problems cover three aspects of Program 06731: numerical accuracy, effectiveness of convergence acceleration, and the proper functioning of search options and communication with other NUSYS programs.

## Accuracy Tests

A number of accuracy tests has been performed using problems from /13/, where these problems are completely described.

The first type of problem uses a 26-group representation of a single homogeneous mixture (Z1-Core of SNEAK-6A), with boundary conditions of zero current at both endpoints. Thus the flux in each group should be independent of position, and one should be able to vary the total length and/or the mesh size without changing the value of $k_{eff}$.

To check this, a first series used different values for the total width ranging 0.1 cm to 1 m. In each case a mesh of ten steps was used.

Table C-I shows the calculated eigenvalues. In all cases, convergence was obtained to $\pm 1 \cdot 10^{-6}$ in three iterations; the true value should be $1.007439 \pm 1 \cdot 10^{-6}$.

Further series involved varying the number of mesh steps from 1 to 144 for a constant value of the step width. For step widths of 10 cm and 1 cm, all values were within the range $\pm 2 \cdot 10^{-6}$ of the value 1.007439. For step width of 0.1 cm, the results are shown in Table C-II; we remark that in this case the dimensions probably do not correspond to the geometry of realistic reactor diffusion problems.

For results previously obtained with Program 06731 (i.e., method (A) in single precision), one should consult /13/.

Table C-I. $K_{eff}$ for Flat Flux Tests,

Ten Step Mesh

| Total Width (cm) | $k_{eff}$ (new 06731) |
|---|---|
| 0.1 | 1.007435 |
| 0.2 | 1.007432 |
| 0.5 | 1.007434 |
| 1.0 | 1.007432 |
| 2.0 | 1.007535 |
| 5.0 | 1.007534 |
| 10. | 1.007435 |
| 20. | 1.007536 |
| 50. | 1.007439 |
| 100. | 1.007439 |
| 200. | 1.007439 |
| 500. | 1.007438 |
| 1000. | 1.007440 |

Table C-II. $K_{eff}$ for Flat Flux Tests,

Mesh Step = 0.1cm

| Number of Steps | $k_{eff}$ (new 06731) |
|---|---|
| 1 | 1.007438 |
| 2 | 1.007438 |
| 3 | 1.007437 |
| 4 | 1.007436 |
| 5 | 1.007435 |
| 7 | 1.007433 |
| 10 | 1.007432 |
| 20 | 1.007426 |
| 50 | 1.007409 |
| 100 | 1.007390 |
| 144 | 1.007371 |

Exact value: $k_{eff} = 1.007439 \pm 1 \cdot 10^{-6}$

A second type of test problem includes the ZPR-III-10 model already used in the main body of this report. In addition to the 26-group bare core model used above, /13/ also used a 26-group model with a core and a blanket region. Furthermore, the cross sections were condensed to a single energy group, giving one-region and two-region models. All four models were established in slab, cylindrical, and spherical geometries.

Table C-III shows 26-group one-zone results for both the new 06731 and the old version. In all cases the error criterion for $k_{eff}$ was $\pm$ $10^{-5}$ between successive iterations, while the pointwise criterion for the fission source was one part in $10^{-4}$. These results appear as the upper curves in Figures 1 and 2 for slab and spherical geometries, respectively.

For these 26-group one-zone problems 06731 now converges to the correct value $k_{eff} = 1$, with error decreasing roughly as $h^2$. In every one of these cases, 06731 gives a noticeably better $k_{eff}$ value than the old version.

Table C-III. $K_{eff}$ for ZPR-II-10 26-Group One-Zone Models

| Geometry | Mesh Steps | $k_{eff}$ (new 06731) | $k_{eff}$ (old 06731) |
|---|---|---|---|
| Sphere | 18 | 1.00106 | 1.00504 |
| | 35 | 1.00028 | 1.00135 |
| | 70 | 1.00006 | 1.00032 |
| | 140 | 1.00001 | 1.00049 |
| Cylinder | 14 | 1.00100 | 1.00431 |
| | 27 | 1.00027 | 1.00118 |
| | 54 | 1.00006 | 1.00028 |
| | 108 | 1.00000 | 1.00050 |
| Slab | 9 | 1.00088 | 1.00298 |
| | 17 | 1.00024 | 1.00086 |
| | 34 | 1.00006 | 1.00027 |
| | 68 | 1.00000 | 1.00054 |
| | 136 | .99999 | 1.00124 |

Exact value: $k_{eff} = 1 \pm 1 \cdot 10^{-6}$

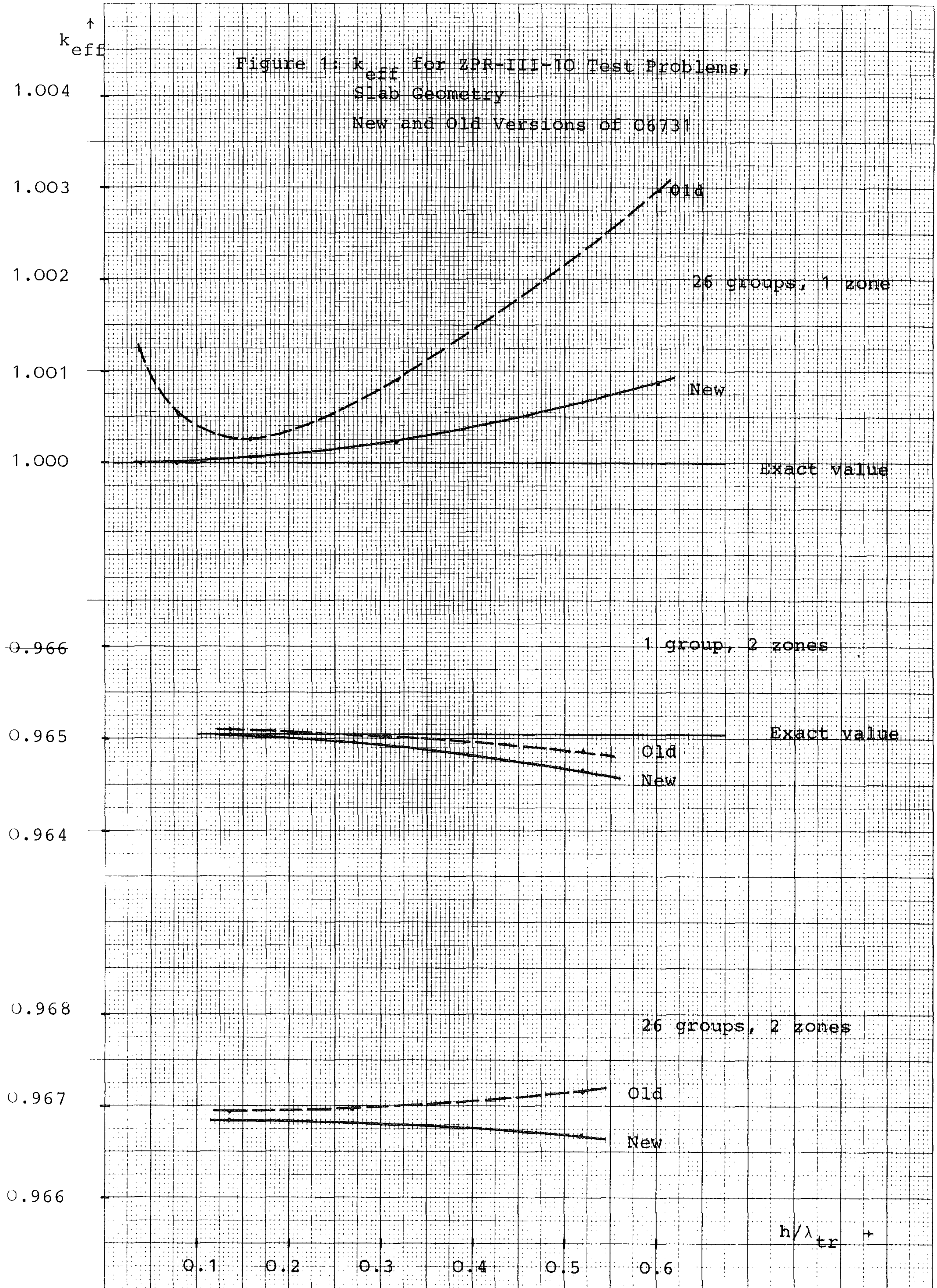


Figure 1: $k_{eff}$ for ZPR-III-10 Test Problems,
Slab Geometry
New and Old Versions of 06731
$k_{eff}$
1.004
1.003
1.002
1.001
1.000
Old
26 groups, 1 zone
New
Exact value
0.966
1 group, 2 zones
0.965
Exact value
Old
New
0.964
0.968
26 groups, 2 zones
0.967
Old
New
0.966
$h/\lambda_{tr}$
0.1
0.2
0.3
0.4
0.5
0.6

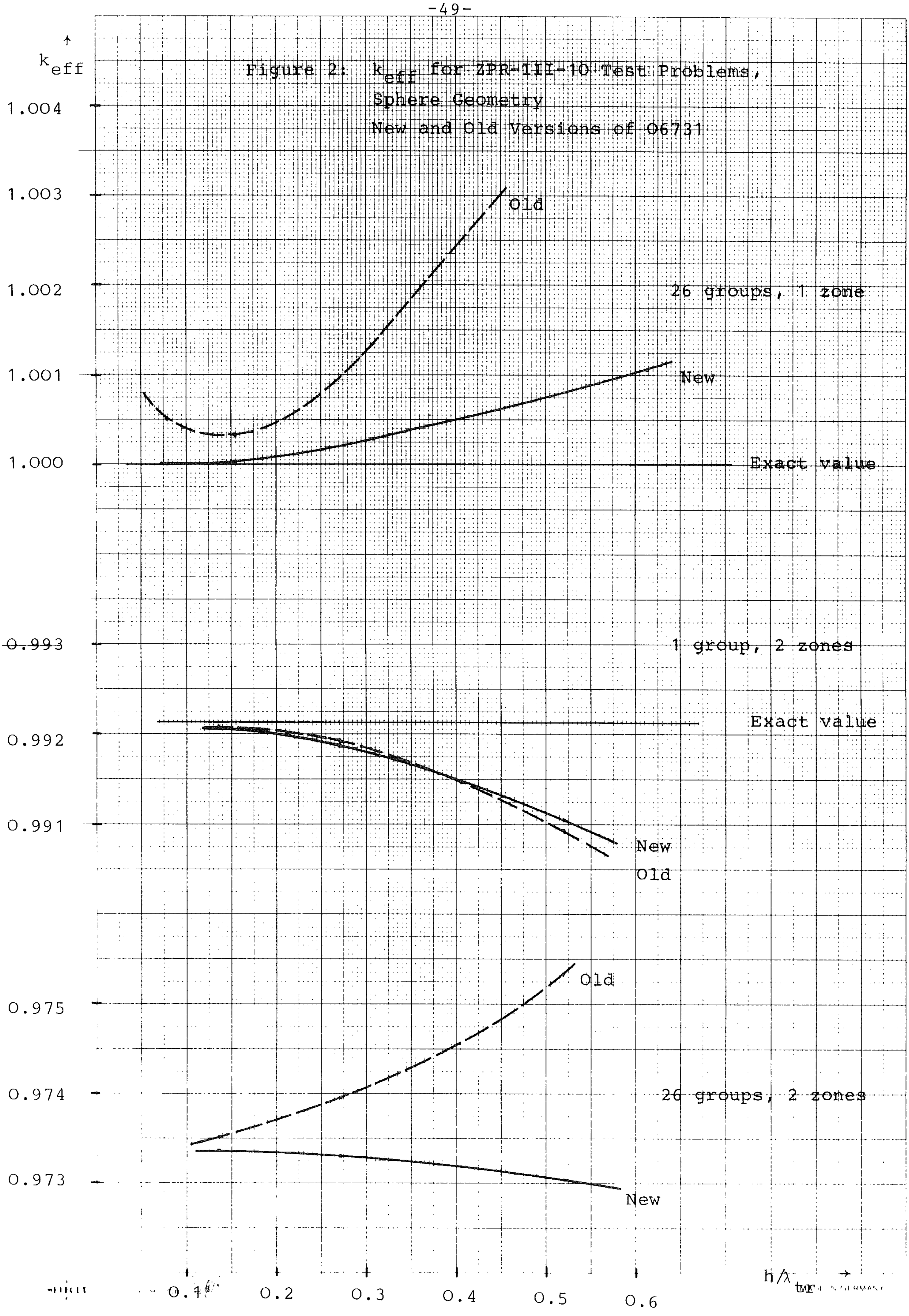


Figure 2: $k_{eff}$ for ZPR-III-10 Test Problems,
Sphere Geometry
New and Old Versions of 06731
$k_{eff}$
1.004
1.003
1.002
1.001
1.000
0.993
0.992
0.991
0.975
0.974
0.973
Old
26 groups, 1 zone
New
Exact value
1 group, 2 zones
Exact value
New
Old
Old
26 groups, 2 zones
New
0.1
0.2
0.3
0.4
0.5
0.6
$h/\lambda_{tr}$

Since the one-group one-zone problems show the same effects, we do not set them forth in detail.

Table C-IV shows one-group two-zone results for the new and old versions of 06731, again in three geometries. For each geometry, $k_{eff}$ for the continuous problem was found by a program which uses the analytic solutions (16) in each zone. We remark that to test the new 06731, a new condensation to one group was performed, using 26-group fluxes calculated by the new version of 06731; for this reason the one-group cross-sections given to the new and old versions of 06731 are slightly different. However, the difference is not noticeable in Figures 1 and 2, where the results appear as the middle curves.

Here one sees that the magnitude of error in $k_{eff}$ for the new version of 06731 is not so different from former results. In fact, in two cases the $k_{eff}$ values for slab geometry were better with the old version, although the finest mesh overshot the true value. To understand this better , values of the flux were also checked.

Table C-V gives the flux found in slab geometry at the far left (where the boundary condition is zero current) and at the material interface. From this table it is clear that the old version of 06731 was not superior for pointwise values. One also sees in the values for the old version how compensation of errors could yield better values of $k_{eff}$.

The bottom curves in Figures 1 and 2 show the results of two-zone 26-group calculations. Since no exact solution of the continous problem is available, one cannot draw rigorous conclusions for these cases. However, one does note that with the new version of 06731, convergence for the 26-group problem resembles that for the condensed one-group problem.

Table C-IV. $K_{eff}$ for ZPR-III-10 One-Group Two-Zone Models

| Geometry | Mesh Steps | $k_{eff}$ error (new 06731) | $k_{eff}$ error (old 06731) |
|---|---|---|---|
| Sphere | 31 | -.00108 | -.00118 |
| | 62 | -.00025 | -.00019 |
| | 124 | -.00006 | -.00006 |
| Exact value: | | .99214 | .99213 |
| Cylinder | 32 | -.00081 | -.00045 |
| | 64 | -.00021 | -.00006 |
| | 128 | -.00005 | -.00003 |
| Exact value: | | .97977 | .97976 |
| Slab | 28 | -.00040 | -.00017 |
| | 56 | -.00009 | .00000 |
| | 112 | -.00002 | +.00006 |
| Exact value: | | .96506 | .96505 |

Table C-V. Flux at Selected Points for One-Group Two-Zone Slab

| Point | Mesh Steps | Flux error (new 06731) | Flux error (old 06731) |
|---|---|---|---|
| Left boundary | 28 | +.0034 | +.0038 |
| | 56 | +.0009 | +.0010 |
| | 112 | +.0003 | +.0002 |
| Exact value: | | 2.1718 | 2.1719 |
| Material interface | 28 | -.0029 | -.0035 |
| | 56 | -.0007 | -.0008 |
| | 112 | -.0002 | -.0001 |
| Exact value: | | 1.0208 | 1.0207 |

Exact values are within ±1 in the least significant figure.

Finally a simple external source problem was tried. Using the 26-group one-zone problems above, one can create an external source problem with known solution as follows. First one does a radius criticality search for, say, the value $k_{eff} = (1.1)^{-1}$. One then takes the resulting geometry and fission source f, and sets

$$S_g = (0.1) \chi_g f$$

Then the solution of this external source problem should have the same solution as the eigenvalue problem with $1/k_{eff} = 1.1$.

Trying this with ten space mesh points in slab geometry produced the following results: an error criterion of one part in $10^{-5}$ for the magnitudes of successive fission source estimates was specified, and the final fission source differed by 3 parts in $10^{-5}$ from that of the eigenvalue problem. The pointwise flux values also differed by about 3 parts in $10^{-5}$ from those for the eigenvalue problem.

## Convergence Acceleration

Good convergence acceleration is important when the dominance ratio $\rho$ is nearly 1. For external source problems, this happens if the reactor model is nearly critical; a $k_{eff}$ eigenvalue calculation may have $\rho$ nearly 1 for a large power reactor.

The convergence acceleration for $k_{eff}$ calculations was checked with a model of the proposed SNR-2 fast power reactor. The problem was supplied by

E. Kiefhaber as a case for which convergence acceleration had previously been unsatisfactory; the dominance ratio is about 0.92. If one sets the pointwise criterion for fission source convergence at one part in $10^{-4}$, one finds that the final estimate of $k_{eff}$ is within $\pm 2 \cdot 10^{-5}$ of the true value for the discrete problem, and this is achieved in fewer than 30 iterations.

For external source problems, one can easily create a test problem with any desired $\rho$ by first performing a radius search for $k_{eff} = \rho$. This was done for the external source problem mentioned above as an accuracy test. With $\rho = (1.1)^{-1}$, the total power for the external source problem changed less than one part in $10^{-5}$ per iteration after about twenty iterations. For $\rho = (1.01)^{-1}$, the same criterion was satisfied after about seventy iterations.

## Proper Functioning

Finally several test problems were run to check that program control functions properly for more complicated calculations. The purpose was not to check numerical accuracy, but simply to verify that the calculations are completed without disruption.

One such test was the large reactor problem mentioned above. This was actually an enrichment iteration, which involves repeated communication between 06731 and another NUSYS program which adjusts the enrichment. Since the enrichment iteration did converge to a solution with the desired properties, we assume that the communication between programs has not been disturbed.

The two types of radius iterations were tried, using one-group two-zone problems from the accuracy test series. The input geometry was perturbed, and the previously obtained values were requested; in both cases the original geometry was found.
Finally, the two types of time-eigenvalue calculations were tried. Although the correct solutions of the problems are not known, the program did produce plausible answers without difficulties.