In summary, the cross correlation and spectral density functions have been obtained for the case of neutrons slowing down in an infinite homogeneous reactor. Given here are several examples of the results.

1. JAMES R. SHEFF and ROBERT W. ALBRECHT, "The Space Dependence of Reactor Noise, Parts I and II," *Nucl. Sci. Eng.*, **24**, 246 (1966); also, **26**, 207 (1966) and **39**, 413 (1970).
2. JAMES R. SHEFF, "Three-Group Space-Dependent Spectral Densities," *Trans. Am. Nucl. Soc.*, **12**, 709 (1969).
3. M. M. R. WILLIAMS, *Random Processes in Nuclear Reactors*, p. 119, Pergamon Press, New York (1974).
4. P. LANGEVIN, "Sur la Theorie du Mouvement Brownien," *Compt. Rendu*, p. 530 (1908).
5. M. S. BARTLETT, *Stochastic Process*, p. 53, Cambridge University Press (1966).
6. L. PAL, "Statistical Fluctuations of Neutron Multiplication," *1958 U.N. Geneva Conf.*, **16**, 687 (1958).
7. JAMES R. SHEFF, "Time and Energy Green's Function for Neutrons Slowing Down," *Trans. Am. Nucl. Soc.*, **38**, 635 (1981).

5. Multiregion Neutronics Model for Slow Transient Analysis of BWRs, *Hiroshi Motoda, Yasunori Bessho, Tamotsu Hayase, Kanji Kato (Hitachi ERL-Japan)*

Simulation technique of transient phenomena is essential for core design and plant control of boiling water reactors (BWRs). Needs to develop practical tools for space-time analyses of reactor core have increased because of increased core size.

An attempt was made to develop a multiregion neutronics model based on coarse-mesh nodal coupling method. First, a three-dimensional steady-state neutronics model[1] of Eq. (1) was extended to a dynamics model taking the balance of neutron source and delayed neutron precursors as given by Eq. (2):

$$S_i = \frac{k_{\infty i}}{\lambda}\left(\sum_j W_{ji}S_j + WS_iS_i\right) , \qquad (1)$$

where

S_i = neutron source density at node i

$k_{\infty i}$ = infinite multiplication factor at node i

W_{ji} = neutron transport kernel from node j to node i (probability that a neutron born at node j is finally absorbed at node i)

WS_i = neutron self-transport kernel

λ = eigenvalue.

$$\left.\begin{aligned} \ell_i \frac{\partial S_i}{\partial t} &= \sum_j W_{ji}S_j + \left(WS_i - \frac{\lambda_0}{k_{\infty i}} - \beta_i\right)S_i + \sum_\ell \lambda_\ell C_{\ell i} \\ \frac{\partial C_{\ell i}}{\partial t} &= \beta_{\ell i}S_i - \lambda_\ell C_{\ell i} \end{aligned}\right\} , \qquad (2)$$

where

$C_{\ell i}$ = ℓ'th-group delayed neutron precursor concentration at node i

ℓ_i = neutron generation time at node i

$\beta_{\ell i}$ = ℓ'th-group delayed neutron precursor fraction at node i

λ_ℓ = ℓ'th-group delayed neutron precursor decay constant.

Next, the reactor core was divided radially into several regions (q = 1, 2, . . . , Q). Equations that describe the time variation of the average neutron source density $n_{q,k}$ and average delayed neutron precursor concentration $m_{\ell_{q,k}}$ within each region q were derived for each axial node (k = 1, 2, . . . , K).

This spatial collapsing was made by taking weighted summation of Eq. (2) over all nodes i within q and k. The equations derived are summarized here:

$$\left.\begin{aligned} \ell_{q,k}\dot{n}_{q,k} &= \sum_{q'} WH_{q',q,k}n_{q',k} + W^-_{q,k+1}n_{q,k+1} \\ &\quad + W^+_{q,k-1}n_{q,k-1} \\ &\quad + \left(W_{q,k} + WS_{q,k} - \frac{\lambda_0}{k_{\infty q,k}} - \beta_{q,k}\right)n_{q,k} \\ &\quad + \sum_\ell \lambda_\ell m_{\ell_{q,k}} \\ \dot{m}_{\ell_{q,k}} &= \beta_{\ell_{q,k}}n_{q,k} - \lambda_\ell m_{\ell_{q,k}} \end{aligned}\right\} , \qquad (3)$$

where $n_{q,k}$ is normalized to unity at initial steady state, i.e., $n_{q,k}(0) = 1.0$.

Coefficients $\ell_{q,k}$, $WH_{q',q,k}$, $W^-_{q,k}$, $W^+_{q,k}$, etc., are defined as shown below:

$$\left.\begin{aligned} \ell_{q,k} &= \sum_{i\epsilon q,k} S_i^*\ell_iS_i^0 \\ WH_{q',q,k} &= \sum_{i\epsilon q,k}\sum_{j\epsilon q',k} S_j^*W_{ij}S_i^0 \\ W^-_{q,k} &= \sum_{i\epsilon q,k} S^*_{i_{k-1}}W_{ii_{k-1}}S_i^0 \\ &\text{etc.,} \end{aligned}\right\} , \qquad (4)$$

where

S_i^0 = initial steady-state neutron source at node i

S_i^* = weight at node i.

The optimal weight S_i^* is shown to satisfy the following equation, adjoint to the steady-state equation for absorption rate $Si/k_{\infty i}$:

$$S_i^* = \frac{k_{\infty i}}{\lambda_0}\left(\sum_j W_{ij}S_j^* + WS_iS_i^*\right) . \qquad (5)$$

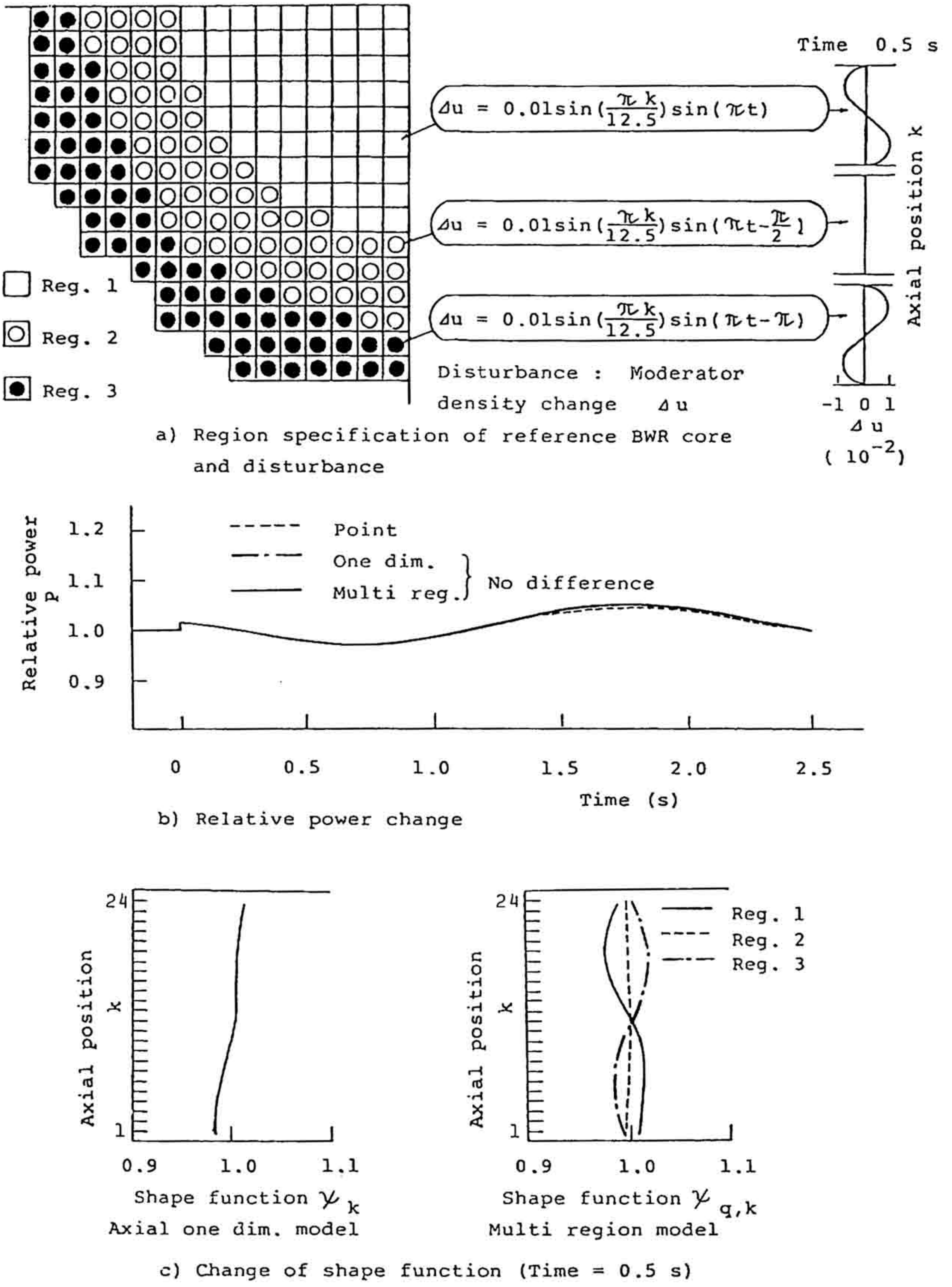
The physical meaning of S_i^* is the importance of a neutron absorbed at node i, namely, absorption importance.

Two approximations can be employed for slow transient phenomena (prompt subcritical): (a) quasi-static[2] and (b) prompt jump. Coefficients of Eq. (4) can be fitted as functions of average moderator density $u_{q,k}$ and average fuel temperature $T_{q,k}$ at each axial node k of each region q for each specific control rod configuration.

The above model was applied to a reference 1000-MW(e) BWR core. An example of transient behavior is shown in Fig. 1. The reactor core was assumed in steady state at 60% power and 40% flow before the transient, and divided radially into three circular regions (Q = 3) and axially into 24 nodes (K = 24).

First, steady-state neutron source and absorption importance were solved for original three-dimensional node i and used to calculate coefficients of Eq. (4). As a disturbance, only moderator density was changed intentionally from its steady-state distribution, as given by

$$\Delta u_{q,k} = 0.01 \sin\left(\frac{\pi k}{12.5}\right) \sin\left[\pi t - (q-1)\frac{\pi}{2}\right] . \qquad (6)$$



a) Region specification of reference BWR core and disturbance

b) Relative power change

c) Change of shape function (Time = 0.5 s)

Fig. 1. An example of transient behavior analysis.

In this figure, results obtained by an axial one-dimensional model, a special case of Q = 1, and a one point model, a special case of the former with constant power distribution with time, are also shown for comparison. It is seen that the reactor power can be well predicted by an axial one-dimensional model, but there exists a notable difference in power distribution for this type of out-of-phase regionwise disturbance. Results of other applications also indicate the necessity of multiregion treatment and practicality of this model.

1. D. L. DELP et al., GEAP-4598, General Electric Co. (1964).

2. K. O. OTT and D. A. MENELEY, *Nucl. Sci. Eng.*, **36**, 402 (1969).

6. A Discrete Sampling Method for Vectorized Monte Carlo Calculations, *Forrest B. Brown, William R. Martin, Donald A. Calahan (Univ of Michigan)*

The discrete sampling method outlined below is a particular extension of Marsaglia's method[1] as applied to discrete distributions. It is faster than the usual method for sampling discrete distributions with large table length N, executes in a fixed time independent of N, and can be efficiently implemented into Monte Carlo codes for parallel and vector processing computers.

The usual method for discrete sampling can be expressed as[2]

$$x = \sup_{F(x_i)<r} x_i \ , \qquad (1)$$

where $F(x_i)$ is the cumulative distribution function (cdf) for the discrete points x_i, $i = 1, 2, \ldots, N$, and r is a random number distributed uniformly on (0,1). This method requires searching a "ladder" of length N for the x_i corresponding to the largest $F(x_i)$ not exceeding r. It is trivial to code and is used almost universally, but suffers from two disadvantages: (a) Its speed is limited by the rate of data transfer from memory to central processing unit (CPU). This rate is generally much lower than CPU cycle time, and the search will be relatively slow for large N. (b) This method will not be efficient on vector processing computers. If it is desired to sample many values of x at once, Eq. (1) will be satisfied at many different ladder positions, and the ladder will in general have to be searched to the end. Many unneeded operations and data transfers are thus required.

In Marsaglia's method, a cdf F(x) is represented as a mixture of conditional distributions:

$$F(x) = \sum_k G(x|k)*h(k) \ , \qquad (2)$$

where h(k) is a discrete probability density function (pdf) for k, and G(x|k) is the conditional cdf for x given a particular value of k. In this method, a particular G(x|k') is selected by sampling k' from h(k), and then x is sampled from G(x|k'). The method is very efficient if G(x|k) is chosen to be an easily sampled approximation when h(k) is large and a more difficult to sample remainder only when h(k) is small. Both G(x|k) and h(k) are chosen *ad hoc* to suit a particular distribution and computer.

In the present method, a discrete cdf is also represented as a mixture of conditional cdfs as in Eq. (2). We choose G(x|k) and h(k) definitively, however, to meet the following conditions: (a) The distributions chosen must be sufficient to exactly represent any arbitrary discrete F(x). (b) Both must be easy to sample for all values of x and k. (c) When many samples are to be made at once, the sampling methods must allow a high degree of parallelism in computation without extra memory fetches or unneeded operations. All three demands are satisfied if h(k) is chosen as a discrete uniform density for $k = 1, 2, \ldots, N'$, where N' is the number of discrete x points having nonzero probability, and G(x|k) is chosen as a discrete binary cdf, i.e.,

$$h(k) = 1/N' \quad \text{for} \quad k = 1, 2, \ldots, N' \ , \qquad (3)$$

$$G(x|k) = \begin{cases} 0 & \text{for } x < x_{1,k} \\ q_k & \text{for } x_{1,k} < x < x_{2,k} \\ 1 & \text{for } x_{2,k} < x \ , \end{cases} \qquad (4)$$

where $\{q_k, x_{1,k}, x_{2,k}\}$, $k = 1, 2, \ldots, N'$ are determined below. The sampling procedure is then as follows: sample k uniformly, then select either $x_{1,k}$ with probability q_k or $x_{2,k}$ with probability $1 - q_k$.

The following algorithm derived from geometric arguments will always determine a sufficient set $\{q_k, x_{1,k}, x_{2,k}\}$ for any discrete F(x). Letting f_i denote $F(x_i) - F(x_{i-1})$ (i.e., f_i, $i = 1, 2, \ldots, N$ is the discrete pdf for the x_i's), and N' denote the number of nonzero f_i's, then the algorithm proceeds by "transferring" 1/N' of the total probability to each of the h(k)'s. This is done sequentially, picking for G(x|k) the pair of x_i's with the largest and smallest (nonzero) remaining probability, choosing a q_k to conserve the smaller probability, and then removing the smallest and a portion of the largest f_i at each stage to conserve probability. Thus,

```
for k = 1, 2, . . . , N':
    $x_{2,k} = x_j$, s.t. $f_j$ = smallest positive $f_m$, $m = 1, \ldots, N$
    $x_{1,k} = x_i$, s.t. $f_i$ = largest $f_m$, $m = 1, \ldots, N$
    $q_k = 1 - N'*f_j$
    $f_j = 0$
    $f_i = f_i - q_k/N'$ .                                        (5)
```

The discrete conditional sampling method described by Eqs. (2), (3), (4) is fast and well suited to parallel or vector computation. When many samples are to be obtained, the first step, selecting k, can be done in parallel without memory fetches. The second step, fetching the appropriate q_k from memory, requires only one memory fetch per sample, regardless of the number of discrete points in F(x). The third step, deciding between $x_{1,k}$ and $x_{2,k}$, can be performed in parallel, and the final step, retrieving the sample, requires only one memory fetch.

In Fig. 1, timing comparisons are presented for discrete sampling via the usual method and our method on the Amdahl 470V/8 computer. It is apparent from Fig. 1 that our method is significantly faster when more than about three ladder entries must be checked. Preliminary results obtained with a Cray-1 simulator[3] show similar gains for the Cray-1 computer, with an additional speedup by a factor of ~2 when vectorized. Application to the discrete portions of multigroup Monte Carlo calculations on the Amdahl 470V/8 using 100 neutron groups[4] yields modest speedups by factors of 3.2 and 1.5 for sampling the ^{235}U fission spectrum and the group-to-group transfer in water, respectively. Although discrete sampling comprises only a minor portion of large production codes[5] such as MORSE, it is expected that our method will be highly valuable for two classes of applications: (a) multigroup Monte Carlo codes for vector computers, where the parallelism of the method can be exploited, and (b) specialized calculations involving very large discrete distributions, since our method executes in a fixed time regardless of the size of the distribution. We are currently using the discrete conditional sampling method presented above for all discrete sampling in a multigroup Monte Carlo code under development for vector computers.