AN ANALYTICAL METHOD FOR THE SOLUTION OF REACTOR DYNAMIC EQUATIONS

F.I. Ibitoye
Centre for Energy Research and Development
Obafemi Awolowo University, Ile-Ife, Nigeria.
Email: fibitoye@yahoo.com
Tel:+234 (0) 803 717 6453

Abstract: Microcomputers are becoming increasingly popular in systems simulation because of their low cost and improved performance. One of the challenges of modelling nuclear reactor dynamics on microcomputers is that of finding robust techniques which guarantee the required level of accuracy and at the same time produce results in reasonable time. In this paper, an analytical method for the solution of nuclear reactor dynamic equations is presented. The method is applied to a linearised high-order deterministic model of a pressurised water reactor plant driven by step-reactivity insertion. A comparison of this method with two other techniques (the matrix exponential and finite difference approximation methods) shows that the analytical method yields the most accurate results on a microcomputer and it is also found to be numerically stable over all integration step-sizes investigated.

Keywords: linearisation, PWR, reactivity, deterministic model, reactor dynamics

1. INTRODUCTION

Modelling dynamical systems on microcomputers involves developing compact algorithms that guarantee some degree of accuracy. In real-time applications, such algorithms must also be fast enough to generate real-time solutions. The demand for low run-time and a reasonable degree of accuracy makes the usual numerical methods impracticable on microcomputers and hence new techniques must be sought. In this paper we present an efficient method for the solution of the dynamic equations of nuclear reactor systems. The method is compared with the matrix exponential method and finite difference approximation by means of a numerical example.

The governing equations of the dynamics of nuclear reactor systems can most often be cast in the general form:

$$\frac{d\mathbf{X}(t)}{dt} = f(\mathbf{F}, \mathbf{X}, t) \qquad (1)$$

where **F** is the vector of driving functions and **X** the state variable vector. The feedback mechanisms of nuclear reactor systems are non-linear in practice, but for some sufficiently small input disturbances it is possible to reduce the dynamic equations to a system of linear equations.

There are various reasons for preferring a linearised model:

(a) The perturbation may be genuinely small enough that the use of a non-linear representation makes little or no difference to model predictions

(b) The computational costs of solving linear model equations are lower, and the low cost makes them suitable for use on small computers, and development of real-time algorithms.

(c) The manipulations and numerical methods of linear system equations are well known and easy to track down for dynamic studies of the plant.

Whatever the reasons for linearisation may be, however, we still recognise the fact that the linearised model is an approximation to the corresponding non-linear version.

Linearisation proceeds by writing each state variable as a sum of its steady state value and the deviation from steady state conditions due to an input disturbance. For relatively small input disturbances products of such deviation of state variables from steady state conditions are small and can be neglected. Thus for small input disturbances the linearised form of eqn. (1) can be written as

$$\frac{d\delta\mathbf{X}(t)}{dt} = \mathbf{A}\delta\mathbf{X}(t) + \mathbf{F}(t) \qquad (2)$$

where **A** is the control matrix of constant coefficients and δ**X** the vector of deviations of state variables from their steady state values due to a small input disturbance **F**. Three numerical methods are described below for the solution of eqn. (2).

Nig. J. Pure Appl Phys. (2004) Vol. 2 pp 1-.... .Printed in Nigeria. ISSN 1596-0862. E-mail: njpapfuta@yahoo.com. For rights of reproduction and web access, contact African Journals Online. UK, www.inasp.ajol/journals/njpap

2.THE MATRIX EXPONENTIAL METHOD

The general solution of eqn. (2) is

$$\delta\mathbf{X}(t) = e^{\mathbf{A}\Delta t}\delta\mathbf{X}(t_0) + \int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{F}(\tau)d\tau\,,$$

$\Delta t = t\text{-}t_0$ (3)

The matrix exponential method [1] stems from the fact that the exponential term in eqn. (3) can be expanded in Taylor series as:

$$e^{\mathbf{A}\Delta t} = \mathbf{I} + \mathbf{A}\Delta t + \frac{(\mathbf{A}\Delta t)^2}{2!} + \frac{(\mathbf{A}\Delta t)^3}{3!} + \cdots \quad (4)$$

If the time interval Δt is appreciably small we can assume that the driving function $\mathbf{F}$ in eqn. (3) is piecewise constant, so that we can write

$$\delta\mathbf{X}(t) = e^{\mathbf{A}\Delta t}\delta\mathbf{X}(t_0) + \left(e^{\mathbf{A}\Delta t} - \mathbf{I}\right)\mathbf{A}^{-1}\mathbf{F}(t) \quad (5)$$

From equation 4 we obtain

$$[e^{\mathbf{A}\Delta t} - \mathbf{I}]\mathbf{A}^{-1} = \begin{pmatrix} \mathbf{I} + \dfrac{\mathbf{A}\Delta t}{2!} + \dfrac{(\mathbf{A}\Delta t)^2}{3!} \\ + \dfrac{(\mathbf{A}\Delta t)^3}{4!} + \cdots \end{pmatrix}\Delta t \quad (6)$$

thus removing the need to compute $\mathbf{A}^{-1}$. Eqn. (5) can therefore be evaluated by using values computed from the series in eqns. (4) and (6). The series expansions are fairly straightforward, since the two can be generated in the same program loop. The choice of integration time-step and the number of terms in the series expansion are very critical in the accuracy of the scheme, however. In this work, the series is terminated after n terms when $\left|\sum_{k=1}^{n} a^k_{\max}\right| \le \varepsilon$, where $a^k_{\max}$ is the maximum element in the k^{th} term of the matric series and ε the unit round-off of the machine, defined as the smallest positive number such that $(1+\varepsilon)\neq 1$. This scheme was found to work satisfactorily. We note, however, that the solution of equation 2 has been reduced to a simple problem of addition and multiplication of matrices.

3. FINITE DIFFERENCE APPROXIMATION

This approach was introduced by Zhiwei *et. al.* [2], and makes use of the posterior difference form of Euler's formulas. For if we write eq. 2 as

$$\delta\dot{\mathbf{X}} = \mathbf{A}\delta\mathbf{X} + \mathbf{F}(t)$$
$$\equiv f(\delta\mathrm{X}, t),$$
$$\delta\mathrm{X}(0) = \delta\mathbf{X}_0$$

then the equivalent difference equations using the posterior derivatives are:

$$\delta\mathbf{X}_{k+1} - \delta\mathbf{X}_k = \Delta t f(\delta\mathbf{X}_{k+1}, t_{k+1})$$
$$\delta\mathbf{X}_0 = \delta\mathbf{X}(0)$$

If after one recursion the incremental vector is ξ_{k+1}, such that

$$\delta\mathbf{X}_{k+1} = \delta\mathbf{X}_k + \xi_{k+1} \quad (7)$$

then

$$\xi_{k+1} = \Delta t f(\delta\mathbf{X}_{k+1}, t_{k+1}).$$

Expanding the right-hand side of the later equation we obtain

$$\xi_{k+1} = \Delta t f_{k+1}(\delta\mathbf{X}_k, t_{k+1}) + \Delta t\, \mathbf{A}\xi_{k+1}\,,$$

so that

(4) $$(\mathbf{I} - \mathbf{A}\Delta t)\xi_{k+1} = \Delta t f_{k+1}(\delta\mathbf{X}_k, t_{k+1})\,. \quad (8)$$

Therefore, using the initial conditions one obtains $f_{k+1}(\delta\mathbf{X}_k, t_{k+1})$, which can then be used to calculate the incremental vector from eqn (8). We then proceed to solve for the new state vector $\delta\mathbf{X}_{k+1}$ (5) from eqn. (7), and the process is repeated until the required transient time is covered.

4. ANALYTICAL METHOD

(6) One possible way of evaluating eqn. (3) is by similarity transformation of the coefficient matrix **A** such that:

$$\mathbf{A} = \mathbf{S}\,\Lambda\,\mathbf{T},\quad \mathbf{S}\,\mathbf{T} = \mathbf{T}\,\mathbf{S} = \mathbf{I} \quad (9)$$

where Λ is a diagonal matrix of the eigenvalues of **A**, **S** the matrix of eigenvectors of **A**, also called the transformation matrix, and **T** the inverse of **S**. **I** is an identity matrix. Thus, $e^{\mathbf{A}\Delta t} = \mathbf{S}e^{\Lambda\Delta t}\mathbf{T}$, and eqn. (3) becomes

$$\delta\mathbf{X}(t) = \mathbf{S}e^{\Lambda(t-t_0)}\mathbf{T}\,\delta\mathbf{X}(t_0) + \int_{t_0}^{t}\mathbf{S}e^{\Lambda(t-\tau)}\mathbf{T}\mathbf{F}(\tau)d\tau \quad (10)$$

or in components form,

$$\delta x_j(t) = \sum_{i=1}^{n}\sum_{k=1}^{n} S_{ji}T_{ik}\left\{e^{\lambda_i\Delta t}\delta x_k(t_0) + \int_{t_0}^{t} e^{\lambda_i(t-\tau)}F_k(\tau)d\tau \right. \quad (11)$$

Quite often, however, especially for dynamical systems, some of the eigenvalues of **A** are complex, and when this occurs, they appear as complex conjugate pairs. Therefore if we write

$$S_{ji} = X_{ji} + \hat{i}Y_{ji},$$
$$T_{ik} = W_{ik} + \hat{i}Z_{ik},$$
$$\lambda_i = \alpha_i + \hat{i}\beta_i,$$

and $e^{i\theta} = \cos\theta + i\sin\theta$ where $\hat{i}^2 = -1$

we obtain

$$\delta x_j(t) = 2\sum_{i=1}^{m}\sum_{k=1}^{n}\Big\{\left(X_{ji}W_{ik} - Y_{ji}Z_{ik}\right)\left[e^{\alpha_i\Delta t}\cos\beta_i\Delta t\,\delta x_k(t_0) + \int_{t_0}^{t} e^{\alpha_i(t-\tau)}\cos\beta_i(t-\tau)F_k(\tau)d\tau\right]$$
$$-\left(X_{ji}Z_{ik} + Y_{ji}W_{ik}\right)\left[e^{\alpha_i\Delta t}\sin\beta_i\Delta t\,\delta x_k(t_0) + \int_{t_0}^{t} e^{\alpha_i(t-\tau)}\sin\beta_i(t-\tau)F_k(\tau)d\tau\right]\Big\}$$

where m is the number of complex conjugate pairs. At this point we shall assume that the disturbance function is a general ramp function of the form $F(t) = a+bt$, which can be used to simulate control rod movements in the reactor core, or for step reactivity insertion when b=0. Using this the solution becomes

$$\delta x_j(t) = 2\sum_{i=1}^{m}\sum_{k=1}^{n}\Big\{\left(X_{ji}W_{ik} - Y_{ji}Z_{ik}\right)\left[e^{\alpha_i\Delta t}\cos\beta_i\Delta t\,\delta x_k(t_0) + F_k(t)\xi_1^i(t) - b_k\xi_2^i(t)\right]$$
$$-\left(X_{ji}Z_{ik} + Y_{ji}W_{ik}\right)\left[e^{\alpha_i\Delta t}\sin\beta_i\Delta t\,\delta x_k(t_0) + F_k(t)\xi_3^i(t) - b_k\xi_4^i(t)\right]\Big\} \qquad (12)$$

where

$$\xi_1^i = \frac{e^{\alpha_i\Delta t}}{\alpha_i^2+\beta_i^2}\left\{\alpha_i\cos\beta_i\Delta t + \beta_i\sin\beta_i\Delta t - \alpha_i e^{-\alpha_i\Delta t}\right\}$$

$$\xi_2^i = \frac{e^{\alpha_i\Delta t}}{\alpha_i^2+\beta_i^2}\left\{\Delta t(\alpha_i\cos\beta_i\Delta t + \beta_i\sin\beta_i\Delta t) - \frac{(\alpha_i^2-\beta_i^2)\cos\beta_i\Delta t + 2\alpha_i\beta_i\sin\beta_i\Delta t}{\alpha_i^2+\beta_i^2} + \frac{\alpha_i^2-\beta_i^2}{\alpha_i^2+\beta_i^2}e^{-\alpha_i\Delta t}\right\}$$

$$\xi_3^i = \frac{e^{\alpha_i\Delta t}}{\alpha_i^2+\beta_i^2}\left\{\alpha_i\sin\beta_i\Delta t - \beta_i\cos\beta_i\Delta t + \beta_i e^{-\alpha_i\Delta t}\right\}$$

$$\xi_4^i = \frac{e^{\alpha_i\Delta t}}{\alpha_i^2+\beta_i^2}\left\{\Delta t(\alpha_i\sin\beta_i\Delta t - \beta_i\cos\beta_i\Delta t) - \frac{(\alpha_i^2-\beta_i^2)\sin\beta_i\Delta t - 2\alpha_i\beta_i\cos\beta_i\Delta t}{\alpha_i^2+\beta_i^2} - \frac{2\alpha_i\beta_i}{\alpha_i^2+\beta_i^2}e^{-\alpha_i\Delta t}\right.$$

For real roots,

$$\delta x_j(t) = \sum_{i=1}^{n}\sum_{k=1}^{n}X_{ji}W_{ik}\left\{\delta x_k(t_0)e^{\alpha_i\Delta t} + F_k(t)\left[e^{\alpha_i\Delta t}-1\right]/\alpha_i \quad b_k\left[(\alpha_i\Delta t-1)e^{\alpha_i\Delta t}+1\right]/\alpha_i^2\right\} \qquad (13)$$

5. COMPARISON OF NUMERICAL TECHNIQUES

We shall apply the three solution techniques discussed above to a linearised deterministic model of a 2200 MW(th) pressurised water reactor (PWR) plant. The main components represented in the model are the reactor core, pressuriser, steam generator, pipings and plenums. Detailed formalism of the model equations has been reported by Kerlin *et. al* [3], and consists of a system of 21 linear ordinary differential equations made up of seven neutronics equations (point kinetics with six groups of delayed neutron), three core heat transfer equations (average values for each of fuel temperature and two coolant nodes), two equations for pressuriser (pressure and a control variable), three equations describing the steam generator (steam pressure, temperature and metal temperature), four equations for the inlet and outlet plenums of the steam generator and reactor core, and two other equations for the cup-mixed hot- and cold-leg coolant temperatures. The neutronics and thermalhydraulic equations are coupled by a reactivity feedback equation. The assumption is that the reactivity feedback can be expressed as a linear combination of the effects of temperature feedbacks from fuel and coolant, primary system pressure feedback, and changes in control rod position.

In order to establish some reference or baseline with which the three methods discussed above can be compared, model equations were first solved using a linear solver for ordinary differential equations, LSODE [4]. LSODE is an initial-value ordinary differential equation solver based on Gear's algorithm, which is essentially a variable-step, variable-order method with automatic error control algorithm [5]. This algorithm is know to be well suited to ill-conditioned problems such as those encountered in nuclear reactor dynamics, where stiffness ratios of the order of 10^6 are not unusual. However, for real-time calculations or problems in which model equations in the form of equation 2 are to be solved repeatedly several times (such as in optimisation problems or simulation packages), the overhead computational cost of using LSODE becomes overbearing on a microcomputer system, and hence the search for more elegant methods, the basis of this paper.

Transient response of the model for the first 100 seconds following the insertion of a \$0.071 step reactivity is shown in Figures 1-3. In Figure 1, the initial response of the system to the input disturbance is a sharp rise in reactor power, occasioned by the release of prompt neutrons. As expected, the effects of the step reactivity insertion on fuel temperature in Figure 2 and on hot-leg temperature in Figure 3 are much slower in time response than for reactor power. The initial tendency is for the state variables to change in response to the input disturbance, but after some time the system stabilises to a new equilibrium state because of the effect of the feedback mechanisms. It becomes imperative therefore that the prediction of the fastest transient in the system (i.e. the power response) will be of utmost importance in any attempt to solve the model equations.

The profile of the reactor power predicted over a transient period of 100 seconds by the three techniques discussed above are compared with that obtained using LSODE. The measure of comparison is the residual error, defined as the sum of squares of the deviation of experimental values from the LSODE solution, that is,

$$R = \sum_{i=1}^{N} \left(x_i^{LSODE} - x_i^{EXPT} \right)^2, \qquad (14)$$

where N is the number of sample points.

6. RESULTS AND DISCUSSION

The residual errors generated by the three methods are plotted as functions of the integration time step in Figure 4. In this figure we observe that the matrix exponential method becomes numerically unstable beyond a step-size greater than 0.05 seconds. Although the method of finite difference approximation remains stable over the step-sizes considered, the residual errors due to this method are of the order of 10^{10}- 10^{13} times those of the analytical method. Also in Figure 4 we notice that out of the three techniques considered, the analytical method has the least residual errors. Furthermore, we do not observe any significant influence of the choice of integration time-step on the residual errors generated by the analytical method.

Figure 5 shows the dependence of program run-time on the choice of step-sizes. Within the range of time-steps where numerical stability is achievable, we notice that the program run-time for the matrix exponential method increases with increasing step-size; this is because of the need to expand the exponential series to higher order terms. On the other hand, for both the analytical technique and finite difference approximation method, the run-times fall gradually with increasing integration time-step. Although the analytical technique generates the least residual errors, it is about 40-50 times slower than the finite difference approximation method, and between 3-20 times slower than the matrix exponential method, when solution is feasible in the latter. At this level, the analytical method has only two advantages over the other methods considered - it is numerically stable, and generates reliable results. It is however possible to optimise the algorithm if we are primarily interested in step reactivity insertions. In this case the input function is a constant and some iterations and other calculations can be avoided. By using the initial condition $\delta X(t_0) = 0$ equations 12 and 13 respectively simplify to

$$\delta x_j(t) = 2\sum_{i=1}^{m}\sum_{k=1}^{n}\left\{\left(X_{ji}W_{ik} - Y_{ji}Z_{ik}\right)F_k(t)\xi_1^i(t) - \left(X_{ji}Z_{ik} + Y_{ji}W_{ik}\right)F_k(t)\xi_3^i(t)\right\} \qquad (15)$$

for complex roots, and

$$\delta x_j(t) = \sum_{i=1}^{n}\sum_{k=1}^{n} X_{ji}W_{ik}F_k(t)\left[e^{\alpha_i \Delta t} - 1\right]/\alpha_i \qquad (16)$$

for real roots.

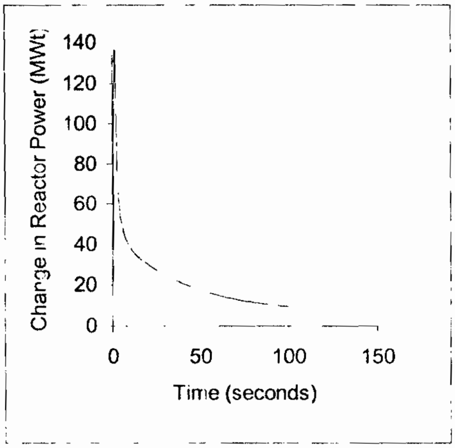

Figure 1: Transient Response of Reactor Power due to 0.071$ Step Reactivity Insertion

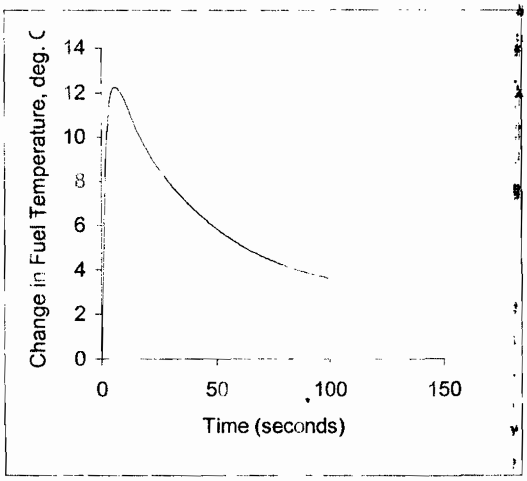

Figure 2: Fuel Temperature Response due to 0.071$ Step Reactivity Insertion

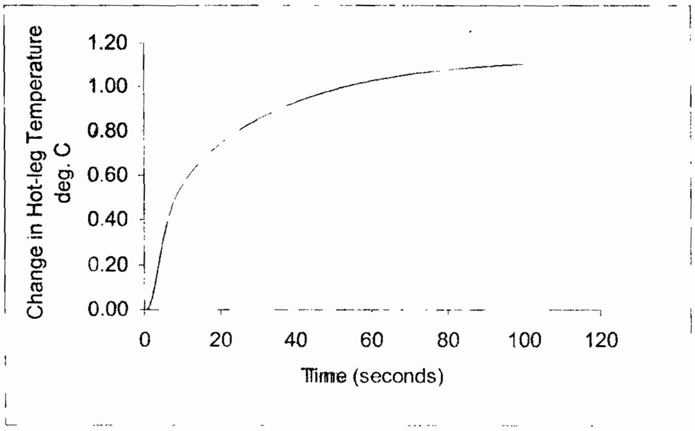

Figure 3: Hot-leg Temperature Response due to 0.071$ Step Reactivity Insertion

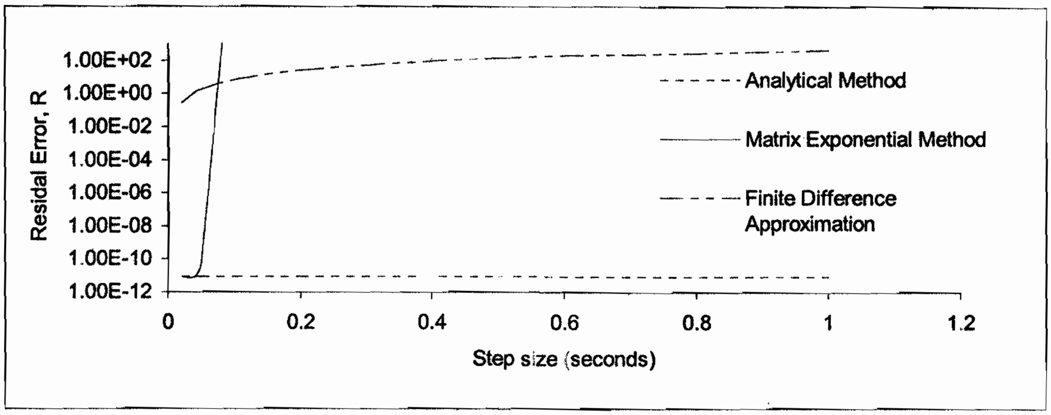

Figure 4: Comparison of Residual Errors Generated by Various Solution Techniques

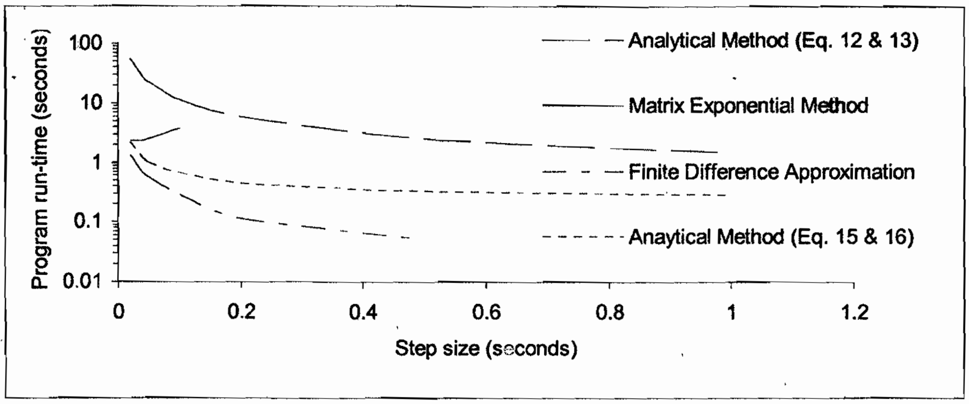

Figure 5: Dependence of Program Run-time on Step-size

These equations generate the same results as equation 12 and 13 for step reactivity insertions, with no loss of accuracy. Program run-times due to the reduced equations are shown in Figure 5, indicating that program run-time can be drastically reduced for step input disturbances, by a factor of 5-25, depending on step-size. Thus, the analytical method enjoys the benefit of a low program run-time, numerical stability, and accurate solution of the dynamic equations.

7. CONCLUSIONS

A numerically stable algorithm for the solution of nuclear reactor dynamic equations has been developed. It has been demonstrated through the example of a high-order model of a PWR[1] power plant that the method produces very accurate results on a microcomputer. The stability of the method stems from the fact that for a dynamically stable system the real parts of all the eigenvalues will be negative, and this removes the problem of numerical overflows when computing the exponential terms in eqns. (12) and (13). Although the formulation of the analytical method in this paper is for ramp input disturbances, the method is easily adaptable for inputs in the form of square waves, pulses and periodic functions. In addition, the methodology in this paper is equally applicable to other configurations of nuclear reactors, as long as the model equations can be cast in form of eqn. (2).

REFERENCES

[1] Ball, S.J. and R.K. Adams 1967. *MATEXP, A general purpose digital computer program for solving ordinary differential equations by the Matrix Exponential Method.* ORNL-TM-1933, Oak Ridge National Laboratory, USA.

[2] Zhiwei, L. and T.W. Kerlin 1983. *A numerical integration approach suitable for simulating PWR dynamics using a microcomputer system.*

[1] Computer source code for the analysis reported in this paper is freely obtainable from the author.

Proc. 5th Symp. Power Plant Dynamics, Control and Testing, Knoxville, Tennessee, USA.

[3] Kerlin, T.W., E.M Katz, J.G. Thakkar and J.E. Strange. 1976. *Theoretical and Experimental Dynamic Analysis of the H.B. Robinson Nuclear Plant*, Nucl. Technol. **30**: 299-316.

[4] Hindmarsh, A. C. 1980. *LSODE and LSODI, Two initial-value ordinary differential equation solvers*, ACM-SIGNUM Newsletter **15**(4): 10-11.

[5] Gear, C.W. 1971. *The automatic integration of ordinary differential equations*. ACM **14**: pp. 176.