INFORMATION TO USERS

This reproduction was made from a copy of a manuscript sent to us for publication and microfilming. While the most advanced technology has been used to photograph and reproduce this manuscript, the quality of the reproduction is heavily dependent upon the quality of the material submitted. Pages in any manuscript may have indistinct print. In all cases the best available copy has been filmed.

The following explanation of techniques is provided to help clarify notations which may appear on this reproduction.

1. Manuscripts may not always be complete. When it is not possible to obtain missing pages, a note appears to indicate this.

2. When copyrighted materials are removed from the manuscript, a note appears to indicate this.

3. Oversize materials (maps, drawings, and charts) are photographed by sectioning the original, beginning at the upper left hand corner and continuing from left to right in equal sections with small overlaps. Each oversize page is also filmed as one exposure and is available, for an additional charge, as a standard 35mm slide or in black and white paper format.*

4. Most photographs reproduce acceptably on positive microfilm or microfiche but lack clarity on xerographic copies made from the microfilm. For an additional charge, all photographs are available in black and white standard 35mm slide format.*

***For more information about black and white slides or enlarged paper reproductions, please contact the Dissertations Customer Services Department.**

U·M·I Dissertation Information Service

University Microfilms International
A Bell & Howell Information Company
300 N. Zeeb Road, Ann Arbor, Michigan 48106

Tsuruoka, Syuji

DYNAMICS AND FEEDBACK CONTROL OF CRYSTAL SIZE DISTRIBUTION IN A CONTINUOUS CRYSTALLIZER

The University of Arizona PH.D. 1986

DYNAMICS AND FEEDBACK CONTROL OF CRYSTAL SIZE DISTRIBUTION IN A CONTINUOUS CRYSTALLIZER

by

Syuji Tsuruoka

A Dissertation Submitted to the Faculty of the

DEPARTMENT OF CHEMICAL ENGINEERING

In Partial Fulfillment of the requirements
For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

1 9 8 6

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Final Examination Committee, we certify that we have read the dissertation prepared by Syuji Tsuruoka

entitled DYNAMICS AND FEEDBACK CONTROL OF CRYSTAL SIZE DISTRIBUTION IN A CONTINUOUS CRYSTALLIZER

and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Ph.D. Chemical Engineering.

Alan D. Randolph	Alan D. Randolph	6/17/86 Date
Thomas W. Peterson	Thomas Peterson	6·17·86 Date
Farhang Shadman	F. Shadman	6/17/86 Date
Ralph Martinez	Ralph Martinez	6/17/86 Date
Ian L. Pepper	Ian L Pepper	6/17/86 Date

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copy of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

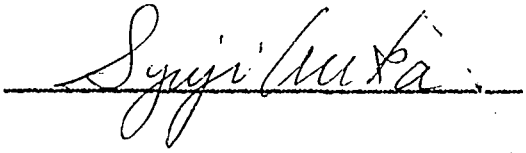
Alan D. Randolph
Dissertation Director

6/17/86
Date

STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirments for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: 

ACKNOWLEDGMENTS

I express my gratitude to Dr. Alan D. Randolph, my advisor, for his guidance in the successful completion of this work.

I am extremely grateful for research assistantship granted by NSF (NSF GRANT CPE-8117753).

The help of Dr. Francois Cellier is greatly appreciated. With his advice, advanced techniques for numerical simulation and designing of a system controller were successfully used for this work.

Finally, I wish to thank Yoko, my wife.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (continued)

LIST OF ILLUSTRATIONS (continued)

LIST OF ILLUSTRATIONS (continued)

LIST OF TABLES

ABSTRACT

A simulation algorithm for crystal size distribution dynamics in a continuous crystallizer was developed using the method of lines. Dimensionless crystal sizes, in vector form, were used as state variables. Simulation results using this algorithm satisfied stability criteria for continuous crystallizers, which had been developed previously using different methods. The use of a state space representation of the algorithm permits the use of well-known theoretical and numerical approaches to the modeling of an experimental R-z crystallizer and for design of a proportional controller for a continuous crystallizer system. Boundary conditions defined by the nucleation/growth rate kinetics were separately written as an auxiliary function so that other kinetics can be substituted without any change of the main algorithm. This implies that the algorithm is applicable for any growth-type particulate system. CSD dynamics from an experimental crystallizer were satisfactorily modelled using this algorithm with reasonable parameters: e.g. the recycle ratios of the fines dissolver and the product classifier, crystal sizes at the upper cut size of the dissolver and at the lower cut size of the classifier, initial CSD, and the form of the upset. Algorithms for controller design using pole placement and optimization techniques were applied to develop a proportional matrix controller for an R-z crystallizer. It was

evident that pole placement is a better method than optimization to design a controller for this crystallizer system. The system poles are concentrated at a point and it is necessary to assign the controller poles further apart to obtain appropriate control. To summarize controller design using the pole placement method, a schematic flow diagram of a system controller using the minimum order Luenberger observer was illustrated. In this example, only a few population densities need to be measured to drive the controller.

CHAPTER 1

INTRODUCTION

The study of control algorithms for crystal size distribution in crystallizers of complex configuration is a topic which might give control of crystallizer systems yielding a more constant quality product. For large scale continuous crystallizers, crystal size distribution is regarded as one of the most important properties, in addition to production rate. Crystal size distribution characterizes a crystal product. For instance, the dissolution rate of crystals is dependent on the crystal size. In the field of pharmacy, crystal size distribution often determines the bioavailability of drugs in the human body. Thus, controlling crystal size distribution is an important objective in many crystallization processes. As another example, a quality photo is made by spreading silver halide crystals having a uniform crystal size distribution on developing paper. In any case, crystal size distribution to a greater or lesser extent determines function and quality of crystalline products. Thus, devising control algorithms for stabilizing crystal size distribution is an industrially desirable technology.

Feedback control algorithms have not been established in either industry nor laboratory for control of crystal size

distribution in continuous crystallizers. Furthermore, a definitive simulation algorithm for dynamic crystal size distribution behavior has not been constructed, although many adequate, but limited, crystal size distribution simulation algorithms do exist for prediction of dynamics. A major difficulty is that the population balance equation phrasing a continuous crystallizer system is a hyperbolic wave equation of heterogeneous first order form, as explained in chapter 3 and Appendix A. This type of equation is not only mathematically asymmetric but also numerically unstable relative to non-hyperbolic types. The hyperbolic equation can generate a numerical shock wave which leads to discontinuity and ruins the numerical analysis. Additionally, since a crystallizer has a growth rate constraint, the equations of crystal size distribution involve complex conservation statements that must be phrased mathematically. The growth rate constraint internally determines supersaturation and hence nucleation rate. These kinetics then feed back to determine the level of solute concentration and the number of the particles in the crystallizer system. Thus, the boundary conditions raise another mathematical difficulty in solving dynamic crystal size distribution equations. A method for solving the population balance equation with the proper boundary conditions using a state space formulation has not yet been demonstrated.

Methods to treat such equations, characterized as distributed parameter systems, were mainly developed in the 1970's in the field of aerospace science. Control algorithms for these

systems, namely a controller design method for multiple-input multiple-output systems expressed by vector-matrix formulae, were developed during this decade.

In this study, mathematical techniques which were newly developed in system theories during this decade are applied for establishing simulation and design algorithms for dynamic crystal size distribution. A simulation algorithm using the method of lines is discussed with mathematical consistancy and illustrated using model dynamics from a continuous classified product crystallizer with fines removal. Some dynamic simulations with or without the corresponding experimental results, are presented using this new algorithm. It is also shown that this simulation predicts results agreeing with a previous classical stability criterion for continuous Mixed Suspension, Mixed Product Removal (MSMPR) crystallizers. Controller design for a continous classified product crystallizer with fines removal is discussed using pole placement and optimization techniques. Finally, an on-line system control diagram with an observer is discussed.

CHAPTER 2

PREVIOUS STUDIES

In the field of crystallizer study, no one has yet been able to completely solve the dynamic crystal size distribution (CSD) equations in conventional variable format using a state space formulation. Previous researchers used parameter-specific models (the R-z crystallizer studied by Beckman, [1977]) or numerically inefficient models (the method of Heun as applied by Nuttall [1971]). In no case has the crystal size distribution control problem been solved directly without simulating the entire distributed dynamic crystal size distribution equations. Previous crystal size distribution control studies simulated particular crystallizer systems and designed controllers for the specific system. Many such control systems required crystallizer information, or control manipulations, that are not realistic or cannot be done with current measurement technology, e.g., Timm and Larson [1968], and Epstain and Sowul [1980]. However, it is interesting to look at these proposed crystal size distribution control schemes and/or crystal size distribution dynamic simulations. No study has efficiently characterized the dynamics of crystal size distribution in a comlpletely model-independent technique.

Sherwin, Shinnar and Katz [1969] investigated an idealized classified product crystallizer. Their particular model of the crystallizer involved two simplifying assumptions. One was that the crystallizer is completely mixed and another is that the classifier is ideal and instantly removes all crystals reaching the classification size, L_p. Thus, no particles are removed up to size L_p with instantaneous removal at size L_p. Using this highly idealized model, Sherwin, et al [1969] concluded that a classified crystallizer may exhibit "self-induced" instability leading to cyclic behavior. The critical value of the nucleation sensitivity parameter (defined as the log derivative of nucleation to growth rate) was approximately two. Note that in their study there exists a product classifier but no fines dissolver. Their results, using a rectangular-shaped crystal size distribution, are plausible and identify product classification as a major cause of crystal size distribution instability. However, no experimental results were produced to support their results. As this study concentrated on crystal size distribution dynamics and stability they did not examine positions of the poles of a characteristic matrix or controllability of the system from the stability analysis (A detailed discussion of controllability and observability is found in Appendix B). Mathematically, their approach did not follow modern state space formulation, but their study is important because it was one of the first to introduce the dynamic population balance with nucleation kinetics.

Nyvlt and Mullin [1970] presented a mathematical model of a continuous stirred crystallizer, which was based on the level of supersaturation. They mathematically demonstrated periodic changes of supersaturation, solid phase content, crystal size, and production rate. Their pivotal point of mathematical development is that the material balance in a crystallizer is expressed by supersaturation, $\Delta c = (c - c_{sat})$; thus, a mass balance gives:

$$\frac{d\Delta c}{dt} = s - k_g A_s (\Delta c)^n - k_n (\Delta c)^m \qquad (2-1)$$

where s, k_g, k_n, A_s, m, and n are supersaturation, a crystal growth rate constant, a nucleation rate constant, the surface area of crystals, the nucleation order, and a crystal growth rate constant, respectively. The conclusion was that the dynamics of the crystallizer system can be expressed by the total amount of solute using supersaturation as an external parameter; hence, crystal size distribution was not considered in the dynamic solution. This model can hardly be correct because there is a strong interaction between the crystal size distribution and the level of supersaturation s. This tactic for approaching the solution was good in a mathematical sense because Equation (2-1) is an ordinary differential equation. Nevertheless, this approach does not attempt an analysis of crystal size distribution dynamics. Furthermore, this method is not always experimentally realistic as it is almost impossible to measure supersaturation in a high yield (Class II) crystal system. On the

other hand, the cycle frequency of supersaturation is one of the key parameters generating crystal size distribution oscillations. The frequency used in their model is not mathematically unique, which means that one can freely choose the oscillation rate for a crystallizer system. The frequency may be regarded as an adjustable parameter. Their model did not explain a priori the dynamics of a crystallizer. The key to any useful a priori crystallizer dynamics model must certainly be a dynamic crystal size distribution model.

Randolph and Larson [1962] carried out dynamic studies of continuous crystallizers using moment equations. The dynamics of a crystallizer was expressed by the total amount of crystal population, crystal surface area, and mass of crystals. Supersaturation was not used as an internal parameter (a high yield, Class II system was assumed but supersaturation could easily be added to the set of state equations). Hence, the mathematical expressions are written with real state variables only. This method is convenient for analysis of the total changes of the dependent variables. Using this technique, they showed that an MSMPR crystallizer is very stable, which agrees with observation. Only operation near an essentially discontinuous change in nucleation rate can cause crystal size distribution cycling. Fines removal improves transient population responses to disturbances while somewhat narrowing the stable operating range relative to the MSMPR case. Nuttall [1971] numerically solved the distributed dynamic crystal size distribution equations using a direct finite difference scheme (Heun's method) but

the dynamic crystal size distribution computations consumed large amounts of computer time.

Randolph, Beer, and Keener [1973] used a spectral density approach to solve for the stability boundaries of the so-called R-z crystallizer (fines dissolving and classified product). This is a process-realistic model which assumes the crystallizer removes crystals at R and z times the mixed removal rate for fines and oversize. The main point in their study is the establishment of stability criteria of the Class II, high yield classified product crystallizer with fines removal. However, their model was not a crystal size distribution simulator and only gives the boundaries of stability, not the dynamic crystal size distribution. Thus, it is of limited value for crystal size distribution studies. Stability regions are described as critical values of the kinetic nucleation/growth rate exponent i, where

$$i = d (\log B^{o}) / d (\log G)$$

versus dimensionless parameters of the crystallizer systems. Randolph and Larson [1971] showed that fines removal as small crystals lowers the nucleation/growth rate sensitivity ratio (the critical value of i where cycling would begin). Classified product removal significantly destabilizes crystal size distribution, also lowering the critical value of i. The importance of the study is that stability limits of an R-z crystallizer system in dimensionless form are determined using the dimensionless parameters: (1) the

crystal size at the upper cut size of a fines dissolver and at the lower cut size of a product classifier, and (2) the recycle ratio of the fines dissolver to the product classifier. These results are not derived from state variable forms using conventional system theories, but they are useful in identifying regions of crystallizer instability where crystal size distribution control might be difficult. Unfortunately no general computer algorithm to evaluate critical stability parameters for a given crystallizer system is given in their paper.

Beckman and Randolph [1977] established a simulator for dynamic crystal size distribution using double Laplace transforms, and Sibert [1972] used the method of characteristics to develop a dynamic crystal size distribution simulator. These simulation programs exist in Department of Chemical Engineering at the University of Arizona, and are named "CYCLER" (using the double Laplace transform) and "MARK V" (using the method of characteristics), respectively. However, neither of these dynamic crystal size distribution simulators is of any value in developing modern state space crystal size distribution control algorithms [Broaman [1970], Carrier [1976], Schetzen [1980], and Rugh[1981]] even though they are useful in demonstrating control strategies by direct simulation.

Several mathematical studies related to characteristics of the partial differential equation (which expresses the dynamic population balance of an R-z crystallizer system) were discussed and submitted in the field of system theories. Leden, Hamza, and

Sheirah [1974] presented different methods for determining parametric models of a one dimensional heat diffusion process which has distributed parameters. They compared a straightforward application of frequency response techniques (which has gained widespread acceptance as an accurate method for determininig thermal diffusivity), with an application of an on-line least squares technique (which has the advantages of being applicable to arbitrary distributed parameter systems and of requiring a small amount of compensation and storage), and with an application of the maximum likelihood method (which is shown to be the most accurate method). Among those methods, our concern is with the last one. With this method the diffusion process is transformed into a finite dimensional system of ordinary differential equations by approximating the partial derivative in the space (z-direction) by finite difference formulae. This method is presently called the method of lines. This method is recommended for accurate estimation of thermal diffusivity, usually producing errors less than 0.01%.

Sheirah, and Hamza [1974] also presented a method based on mathematical programming for optimal control of distributed parameter systems. According to their results, the problem is reduced to a quadratic programming problem for linear systems. The method is restricted to systems which can be represented by discrete time models having coefficients which are functions of the spatial variables. It is also assumed that the control function is spatially concentrated. If the control is free, an unconstrained quadratic programming problem results, reducing the optimal control

sequence over a planning horizon. Their results also depend on the transient period of the system and, for time-varying systems, on the time required to perform the identification.

A summary of reports related to the method of lines, including higher order partial differential equations, is given by Carver, et al [1976], presented in a workshop on computer aided techniques for solving differential equations. A summary of the workshop is published as a text, including instructions for a package program called FORSIM. The text consists of outlines of ordinary differential equations and numerical integration methods in matrix form, introduction of FORSIM, solutions of partial differential equations using the methods of lines, and treatment of large simulations. Among these topics, a pivotal discussion focuses on the characteristics of the method of lines and the use of this method to solve non-linear partial differential equations.

It is easy to find terminology on second order partial differential equations but not as easy to find terminology on first order ones. The reason is that second order partial differential equations are relatively easier to treat than equations of higher order. Second order equations can often have symmetrical expressions. The methods for classifying higher and first order partial differential equations are deeply based in the theory of characteristics. According to Carver, et al. [1976], the classification of first order partial differential equations is as follows. A first order partial differential equation is written as;

$$A \frac{\partial u}{\partial x} + B \frac{\partial u}{\partial y} = d \qquad (2\text{-}2)$$

where $A = [\, a_{ij} \,]$ and $B = [\, b_{ij} \,]$ are n x n matrices, and u and d are the column vectors $\{ u_1, \ldots, u_n \}'$ and $\{ d_1, \ldots, d_n \}'$, where " ' " (used hereafter) is the transpose of the vector. If the equation;

$$\det \left[A - \lambda B \right] = 0 \qquad (2\text{-}3)$$

has no real roots , the system is elliptic. If it has n distinct real roots $_n$, the system is totally hyperbolic. If it has an multiple root , the system is totally parabolic. The system simulation which is characterized as an elliptic or a parabolic one is not numerically difficult, but if the system is characterized totally or partially as a hyperbolic one, it is difficult to simulate numerically because the system generates steep fronted shock waves. A treatment of a simple hyperbolic first order partial differential equation;

$$\frac{\partial u}{\partial t} = - c \frac{\partial u}{\partial x} \qquad (2\text{-}4)$$

is explained Chapter 8 of the FORSIM text. The method of lines is discussed in detail in Appendix A.

CHAPTER 3

THEORY

A continuous crytallizer of complex configulation is schematically illustrated in Figure 3-1, which is a Class II (high yield) classified product crystallizer with fines removal, refered to as an R-z crystallizer. [Note: the most common crystallizers found in industry have the simple MSMPR, not the R-z configulation.] A typical crystal size distribution of the crystallizer is presented as population density in Figure 3-2. The steady state crystal size distribution equation for the system is expressed in population density as;

$$n = f (L_F, L_P, R, z, x) \tag{3-1}$$

or

$$\frac{n}{\bar{n}^o} = H(x) \exp(-Rx) + \{ H(x-x_F) - H(x-x_P) \}$$

$$X \{ \exp[-(R - 1)x_F] \exp(-x) \} + H(x - x_P) \tag{3-2}$$

$$X \{ \exp[(z - 1)x_P - (R - 1)x_F] \exp(-zx)\}$$

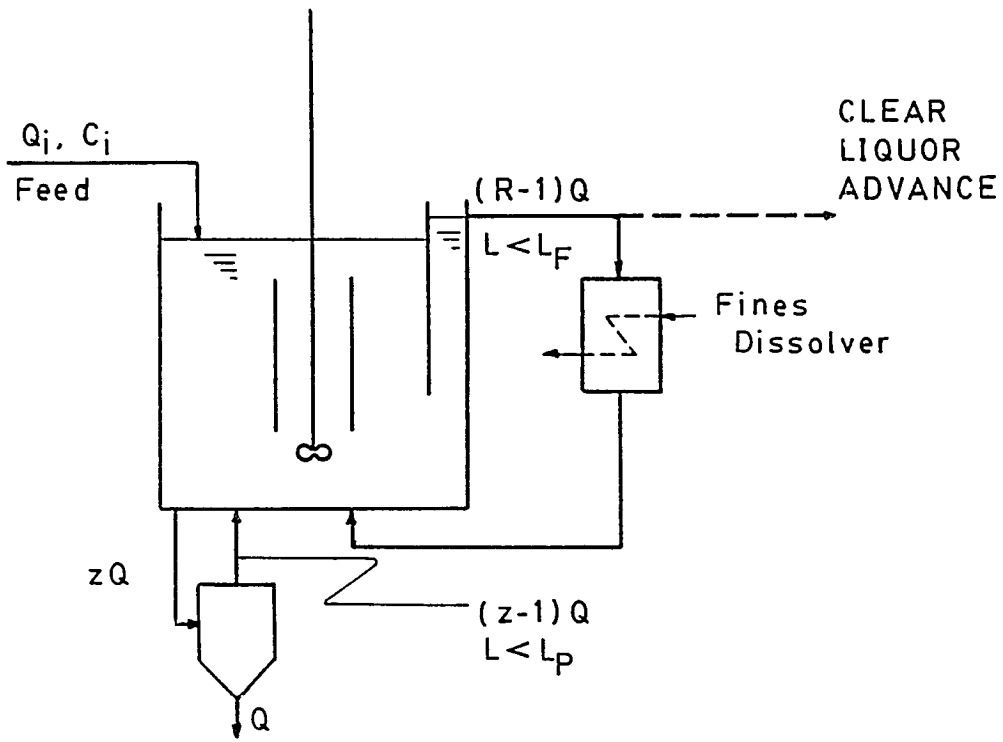


Figure 3-1 Idealized mode of R-z fines dissolver/classified product crystallizer.

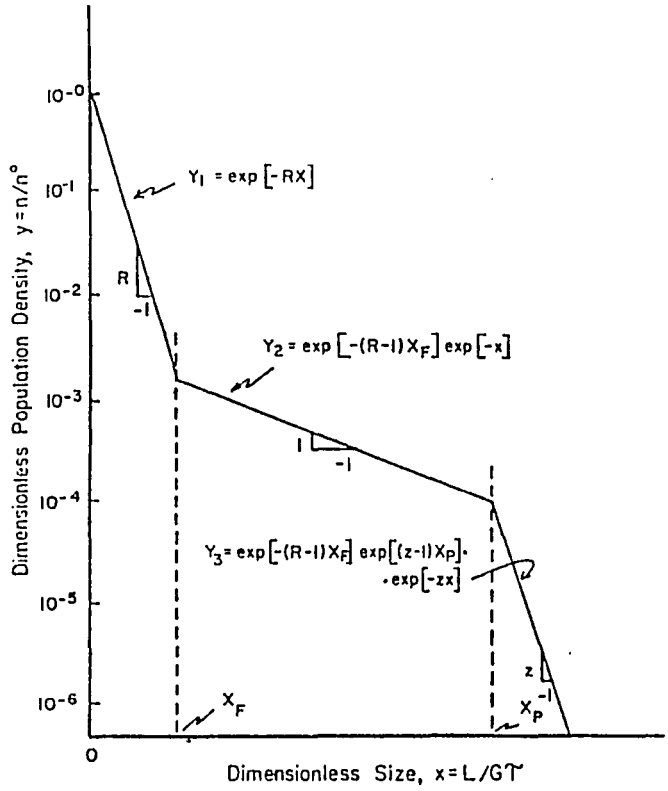


Figure 3-2 Steady state population distribution for the idealized crystallizer.

where H(x) is a Heaviside step function. [See Randolph, Beer, and Keener [1973] for derivation of this distribution.] x is defined as a dimensionless particle size:

$$x = \frac{LQ_P}{VG} \tag{3-3a}$$

Hence, x_F and x_P are defined as;

$$x_F = \frac{L_F}{\bar{L}} \tag{3-3b}$$

$$x_P = \frac{L_P}{\bar{L}} \tag{3-3c}$$

where $\bar{L}=GV/Q_P=G\tau$, a characteristic particle size for the system.

Randolph, Beer, and Keener [1973] show that the growth rate is constrainted for a Class II system. The value predicted for the R-z crystallizer in terms of the external production rate is;

$$G = \left(\frac{P_E}{6 Q_P D_P}\right)^{\frac{1-j}{i+3}} \left(6 D_P\right)^{\frac{-j}{i+3}} \left(k_N \rho k_v \tau^4\right)^{\frac{-1}{i+3}} \tag{3-4}$$

where

$$D_P = \frac{w(Rx_F)}{R^4} + \exp[\ -(R-1)x_F\ ][\ w(x_P) - w(x_F)\ ] + \frac{1}{z^3}\exp[\ (z-1)x_P - (R-1)x_F\ ][\ 1 - w(x_F)\ ] \tag{3-5}$$

and

$$w(x) = \frac{1}{6}\int_0^x x\ \exp(-x)dx \tag{3-6}$$

The dynamic population balance is described as;

$$\frac{\partial n}{\partial t} + G\ \frac{\partial n}{\partial L} + \frac{Q(L)n}{V} = 0 \tag{3-7}$$

where population density in the feed liquor is regarded as zero, i.e., clear liquor is assumed. When no clear liquor is assumed, the external input in population density is a function of time, $\underline{r}(t)$, and is substituted in the right-hand side of Equation (3-7).

The mass balance constraint of the system illustrated in Figure 3-1 and shown Equation (3-4) gives a growth rate equation (Nuttall, [1971]), and Randolph, Beer, and Keener [1973])as;

$$G = \frac{P_E + (R - 1) Q_P \rho k_v \int_0^{L_F} n L^3 dL}{\left(\frac{\rho V}{2}\right) A} \quad (3\text{-}8)$$

A population density boundary condition is given by the nucleation kinetics equation. Thus, as conventionally written for mixed magma crystallizers (Randolph and Larson, [1971]) nucleation kinetics for a Class II system are given as;

$$n^o = \frac{B^o}{G} \quad (3\text{-}9)$$

with

$$B^o = k_N M_t^j G^i \quad (3\text{-}10)$$

which describes secondary nucleation rate in terms of slurry density and growth rate (the latter as a surrogate variable for the unmeasurable supersaturation level).

In the above model, the classical size-independent growth rate hypothesis is used. Recently, several researchers have studied growth rate dispersion (Randolph and White, [1977], Berglund and Larson, [1982], and Blem and Ramanarayanan [1985]). Inclusion of

growth rate as a statistical variable would require a stochastic formulation of the population balance and is beyond the scope of this work. Fortunately, the phenomenon of growth rate dispersion is usually only observed at small particle sizes and can be ignored in a crystal size distribution dynamics and control study. The purpose of this study is to establish an algorithm for solving the dynamic crystal size distribution equations and to design a crystal size distribution controller. Therefore, to avoid involving any mathematical complexities (e.g. going to a stochastic formulation), the classic growth rate kinetics equation will be used. However, the calculation routine of the boundary condition written as nucleation kinetics will be reserved for an auxiliary routine so that one can substitute any hypothesis for the auxiliary routine instead of Equations (3-9) and (3-10) without disrupting the main control algorithm.

The set of dynamic crystal size distribution equations are conveniently made non-dimensional and normalized (if appropriate) by dividing by their steady state values;

$$x = \frac{LQ_p}{VG} \tag{3-11a}$$

$$\theta = \frac{tQ_p}{V} \tag{3-11b}$$

$$\phi = \frac{G(t)}{\bar{G}} \tag{3-11c}$$

$$u = \frac{n(t,L)}{\bar{n}^o} \tag{3-11d}$$

$$h(x) = \frac{Q(L)}{Q_P} \begin{cases} R, & x<xPvFPv \\ 1, & x \text{ in } (x_F,x_P) \\ z, & x>x_P \end{cases} \tag{3-11e}$$

for the R-z crystallizer, and

$$f_{j,k} = \frac{\int_0^{L_k} L^j n \, dL}{\int_0^{L_k} L^j \bar{n} \, dL} \tag{3-11f}$$

Substitution of equations (3-11) give a dimensionless and normalized form of the set of dynamic crystal size distribution equations as

$$\frac{\partial u}{\partial \theta} + \phi \frac{\partial u}{\partial x} + hu = 0 \tag{3-12}$$

$$\phi = \frac{1 + \beta f_{3,1}}{1 + \beta} \quad \frac{1}{f_2} \tag{3-13}$$

$$u_0 = f_3^j \phi^{i-1} \tag{3-14}$$

$$\beta = \frac{(R - 1)}{6 D_P} \int_0^{x_f} x^3 \bar{u} \, dx \tag{3-15}$$

where u_0 represents nucleation rate as the boundary condition. In Equations (3-12) through (3-15), the independent arguments θ and x are not shown in the functions of the dependent variables u to simplify the presentation. Notice that Equation (3-15) represents the ratio of the fines mass dissolved to external production. A detailed derivation of the above equations can be found elsewhere, e.g., Randolph, Beer, and Keener [1973].

Equation (3-12), a first order partial differential equation, becomes a heterogeneous first order partial differential equation after substituting the right hand side of the equation for an external input vector $\underline{r}(\theta)$. Including the external input, there exist three methods to solve this set of equations.

a) The method of characteristics

b) Finite difference methods

c) The method of lines

In addition to the above techniques, Beckman and Randolph [1977] solved the set of the equations using multiple Laplace transforms, but this method is inappropriate for the control problem.

Among these three methods, the method of lines is feasible for both simulation and control studies of crystal size distribution. The others have defects. The method of characteristics is a natural way of simulating wave-like hyperbolic equations, but requires excessive logic and internal computer memory. It is also not readily adaptable to state space formulation. The finite difference method is a straightforward way to solve a wave equation such as Equation (3-12) but requires excessive computer time. As the time increment becomes smaller, the size element decreases as Δx^2 to maintain stable solutions. This numerical stability limitation is exacerbated with steep wave or pulse disturbances requiring a small step size in the x-axis to avoid truncation errors. This means that a large computer for memory and long calculation times are required to implement the program. For instance, using a second order correct central difference method, the maximum step size in the time axis for Equation (3-12) becomes roughly 0.0001, which barely satisfies the criterion for numerical stability [Vemuri and Karplus [1981]]. Therefore, finite difference methods are not feasible for this problem. The method of lines appears to be the most appropriate technique for simulation of the dynamic crystal size distribution equations. The mathematical development using the method of lines is summarized in Appendix A.

Results using the method of lines are expressed in discrete form as;

$$\underline{u}_{m+1} = \Phi_{m+1}\underline{u}_m + \Delta\theta \frac{\left[\Phi_m B\underline{r}_{m+1} + \Phi_{m+1} B\underline{r}_m\right]}{2} \qquad (3\text{-}16)$$

$$\Phi_m = \exp\left[A_m \Delta\theta\right] \qquad (3\text{-}17)$$

A flow chart of the simulation algorithm for a VAX-VMS computer system is schematically illustrated in Figure 3-3.

The computer program is written in dimensionless and normalized form, but one can enter dimensional data also. The algorithm begins with choosing whether dimensional or dimensionless form is used for data entry. In the former case, the initial conditions for the given data will be calculated by subroutine DDCOND which is summarized in Appendix C. For dimensionless entry, the initial condition of the dimensionless crystal size distribution can be set freely by users. In the next step, information about an R-z crystallizer is entered, then dynamic simulation starts. The simulation results consisting of time, nucleation rate, growth rate, slurry density or third moment, and four chosen crystal size population densities, are printed out. Details of the program are is presented in Appendix D.

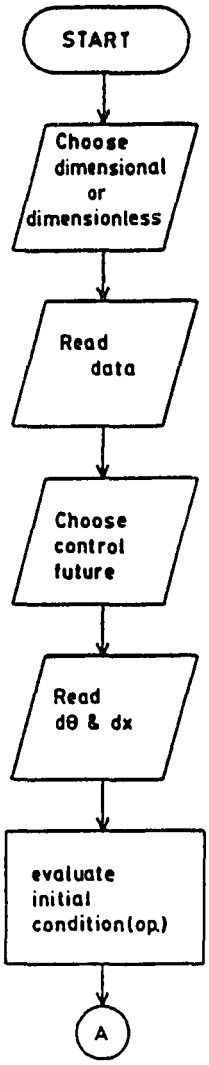


Figure 3-3(a)
A schematic flow chart of a simulation program "DYNE" which is written using the method of lines.

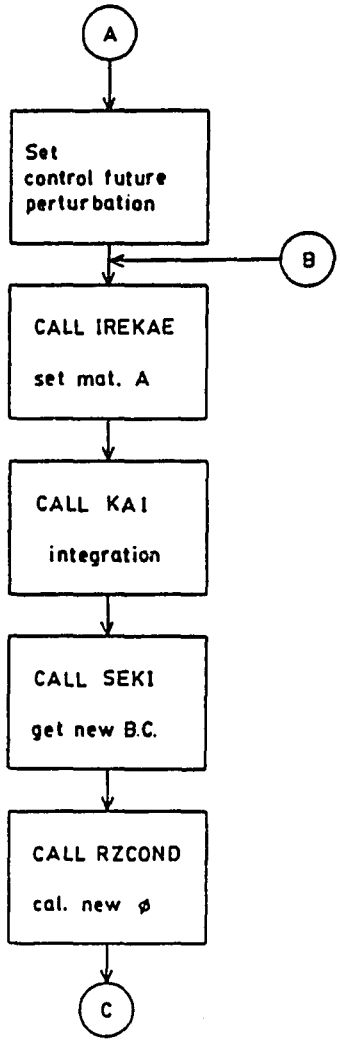


Figure 3-3(b)
A schematic flow chart of a simulation program "DYNE" which is written using the method of lines (continued).

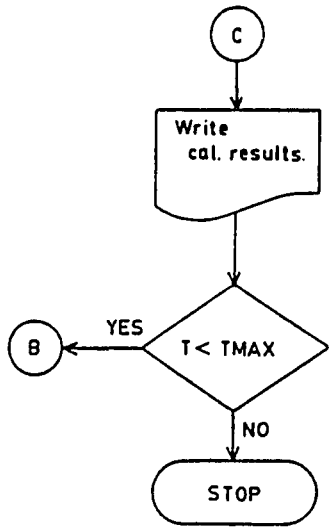


Figure 3-3(c)
A schematic flow chart of a simulation program "DYNE" which is written using the method of lines (continued).

CHAPTER 4

COMPUTER SIMULATION

Stability

In order for the calculation algorithm of the dynamic crystal size distribution, developed in the former chapter, to be valid, it is imperative that the simulation results using the algorithm follow the classical stability criteria. Randolph and Larson [1962] showed the dynamics of crystal size distribution in a Class II MSMPR crystallizer and derived the classic stability constraint that

$$i \equiv \frac{d\ (\log B^{o})}{d\ (\log G)} < 21 \qquad (4\text{-}1)$$

for a stable crystal size distribution. The above result is independent of the specific form of nucleation/growth rate kinetics. Randolph, Beer, and Keener [1973] investigated stability criteria of crystal size distribution in the classified product crystallizer with fines removal, namely the R-z crystallizer, using spectral techniques. A typical stability chart where the crystal size at the lower cut size of the product classifier x_p is a parameter and a recycle ratio of the classifier z is equal to seven is illustrated in Figure 4-1.

Simulation results for an MSMPR crystallizer and an R-z crystallizer are illustrated in Figures 4-3 and 4-4 respectively. The simulation conditions and parameters used are summarized in Table 4-1, and the pulse configurations of disturbances are schematically illustrated in Figure 4-2.

These simulation results correlate well with the theoritical criteria of Randolph, Beer, and Keener, [1973]. In comparing these simulation results to the MSMPR crystallizer with the stability constraint described in Equation (4-1), it is evident that the contours satisfy the constraint. Around i=21, dimensionless nuclei density yields bounded oscillations; for i>21, the curve diverges, but for i<21, the curves converge. In addition to the above correspondences, the contours in Figure 4-4 also illustrate that these simulation results are in agreement with Randolph, Beer, and Keener's criteria. In these simulations, when the recycle ratio of the fines dissolver R is set to 8.5 (the fines dissolver x_F is set to 0.2), then the fines dissolving parameter becomes $\lambda = 1.5$ by definition. [Note: $\lambda = x_F(R-1)$, the dissolving parameter.] Thus, from Figure 4-1, the critical point (assuming the crystal size at the lower cut size of the product classifier x_P equals three) becomes a nucleation sensitivity parameter of i=3.8. This means that the crystallizer under the above condition becomes unstable for nucleation/growth rate sensitivity $i \geq 3.8$. Figure 4-4 shows that the crystallizer system assumed for these simulations converges at i = 3, and diverges at i = 6 and i=10. The simulation algorithm is, thus,

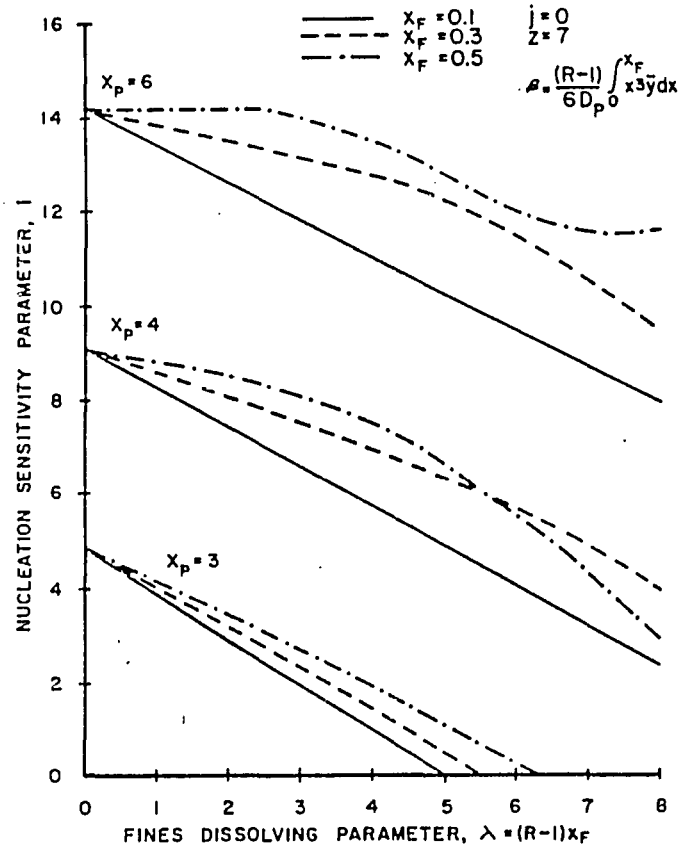


Figure 4-1 Crystallizer stability for the R-z crystallizer with fines dissolving and classified product removal. (magma-independent nucleation), [Randolph, Beer, and Keener [1973]].

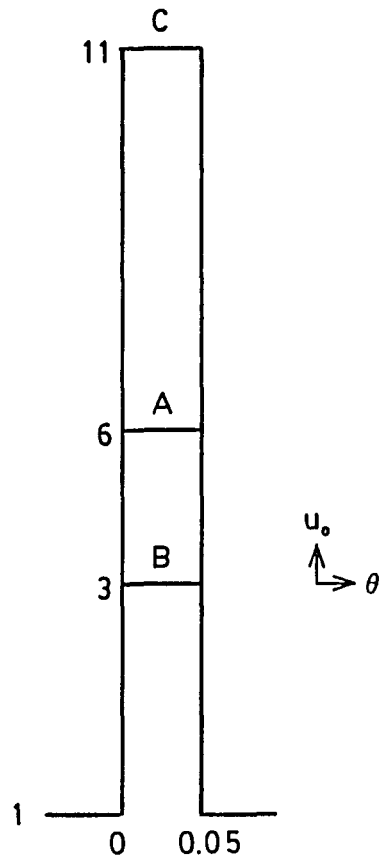


Figure 4-2 Nuclei density pulse configulation.

TABLE 4-1 Simulation conditions for stability tests

RUN	x_F	x_P	R	z	i	j	dθ	dx	PULSE
1-1	0	10	1	1	10	0	0.05	0.5	A
1-2	0	10	1	1	18	0	0.05	0.5	A
1-3	0	10	1	1	21	0	0.05	0.5	A
1-4	0	10	1	1	25	0	0.05	0.5	A
2-1	0	10	1	1	3	0	0.05	0.5	A
2-2	0	10	1	1	6	0	0.05	0.5	A
2-3	0	10	1	1	10	0	0.05	0.5	A

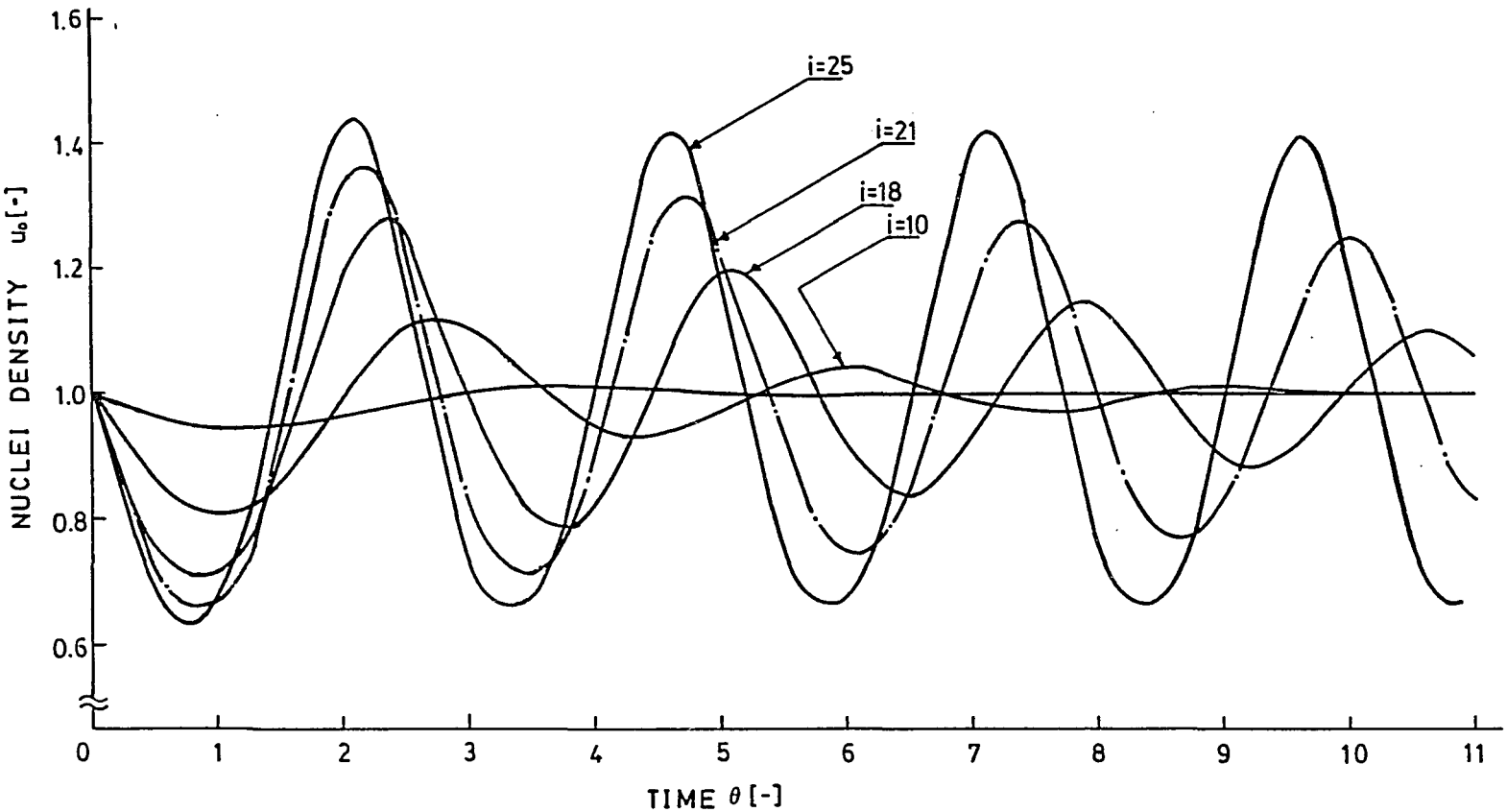


Figure 4-3 Simulation results for an MSMPR crystallizer with various nuclei sensitivities.

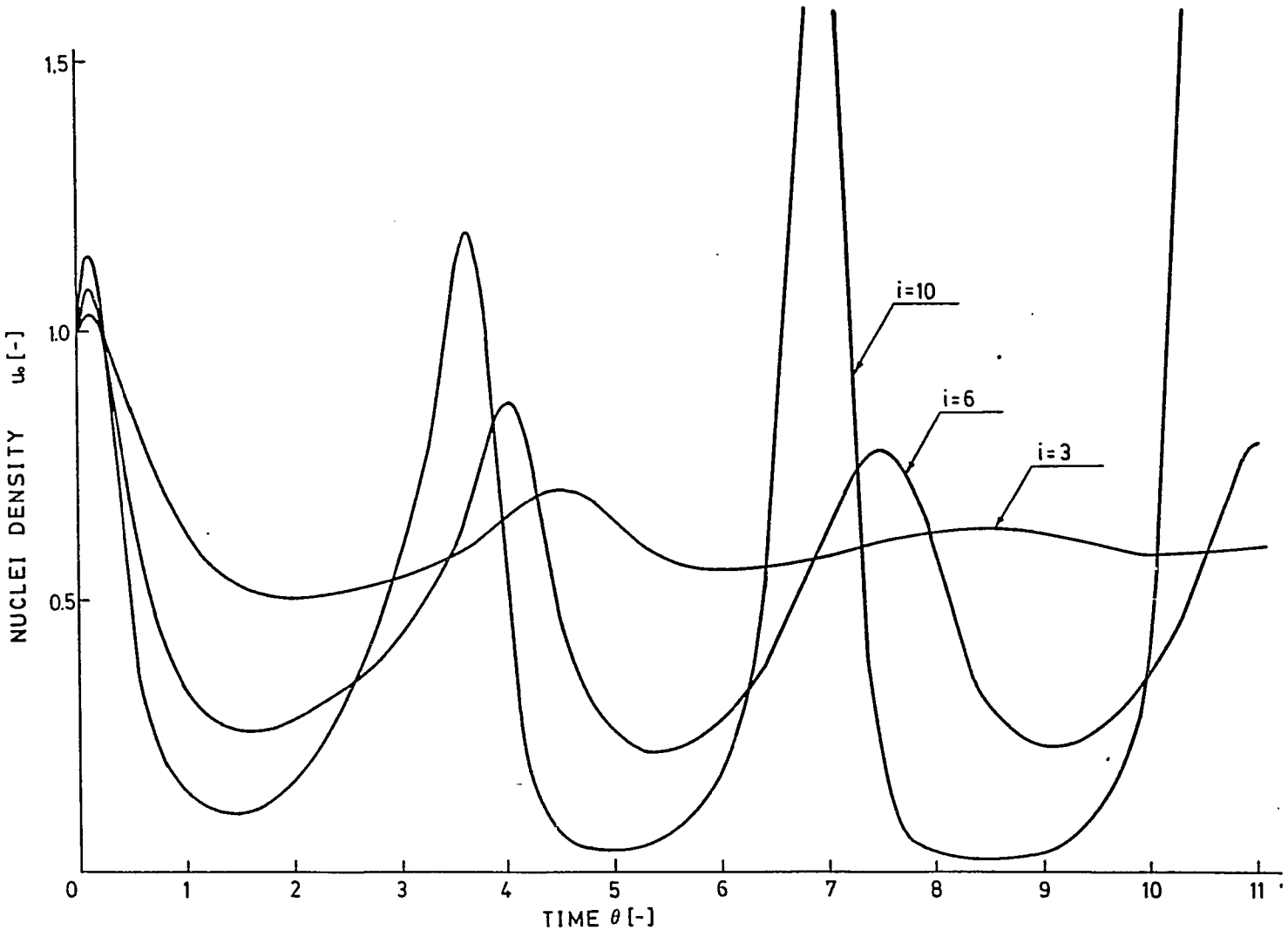


Figure 4-4(a) Simulation results in nuclei density for of R-z crystallizer with various nuclei sensitivities, where R=8.5, z=7, x_F=0.2, x_P=3, and j=0.

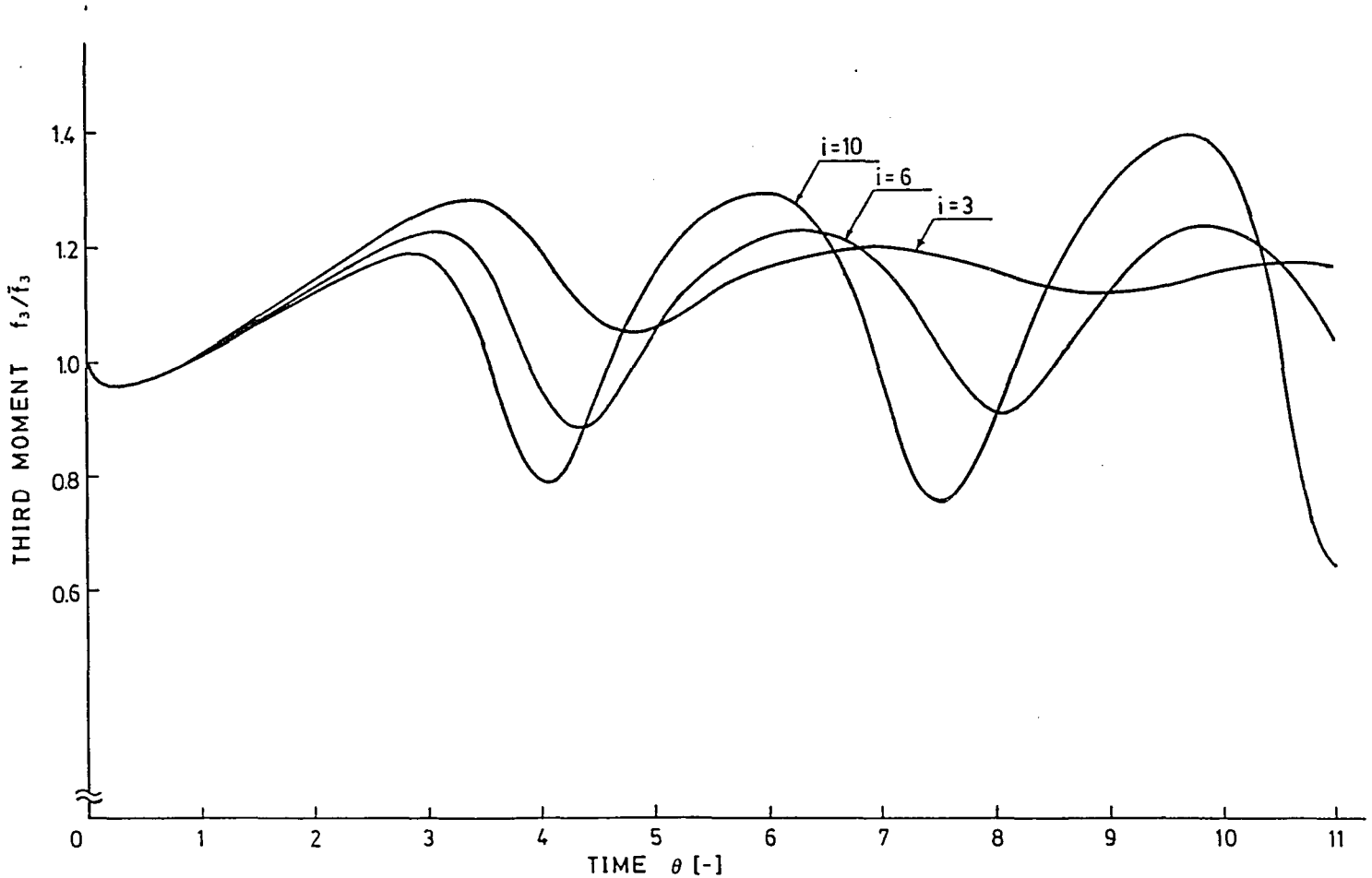


Figure 4-4(b) Simulation results of the third moment for an R-z crystallizer with various nuclei sensitivities, where R=8.5, z=7, x_F=0.2, x_P=3, and j=0.

in accordance with the previous criteria for an R-z crystallizer. These two cases support the validity of the proposed algorithm for dynamic crystal size distribution, at least with regard to stability.

The boundary between stable and unstable conditions for both an MSMPR and an R-z crystallizer is not clearly distinctive; it is rather vague. The stability criteria do not predict the vagueness and no one has yet experimentally or theoretically discussed the sensitivity to the parameter i around its critical value. However, it is intuitively understood that the effect of changing the parameter i around i=21 is reduced compared to that around i=2. For instance, assuming $\omega=0.8$ at the critical point where cycling begins, then;

$$(\Delta\omega)^i = 0.8^3 - 0.8^2 = -0.13 \qquad (4\text{-}2)$$

or

$$(\Delta\omega)^i = 0.8^{22} - 0.8^{21} = -1.84 \times 10^{-3} \qquad (4\text{-}3)$$

The ratio of $\Delta\omega^i$ in Equation (4-3) to that in Equation (4-2) is 0.01, which indicates a smaller sensitivity to the parameter i at larger values of i. Note that a discussion of sensitivity for state variable formulae around on unstable point can be found elsewhere (Kailath, [1981]).

Effect of calculation range of crystal size

The mathematical algorithm for dynamic crystal size distribution has been tested relative to stability criteria in the previous chapter, but the numerical characteristics which should be considered in implementing computer calculations have not been clarified. Among previous studies related to computer simulation of crystallizers, no one discussed the characteristics, though there is always a practical problem in computing large simulations. Numerical characteristics of the method of lines are discussed in Appendix A. In this chapter, the effective calculation range for the crystal size will be discussed.

A computer has finite memory storage, though crystal size stretches from zero to infinite. As is mentioned in Appendix A, the grid size of the crystal size axis is an important factor in obtaining a precise simulation result so that one needs to find an effective upper limit of crystal size which is accommodated to computer storage size. Note that storage size means the size of the image file which is determined by disc capacity of a main frame computer, or diskette capacity of a personal computer. Furthermore, dynamic simulator DYNE discribed in Appendix F has seven independent square matrices performing general calculation and does not have any function which reduces the matrix size. Thus, determination of the limit of crystal size and the grid size raises a most important problem in obtaining an effective simulation. The effective upper limit of crystal size will be sought using numerical evaluation with

the algorithm developed in Chapter 3.

To do the evaluation, first, calculation of ranges of crystal sizes for an MSMPR crystallizer were varied; then, the same operation was implemented for an R-z crystallizer. The parameters used for those calculations are summarized in Table 4-2. Note that the grid sizes were minimized in each calculation because the narrower grid size gives a more precise simulation, as explained in Appendix A.

Figures 4-5(a) and 4-5(b) show an MSMPR crystallizer simulation for different upper limits of the dimensionless crystal size x. The stability criteria reveal that the contour converges for a nucleation sensitivity index i=18. Although the contour at the upper limit of x=10 converges, the contour at x=5 diverges in nuclei density and in the third moment. The same evaluation was implemented for the R-z crystallizer system discussed in the previous section. Since an R-z crystallizer system is more unstable than an MSMPR one, the former divergence appears to be more sensitive. The simulation results for the R-z crystallizer system are presented in Figures 4-6. These figures imply that the upper limit for calculation of crystal size, namely the maximum crystal size, may be set to a dimensionless crystal size x=7.5 in order to hold numerical consistancy for the simulation. This critical value may change with changing recycle ratios of the fines dissolver and of a product classifier because the configuration of crystal size distribution is dependent on those ratios. Comparing data with different ratios, the effective minimum dimensionless crystal size,

TABLE 4-2 Simulation conditions for numerical tests

RUN	x_F	x_P	R	z	i	j	$d\theta$	dx	x-range	PULSE
3-1	0	10	1	1	18	0	0.05	0.5	5	A
3-2	0	10	1	1	18	0	0.05	0.01	10	A
4-1	0.2	3	8.5	7	6	0	0.05	0.2	10	B
4-2	0.2	3	8.5	7	6	0	0.05	0.1	5	B
4-3	0.2	3	8.5	7	6	0	0.05	0.1	5	B
4-4	0.2	3	8.5	7	6	0	0.05	0.15	5	B

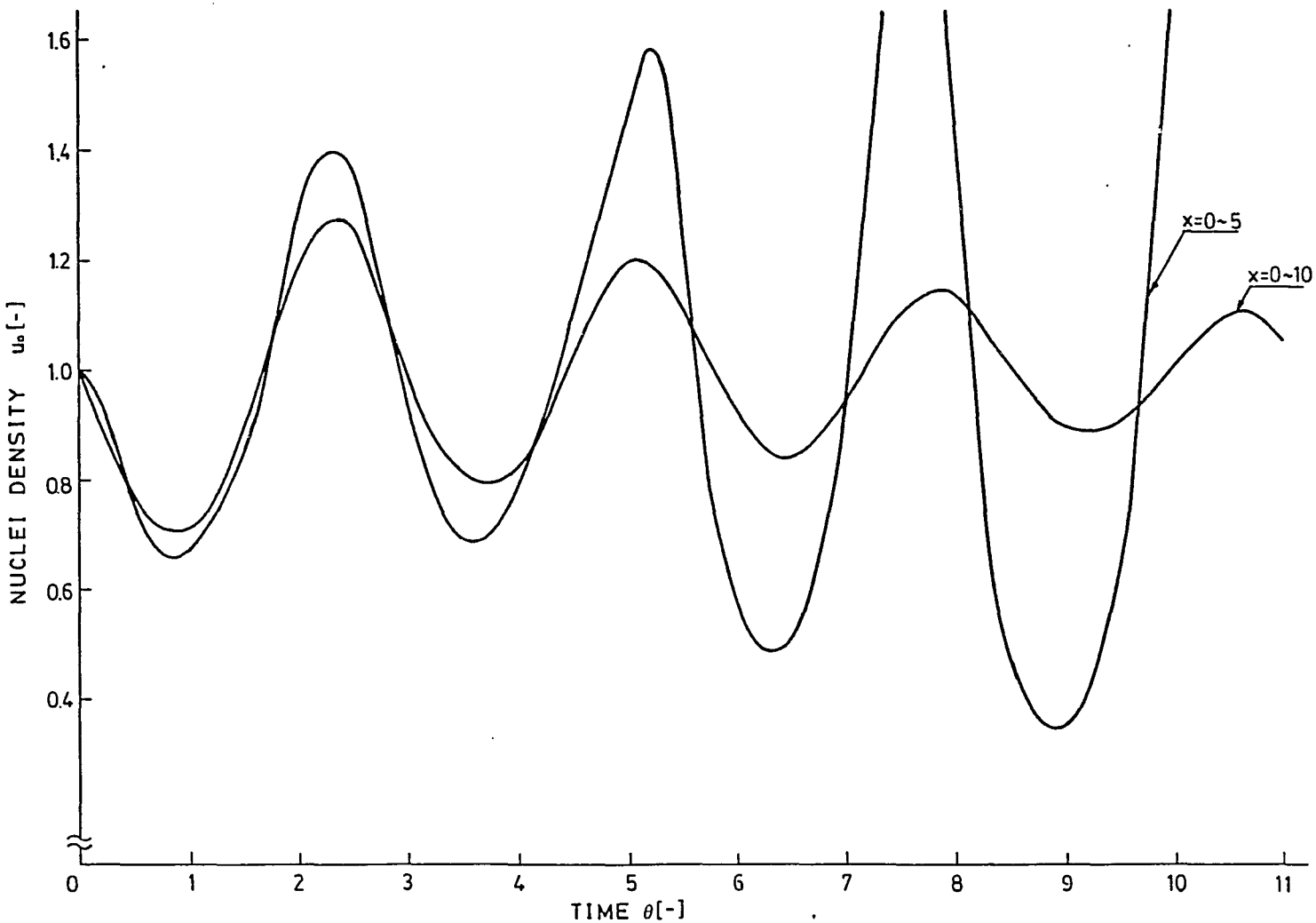
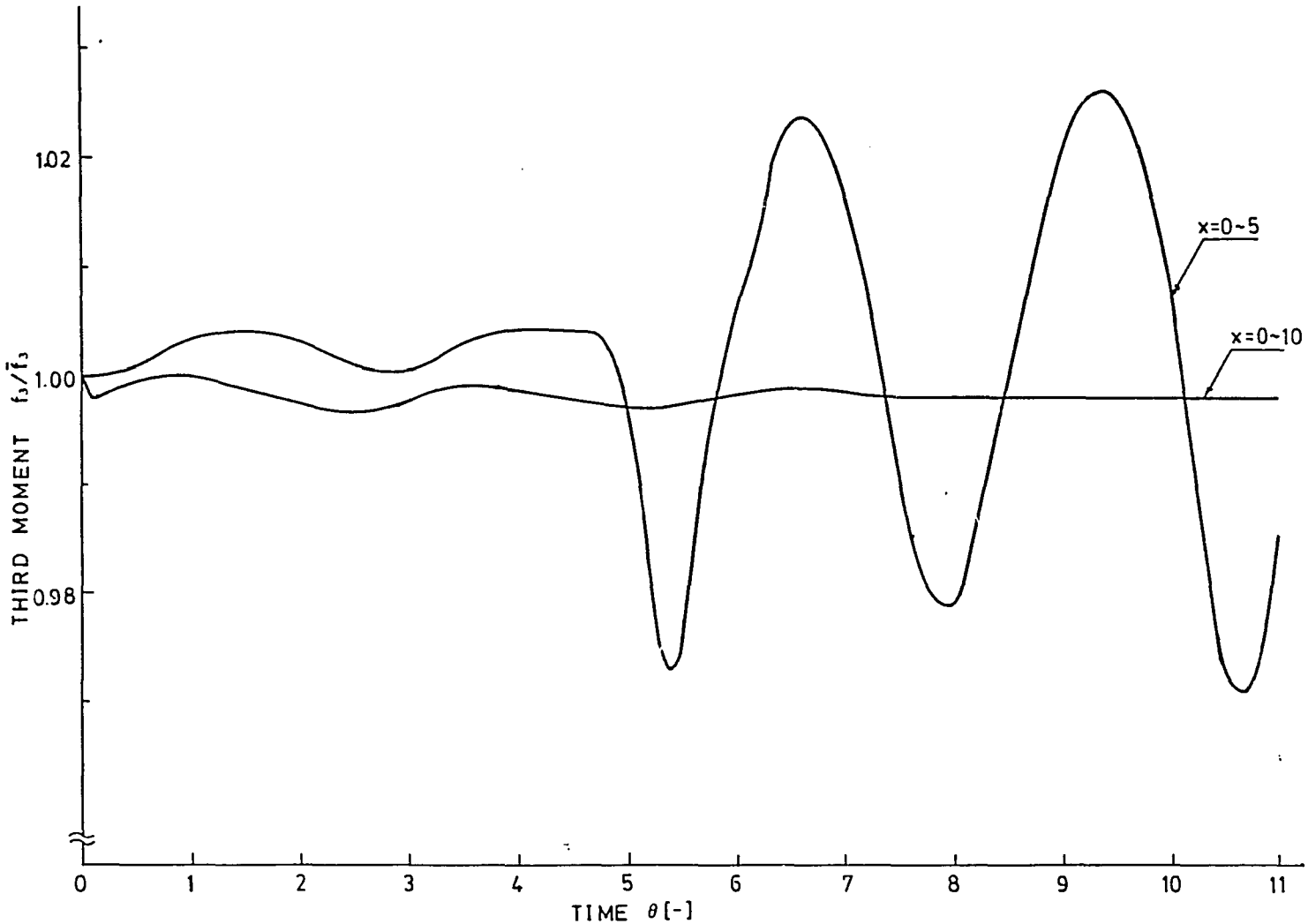


Figure 4-5(a) Simulation results of nuclei density for an MSMPR crystallizer changing calculation range of dimensionless crystal size x.



Figuer 4-5(b) Simulation results of the third moment for an MSMPR crystallizer changing calculation range of dimensionless crystal size x.

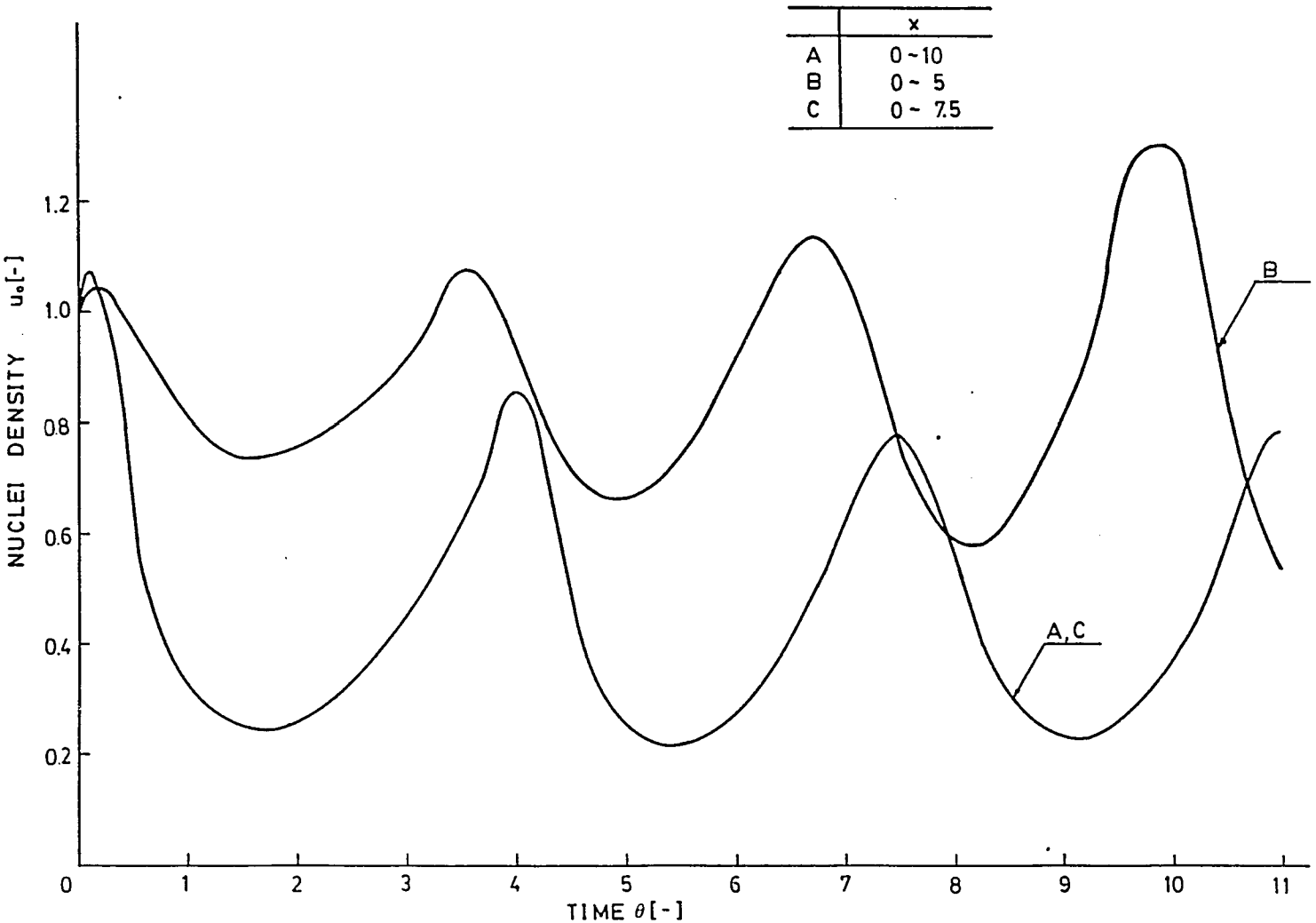


Figure 4-6(a) Simulation results of nuclei density for an R-z crystallizer changing calculation range of dimensionless crystal size x, where x_F=0.2, x_P=3, R=8.5, z=7, i=6, and j=0.

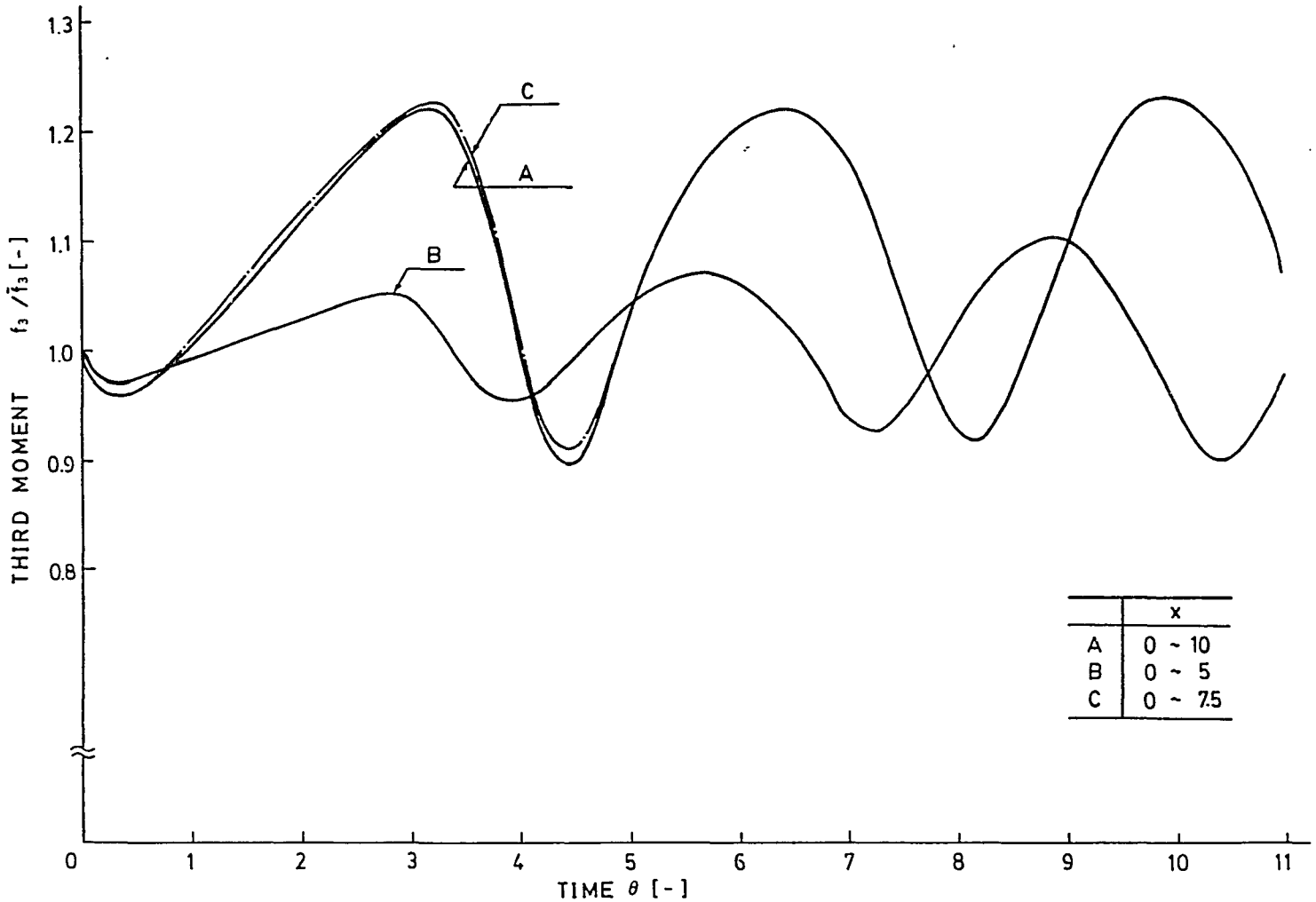


Figure 4-6(b) Simulation results of the third moment for an R-z crystallizer changing calculation range of dimensionless crystal size x, where x_F=0.2, x_P=3, R=8.5, z=7, i=6, and j=0.

x=10, is adopted throughout this study except in specific cases. This value automatically gives a corresponding dimensionless grid size of $\Delta x=0.2$.

Propagation of nucleation disturbances

Crystal size distribution, as is mentioned repeatedly, is an important factor in evaluating a crystallizer system; the dynamic crystal size distribution is especially important in analizing the performance of the crystallizer system. In previous studies, dynamics were often illustrated in the simulation by responses of the moments, but not by response of the crystal size distribution at specific sizes. Thus, typical crystallizers will be investigated for propagation of a disturbance at size equal zero, i.e., nuclei density. It seems that a three dimensional graphic method visually helps one to understand the characteristics of the crystallizer systems, but the EVAX-I, VAX-VMS system at the University of Arizona does not provide a CAL-COMP graphic terminal, and the storage for simulation histories is limited so that the three dimensional graph could not be drafted. However, some specific population density values were printed out. Note that simulater DYNE can print out all grid points on the crystal size axis, but that increases the calculation time. Printing out each dimensionless size point is not practically feasible. The parameters used in these simulations are summarized in Table 4-3.

The simulation results are shown in Figures 4-7. Three

different systems are presented: an MSMPR crystallizer, an R-z crystallizer without a product classifier, and an R-z one with a product classifier. These contours are in accordance with the theory that fines removal narrows the stable region (Randolph A.D., and M.A. Larson [1971]). In addition to this narrowing, it can be seen that classification of product amplifies the wave height of the response to a pulse type disturbance, which propagates along the crystal size axis. The fundamental characteristics of response to an upset in the R-z crystallizer, namely response time and configurations of waves, are not affected by the product classification.

Effects of recycle ratios

The effects of recycle ratios of a fines dissolver and a product classifier, namely R and z respectively, for dynamic responses to an R-z crystallizer are experimentally known. In this section, those dynamics will be summarized with numerical analysis using the proposed algorithm for dynamic crystal size distribution. The effects caused by product classifier ratio z, were briefly discussed in the previous section. The simulations of this section are summarized in Table 4-4.

The simulation results for various fines dissolver ratios and product classifier ratios are presented in Figures 4-8(a) and (b), and Figures 4-9(a) and (b) respectively. It is evident from Figures 4-8 that offsets for nuclei density increase with an increase

TABLE 4-3 Simulation conditions for typical R-z crystallizers

RUN	x_F	x_P	R	z	i	j	dθ	dx	PULSE
5-1	0	10	1	1	3	1	0.05	0.2	C
5-2	1	5	5	5	3	1	0.05	0.2	C
5-3	1	10	5	1	3	1	0.05	0.2	C

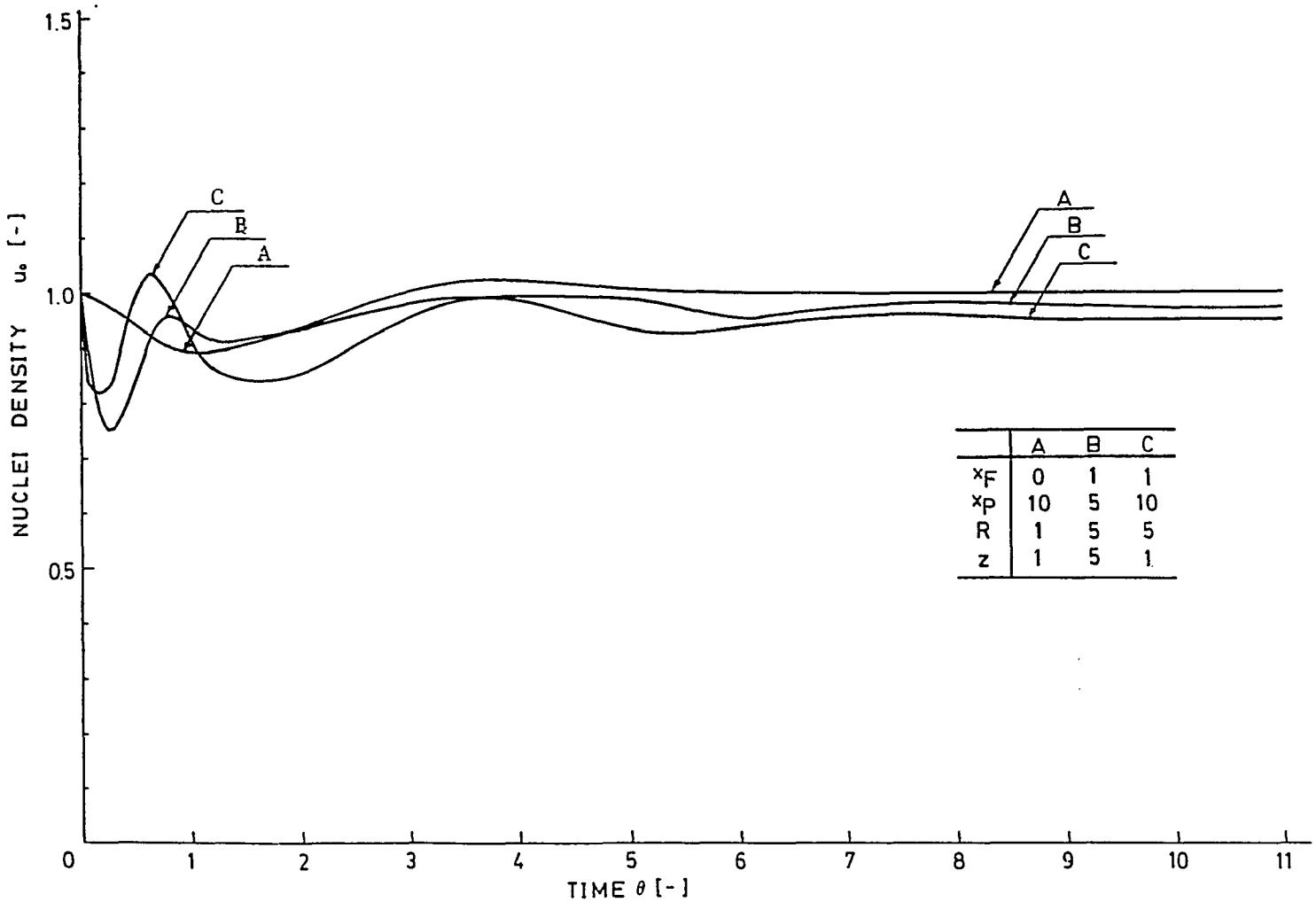


Figure 4-7(a) Propagation of nuclei density upset at x=0.

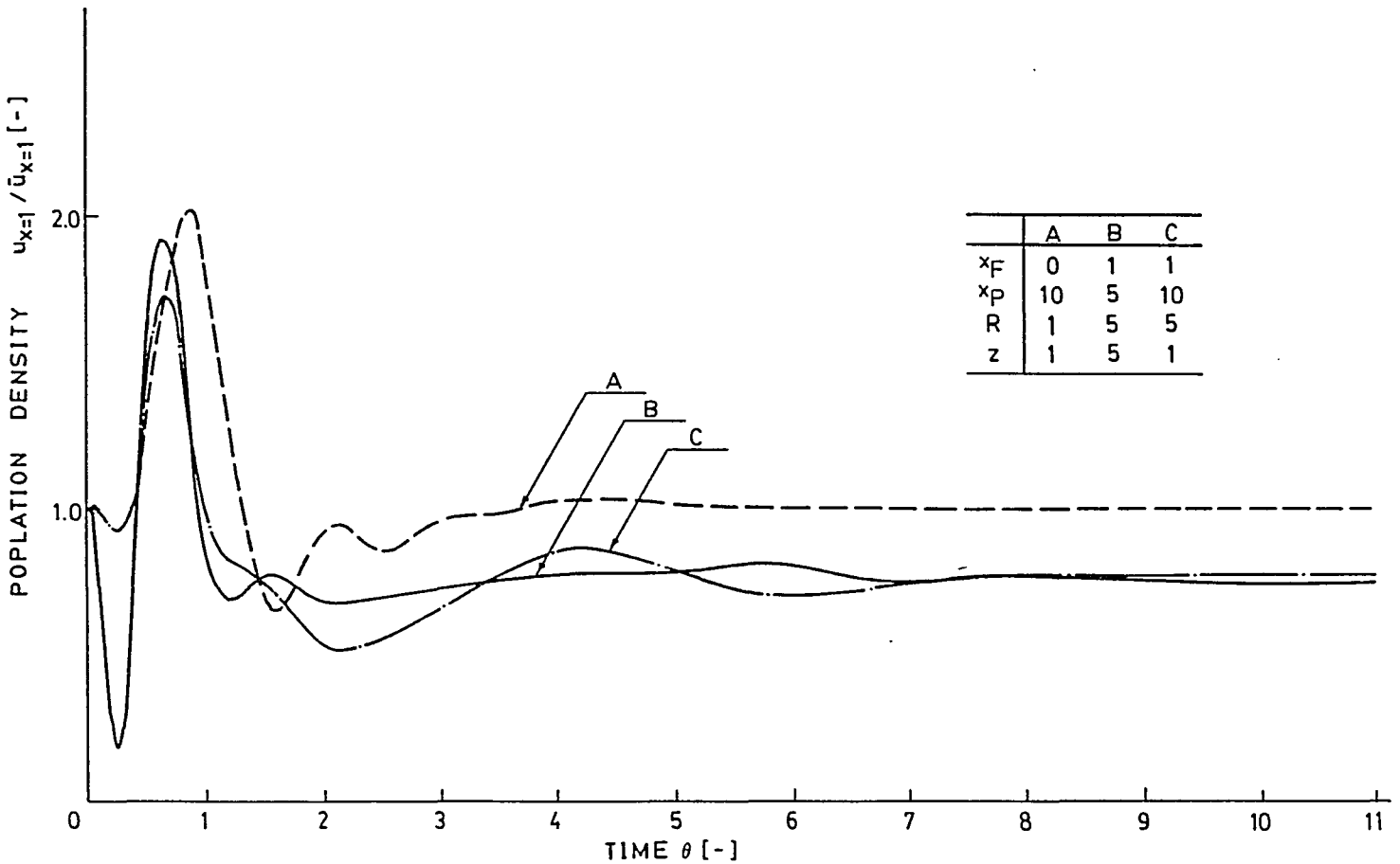


Figure 4-7(b) Propagation of nuclei density upset at x=1.

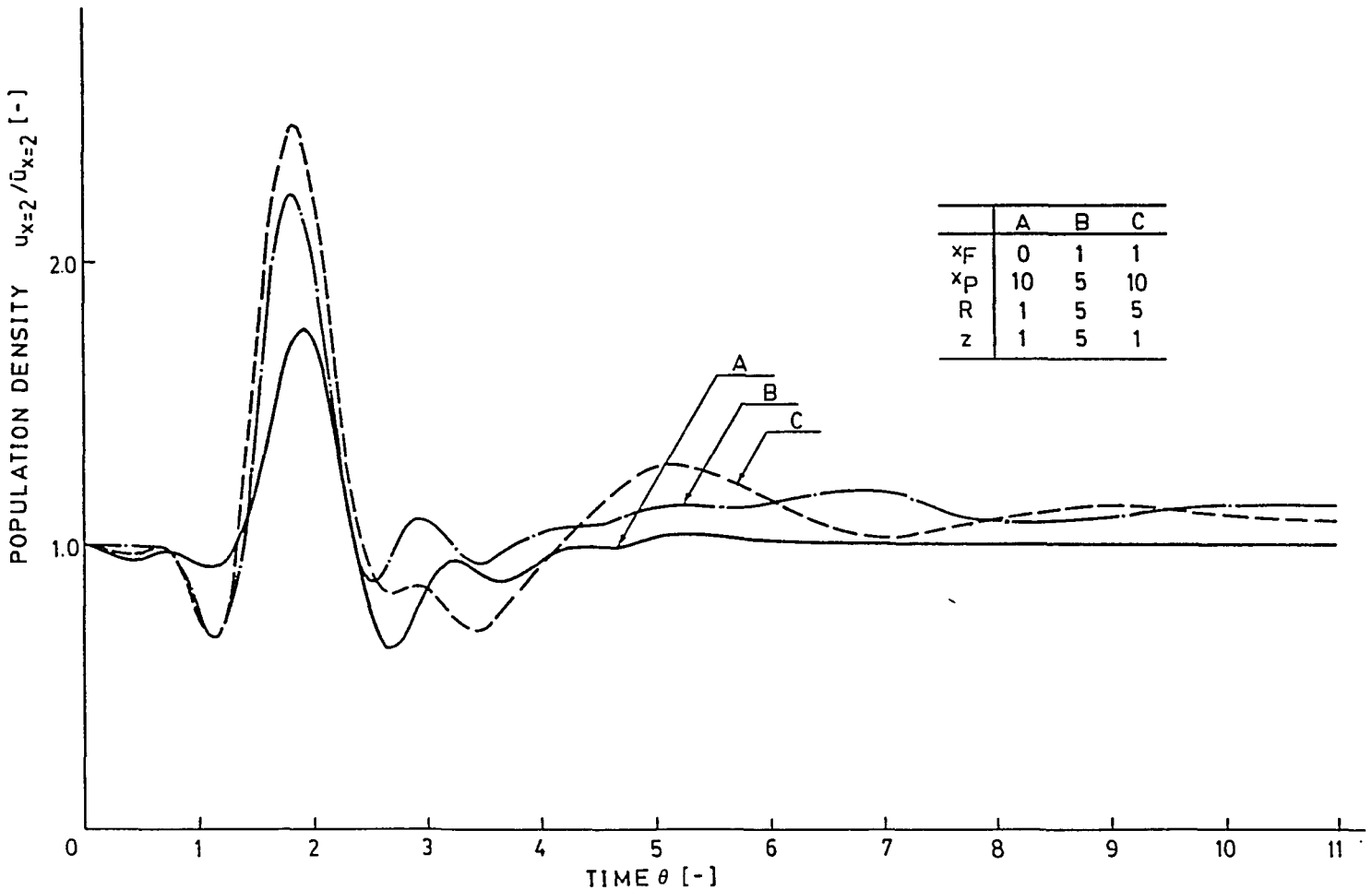


Figure 4-7(c) Propagation of nuclei density upset at x=2.

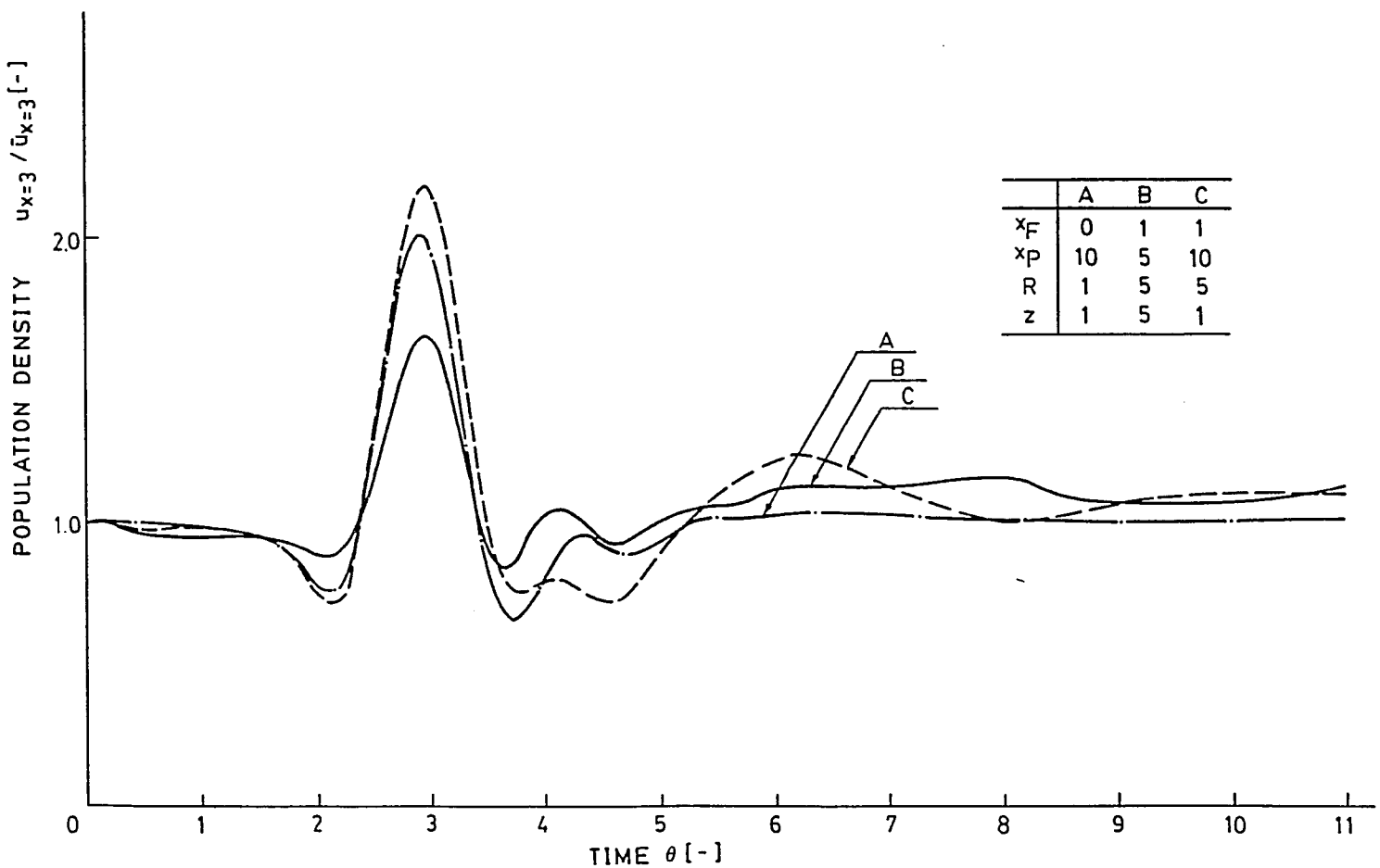


Figure 4-7(d) Propagation of nuclei density upset at x=3.

in the recycle ratio of the fines dissolver, R. This means that using the higher recycle ratio to destruct fines causes not only narrowing of the system stability region but also reducing the ability of recovery to the set point. In other words, the system with the higher recycle ratio is not robust, but still stable under those conditions. Meanwhile, it is also evident from Figures 4-9 that variation of the recycle ratio of the product classifier z affects the dynamics of the crystallizer system only minimally. This means that one can choose the recycle ratio z freely under the stable condition for a specific crystal size at the lower cut size of the product classifier. This is a desirable condition because crystal size distribution can be narrowed with an increase of the recycle ratio of the product classifier and the quality of products from the crystallizer. Crystal size regulation becomes higher with large value of z. Note that increasing the quality using the product classifier also increases the operational cost, consuming more energy; thus, there exists an optimal point for the product classifier in energy and quality. The discussion of efficiency of a liquid cyclone can be found elsewhere (e.g., Perry's Handbook). Consequently, the higher recycle ratio of the fine dissolver is not recommended but that of the product classifier can be independently and freely chosen considering the system economy and the product quality desired.

TABLE 4-4(a) Simulation conditions for R-variation tests

RUN	x_F	x_P	R	z	i	j	$d\theta$	dx	PULSE
6-1	0.2	3	3	7	3	0	0.05	0.2	B
6-2	0.2	3	8.5	7	3	0	0.05	0.2	B
6-3	0.2	3	13.5	7	3	0	0.05	0.2	B
6-4	0.2	3	18	7	3	0	0.05	0.2	B

TABLE 4-4(b) Simulation conditions for z-variation tests

RUN	x_F	x_P	R	z	i	j	$d\theta$	dx	PULSE
7-1	0.2	3	8.5	4	3	0	0.05	0.2	B
7-2	0.2	3	8.5	7	3	0	0.05	0.2	B

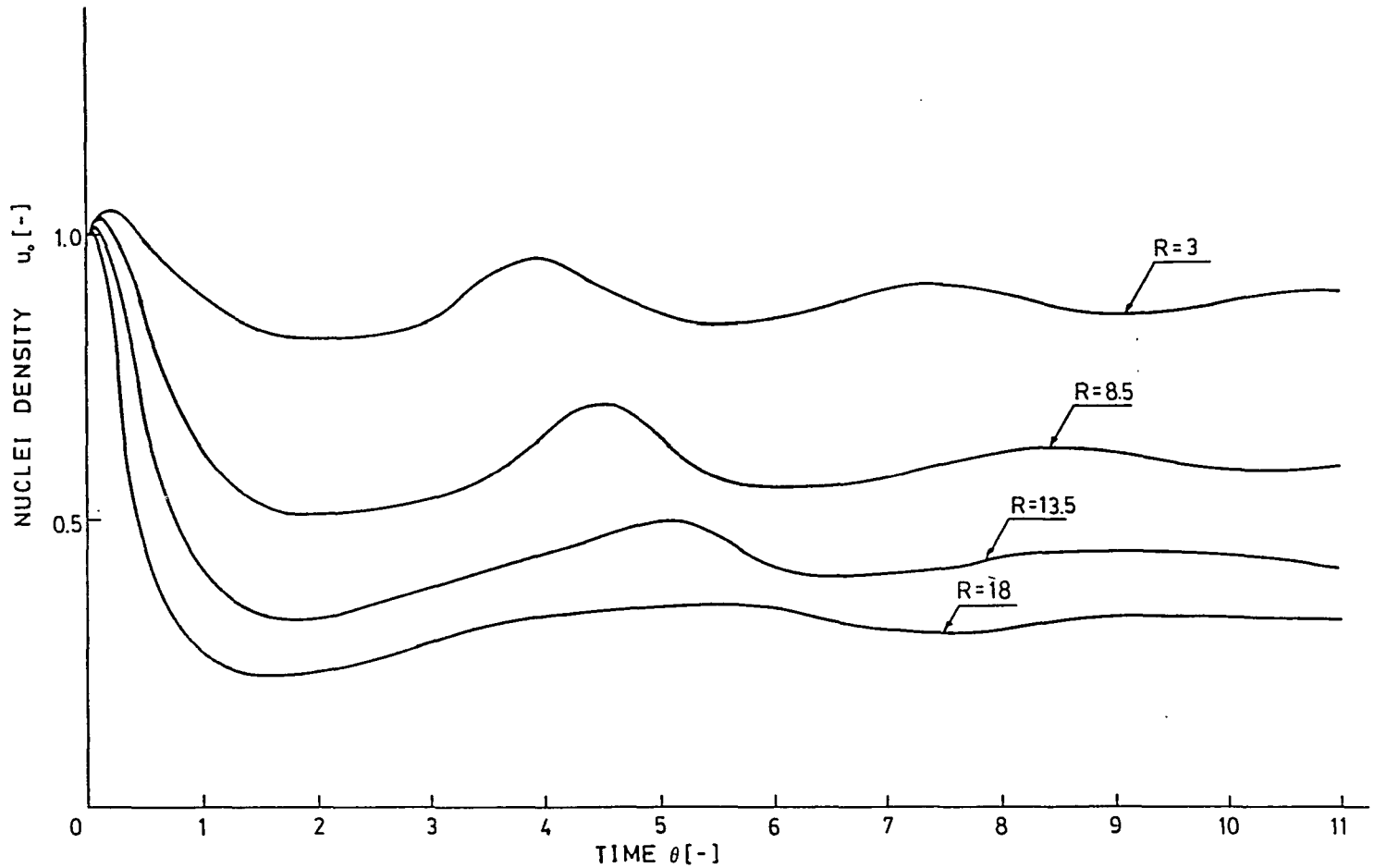


Figure 4-8(a) Simulation results of nuclei density for various recycle ratio R, where x_F=0.2, x_P=3, z=7, i=1, and j=7.

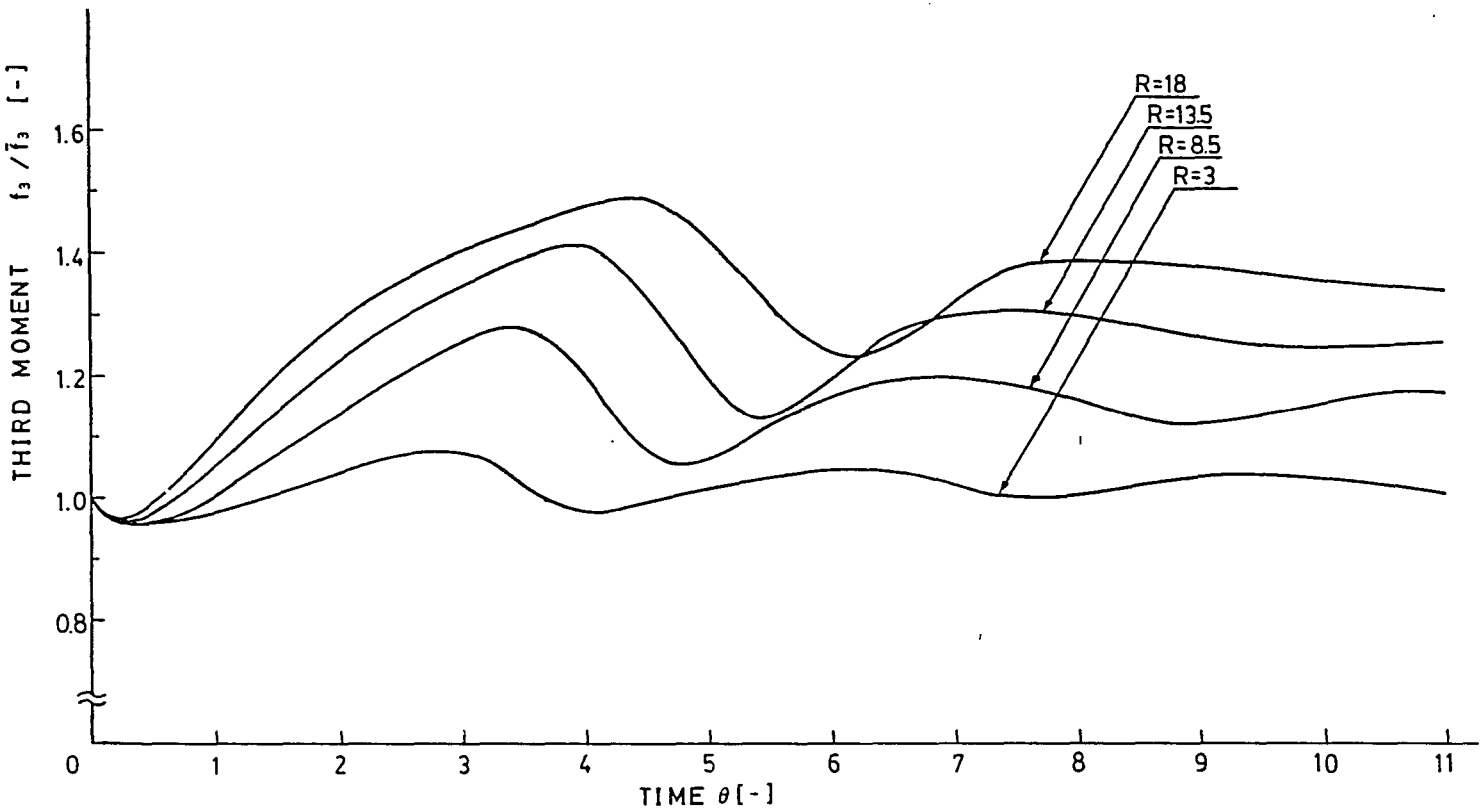


Figure 4-8(b) Simulation results of the third moment for various recycle ratio R, where x_F=0.2, x_P=3, z=7, i=1, and j=7.

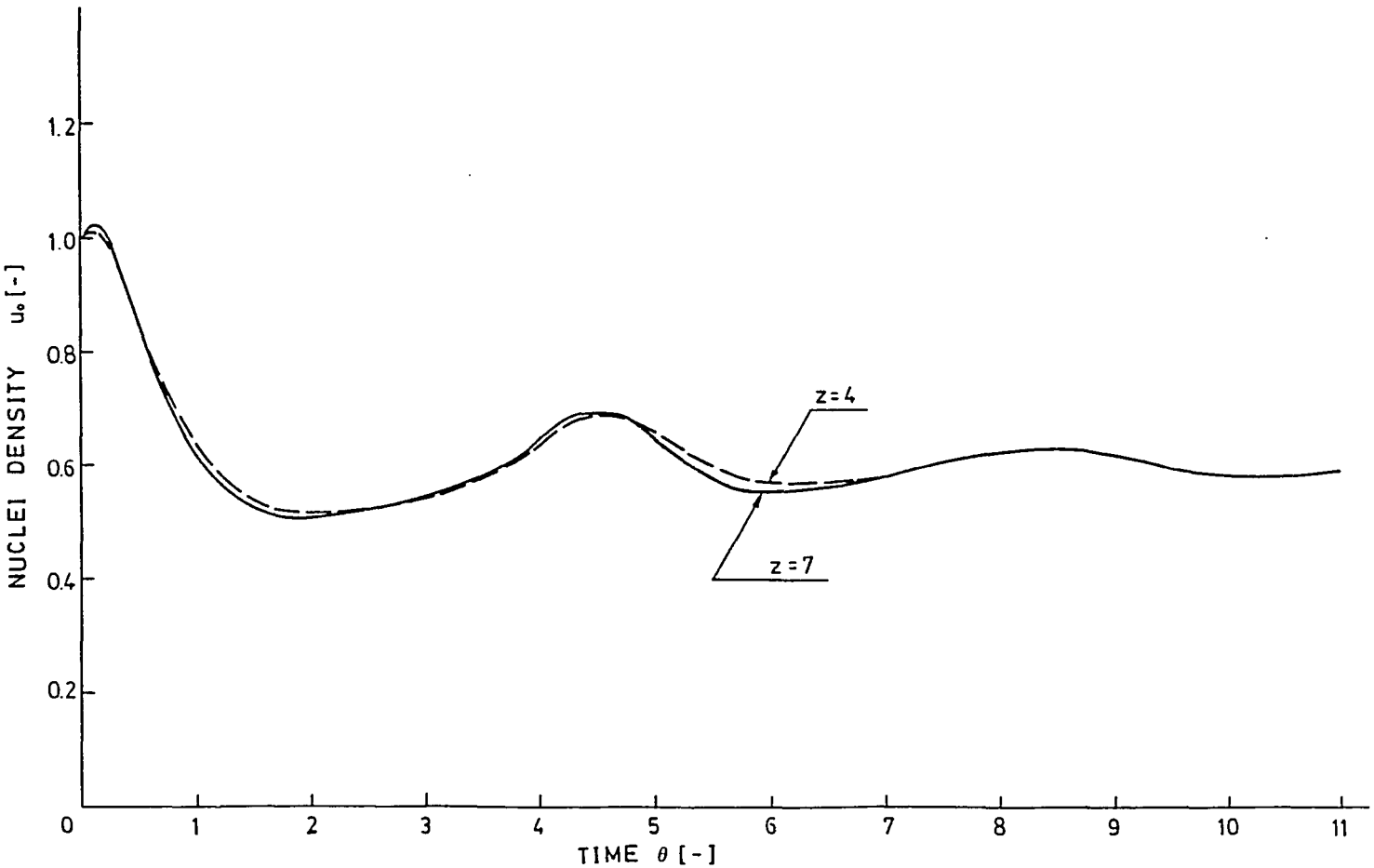


Figure 4-9(a) Simulation results of nuclei density for various recycle ratio z, where x_F=0.2, x_P=3, R=8.5, i=3, and j=7.

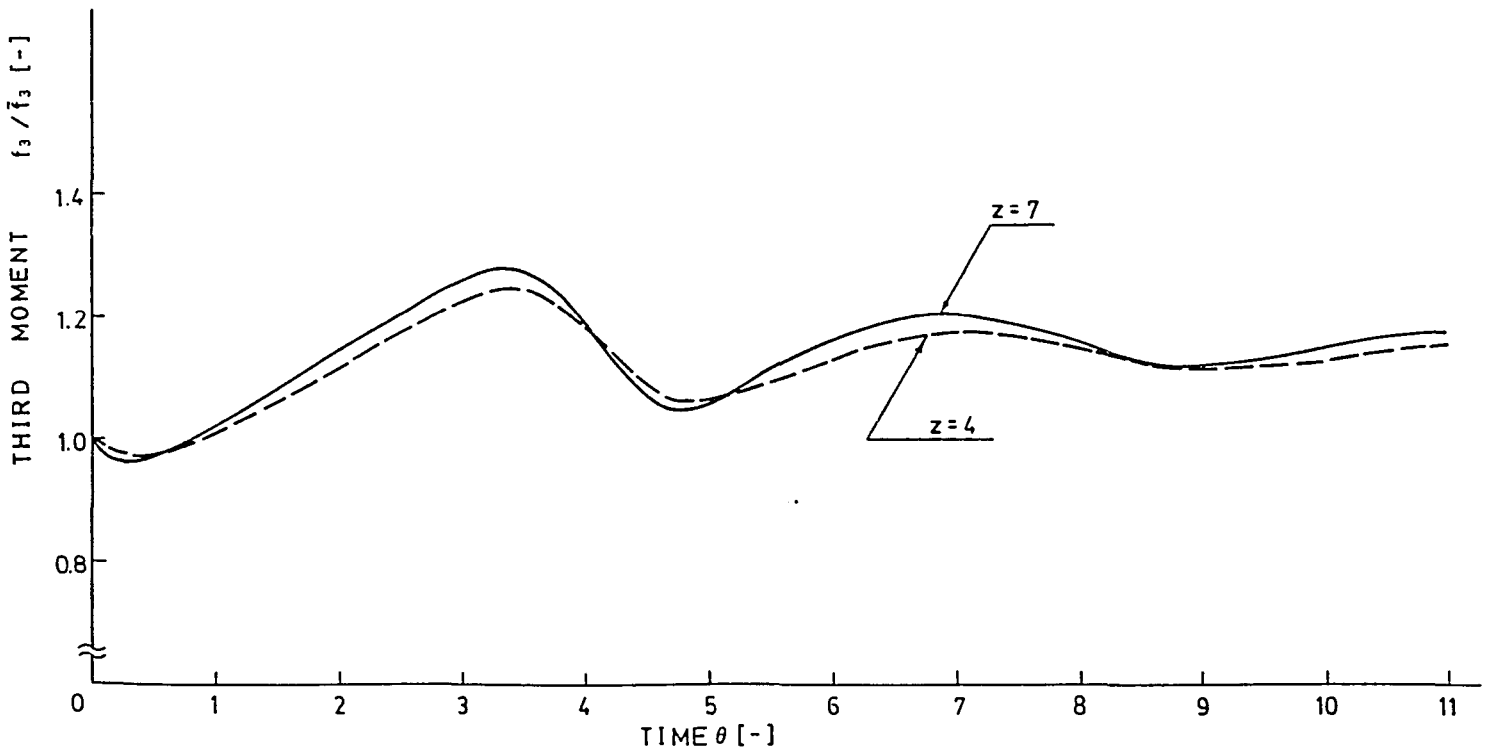


Figure 4-9(b) Simulation results of the third moment for various recycle ratio z, where x_F=0.2, x_P=3, R=8.5, i=3, and j=7.

Effects of cut sizes

The effects of cut sizes of a fines dissolver (x_F) and a product classifier (x_P) for dynamic responses to an R-z crystallizer were predicted by Randolph, Beer, and Keener [1973]. The prediction is summarized that critical nucleation sensitivity parameter i increases with an increase of x_P and a decrease of x_F, in other words, the crystallizer becomes stable for large x_P and small x_F. Figures 4-10 and 4-11 shows that the flutuation vary corresponding to x_P or x_F changes. The simulation conditions are summarized in Table 4-5. It is interesting that response time is shorter in both cases when the fluctuations are large. Thus, to obtaina stable condition, a large lower cut size of the product classifier(x_P) and a small upper cut size of the fines dissolver should be chosen, onthe other hand, to obtain a fast response, a small x_P and large x_F should be chosen as long as those parameters do not make the crystallizer unstable.

TABLE 4-5 Simulation conditions for x_F and x_P variation

RUN	x_F	x_P	R	z	i	j	dθ	dx	PULSE
8-1	1	3	5	5	3	1	0.05	0.2	C
8-2 (5-2)	1	5	5	1	3	1	0.05	0.2	C
8-3 (5-3)	1	10	5	5	3	1	0.05	0.2	C
9-1	0	5	5	5	3	1	0.05	0.2	C
9-2 (5-2)	1	5	5	5	3	1	0.05	0.2	C
9-3	3	5	5	5	3	1	0.05	0.2	C

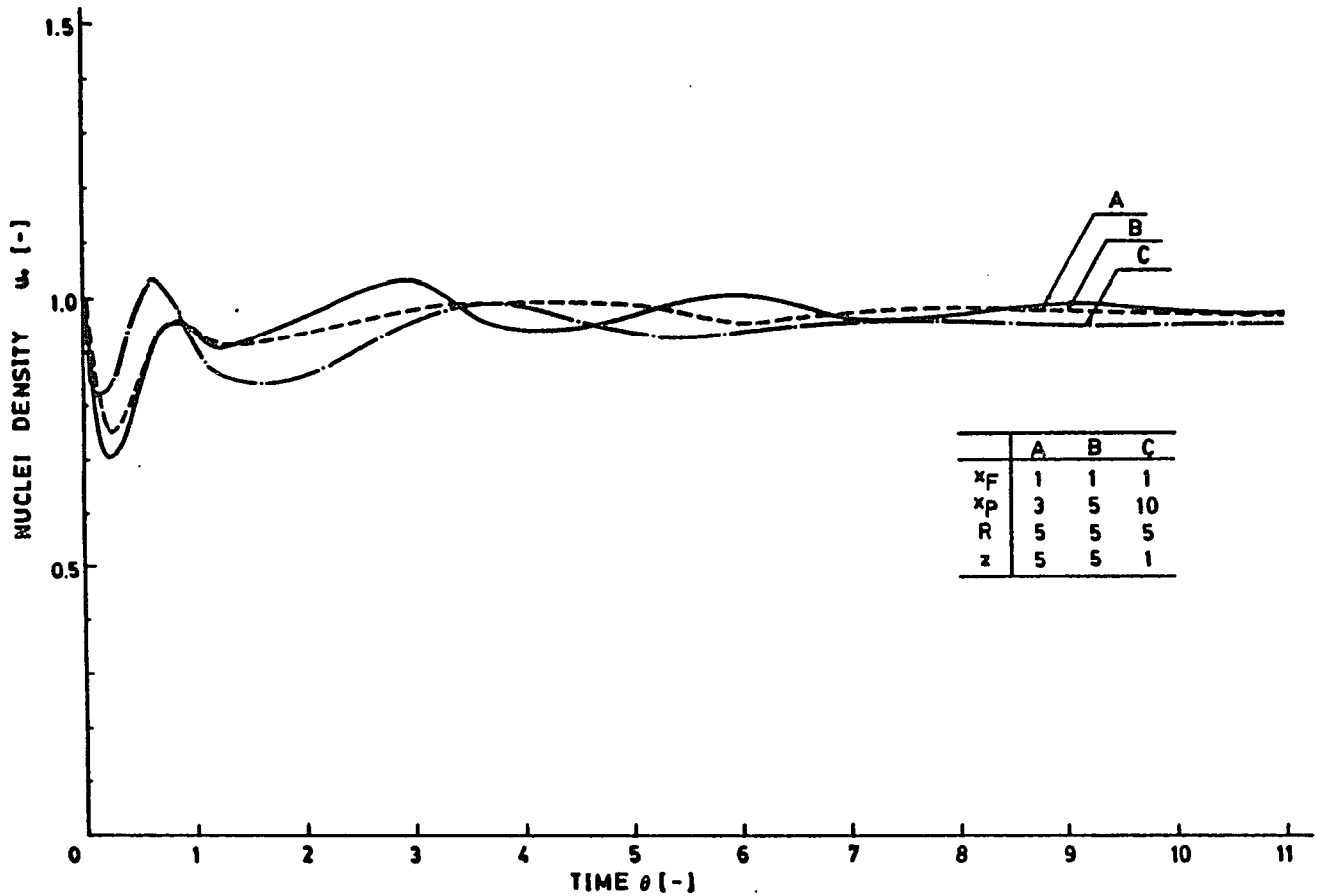
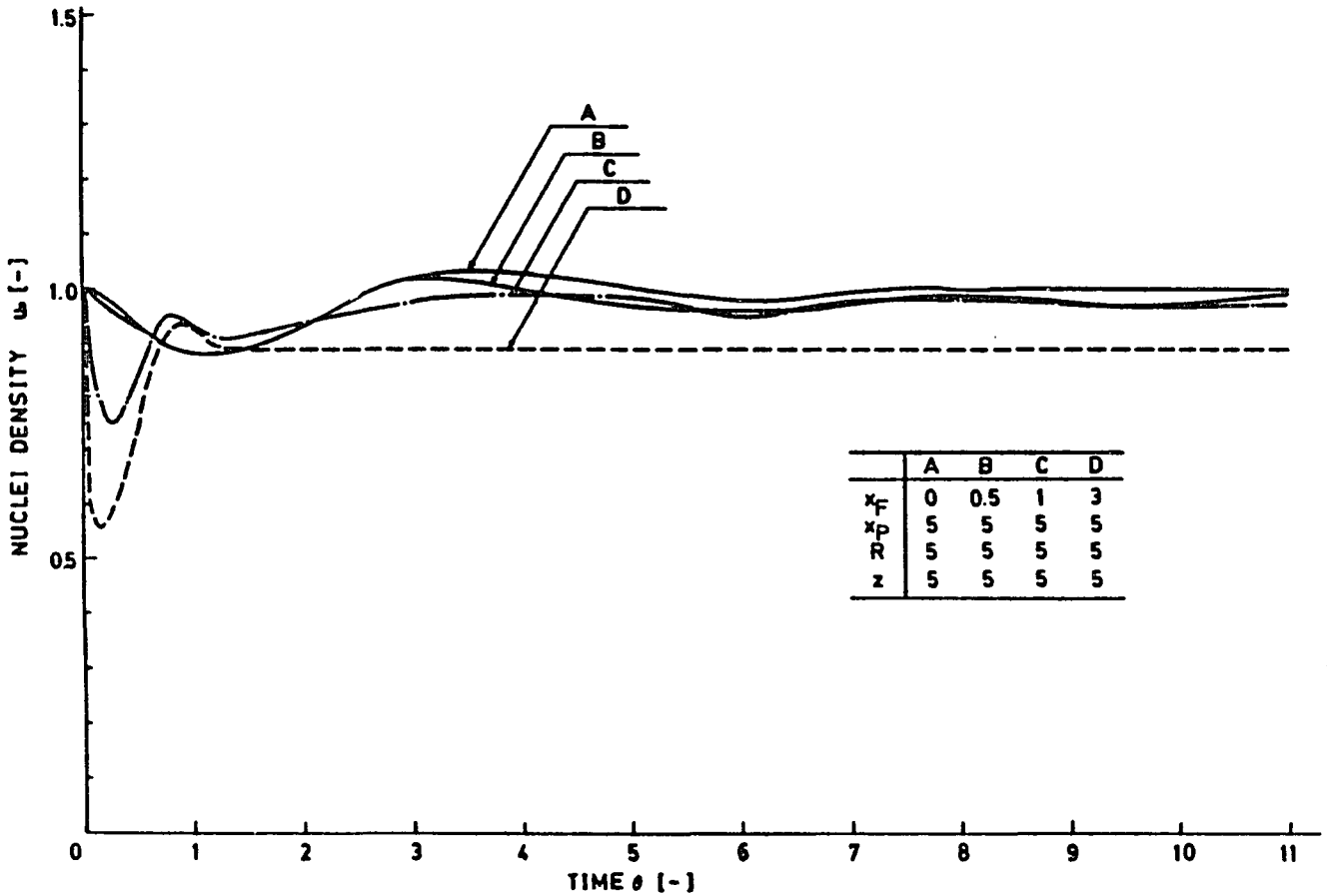


Figure 4-10 Simulation results of nuclei density for various lower cut sizes of the poroduct classifier, where x_F=1, R=5, z=5 (but z=1 for B), i=3, and j=1.



	A	B	C	D
x_F	0	0.5	1	3
x_P	5	5	5	5
R	5	5	5	5
z	5	5	5	5

Figure 4-11 Simulation results of nuclei density for various upper cut sizes of the fines dissolver, where x_p=5, R=5, z=5, i=3, and j=1.

CHAPTER 5

SIMULATION OF AN R-z CRYSTALLIZER

Experimental data

In the previous chapter, a theoretical-numerical analysis is done to justify the simulation algorithm for dynamic crystal size distribution (as developed in Chapter 3). In this chapter, to further validate this simulation algorithm and to seek a predictive model for the dynamics of an R-z crystallizer, a set of experimental results for an R-z crystallizer will be compared with the simulation results. Only a few published results are applicable for comparison between theoretical and experimental models. The information and data necessary for implementation of the simulation are often defectively written. To avoid such estimations and to minimize the degrees of freedom in constructing a simulation, a system which has the maximum amount of data available will be adopted. One set of data reported for an R-z crystallizer will be used [Randolph, Chen, and Tavana [1986]]. The crystallizer will be modeled and the experimental results will be compared to the simulation results.

The experimental apparatus and operation are described below. The quotation is rather long but it is important to adequetly describe the crystallizer system. Furthermore, a program to monitor the data from a CILAS Granulometer 715 particle counter

and analyzer with a Hewlett Packard HP-85 personal computer was developed as a part of this study. Therefore, the experiment is extensively referred to in this study. A sketch of the apparatus is illustrated schematically in Figure 5-1.

The KCL crystallizer consists of a well-mixed draft-tube-baffled 18 liter vessel, operated at 37.8°C, with an associated fines dissolution system shown in Figure 5-1. Saturated potassium chloride solution at 54.4°C was fed from a 200 liter feed tank. A double draw-off product removal system was used giving crystal product residence times of 180 to 240 minutes, depending on the ratio of overflow to underflow streams. Suspension densities ranged from 30 to 120 grams of crystals per liter of clear liquor. The fines stream was removed from an internal fines trap which classified the crystals with a cut-size of approximately 180 micrometers. A separate fines classifier stream continuously flowed through the CILAS particle sizing sample cell. The sample window was a rectangular slit (about 5 cm X 5 cm) having a flow thickness of 1 mm. No plugging of this sample window was observed due to particles in the sample stream. However, the sample window was flushed on a two-hour basis with the dissolved fines stream to prevent salt encrustation.

Crystal dissolution was achieved by raising the fines stream temperature from 54.4 to 60.0 °C. This fines stream was then cooled and returned to the crystallizer. The system was assumed to be at steady state when the changes with respect to time of the weight

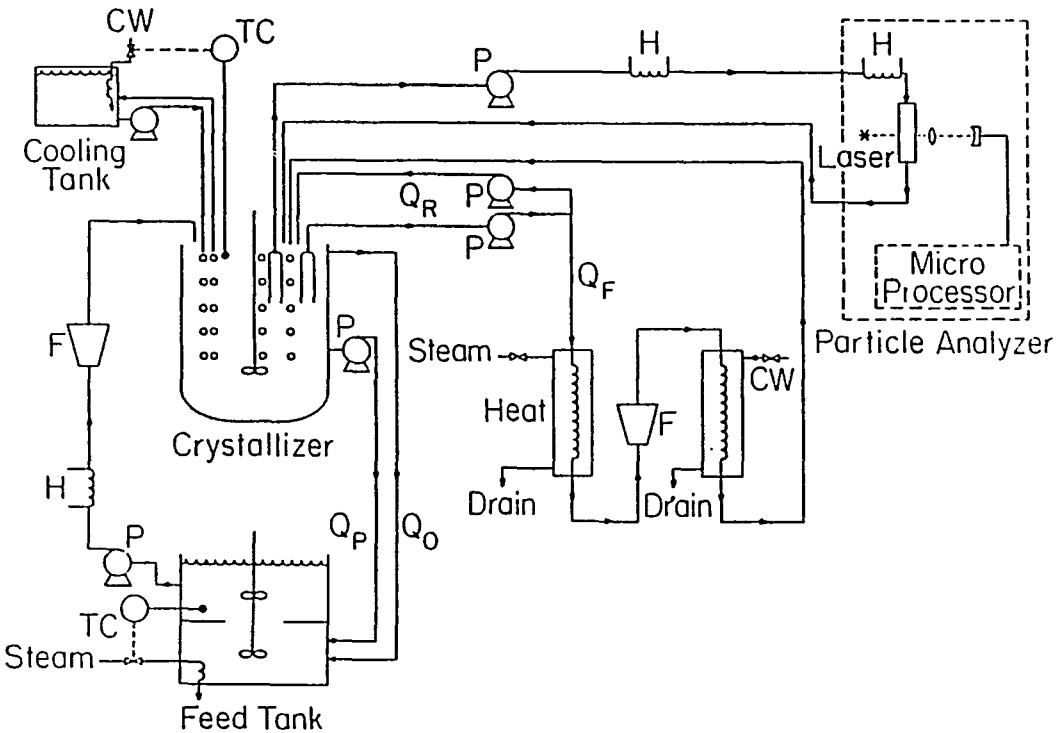


Figure 5-1 Schematic flow diagram of an 18 liter crystallizer (After A. Tavana).

fraction of the crystal product in sieve range i were minimal.

The crystallizer was first brought to steady state and then challanged with a sudden nucleation pulse obtained by upsetting the temperature. The temperature was upset by shutting off the cooling water until the temperature rose (by the hot in coming feed brine) to the desired amount (around 7.2 to 8.3°C). The cooling water was then opened to initiate temperature control. The sudden drop in the crystallizer temperature resulted in a fairly reproducible burst of nuclei. Crystal size distribution was measured throughout the experiment by sieve analysis as well as by on-line measurement of nuclei density in the fines stream. Each run was 30 to 50 hours in length.

The parameters describing the experiments are summarized in Table 5-1. The experimental devices and manipulation for the control study will be summarized in Chapter 6. Additionally, the crystal size distribution data at a steady state point before the upset are summarized in Table 5-2 [Courtesy of A. Tavana, Department of Chemical Engineering, University of Arizona]. The nucleation density tabulated in the last row in Table 5-2 was obtained by Tavana.

TABLE 5-1 Range of variables and crystallizer specifications (After A. Tavana).

Run	Feed (ml/min)	Q_o/Q_u	Q_F (ml/min)	RPM	Temperature, °F			Rise in Crystallizer Temperature during upset (°F)
					Feed	Crystallizer	Fines Destruction Loops	
7/30/85 (#35)	300	2	1000	500	130	100	130	13
8/5/85 (#36)	300	3	Control	500	130	100	130	14.5

Concentration of feed (for all the runs): 300g KCL/L, 300g NaCl/L, 10g $MgSO_4$/L

Crystallizer dimensions: Diameter = 31 cm., Length = 42 cm., Working volume = 18 L

Two sets of concentric 1/2" (4" diameter) and 3/8" (8" diameter) stainless steel cooling coils are used inside the crystallizer. Three vertical baffles are attached to the coils which rest against the tank walls. Three other baffles divide the annulus between the coils. The impeller, located within the inner coil, was a 3-bladed marine-type propeller of 3.7 cm radius.

TABLE 5-2 Crystal size distribution by sieve (After A. Tavana)

SIEVE	L X 10^6 [m]	n(L) X 10^{-11} [#/m^4]
6	921	1.16
7	771	2.25
8	651	2.38
9	548	2.51
10	460	3.08
11	387	2.80
nuclei	0	5.4 X 10^5 *

*Evaluated from extraporlation.

Evaluation and modeling of the crystallizer system

As the experimental system is not identical to the model algorithm developed in Chapter 3 in the initial steady state crystal size distribution, addition of a product classifier is necessary to model the experimental system. Undoubtedly, product classification was occuring in this system at the larger sizes. Such classification can easily be modeled as though an external classifier were present. The parameters used in the computer simulation are listed in Table 5-3. The first four parameters in the table are used to fit the initial crystal size distribution in the vessel. The rest are used for conversion from dimensional forms to dimensionless ones, or vice versa. The numbers corresponding to the notations on the parameters were calculated or obtained from experimental data. A typical contour for the steady state crystal size distribution of the 18 liter crystallizer in this experiment is illustrated in Figure 5-2 [after Randolph et al [1986]]. This figure indicates an anomaly compared to a standard crystal size distribution from an R-z crystallizer, namely the crystal population does not decay properly (exponentially) in the size range L in (L_F, L_P).

It is evident from the tables and figures that there exists some freedom to choose the crystal size at the lower cut size and the recycle product classifier ratio, as well as the slope between the crystal sizes at the upper cut size of the fines dissolver and lower cut size of the product classifier. Indeed, an external product

TABLE 5-3 Summary of experimental data for simulations.

Parameters	Experimental data
x_F	1 [-]
x_P	N/A
R	13 [-]
z	N/A
$\bar{G}$	$\simeq$ 1 [μm/min]
$\bar{n}^o$	5.4×10^{16} [$1/m^3 \cdot m$]
τ	180 [min]
Q_P	100 [mℓ/min]
V	18 [ℓ]

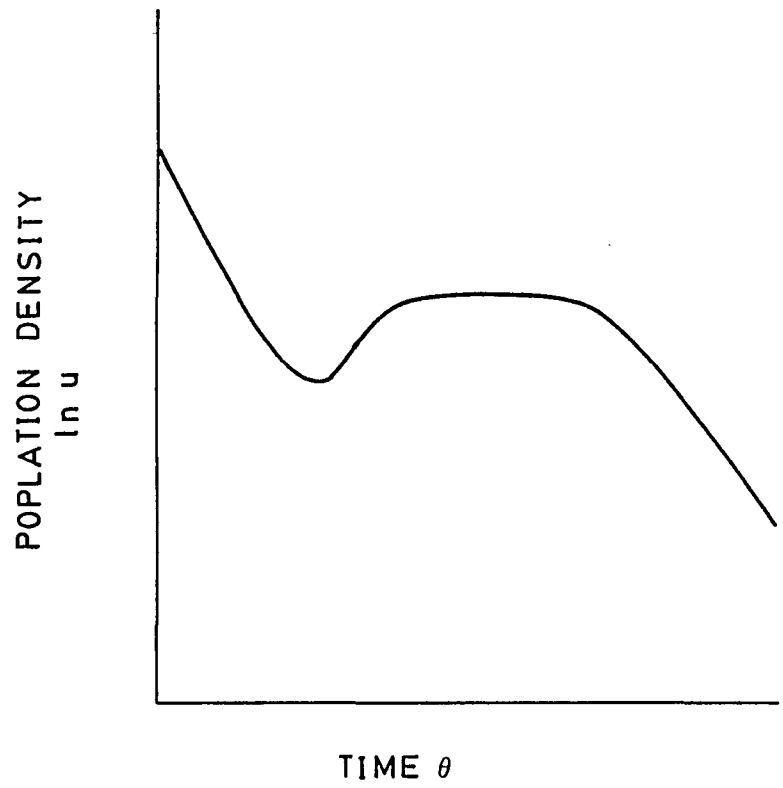


Figure 5-2 Schematic crystal size distribution in 18 liter crystallizer.

classifier did not exist in the experimental system. Crystal classification apparently occured at the bottom of the 18 liter crystallizer because of its imperfect mixing. Hence, it is reasonable to assume a fictituous classifier. The steady state crystal size distribution data listed in Table 5-2 show that the recycle ratio of the fictitiuous product classifier z, is approximately equal to one. Next, the slope between the crystal sizes at the upper cut size of the fines dissolver and the lower cut size of the product classifier can be also evaluated as approximately equal to one from the same table. However, as is shown in Figures 5-3(a) and 5-3(b), the slope is unexpectedly evaluated as zero. The steady state population densities at crystal sizes $L=4.58X10^{-4}$ meters and $L=6.51X10^{-4}$ meters are seen to be approximatedly equal, in other words, no exponential decay occured. Thus, the slope cannot be determined from the experimental data; it is in the range from zero to one. The crystal size at the lower cut size of the product classifier cannot be determined either. It seems that this crystal size x_p may be from 4 to 5 (dimensionless form), or $7X10^{-4}$ to $9X10^{-4}$ meters (dimensional form). Additionally, adopting a value less than one for the slope between the crystal sizes at upper cut size of the fines dissolver and at the lower cut size of the product classifier, z cannot be determined from data. Thus, z may be chosen appropriately for simulation results. Consequently, those indeterminable parameters can be summarized as follows;

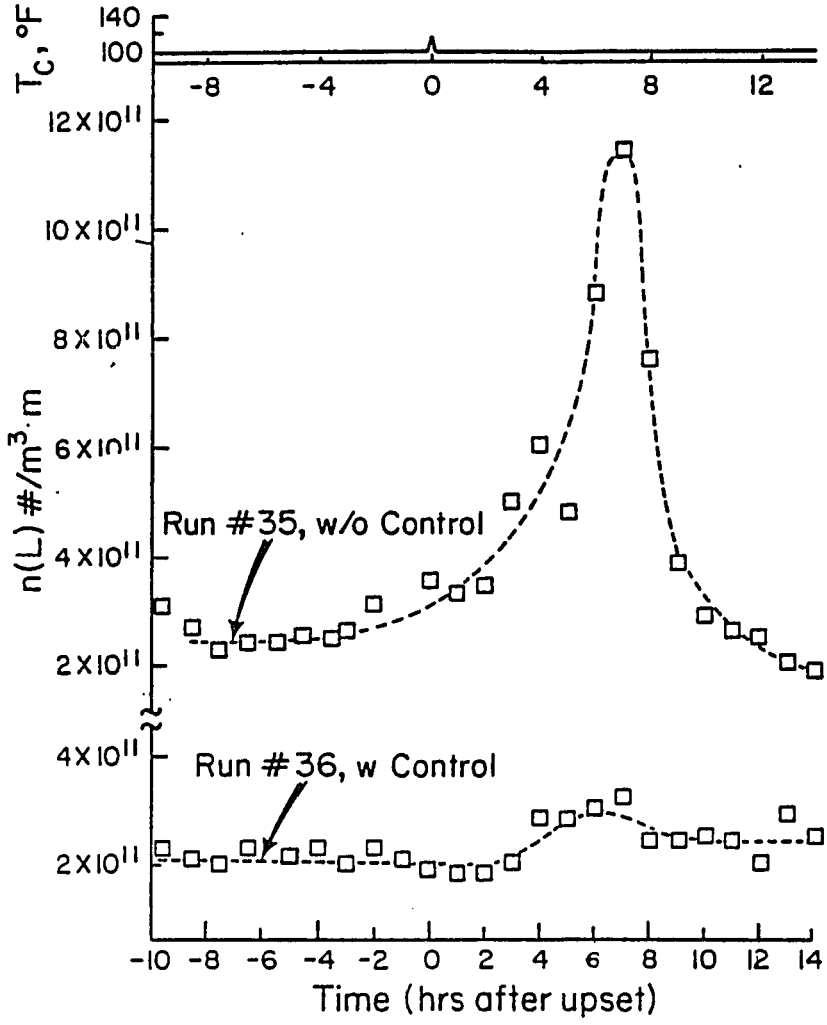


Figure 5-3(a) Population density vs. time at an average size of $4.58X10^{-4}$[m] with and without control (After A. Tavana).

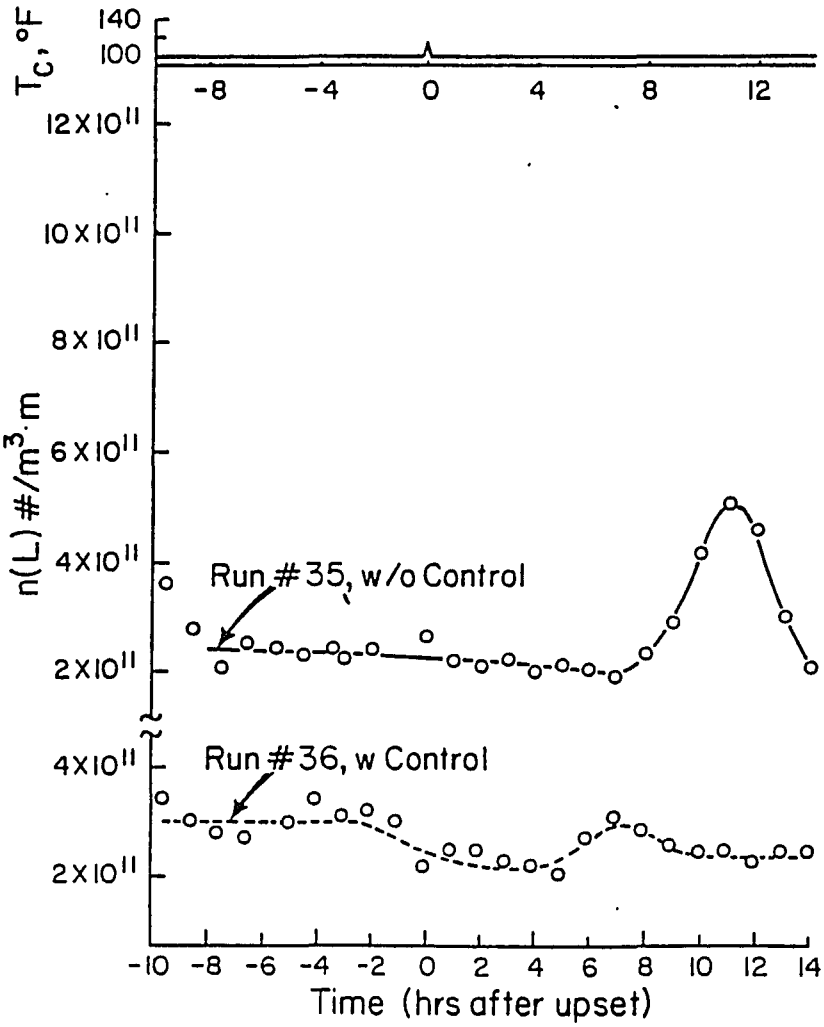


Figure 5-3(b) Population density vs. time at an average size of $6.51X10^{-4}$ [m] with and without control (After A.Tavana).

$z \simeq 1$ or free

$x_p \simeq 4$ to 5

(the slope between x_F and x_p) $\simeq 0$ to 1

The above discussion is a framework for the crystallizer system where the parameters are determined factually or at least by an educated guess. Meanwhile, the discussion does not involve the intensity and shape of the nuclei density upset. To implement the simulation, the upset needs to be established. The problem is that no positive imformation for determining the upset is available. As quoted, it was only known that the upset was in nuclei density, was large, and was reproducible. Thus, the intensity and shape of the upset must be estimated. Note that in a crystallizer simulation, given a certain upset, different frameworks do not identically respond. For each crystallizer framework, an upset will give a response for that system only. Thus, once a parameter is changed, the upset might have to be re-posed.

The temperature profile for creating the nucleation upset uses step change so it is reasonable to assume the upset may be a pulse. The pulse tails off experimentally [Randolph et al [1986]]. A plot of nuclei density from the CILAS particle analyzer shows this

tailing behavior. Apparently, homogeneous nucleation first occured, followed by continuous secondary nucleation. It is impossible to distinguish these forms of nucleation; thus, for simplification the upset in the computer simulation will be described as a rectangular pulse. The intensity of the pulse will be determined by comparing the simulation results with the experimental ones.

Simulation

Simulations under the conditions discussed above were implemented with EVAX-I computer system at the University of Arizona. The indefinite parameters and pulse intensities were varied. The slopes between the crystal sizes at the upper cut size of the fines dissolver and lower cut size of the fictitiuous classifier were taken at the values of 0, 0.5 and 1. A removal rate of zero made the simulated crystallizer unstable and the solution diverged; thus, particle removal rates of zero were not adopted. Randolph, Beer, and Keener [1973] did not discuss crystallizer behavior for this value (zero), but crude parameter estimation in a KCl crystallizer does indicate the removal rate for x in (x_F, x_P) should be zero. This result is physically impossible, of course. The sensitivity criteria needs to be studied for various values of the parameters in detail, but that is beyond the scope of this study. The simulation results seemed to correspond to the experimental responses of the crystallizer as drawn in Figures 5-4(a) and 5-4(b). The parameters used for these simulations are summarized in Table 5-4.

It is evident that in each case the level of initial population density of the simulation does not correspond to that of the experiment. The reason has been already discussed, but it is mentioned again that the ideal crystal size distribution form for an R-z crystallizer was not achieved in the experiments. Previous studies [Nuttall [1971], Randolph, Beckman, and Kraljevich [1973], and Randolph, White, and Chi-Chu David [1981]] show that it is possible to establish a theoretical crystal size distribution as a pre-upset initial condition. A difference between Randolph, Chen and Tavana's experiments and previous studies is the recycle ratio of the fines dissolver, R; the ratio of the former is 13, meanwhile that of the latter is around five. As is mentioned in Chapter 3 and Chapter 4, since the higher recycle ratio of the fines dissolver narrows stability of the crystallizer, any small fluctuation including the initiation of the crystallizer in the study by Randolph, Chen, and Tavana might generate a disturbance which was relatively unstable. The fluctuation might shift the crystal size distribution from the typical R-z crystallizer. Hence, the skewness illustrated in Figure 5-2 might be avoided by reducing the recycle ratio of the fines dissolver.

It is evident that the contours of the pulse propagation by the simulation correspond to those of the experimental results except for their intensities. For the population density at $L=6.51X10^{-4}$ meters, the experimental pulse is almost identical to the simulation pulse in shape. Thus, the simulation can represent the dynamic

TABLE 5-4 Summary of parameters used for simulations

PARAMETERS	FOR BRZ=1*	FOR BRZ=2*
x_F	1	1
x_P	10	6
R	13	13
z	1	5
$d\theta$	0.05	0.05
dx	0.2	0.2

* BRZ means slope between x_F and x_P.

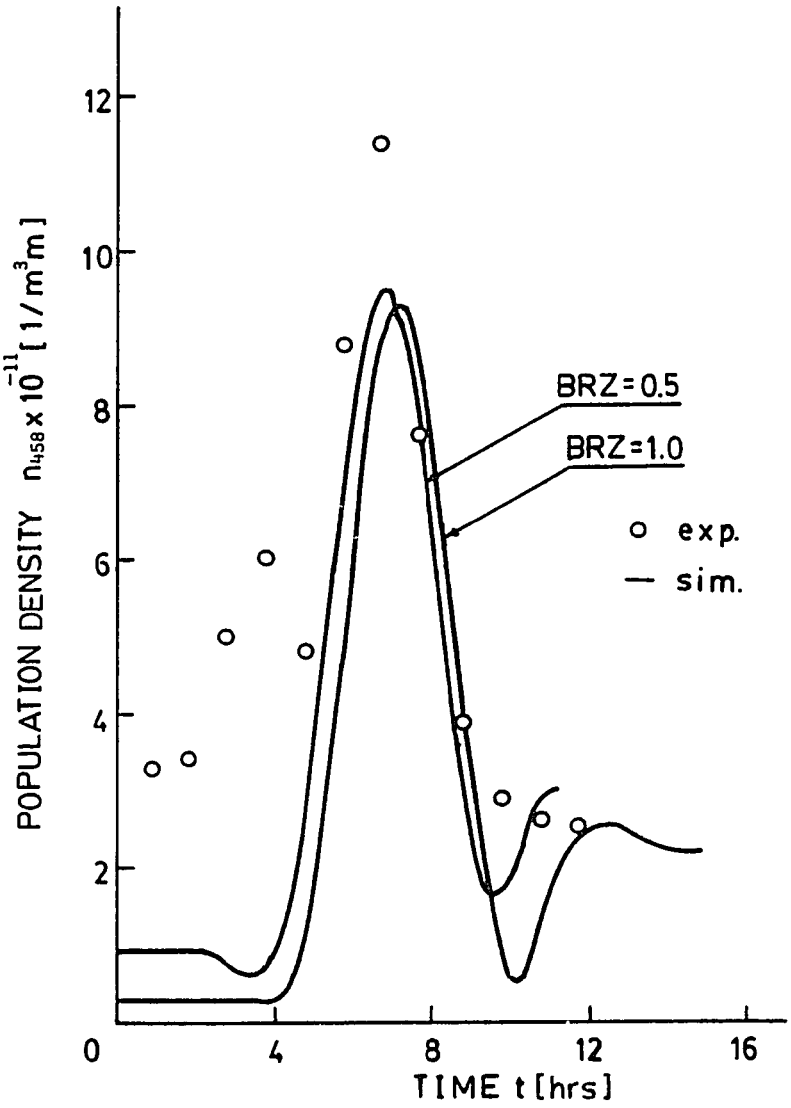


Figure 5-4(a) Experimental and theoretical population density at $L=4.58X10^{-4}$[m].

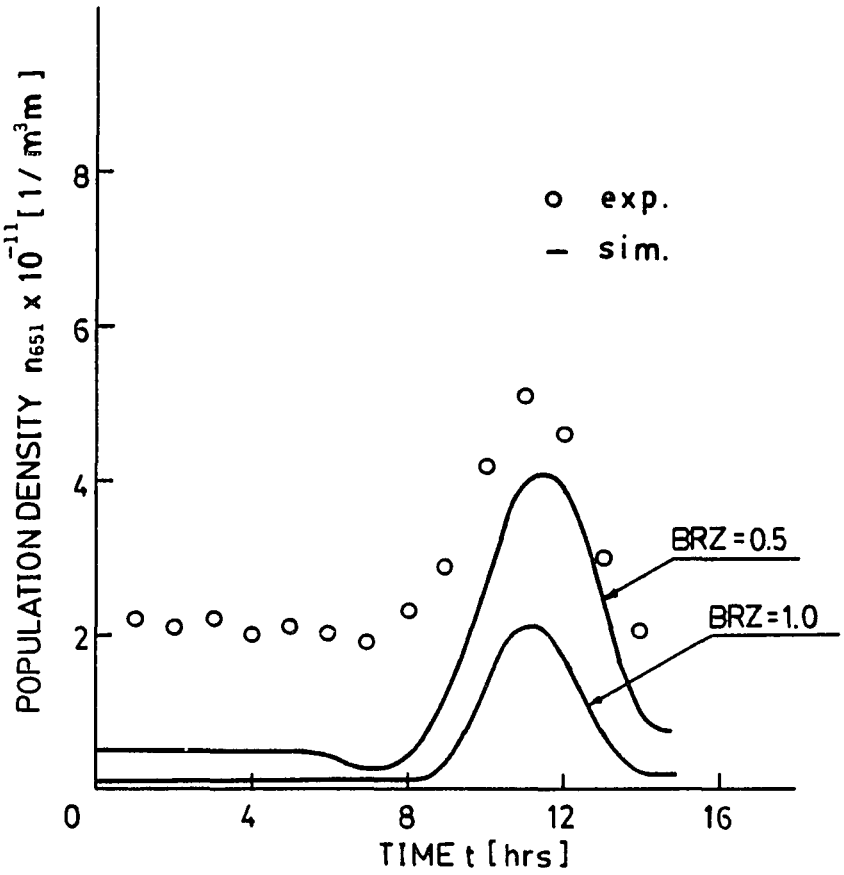


Figure 5-4(b) Experimental and theoretical population density at $L=6.51X10^{-4}$[m].

response of the R-z crystallizer. Since the nucleation kinetics used in this study are characterized as an auxiliary function, the equations for nucleation and growth rate can be easily substituted with other form. This simulation algorithm should be versatile enough for prediction of crystal size distribution dynamics in crystallizers with any configuration on any nucleation kinetics.

CHAPTER 6

CONTROLLER DESIGN FOR A CONTINUOUS CRYSTALLIZER

System theories suggest two strategies to design a controller for a continuous crystallizer. Each strategy should have significant design characteristics different from the other. Pole placement can determine a desired dynamic response of a system in response time, overshoot, and decay ratio of the system. Typical dominant poles are assigned on the lines of $z = (-1 \pm i)a$, where i indicates imaginary numbers and a is an arbitrary positive number. This method gives the configuration for dynamic response, but does not guarantee that the performance index of the system becomes minimal. On the other hand, optimization can determine an optimal controller for a given state of the variables, and the performance index will be minimal by definition. The optimal controller is not guaranteed to behave appropriately. Either method may be chosen to design the controller depending on the characteristics of a given system. As a typical example, pole assignment results in a proper dynamic response for an upset, but optimization provides a minimal performance index for a stable or quasi-stable system. Both methods are detailed as pole placement and optimization problems, respectively, in Appendix B.

In this chapter, the controller design problem for an R-z crystallizer will be discussed using both pole placement and optimization techniques. The focus in treating the controller design problem is on evaluation of characteristics of the system. Especially, an R-z crystallizer system is expressed as a multiple input-multiple output system; there is no simple index expressing characteristics of the system. Thus, the time variant system matrix A in Equation (6-1) is approximated by an invariant matrix, and hence becomes a time discrete system in order to simplify the problem. Also, a multiple pole in the characteristic matrix raises a sensitivity problem for the system. This problem will be treated by discussion of an actual controller design.

Modification of the dynamic crystallizer

The dynamic crystal size distribution is expressed in the state variable form as;

$$\underline{u}(\theta) = A(\theta)\underline{u}(\theta) + B\underline{r}(\theta) \tag{6-1}$$

and from Equations (3-13) and (3-14), the boundary condition is written as;

$$u_0 = f_3^j \left(\frac{1 + \beta f_{3,1}}{1 + \beta} \quad \frac{1}{f_2} \right)^i \tag{6-2}$$

where

$$\beta = \frac{(R-1)}{6 D_P} \int_0^{x_f} x^3 \bar{u} \, dx \qquad (6\text{-}3)$$

Matrix A is expressed by Equation (6-4).

To develop a computer algorithm to simulate the dynamic response of crystal size distribution, Equation (6-1) is integrated and then approximated by the discrete formula in Section 3. However, to develop the form for the controller design, the discrete form is instead directly modified from Equation (6-1). Equations (6-1) through (6-4) are schematically illustrated in Figure 6-1. It is evident that nucleation kinetics give complex auxiliary loops in the diagram. To delete the auxiliary subsystem, Equation (6-1) is modified into the discrete form;

$$\dot{\underline{u}}_m = A_m \underline{u}_m + B \underline{r}_m \qquad (6\text{-}5)$$

Hereafter, the subscript m is omitted simplifying the equations. Thus, Equation (6-5) is rewritten as;

$$\dot{\underline{u}} = A\underline{u} + B\underline{r} \qquad (6\text{-}6)$$

The diagram corresponding to Equation (6-6) is schematically

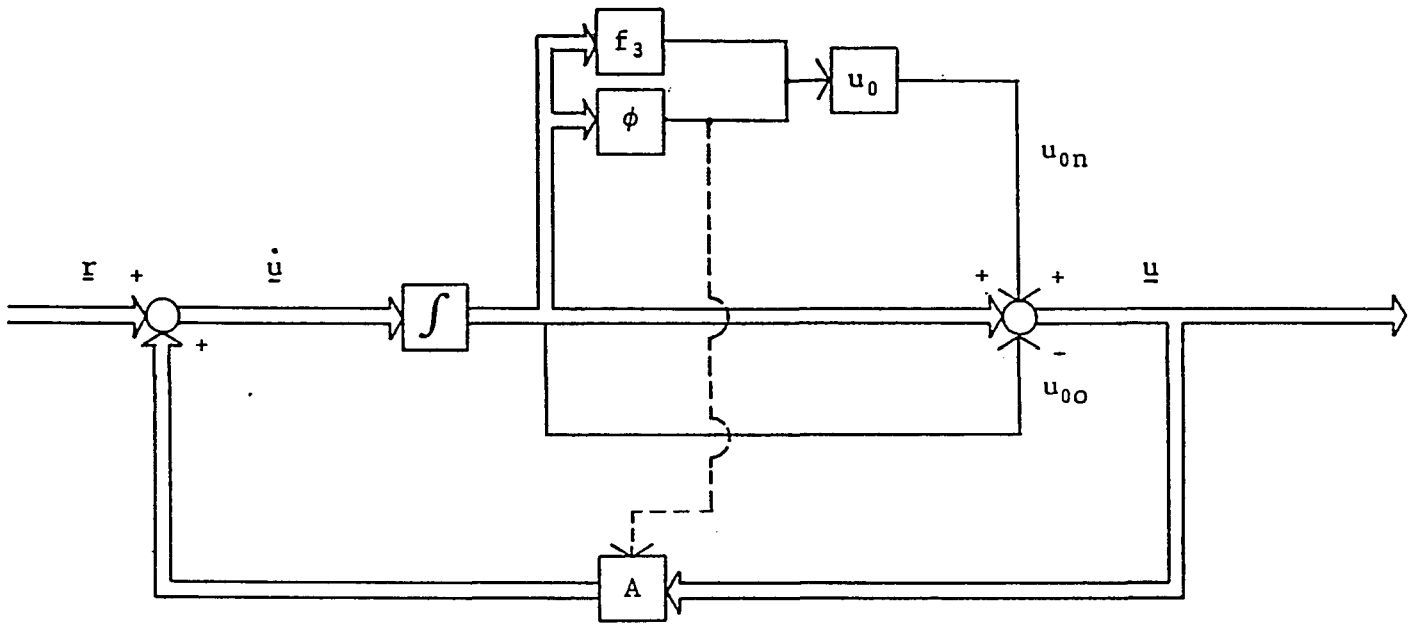


Figure 6-1 Schematic diagram of an R-z crystallizer.

illustrated in Figure 6-2, where matrix A becomes an invariant matrix. Thus, the original system is modified and expressed by invariant matrices only. Also, the complex auxiliary system is expelled from the original system, but is reserved as the reset function after each time segment. The system expressing (6-6) can be treated as a standard system.

A proportional controller can be imposed into the system expressed by Equation (6-6) and is schematically illustrated by the notation K in Figure 6-3. The controller is constant as a proportional control constant during the time increment θ, but is varied with each time segment change. This does not raise any problems when the system has an on-line controller evaluation system. The controller action is necessarily constant; thus, it would be easy to apply in most present industrial crystallizers. Looking at a continuous crystallizer system, the response to a disturbance is regarded as a bounded oscillation. The variation is not wide. Thus, it might be possible to implement a constant controller including the extreme values of the controller. First, the variation of the controller is governed by that of the dominant pole of the crystallizer system (using the pole placement technique). In a mathematical expression, from Equation (B-6) in Appendix B, the controller is expressed as;

$$K = B^{-1}(M - A) \qquad (B-6)$$

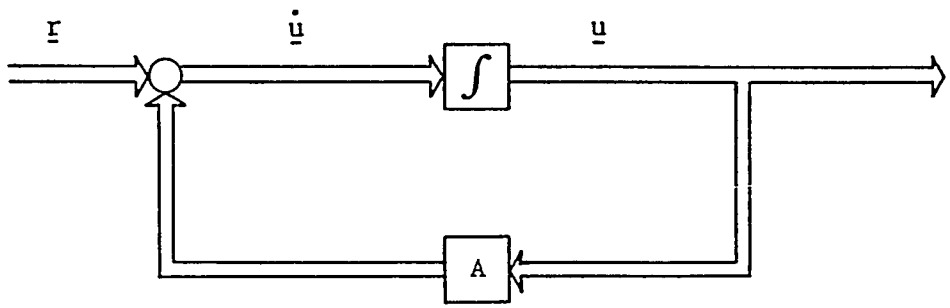


Figure 6-2 The modified diagram of the R-z crystallizer based on Equation (6-6).

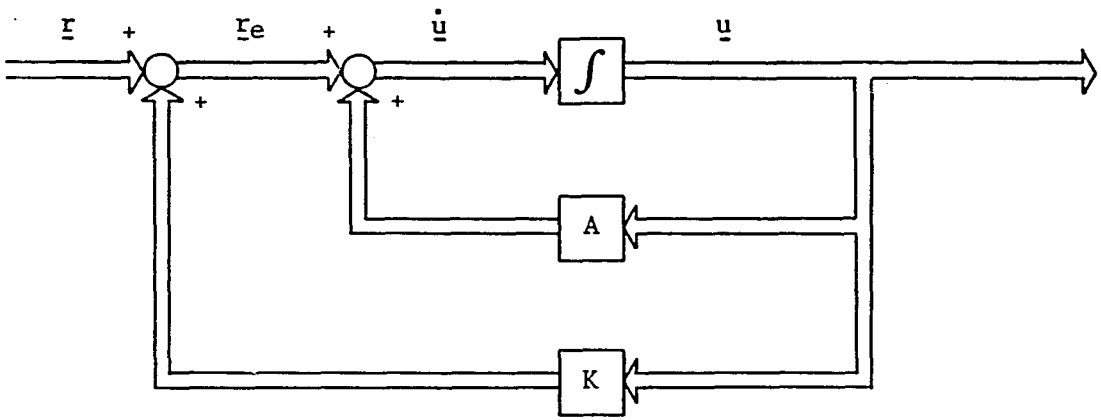


Figure 6-3 A schematic diagram of the R-z crystallizer with proportional controller K.

where matrix M includes the required poles which are fixed, and only matrix A is a parameter. Thus, controller K is a function of the dominant pole of matrix A. To evaluate fluctuations of controller K, it is necessary to obtain the upper and lower extreme dominant poles from dynamic behavior. The rest of the dominant poles are allocated between the extreme poles. Using the optimization technique, the controller K is determined from Equation (B-26);

$$0 = KA + A'K + C'QC - KBR^{-1}B'k \qquad \text{(B-26)}$$

In this equation, matrices B, C, Q, and R are invariant in time. Thus, the controller is intrinsically a function of matrix A, or a function of the dimensionless growth rate which is also bounded in the process dynamics. Consequently, the variation of controller K is determined by the range of the dimensionless growth rate change. In either case, there is no theory for determinating a value of fixed controller K. The value K must be evaluated by simulation of the system dynamics.

Previous studies assumed optimal proportional constants for the controllers to design an optimal control system. In these studies, except for optimization of stable systems, the scalar variables of controller K were assumed and then calculated using various optimization methods. Techniques using scalar controllers correspond to using a scalar k instead of matrix K in Equation (B-26); thus,

$$0 = kA + kA' + C'QC - kBR^{-1}B'k \tag{6-7}$$

or

$$0 = K_S A + A'K_S + C'QC - K_S BR^{-1}B'K_S \tag{6-8}$$

where

$$K_S = \begin{bmatrix} k & & & \\ & \ddots & & 0 \\ & & \ddots & \\ 0 & & & k \end{bmatrix} \tag{6-9}$$

Matrix K_S is an approximate solution to Equation (B-26); however, matrix K_S does not make any sense when using a state space formulation. Matrix K_S removes any effect on the state variables by other state variables in columns and rows. In other words, only the diagonal elements of the system matrices remain to evaluate the next position of the state variables. It is evident that techniques using matrix K_S are not proper for the system. Thus, in considering characteristics of crystal size distribution, it is not accurate to mathematically use matrix K_S. An example using matrix K_S will be shown later.

In the above discussion, proportional-integral or proportional-differential-integral control is not referred to. The reason is that matrix equations in general cannot be used in conjunction with integration and differentiation. This is a mathematical problem; thus, only proportional control will be treated in this study.

Modeling of a controller for an R-z crystallizer

From the feedback control system illustrated in Figure 6-3, it is seen that the state variable is a vector of population densities corresponding to crystal sizes. The population density vector is fed back as the manipulator multiplying the controller, hence, $\underline{Ku}$. Physically this means that the crystal size distribution (population density) is measured at the outlet or in the vessel, and that the measured value adjusts crystal size distribution in a feed flow by means of adding or subtracting the number of particles at particular sizes. To do so, well-classified particles and particle separators and feeders must be prepared. Such an operation is theoretically possible, but is probably not practically feasible. If two crystallizers in series are used and the crystal size distribution of the second vessel is fed back to the first vessel, then it might be possible. Even in this case, control of crystal size distribution in the first vessel remains as an unsolved problem. Besides, it is usually assumed there are no particles in the feed liquor. Thus, limitations to a feedback controller of an R-z

crystallizer must be considered. Direct and indirect variables for input to and/or output from the controller must be revised in order to test physical feasibility. In the field of crystallizer control, this kind of discussion is rare. Therefore, even though it may sound tedious, general requirements as well as physical feasibility of the controller signals will be summarized in the following discussion.

A desirable variable as an input to-outout from controller must provide the following conditions;

1) Fast transient response
2) Represention of characteristics of the system
3) Proper dynamic sensitivity and stability

One of the most important conditions in choosing the variable from the system to the controller is, of course, its response speed to system fluctuations. Slower response, including dead time, deteriorates system stability. For example, in controlling nuclei density, nuclei density might be the input variable to the controller. Nuclei density is spontaneous itself and its response is faster than other variables such as growth rate. Thus, to control crystal size distribution, population densities are a strong candidate for the input variable to the controller.

The input variable necessarily expresses characteristics of the response to fluctuations in the system. For instance, when

nucleation rate increases, the input variable must also indicate that the system turns upward. However, characteristics of an R-z crystallizer are too complicated to describe using such a simple phrase because the system is expressed by matrices. It is difficult to judge the response tendency from the system matrix; elements of a matrix indicate ascent and descent simultaneously. Thus, observation points, or observation of crystal sizes, must be specified to control the system. In the case of the R-z crystallizer, the synthesized tendency of the response to fluctuations is characterized by the third moment as is mentioned later, but this tendency does not represent characteristics of crystal size distribution.

The third point which needs to be considered is the dynamic sensitivity of the response. Since the growth rate and third moment are raised to an exponent and fed forward in order to reset the boundary condition after each time segment, those fluctuations are, respectively, amplified i and j times from the original level. Thus, powers of growth rate and the third moment, respectively, may be the most sensitive variables to system fluctuations. They certainly affect stability (the nucleation sensitivity parameter i itself is an index of a stability criterion for continuous crystallizers). If the numbers i and j are relatively large, fluctuations of nuclei density become larger. Growth rate and the third moment are stiff relative to other variables; thus, they are still candidates for input variables to the controller. Thus, the

above three conditions must be checked when designing a crystallizer controller.

Consideration of physical conditions for these variables will be discussed. The variables which are candidates for input to and output from the controller and which have been used as the controller parameters in previous studies are listed as follows:

A. Controller input (from the system to the controller)

1) Population densities, including nuclei density (or crystal size distribution).
2) Growth rate.
3) Moments.
4) Mixed product rate
5) Solute concentration.

B. Controller output, or manipulation variables (from the controller to the system)

1) Population densities including nuclei density, (or crystal size distribution).
2) Recycle ratios of the fines dissolver and the product classifier.
3) Crystal sizes at the upper cut size of the fines dissolver and the lower cut size of the product classifier.
4) Solute concentration in feed liquor.
5) Temperature in feed liqour or the vessel.

The above input variables can be distinguished between physically observable and physically non-observable. The former group includes population densities, mixed product rate, and solute concentration. The latter group includes the other variables plus nuclei density, which are calculated using the former. Some of the variables belonging to the former group are not appropriate as input variables. Solute concentration may be excluded from the candidates, for mass conservation in a crystallizer system does not consist of solute concentration only, and it is difficult to predict the relationship between supersaturation and nuclei density. Thus, solute concentration does not contain sufficient information to adjust the controller. Next, the mixed product rate is not a proper input variable to the controller because the rate indicates the total production rate and does not characterize crystal size distribution. The rate must be used for controlling the total production rate. On the other hand, population densities characterize crystal size distribution and are measureable variables, at least in the smaller size ranges, Thus, the measureable range of crystal size and the number of the measureable points are limited. As long as a full order observer is used, this limitation deteriorates the controller design algorithm. Using the theorem of a minimum order Luenberger observer, a partial observation of state variables still satisfies the controller design alogrithm (Appendix B). Consequently, population densities in the smaller size ranges are unique measurable variables which might be used as inputs to the controller.

Nuclei density, growth rate, and the third moment are indirectly observed from population densities and include characteristic information of the crystallizer system. As was mentioned before, observation points must be specified based on the objective of control; thus, those that include characteristics of crystal size distribution intrinsically are advantageous to reduce the number of inputs to the controller. The relationship among nuclei density, growth rate, and the third moment is described in Equation (6-2); nuclei density is a function of growth rate and third moment. Thus, nuclei density involves characteristics of growth rate and the third moment. In other words, nuclei density involves mass conservation of solute as growth rate, and the total integration of crystal size distribution as the third moment. Also, nuclei density is a boundary condition, and hence one of the state variables. Consequently, nuclei density and population densities are chosen for input variables to the controller.

The manipulation variables listed include those that must be set points of an R-z crystallizer. The crystal size at the lower cut size and recycle ratio of the product classifier as well as the temperature in the feed liquor and vessel must be constant. The crystal size at the lower cut size and recycle ratio of the product classifier determine product quality. Thus, these parameters are set points of the crystallizer system. Next, temperature is seldom changed in order to control a continuous crystallizer. Temperature determines nucleation kinetics, and hence the fundamental constants

of the system. Particularly because solubility and supersaturation are strongly expressed as functions of temperature, changing temperature changes the physical constants of the system. Hence, temperature must be excluded from the candidates for manipulative variables. In addition to these variables, solute concentration cannot be a variable because supersaturation cannot be controlled, as mentioned above.

The remaining variables are population densities, recycle ratio of the fines dissolver, and the crystal size at the upper cut size of the fines dissolver. Among these, the first and second variables are equivalent in their effect on crystal size distribution. The dissolved mass of the fines is proportional to the recycle ratio. Also, the effect of population densities at the larger sizes on nuclei density is probably negligible compared to the effect of population densities at a size smaller than the crystal size at the cut size of the fines dissolver. The target for controlling the crystallizer system is regulation of the nuclei and smaller crystal size population densities; thus, the recycle ratio may be regarded as equivalent in effect to population density and hence a pseudo state variable. On the other hand, the crystal size at the upper cut size of fines dissolver, L_F (or in dimensionless form x_F) is not favorable as a manipulative variable. Varying the crystal cut size changes the characteristics of system matrix A in Equation (6-1) or (6-6). Crystal size distribution is represented using a non-continuous function h(x), and the break points are the

crystal sizes at the upper cut size of the fines dissolver x_F and at the lower cut size of the product classifier, x_p. These break points govern mathematical characteristics of matrix A; namely, the eigen values of the matrix. Variation of the break points drastically shifts the allocation of the system poles and hence the dominant poles. This causes not only a deterioration of the numerical credibility of the discrete algorithm expressed by Equation (6-6) but also renders controller capability meaningless, because the controller is designed based on the dominant poles. In other words, variation of the break points during operation mathematically means that non-continuity of the crystal size distribution is brought into the time sequence. Variation of the crystal size at the upper cut size of the fines dissolver may be experimentally and analytically fragmentary. Hence, crystal size should not be the manipulation variable. Consequently, the most favorable analytical manipulation variable is the recycle ratio of the fines dissolver.

As a summary of the above discussion, the controller input and output variables proposed are tabulated in Table 6-1.

TABLE 6-1 Summary of controller input and output variables

objective	variables
Controller input	specified population densities including nuclei density as estimated from population densities at a finite size
Controller output	recycle ratio of the fines dissolver

Discussion

The experimental design of a control study which was done by Randolph et al [1986], described in Chapter 5, will again be summarized here. The fundamental experimental procedure was the same with or without control.

The control study was implemented using a program in a computer (HP-85) which used the current infered value of n^o to calculate the updated set points of Q_F for the next sample period, as per Equation (6-10);

$$\frac{Q_F - \bar{Q}_F}{\bar{Q}_F} = K_c \frac{n^o - \bar{n}^o}{\bar{n}^o} \qquad (6\text{-}10)$$

A three point running time average of n^o was used in the control algorithm. Thus,

$$\bar{n}_t{}^o = \frac{\bar{n}_{t-2}{}^o + 2\bar{n}_{t-1}{}^o + n_t{}^o}{4} \qquad (6\text{-}11)$$

where "-" represents the time average and subscript t represents the value at sample period t (Note: the sample period here means the sample hold time). A sampled data time of 10 minutes was used,

which permitted manual adjustment of the fines dissolver flow Q_F. A sampled data time as low as one minute would be possible with the system if Q_F were adjusted automatically. The sampled data time of 10 minutes was still short compared to the product residence time =180-240 minutes. An important aspect of the experimental apparatus was that the fines dissolver stream was adjusted by recycle of a portion directly back to the crystallizer. The purpose of this recycle was to vary Q_F without changing the cut size of the fines being removed from the crystallizer.

For the above crystallizer, one needs to test controllability, observability, and the pole allocations of the system before attempting controller design. Controllability is evident from Figure 6-2; all of the state variables are theoretically controllable because matrix A is regular. Also, it is dual for observability of the system. A detailed theory of controllability and observability can be found elsewhere (Appendix B). Next, the pole allocations will be characterized. As mentioned before, the characteristic matrix A of the system is not invariant with time, but the time discrete formula is applicable to system theories. Furthermore, growth rate variations for an R-z crystallizer are stiff relative to fluctuations of other variables, which means that the width and cycle period of growth rate oscillation are narrow and long, respectively. The order of growth rate change is much smaller than the order of other variables in matrix A. Growth rate change

is of the order of 0.1, while the order of the function h(x) is 1 and more. Thus, growth rate may be fixed equal to unity for controller design. Consequently, the pole allocations are calculated using MATLAB, a package program for matrix calculations [Moler, C. [1980]]. Eigen values of matrix A for various recycle ratios of the fines dissolver are tabulated in Table 6-2.

At present, MATLAB is the most convenient package program to obtain numerical answers to matrix problems. The numerical credibility of the calculations is guaranteed only up to 10X10 matrices. Larger matrices can not be implemented without numerical and truncation errors [Forsythe, Malcolm, Moller [1977]]. A major interest in crystallizer systems is the fluctuation of nucleation and its propagation behavior. Thus, to demonstrate crystal size distribution control techniques for a crystallizer, the five dimensionless crystal sizes are used in each calculation. As QR-decomposition is used for the matrices, the original matrix size must be one half of the maximum allowable size for MATLAB.

Table 6-2 shows that overshoot decreases with an increase of the recycle ratio of the fines dissolver; the ratio of the real part to the imaginary part for one of the dominant pole represents the intensity of overshoot. This means that a crystallizer with a large recycle ratio of the fines dissolver is overdamped. The poles for this system gather at a point so the crystallizer is ill-posed. If the model accurately represents the physical system, this means that the real system may become sensitive around this operating point. A

TABLE 6-2 Eigen values for various R.

R	Eigen values
1	-0.9999 + 3.3333I
	-1.1535 + 0.1917I
	-1.1854 - 0.1562I
	-0.8114 + 0.1472I
	-0.8497 - 0.1826I
	-1.0000 - 3.3333I
5	-5.0000 + 3.3334I
	-5.1812 + 0.0886I
	-4.9078 + 0.1756I
	-5.0831 - 0.1778I
	-4.8279 - 0.0864I
	-5.0000 - 3.3333I
9	-8.9999 + 3.3334I
	-9.0463 + 0.2186I
	-9.2159 - 0.0411I
	-8.9614 - 0.2203I
	-8.7764 + 0.0428I
	-9.0000 - 3.3333I
13	-13.0000 + 3.3334I
	-13.1761 + 0.1760I
	-12.8258 + 0.1715I
	-13.1727 - 0.1778I
	-12.8254 - 0.1697I
	-13.0000 - 3.3334I

crystallizer with a small fines dissolver ratio has a large overshoot relative to one with a large recycle ratio. This means that oscillations last longer and are slow to reach the response time. Thus, a crystallizer system with a large fines dissolver ratio is stiff because of the large absolute real numbers of these dominant poles. The system is stable but it is hard to analyze the system for sensitivity. The controller poles should be separated. In the case of a small fines dissolver ratio, overshoot and response time must be minimized by controller design. For each case, the pole placement technique is a more convenient way to design the controller than using optimization because the poles can be assigned freely in order to separate the original pole allocation. In either case (i.e. high or low fines dissolving ratio R), the pole placement techniques are recommended for controller design. An example of controller design using pole placement will be shown at the end of this chapter.

The concept of the pole placement method is totally different from that of the optimization techniques. Pole placement assigns the controller poles desired, and then the controller is designed. Surprisingly the pole placement technique has not been used in many previous studies. Optimization techniques are still the most popular approach, as discussed below.

Before starting the discussion, it is desireable to have an insight into the actual crystallizer. The crystallizer model can be theoritically described as a multiple input-multiple output system,

and is evidently controllable and observable. Those conditions are also true for the actual crystallizer system; one can observe crystal size distribution through the whole range with suitable instruments or experimental procedures. This fact provides a full order observer. However, only one physical manipulation device, a fines dissolver, exists for most crystallizer configurations. This means that only one circuit of the signal receptor for the controller output is available in the crystallizer. It is physically possible, but impractical, to install multiple size range dissolvers to realize the full theoretical approach of controller design. It would be folly to dissolve crystals over the entire size range so this approach is not feasible in industry. Although multiple dissolvers are assumed and the signal flow diagram is completely constructed, a discussion of controller design must be limited in light of the fact that only a single dissolver is feasible.

There is no theorem for controller design of a single manipulator from a multiple output controller. To averge or characterize the output, it is supposed that an arithmetic, geometric, or some other average methods are used. However, in the case of the R-z crystallizer, irregular nucleation causes considerable fluctuations in the system. Hence, the characteristics of system control are specifically concentrated on the dynamics of population density at zero size and its neighbors. Of course, propagation of fluctuations moves from zero size to the larger crystal sizes. Backward propagation of the waves does not occur.

The effect of no back-propagation is quantitatively explained by numerical calculation of Equation (6-1). As represented in Appendix A, population density $\underline{u}$ is affected by four others: u_{l-2}, u_{l-1}, u_{l+1}, and u_{l+2}. But, for the region of crystal sizes smaller than x_F, the ratio of u_l/u_{l+1} is

$$\frac{u_l}{u_{l+1}} = \frac{e^{-Rx_l}}{e^{-R(x_l + \Delta x)}} = e^{R(\Delta x)} \tag{6-12}$$

Since the elements of matrix A are symmetrical except for both ends of the elements, the larger crystal size effect is $e^{-2R(\Delta x)}$ times the smaller crystal size effect. Therefore, population densities at the small crystal sizes may be chosen as input variables to the controller which would then drive the fines dissolver ratio. If it is necessary to choose one input variable, it should be nuclei density. The credibility of this discussion needs to be proved by experiment, but it will be shown later that the above controller assumption is proper.

Optimal values of the controller for various fines dissolver ratios with nuclei density as input and the recycle ratio as output using Equations (B-28) through (B-31) are drawn in Figure 6-4. MATLAB was used for these calculations. It is evident that the amount of control action decreases with an increase in the fines dissolving ratio. This means that a large fines dissolver ratio

narrows the range of stability; the system becomes non-robust. This supports the statement that a large fines dissolving ratio narrows the stability limits of the system [Randolph, et al [1973]]. Note that the figure is constructed using one-dimensional optimization on nuclei density for the R-z crystallizer. An optimal controller using multiple population densities, including nuclei density, is expressed in matrices and corresponds to the characteristic matrices in Table 6-1. These matrices are tabulated in Table 6-3. It is evident from Table 6-3 that the elements in the first row and column correlate with the variation illustrated in Figure 6-4. There is no theoretical correlation between elements of the matrix and the scalar values, for an element of a matrix cannot represent the matrix. Thus, the optimal values illustrated in Figure 6-4 must be understood as the indices which represent a correlation of the characteristics of the crystallizer.

An experimental R-z crystallizer was simulated with various controllers (as defined by differing proportional constants). These simulation results are schematically illustrated in Figures 6-5(a), (b), and (c). Figures 6-5(a) and (b) show population densities at dimensionless crystal sizes x=2.6 and x=3.8 that correspond to crystal sizes $L=4.58\times10^{-4}$ meters and $L=6.51\times10^{-4}$ meters, respectively. Figure 6-5(c) shows the normalized third moment. The controller gain was assigned values 0.2, 0.5, and 1.0, respectively. The controller algorithm in the simulation was identical to Equations (6-10) and (6-11), but sample hold time was

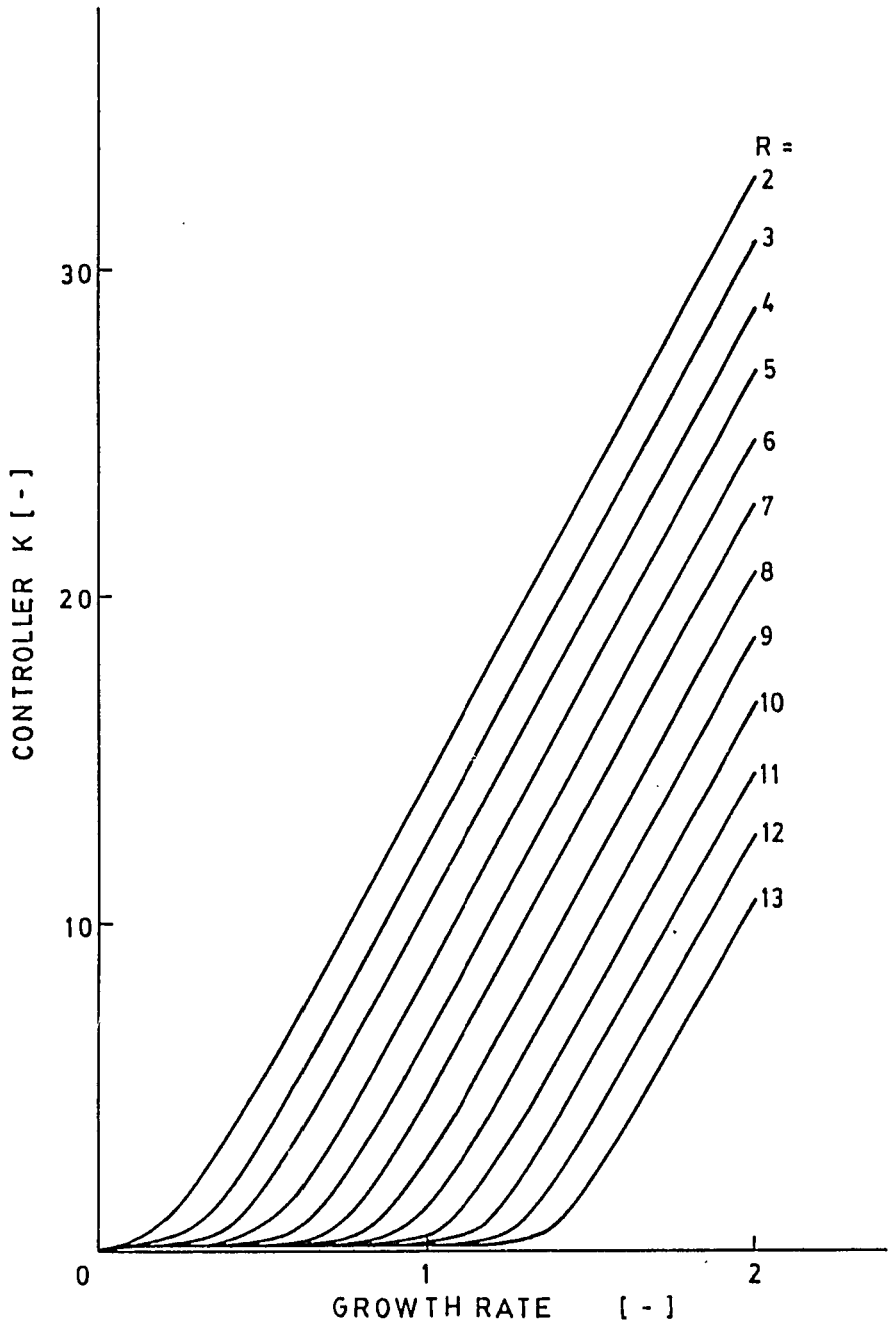


Figure 6-4 Optimal controllers to growth rate change for various recycle ratio R, where x_F=1.0, x_P=6, i=3, and j=1.

TABLE 6-3(a) Optimal controller K for various R.

R	Controller K
1	COLUMNS 1 THRU 4 2.9395 - 0.0000I -3.9386 + 0.0000I 2.8963 - 0.0000I -0.9561 + 0.0000I -3.9386 + 0.0000I 7.1633 - 0.0000I -5.5140 + 0.0000I 2.1675 - 0.0000I 2.8968 - 0.0000I -5.5140 + 0.0000I 5.2992 - 0.0000I -2.2951 + 0.0000I -0.9561 + 0.0000I 2.1675 - 0.0000I -2.2951 + 0.0000I 1.0653 - 0.0000I 0.0980 - 0.0000I -0.3905 + 0.0000I 0.5692 - 0.0000I -0.4739 + 0.0000I 0.0005 + 0.0000I 0.0397 - 0.0000I -0.0805 + 0.0000I 0.0765 - 0.0000I COLUMNS 5 THRU 6 0.0980 - 0.0000I 0.0005 + 0.0000I -0.3905 - 0.0000I 0.0396 - 0.0000I 0.5692 + 0.0000I -0.0805 + 0.0000I -0.4739 - 0.0000I 0.0765 - 0.0000I 0.4416 + 0.0000I -0.0212 + 0.0000I -0.0212 + 0.0000I 0.0549 - 0.0000I
5	COLUMNS 1 THRU 4 0.7098 - 0.0000I -1.0570 + 0.0000I 0.9109 + 0.0000I -0.4131 - 0.0000I -1.0570 + 0.0000I 2.0787 - 0.0000I -1.8062 - 0.0000I 0.8820 + 0.0000I 0.9109 - 0.0000I -1.8062 + 0.0000I 1.8103 + 0.0000I -0.8763 - 0.0000I -0.4131 + 0.0000I 0.8820 + 0.0000I -0.8763 - 0.0000I 0.5772 + 0.0000I 0.1044 - 0.0000I -0.2463 + 0.0000I 0.2590 - 0.0000I -0.1544 - 0.0000I -0.0160 - 0.0000I 0.0405 + 0.0000I -0.0427 - 0.0000I 0.0199 + 0.0000I COLUMNS 5 THRU 6 0.1044 + 0.0000I -0.0160 - 0.0000I -0.2463 - 0.0000I 0.0405 + 0.0000I 0.2590 + 0.0000I -0.0427 - 0.0000I -0.1544 - 0.0000I 0.0199 + 0.0000I 0.1509 + 0.0000I 0.0066 + 0.0000I 0.0068 - 0.0000I 0.0349 + 0.0000I

TABLE 6-3(b) Optimal controller K for various R (continued).

R	Controller K
9	COLUMNS 1 THRU 4 0.2115 - 0.0000I -0.2326 + 0.0000I 0.1862 - 0.0000I -0.0785 + 0.0000I -0.2326 + 0.0000I 0.4213 - 0.0000I -0.3062 + 0.0000I 0.1376 - 0.0000I 0.1862 - 0.0000I -0.3062 + 0.0000I 0.3181 - 0.0000I -0.1222 + 0.0000I -0.0785 + 0.0000I 0.1378 - 0.0000I -0.1222 + 0.0000I 0.1163 - 0.0000I 0.0182 + 0.0000I -0.0348 - 0.0000I 0.0321 + 0.0000I -0.0185 - 0.0000I -0.0027 + 0.0000I 0.0056 - 0.0000I -0.0044 + 0.0000I -0.0018 - 0.0000I COLUMNS 5 THRU 6 0.0182 - 0.0000I -0.0027 + 0.0000I -0.0348 + 0.0000I 0.0056 - 0.0000I 0.0321 - 0.0000I -0.0044 + 0.0000I -0.0185 + 0.0000I -0.0018 - 0.0000I 0.0625 - 0.0000I 0.0098 + 0.0000I 0.0098 + 0.0000I 0.0266 + 0.0000I
13	COLUMNS 1 THRU 4 0.0933 + 0.0000I -0.0683 - 0.0000I 0.0490 + 0.0000I -0.0180 - 0.0000I -0.0683 - 0.0000I 0.1232 + 0.0000I -0.0625 + 0.0000I 0.0244 - 0.0000I 0.0490 + 0.0000I -0.0625 - 0.0000I 0.0351 + 0.0000I -0.0190 - 0.0000I -0.0180 - 0.0000I 0.0244 + 0.0000I -0.0190 - 0.0000I 0.0476 - 0.0000I 0.0032 + 0.0000I -0.0049 + 0.0000I 0.0040 - 0.0000I -0.0030 + 0.0000I -0.0004 - 0.0000I 0.0008 + 0.0000I 0.0000 - 0.0000I -0.0029 + 0.0000I COLUMNS 5 THRU 6 0.0032 + 0.0000I -0.0004 - 0.0000I -0.0049 + 0.0000I 0.0008 - 0.0000I 0.0040 + 0.0000I 0.0000 - 0.0000I -0.0030 + 0.0000I -0.0029 - 0.0000I 0.0399 - 0.0000I 0.0072 - 0.0000I 0.0072 - 0.0000I 0.0219 + 0.0000I

TABLE 6-4 Matrices Q and R used for calculations.

Matrices	Elements					
R	1	0	0	0	0	0
	0	1	0	0	0	0
	0	0	1	0	0	0
	0	0	0	1	0	0
	0	0	0	0	1	0
	0	0	0	0	0	1
Q	1	0	0	0	0	0
	0	1	0	0	0	0
	0	0	1	0	0	0
	0	0	0	1	0	0
	0	0	0	0	1	0
	0	0	0	0	0	1

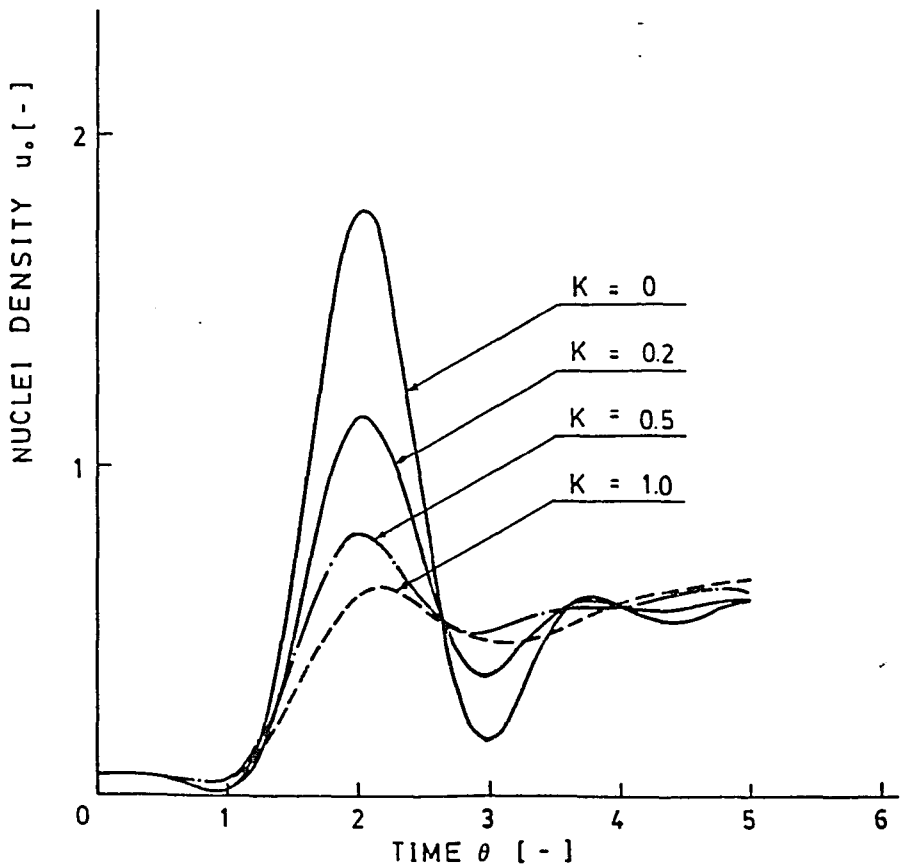


Figure 6-5(a) Variation of response in population density at x=2.6 to an upset for various proportional controllers.

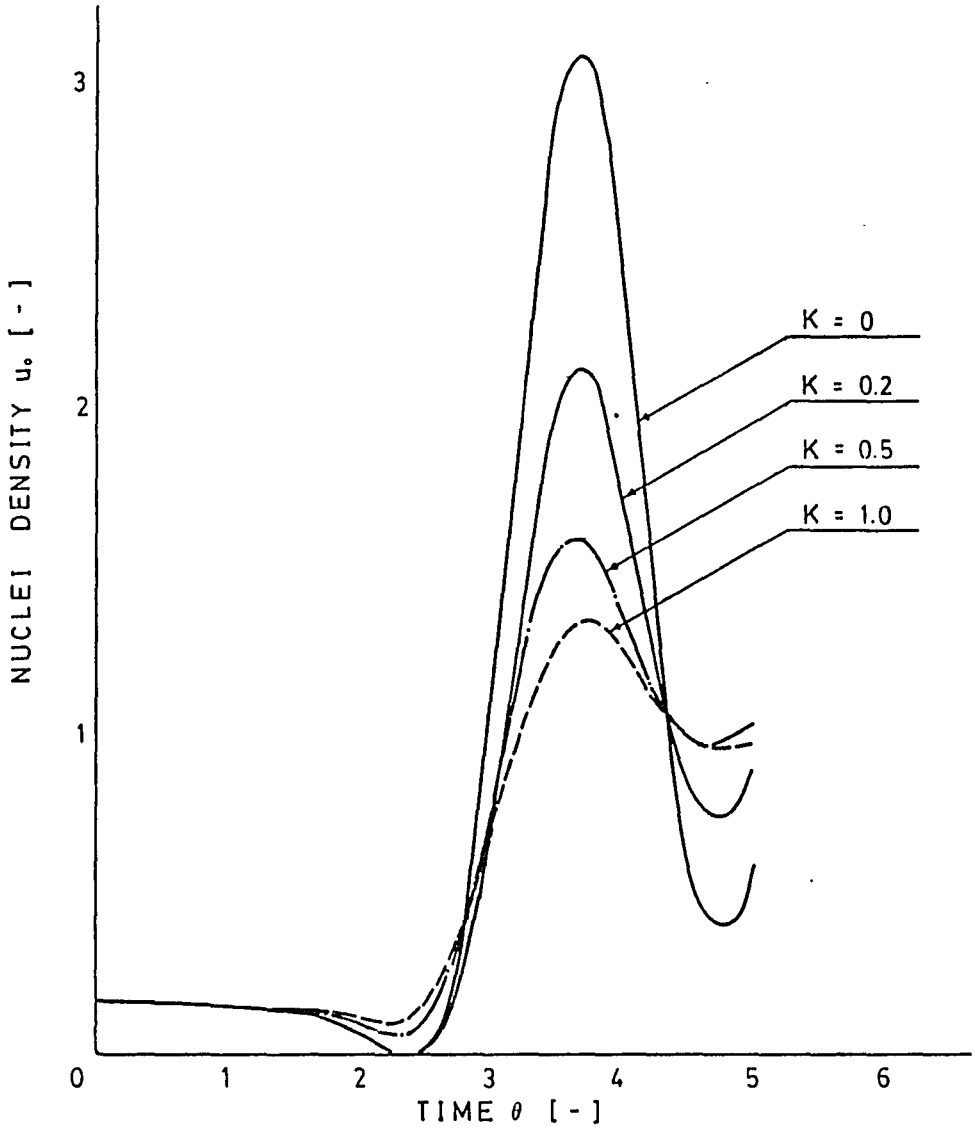


Figure 6-5(b) Variation of response in population density at x=3.8 to an upset for various proportional controllers.

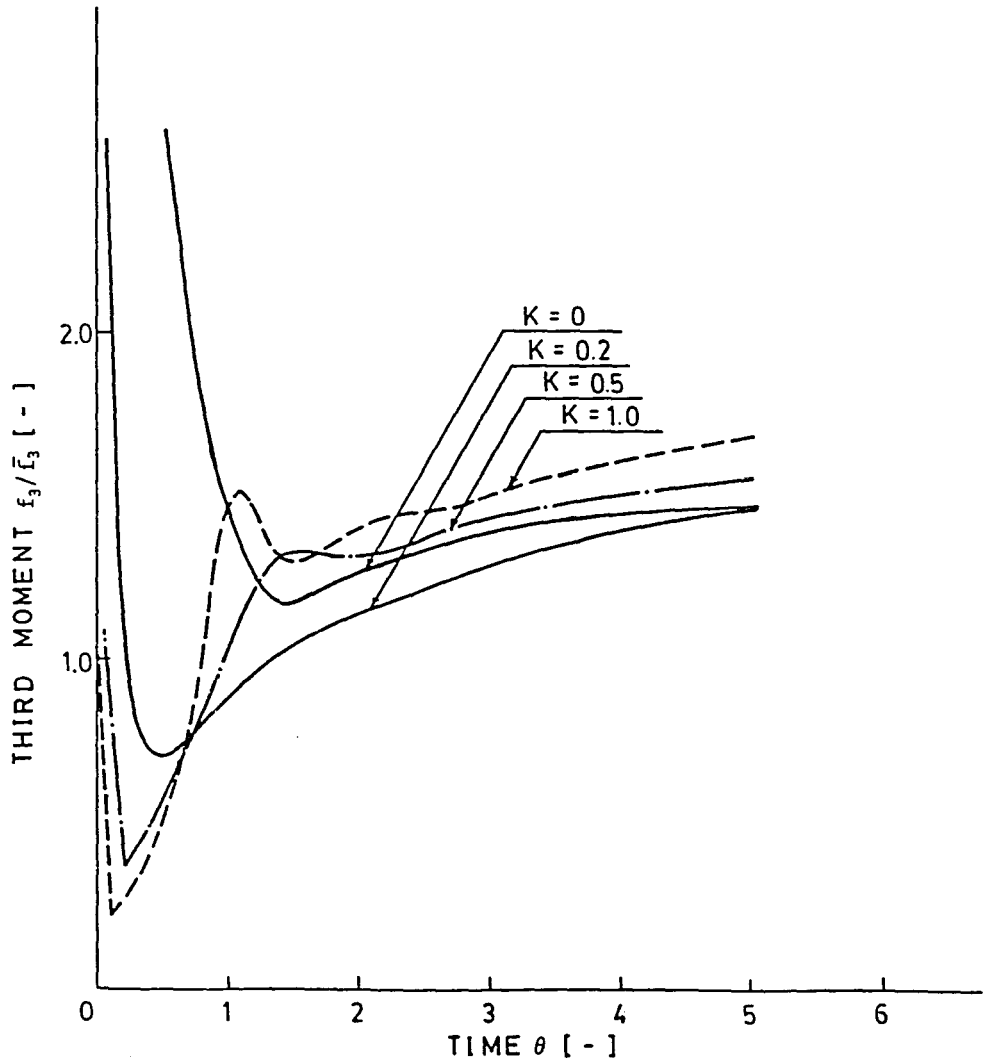


Figure 6-5(c) Variation of response in the third moment to an upset for various proportional controllers.

set by a dimensionless time θ=0.05 corresponding to 9 minutes in real time. This amount of time is approximately equal to the sample hold time of 10 minutes in the experimental study.

From these simulation, it is evident that a larger value of the controller gain reduces the magnitude of the disturbance pulse, but the controller does not stabilize the system as well, compared to the smaller controller gain. Synthesizing these three graphs, it seems that a controller of K=0.5 is quite effective in reducing the response time and fluctuations in population density at x=2.6 (or $L=4.58\times10^{-4}$ meters). From Figure 6-4, the optimal controller gain for R=13 might be 0.1 or less. The steady state recycle ratio of the laboratery fines dissolver was equal to 13, but decreases to around 11 after the initial upset as indicated by the control algorithm of Equations (6-6) and (6-7) (Figure 6-6). For R=11 and ϕ=1, an optimal controller gain of 0.5 is obtained from Figure 6-4. The overall optimal controller gain for the R-z crystallizer with fines dissolving as the manipulated variable becomes approximately K=0.5. It is again emphasized that calculation of the optimum controller does not provide proper pole assignment. This means that the system poles are possibly ill-posed, even though the controller is optimized.

An improper optimization is presented in Figure 6-7. The algorithm of Equation (B-26) using a unique value K_s as per Equation (6-7) for diagonal elements but zeros for all others corresponds essentially to the optimization technique discussed

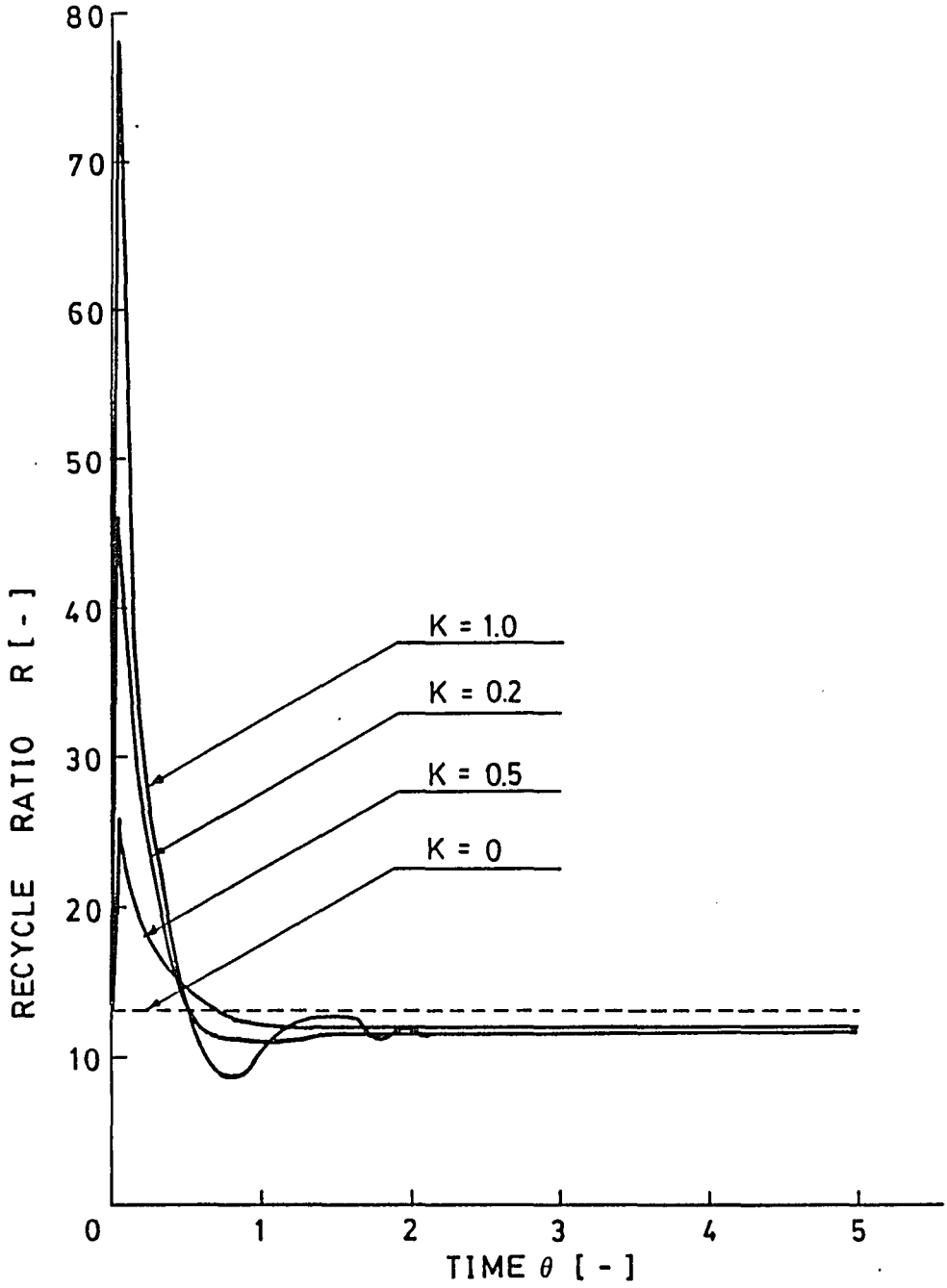


Figure 6-6 Variation of recycle ratio R to time, where x_F=1.0, x_p=3.0, z=5, i=3, and j=1.

previously, but without consideration of pole allocation in the system. It is evident that it is impossible to obtain any optimal controller using a minimal performance index based on population density. It is concluded that conventional methods of optimization with a scalar controller are not applicable for an R-z crystallizer if the ultimate object of control is controlling crystal size distribution.

The synthesized figures of the experimental and simulational results are drawn in Figures 6-8(a) and (b).

A flow diagram describing a controller and the minimum order Luenberger observer is schematically illustrated in Figure 6-9, where the pole placement technique is used. The observer tracks the state variables reconstructed in the model to real ones. In the flow diagram, a few adjoining population densities as well are measured. This measurement permits the real output and nuclei density to be calculated (estimated). Advantages of using this system are summarized as follows. First, the controller poles can be chosen freely as mentioned above. Additionally, the use of a few population densities as calibration variables to the reconstructed model eliminates measurement of population densities at all sizes. [Such measurement would be impractical with current on-line sizing instruments.] An effective dummy crystal size distribution is numerically calculated in the model. Examples of controller design with the minimum order Luenberger observer can be found elsewhere [e.g. Kailath [1981], and Cellier [1985]]. Note that the above

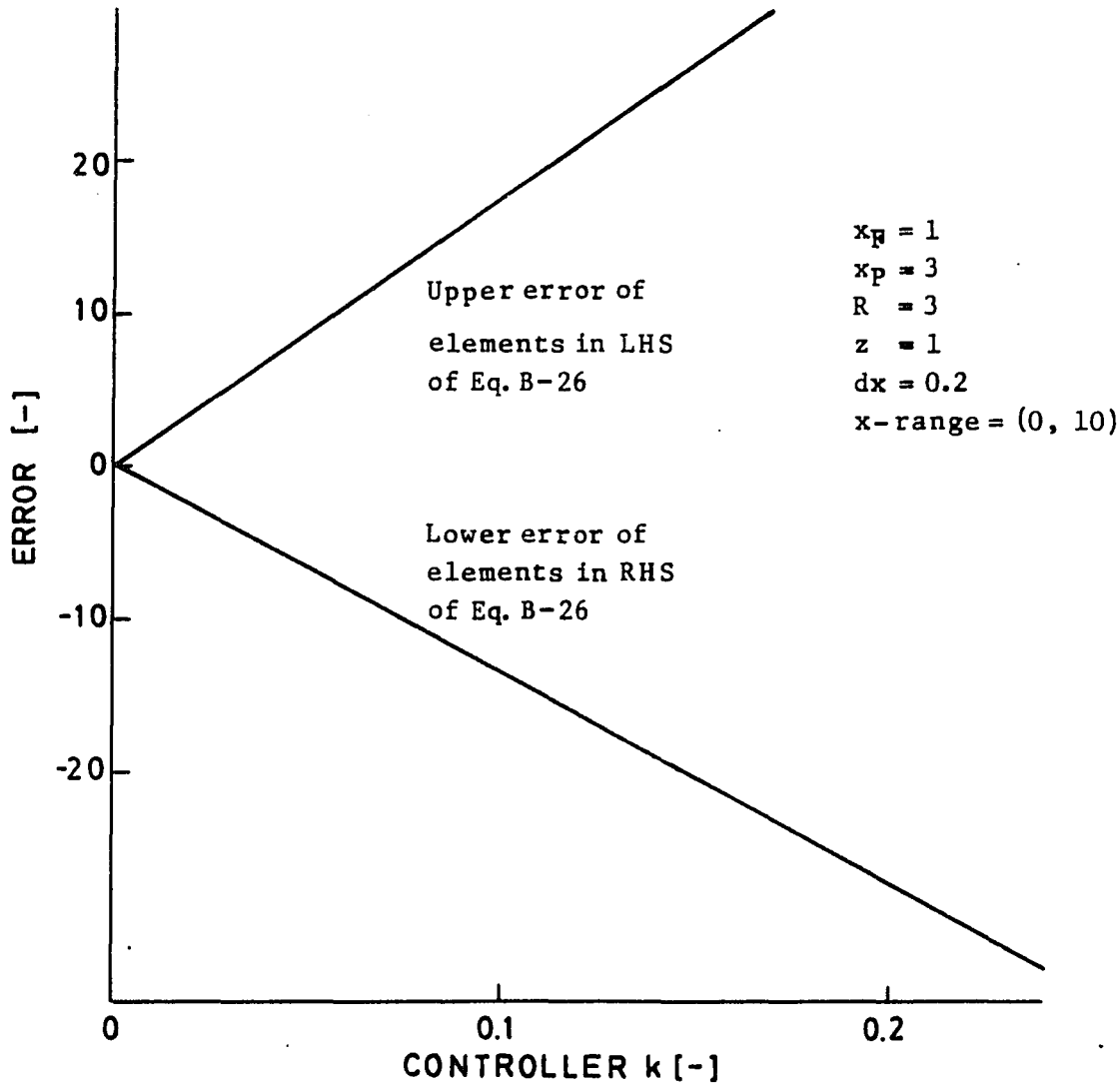


Figure 6-7 An example of improper optimization using matrix K_s, where its unique diagonal element k is equal to 0.5.

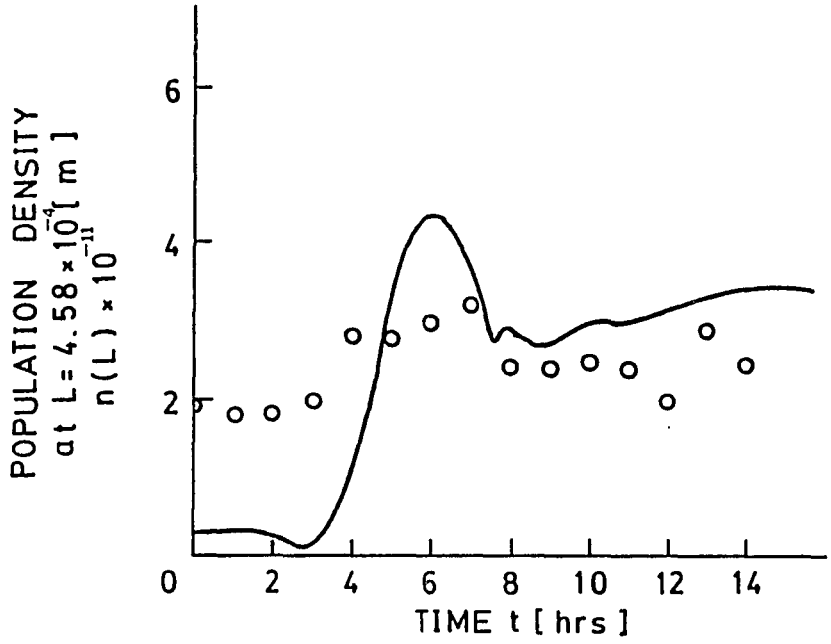


Figure 6-8 Experimental and theoretical population density at $L=4.58X10^{-4}$[m] with controller K=0.5.

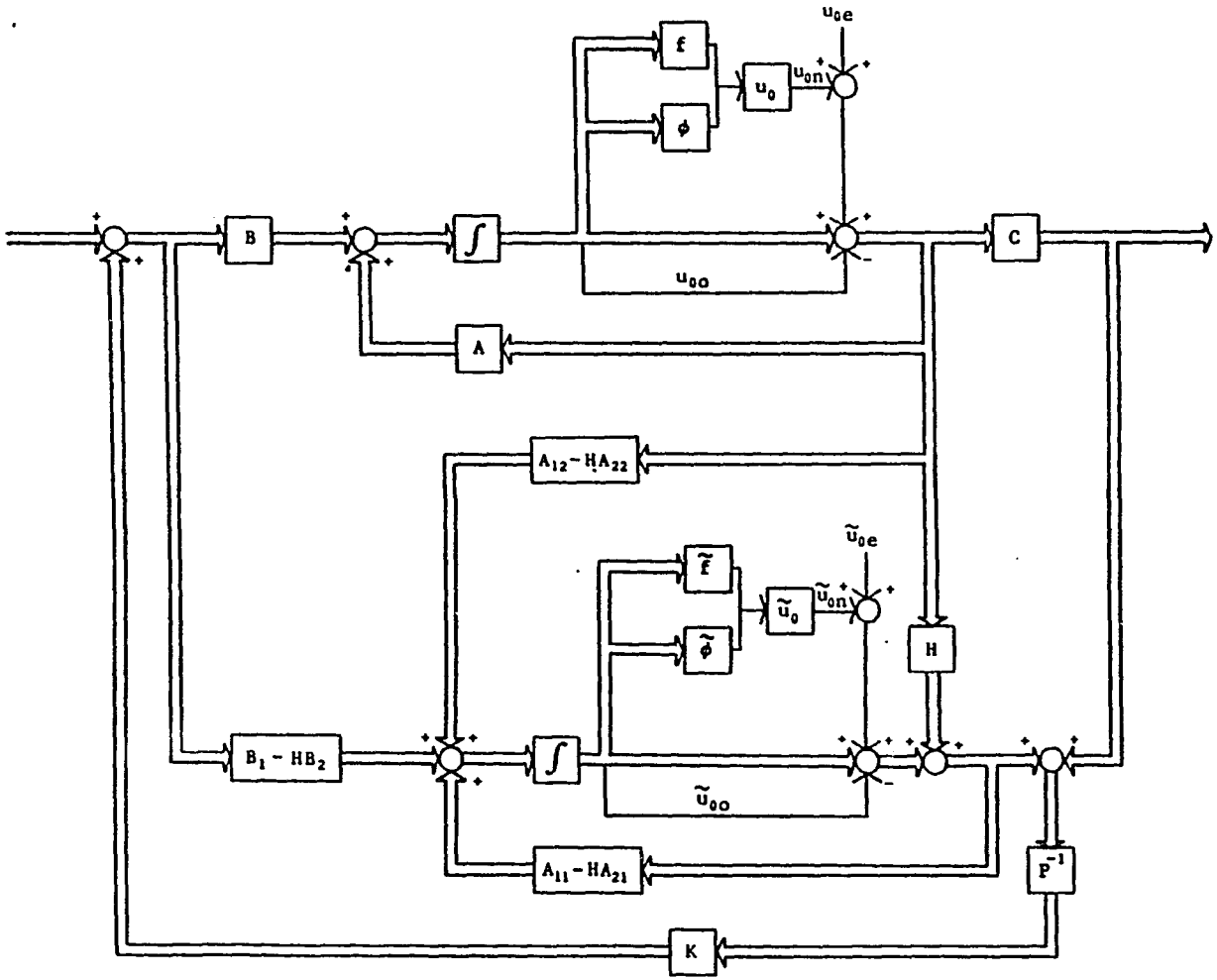


Figure 6-9 Schematic flow diagram of an R-z crystallizer with a proportional controller and the minimum order Luenberger observer.

conception of controller design has already been used for a program, "SYUJI II", which was used in the KCl control study described previously [Randolph, et al [1986]] based on slurry density as a state variable instead of crystal size distribution. It is necessry to further study how to treat the difference in the number of variables between the controller-input-output and the manipulator, but the flow diagram in Figure 6-9 should be one of the best systems to control crystal size distribution.

An example of controller design for an MIMO system (Cellier [1985]) is summarized below.

Example of controller design for an MIMO system

A simple example of controller design for a multiple input-multiple output system will be presented. According to Cellier (Cellier [1985]), QR-decomposition (orthogonal-triangular decomposition) is a numerically stable algebraic operation for an MIMO system to obtain the null space of a matrix. This is mathematically expressed as;

$$[A + BK]\underline{v}_i = \lambda \underline{v}_i \tag{6-13}$$

or

$$[A - \lambda_i I]\underline{v}_i = -BK\underline{v}_i = -B\underline{q}_i \tag{6-14}$$

where λ_i is a desired pole and $\underline{v}_i$ is an eigen vector.

Suppose a system is given as follows:

$$\dot{\underline{u}} = \begin{bmatrix} 0 & 1 \\ 4 & 0 \end{bmatrix} \underline{u} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \underline{r}_e \qquad (6\text{-}15)$$

where $\underline{u}$ and $\underline{r}_e$ correspond to the symbols in Figure 6-3 respectively. Let us assume that the desired poles are -2 and -4, meanwhile the system poles are -2 and 2. Equation (6-14) can be rewritten as;

$$[A - \lambda_i I, \; B] \begin{bmatrix} \underline{v}_i \\ \underline{q}_i \end{bmatrix} = 0 \qquad (6\text{-}16)$$

Thus, $[\underline{v}_i' \; \underline{q}_i']$ must be in the null space of $S=[A-\lambda_i I, \; B]$, and

$$\begin{aligned} K &= [\underline{q}_1, \; \cdots \; , \underline{q}_n][\underline{v}_1, \; \cdots \; , \underline{v}_n]^{-1} \\ &= Q V^{-1} \end{aligned} \qquad (6\text{-}17)$$

Hence, choosing V=I, for λ_1=-2,

$$S_1 = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 2 & 0 & 1 \end{bmatrix} \tag{6-18}$$

and for $\lambda_2 = -4$

$$S_2 = \begin{bmatrix} 4 & 1 & 1 & 0 \\ 4 & 4 & 0 & 1 \end{bmatrix} \tag{6-19}$$

The null space $[\underline{v}_i', \underline{q}_i']$ can be calculated using QR-decomposition. The control action K is then calculated as;

$$K = \begin{bmatrix} 2 & -1 \\ 4 & -4 \end{bmatrix} \tag{6-20}$$

The algorithm of QR-decomposition can be found elsewhere (e.g. Moler [1981]). The system expressed in Equation (6-15) together with the controller can be written as;

$$\dot{\underline{u}} = \begin{pmatrix} 2 & 0 \\ 8 & -4 \end{pmatrix} \underline{u} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \underline{r}_e \tag{6-21}$$

Using an external input $\underline{r}$ with coefficient E, Equation (6-21) can be rewritten as;

$$\dot{\underline{u}} = \begin{pmatrix} 2 & 0 \\ 8 & -4 \end{pmatrix} \underline{u} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} [E\underline{r} + K\underline{u}] \qquad (6\text{-}22)$$

Calibration of the matrix E can be determined using the infinite response to step changes as the external input $\underline{r}$. Details can be found elsewhere (Cellier [1985], and Forsythe, et al [1977]).

SUMMARY AND CONCLUSIONS

A numerical algorithm to simulate the dynamics of crystal size distribution was developed using the method of lines. This algorithm has several advantages. First, the new algorithm demonstrates the numerical stability criteria for continuous crystallizers which were previously developed. It is apparent that the algorithm adequetly represents the dynamics of crystal size distribution in a continuous crystallizer. Second, grid number for numerical calculation could be reduced using the method of lines. The grid number is roughly one half of that which is determined by the finite difference method. Thus, computational time can be reduced drastically and implementation of the simulator becomes feasible for a computer. Third, since certain points of crystal size were used as state veriables in the algorithm, conventional analytical methods in system theories could be adopted to analyze the crystallizer system and to design its controller. Finally, nucleation/growth rate kinetics were designed as an auxiliary function in the algorithm so that the main algorithm was independent of kinetics; thus, the kinetics can easily be substituted. Also, it is possible to define an arbitrary particle size distribution as an initial condition; therefore, the algorithm can be utilized for analysis of any growth-type particulate systems.

The developed algorithm was used for simulation of an experimental crystallizer system with fines removal in order to verify its credibility. Although the experimental data could not provide all the parameters desired for implemetation of the algorithm, simulation results were almost identical to the experimental ones. It was necessary to construct a simulation model corresponding to the experimental crystallizer; the simulator needed the parameters: recycle ratio of the fines dissolver (R) and the product classifier (z), crystal sizes at the upper cut size of the dissolver (x_F) and the lower cut size of the classifier (x_P), the increment of time ($d\theta$) and crystal size (dx), initial crystal size distribution, and configulation of upset. Among these parameters, R and x_F were determined from experimental data. The parameters $d\theta$ and dx were accomodated from computer specification. However, z and x_P were necessarily estimated using experimental data, assuming that a fictitious product classifier was installed in the system. The configulation of the upset was evaluated from simulation results using various rectangular-shaped upsets. Simulation results showed that the R-z crystallizer system was properly simulated in this dynamic configulation. Fluctuations in crystal size distribution, pulse propagation, and elapse time of response to the upset were well-modeled. Thus, it is possible to predict the dynamics of an R-z crystallizer given a certain set of parameters. However, it is desirable to experimentally investigate the uncertainty of the parameters z and x_P, and pulse configulation.

A design algorithm for a proportional controller was achieved using pole placement and optimization techniques. It was found that pole placement is a better approach than optimization for crystallizer controller design, as the system poles are concentrated at a point.

Differeces between practical and theoretical conditions for an R-z crystallizer were evaluated. Considering characteristics and sensitivity of parameters, it is evident that nuclei and population densities are suitable for the controller-input-output variables and that the fines dissolving ratio R is proper for the manipulation variable. Realistically, only one fines dissolver is installed in a crystallizer. Nuclei density apparently indicates the total system dynamic tendency. Thus, nuclei density was chosen as a signal to design an actual controller.

Optimization techniques were used to compare the results of the theoretical development in this study with previous controller research and an experimental control study. The theoretical discussion verified that scalar operation using a controller to optimize crystal size distirbution does not make sense from system theory; matrix operation must be adopted to realize the system whenever optimization in crystal size distribution is considered. This reason is that crystal size distribution consists of a vector in crystal size and amount. The results of this study showed that the optimal controller in nuclei density is almost identical to that in slurry density which was used in the experiment. Thus, this method

is effective in designing an optimal controller for an continuous crystallizer.

Finally, a control algorithm using pole placement techniques with the minimum order Luenberger observer was achieved. Using this method, controller poles can be chosen freely and poles can be properly assigned for an R-z crystallizer. An observer which tracks the reconstructed model to the real system can be advantageously designed for an actual crystallizer susytem using the pole placement techniques so that it is possible to design an effective real-time on-line controller for crystal size distribution. The observer contributes to a reduction of the number of measurement points in population densities. It is necessary to experimentally implement the algorithm shown in Figure 6-9 in order to confirm its efficacy.

APPENDIX A

MATHEMATICAL DEVELOPMENT OF THE DYNAMIC CSD EQUATION

In this section, a mathematical development of the dynamic crystal size distribution equation will be discussed. A brief review of the basic theory of partial differential equations with computers and discussions of the formulae used to approximate space derivatives with the method of lines, which is used in this study, can be found elsewhere [Carver, et al [1976]]. The the method of lines technique is summarized as follows.

1) The versatility of the method of lines is obvious, as it is completely modular in its approach.

2) In practice, more accurate formulae are recommended; higher order approximations in space allow the time integration to proceed more efficiently and the stiffness of the equation system is reduced.

3) Increasing the number of spatial divisions will increase accuracy, but it is also established that this increases the stiffness of the equations thus requiring a lower maximum integration step size.

4) The error is considerably greater for asymmetric formulae, so for a uniform error profile, formulae of higher order

should be used at the end points or error will propagate from these points.

Items (2) and (3) are important in writing a computer program, as both the higher order approximation and the larger number of spatial divisions increase the dimension of the matrix. As memory storage capacity for most computers defines the block size of the image files, the order of approximation and the number of spatial divisions need to be considered carefully before programming.

Generally speaking, hyperbolic wave equations are more difficult to solve because they generate wave solutions. Frequently these waves become steep fronted, or shock waves. Carver, et al [1976] recommend that a first order hyperbolic equation be converted to a second order wave equation if possible. For the simplest case of a hyperbolic wave equation, comparisons between the true solution and the numerical ones are illustrated in Figure A-1.

In the case of this study, the starting point is the equation;

$$\frac{\partial u}{\partial \theta} + \phi(\theta) \frac{\partial u}{\partial x} + h(x)u = 0 \qquad \text{(A-1)}$$

The boundary condition is described as;

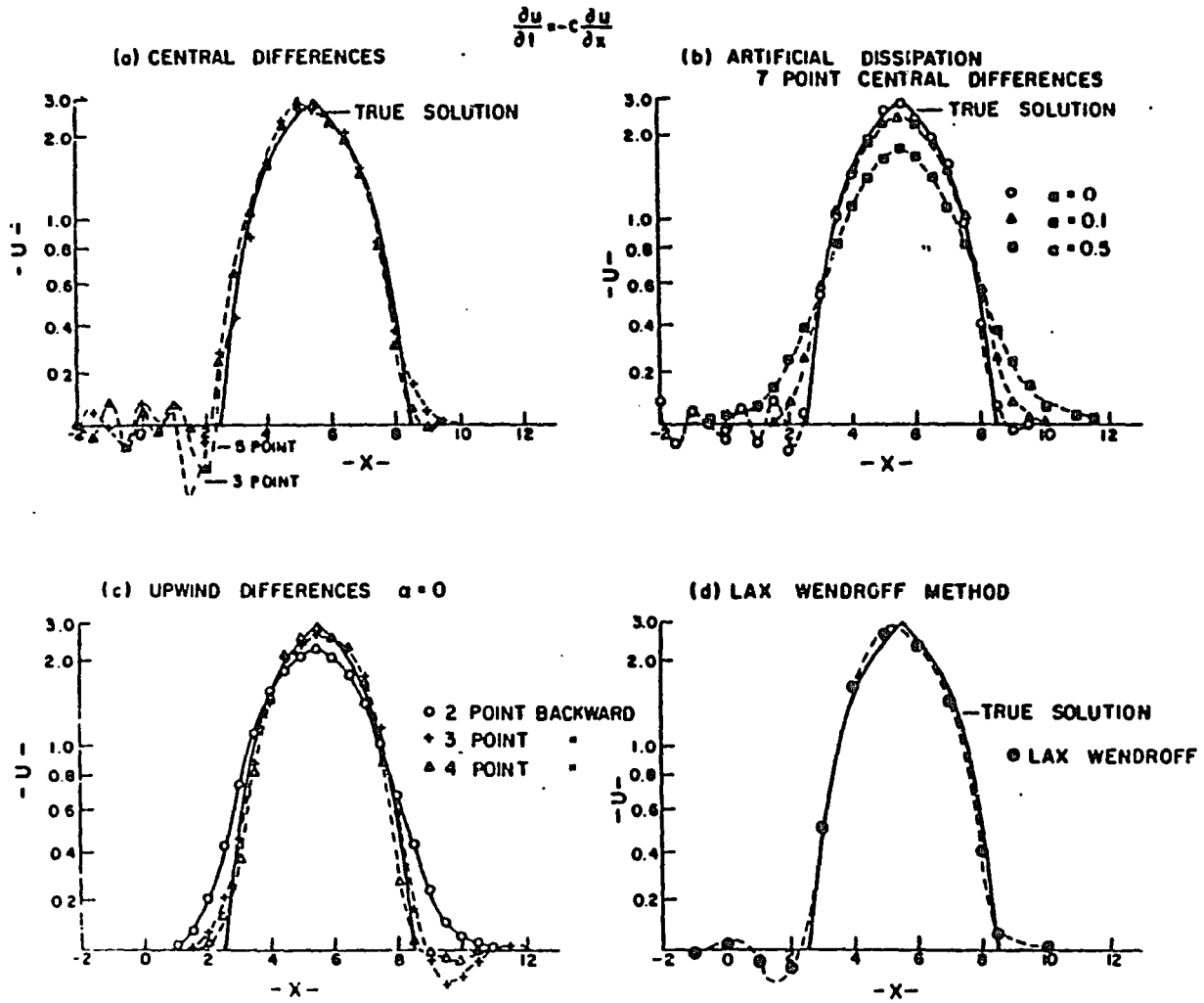


Figure A-1 Analytical techniques in the first order Hyperbolic equation [Carver et al [1976]].

$$\phi = \frac{1 + \beta f_{3,1}}{1 + \beta} \quad \frac{1}{f_2} \tag{A-2}$$

$$u_0 = f_3^{\,j}\, \phi^{\,i-1} \tag{A-3}$$

$$\beta = \frac{(R - 1)}{6 D_P} \int_0^{x_f} x^3 \bar{u}\, dx \tag{A-4}$$

The initial condition of CSD u, is defined as an arbitrary distribution function of dimensionless crystal size, x.

Since Equation (A-1), a first order partial differential equation, becomes a heterogeneous first order partial differential equation adding an external input $\underline{r}(t)$, (which is usually zero for an industrial crystallizer), to the right hand side of Equation (A-1), it is hard to convert Equation (A-1) with the external input to a second order wave equation so Equation (A-1) must be solved directly. As mentioned above, increasing the spatial step size gives a more accurate calculation result, but the number of steps is limited by the storage capacity of computer memory. Also, a symmetrical form is recommended. Thus, a fourth order correct central difference approximation is adopted. This gives a neat form, suitable for designing a controller.

We are interested in fourth order approximations of $\partial u/\partial x$ and proceed by writing a Tayler series expansions for u_{l-2}, u_{l-1}, u_{l+1}, and u_{l+2} about the grid point l. Thus,

$$u_{\ell-2} = u_\ell - 2\delta u'_\ell + \frac{4}{2!}\delta^2 u''_\ell - \frac{8}{3!}\delta^3 u'''_\ell + \frac{16}{4!}\delta^4 u''''_\ell + \cdots \tag{A-5a}$$

$$u_{\ell-1} = u_\ell - \delta u'_\ell + \frac{1}{2!}\delta^2 u''_\ell - \frac{1}{3!}\delta^3 u'''_\ell + \frac{1}{4!}\delta^4 u''''_\ell + \cdots \tag{A-5b}$$

$$u_{\ell+1} = u_\ell + \delta u'_\ell + \frac{1}{2!}\delta^2 u''_\ell + \frac{1}{3!}\delta^3 u'''_\ell + \frac{1}{4!}\delta^4 u''''_\ell + \cdots \tag{A-5c}$$

$$u_{\ell+2} = u_\ell + 2\delta u'_\ell + \frac{4}{2!}\delta^2 u''_\ell + \frac{8}{3!}\delta^3 u'''_\ell + \frac{16}{4!}\delta^4 u''''_\ell + \cdots \tag{A-5d}$$

where

$$\delta = (\Delta x) \tag{A-6}$$

Multiplying these four equations respectively by +1, -8, +8, and -1, and adding, one gets

$$u'_\ell = \left(\frac{\partial u}{\partial x}\right)_{x=x_\ell} \tag{A-7}$$

$$= \frac{(u_{\ell-2} - 8u_{\ell-1} + 8u_{\ell+1} - u_{\ell+2})}{12\delta} + O(\delta^4)$$

which is the desired fourth order accurate formula. This formula is applicable at all points except those that are one mesh away from the boundary and those on the boundary. This problem can be overcome by using two fictitiuous grid points beyond the boundary calculated using a cubic extrapolation formula. Specifying the fictitiuous grid points beyond the left boundary by subscripts -1 and -2, and those beyond the right boundary by subscripts L+1 and L+2, one can write a cubic extrapolation formula [See Vemuri and Karplus, [1981], Chapter 4] as;

$$u_{-1} = 4u_0 - 6u_1 + 4u_2 - u_3 \qquad \text{(A-8a)}$$

$$u_{-2} = 4u_{-1} - 6u_0 + 4u_1 - u_2 \qquad \text{(A-8b)}$$

$$u_{L+1} = 4u_L - 6u_{L-1} + 4u_{L-2} - u_{L-3} \qquad \text{(A-8c)}$$

$$u_{L+2} = 4u_{L+1} - 6u_L + 4u_{L-1} - u_{L-2} \qquad \text{(A-8d)}$$

Returning to the dynamic crystal size distribution equation, one can express Equation (A-1) as a semidiscrete approximate formula using Equation (A-7);

$$\frac{du_\ell}{d\theta} + \phi(\theta)\ \frac{u_{\ell-2} - 8u_{\ell-1} + 8u_{\ell+1} - u_{\ell+2}}{12\delta} + h_\ell u_\ell \qquad (A-9)$$

$$= 0$$

where the residue $O(\delta^4)$ is ignored and the external input is omitted. Substitution of Equations (A-8) into Equation (A-9) gives a vector-matrix equation;

$$\dot{\underline{u}}(\theta) = A(\theta)\underline{u}(\theta) \qquad (A-10)$$

$$\underline{u} = (u_0, u_1, \ldots, u_L)' \qquad (A-11)$$

where

$$A(\theta) = A[\phi(\theta)] \qquad (A-12)$$

and is written out in Equation (A-13).

Since the external input $\underline{r}(\theta)$ is given independently, the general expression of the dynamic crystal size distribution equation is written as;

$$\dot{\underline{u}}(\theta) = A(\theta)\underline{u}(\theta) + B\underline{r}(\theta) \qquad (A-14)$$

$$A(\theta) = \frac{\phi}{12\delta}\begin{bmatrix} -22+\frac{12h_0\delta}{\phi}, & 36, & -18, & 4, & 0, & \cdots & , 0 \\ -4, & -6+\frac{12h_1\delta}{\phi}, & 12, & -2, & 0, & \cdots & , 0 \\ 1, & -8, & \frac{12h_2\delta}{\phi}, & 8, & -1, & 0, \cdots & , 0 \\ \vdots & & & & & & \\ 0, \cdots, 0, & 1, & -8, & \frac{12h_{1-1}\delta}{\phi}, & 8, & -1, & 0, \cdots, 0 \\ \vdots & & & & & & \\ 0, & \cdots & , 0, & 2, & -12, & 6+\frac{12h_{L-1}\delta}{\phi}, & 4 \\ 0, & \cdots & , 0, & 4, & 18, & -36, & 22+\frac{12h_L\delta}{\phi} \end{bmatrix} \qquad \text{(A-13)}$$

where B is an arbitrary constant coefficient matrix. Although matrix B is defined as the amplitude ratio of a gain which gives a unit response for a unit step change, B is expressed as an unit matrix for convenience at this point. This problem is beyond the subject of this appendix. It will be discussed in the chapter on controller design.

An analytical solution of Equation (A-14) can be obtained by formal integration [Vemuri and Karplus [1981]];

$$u(\theta) = \Phi(\theta, \theta_0)\,\bar{u} + \int_{\theta_0}^{\theta} \Phi(\theta\ \sigma)\, B\, \underline{r}(\sigma)\, d\sigma \qquad \text{(A-15)}$$

$$\Phi(\theta, \tau) = \exp\left[\int_{\tau}^{\theta} A(\sigma) d\sigma\right] \qquad \text{(A-16)}$$

According to Kailath [1981], it is only possible for Equation (A-15) and (A-16) to yield $\underline{u}(\theta)$ numerically if matrix A is of upper or lower triangular form, or if matrix A is unitary; otherwise, it is impossible. In this study, the matrix size of matrix A is expected to be more than 10 x 10 so one can no longer calculate Equation (A-15) and (A-16) by hand. Thus, it is assumed that matrix $A(\theta)$ is invariant during time increment $\Delta\theta$. This assumption easily gives the time discrete formulae of Equation (A-15) and (A-16);

$$\underline{u}_{m+1} = \Phi_{m+1}\,\underline{u}_m + \int_0^{\theta_m} \Phi_m(\theta - \sigma)\, B\, \underline{r}_m\, d\sigma \qquad \text{(A-17)}$$

$$\Phi_m = \exp\left[A_m \theta \right] \qquad \text{(A-18)}$$

If the time increment $\Delta\theta$ is small enough and unique, the second term of the right hand side of Equation (A-17) can be rewritten using trapezoidal integration and Equations (A-17) and (A-18) become the simpler expressions;

$$\underline{u}_{m+1} = \Phi_{m+1}\,\underline{u}_m + \Delta\theta\, \frac{\left[\Phi_m B \underline{r}_{m+1} + \Phi_{m+1} B \underline{r}_m\right]}{2} \qquad \text{(A-19)}$$

$$\Phi_m = \exp\left[A_m \Delta\theta \right] \qquad \text{(A-20)}$$

In this study, integration is done analytically, but numerical methods are provided in the FORSIM package program.

APPENDIX B

SUMMARY OF POLE PLACEMENT AND OPTIMIZATION PROBLEMS

Modern system theories consist of two major methods: (B-1) pole placement or pole assignment and (B-2) optimization problems. Those studies have been mainly pursued in the field of aerospace engineering. The design methodology for multiple input-multiple output systems, which has been developed this decade, has not been introduced to the field of crystallization. In the field of chemical engineering, a few applications of the techniques for heat transfer problems were studied [Leden, Hamza, and Sheirah [1973], and Sheirah and Hamza [1974]]. Thus, in this appendix, the concept of the methodology will be summarized covering the ideas necessary for this study. One may find details in the books on system theory written after 1980, such as "Linear System" by Kailath [1981].

Pole placement

Controllable poles of a system can be placed freely by state feed back, which is called pole placement problem or pole assignment problem. Notice that controllablity of a system is generally defined for the minimal realization in which ranks of controllability and observability matrices are identical to the dimension of the system matrix. The uncontrollable poles are not affected by state

feed back, nor either are zeros. Linear algebra theories and theorems contribute to those derivations. As a general rule, the theorem of state feed back states as follows [Cellier [1985]].

By feeding back a linear combination of all states, any controllable system is able to be brought to a closed-loop form where the poles can be chosen freely. Suppose a given system;

$$\dot{\underline{x}}=A\underline{x}+B\underline{u} \tag{B-1}$$

$$\underline{y} = C\underline{x} \tag{B-2}$$

Apparently, state feed back $\underline{u}$ becomes

$$\underline{u} = K\underline{x} + E\underline{r} \tag{B-3}$$

where matrices K and E are defined as is described in Figure B-1. Hence, closed-loop system equation is given as;

$$\dot{\underline{x}} = (A + BK)\underline{x} + BE\underline{r} \tag{B-4}$$

$$\underline{y} = (C + DK)\underline{x} + DE\underline{r} \tag{B-5}$$

If A is expressed as the controller canonical form, then the

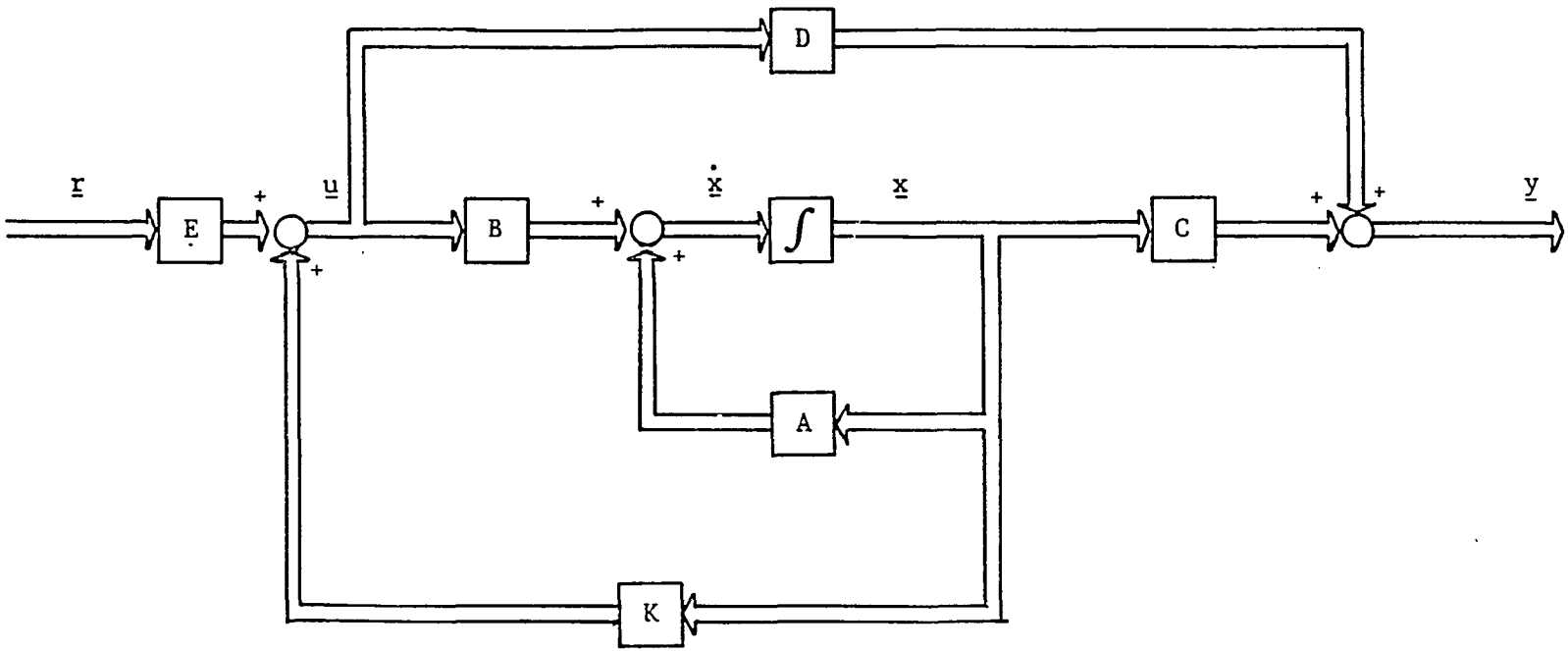


Figure B-1 A schematic flow diagram of multiple input-multiple output system.

term (A + BK) still remains in the controller canonical form. It means that the term (A + BK) has the same structure as matrix A has. Now placing poles wanted as matrix M, controller K is calculated as;

$$K = {}^{-1}(M - A) \tag{B-6}$$

Note that if the system expressed as Equation (B-1) is not controllable, the algorithm can be applied only to the controllable part of the matrix and that the pole placement problem for the multiple input-multiple output problem does not give any unique solution.

In the above discussion, it is assumed that all of state variables are observable; however, that is relatively rare. Meanwhile, the number of measuring points needs to be reduced to the dimension of the model of observer.

When the state variables are not physically accessible to measurement, one cannot get them back by

$$\underline{x} = C^{-1}\underline{y} \tag{B-7}$$

because matrix C is usually not square. Then, a model of the system has to be established and an inner control loop to track the model states to the real ones has to be designed. The difference, or

tracking error, between measured and reconstructed outputs is fed back into the model in order to track the reconstructed state $\hat{x}$. It is not necessary to estimate all n states. Measuring p output, then (n-p) states should be estimated, but p states can be obtain directly. This is called minimum order Luenberger observer [Kailath [1981]] which is schematically illustrated in Figure B-2.

The procedure to obtain the observer is outlined as follows. Assume that a system expressed by Equations (B-1) and (B-2) is observable, but

$$\text{RANK}(C) = p < \text{RANK}(A) = n \tag{B-8}$$

It is always possible to find a transformation matrix with a null space T of matrix C such as

$$P = \begin{pmatrix} T \\ C \end{pmatrix} \tag{B-9}$$

Hence, the new state variable is defined as

$$\underline{\xi} = P\underline{x} = \begin{pmatrix} T \\ C \end{pmatrix} \underline{x} = \begin{pmatrix} \underline{w} \\ \underline{y} \end{pmatrix} \tag{B-10}$$

The new system may be expressed as;

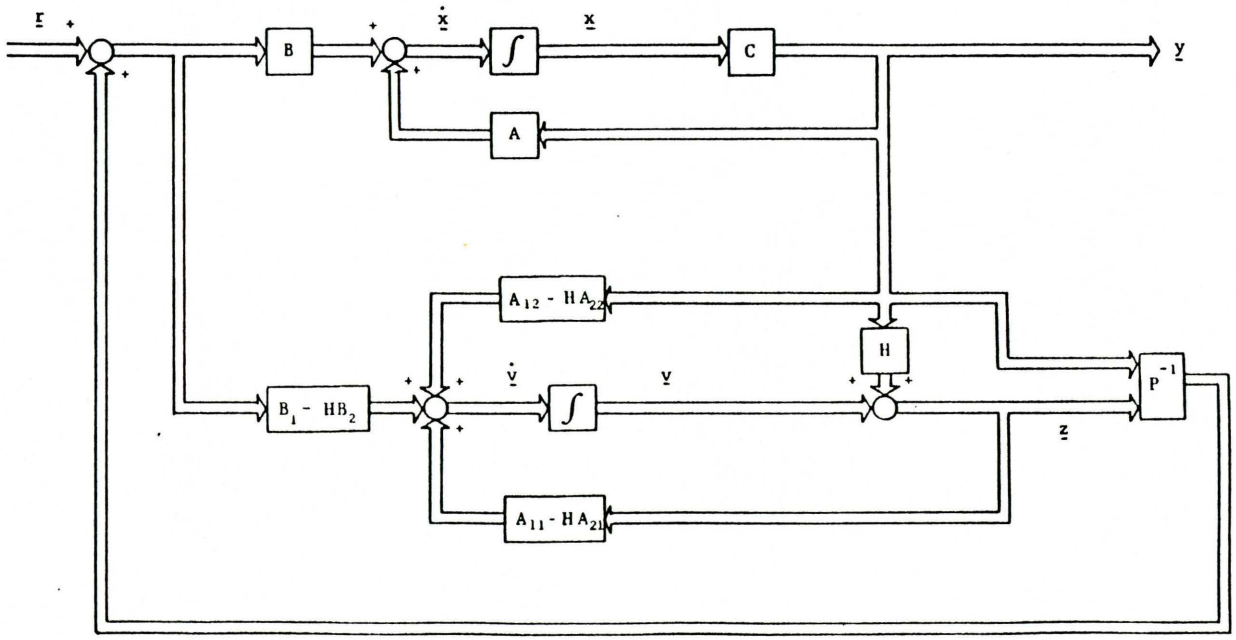


Figure B-2 A schematic flow diagram of multiple input-multiple output system with the minimum order Luenberger observer.

$$\dot{\underline{\xi}} = \begin{pmatrix} \dot{\underline{w}} \\ \dot{\underline{y}} \end{pmatrix} = PAP^{-1} \begin{pmatrix} \underline{w} \\ \underline{y} \end{pmatrix} + PB\underline{u} \tag{B-11}$$

$$= \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} \underline{w} \\ \underline{y} \end{pmatrix} + \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} \underline{u}$$

$$\underline{y} = \begin{bmatrix} 0 & \underbrace{\begin{matrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{matrix}}_{p} \end{bmatrix} \underline{\xi} \tag{B-12}$$

Alternatively, the system can be described as

$$\dot{\underline{w}} = A_{11}\underline{w} + \begin{bmatrix} A_{12} & B \end{bmatrix} \begin{bmatrix} \underline{y} \\ \underline{u} \end{bmatrix} \tag{B-13}$$

$$\underline{y}_r = A_{21}\underline{w} \tag{B-14}$$

The detail description of the derivation of Equations (B-13) and (B-14) is written by Kailath [1881], but note that the subscripts order of vectors and matrices are opposite to the above way. At any

rate, the dummy output from the model becomes

$$\underline{v} = (A_{11} - HA_{21})\underline{v} + (A_{12} - HA_{22} + A_{11}H - HA_2H)\underline{y} + (b_1 - HB_2)\underline{u} \tag{B-15}$$

The error of the observer decreases with the stablepoles of matrix $(A_{11} - HA_{21})$.

Optimization

A general problem of optimal control is stated so that performance index PI for a state variable $\underline{x}(t)$ is minimized. Suppose that a system is described by time t, state variable $\underline{x}(t)$, and input $\underline{u}(t)$;

$$\underline{\dot{x}} = \underline{f}\ (\underline{x},\ \underline{u},\ t) \tag{B-16}$$

$$PI = \int_0^{t_1} \varphi(\ \underline{x}, \underline{u}, t\ )\, dt \tag{B-17}$$

For Eqations (B-16) and (B-17), matrix Hamilton's formation is given as

$$\phi(\ \underline{x}, \underline{\dot{x}}, \underline{u}, \underline{\psi}, t\ ) = \varphi(\ \underline{x}, \underline{u}, t\ ) + \psi(t)'\, \underline{f}(\ \underline{x}, \underline{u}, t\ ) - \underline{\psi}'(t)\, \underline{\dot{x}}(t) \tag{B-18}$$

where Hamiltonian H and condition for $\psi(t)$ are written respectively as;

$$H(\underline{x}, \underline{u}, t, \underline{\psi}) = \varphi(\underline{x}, \underline{u}, t) + \underline{\psi}'(t) \cdot f(\underline{x}, \underline{u}, t) \tag{B-19}$$

$$\dot{\underline{\psi}} = - \frac{\partial \varphi}{\partial \underline{x}} \tag{B-20}$$

Equation (B-19) is called Hamilton function. For a linear system such as Equation (B-1), the performance index PI is defined as;

$$PI = \underline{y}'(t_1) \cdot S \underline{y}(t_1) + \int_0^{t_1} \Big[\underline{y}'(t)\, Q(t)\, \underline{y}(t) + \underline{u}'(t)\, R(t)\, \underline{u}(t) \Big] dt \tag{B-21}$$

where matrices S and Q are positive-semidefinite symmetric matrices and matrix R is a positive-definite symmetric matrix. Furthermore, the following assumption is generally set;

$$t_{11} = (\text{ fixed }) \tag{B-22a}$$

$$\underline{x}(t_1) = (\text{ free }) \tag{B-22b}$$

$$\underline{u}(t) = (\text{ not limited }) \tag{B-22c}$$

Using Hamiltone's formation and calculating it tediously, Equation (B-21) can be converted to a boundary value problem;

$$\begin{bmatrix} \dot{\underline{x}} \\ \dot{\underline{\psi}} \end{bmatrix} = \begin{bmatrix} A & -\frac{1}{2} B R^{-1} B' \\ -2 C'QC & -A' \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{\psi} \end{bmatrix}. \tag{B-23}$$

The performance index is now algebraically determined.

In general, it is known that a boundary value problem involving a matrix equation cannot be solved numerically except in a few special cases. It is necessary to change the form of Equation (B-23) to the initial value problem using an equation [Lapidus, and Luus [1967]];

$$\underline{\psi}(t) = 2K(t)\underline{x}(t) \tag{B-24}$$

Equation (B-23) is rewritten as Riccati-Differential equation;

$$-K = kA + A'K + C'QC - KBR^{-1}B'K \tag{B-25}$$

Thus, the optimal condition for constant K when t_1 goes to infinite is written as;

$$0 = KA + A'K + C'QC - KBR^{-1}B'K \qquad \text{(B-26)}$$

Equation (B-26) is called Matrix-Riccati equation.

Although Equation (B-26) is expressed in the simple formula and it is assumed that optimal controller K may be numerically obtained by Newton's method, it it quite difficult to solve the equation for matrix K because of its vector-matrix form. An alternative way to solve for matrix K is introduced using the technique of pole placement problem [Cellier and Rinuvall [1983]]. The Matrix-Riccati equation is converted into a linear equation using a new variable such that;

$$\dot{\theta} = E\theta \qquad \text{(B-27)}$$

Matrix E is defined as;

$$E = \begin{bmatrix} A & -BRB' \\ -C'QC & -A' \end{bmatrix} \qquad \text{(B-28)}$$

Then, the eigen space V and modal matrix are evaluated. The eigen vector $\underline{v}_i$ corresponding to the eigen value λ_i which is on the left half plane of the complex plane is chosen and eigen space V* will be

reconstructed;

$$V^* = \overset{n}{\left[\begin{array}{c} \\ \\ \end{array}\right]} 2n \qquad \text{(B-29)}$$

Eigen space V^* is split into the upper and lower part;

$$V^* = \left[\begin{array}{c} V_1 \\ \\ V_2 \end{array}\right] \qquad \text{(B-30)}$$

Null space V_2 of space V_1 is determined as:

$$P = V_2 V_1^{-1} \qquad \text{(B-31)}$$

Thus, optimal controller matrix K is determined as;

$$K = B^{-1} R^{-1} P \qquad \text{(B-32)}$$

This algorithm is available for a linear system described by the form such as that of Equation (B-1). Matrix A here is expectedly regular. Also, this algorithm is believed to be the most numerically stable one.

APPENDIX C

STEADY STATE CONDITIONS FOR A GIVEN CRYSTALLIZER

In analyzing an R-z crystallizer dimensionally, it is necessary to evaluate the steady state conditions for a given crystallizer. After Beckman and Randolph [1977], the solute balance equation is expressed as:

$$\frac{F}{\tau}(c_F - \gamma c) + \frac{P_R}{V}\left(1 - \frac{c}{\rho}\right)\left(\frac{R - \gamma F}{R - 1}\right) = (\rho - c)\frac{k_A G m_2}{2} + \epsilon_c \frac{dc}{dt} \qquad (C\text{-}1)$$

where

$$\epsilon_c = \frac{V}{V_L} \qquad (C\text{-}2a)$$

$$\gamma = (Q_P - Q_{CLA}) / Q_F \qquad (C\text{-}2b)$$

Growth rate is assumed to be proportional to supersaturation as;

$$G = k_g(c - c_s) \qquad (C\text{-}3)$$

Equations (C-1) and (C-3) must be satisfied simultaneously, though growth rate is apparently quadratic to c_s in Equation (C-1), but proportional to c_s in Equation (C-3). Beckman and Randolph [1977] solved these equations using the Class II (high yield) assumption:

$$\frac{dc}{dt} = 0 \tag{C-4}$$

The condition expressed by Equation (C-4) gives Equations (15) through (22) in their paper; however, Equations (16) and (18) give infinite intermediate parameters and hence indifinite concentration at t=0. Thus, Equations (C-1) and (C-3) will be solved without using Equation (C-4) avoiding singularity at t=0.

Substitution of Equation (C-3) into Equation (C-1) gives;

$$\frac{F}{\tau} c_F - \gamma c + \frac{P_R}{V}\left(1 - \frac{c}{\rho}\right)\left(\frac{R - \gamma F}{R - 1}\right) = (\rho - c)\frac{k_A k_g (c - c_s) m_2}{2} + \epsilon_c \frac{dc}{dt} \tag{C-5}$$

Define supersaturation ratio as;

$$x = \frac{c - c_s}{c_s} \tag{C-6}$$

Using Equations (C-6) and (3-11) in the text, Equation (C-5) can be converted the non-dimensionalized and normarlized form as;

$$\frac{dx}{d\theta} = \frac{k_A f_2' g k_g'}{2\epsilon_c} x^2 - \frac{1}{\epsilon_c}\left\{ \frac{k_V f_{3,1} g (R - \gamma F)^2}{(R - 1)} + \frac{(rp - 1) k_A f_2' g k_g'}{2} \right\} x + \left\{ (r_p - 1) \frac{k_V f_{3,1} g (R - \gamma F)^2}{(R - 1)} + F (x_{cF} + 1 - \gamma) \right\} \tag{C-7}$$

or

$$x' = ax^2 + bx + c \tag{C-8}$$

where

$$g = (\bar{G}\tau)\bar{n}^{\circ} \tag{C-9}$$

$$r_p = \frac{\rho}{c_s} \tag{C-10}$$

$$a = \frac{k_A f_2' g k_g}{2\epsilon_c} \tag{C-11}$$

$$b = \frac{-1}{\epsilon_c}\left\{ \frac{k_V f_{3,1} g\,(R - \gamma F)^2}{(R - 1)} + \frac{(r_p - 1)k_A f_2' g k_g'}{2} \right\} \qquad (C\text{-}12)$$

$$c = \left\{ \frac{(r_p - 1)k_V f_{3,1}' g (R - \gamma F)^2}{(R - 1)} - F(x_{cF} + 1 - \gamma) \right\} \qquad (C\text{-}13)$$

Equation (C-8), furthermore, can be rewritten as;

$$\int_{X_0}^{X} \frac{2}{aX^2 + bX + c}\, dX = \int_0^{\theta} d\theta \qquad (C\text{-}14)$$

The general solutions for Equation (C-14) are given as;

$$\theta = \frac{1}{\sqrt{b^2 - 4ac}}\left\{ \ell n \frac{(2ax + b - \sqrt{b^2-4ac})(2ax_0 + b + \sqrt{b^2-4ac})}{(2ax_0 + b + \sqrt{b^2-4ac})(2ax + b - \sqrt{b^2-4ac})} \right\} \qquad (C\text{-}15)$$

for the condition $b^2-4ac>0$, or

$$\theta = \frac{1}{a\left(x_0 + \frac{a}{2}\right)} - \frac{1}{a\left(x + \frac{b}{2}\right)} \qquad (C\text{-}16)$$

for the condition $b^2-4ac=0$. Hence, x can be obtained as;

$$x = \frac{\beta\, e^{\theta\sqrt{b-4ac}} - \alpha A}{2a(A - e^{\theta\sqrt{b-4ac}})} \qquad (C\text{-}17)$$

for $b^2-4ac>0$, or

$$x = \frac{x_0\left(1 - \frac{ab\theta}{2}\right) - \frac{ab^2\theta}{4}}{1 - a\theta x_0 - \frac{ab\theta}{2}} \qquad (C\text{-}18)$$

for $b^2-4ac=0$, where

$$\alpha = b - \sqrt{b - 4ac} \tag{C-19}$$

$$\beta = b + \sqrt{b - 4ac} \tag{C-20}$$

$$A = \frac{2ax_0 + \beta}{2ax_0 + \alpha} \tag{C-21}$$

Therefore, steady state condition x_s can be defined for $\theta \to \infty$; thus, Equation (C-14) and (C-15) become

$$x_s = -\frac{\beta}{2\alpha} \tag{C-22}$$

and

$$x_s = -\frac{b}{2} \tag{C-23}$$

for $b^2-4ac>0$ and $b^2-4ac=0$ respectively.

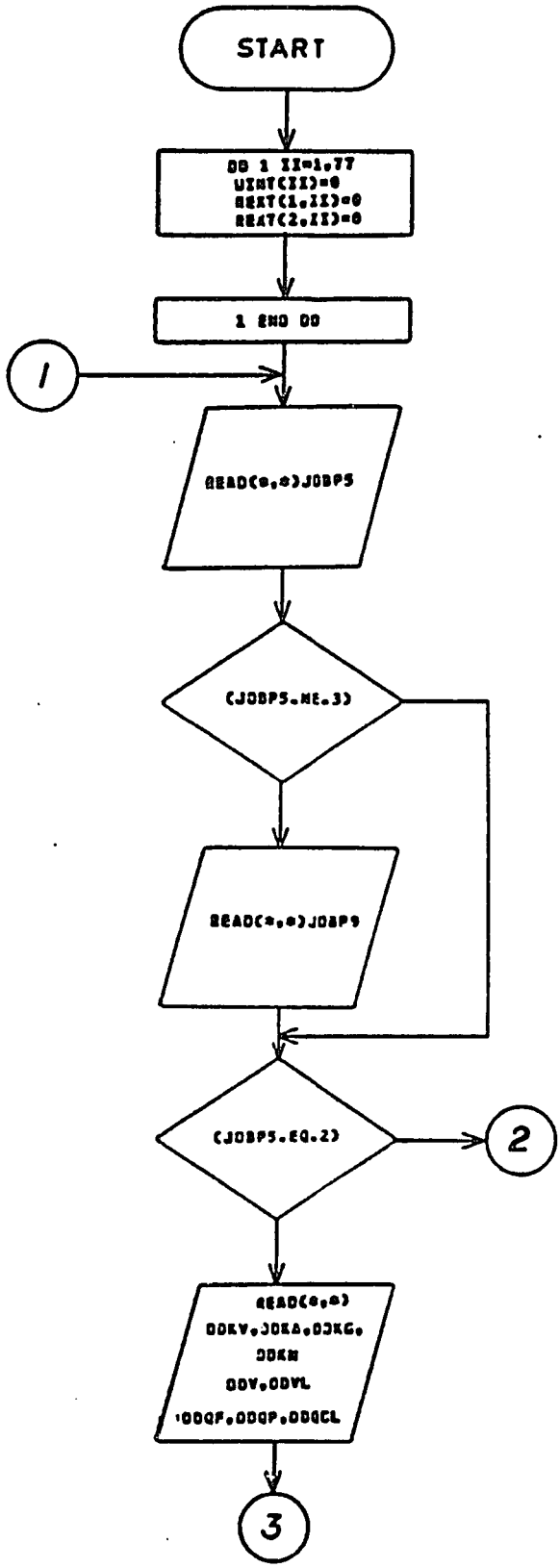
As a summary, an algorithm to determine steady state condition is described as follows. First, it is necessary to evaluate;

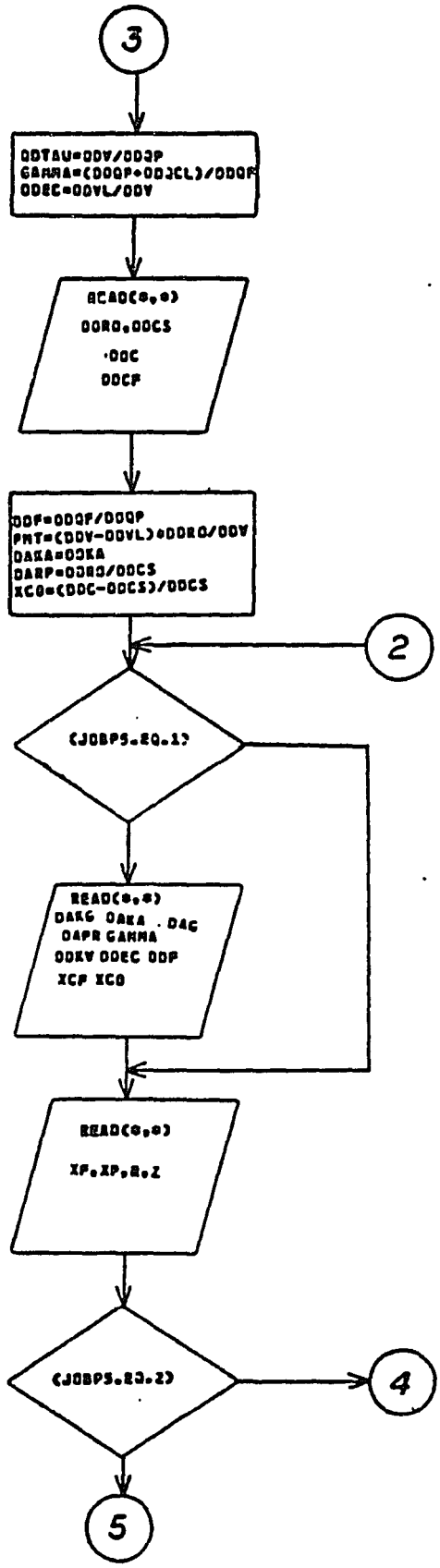
$$D = b^2 - 4ac \tag{C-24}$$

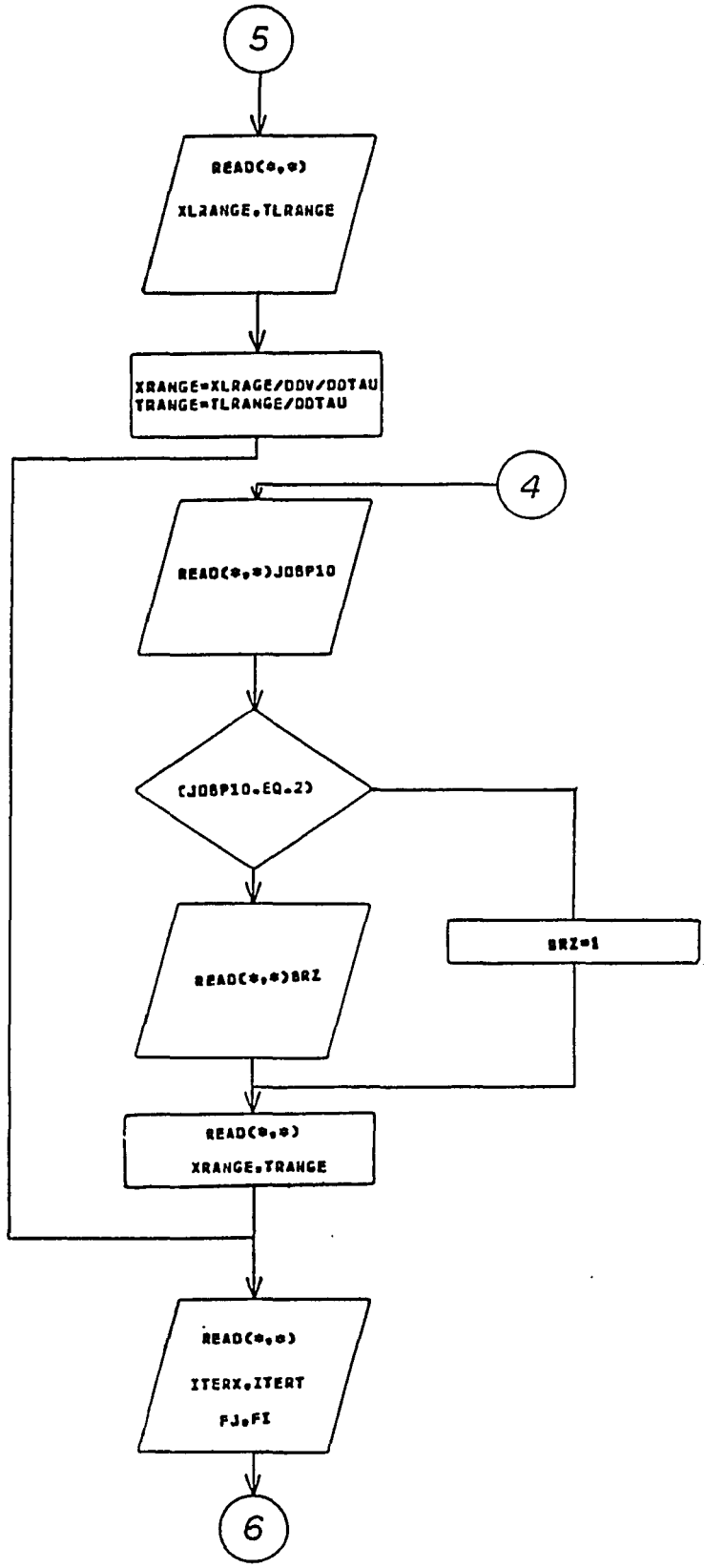
If D is negative, calculation is not advanced because such a system no longer exists. If D is positive or zero, Equation (C-22) or (C-23) is used to obtain steady state supersaturation ratio x_s, and then growth rate G is determined using Equation (C-3).

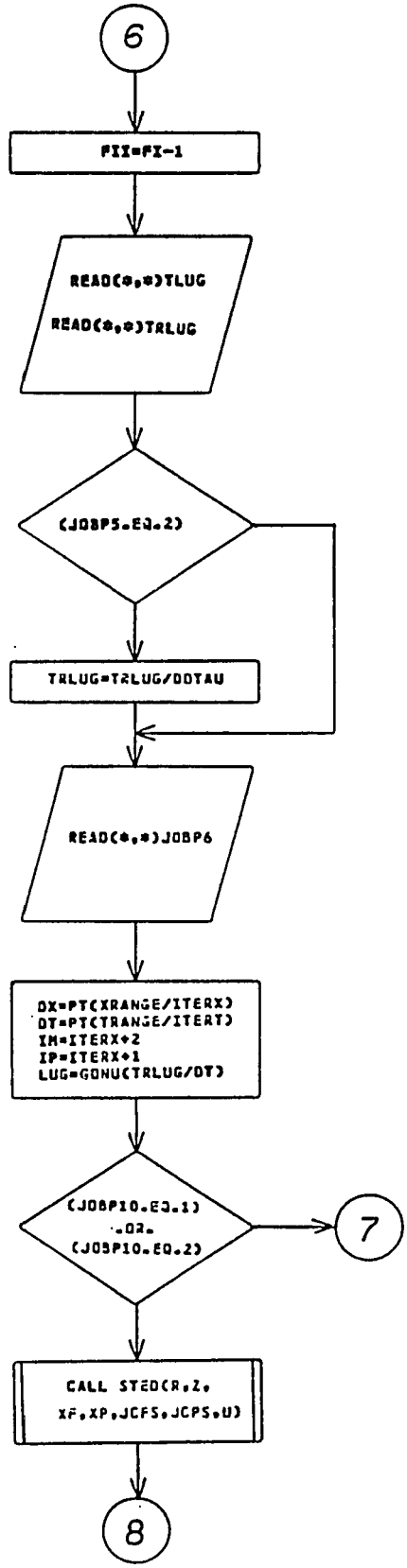
APPENDIX D

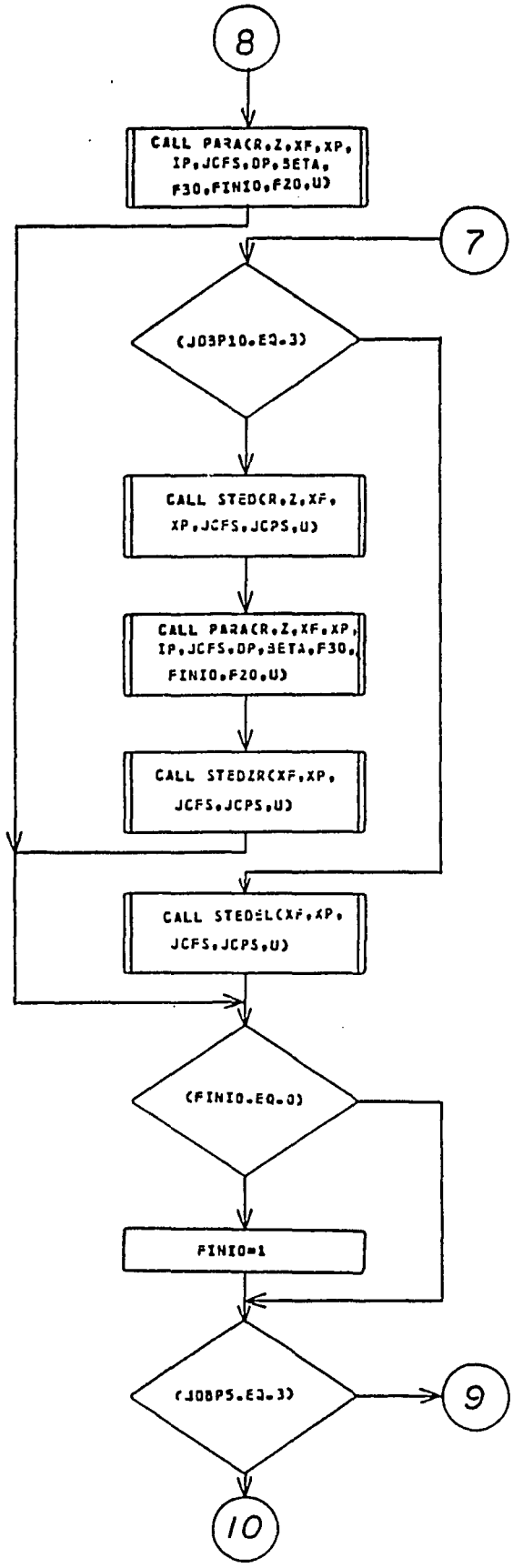
FLOW CHART AND COMPUTER PROGRAM OF "DYNE"

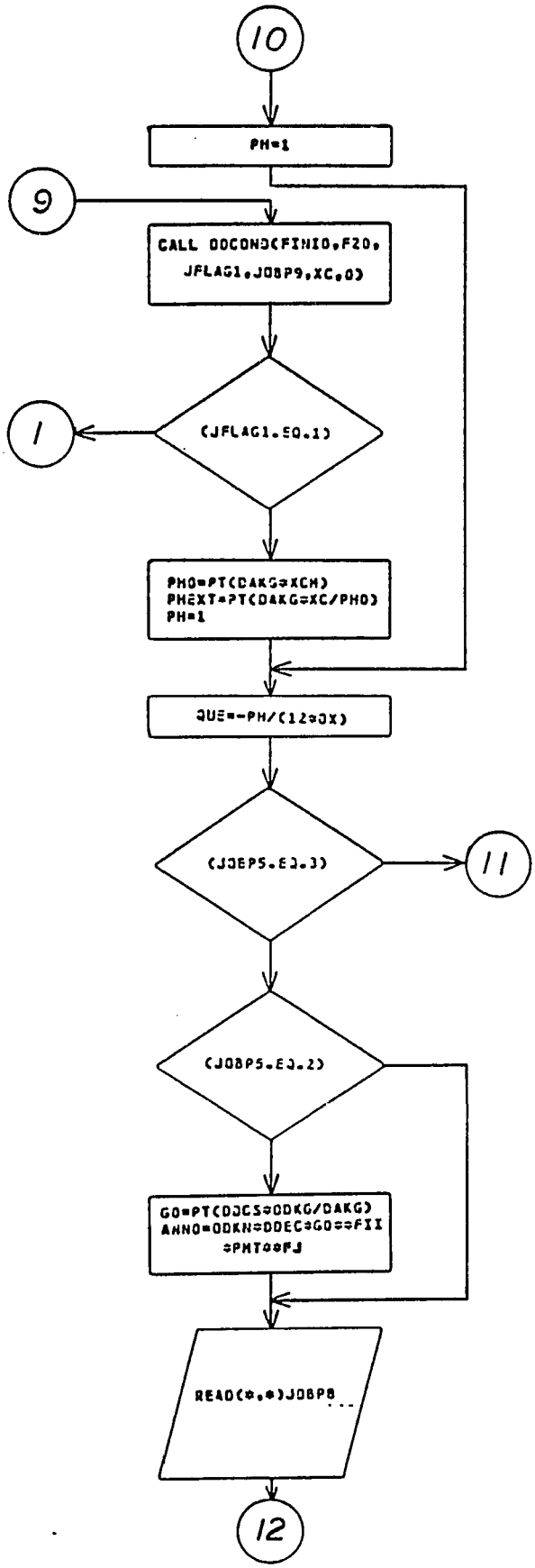

START
DO 1 II=1,77
UINT(II)=0
REAT(1,II)=0
REAT(2,II)=0
1 END DO
1
READ(0,0)JOBP5
(JOBP5.NE.3)
READ(0,0)JOBP9
(JOBP5.EQ.2)
2
READ(0,0)
DDKV,JOKA,DJKG,
JDKN
DDV,DDVL
,DDQF,DDQP,DDQCL
3

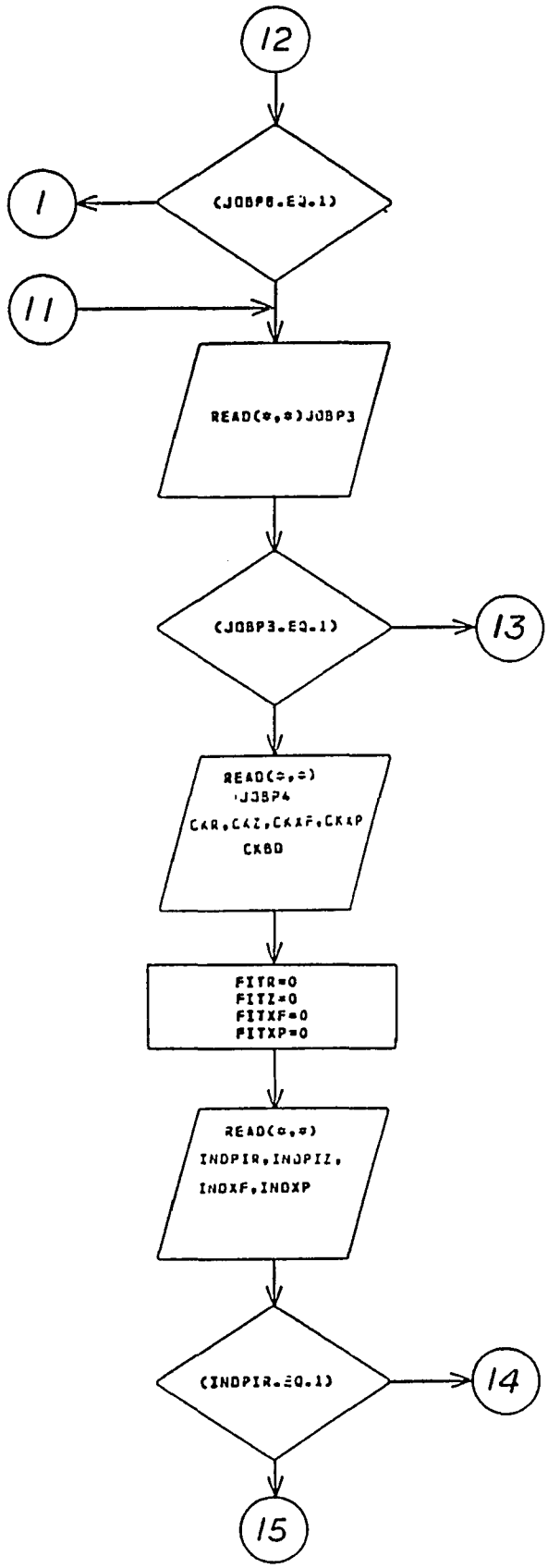

3
ODTAU=ODV/ODDP
GAMMA=(DDDP+ODJCL)/ODDF
ODEC=ODVL/ODV
READ(0,0)
DDRO,DDCS
·DDC
DDCF
DDF=DDDF/DDDP
PMT=(DDV-DDVL)*DDRO/DDV
DAKA=DJKA
DARP=DDRJ/DDCS
XC0=(DDC-DDCS)/DDCS
2
(JOBPS.EQ.1)
READ(0,0)
DAKG DAKA . D4G
DAPR GAMMA
DDKV DDEC DDP
XCP XC0
READ(0,0)
XP,XP,R,Z
(JOBPS.EQ.2)
4
5

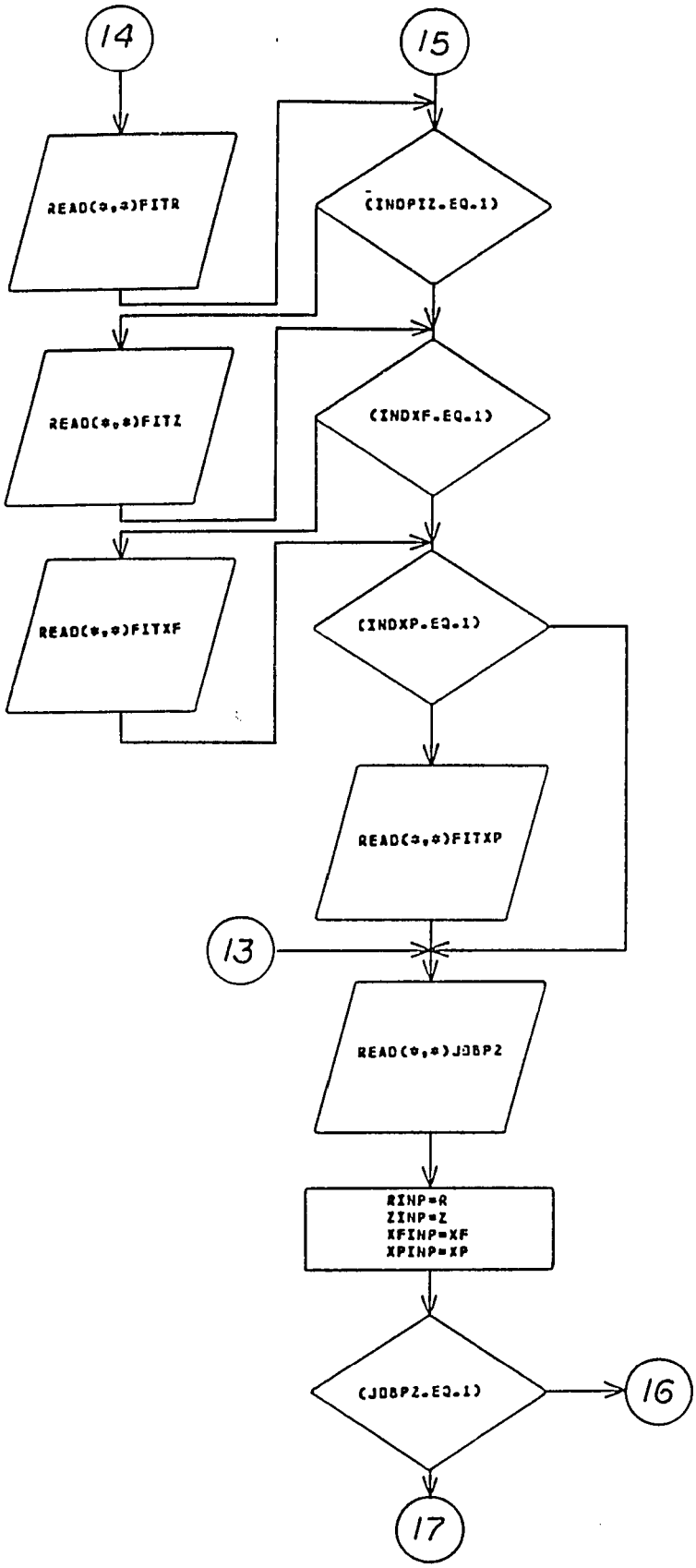

5
READ(*,*)
XLRANGE,TLRANGE
XRANGE=XLRAGE/DDV/DDTAU
TRANGE=TLRANGE/DDTAU
4
READ(*,*)JOBP10
(JOBP10.EQ.2)
READ(*,*)BRZ
BRZ=1
READ(*,*)
XRANGE,TRANGE
READ(*,*)
ITERX,ITERT
FJ,FI
6

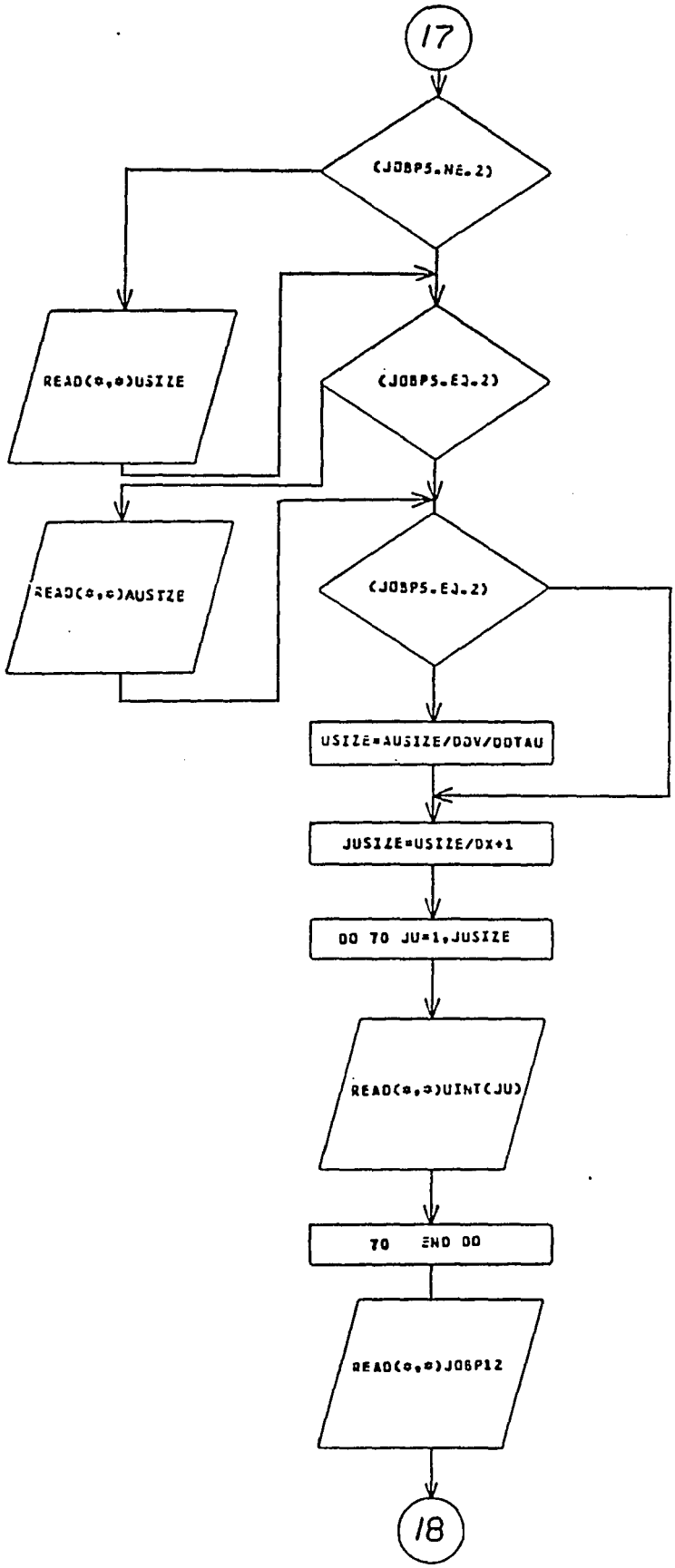

6
FII=FI-1
READ(*,*)TLUG
READ(*,*)TRLUG
(JOBP5.EQ.2)
TRLUG=TRLUG/DDTAU
READ(*,*)JOBP6
DX=PT(XRANGE/ITERX)
DT=PT(TRANGE/ITERT)
IM=ITERX+2
IP=ITERX+1
LUG=GONU(TRLUG/DT)
(JOBP10.EQ.1)
.OR.
(JOBP10.EQ.2)
7
CALL STEO(R,Z,
XF,XP,JCFS,JCPS,U)
8

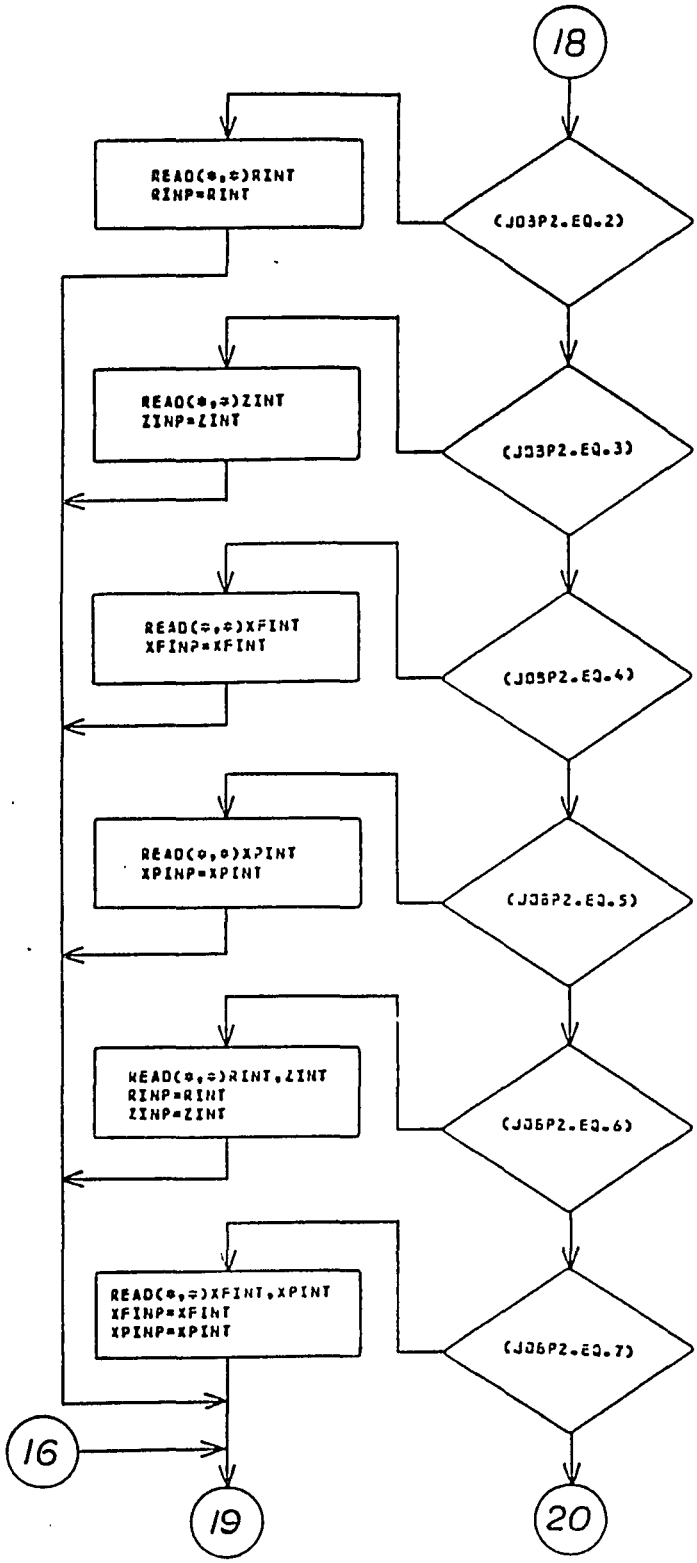

8
CALL PARA(R,Z,XF,XP,
IP,JCFS,DP,BETA,
F30,FINIO,FZO,U)
7
(JOBP10.EQ.3)
CALL STEDCR,Z,XF,
XP,JCFS,JCPS,U)
CALL PARA(R,Z,XF,XP,
IP,JCFS,DP,BETA,F30,
FINIO,FZO,U)
CALL STEDZR(XF,XP,
JCFS,JCPS,U)
CALL STEDEL(XF,XP,
JCFS,JCPS,U)
(FINIO.EQ.0)
FINIO=1
(JOBPS.EQ.3)
9
10

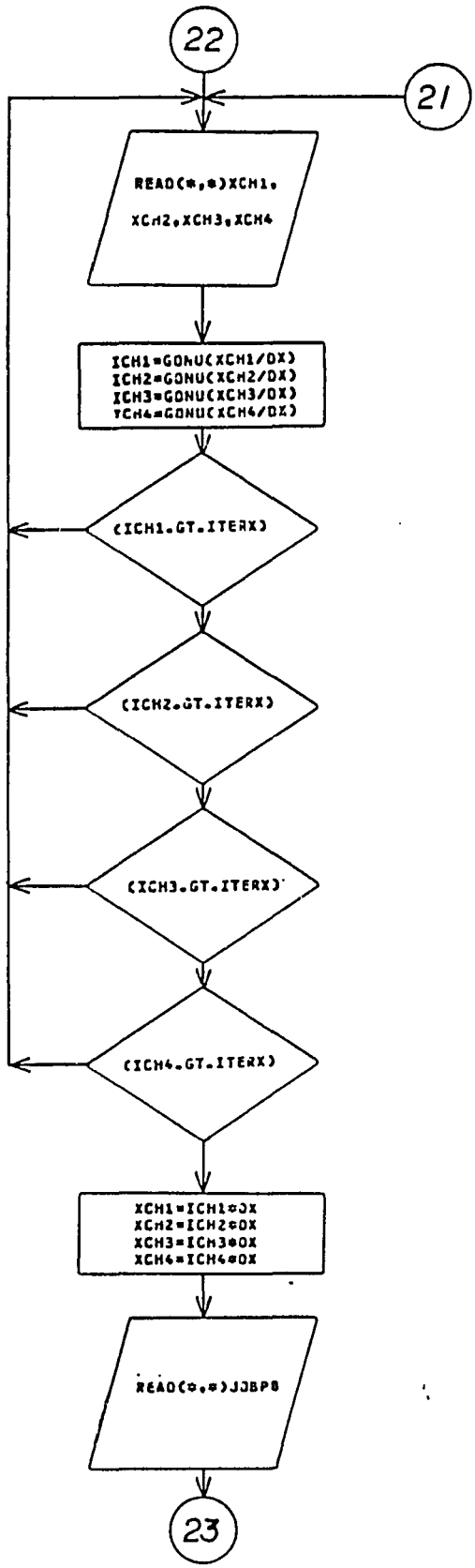

10
PH=1
9
CALL DOCOND(FINIO,F20,
JFLAG1,J08P9,XC,0)
1
(JFLAG1.EQ.1)
PH0=PT(DAKG*XCM)
PH2XT=PT(DAKG*XC/PH0)
PH=1
QUE=-PH/(12*DX)
(J08P5.EQ.3)
11
(J08P5.EQ.2)
GO=PT(DJCS*ODKG/DAKG)
ANNO=ODKN=DDEC*GO==FII
*PHT**FJ
READ(*,*)J08P8
12

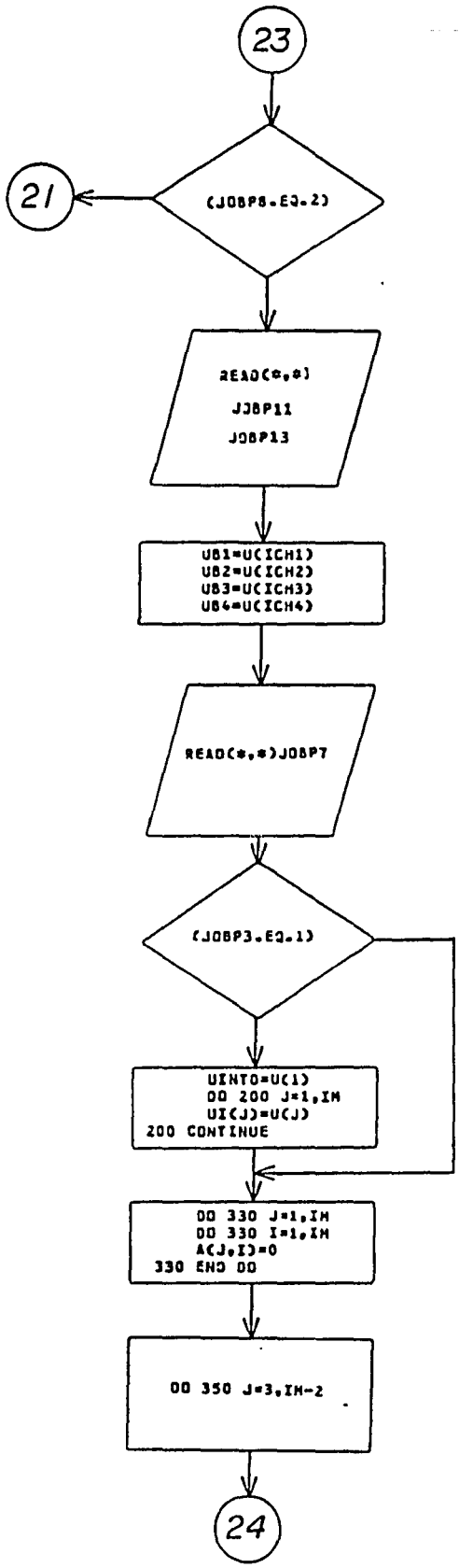

12
1
(JOBP8.EQ.1)
11
READ(*,*)JOBP3
(JOBP3.EQ.1)
13
READ(*,*)
JOBP4
CAR,CAI,CKXF,CKXP
CKBD
FITR=0
FITI=0
FITXF=0
FITXP=0
READ(*,*)
INDPIR,INDPII,
INDXF,INDXP
(INDPIR.EQ.1)
14
15

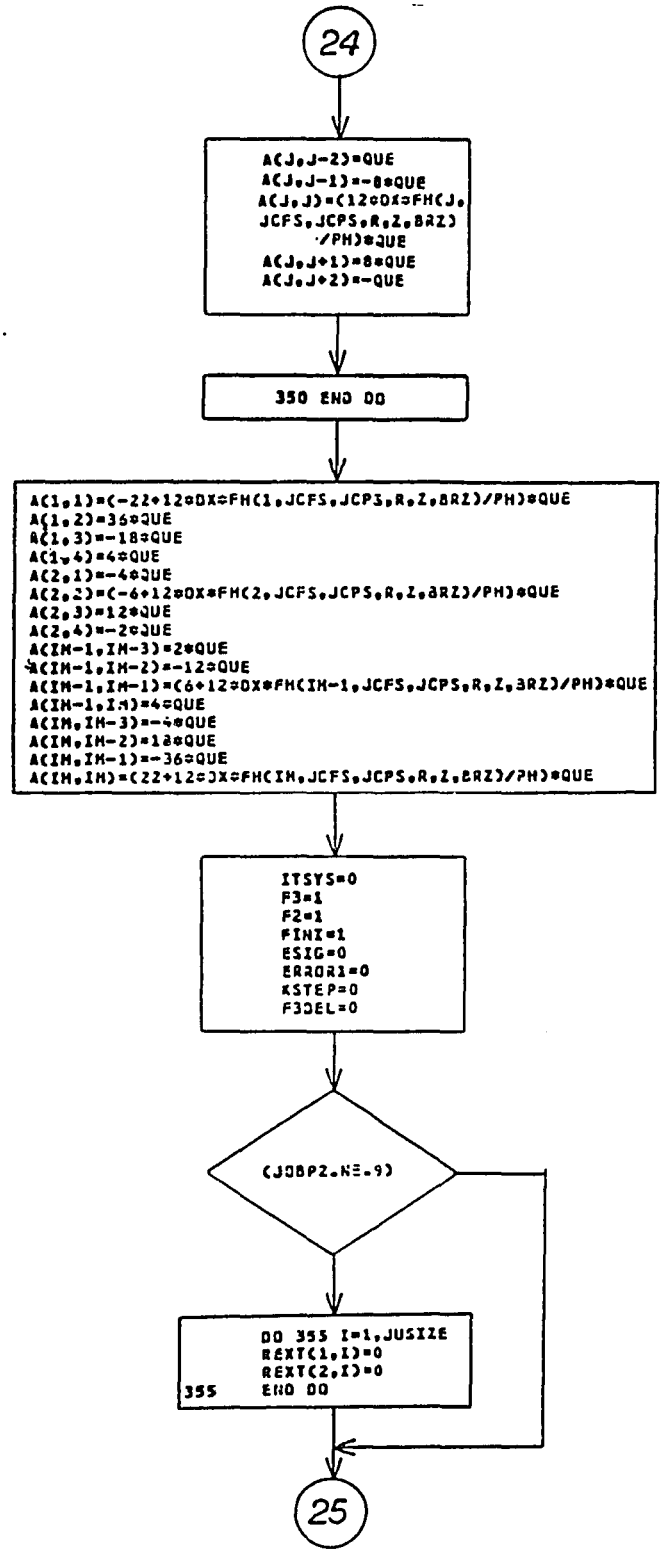

14
15
READ(*,*)FITR
(INDPIZ.EQ.1)
READ(*,*)FITZ
(INDXF.EQ.1)
READ(*,*)FITXF
(INDXP.EQ.1)
READ(*,*)FITXP
13
READ(*,*)JOBP2
RINP=R
ZINP=Z
XFINP=XF
XPINP=XP
(JOBP2.EQ.1)
16
17

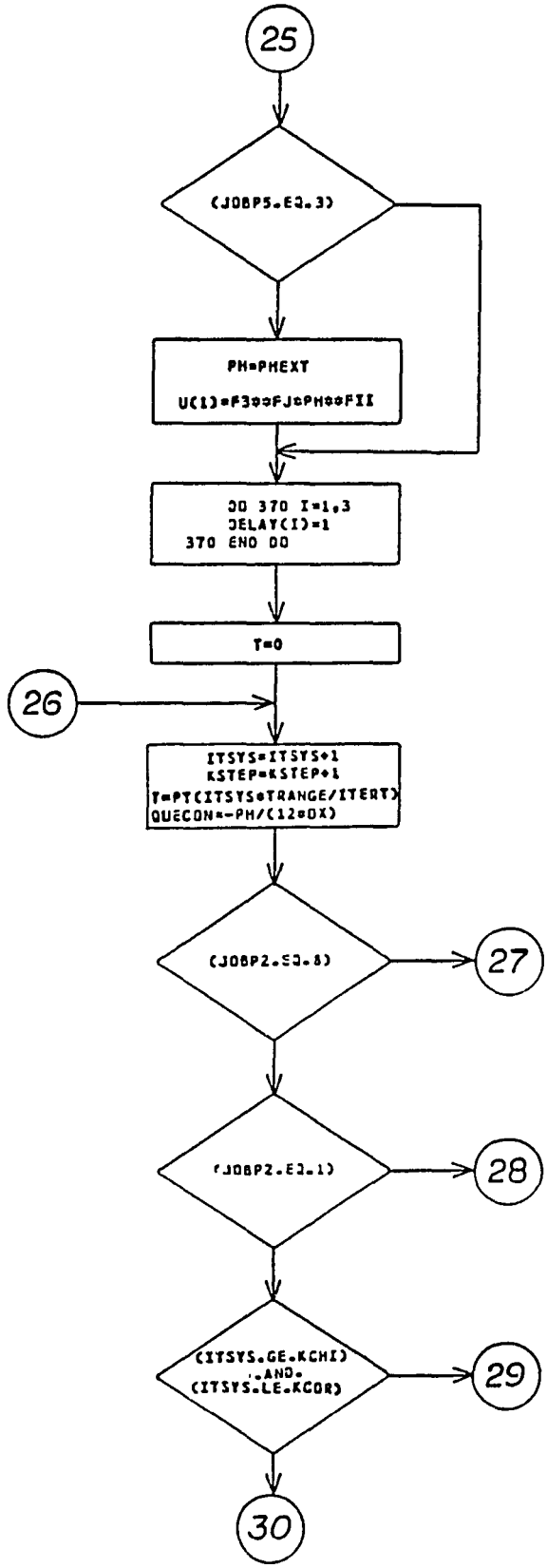

17
(JOBPS.NE.2)
READ(*,*)USIZE
(JOBPS.EQ.2)
READ(*,*)AUSIZE
(JOBPS.EQ.2)
USIZE=AUSIZE/DDV/DDTAU
JUSIZE=USIZE/DX+1
DO 70 JU=1,JUSIZE
READ(*,*)UINT(JU)
70 END DO
READ(*,*)JOBP12
18

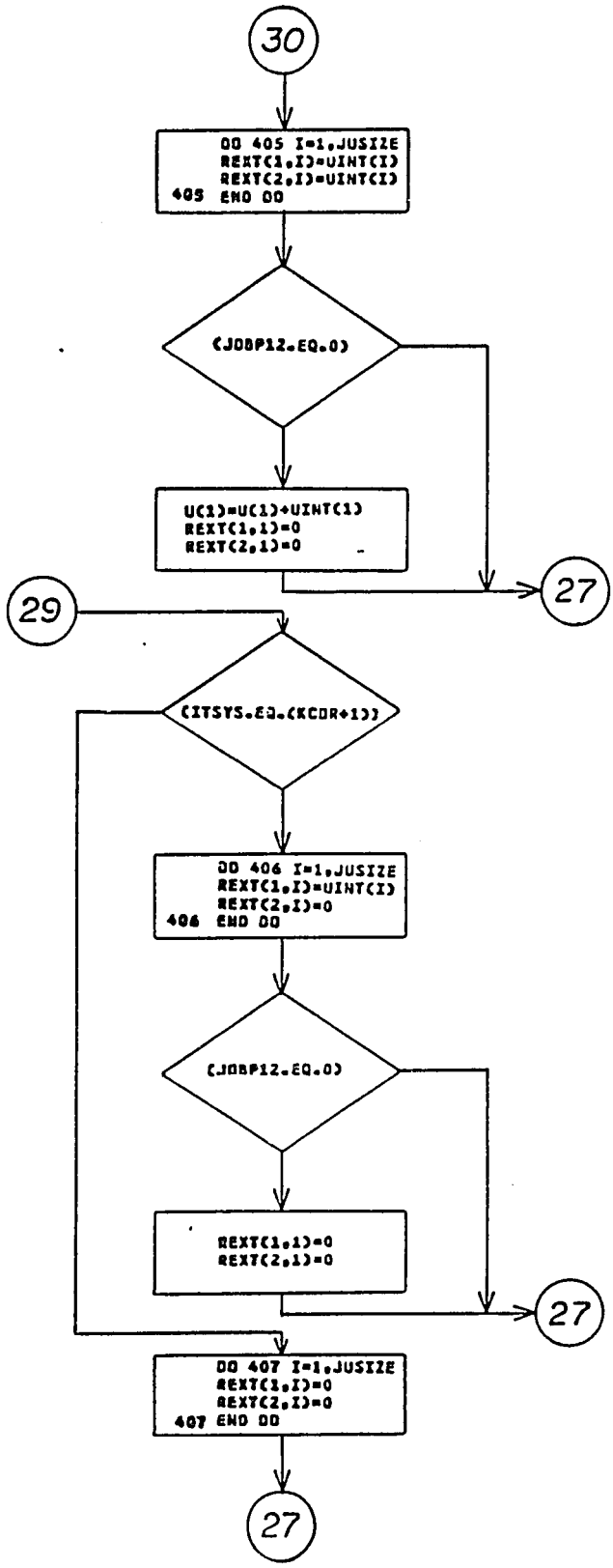

18
READ(*,*)RINT
RINP=RINT
(JO3P2.EQ.2)
READ(*,*)ZINT
ZINP=ZINT
(JO3P2.EQ.3)
READ(*,*)XFINT
XFINP=XFINT
(JO5P2.EQ.4)
READ(*,*)XPINT
XPINP=XPINT
(JO6P2.EQ.5)
READ(*,*)RINT,ZINT
RINP=RINT
ZINP=ZINT
(JO6P2.EQ.6)
READ(*,*)XFINT,XPINT
XFINP=XFINT
XPINP=XPINT
(JO6P2.EQ.7)
16
19
20

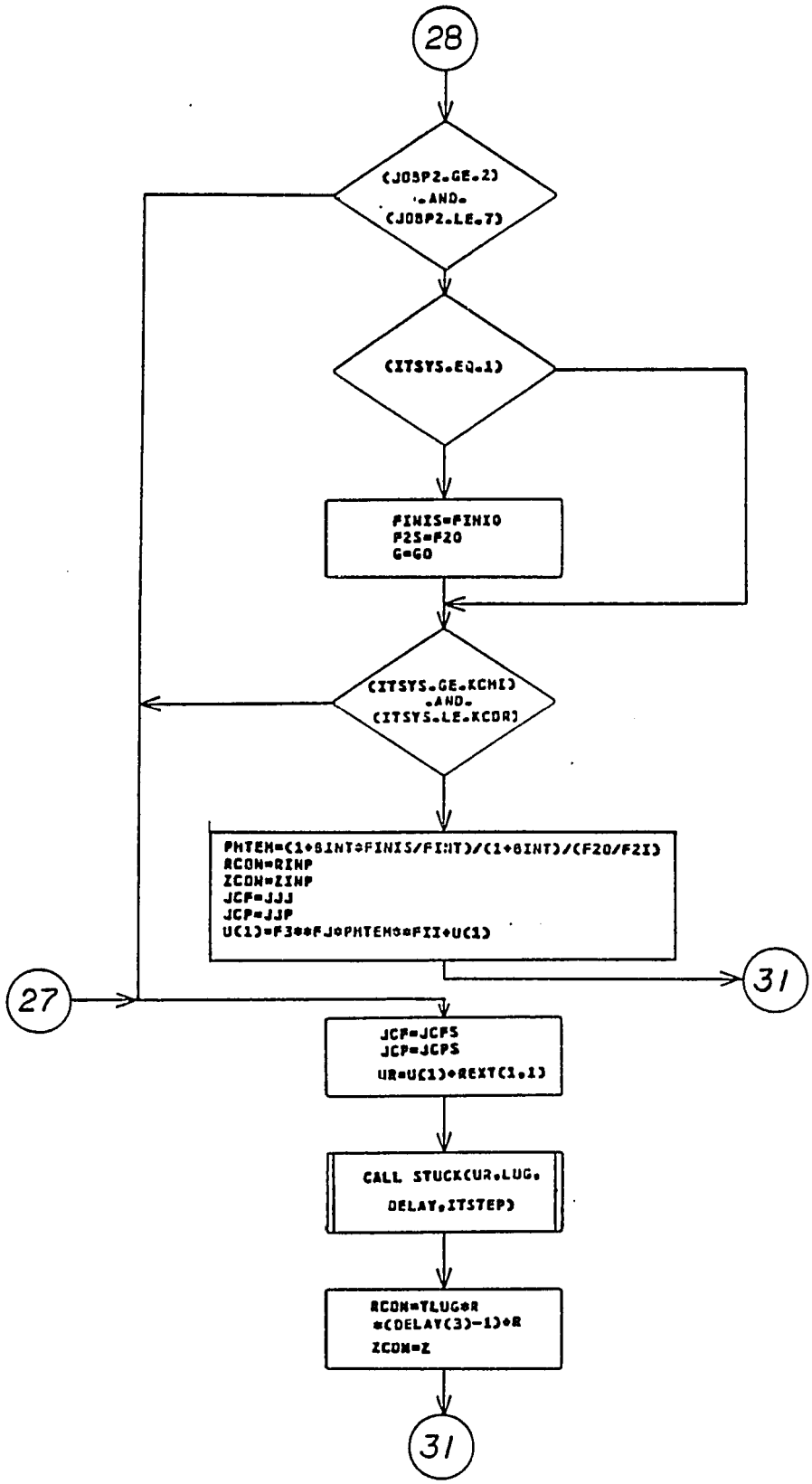

22
21
READ(*,*)XCH1,
XCH2,XCH3,XCH4
ICH1=GONU(XCH1/DX)
ICH2=GONU(XCH2/DX)
ICH3=GONU(XCH3/DX)
ICH4=GONU(XCH4/DX)
(ICH1.GT.ITERX)
(ICH2.GT.ITERX)
(ICH3.GT.ITERX)
(ICH4.GT.ITERX)
XCH1=ICH1*DX
XCH2=ICH2*DX
XCH3=ICH3*DX
XCH4=ICH4*DX
READ(*,*)JJBP8
23

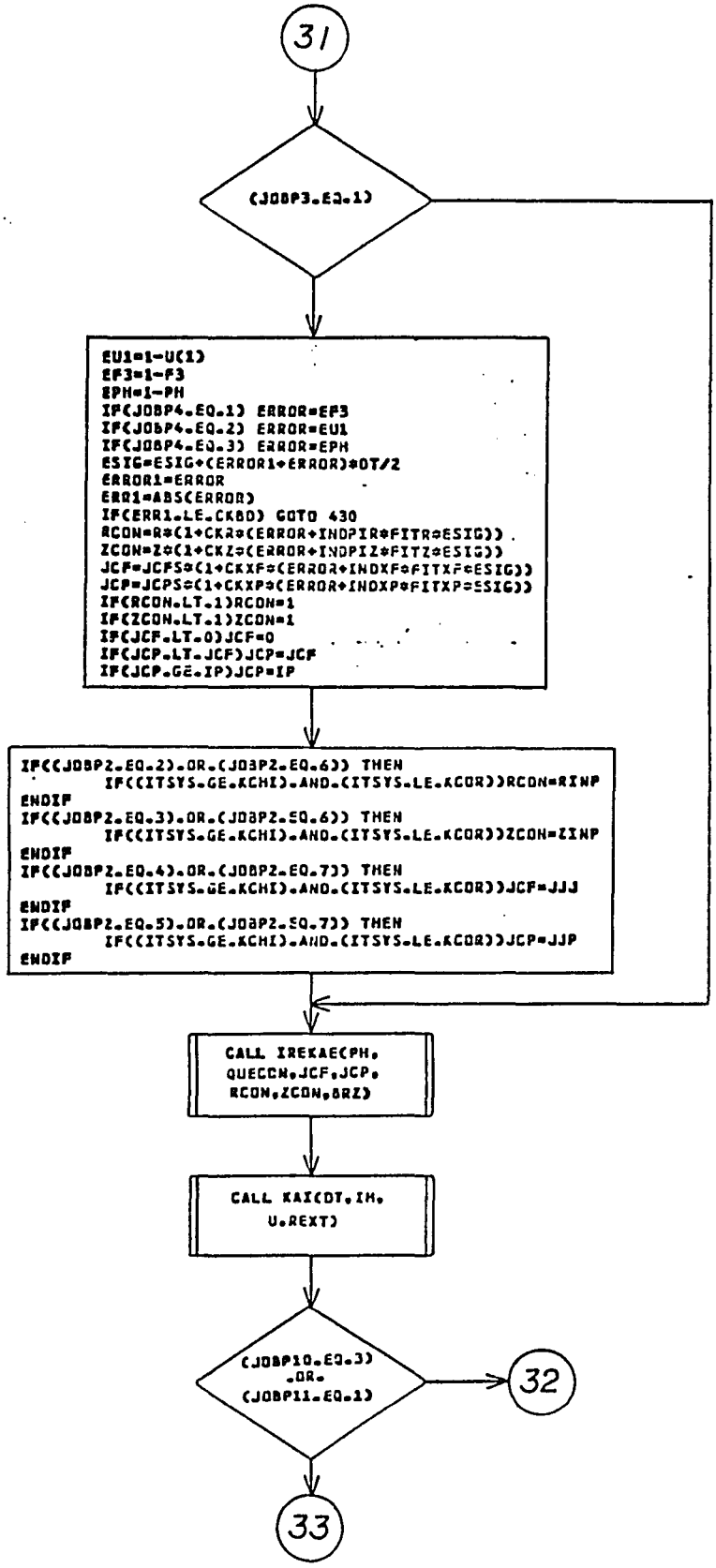

23
21
(JOBP8.EQ.2)
READ(*,*)
JOBP11
JOBP13
UB1=U(ICH1)
UB2=U(ICH2)
UB3=U(ICH3)
UB4=U(ICH4)
READ(*,*)JOBP7
(JOBP3.EQ.1)
UINTO=U(1)
DO 200 J=1,IM
UI(J)=U(J)
200 CONTINUE
DO 330 J=1,IM
DO 330 I=1,IM
A(J,I)=0
330 END DO
DO 350 J=3,IM-2
24

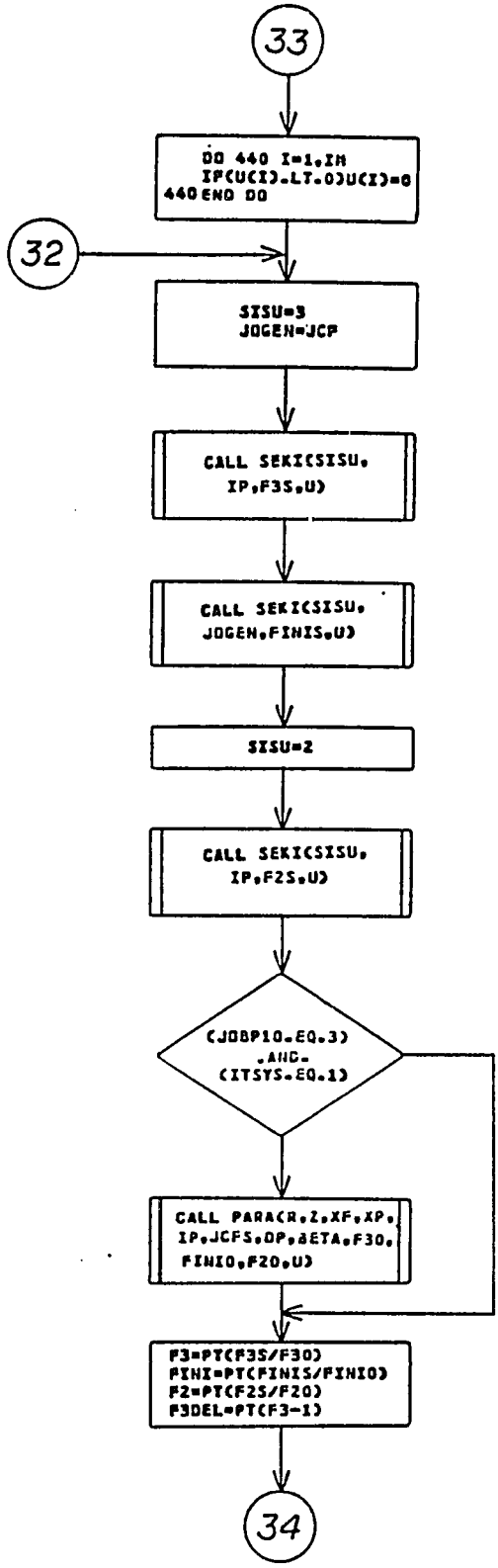

24
A(J,J-2)=QUE
A(J,J-1)=-8*QUE
A(J,J)=(12*DX*FH(J,
JCFS,JCPS,R,Z,BRZ)
/PH)*QUE
A(J,J+1)=8*QUE
A(J,J+2)=-QUE
350 END DO
A(1,1)=(-22+12*DX*FH(1,JCFS,JCPS,R,Z,BRZ)/PH)*QUE
A(1,2)=36*QUE
A(1,3)=-18*QUE
A(1,4)=4*QUE
A(2,1)=-4*QUE
A(2,2)=(-6+12*DX*FH(2,JCFS,JCPS,R,Z,BRZ)/PH)*QUE
A(2,3)=12*QUE
A(2,4)=-2*QUE
A(IM-1,IM-3)=2*QUE
A(IM-1,IM-2)=-12*QUE
A(IM-1,IM-1)=(6+12*DX*FH(IM-1,JCFS,JCPS,R,Z,BRZ)/PH)*QUE
A(IM-1,IM)=4*QUE
A(IM,IM-3)=-4*QUE
A(IM,IM-2)=12*QUE
A(IM,IM-1)=-36*QUE
A(IM,IM)=(22+12*DX*FH(IM,JCFS,JCPS,R,Z,BRZ)/PH)*QUE
ITSYS=0
F3=1
F2=1
FINI=1
ESIG=0
ERROR1=0
KSTEP=0
F3DEL=0
(JOBPZ.NE.9)
DO 355 I=1,JUSIZE
REXT(1,I)=0
REXT(2,I)=0
355 END DO
25

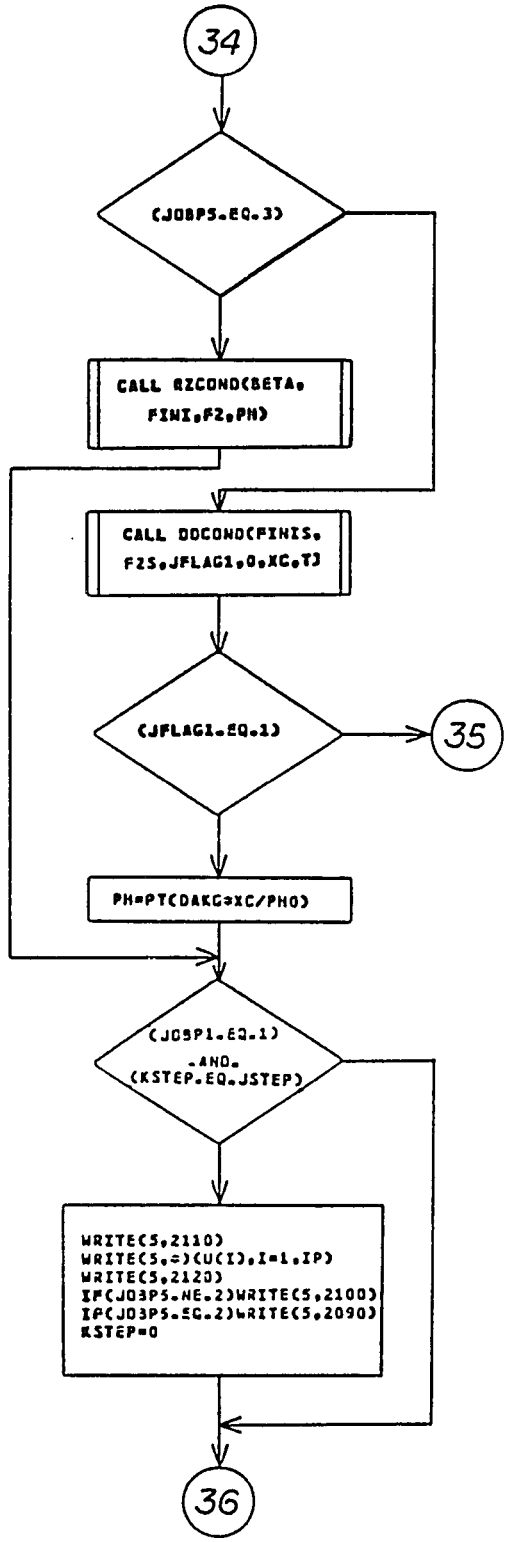

25
(JOBP5.EQ.3)
PH=PHEXT
U(I)=F3**FJ*PH**FII
DO 370 I=1,3
DELAY(I)=1
370 END DO
T=0
26
ITSYS=ITSYS+1
KSTEP=KSTEP+1
T=PT(ITSYS*TRANGE/ITERT)
QUECON=-PH/(12*DX)
(JOBP2.EQ.8)
27
(JOBP2.EQ.1)
28
(ITSYS.GE.KCHI)
.AND.
(ITSYS.LE.KCOR)
29
30

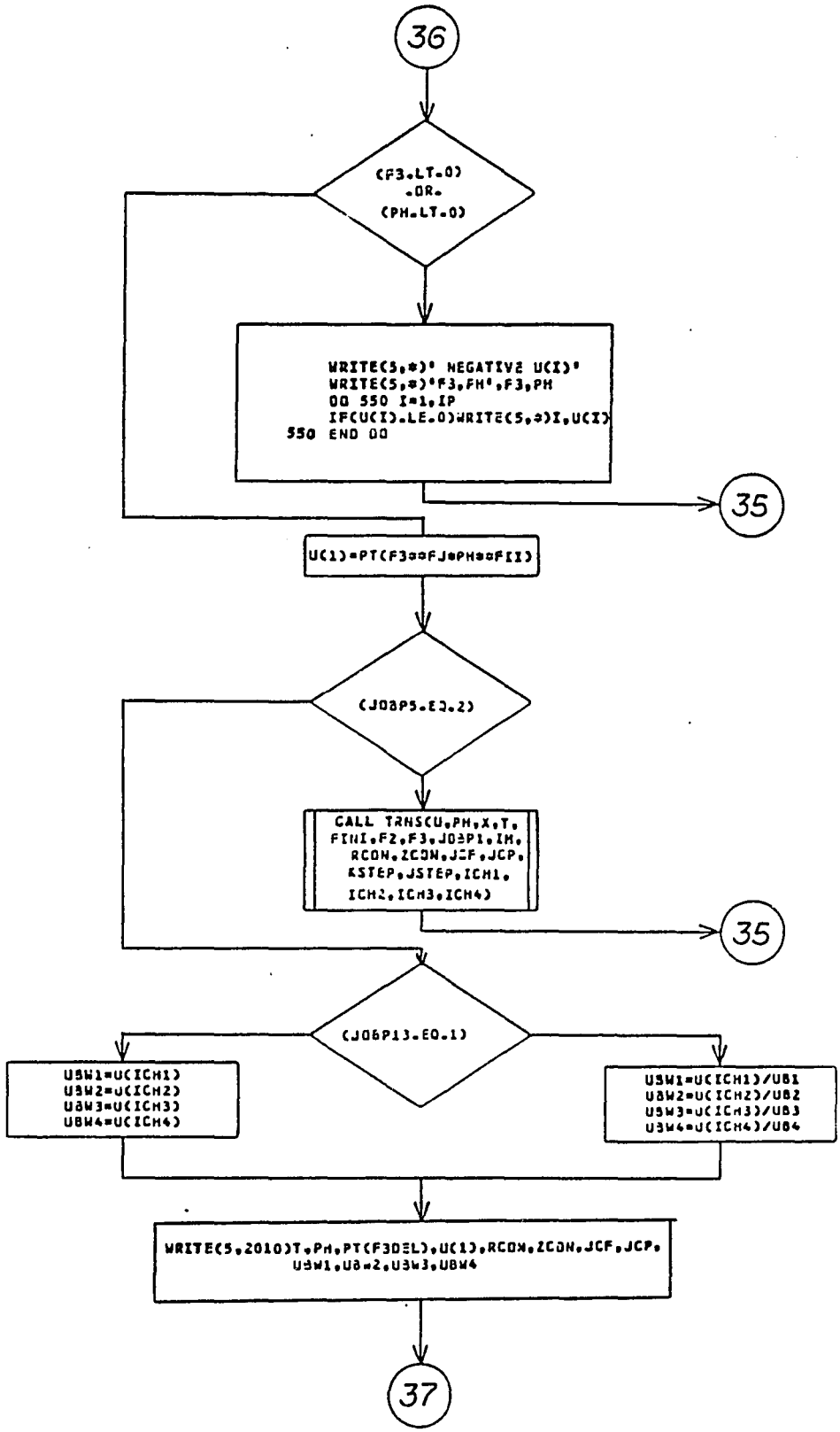

30
DO 405 I=1,JUSIZE
REXT(1,I)=UINT(I)
REXT(2,I)=UINT(I)
405 END DO
(JOBP12.EQ.0)
U(1)=U(1)+UINT(1)
REXT(1,1)=0
REXT(2,1)=0
27
29
(ITSYS.EQ.(KCOR+1))
DO 406 I=1,JUSIZE
REXT(1,I)=UINT(I)
REXT(2,I)=0
406 END DO
(JOBP12.EQ.0)
REXT(1,1)=0
REXT(2,1)=0
27
DO 407 I=1,JUSIZE
REXT(1,I)=0
REXT(2,I)=0
407 END DO
27

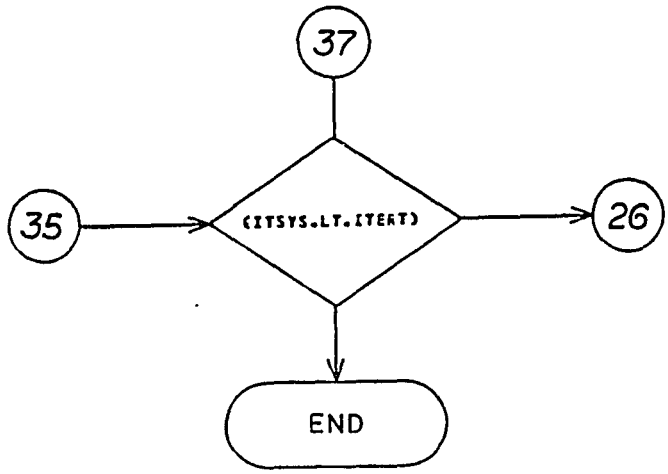

28
(JOBP2.GE.2)
.AND.
(JOBP2.LE.7)
(ITSYS.EQ.1)
FINIS=FINIO
F2S=F2O
G=GO
(ITSYS.GE.KCHI)
.AND.
(ITSYS.LE.KCOR)
PHTEM=(1+BINT*FINIS/FINT)/(1+BINT)/(F2O/F2I)
RCON=RINP
ZCON=ZINP
JCF=JJJ
JCP=JJP
U(1)=F3**FJ*PHTEM**FII*U(1)
31
27
JCF=JCFS
JCP=JCPS
UR=U(1)*REXT(1,1)
CALL STUCK(UR,LUG,
DELAY,ITSTEP)
RCON=YLUG*R
*(DELAY(3)-1)*R
ZCON=Z
31


31
(JOBP3.EQ.1)
EU1=1-U(1)
EF3=1-F3
EPH=1-PH
IF(JOBP4.EQ.1) ERROR=EF3
IF(JOBP4.EQ.2) ERROR=EU1
IF(JOBP4.EQ.3) ERROR=EPH
ESIG=ESIG+(ERROR1+ERROR)*DT/2
ERROR1=ERROR
ERR1=ABS(ERROR)
IF(ERR1.LE.CK80) GOTO 430
RCON=R*(1+CK2*(ERROR+INDPIR*FITR*ESIG))
ZCON=Z*(1+CK2*(ERROR+INDPIZ*FITZ*ESIG))
JCF=JCFS*(1+CKXF*(ERROR+INDXF*FITXF*ESIG))
JCP=JCPS*(1+CKXP*(ERROR+INDXP*FITXP*ESIG))
IF(RCON.LT.1)RCON=1
IF(ZCON.LT.1)ZCON=1
IF(JCF.LT.0)JCF=0
IF(JCP.LT.JCF)JCP=JCF
IF(JCP.GE.IP)JCP=IP
IF((JOBP2.EQ.2).OR.(JOBP2.EQ.6)) THEN
IF((ITSYS.GE.KCHI).AND.(ITSYS.LE.KCOR))RCON=RINP
ENDIF
IF((JOBP2.EQ.3).OR.(JOBP2.EQ.6)) THEN
IF((ITSYS.GE.KCHI).AND.(ITSYS.LE.KCOR))ZCON=ZINP
ENDIF
IF((JOBP2.EQ.4).OR.(JOBP2.EQ.7)) THEN
IF((ITSYS.GE.KCHI).AND.(ITSYS.LE.KCOR))JCF=JJJ
ENDIF
IF((JOBP2.EQ.5).OR.(JOBP2.EQ.7)) THEN
IF((ITSYS.GE.KCHI).AND.(ITSYS.LE.KCOR))JCP=JJP
ENDIF
CALL IREKAE(PH,
QUECON,JCF,JCP,
RCON,ZCON,BRZ)
CALL KAI(DT,IM,
U,REXT)
(JOBP10.EQ.3)
.OR.
(JOBP11.EQ.1)
32
33

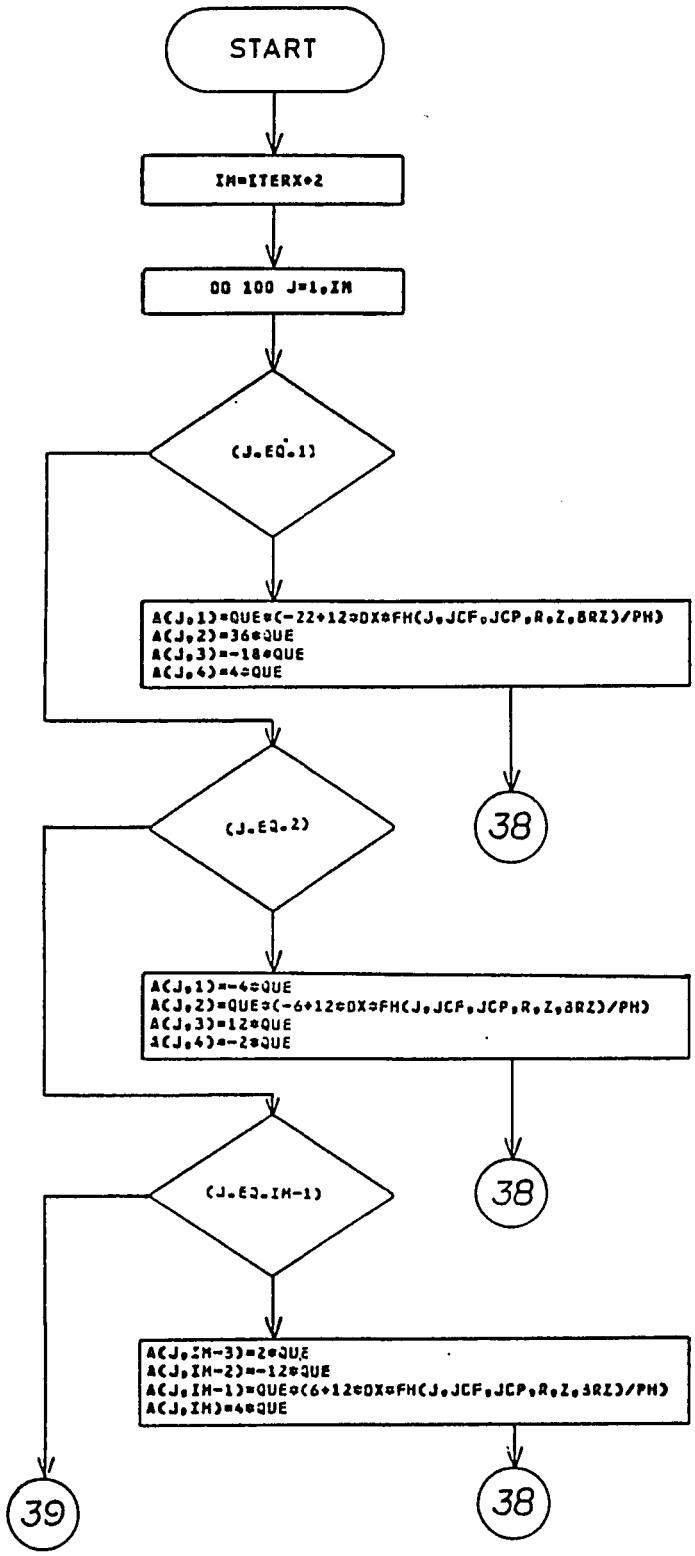

33
DO 440 I=1,IM
IF(U(I).LT.0)U(I)=0
440 END DO
32
SISU=3
JOGEN=JCP
CALL SEKI(SISU,
IP,F3S,U)
CALL SEKI(SISU,
JOGEN,FINIS,U)
SISU=2
CALL SEKI(SISU,
IP,F2S,U)
(JOBP10.EQ.3)
.AND.
(ITSYS.EQ.1)
CALL PARA(R,Z,XF,XP,
IP,JCFS,DP,BETA,F30,
FINIO,F20,U)
F3=PT(F3S/F30)
FINI=PT(FINIS/FINIO)
F2=PT(F2S/F20)
F3DEL=PT(F3-1)
34

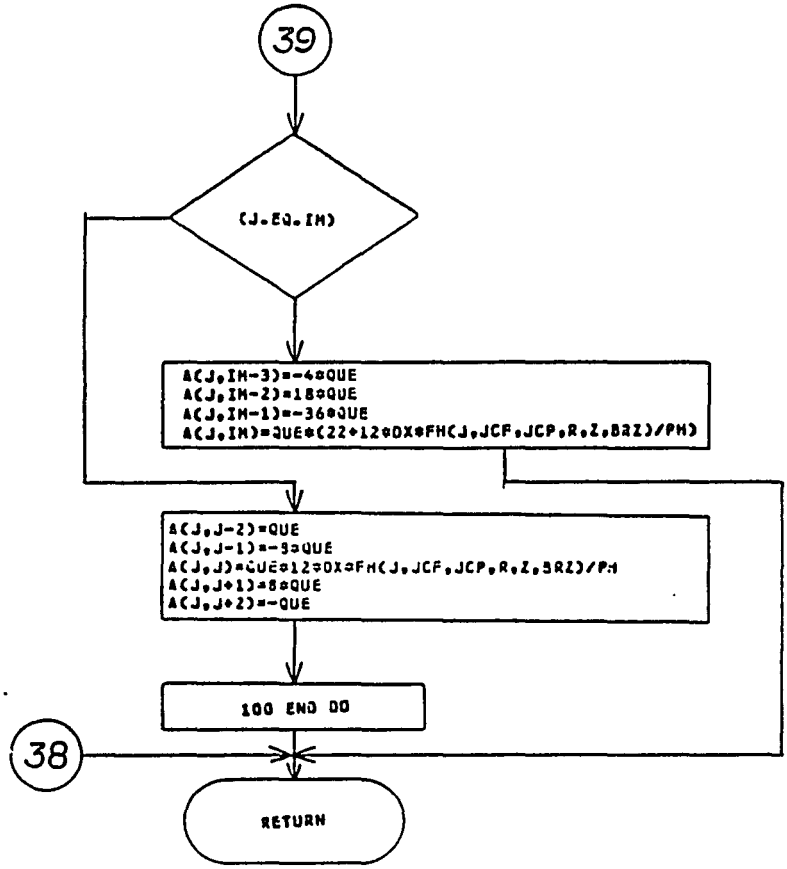

34
(JOBP5.EQ.3)
CALL RZCOND(BETA,
FINI,F2,PH)
CALL DOCOND(FINIS,
F2S,JFLAG1,0,XC,T)
(JFLAG1.EQ.1)
35
PH=PT(DAKG*XC/PH0)
(JOBP1.EQ.1)
.AND.
(KSTEP.EQ.JSTEP)
WRITE(5,2110)
WRITE(5,*)(U(I),I=1,IP)
WRITE(5,2120)
IF(JOBP5.NE.2)WRITE(5,2100)
IF(JOBP5.EQ.2)WRITE(5,2090)
KSTEP=0
36


36
(F3.LT.0)
.OR.
(PH.LT.0)
WRITE(5,*)' NEGATIVE U(I)'
WRITE(5,*)'F3,PH',F3,PH
DO 550 I=1,IP
IF(U(I).LE.0)WRITE(5,*)I,U(I)
550 END DO
35
U(1)=PT(F3**FJ*PH**FI)
(JOBP5.EQ.2)
CALL TRNSCU,PH,X,T,
FINI,F2,F3,JOBP1,IM,
RCON,ZCON,JCF,JCP,
KSTEP,JSTEP,ICH1,
ICH2,ICH3,ICH4)
35
(JOBP13.EQ.1)
U8W1=U(ICH1)
U8W2=U(ICH2)
U8W3=U(ICH3)
U8W4=U(ICH4)
U8W1=U(ICH1)/UB1
U8W2=U(ICH2)/UB2
U8W3=U(ICH3)/UB3
U8W4=U(ICH4)/UB4
WRITE(5,2010)T,PH,PT(F3DEL),U(1),RCON,ZCON,JCF,JCP,
U8W1,U8W2,U8W3,U8W4
37

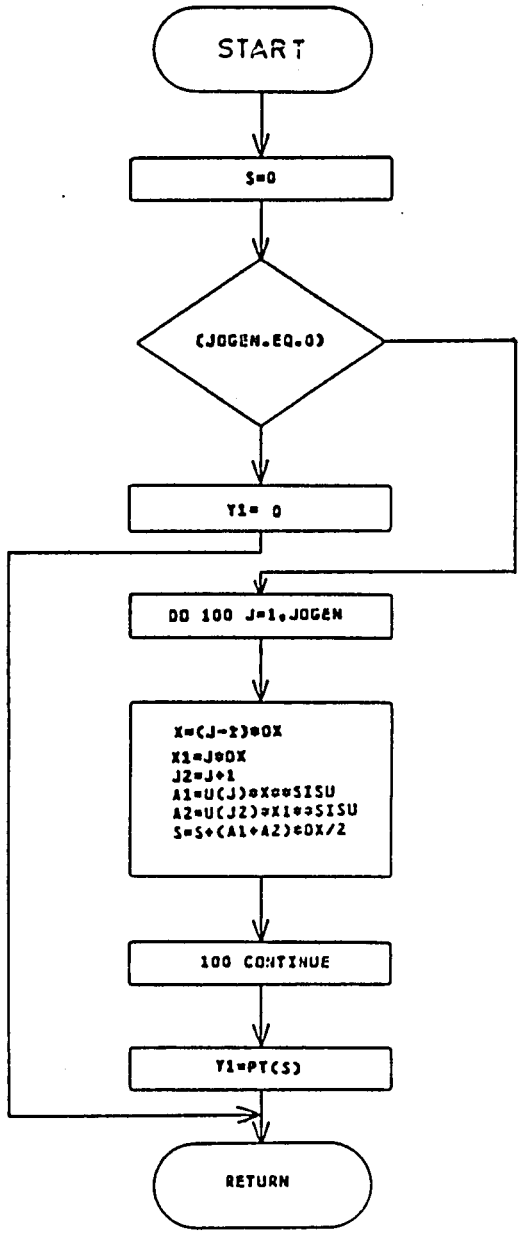

37
35
(ITSYS.LT.ITEXT)
26
END

SUBROUTINE IREKA6(PH,QUE,JCF,JCP,R,Z,BRZ)

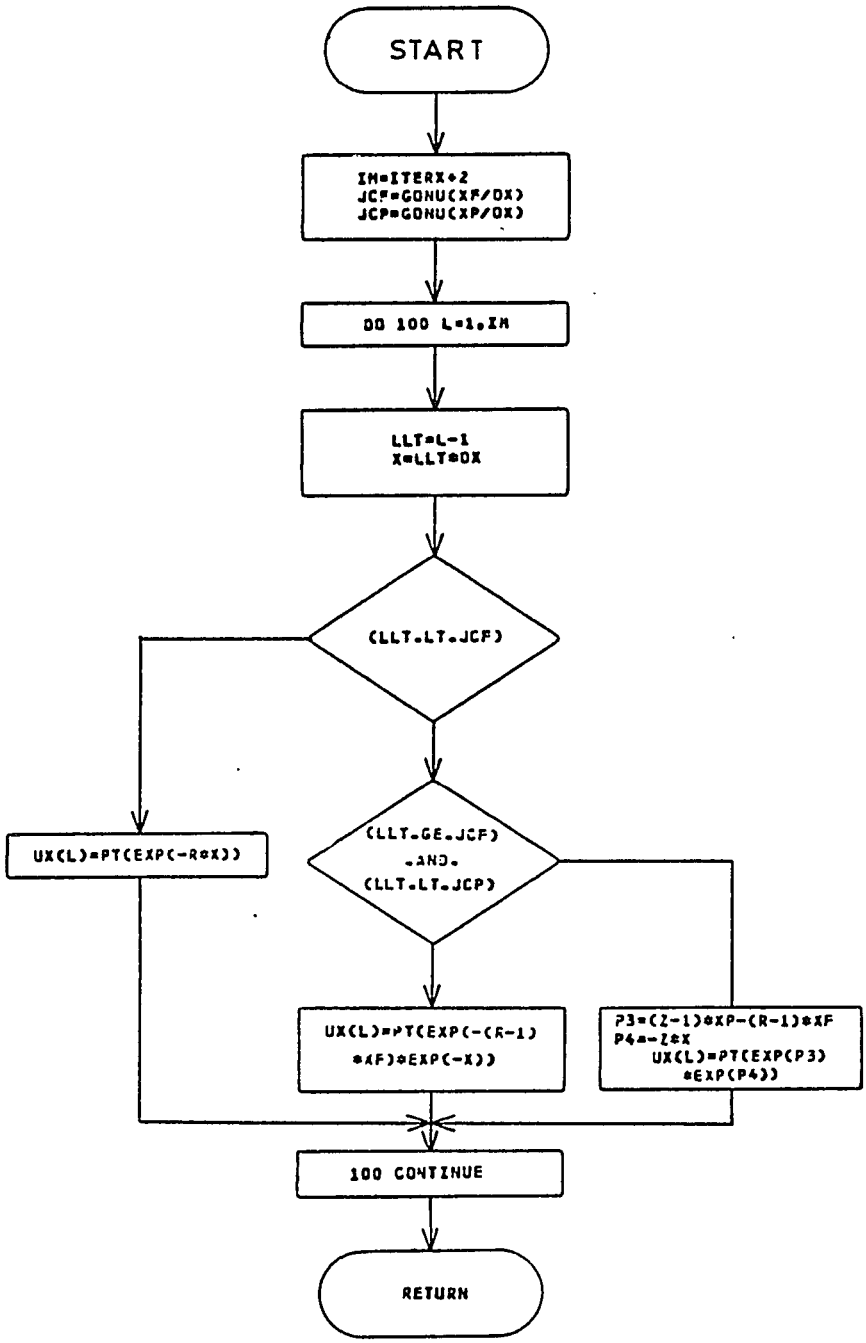



39
(J.EQ.IM)
A(J,IM-3)=-4*QUE
A(J,IM-2)=18*QUE
A(J,IM-1)=-36*QUE
A(J,IM)=QUE*(22+12*DX*FH(J,JCF,JCP,R,Z,BRZ)/PH)
A(J,J-2)=QUE
A(J,J-1)=-9*QUE
A(J,J)=QUE*12*DX*FH(J,JCF,JCP,R,Z,BRZ)/PH
A(J,J+1)=8*QUE
A(J,J+2)=-QUE
100 END DO
38
RETURN

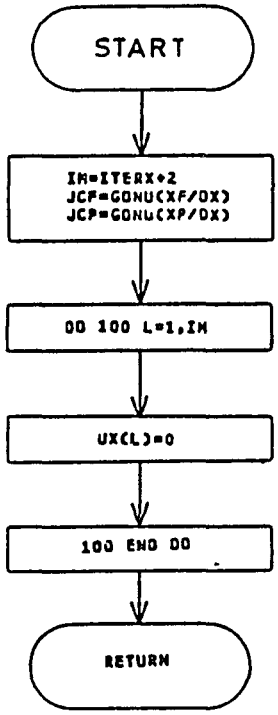
SUBROUTINE SEKI(SISU,JOGEN,Y1,U)



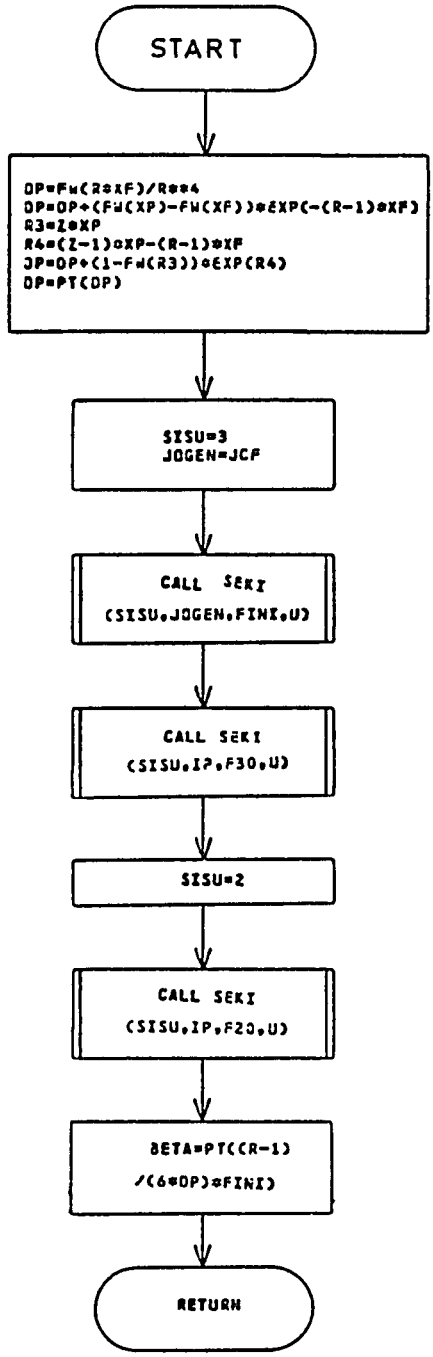
SUBROUTINE STED(R,Z,XF,XP,JCF,JCP,UX)



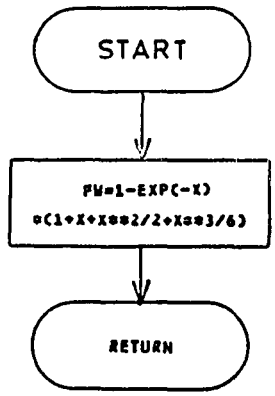
SUBROUTINE STEDIR(XP,XP,JCF,JCP,UX)



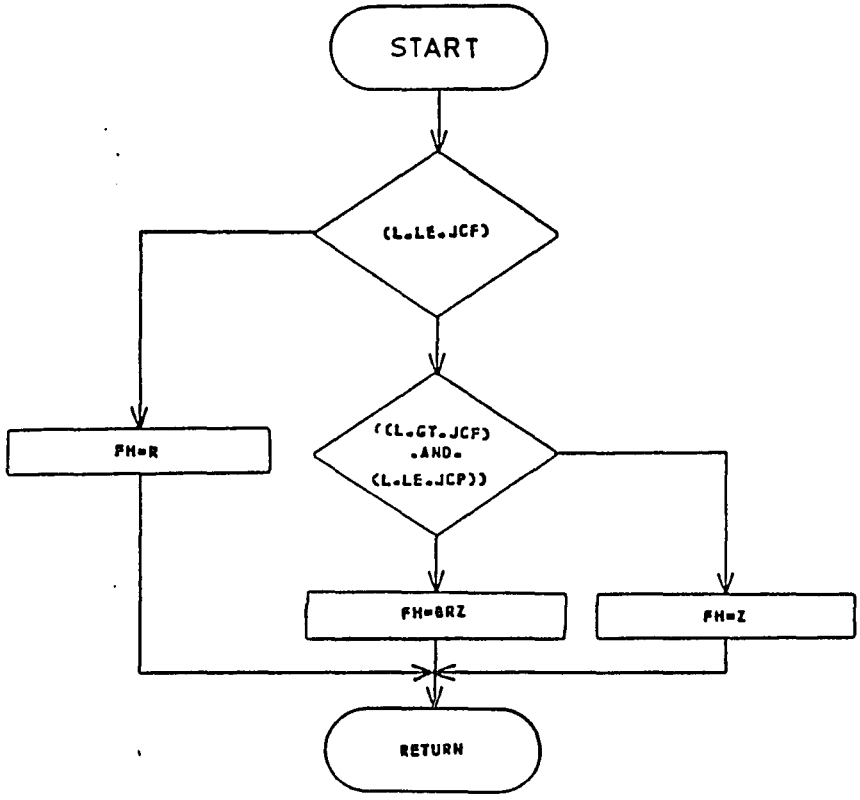
SUBROUTINE PARA(R,Z,XF,XP,IP,JCF,DP,BETA,F30,FINI,F20,U)



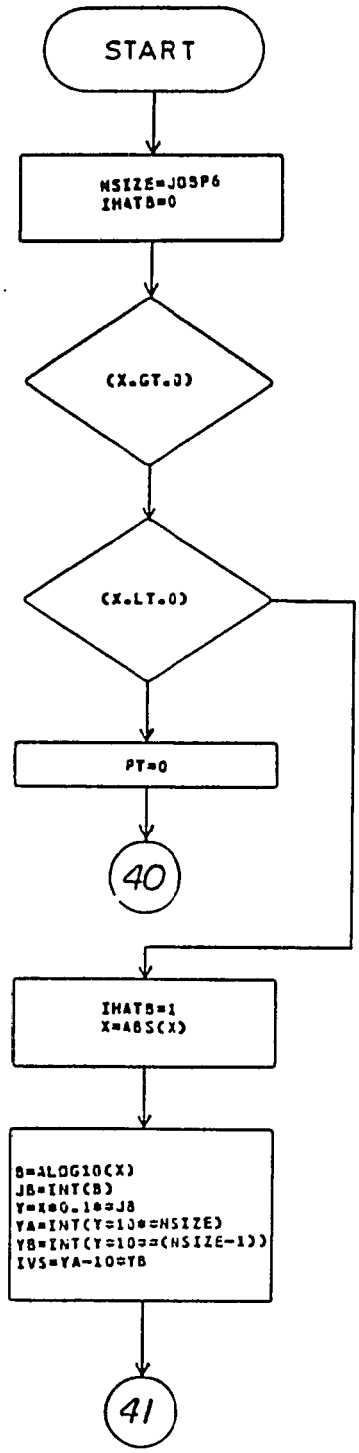
FUNCTION FW(X)

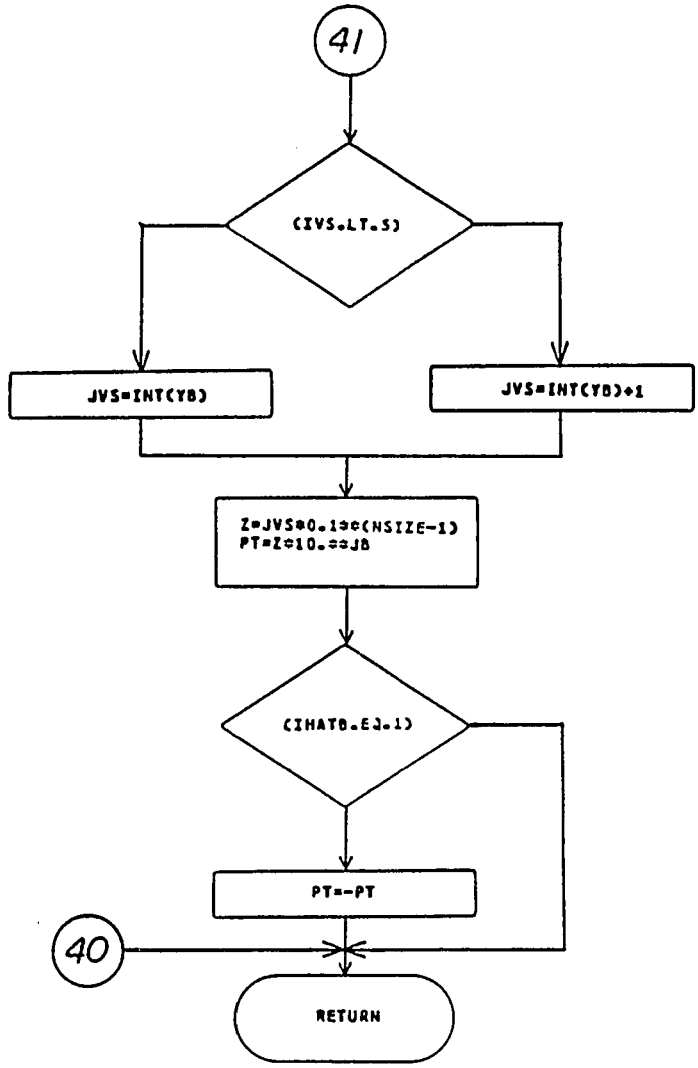


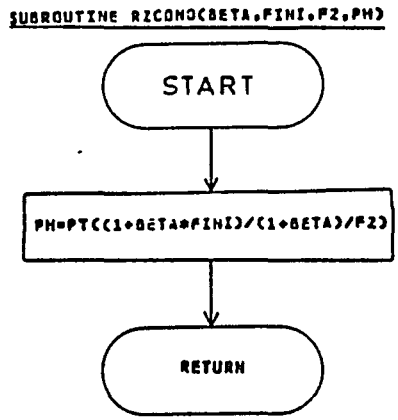
FUNCTION FH(L,JCF,JCP,R,Z,BRZ)



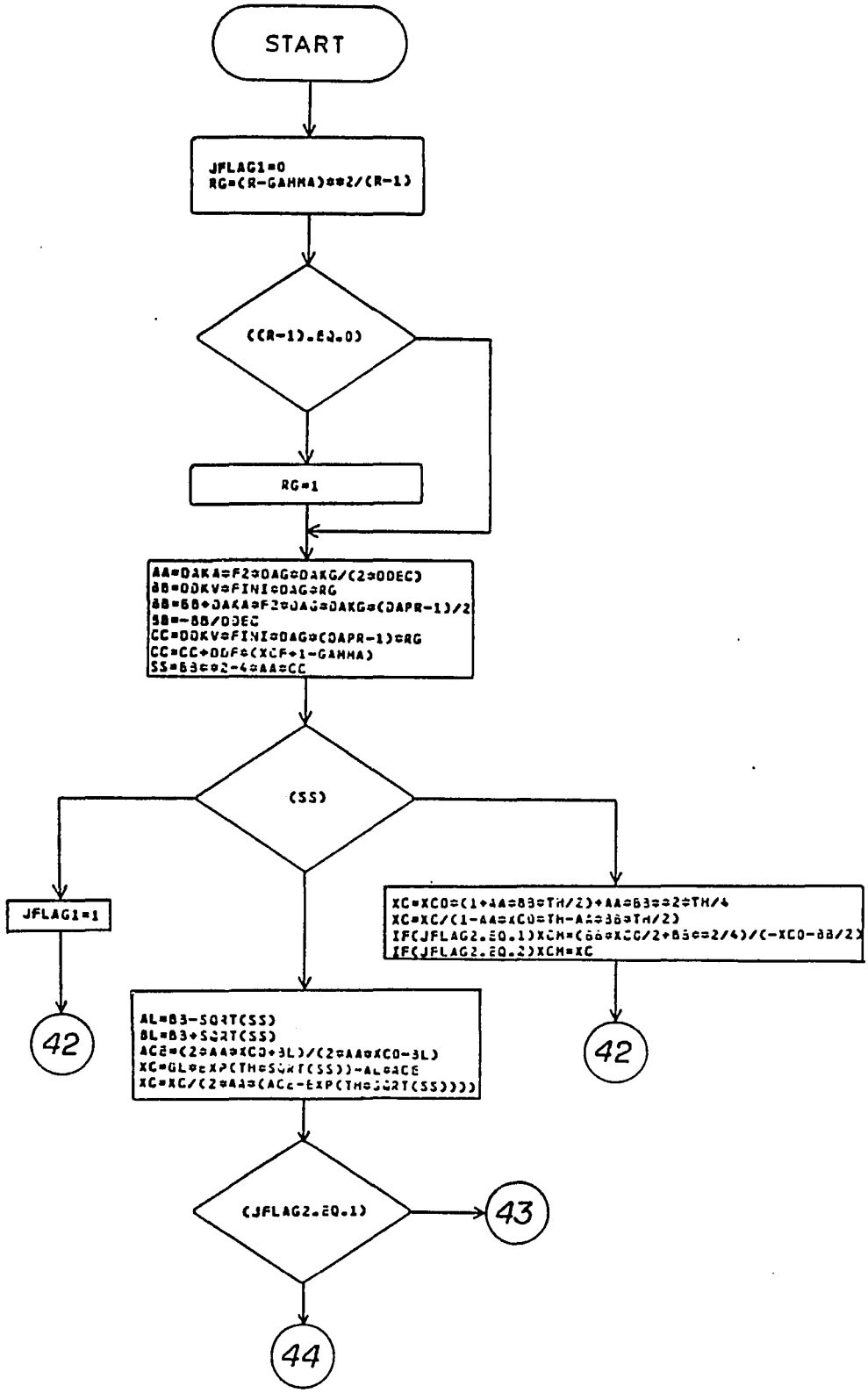
FUNCTION PT(X)

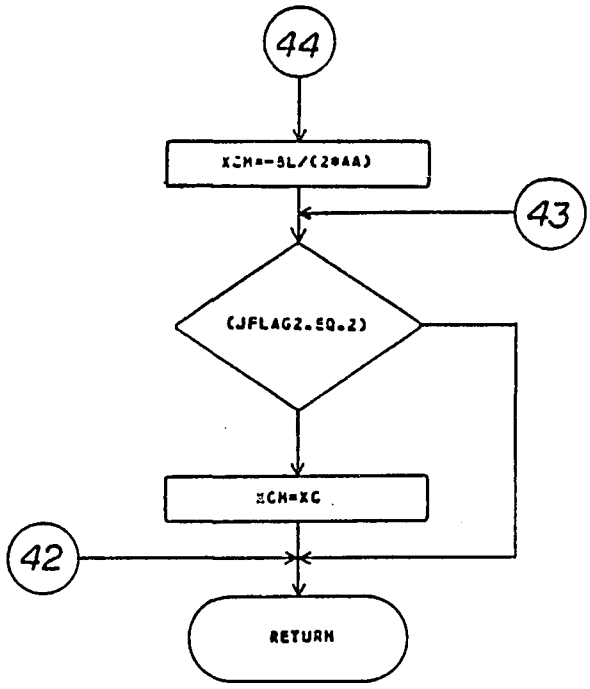



41
(IVS.LT.5)
JVS=INT(YB)
JVS=INT(YB)+1
Z=JVS*0.1**(NSIZE-1)
PT=Z*10.**JB
(IHATB.EQ.1)
PT=-PT
40
RETURN

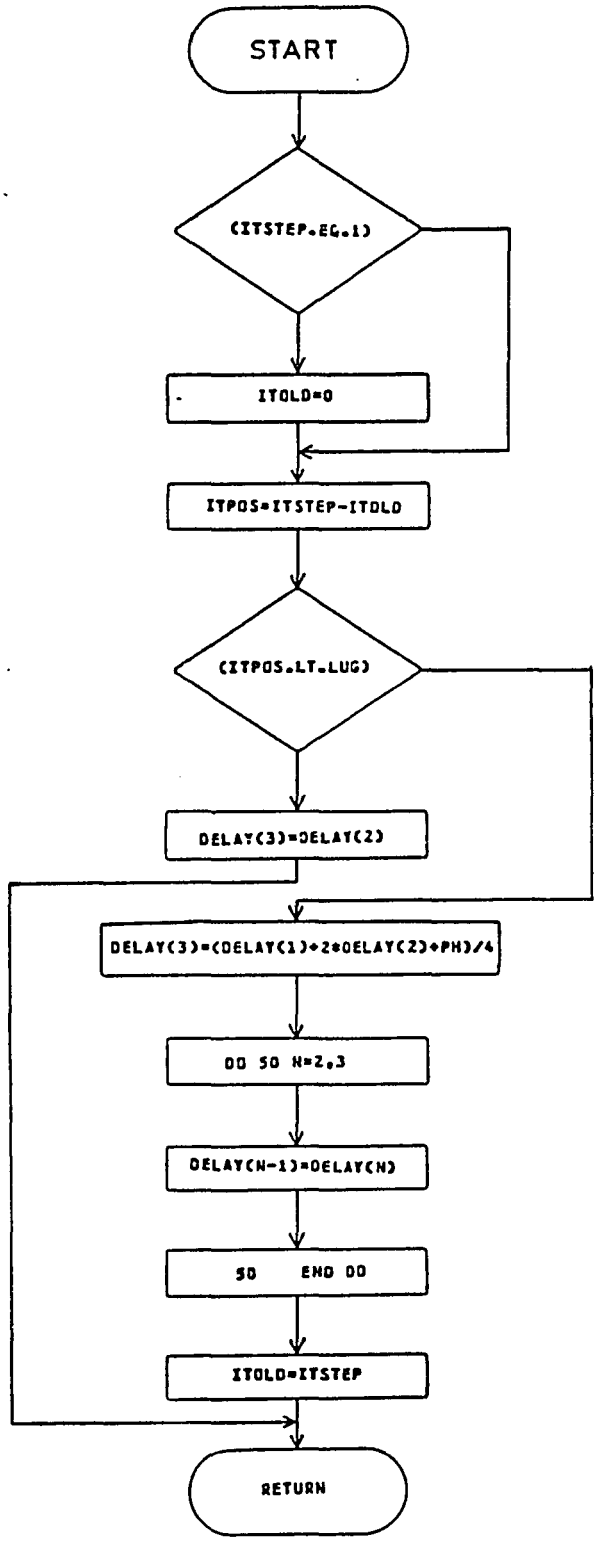

SUBROUTINE RICONO(BETA,FINI,FZ,PH)
START
PH=PT((1+BETA*FINI)/(1+BETA)/FZ)
RETURN

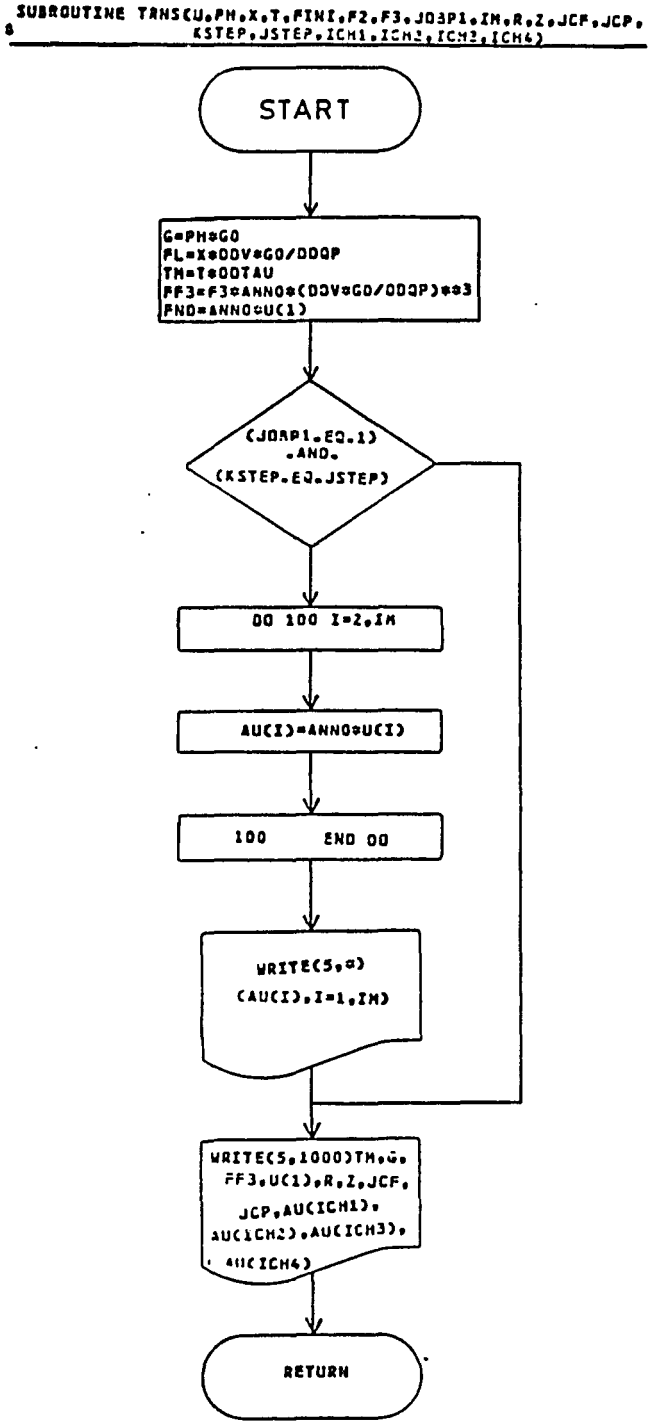
SUBROUTINE ODCOND(FINI,F2,JFLAG1,JFLAG2,XC,TH)



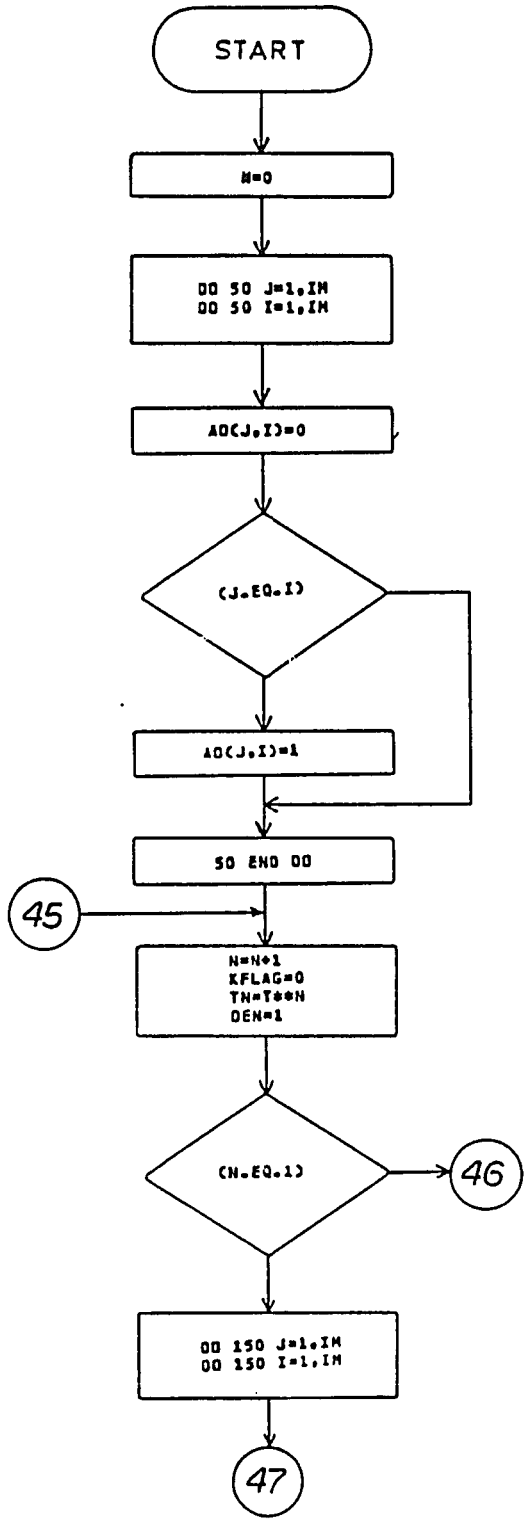

44
XCH=-SL/(2*AA)
43
(JFLAG2.EQ.2)
ZCH=XC
42
RETURN

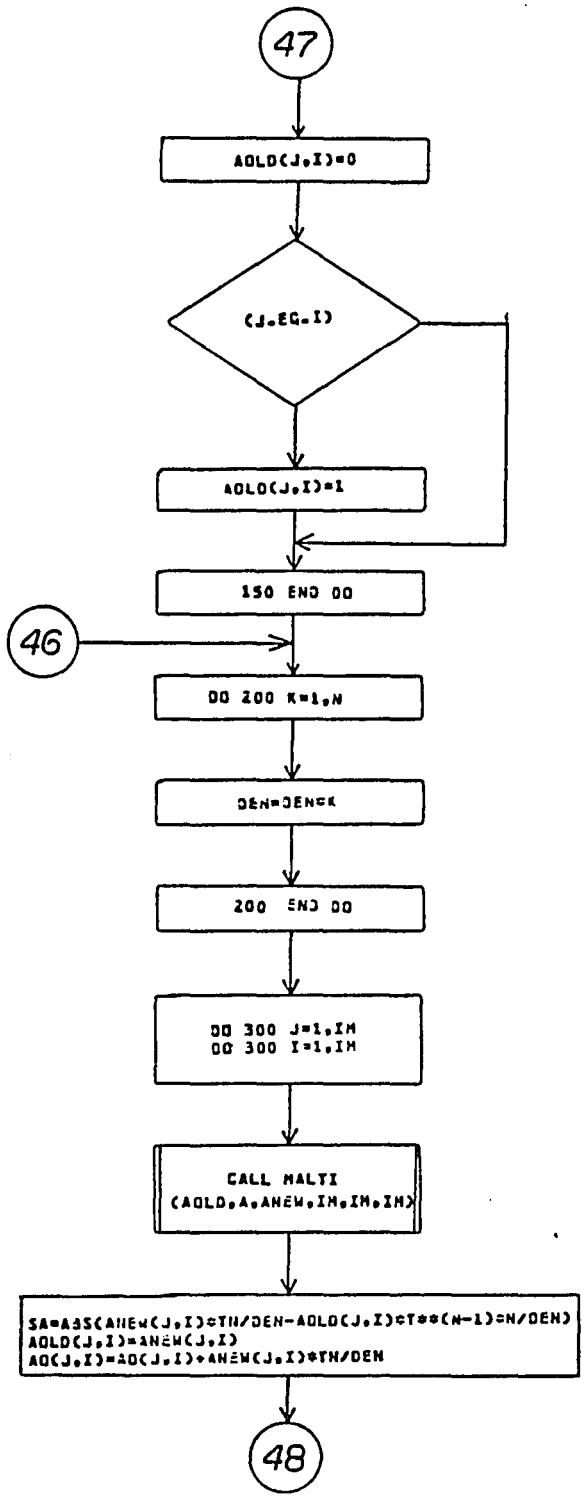
SUBROUTINE STUCK(PH,LUG,DELAY,ITSTEP)

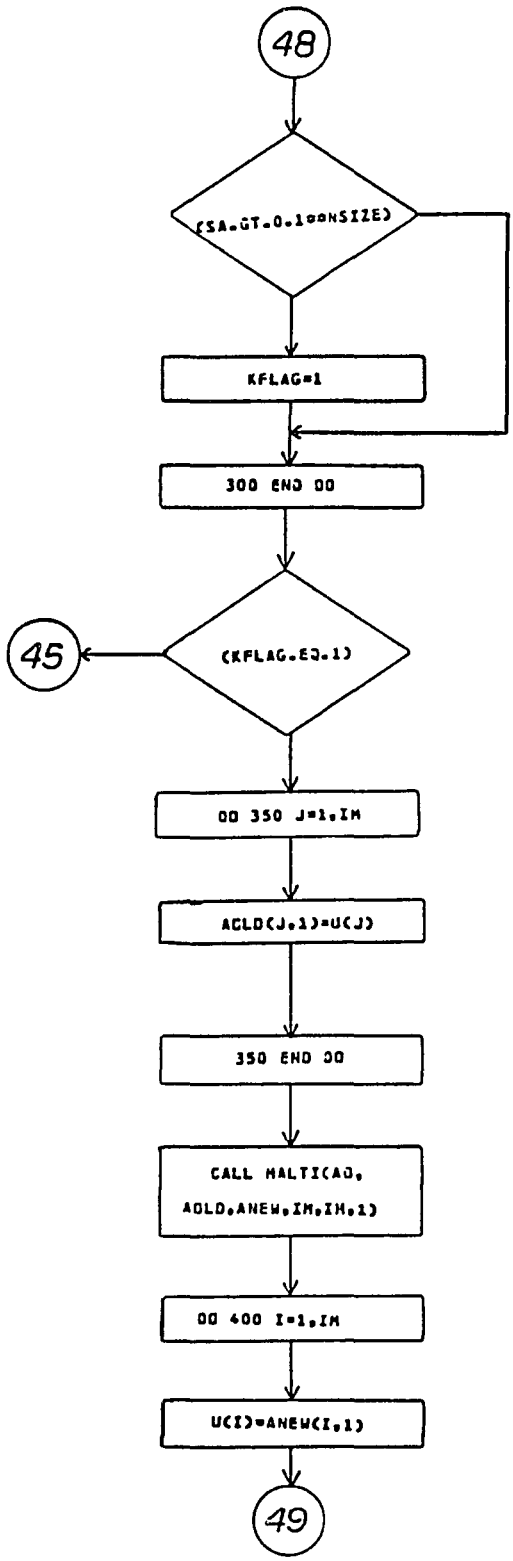


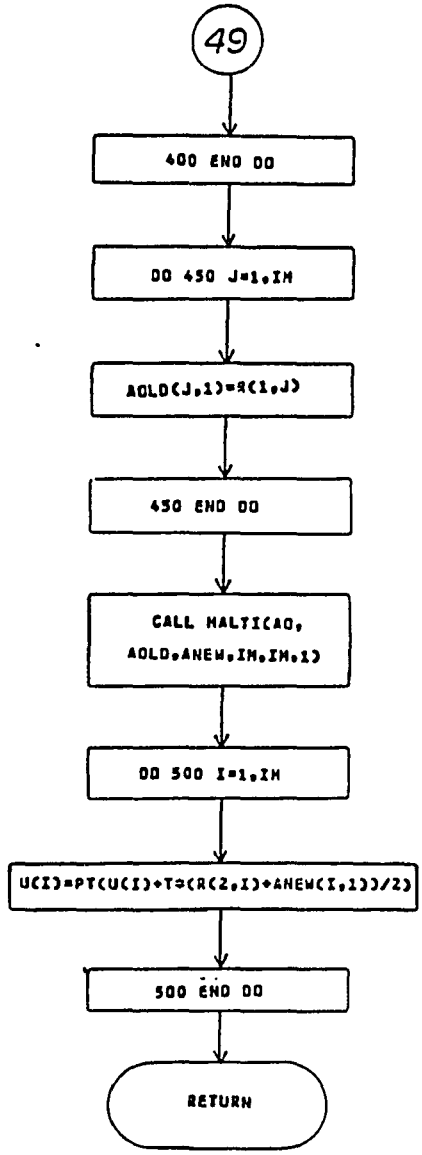

SUBROUTINE TANS(U,PH,X,T,FINI,F2,F3,JOBPI,IM,R,Z,JCF,JCP,
& KSTEP,JSTEP,ICH1,ICH2,ICH3,ICH4)
START
G=PH*GO
FL=X*ODV*GO/ODOP
TM=T*OOTAU
FF3=F3*ANNO*(ODV*GO/ODOP)**3
FND=ANNO*U(1)
(JOBPI.EQ.1)
.AND.
(KSTEP.EQ.JSTEP)
DO 100 I=2,IM
AU(I)=ANNO*U(I)
100 END DO
WRITE(5,*)
(AU(I),I=1,IM)
WRITE(5,1000)TM,G,
FF3,U(1),R,Z,JCF,
JCP,AU(ICH1),
AU(ICH2),AU(ICH3),
AU(ICH4)
RETURN

SUBROUTINE KAI(T,IM,U,R)

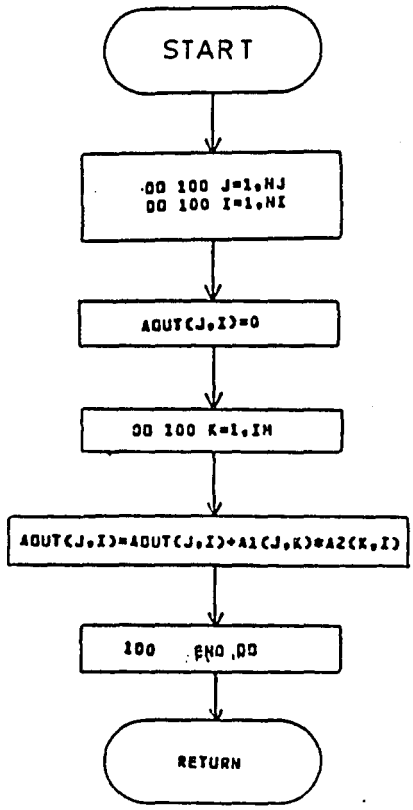



47
AOLD(J,I)=0
(J.EQ.I)
AOLD(J,I)=1
150 END DO
46
DO 200 K=1,N
DEN=DEN+K
200 END DO
DO 300 J=1,IM
DO 300 I=1,IM
CALL MALTI
(AOLD,A,ANEW,IM,IM,IM)
SA=ABS(ANEW(J,I)*TN/DEN-AOLD(J,I)*T**(N-1)*N/DEN)
AOLD(J,I)=ANEW(J,I)
AO(J,I)=AO(J,I)+ANEW(J,I)*TN/DEN
48

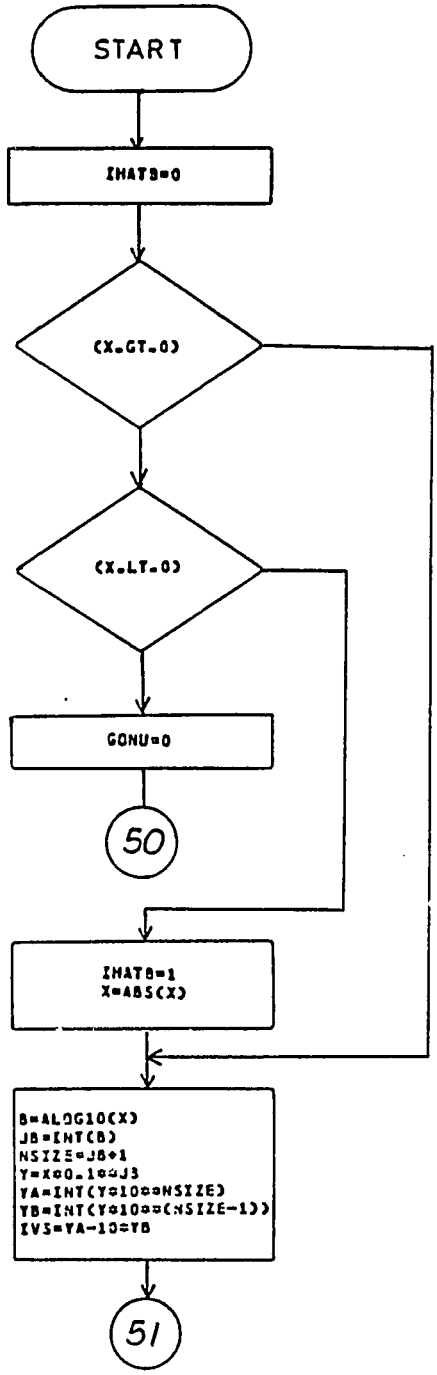

48
(SA.GT.0.1**NSIZE)
KFLAG=1
300 END DO
45
(KFLAG.EQ.1)
DO 350 J=1,IM
AOLD(J,1)=U(J)
350 END DO
CALL MALTI(AO,
AOLD,ANEW,IM,IM,1)
DO 400 I=1,IM
U(I)=ANEW(I,1)
49

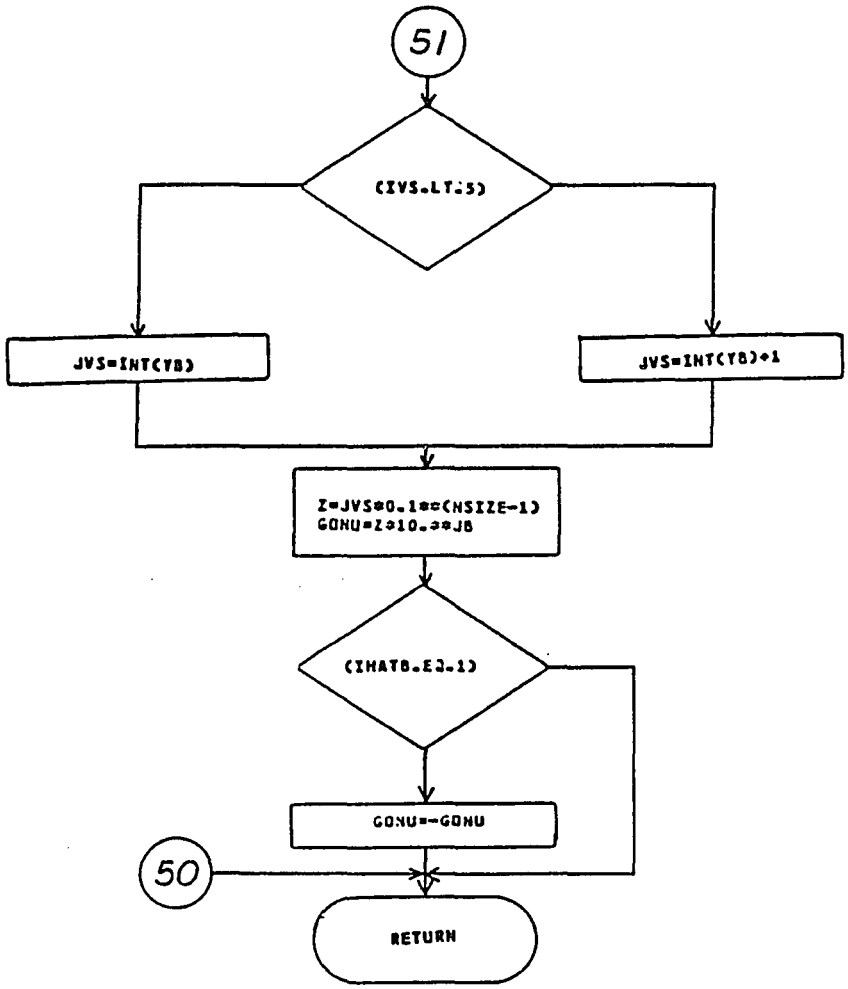

49
400 END DO
DO 450 J=1,IM
AOLD(J,1)=R(1,J)
450 END DO
CALL MALTI(AO,
AOLD,ANEW,IM,IM,1)
DO 500 I=1,IM
U(I)=PT(U(I)+T*(R(2,I)+ANEW(I,1))/2)
500 END DO
RETURN

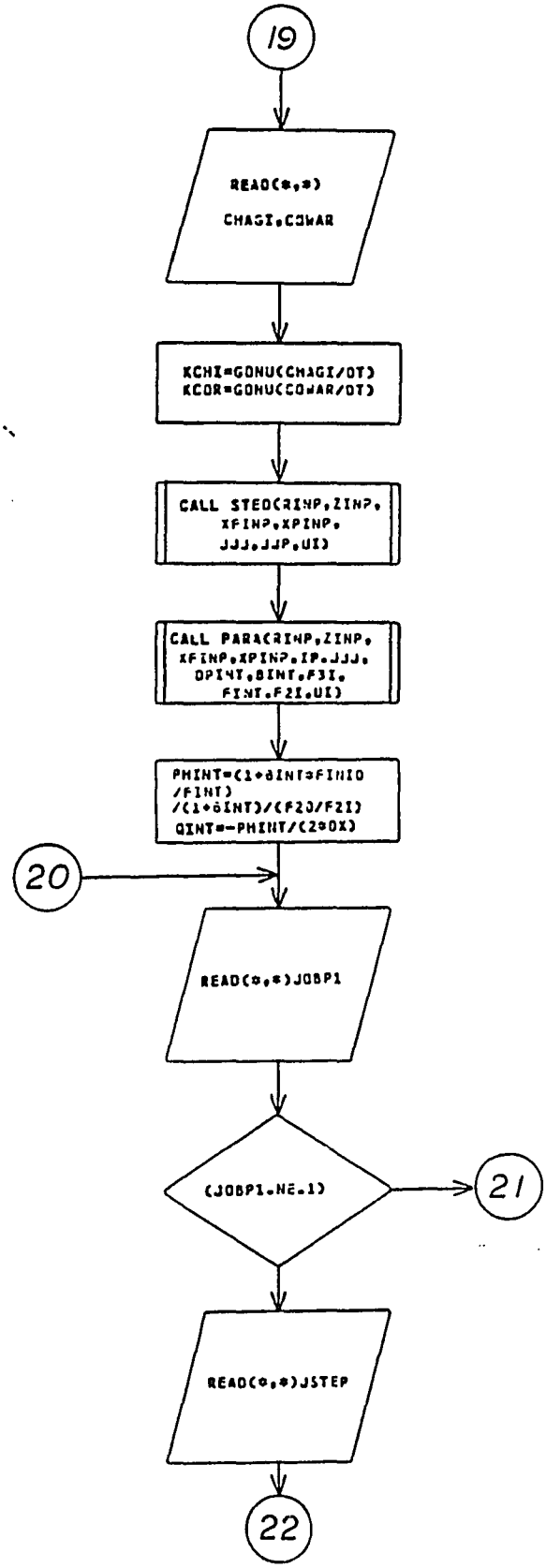
SUBROUTINE MALTI(A1,A2,AOUT,IM,NJ,NI)



FUNCTION GONU(X)




51
(IVS.LT.5)
JVS=INT(YB)
JVS=INT(YB)+1
Z=JVS*0.1**(NSIZE-1)
GONU=Z*10.**JB
(IHATB.EQ.1)
GONU=-GONU
50
RETURN


19
READ(*,*)
CHAGI,COWAR
KCHI=GONU(CHAGI/OT)
KCOR=GONU(COWAR/OT)
CALL STED(RINP,ZINP,
XFINP,XPINP,
JJJ,JJP,UI)
CALL PARA(RINP,ZINP,
XFINP,XPINP,IP,JJJ,
DPINT,BINT,F3I,
FINT,F2I,UI)
PHINT=(1+BINT*FINIO
/FINT)
/(1+BINT)/(F2J/F2I)
QINT=-PHINT/(2*DX)
20
READ(*,*)JOBP1
(JOBP1.NE.1)
21
READ(*,*)JSTEP
22

```
*       THE COMPUTER SIMULATION OF R-Z CRYSTALLIZER
*       WRITTEN BY SYUJI TSURUOKA IN OCT.1985
*
*       THE ALGORYTHM WHICH IS USED IN THIS PROGRAM
*       IS REFERED TO THE METHODS OF LINES.
      COMMON/BLOCK1/A(77,77),ITERX,DX,IXINT
      COMMON/BLOCK2/JOBP6
      COMMON/BLOCK3/DDQP,DDQF,DDKV,DDKA,DDRO,RVM31,RVM2,DDTAU,DDKG,DDCS
      COMMON/BLOCK4/ DDEC,DDF,DDCF,DDKN,DDCO,DDV,GO,ANNO,PMT
      COMMON/BLOCK5/DAKA,DAG,DAKG,DAPR,XCM,XCO
      DIMENSION U(77),UI(77),DELAY(77),UINT(77),REXT(2,77)
      DATA RINT,ZINT,XFINT,XPINT/0.,0.,0.,0./
      DATA TLUG,T/0.0,0.0/
      DATA ICH1,ICH2,ICH3,ICH4,JOBP9/1,2,3,4,2/
      DATA DDKV,DDKA,DDKG,DDKN,DDV,DDVL,DDQF,DDQP,DDQCL,DDRO,DDCS,
     $ DDCF,DAKG,DAKA,DAG,DAPR,GAMMA,DDEL,XCF/19*0.0/
      OPEN(UNIT=5,FILE='YOKO.DAT',STATUS='NEW')
      WRITE(5,1500)
*
*     INITIAL CONDITIONS ARE DEFINED IN DIMENSIONLESS VARIABLES.
*     OTHERWISE, ONE CAN CHOOSE DIMENSIONAL CONDITIONS, BUT THOSE
*     VALUES ARE TRANRATED INTO DIMENSIONLESS FORM.
*     THE INFINITE CONDITION CORRESPONDING TO THE ABOVE ENTRY IS
*     IS EXPECTED  AND CALUCULATED AS A UNIQUE STEADY STATE CONDITION.
*     THE ENTRY IS REGARDED AS EXTERNAL PERTURBATION AS LOG AS IT IS NOT
*     IDENTICAL TO THE INFINITE CONDITION.
*     CONCENTRATION IN CRYSTALLIZER IS FREELY CHOSEN IN THE ALGORITHM.
*
*     THIS PROGRAM WAS PROVIDED FOR THE BENCH SCALE CRYSTALLIZER
*     LOCATED AT RM. 26
*
*       THE NEXT LINES CREAT IDENTITY MATRICES.
*       THE NUMBER OF II CORESPONDS TO ARRAY SIZE
*       UINT = DISTURBAMNCE OF U
*       REXT(1,*) = THE LEFT VALUE OF THE EXTERNAL INPUT
*       REXT(2,*) = THE RIGHT VALUE OF THE EXTERNAL INPUT
*
      DO 1 II=1,77
      UINT(II)=0
      REXT(1,II)=0
      REXT(2,II)=0
    1 END DO
*       CHOOSE GROWTH RATE CONSTRAINT
   50 WRITE(*,*)' JOBP5=1 => DIMENSIONLESS '
      WRITE(*,*)' JOBP5=2 => DIMENSIONAL'
      WRITE(*,*)' JOBP5=3 => INITIALLY STEADY STATE EXPECTED AND'
      WRITE(*,*)'              DIMENSIONLESS FORM ON R-Z TYPE'
      READ(*,*)JOBP5
      IF(JOBP5.NE.3) THEN
         WRITE(*,*)' WHICH ASSUMPTION WILL YOU USE?'
         WRITE(*,*)' INITIAL CONDITIONS ARE:'
         WRITE(*,*)' A) BASED ON INFINITE CONDITIONS =>1'
         WRITE(*,*)' B) BASED ON T=0 CONDITIONS      =>2'
         READ(*,*)JOBP9
      ENDIF
*
*
*       DDC     = SOLUTE CONCENTRATION
*       DDCF    = FEED LIQUOR SOLUTE CONCENTRATION
*       DDCO    = INITIAL SOLUTE CONCENTRATION
```

```
*       DDCS    = SATURATED SOLUTE CONCENTRATION
*       DDF     = QF/QP
*       GO      = GROWTH RATE AT INFINITE TIME
*       DDKA    = AREA SHAPE FACTOR
*       DDKG    = GROWTH RATE KINETIC CONSTANT
*       DDKN    = NUCLEATION RATE KINETIC CONSTANT
*       DDKV    = VOLUMETRIC SHAPE FACTOR
*       DDV     = SLURRY VOLUME
*       DDVL    = CLEAR LIQUOR VOLUME
*       DDQP    = SLURRY BASED LIQUOR PRODUCT RATE
*       DDQF    = SLURRY BASED FINES DESTRUCTION RATE
*       DDQCL   = SLURRY FLOW RATE OF CLEAR LIQUOR ADVANCE
*       DDRO    = CRYSTAL DENSITY
*       GAMMA   = (DDQP+DDQCL)/DDQF
*       DDEC    = VOID RATE
*       DDTAU   = RESIDENCE TIME
*       DAKA    = DDKA
*       DAKG    = KG' =DDCS**DDKG/GO
*       DAG     = (GO*DDTAU)**4*N00
*       DAPR    = DDRO/DDCS
*       XC      = DIMENSIONLESS SOLUTE CONCENTRATION
*       XCF     = DIMENSIONLESS FEED SOLUTE CONCENTARTION
*       XCO     = DIMENSIONLESS INITIAL SOLUTE CONCENTRATION
*       XCM     = DIMENSIONLESS SOLUTE CONCENTRATION AT INFINITE
*       IF JOBPS=2, THEN READ NECESSARILY DIMENSIONAL VALUES
      IF(JOBPS.EQ.2) THEN
        WRITE(*,*)' ENTER DIMENSIONAL VALUES '
        WRITE(*,*)' KV,KA,KG,KN'
        READ(*,*)DDKV,DDKA,DDKG,DDKN
        WRITE(*,*)' SLURRY AND CLEAR LIQUOR VOLUME IN X-ER'
        READ(*,*)DDV,DDVL
        WRITE(*,*)' QF, QP, QCLA'
        READ(*,*)DDQF,DDQP,DDQCL
        DDTAU=DDV/DDQP
        GAMMA=(DDQP+DDQCL)/DDQF
        DDEC=DDVL/DDV
        WRITE(*,*)' CRYSTAL DENSITY, AND SATURATED SOLUTE CONCENTRATION'
        READ(*,*)DDRO,DDCS
        WRITE(*,*)' SOLUTE CONCENTERATION'
        READ(*,*)DDC
        WRITE(*,*)' FEED LIQUOR SOLUTE CONCENTRATION'
        READ(*,*)DDCF
        DDF=DDQF/DDQP
        PMT=(DDV-DDVL)*DDRO/DDV
        DAKA=DDKA
        DARP=DDRO/DDCS
        XCO=(DDC-DDCS)/DDCS
      ENDIF
      IF(JOBPS.EQ.1) THEN
        WRITE(*,*)' ENTER  DIMENSIONLESS VALUS'
        WRITE(*,*)' KG-PRIME = CS*KG/GO, WHERE GO IS REFERENCE G'
        READ(*,*)DAKG
        WRITE(*,*)' SHAPE FACTOR FOR AREA, KA'
        READ(*,*)DAKA
        WRITE(*,*)' G =(GO*TAU)**4*NOO, WHERE NOO IS NUCLEI DENSITY'
        WRITE(*,*)' AS REFERENCE'
        READ(*,*)DAG
        WRITE(*,*)' RATIO OF DENSITY/SATURATED CONCENTRATION'
        READ(*,*)DAPR
        WRITE(*,*)' GAMMA'
```

```
        READ(*,*)GAMMA
        WRITE(*,*)' F'
        READ(*,*)DDF
        WRITE(*,*)' SHAPE FACTOR FOR MASS, KV'
        READ(*,*)DDKV
        WRITE(*,*)' RATIO OF VL/V '
        READ(*,*)DDEC
        WRITE(*,*)' FEED LIQUOR SOLUTE CONCENTRSTION XCF'
        READ(*,*)XCF
        WRITE(*,*)' INITIAL SOLUTE CONCENTRATION XC AS REFERENCE'
        READ(*,*)XCO
      ENDIF
*       READ INITIAL CONDITIONS AND INITIAL DATA
*
      WRITE(*,1000)
*             XF= FINE REMOVEL SIZE
*             XP= PRODUCT CUT SIZE
*             R=  RECICLE RATIO OF DISOLVER
*             Z=  RECICLE RATIO OF PRODUCTS CLASSIFIER
*             BRZ=GRADIENT BETWEEN R AND Z
      READ(*,*)XF,XP,R,Z
*       READ XRANGE = PARTICLE SIZE RANGE
*            TRANGE = TIME LENGTH OBSERVED
*            ITERX  = A NUMBER OF DISCRETIZED POINTS OF PARTICLE SIZE
*            ITERT  = A NUMBER OF DISCRETIZED POINTS OF TIME LENGTH
      IF(JOBP5.EQ.2) THEN
         WRITE(*,*)' ENTER CALUCULATION RANGE OF L AND T'
         READ(*,*)XLRANGE,TLRANGE
         XRANGE=XLRAGE/DDV/DDTAU
         TRANGE=TLRANGE/DDTAU
         GOTO 55
      ENDIF
      WRITE(*,*)' CHOOSE A NONDIMENSIONAL INITIAL CONDITION'
      WRITE(*,*)' STANDARD        => 1'
      WRITE(*,*)' MODIFIED R-Z    => 2'
*       MODIFIED R-Z MEANS THAT SLOPE BETWEEN R & Z IS BRZ INSTED 1
      WRITE(*,*)' NON-PARTICLE    => 3'
      WRITE(*,*)' ELSE            => 4'
      READ(*,*)JOBP10
      IF(JOBP10.EQ.2) THEN
         WRITE(*,*)' ENTER GRADIENT BETWEEN R & Z'
         READ(*,*)BRZ
      ELSE
         BRZ=1
      ENDIF
      WRITE(*,1010)
      READ(*,*)XRANGE,TRANGE
   55 WRITE(*,1020)
      READ(*,*)ITERX,ITERT
*       READ    FI = NUCLEATION SESITIVITY
*               FJ = A NUMBER OF CONSTANT
      WRITE(*,1040)
      READ(*,*)FJ,FI
      FII=FI-1
*       GROTH RATE TIME-LUG SET
      WRITE(*,*)' ENTER PROPORTIONAL CONSTANT KC  '
      READ(*,*)TLUG
      WRITE(*,*)' ENTER SAMPLE HOLD TIME OF CONTROLLER'
      READ(*,*)TRLUG
      IF(JOBP5.EQ.2) THEN
```

```
         TRLUG=TRLUG/DDTAU
       ENDIF
       WRITE(*,*)' ENTER VALID DIGID NUMBER OF CALCULATION'
       READ(*,*)JOBP6
*        SYSTEM CONSTANTS
*        DX     = SIZE INCREMENT
*        DT     = TIME INCREMENT
*        IM     = MATRIX SIZE WITH A DUMMY POINT
*        IP     = REQUIRED MATRIX SIZE
       DX=PT(XRANGE/ITERX)
       DT=PT(TRANGE/ITERT)
       IM=ITERX+2
       IP=ITERX+1
       LUG=GONU(TRLUG/DT)
*
*
       IF((JOBP10.EQ.1).OR.(JOBP10.EQ.2)) THEN
          CALL STED(R,Z,XF,XP,JCFS,JCPS,U)
          CALL PARA(R,Z,XF,XP,IP,JCFS,DP,BETA,F30,FINI0,F20,U)
       ELSE IF(JOBP10.EQ.3) THEN
*        IN THIS CASE, STEADY STATE CSD FOR INITIAL CONDITIONS
*        ARE TAKEN AS A BASE OF CALCULATION
          CALL STED(R,Z,XF,XP,JCFS,JCPS,U)
          CALL PARA(R,Z,XF,XP,IP,JCFS,DP,BETA,F30,FINI0,F20,U)
          CALL STEDZR(XF,XP,JCFS,JCPS,U)
       ELSE
*        WRITE A FUNCTION AS YOU WANT.   SO FAR NO FUNCTION IS INSTALLED.
          CALL STEDEL(XF,XP,JCFS,JCPS,U)
       ENDIF
       IF(FINI0.EQ.0)FINI0=1
       IF(JOBP5.EQ.3) THEN
         PH=1
         GOTO 60
       ENDIF
       CALL DDCOND(FINI0,F20,JFLAG1,JOBP9,XC,0)
       IF(JFLAG1.EQ.1)GOTO 50
       PH0=PT(DAKG*XCM)
       PHEXT=PT(DAKG*XC/PH0)
       PH=1
   60  QUE=-PH/(12*DX)
       IF(JOBP5.EQ.3)GOTO 65
       IF(JOBP5.EQ.2)THEN
          G0=PT(DDCS*DDKG/DAKG)
          ANN0=DDKN*DDEC*G0**FII*PMT**FJ
       ENDIF
       WRITE(*,2030)PH0,PHEXT,PH
       WRITE(*,*)' IF NOT SUTICEFIED, ENTER 1 ELSE 0'
       READ(*,*)JOBP8
       IF(JOBP8.EQ.1)GOTO 50
       WRITE(5,2030)PH0,PHEXT,PH
       WRITE(5,2060)
   65  WRITE(5,2000)XF,XRANGE,ITERX,DX,XP,TRANGE,ITERT,DT,R,
      $ Z,FI,FJ
       WRITE(5,2070)DDKV,DDKA,DDKG,DDKN,DDV,DDVL,DDQF,DDQP,DDQCL,
      $ DDR0,DDCS,DDC0,DDCF,DAKG,DAKA,DAG,DAPR,DCEC,XCF,XC0,GAMMA,
      $ DDF
       WRITE(5,1090)TLUG,TRLUG
*        SET CONTROL FUTURES
*        JOBP1 = FLAG OF OUTPUT CONTROL OF ALL OF U(I)
*        JOBP2 = FLAG OF PERTURBATION ARGUMENT
```

```
*       JOBP3 = FLAG OF PI-CONTROL
*       JOBP4 = FLAG OF CONTROL FACTOR
*       JOBP5 = FLAG OF GROWTH RATE CONSTRAINT
*       JOBP6 = THE DIGIT SIZE
*       JOBP7 = FLAG OF DIMENSIONAL VALUE OUTPUT
*       JOBP8 = FLAG OF GROTH RATE CHECK
*       JOBP9 = FLAG OF AN INITIAL CONDETION CHOSEN
*       JOBP10= FLAG OF A NONDIMENSIONAL INITIAL DISTRIBUTION
*       JOBP11= FLAG OF ZERO COMPENSATION OF U(I)
*       JOBP12= FLAG OF U(1) INPUT POSITION
*       JOBP13= FLAG OF U(ICH*);1=INTRINSIC,2=RATIO
*       JFLAG1= ERROR FLAG; 1=ERROR OCCURS
*       JFLAG2= TIME SEQUENCE CONTROL FLAG
*       CK*   = PROPORTIONAL CONSTANT
*       CKBD  = BOUND LIMIT
*       FIT*  = RESET RATE
*       IND** = SWITCH OF PI-CONTROLLER
      WRITE(*,*)' CONTROLLER ON =>1, ELSE =>0'
      READ(*,*)JOBP3
      IF(JOBP3.EQ.1) THEN
        WRITE(*,1045)
        WRITE(5,1045)
        READ(*,*)JOBP4
        WRITE(5,*)JOBP4
        WRITE(*,*)' ENTER PROPORTIONAL CONSTANTS'
        WRITE(*,*)' CKR,CKZ,CKXF,CKXP'
        READ(*,*)CKR,CKZ,CKXF,CKXP
        WRITE(*,*)' ENTER ERROR BOUND RANGE ON CONTROLLER'
        READ(*,*)CKBD
        FITR=0
        FITZ=0
        FITXF=0
        FITXP=0
        WRITE(*,*)' PI CONTROL =>1, ELSE =>0 ON R,Z,XF,XP'
        READ(*,*)INDPIR,INDPIZ,INDXF,INDXP
        WRITE(*,*)'ENTER RESET RATE'
        IF(INDPIR.EQ.1) THEN
                WRITE(*,*)' ON R'
                READ(*,*)FITR
        ENDIF
        IF(INDPIZ.EQ.1) THEN
                WRITE(*,*)' ON Z'
                READ(*,*)FITZ
        ENDIF
        IF(INDXF.EQ.1) THEN
                WRITE(*,*)' ON XF'
                READ(*,*)FITXF
        ENDIF
        IF(INDXP.EQ.1) THEN
                WRITE(*,*)' ON XP'
                READ(*,*)FITXP
        ENDIF
        WRITE(*,*)' END OF INPUT OF RESET RATE'
        WRITE(5,2040)CKR,CKZ,CKXF,CKXP,FITR,FITZ,FITXF,FITXP
      ENDIF
      WRITE(*,1050)
      WRITE(5,1050)
      READ(*,*)JOBP2
      WRITE(5,*)' CHOSEN => ',JOBP2
      WRITE(5,2120)
```

```
      IF(JOBP2.NE.8)WRITE(*,1060)
      RINP=R
      ZINP=Z
      XFINP=XF
      XPINP=XP
      IF(JOBP2.EQ.1) THEN
        WRITE(*,*)' ENTER THE SIZE RANGE.  0 MEANS U(1)'
        IF(JOBP5.NE.2)READ(*,*)USIZE
        IF(JOBP5.EQ.2)READ(*,*)AUSIZE
        IF(JOBP5.EQ.2)USIZE=AUSIZE/DDV/DDTAU
        JUSIZE=USIZE/DX+1
        WRITE(*,*)' ENTER DISTRIBUTED AMMONUT'
        DO 70 JU=1,JUSIZE
        WRITE(*,*)'  U(',JU,')'
        READ(*,*)UINT(JU)
 70     END DO
        WRITE(*,*)' U(1) IS INTERNAL, OR EXTERNAL?'
        WRITE(*,*)' INTERNAL =>0, EXTERNAL =>1'
        READ(*,*)JOBP12
       ELSEIF(JOBP2.EQ.2) THEN
        READ(*,*)RINT
        RINP=RINT
      ELSEIF(JOBP2.EQ.3) THEN
        READ(*,*)ZINT
        ZINP=ZINT
      ELSEIF(JOBP2.EQ.4) THEN
        READ(*,*)XFINT
        XFINP=XFINT
      ELSEIF(JOBP2.EQ.5) THEN
        READ(*,*)XPINT
        XPINP=XPINT
      ELSEIF(JOBP2.EQ.6) THEN
        READ(*,*)RINT,ZINT
        RINP=RINT
        ZINP=ZINT
      ELSEIF(JOBP2.EQ.7) THEN
        READ(*,*)XFINT,XPINT
        XFINP=XFINT
        XPINP=XPINT
      ELSE
        GOTO 100
      ENDIF
      WRITE(*,1080)
      WRITE(5,1080)
*       SET CHAGI = THE BEGINNING OF PERTURBATION
*            COWAR = THE END OF PERTURBATION
      READ(*,*)CHAGI,COWAR
      WRITE(5,*)CHAGI,COWAR
      KCHI=GONU(CHAGI/DT)
      KCOR=GONU(COWAR/DT)
      WRITE(5,*)KCHI,KCOR
*       THE FOLLOWING IS PERTURBATION CONDITION CALUCULATION
      CALL STED(RINP,ZINP,XFINP,XPINP,JJJ,JJP,UI)
      CALL PARA(RINP,ZINP,XFINP,XPINP,IP,JJJ,DPINT,BINT,F3I,FINT,F2I,UI)
      PHINT=(1+BINT*FINID/FINT)/(1+BINT)/(F2D/F2I)
      QINT=-PHINT/(2*DX)
      DO 95 I=1,JUSIZE
      WRITE(5,1110)I,UINT(I)
 95   END DO
      IF(JOBP12.EQ.0)WRITE(5,*)' U(1) IS INTERNAL DISTERBANCE'
```

```
      IF(JOBP12.EQ.1)WRITE(5,*)' U(1) IS EXTERNAL DISTERBANCE'
      WRITE(5,1070)RINP,ZINP,XFINP,XPINP
  100 WRITE(*,2030)
      READ(*,*)JOBP1
      IF(JOBP1.NE.1)GOTO 150
      READ(*,*)JSTEP
  150 WRITE(*,*)' CHOOSE 4 OUTPUT POINTS OF U(I)'
      READ(*,*)XCH1,XCH2,XCH3,XCH4
      ICH1=GONU(XCH1/DX)
      ICH2=GONU(XCH2/DX)
      ICH3=GONU(XCH3/DX)
      ICH4=GONU(XCH4/DX)
      IF(ICH1.GT.ITERX)THEN
         WRITE(*,*)' ENTERED POINTS ARE BEYOND CALCULATION RANGE'
         GOTO 150
      ELSEIF(ICH2.GT.ITERX)THEN
         WRITE(*,*)' ENTERED POINTS ARE BEYOND CALCULATION RANGE'
         GOTO 150
      ELSEIF(ICH3.GT.ITERX)THEN
         WRITE(*,*)' ENTERED POINTS ARE BEYOND CALCULATION RANGE'
         GOTO 150
      ELSEIF(ICH4.GT.ITERX)THEN
         WRITE(*,*)' ENTERED POINTS ARE BEYCND CALCULATION RANGE'
         GOTO 150
      ENDIF
      XCH1=ICH1*DX
      XCH2=ICH2*DX
      XCH3=ICH3*DX
      XCH4=ICH4*DX
      WRITE(*,*)' CHOSEN POINTS BY COMPUTER ARE'
      WRITE(*,*)XCH1,XCH2,XCH3,XCH4
      WRITE(*,*)' IF NOT SATICFIED, ENTER 2, ELSE 0'
      READ(*,*)JOBP8
      IF(JOBP8.EQ.2)GOTO 150
      WRITE(*,*)' PRINT OUT IS INTRINSIC =>1, OR RATIO =>2'
      READ(*,*)JOBP13
      WRITE(*,*)' ZERO COMPENSATOR OF U(I)'
      WRITE(*,*)'    ON => 1 ,  OFF => 0'
      READ(*,*)JOBP11
      IF(JOBP11.EQ.0)WRITE(5,2150)
      IF(JOBP11.EQ.1)WRITE(5,2160)
      UB1=U(ICH1)
      UB2=U(ICH2)
      UB3=U(ICH3)
      UB4=U(ICH4)
      WRITE(*,*)' DIMENSIONAL VALUES ARE TYPED OUT IN 1'
      READ(*,*)JOBP7
      IF(JOBP3.EQ.1) THEN
        UINT0=U(1)
        DO 200 J=1,IM
        UI(J)=U(J)
  200 CONTINUE
      ENDIF
      IF(JOBP1.EQ.1)WRITE(5,*)(U(I),I=1,IP)
      DO 330 J=1,IM
      DO 330 I=1,IM
      A(J,I)=0
  330 END DO
      DO 350 J=3,IM-2
      A(J,J-2)=QUE
```

```
      A(J,J-1)=-8*QUE
      A(J,J)=(12*DX*FH(J,JCFS,JCPS,R,Z,BRZ)/PH)*QUE
      A(J,J+1)=8*QUE
      A(J,J+2)=-QUE
  350 END DO
      A(1,1)=(-22+12*DX*FH(1,JCFS,JCPS,R,Z,BRZ)/PH)*QUE
      A(1,2)=36*QUE
      A(1,3)=-18*QUE
      A(1,4)=4*QUE
      A(2,1)=-4*QUE
      A(2,2)=(-6+12*DX*FH(2,JCFS,JCPS,R,Z,BRZ)/PH)*QUE
      A(2,3)=12*QUE
      A(2,4)=-2*QUE
      A(IM-1,IM-3)=2*QUE
      A(IM-1,IM-2)=-12*QUE
      A(IM-1,IM-1)=(6+12*DX*FH(IM-1,JCFS,JCPS,R,Z,BRZ)/PH)*QUE
      A(IM-1,IM)=4*QUE
      A(IM,IM-3)=-4*QUE
      A(IM,IM-2)=18*QUE
      A(IM,IM-1)=-36*QUE
      A(IM,IM)=(22+12*DX*FH(IM,JCFS,JCPS,R,Z,BRZ)/PH)*QUE
      ITSYS=0
      F3=1
      F2=1
      FINI=1
      ESIG=0
      ERROR1=0
      KSTEP=0
      F3DEL=0
      IF(JOBP2.NE.9) THEN
         DO 355 I=1,JUSIZE
         REXT(1,I)=0
         REXT(2,I)=0
  355    END DO
      ENDIF
      IF(JOBP5.EQ.3)GOTO 360
      PH=PHEXT
      WRITE(*,*)F3,PH,FJ,FII
      U(1)=F3**FJ*PH**FII
  360 DO 370 I=1,3
      DELAY(I)=1
  370 END DO
      IF(JOBP5.EQ.2) THEN
         WRITE(5,2090)
         IF(JOBP9.EQ.1)WRITE(5,2130)
         IF(JOBP9.EQ.2)WRITE(5,2140)
      ELSE
         WRITE(5,2100)
         IF(JOBP9.EQ.1)WRITE(5,2130)
         IF(JOBP9.EQ.2)WRITE(5,2140)
      ENDIF
      WRITE(5,2050)ICH1,ICH2,ICH3,ICH4,XCH1,XCH2,XCH3,XCH4
      T=0
      WRITE(5,2010)T,PH,F3DEL,U(1),R,Z,JCFS,JCPS,U(ICH1),
     1 U(ICH2),U(ICH3),U(ICH4)
C       START SYSTEM TIME ITERATION
      WRITE(*,1510)
      WRITE(*,*)'            START EXECUTION'
  400 ITSYS=ITSYS+1
      KSTEP=KSTEP+1
```

```
      T=PT(ITSYS*TRANGE/ITERT)
      QUECON=-PH/(12*DX)
*       SET PERTURBATIN
      IF(JOBP2.EQ.8) GOTO 410
      IF(JOBP2.EQ.1) THEN
        IF((ITSYS.GE.KCHI).AND.(ITSYS.LE.KCOR)) THEN
                DO 405 I=1,JUSIZE
                REXT(1,I)=UINT(I)
                REXT(2,I)=UINT(I)
  405           END DO
                IF(JOBP12.EQ.0) THEN
                   U(1)=U(1)+UINT(1)
                   REXT(1,1)=0
                   REXT(2,1)=0
                ENDIF
        ELSEIF(ITSYS.EQ.(KCOR+1)) THEN
                DO 406 I=1,JUSIZE
                REXT(1,I)=UINT(I)
                REXT(2,I)=0
  406           END DO
                IF(JOBP12.EQ.0) THEN
                   REXT(1,1)=0
                   REXT(2,1)=0
                ENDIF
        ELSE
                DO 407 I=1,JUSIZE
                REXT(1,I)=0
                REXT(2,I)=0
  407           END DO
        ENDIF
        GOTO 410
*
      ELSEIF((JOBP2.GE.2).AND.(JOBP2.LE.7)) THEN
        IF(ITSYS.EQ.1) THEN
                FINIS=FINI0
                F2S=F20
                G=G0
        ENDIF
        IF((ITSYS.GE.KCHI).AND.(ITSYS.LE.KCOR)) THEN
                PHTEM=(1+BINT*FINIS/FINT)/(1+BINT)/(F20/F2I)
                RCON=RINP
                ZCON=ZINP
                JCF=JJJ
                JCP=JJP
                U(1)=F3**FJ*PHTEM**FII+U(1)
                GOTO 420
        ELSE
                GOTO 410
        ENDIF
      ENDIF
  410 JCF=JCFS
      JCP=JCPS
*
*
      UR=U(1)+REXT(1,1)
      CALL STUCK(UR,LUG,DELAY,ITSTEP)
      RCON=TLUG*R*(DELAY(3)-1)+R
      ZCON=Z
*       CALUCULATE ERRORS AND CONTROLLER
  420 IF(JOBP3.EQ.1) THEN
```

```
      EU1=1-U(1)
      EF3=1-F3
      EPH=1-PH
      IF(JOBP4.EQ.1) ERROR=EF3
      IF(JOBP4.EQ.2) ERROR=EU1
      IF(JOBP4.EQ.3) ERROR=EPH
      ESIG=ESIG+(ERROR1+ERROR)*DT/2
      ERROR1=ERROR
      ERR1=ABS(ERROR)
      IF(ERR1.LE.CKBD) GOTO 430
      RCON=R*(1+CKR*(ERROR+INDPIR*FITR*ESIG))
      ZCON=Z*(1+CKZ*(ERROR+INDPIZ*FITZ*ESIG))
      JCF=JCFS*(1+CKXF*(ERROR+INDXF*FITXF*ESIG))
      JCP=JCPS*(1+CKXP*(ERROR+INDXP*FITXP*ESIG))
      IF(RCON.LT.1)RCON=1
      IF(ZCON.LT.1)ZCON=1
      IF(JCF.LT.0)JCF=0
      IF(JCP.LT.JCF)JCP=JCF
      IF(JCP.GE.IP)JCP=IP
      IF((JOBP2.EQ.2).OR.(JOBP2.EQ.6)) THEN
            IF((ITSYS.GE.KCHI).AND.(ITSYS.LE.KCOR))RCON=RINP
      ENDIF
      IF((JOBP2.EQ.3).OR.(JOBP2.EQ.6)) THEN
            IF((ITSYS.GE.KCHI).AND.(ITSYS.LE.KCOR))ZCON=ZINP
      ENDIF
      IF((JOBP2.EQ.4).OR.(JOBP2.EQ.7)) THEN
            IF((ITSYS.GE.KCHI).AND.(ITSYS.LE.KCOR))JCF=JJJ
      ENDIF
      IF((JOBP2.EQ.5).OR.(JOBP2.EQ.7)) THEN
            IF((ITSYS.GE.KCHI).AND.(ITSYS.LE.KCOR))JCP=JJP
      ENDIF
  430 ENDIF
      CALL IREKAE(PH,QUECON,JCF,JCP,RCON,ZCON,BRZ)
      CALL KAI(DT,IM,U,REXT)
*       ZERO COMPENSATER; IF U<0 AT JOBP10=3, UIE RESET TO ZERO
      IF((JOBP10.EQ.3).OR.(JOBP11.EQ.1)) THEN
         DO 440 I=1,IM
         IF(U(I).LT.0)U(I)=0
  440    END DO
      ENDIF
*       NEW BOUNDARY CONDITIONS EVALUATED
      SISU=3
      JOGEN=JCF
      CALL SEKI(SISU,IP,F3S,U)
      CALL SEKI(SISU,JOGEN,FINIS,U)
      SISU=2
      CALL SEKI(SISU,IP,F2S,U)
      IF((JOBP10.EQ.3).AND.(ITSYS.EQ.1)) THEN
         CALL PARA(R,Z,XF,XP,IP,JCFS,DP,BETA,F30,FINI0,F20,U)
      ENDIF
      F3=PT(F3S/F30)
      FINI=PT(FINIS/FINI0)
      F2=PT(F2S/F20)
      F3DEL=PT(F3-1)
      IF(JOBP5.EQ.3) THEN
         CALL RZCOND(BETA,FINI,F2,PH)
         GOTO 500
      ENDIF
      CALL DOCOND(FINIS,F2S,JFLAG1,0,XC,T)
      IF(JFLAG1.EQ.1) THEN
```

```
          WRITE(*,*)' IMAGINARY CONCENTRATION OCCURED. NO PATH TO GO'
          GOTO 999
       ENDIF
       PH=PT(DAKG*XC/PH0)
  500  IF((JOBP1.EQ.1).AND.(KSTEP.EQ.JSTEP))THEN
          WRITE(5,2110)
          WRITE(5,*)(U(I),I=1,IP)
          WRITE(5,2120)
          IF(JOBP5.NE.2)WRITE(5,2100)
          IF(JOBP5.EQ.2)WRITE(5,2090)
          KSTEP=0
       ENDIF
       IF((F3.LT.0).OR.(PH.LT.0)) THEN
          WRITE(5,*)' NEGATIVE U(I)'
          WRITE(5,*)'F3,PH',F3,PH
          DO 550 I=1,IP
          IF(U(I).LE.0)WRITE(5,*)I,U(I)
  550     END DO
          GOTO 999
       END IF
  570  U(1)=PT(F3**FJ*PH**FII)
       WRITE(*,*)T,U(1),F3DEL,PH
       IF(JOBP5.EQ.2) THEN
           CALL TRNS(U,PH,X,T,FINI,F2,F3,JOBP1,IM,RCON,ZCON,JCF,JCP,
      $                KSTEP,JSTEP,ICH1,ICH2,ICH3,ICH4)
           GOTO 600
       ENDIF
       IF(JOBP13.EQ.1) THEN
          UBW1=U(ICH1)
          UBW2=U(ICH2)
          UBW3=U(ICH3)
          UBW4=U(ICH4)
       ELSE
          UBW1=U(ICH1)/UB1
          UBW2=U(ICH2)/UB2
          UBW3=U(ICH3)/UB3
          UBW4=U(ICH4)/UB4
       ENDIF
       WRITE(5,2010)T,PH,PT(F3DEL),U(1),RCON,ZCON,JCF,JCP,
      $             UBW1,UBW2,UBW3,UBW4
  600  IF(ITSYS.LT.ITERT) GOTO 400
 1000  FORMAT(' ENTER XF,XP,R,Z')
 1010  FORMAT(' ENTER CALCULATION RANGE ON EACH AXIS'/
     1       ' X-RANGE, AND THEN T-RANGE')
 1020  FORMAT(' ENTER THE NUMBER OF ITERATION'/
     1       ' X-AXIS AND THEN T-AXIS')
 1040  FORMAT(' ENTER J AND I')
 1045  FORMAT(' CONTROL FACTOR: F3 =>1, U(1) =>2, PHAI =>3')
 1050  FORMAT(' CHOOSE AND ENTER THE CORRESPONDING NUMBER OF
     1 PERTURBATION  VARIABLE(S)'
     2 /' U(I) =>1'/' R    =>2'/' Z    =>3'/' XF   =>4'/' XP   =>5'/
     3' R&Z  =>6'/' XF&XP=>7'/' NONE =>8'/' REXT(I)<>0 =>9'/)
 1060  FORMAT(' ENTER THE VALUE OF THE PERTURBATION VARIABLE(S)')
 1070  FORMAT(//' ********** PERTURBATIN **********'/
     1 ' R=',E11.4,5X,'Z=',E11.4,5X,'XF=',E11.4,5X,'XP=',E11.4//)
 1080  FORMAT(' ENTER PERTURBATION STARTING POINT AND ENDING POINT')
 1090  FORMAT(' ENTER PROPORTIONAL COSTANT',E11.4/' SET SAMPLE
     1 HOLD TIME      ',E11.4/)
 1100  FORMAT(' ********** SUMMARY OF EXTERNAL INPUT **********'/)
 1110  FORMAT(' U(',I4,') = ',E11.4)
```

```
 1500 FORMAT(/////)
 1510 FORMAT(1H1)
 2000 FORMAT(' XF = ',E11.4,5X,'X-RANGE = ',E11.4,5X,'ITERX = ',
     1 I5,5X,'DX = ',E11.4/' XP = ',E11.4,5X,'T-RANGE = ',E11.4,
     2 5X,'ITERT = ',I5,5X,'DT = ',E11.4/' R  = ',E11.4
     3 /' Z  = ',E11.4/' I  = ',E11.4/' J  = ',E11.4//)
 2010 FORMAT(6(1X,E11.4), 2(1X,I3),4(1X,E11.4))
 2030 FORMAT(' IF YOU WOULD LIKE TO PRINT OUT, ENTER 1'/
     1' THEN, ENTER THE INTERVAL(INTEGER EXPECTED)')
 2040 FORMAT(' PROPORIOTNAL CONSTANTS TO CONTROL R,Z,XF,& XP'/
     1 ' FOR R =>',E10.3,3X,'FOR Z=>',E10.3,3X,'FOR XF=>',E10.3,3X,
     2 'FOR XP=>',E10.3/' FITR=',E10.3,3X,'FITZ=',E10.3,3X,'FITXF=',
     3 E10.3,3X,'FITXP=',E10.3/)
 2050 FORMAT('     TIME',7X,'GROTH',6X,'D(MT/F3)',5X,'NO/U(1)',8X,
     1 'R',11X,'Z',7X,'JCF',1X,'JCP',3(3X,'U(',I3,')',3X)
     2 3X,'U(',I3,')'/16X,'RATE',63X,3(F3.4,4X),F8.4//)
 2060 FORMAT(' ******** DATA ECHO SUMMARY ********')
 2070 FORMAT(' -----DIMENSIONAL VALUES -----'/
     1 ' KV = ',E10.3,3X,'KA = ',E10.3,3X,'KG = ',E10.3,3X,
     2         'KN = ',E10.3,3X,'V  = ',E10.3,3X,'VL = ',E10.3,3X,
     3         'QF = ',E10.3/
     4        ' QP = ',E10.3,3X,'QCL= ',E10.3,3X,'RHO= ',E10.3,3X,
     5         'CS = ',E10.3,3X,'CO = ',E10.3,3X,'CF = ',E10.3//
     6' ----- DIMENSIONLESS VALUES -----'/
     7' KG = ',E10.3,3X,'KA = ',E10.3,3X,'G  = ',E10.3,3X,
     8 'PR = ',E10.3,3X,'EC = ',E10.3,3X,'XCF= ',E10.3,3X,
     9 'XCO= ',E10.3/' GAMMA = ',E10.3,18X,'F  = ',E10.3//)
 2080 FORMAT(' PH(AT T=INFINITE)   = ',E10.3/
     1       ' PH(RATIO AT T=0)    = ',E10.3/
     2       ' PH(RATIO AT STEADY) = ',E10.3//)
 2090 FORMAT(1H1,' ********** DIMESIONAL OUTPUT **********'/)
 2100 FORMAT(1H1,' ********** DIMENSIONLESS OUTPUT **********'/)
 2110 FORMAT(//'  ---------- DISTRIBUTION ON U(I) ----------')
 2120 FORMAT(//)
 2130 FORMAT(' >>> INITIAL CONDITIONS ARE BASED ON INFINIT ONES <<<'//)
 2140 FORMAT(' >>> INITIAL CONDITIONS ARE BASED ON T=0 ONES <<<'//)
 2150 FORMAT(//' >>> ZERO COMPENSATOR OF U(I) WAS OFF <<<'//)
 2160 FORMAT(//' >>> ZERO COMPENSATOR OF U(I) WAS ON <<<'//)
  999 STOP
      END
*
*
*
*
      SUBROUTINE IREKAE(PH,QUE,JCF,JCP,R,Z,BRZ)
*       THIS SUBROUTINE CALUCULATES MATRIX A WITH A NEW VALUES OF
*       QUE,FH,AND PH.
*
*       ON ENTRY
*       PH=GROWTH RATE
*       QUE=CONSTANT
*       JC*=CUT SIZE POINTS
*       ITERX=ITERATION NUMBER ON X
*       DX=INCREMENT OF X
*
*       ON RETURN
*       A=MATRIX A
      COMMON/BLOCK1/A(77,77),ITERX,DX,IXINT
      IH=ITERX+2
      DO 100 J=1,IH
```

```
*       FIRST ROW
      IF(J.EQ.1) THEN
        A(J,1)=QUE*(-22+12*DX*FH(J,JCF,JCP,R,Z,BRZ)/PH)
        A(J,2)=36*QUE
        A(J,3)=-18*QUE
        A(J,4)=4*QUE
*       SECOND ROW
      ELSEIF(J.EQ.2) THEN
        A(J,1)=-4*QUE
        A(J,2)=QUE*(-6+12*DX*FH(J,JCF,JCP,R,Z,BRZ)/PH)
        A(J,3)=12*QUE
        A(J,4)=-2*QUE
*       (IM-1)TH ROW
      ELSEIF(J.EQ.IM-1) THEN
        A(J,IM-3)=2*QUE
        A(J,IM-2)=-12*QUE
        A(J,IM-1)=QUE*(6+12*DX*FH(J,JCF,JCP,R,Z,BRZ)/PH)
        A(J,IM)=4*QUE
*       IM-TH ROW
      ELSEIF(J.EQ.IM) THEN
        A(J,IM-3)=-4*QUE
        A(J,IM-2)=18*QUE
        A(J,IM-1)=-36*QUE
        A(J,IM)=QUE*(22+12*DX*FH(J,JCF,JCP,R,Z,BRZ)/PH)
*       J-TH ROW
      ELSE
        A(J,J-2)=QUE
        A(J,J-1)=-9*QUE
        A(J,J)=QUE*12*DX*FH(J,JCF,JCP,R,Z,BRZ)/PH
        A(J,J+1)=8*QUE
        A(J,J+2)=-QUE
      ENDIF
  100 END DO
      RETURN
      END
*
*
*
      SUBROUTINE SEKI(SISU,JOGEN,Y1,U)
*       SUBROUTINE SEKI INTEGRATES MOMENT EQUATIONS WITH
*       TREPEZOIDAL INTEGRATION
*
*       ON ENTRY
*       U=VECTOR U
*       ITERX=NUMBER OF ITERATION ON X
*       SISU=POWER NUMBER ON X
*       JOGEN=UPPER LIMIT OF INTEGRATION
*       DX=INCREMENT ON X
*
*       ON RETURN
*       Y1=RESULTS
      COMMON/BLOCK1/A(77,77),ITERX,DX,IXINT
      DIMENSION U(77)
      S=0
      IF(JOGEN.EQ.0)THEN
         Y=0
         GOTO 999
      ENDIF
      DO 100 J=1,JOGEN
      X=(J-1)*DX
```

```
      X1=J*DX
      J2=J+1
      A1=U(J)*X**SISU
      A2=U(J2)*X1**SISU
      S=S+(A1+A2)*DX/2
  100 CONTINUE
      Y1=PT(S)
  999 RETURN
      END
*
*
*
      SUBROUTINE STED(R,Z,XF,XP,JCF,JCP,UX)
*       SUBROUTINE STED EVALUATES STEADY STATE VALUS OF
*       JCF,JCP,AND U(I)
*
*       ON ENTRY
*       X*=CUT SIZE POINTS
*       DX=INCREMENT ON X
*       R=R
*       Z=Z
*       ITERX=NUMBER IF ITERATION ON X
*
*       ON RETURN
*       JC*=CUT SIZE POINTS TO CONTROL DESCRETED VARIABLES
*       UX=VECTOR OF CSD IN STEADY CONDITION
*
      COMMON/BLOCK1/A(77,77),ITERX,DX,IXINT
      DIMENSION UX(77)
      IM=ITERX+2
      JCF=GONU(XF/DX)
      JCP=GONU(XP/DX)
      DO 100 L=1,IM
      LLT=L-1
      X=LLT*DX
      IF(LLT.LT.JCF) THEN
        UX(L)=PT(EXP(-R*X))
      ELSEIF((LLT.GE.JCF).AND.(LLT.LT.JCP)) THEN
        UX(L)=PT(EXP(-(R-1)*XF)*EXP(-X))
      ELSE
        P3=(Z-1)*XP-(R-1)*XF
        P4=-Z*X
        UX(L)=PT(EXP(P3)*EXP(P4))
      ENDIF
  100 CONTINUE
      RETURN
      END
*
*
*
      SUBROUTINE STEDZR(XF,XP,JCF,JCP,UX)
*       SUBROUTINE STEDZR DEFINES INITIAL DISTRIBUTION OF U
*       AS ZEROS.
*       ON ENTRY
*       ITERX = MATRIX SIZE
*       XF,XP = CUT SIZE POINTS
*
*       ON RETURN
*       JC* = CUT SIZE POINTS TO CONTROL DESCRETE VARIABLES
*       UX = VECTOR OF CSD IN STEADY CONDITION
```

```
      COMMON/BLOCK1/A(77,77),ITERX,DX,IXINT
      DIMENSION UX(77)
      IM=ITERX+2
      JCF=GONU(XF/DX)
      JCP=GONU(XP/DX)
      DO 100 L=1,IM
      UX(L)=0
  100 END DO
      RETURN
      END
*
*
*
      SUBROUTINE STEDEL(XF,XP,JCF,JCP,U)
      WRITE(*,*)' NOT AVAILABLE '
      RETURN
      END
*
*
*
      SUBROUTINE PARA(R,Z,XF,XP,IP,JCF,DP,BETA,F30,FINI,F20,U)
*       SUBROUTINE PARA EVALUATES STEADY STATE VALUES OF DP,BETA,
*       F3,F2,AND F31
*
*       ON ENTRY
*       R=R
*       Z=Z
*       XF,XP=CUT SIZE POINTS
*       IP=MATRIX SIZE
*       JCF=CUT SIZE POINT CALUCULATED BY SUBROUTINE STED
*
*       ON RETURN
*       SISU=POWER NUMBER
*       JOGEN=UPPER LIMIT OF INTEGRATION
*       F***=I-TH MOMENTS
*       BETA=CONSTANT
*       DP=CONSTANT
      COMMON/BLOCK1/A(77,77),ITERX,DX,IXINT
      DIMENSION U(77)
      DP=FW(R*XF)/R**4
      DP=DP+(FW(XP)-FW(XF))*EXP(-(R-1)*XF)
      R3=Z*XP
      R4=(Z-1)*XP-(R-1)*XF
      DP=DP+(1-FW(R3))*EXP(R4)
      DP=PT(DP)
      SISU=3
      JOGEN=JCF
      CALL SEKI(SISU,JOGEN,FINI,U)
      CALL SEKI(SISU,IP,F30,U)
      SISU=2
      CALL SEKI(SISU,IP,F20,U)
      BETA=PT((R-1)/(6*DP)*FINI)
      RETURN
      END
*
*
*
      FUNCTION FH(L,JCF,JCP,R,Z,BRZ)
      IF(L.LE.JCF) THEN
```

```
         FH=R
      ELSE IF((L.GT.JCF).AND.(L.LE.JCP)) THEN
         FH=BRZ
      ELSE
         FH=Z
      END IF
      RETURN
      END
*
*
*
      FUNCTION FW(X)
*       FUNCTION FW CALCULATES GAMMA FUNCTION
      FW=1-EXP(-X)*(1+X+X**2/2+X**3/6)
      RETURN
      END
*
*
*
      FUNCTION PT(X)
      COMMON/BLOCK2/JOBP6
*       FUNCTION PT TRANCATES ARGUMENTS BY *.**, THREE DISITS.
*       ENTRY IS X AND RETURN IS PT.
      NSIZE=JOBP6
      IHATB=0
*
*       IF X IS ZERO, RETURN ZERO.   IF X IS NEGATIVE,
*       TAKE ABSOLUTE VALUE OF X.
      IF(X.GT.0) GOTO 10
      IF(X.LT.0) GOTO 5
      PT=0
      GOTO 20
*       SET FLAG AS IHATB=1 WHICH MEANS ORIGINAL X IS NEGATIVE.
    5 IHATB=1
      X=ABS(X)
*       FIND HIGHEST NUMBER OF DIGIT OF X
   10 B=ALOG10(X)
      JB=INT(B)
      Y=X*0.1**JB
      YA=INT(Y*10**NSIZE)
      YB=INT(Y*10**(NSIZE-1))
      IVS=YA-10*YB
      IF(IVS.LT.5) THEN
        JVS=INT(YB)
      ELSE
        JVS=INT(YB)+1
      ENDIF
      Z=JVS*0.1**(NSIZE-1)
      PT=Z*10.**JB
      IF(IHATB.EQ.1)PT=-PT
   20 RETURN
      END
*
*
*
      SUBROUTINE RZCOND(BETA,FINI,F2,PH)
*       SUBROUTINE RZCOND GIVES GROWTH RATE OF R-Z CRYSTALLIZER
*       WITH NO ADVANCED LIQUOR
      PH=PT((1+BETA*FINI)/(1+BETA)/F2)
      RETURN
```

```
      END
*
*
      SUBROUTINE DDCOND(FINI,F2,JFLAG1,JFLAG2,XC,TH)
*       SUBROUTINE DDCOND CALUCURATES GROTH RATE CONDITION
      COMMON/BLOCK3/DDSP,DDQF,DDKV,DDKA,DDRC,RVM31,RVM2,DDTAU,DDKG,DDCS
      COMMON/BLOCK4/DDEC,DDF,DDCF,DDKN,DDCO,DDV,S0,ANNO,PMT
      COMMON/BLOCK5/DAKA,DAG,DAKG,DAPR,XCM,XCO
*       ON ENTRY
*       FINI    = F31, PARTIAL THIRD MOMENT
*       F2      = F2, SECOND MOMENT
*       TH      = DIMENSIONLESS TIME
*       JFLAG2 =INITIATION CONTROL FLAG
*       ON RETURN
*       JFLAG1 = ERROR FLAG
*       XC      =DIMENSIONLESS SOLUTE CONCENTRATION
      JFLAG1=0
      RG=(R-GAMMA)**2/(R-1)
      IF((R-1).EQ.0)RG=1
      AA=DAKA*F2*DAG*DAKG/(2*DDEC)
      BB=DDKV*FINI*DAG*RG
      BB=BB+DAKA*F2*DAG*DAKG*(DAPR-1)/2
      BB=-BB/DDEC
      CC=DDKV*FINI*DAG*(DAPR-1)*RG
      CC=CC+DDF*(XCF+1-GAMMA)
      SS=BB**2-4*AA*CC
      IF(SS)100,200,300
  100 WRITE(*,*)' IMAGINARY CONCENTRATION OCCURS. NO PASS TO SOLVE'
      JFLAG1=1
      GOTO 999
  200 XC=XCO*(1+AA*BB*TH/2)+AA*BB**2*TH/4
      XC=XC/(1-AA*XCO*TH-AA*BB*TH/2)
      IF(JFLAG2.EQ.1)XCM=(BB*XCO/2+BB**2/4)/(-XCO-BB/2)
      IF(JFLAG2.EQ.2)XCM=XC
      GOTO 999
  300 AL=BB-SQRT(SS)
      BL=BB+SQRT(SS)
      ACE=(2*AA*XCO+BL)/(2*AA*XCO-BL)
      XC=BL*EXP(TH*SQRT(SS))-AL*ACE
      XC=XC/(2*AA*(ACE-EXP(TH*SQRT(SS))))
      IF(JFLAG2.EQ.1)XCM=-BL/(2*AA)
      IF(JFLAG2.EQ.2)XCM=XC
  999 RETURN
      END
*
*
*
      SUBROUTINE STUCK(PH,LUG,DELAY,ITSTEP)
*       SUBROUTINE STUCK DOES STUCK THE CURRENT PH AND ROLL UP
*       DELAY(I)
      DIMENSION DELAY(3)
      IF(ITSTEP.EQ.1)ITOLD=0
      ITPOS=ITSTEP-ITOLD
      IF(ITPOS.LT.LUG) THEN
         DELAY(3)=DELAY(2)
      ELSE
         DELAY(3)=(DELAY(1)+2*DELAY(2)+PH)/4
         DO 50 N=2,3
         DELAY(N-1)=DELAY(N)
   50    END DO
```

```
          ITOLD=ITSTEP
        ENDIF
  100   RETURN
        END
*
*
*
        SUBROUTINE TRNS(U,PH,X,T,FINI,F2,F3,JOBP1,IM,R,Z,JCF,JCP,
       $                KSTEP,JSTEP,ICH1,ICH2,ICH3,ICH4)
*         SUBROUTINE TRNS TRANSRATES DIMENSIONLESS ARGUMENTS TO
*         DIMENSIONAL ARGUMENTS
        COMMON/BLOCK3/DDQP,DDQF,DDKV,DDKA,DDRO,RVM31,RVM2,DDTAU,DDKG,DDCS
        COMMON/BLOCK4/DDEC,DDF,DDCF,DDKN,DDCO,DDV,GO,ANNO,PMT
        DIMENSION U(77),AU(77)
*         ON ENTRY
*         U        = MATRIX
*         PH       = DIMENSIONLESS GROWTH RATE
*         X        = DIMENSIONLESS PARTICLE SIZE
*         FINI     = DIMENSIONLESS PARTIAL THIRD MOMENT
*         F2       = DIMENSIONLESS SECOND MOMENT
*         F3       = DIMENSIONLESS THIRD MOMEMT
*         JOBP1    = U-OUTPUT CONTROL FLAG
*         IM       = MATRIX SIZE
*         ICH*     = OUTPUT ALLOCATIONS OF U(I)
*         KSTEP    = COUNTER ON U-OUTPUT
*         JSTEP    = INTERVAL ON U-OUTPUT
        G=PH*GO
        FL=X*DDV*GO/DDQP
        TM=T*DDTAU
        FF3=F3*ANNO*(DDV*GO/DDQP)**3
        FNO=ANNO*U(1)
        IF((JOBP1.EQ.1).AND.(KSTEP.EQ.JSTEP)) THEN
            DO 100 I=2,IM
            AU(I)=ANNO*U(I)
  100       END DO
        WRITE(5,*)(AU(I),I=1,IM)
        ENDIF
        WRITE(5,1000)TM,G,FF3,U(1),R,Z,JCF,JCP,AU(ICH1),AU(ICH2),AU(ICH3),
       $ AU(ICH4)
 1000   FORMAT(6(1X,E11.4),2(1X,I3),4(1X,E11.4))
        RETURN
        END
*
*
*
          SUBROUTINE KAI(T,IM,U,R)
*
*         SUBROUTINE COMP LOOKS FOR CONVERGENCE OF THE SERIES,
*         F=I+A*T+(A*T)**2!+...
*         AND CALCULATES F AS A RESULT
*
*         ON ENTRY
*         AI       = MATRIX A WHICH IS EVALUATED IN SUBROUTINE IREKAE
*         T        = TIME INCREMENT
*         IM       = MATRIX SIZE
*         U        = VECTOR OF U(I) IN MAINSUKI
*         R        = VECTOR OF REXT(J,I) IN MAINSUKI
*
*         ON RETURN
*         U        = VECTOR OF U(I) IN MAINSUKI
```

```
*
*
      COMMON/BLOCK1/A(77,77),ITERX,DX,IXINT
      COMMON/BLOCK2/NSIZE
      DIMENSION AO(77,77),ANEW(77,77),AOLD(77,77),U(77),R(2,77)
*
      N=0
      DO 50 J=1,IM
      DO 50 I=1,IM
      AO(J,I)=0
      IF(J.EQ.I)AO(J,I)=1
   50 END DO
  100 N=N+1
      KFLAG=0
      TN=T**N
      DEN=1
      IF(N.EQ.1) THEN
      DO 150 J=1,IM
      DO 150 I=1,IM
      AOLD(J,I)=0
      IF(J.EQ.I)AOLD(J,I)=1
  150 END DO
      ENDIF
      DO 200 K=1,N
      DEN=DEN*K
  200 END DO
      CALL MALTI(AOLD,A,ANEW,IM,IM,IM)
      DO 300 J=1,IM
      DO 300 I=1,IM
      SA=ABS(ANEW(J,I)*TN/DEN-AOLD(J,I)*T**(N-1)*N/DEN)
      AOLD(J,I)=ANEW(J,I)
      AO(J,I)=AO(J,I)+ANEW(J,I)*TN/DEN
      IF(SA.GT.0.1**NSIZE)KFLAG=1
  300 END DO
      IF(KFLAG.EQ.1)GOTO 100
      DO 350 J=1,IM
      AOLD(J,1)=U(J)
  350 END DO
      CALL MALTI(AO,AOLD,ANEW,IM,IM,1)
      DO 400 I=1,IM
      U(I)=ANEW(I,1)
  400 END DO
      DO 450 J=1,IM
      AOLD(J,1)=R(1,J)
  450 END DO
      CALL MALTI(AO,AOLD,ANEW,IM,IM,1)
      DO 500 I=1,IM
      U(I)=PT(U(I)+T*(R(2,I)+ANEW(I,1))/2)
  500 END DO
      RETURN
      END
*
*
      SUBROUTINE MALTI(A1,A2,AOUT,IM,NJ,NI)
*       SUBROUTINE MALTI CALCURATES A MATRIX X A MATRIX
*       BOTH MATRICIES ARE EXPECTED SQUARE
*
*       ON ENTRY
*       A1       = LEFT MATRIX
*       A2       = RIGHT MATRIX
```

```
*       IM      = MATRIX SIZE
*
*       ON RETURN
*       AOUT    = A1*A2
*
      DIMENSION A1(77,77),A2(77,77),AOUT(77,77)
*
*
      DO 100 J=1,NJ
      DO 100 I=1,NI
      AOUT(J,I)=0
      DO 100 K=1,IM
      AOUT(J,I)=AOUT(J,I)+A1(J,K)*A2(K,I)
  100 END DO
      RETURN
      END
*
*
*
      FUNCTION GONU(X)
*       FUNCTION GONU TRANCATES ARGUMENTS AT *.**.
*       ENTRY IS X AND RETURN IS GONU.
*
*       IHATB = FLAG FOR SIGNS OF X;POSITIVE=0,NEGATIVE=1
*
      IHATB=0
*
*       IF X IS ZERO, RETURN ZERO.   IF X IS NEGATIVE,
*       TAKE ABSOLUTE VALUE OF X.
      IF(X.GT.0) GOTO 10
      IF(X.LT.0) GOTO 5
      GONU=0
      GOTO 20
*       SET FLAG AS IHATB=1 WHICH MEANS ORIGINAL X IS NEGATIVE.
    5 IHATB=1
      X=ABS(X)
*       FIND FLOATING POINT OF X
   10 B=ALOG10(X)
      JB=INT(B)
      NSIZE=JB+1
      Y=X*0.1**JB
      YA=INT(Y*10**NSIZE)
      YB=INT(Y*10**(NSIZE-1))
      IVS=YA-10*YB
      IF(IVS.LT.5) THEN
        JVS=INT(YB)
      ELSE
        JVS=INT(YB)+1
      ENDIF
      Z=JVS*0.1**(NSIZE-1)
      GONU=Z*10.**JB
      IF(IHATB.EQ.1)GONU=-GONU
   20 RETURN
      END
*
```

NOMENCALTURE

(NOT INCLUDING LETTERS USED IN APPENDICES B AND C)

A coefficient in Equation (2-2) or a matrix defined in Equation (A-13).

A_s surface area of crystals.

B a constant matrix.

B^o nucleation rate.

C concentration of solute.

c constant in Equation (2-4).

Δc supersaturation.

D diffusivity.

D_p dimensionless function in steady state growth rate equation.

d constant in Equation (2-2).

$f_{j,k}$ dimensionless normalized moments of distribution growth rate equation.

G dynamic growth rate.

$\bar{G}$ steady growth rate.

H Heaviside step function.

h a removal function.

i nucleation growth kinetic sensitivity parameters.

j magma-dependent nucleation kinetics exponent.

K proportional controllers.

K_c proportional constant for the experimental controller.

K_s a matrix defined by Equation (6-9).

k a constant defined by Equation (6-7).

k_g crystal growth rate constant.

k_N nucleation rate constant.

k_v shape factor for volume.

L crystal size.

L_F particle size at the upper cut size of the fines dissolver.

L_P particle size at the lower cut size of the product classifier.

M a matrix.

M_t suspension density.

m nucleation order.

n dynamic population densities.

$\bar{n}$ steady population densities.

n^o dynamic nuclei density.

$\bar{n}^o$ steady nuclei density.

$n_t{}^o$ nuclei density at t.

$\bar{n}_t{}^o$ nuclei density defined by Equation (6-11).

P_E external production rate.

Q a matrix in Equation (B-26).

Q_F feed rate.

Q_P mixed production rate.

$Q(L)$ flow rate as a function of L.

R a matrix in Equation (B-26) or recycle ratio of the fines dissolver.

r external input.

S parameters in frequency domain.

s supersaturation rate.

t time.

u dynamic dimensionless population densities.

$\bar{u}$ steady dimensionless population densities.

u_o dynamic nuclei density.

$\bar{u}_o$ steady nuclei density.

V crystallizer volume.

w gamma function.

x dimensionless crystal size.

x_F dimensionless crystal size at the upper cut size of the fines dissolver.

x_P dimensionless crystal size at the lower cut size of the product classifier.

y output from a system.

z recycle ratio of the product classifier or a coodinate in Appendix B.

GREEK LETTERS

δ dimensionless particle size increment.

θ dimensionless time.

$\Delta\theta$ dimensionless time increment.

λ eigen values.

ρ crystal density.

Φ transition matrices.

ϕ dimensionless growth rate.

ω a parameter in Equation (4-2) and (4-3).

NOMENCLATURE IN APPENDIX B

A a matrix defined in Equation (B-1).

B a matrix defined in Equation (B-1).

C a matrix defined in Equation (B-2).

D a matrix defined in Figure B-1.

E a matrix defined in Figure B-1.

H Hamilton function.

K controllers.

M a matrix including desired poles.

n rank of matrix A

P a matrix defined by Equation (B-9) or (B-31).

p rank of matrix C.

Q a positive-semidefinite symmetric matrix.

R a positive-definite symmetric matrix.

r an input vector.

S a positive-semidefinite symmetric matrix.

T a null space of matrx C.

u an input vector.

V^* an eigen space constructed by negative eigen value $_i$.

x a state variable.

y an output vector.

GREEK LETTERS

θ a new state variable defined by Equation (B-27).

λ eigen values.

ξ a state variable defined by Equation (B-10).

ϕ Lagrange function.

φ a function in Equation (B-17).

ψ a function in Equation (B-18).

NOMENCALTURE IN APPENDIX C

A a parameter defined by Equation (C-21).

a a parameter defined by Equation (C-11).

b a parameter defined by Equation (C-12).

c solute concentration or a parameter defined by Equation (E-14) thereafter.

c_F feed liquor solute concentration.

c_s saturated solute concentration.

F ratio of feed rate to produce rate, Q_F/Q_P.

f_2' the dimensionless normarized second moment in crystallizer.

$f_{3,1}'$ the dimensionless normarized third moment of distribution growth rate equation.

G growth rate.

$\bar{G}$ steady growth rate.

g a dimensionless gorwth rate defined by Equation (C-9).

k_A area shape factor.

k_g growth rate kinetic constant.

k_g normarized growth rate kinetic constant.

k_v volumatric shape factor.

m_2 the second moment in crystallizer.

$\bar{n}^o$ steady nuclei density.

P_R solid rate destroyed in a fines destruction system.

Q_{CLA} slurry flow rate of clear liquor advance.

Q_F slurry based liquor feed rate.

Q_P slurry based liquor product rate.

R recycle ratio of a fines dissolver.

r_p a parameter defined by Eqaution (C-10).

t time.

V crystallizer volume.

x dimensionless supersaturation ratio defined by Equation (C-6).

x_0 dimensionless supersaturation ratio at t=0.

x_{cF} dimensionless supersaturation ratio in feed liquor.

x_s dimensionless supersaturation ratio defined by Equation (C-22) or (C-23).

GREEK LETTERS

α a parameter defined by Equation (C-19).

β a parameter defined by Equation (C-20).

γ $(Q_P + Q_{CLA})/Q_F$.

ϵ_c void or liquor fraction.

θ dimensionless time.

ρ crystal density.

τ residence time, V/Q_P.

REFERENCES

1. Beckman, J. R., and A. D. Randolph, "Crystal Size Distribution Dynamics in a Classified Crystallizer: Part II, Simulated Control of Crystal Size Distribution", AIChE J., **23**, 510 (1977).

2. Berglund, K. A., and M. A. Larson, "Growth of Contact Nuclei of Citric Acid", AIChE Symp. Ser., **78**, 215, 9 (1982).

3. Blem, K. E., and K. A. Ramanarayanan, "Generation and Growth of Secondary Ammonium Dihydrogen Phosphate Nuclei", AIChE Annual Meeting Chicago, Nov. 1985 (1985).

4. Broman, A., "Introduction to Partial Differential Equations: from Fourier Series to Boundary-Value Problems", Addison-Wesley Pub. Co., Reading, Massachusetts, (1970).

5. Calver, M. B., J. M. Blair, W. N. Selander, and D. G. Stewart, "Short course and workshop on numerical solution of ordinary and partial differential equatins using FORSIM", Mathematics and Computation Branch, Chalk River Nuclear Laboratories, Atomic Energy of Canada Limited, Chalk River, Ontario KOJ IJO (1976).

6. Carrier, G. F., "Partial differential Equation: Theory and Practice", Academic Press, (1976).

7. Cellier, F., "ECE501 text", Department of electrical and computer engineering, the University of Arizona, Arizona, (1985).

8. Cellier, F., and M. Rinuvall, "Computer Aided Control System Design", First European Simulation Congress ESC83, Aachen, September 1983 Proceedings, (1983).

9. Epstein, M. A. F., and L. Sowul, "Phase Space Analysis of Limit Cycle Development in CMSMPR Crystallizers Using Three-dimentional Computer Graphics", AIChE Symp. Ser., No193, **76**, 6 (1980).

10. Forsythe, G. E., M. N. Malcolm and C. B. Moler, "Computer Methods for Mathematical Computation', Printice-Hall, (1977).

11. Kailath, T., "Linear System", Printice-Hall, New York, (1981).

12. Lapidus, L., and R. Luus, "Optimal Control of Engineering Process", Blaisdell Publishing Company, Waltham, Massachusetts, (1967).

13. Leden, B., M. H. Hamza, and M. A. Sheirah, "Different Methods for Estimation of thermal Diffusivity of a Heat Diffusion Process", Proc. of the IFAC Symposium on Parameter Estimation and State Identification, (Eylihoff, ed.), Delft 1973, 639, (1974).

14. Moler, C. B., "MATLAB Users' Guide", Department of Computer Science, University of New Mexico, New Mexico, (1981).

15. Nuttall, H. E., "Computer Simulation of Steady State and Dynamic Crystallizers", Ph.D. dissertation, University of Arizona, Arizona, (1971).

16. Nyvlt, J., and J. W. Mullin, "The periodic behaviour of continuous crystallizers", Chem. Eng. Sci., 25, 131, (1970).

17. Randolph, A. D., "Size Distribution Dynamics in a Mixed Suspension", Ph.D. Thesis, Iowa State University, Ames, Iowa, (1962).

18. Randolph, A. D., J. R. Beckman, and Z. I. Kraljevich, "Crystal Size Distributin dynamics in a Classified Crystallizer: Part I, Experimental and Theoritical Study of Cycling in a Potassium Chloride Crystallizer", AIChE J., 23, 500, (1973).

19. Randolph, A. D., G. L. Beer, and J. P. Keener, "Stability of Class II Classified Product Crystallizer with Fines Removal", AIChE J., 19, 1140, (1973).

20. Randolph, A. D., and M. A. Larson, "Theory of Particle Process", Academic Press, (1971).

21. Randolph, A. D., L. Chen, and A. Tavana, "Feed Back Control of CSD in a KCL Crystallizer with Fines Dissolving", AIChE Meeting at New Orleans, (1986).

22. Randolph, A. D., and E. T. White, "Modeling Size Distribution in the Prediction of Crystal Size Distribution", Chem. Eng. Sci., 32, 1067, (1977).

23. Randolph, A. D., E. T. White, and Chi-Chu David Low, "On-Line measurement of Fine Crystal Response to Crystallizer Disturbance", Ind. Eng. Chem. Process Des. Dev., 20, 496, (1981).

24. Rugh, W. J., "Nonlinear System Theory", The Johns Hopkins University Press, Baltimore, Maryland, (1981).

25. Schetzen, M., "The Volterra and Wiener Theories of Nonlinear Systems", John Willey & Sons, New York, (1980).

26. Sheirah, M. A. and M. H. Hamza, "Optimal Control of Distributed Parameter Systems", Int. J. of Control, 19, No.5, 891, (1974).

27. Sherwin, M., R. Shinner, and S. Katz, "Dynamic Behavior of The Isothermal Well-stired Crystallizer with Classified Outlet", Chem. Eng. Prog. Symp. Series, 65, 75, (1969).

28. Sibert, W. P., "User Manual for MARK V", Department of Chemical Engineering, University of Arizona, Arizona, (1973).

29. Timm, D. C., and M. A. Larson, "Effect of Nucleation Kinetics on the Dynamic Behavior of a Continuous Crystallizer", AIChE J., 14, 453, (1968).

30. Vemuri, V., and W. J. Karplus, "Digital Computer Treatment of Partial Differential Equations", Prentice-Hall, (1981).