Numerical Bifurcation Analysis and Parallel Processing

Raymond A. Adomaitis† and Ali Çinar

Department of Chemical Engineering
Illinois Institute of Technology
Chicago, Illinois 60616

Abstract

A numerical scheme has been developed for determining the global static bifurcation characteristics of a nonlinear system. Based on the desirable properties of equilibrium solution surfaces in certain coordinate systems, its utility also extends to the task of classifying the different multiplicity patterns found in the remaining parameter space. An outline of the numerical technique and a discussion of its inherent parallelizability will be presented in this paper.

FG02-87ER 13696

To be presented as in the session titled "The Use of Advanced Computer Architectures in Chemical Engineering" at the 1988 A.I.Ch.E. Annual Meeting, Washington, D.C.
Nov. 27- Dec. 2, 1988

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

† Present address: Department of Chemical Engineering, Princeton University, Princeton, New Jersey 08544

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Introduction

In the analysis of nonlinear systems, one may wish to present the description of the system's equilibrium behavior in the form of plots of the value of a distinguished state variable as a function of a distinguished parameter. Called bifurcation plots, their construction poses large computational problems in the case of nonlinear systems described by a large vector of equations. Some of the more subtle problems (as opposed to the more obvious compute-intensive ones) include isolated solution branches—segments of the equilibrium solution arc which make the 2-dimensional continuation schemes used for speed difficult to automate. Other issues that must be addressed involve the qualitative changes in bifurcation behavior that occur in the remaining parameter space and the lack of multivariate numerical schemes useful in the study of this phenomenon.

A `FORTRAN` bifurcation analysis package has been developed by the authors that does both bifucation diagram construction and the numerical singularity analysis used in classifying the different multiplicity patterns exhibited by an arbitrarily large nonlinear system in a user-selectable parameter plane. The numerical schemes developed take full advantage of the desirable behavior of static equilibrium solution surfaces residing in spaces chosen with cognizance of the Nice Hypersurface Theory. We will not present a detailed exposition of this theory as that discussion has been directed to another outlet (Adomaitis and Çinar, 1988).

This paper will focus on the development of the bifurcation analysis code with emphasis on parallel processing and how well it meshes with our numerical techniques. It will begin with a prerequisite introduction to singularity theory by way of a simple example followed by an outline of the numerical scheme developed for finding singular sets on a solution manifold. The features of the technique that make the emulating code a simple target for parallelization will be discussed along with its natural extension to solution manifold construction. An application of `BIFF`, the bifurcation analysis package, to the analysis of a tubular laboratory-scale chemical reactor will conclude this paper.

Nice Solution Surfaces

As was mentioned in the introduction, both the construction of bifurcation diagrams and the task of classifying the different bifurcation behaviors benefit by an understanding of static equilibrium solution surfaces in appropriate (state; parameter)-spaces. Consider now such a surface in $(u; \phi, \psi)$-space (Figure 1) and the implications of the following theorem:

Theorem (The definition of a nice space). *Given the set $\{u, \phi\}$ where u is the distinguished state variable, ϕ is the distinguished (bifuration) parameter, and $\phi \in \mathcal{S}_e \cup \mathcal{S}_{ne}$, the equilibrium solution manifold in $(u, \phi, \psi_1, \psi_2, \ldots, \psi_n)$-space is unique and well-behaved when $\psi_i \in \mathcal{S}_e$ where $i = 1, 2, \ldots, n$.*

Proof. The proof of the above theorem is actually more of a definition since the $\mathcal{S}$-sets need to be defined:

$\mathcal{S}_{ee}$= The set of parameters of a nonlinear system whose members consist of parameters that can be expressed explicitly as functions of the remaining parameters and state variables.

$\mathcal{S}_{ie}$= The set of parameters of the same system that are implicitly extractable. In other words,

$$\mathcal{S}_{ne} \cap \mathcal{S}_{ie} = \emptyset \qquad \text{and if} \qquad \psi \in \mathcal{S}_{ie}, \qquad \frac{\partial x}{\partial \psi} \neq 0$$

over the relevant ranges of the states and parameters, where x is any state or parameter.

$\mathcal{S}_e$= $\mathcal{S}_{ie} \cup \mathcal{S}_{ee}$

$\mathcal{S}_{ne}$= The set of parameters that are neither explicitly nor implicitly extractable.

So we see that the uniqueness of nice surfaces is assured by their definition. The well-behavedness of the solution manifold stems from the continuous nature of most physical models and can be proven by the implicit function theorem. []

The theorem's utility should not be assumed to be proportional to its simplicity: for example, in all of the nonadiabatic chemical reactors studied by the authors to this time, v_{in} (the reactant stream feed temperature) $\in \mathcal{S}_e$. Thus, when the creation of bifurcation plots in the (effluent reactant concentration, feed flowrate)-plane† is the task at hand, *the construction of the solution manifold in $(u; Q, v_{in})$-space guarantees the elucidation of all*

† The feed flowrate Q in nonadiabatic reactors is generally an $\mathcal{S}_{ne}$-parameter, hence, the troublesome isolated solution branches.

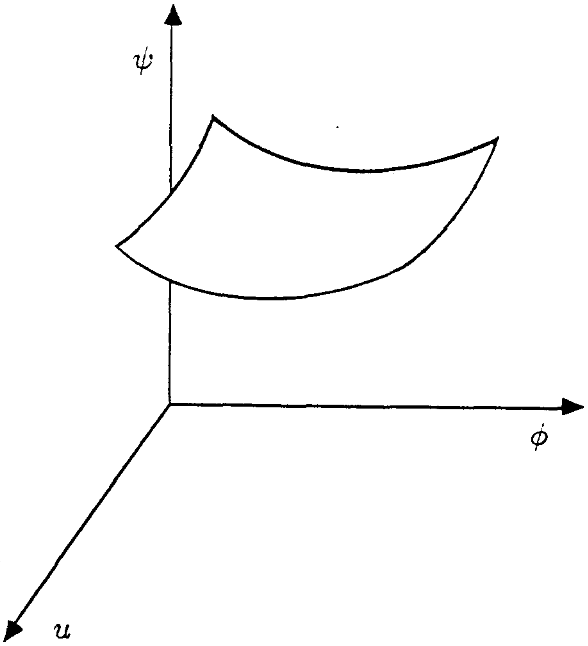


Figure 1. An equilibrium solution manifold in a nice space.

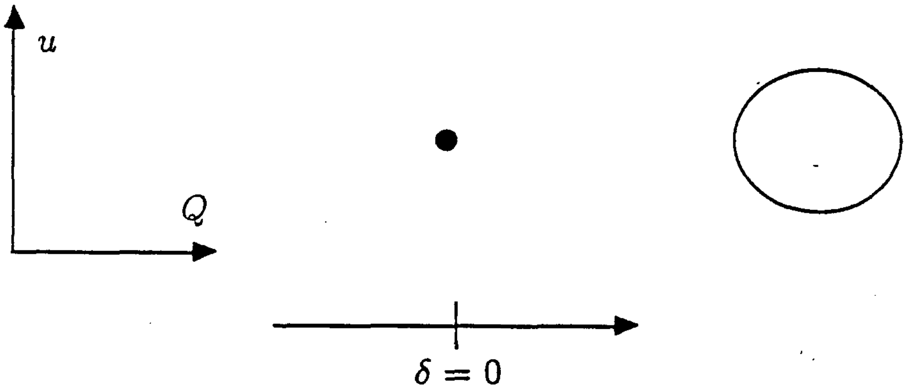


Figure 2. The appearance of an isolated solution branch. The singularity occurs at $\delta = 0$.

static equilibrium solutions. As will be illustrated in the next section, a second benefit of thinking in terms of solution manifolds is enjoyed in a numerical singularity analysis scheme used for classifying bifucation behaviors.

Singularity Theory

To introduce the singularity theory and some of the properties of universal unfoldings in nice spaces, consider an example of an isolated solution branch in the bifurcation diagram plane. If one observes the behavior of the isola as another parameter is varied, one would find a unique value of this secondary parameter at which the isola disappears (see Figure 2). It is easy to envision the general shape of the solution manifold that would give rise to such behavior: a smooth surface with a local minimum defining the point of disappearance. This geometrical information can be taken advantage of in the development of a numerical scheme for finding these points since it can be formally proven that the neighborhood of any singular point has generic characteristics particular to that type of singularity. For example, the manifold surrounding the isola center singularity on an arbitrary solution surface [defined by the scalar equation $g(u; Q, v_{in}) = 0$ where $v_{in} \in \mathcal{S}_e$] in the neighborhood of the singular point $(u^*; Q^*, v_{in}^*)$ satisfies

$$g(\bar{u}; \bar{Q}, \bar{v}_{in}) = S(\bar{u}; \bar{Q}, \bar{v}_{in}) G(x; \lambda, \delta)$$

where

$$\bar{u} = u - u^* \qquad \bar{Q} = Q - Q^* \qquad \bar{v}_{in} = v_{in} - v_{in}^*$$

$S \neq 0$, and in the case of the isola center singularity,

$$G(x; \lambda, \delta) = x^2 + \lambda^2 - \delta = 0$$

where the following diffeomorphisms,

$$x = \Xi(\bar{u}; \bar{Q}) \qquad \lambda = \Lambda(\bar{Q}) \qquad \delta = \Delta(\bar{v}_{in})$$

are defined by the singularity theory (Golubitsky and Schaeffer, 1985).

Because the singular point satisfies $x = 0$, $\lambda = 0$, and therefore $\delta = 0$, the conditions

$$\frac{\partial \delta}{\partial x} = \frac{\partial \delta}{\partial \lambda} = 0$$

are also satisfied. With this and the arguments of the diffeomorphisms in mind, we find

$$\frac{d\Delta}{dv_{in}}\frac{\partial v_{in}}{\partial Q} = 2\lambda\frac{d\Lambda}{dQ} + 2x\frac{\partial \Xi}{\partial Q}$$

and

$$\frac{d\Delta}{dv_{in}}\frac{\partial v_{in}}{\partial u} = 2x\frac{\partial \Xi}{\partial u}.$$

Thus, the singular point (u^*, Q^*) can be determined by iterating on the following Newtonian convergence schemes:

$$u^{\nu+1} = u^{\nu} - \left(u^{\nu} - u^{\nu-1}\right)\frac{\partial v_{in}\left(u^{\nu}\right)}{\partial u}\left[\frac{\partial v_{in}\left(u^{\nu}\right)}{\partial u} - \frac{\partial v_{in}\left(u^{\nu-1}\right)}{\partial u}\right]^{-1}$$

and as $\partial v_{in}/\partial u \to 0$ (obtained by iterating on the above) or by using the points (u, Q^{ν}) and $(u, Q^{\nu-1})$,

$$Q^{\nu+1} = Q^{\nu} - \left(Q^{\nu} - Q^{\nu-1}\right)\frac{\partial v_{in}\left(Q^{\nu}\right)}{\partial Q}\left[\frac{\partial v_{in}\left(Q^{\nu}\right)}{\partial Q} - \frac{\partial v_{in}\left(Q^{\nu-1}\right)}{\partial Q}\right]^{-1}$$

where ν denotes the iterate number.

The moral of the story is that differential information computed on the solution surface can be used as a guide in finding singular sets. Furthermore, if one begins with a reasonable estimate of the singularity's location, the convergence of our algorithm to the singular point is *guaranteed.*

Computation of Manifold Gradients

In each orthogonal step of the Newtonian convergence scheme, the value of the gradient of v_{in} with respect to one of the axes of the bifurcation parameter plane is required at two different points on the manifold. The simplest means of computing these values is by finite-difference using the cross-shaped set of points illustrated in Figure 3 (b++ is used in determining the hysteresis singularity, the other codimension-1 singularity). From this set of points, the second derivatives with respect to u and Q can also be computed. These are useful in identifying the singularity captured, for if both derivatives are of the same sign the isola center singularity has been found; if not, the simple bifurcation has been captured.

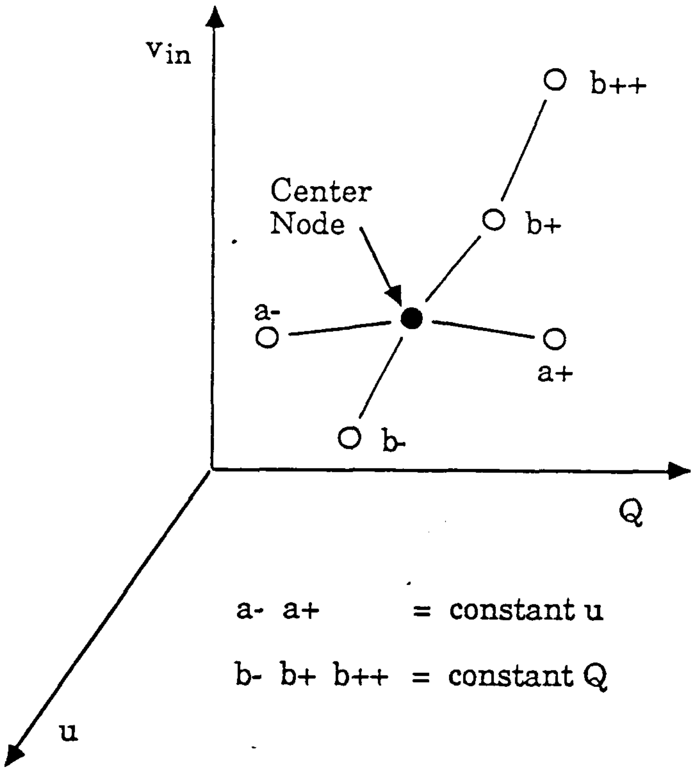


Figure 3. The node net used in finding and identifying codimension-1 singularites.

With the outline of the numerical algorithm at hand finally complete, we can now move on to the central issue of this paper—the parallelization of the analysis technique. First, it should be noted that many mathematical and numerical details of our analysis scheme have changed and continue to change in time, hence, coarse parallelization would make for less painful code changes. Combine this requirement with the notion of computing independent points on a solution manifold and the obvious parallelization scheme illustrated in Figure 4 emerges. In the processor activity chart of Figure 4, we see that (for large models) there are two computational steps that are much larger that the rest. The first involves determining the equilibrium point corresponding to the $(u^{\nu+1}; Q^{\nu+1})$ dictated by the singularity analysis algorithm†. This allows the computation of the satellite points of the node net to then be computed via the efficient predictor-corrector arc-length continuation scheme discussed in Adomaitis and Çinar (1987). Iterating on this singularity analysis algorithm will produce the singular point in short order. After that, the value of a tertiary distinguished parameter can be changed by an increment and the singularity analysis scheme is once again set in motion.

We note from the processor activity chart that all of the processors do the same thing except for the parallelized DO-loop. While other users of the campus mainframe may not appreciate this inefficient use of processors that might otherwise be used by those users, it does allow for the minimum number of program variables that must be classified a global over the processors. Keeping these variables to a minimum helps minimize some of the subtle bugs‡ that arise when using parallelized code on multiple processors.

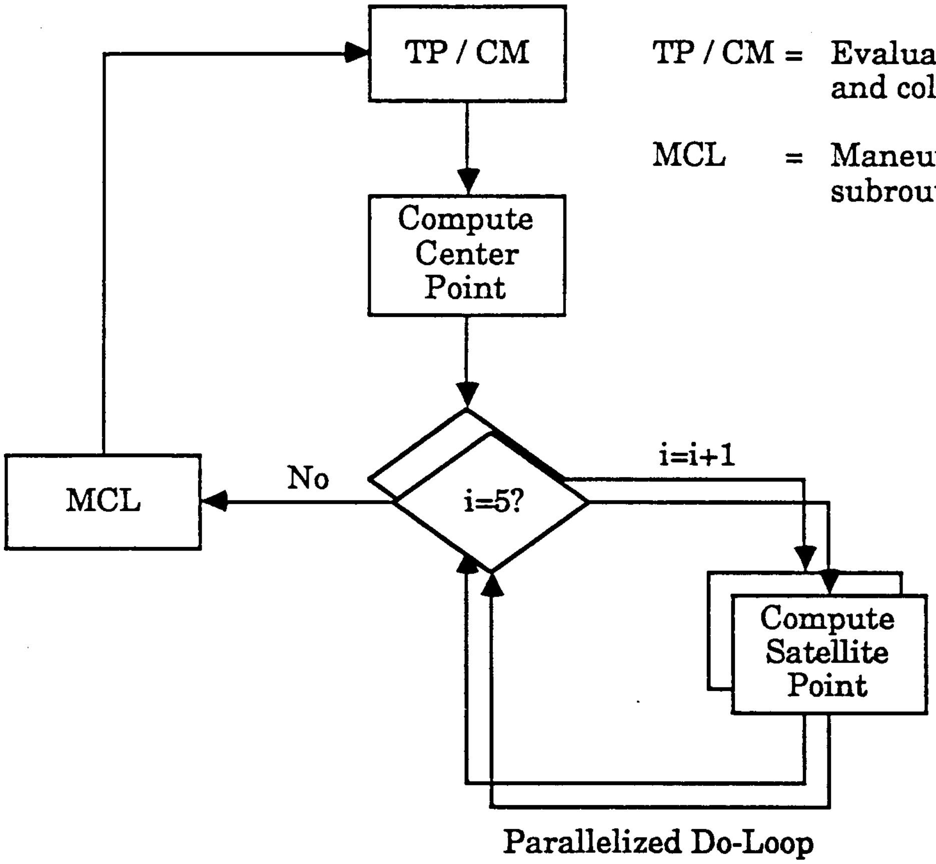
Determining the benefits of parallelization is an easy task when considering large

† This step and the next consume large amounts of CPU time because they amount to several Newton-Raphson iterations on a very large vector of nonlinear algebraic equations.

‡ One of most sinister of these bugs is encountered when a program variable, say a, is defined as global over the processors and the statement

$$\mathrm{a} = \mathrm{a} + \epsilon \qquad \text{where} \qquad \epsilon \ll 1$$

is found in a segment of code run on more than one processor. This will result in a slow accumulation of error that may not be noticed until disaster strikes.



TP / CM = Evaluate transport parameters and collocation matrices

MCL = Maneuvering control logic subroutine

Time →

CPU 1	TP / CM	Compute Center Point	Comp. Satellite Pt. 1	I / O	MCL
" 2	"	"	" 2	Idle	"
" 3	"	"	" 3	"	"
" 4	"	"	" 4	"	"

1 iteration on the Newtonian convergence scheme

Figure 4. A flowchart and processor activity chart describing the computational side of the numerical singularity analysis technique.

models (more than 20, or so, equations). For the isola center and simple bifurcation singularity searches, the effective speed-up over sequential code is

$$\frac{\text{compute 5 equilibrium points}}{\text{compute 2 equilibrium points}} = 2.5 \text{ times faster}$$

and for the hysteresis singularity search,

$$\frac{\text{compute 6 equilibrium points}}{\text{compute 2 equilibrium points}} = 3 \text{ times faster.}$$

The results of one application of BIFF can be seen in Figure 5 where the classification of six bifurcation behaviors is shown in the (u_{in}, v_{in})-plane [the (feed CO concentration, feed temperature)-plane]. The reactor model describes a tubular packed-bed autothermal CO oxidation reactor and consists of 7 PDE's. Orthogonal collocation on finite elements was used to simplify it to slightly more than 100 ODE's. More details of the reactor model and its simplifications can be found in Adomaitis (1988).

Manifold Construction

By providing a classification of the different bifurcation behaviors exhibited by a chemical reactor, the values of the selectable remaining parameters can be chosen so that the reactor behaves in a desirable way. The remaining task involves the actual bifurcation diagram construction, a job which also benefits from nice surfaces and parallelized algorithms.

As shown in Figure 1, the equilibrium solution manifold in a nice space is fairly well-behaved. It is easily traced by the continuation scheme discussed in Adomaitis and Çinar (1987), but a much faster algorithm is proposed in Figure 6. In this scheme, a zig-zagging ribbon of manifold is computed by determining three points at one time along with an evaluation of the stability of the system linearized about the center point. This scheme takes full advantage of the predictor-corrector arc-length continuation technique since the satellite points radiate from a previously determined satellite point (the center point or point 1 of Figure 6). Computationally speaking, this scheme is very efficient since not much time is spent on redundant calculations, and since the time needed to do the stability check is on the same order as the computation of a satellite point, little time is wasted in the resynchronization of the programs after the parallelized-DO. All of the bifurcation diagrams of Figure 5 were computed by this scheme.

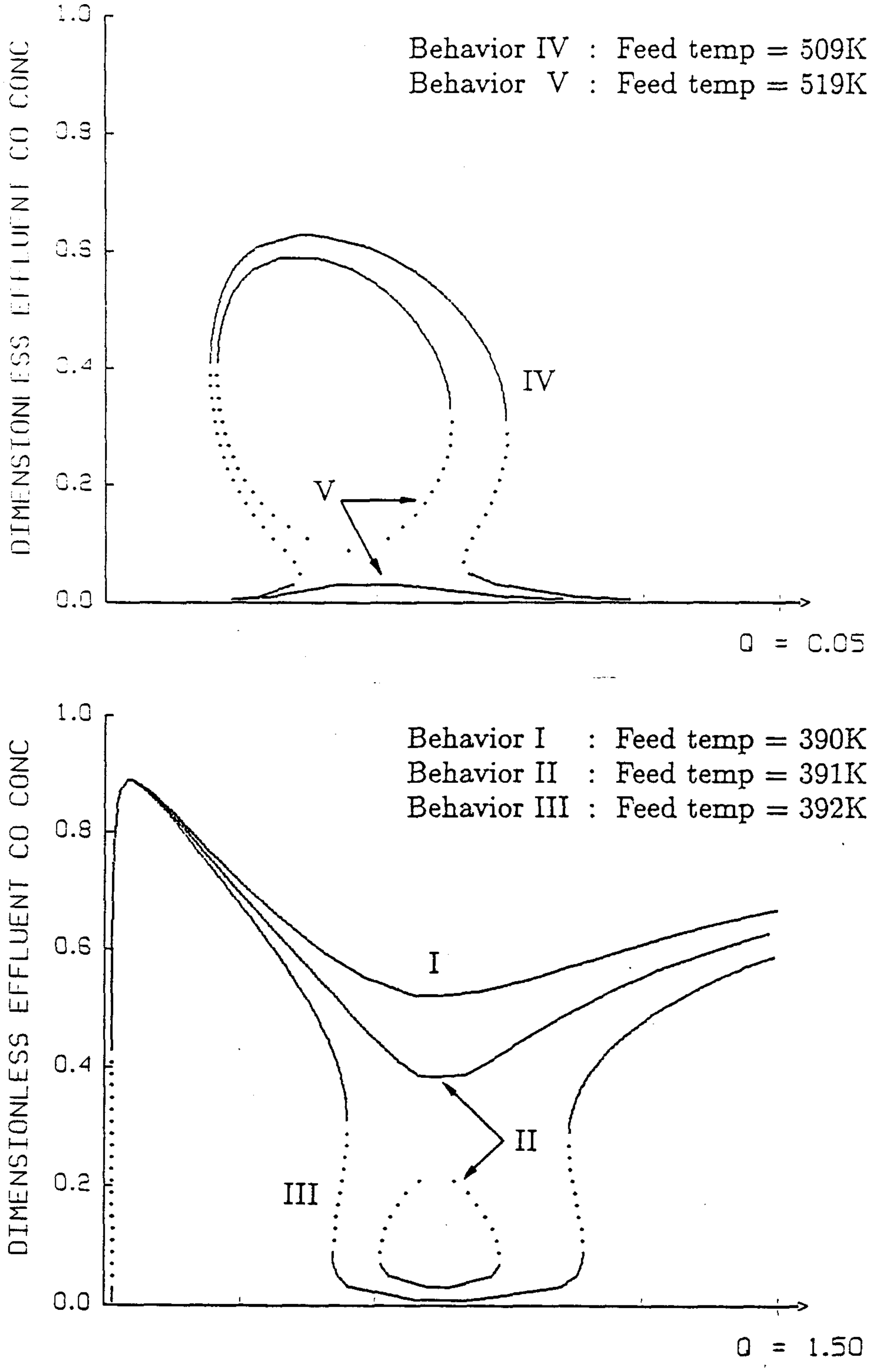


Figure 5. Continued.

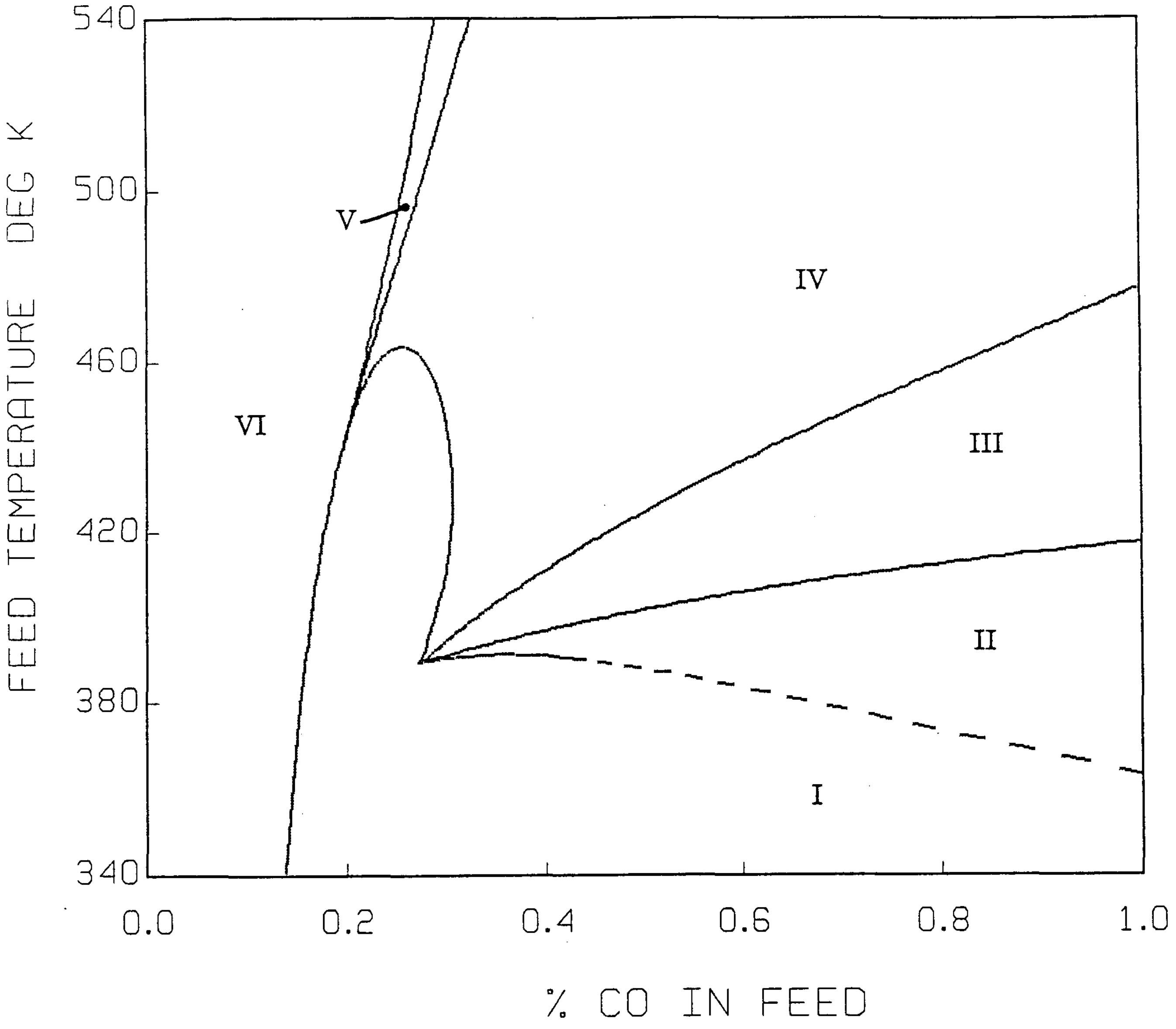
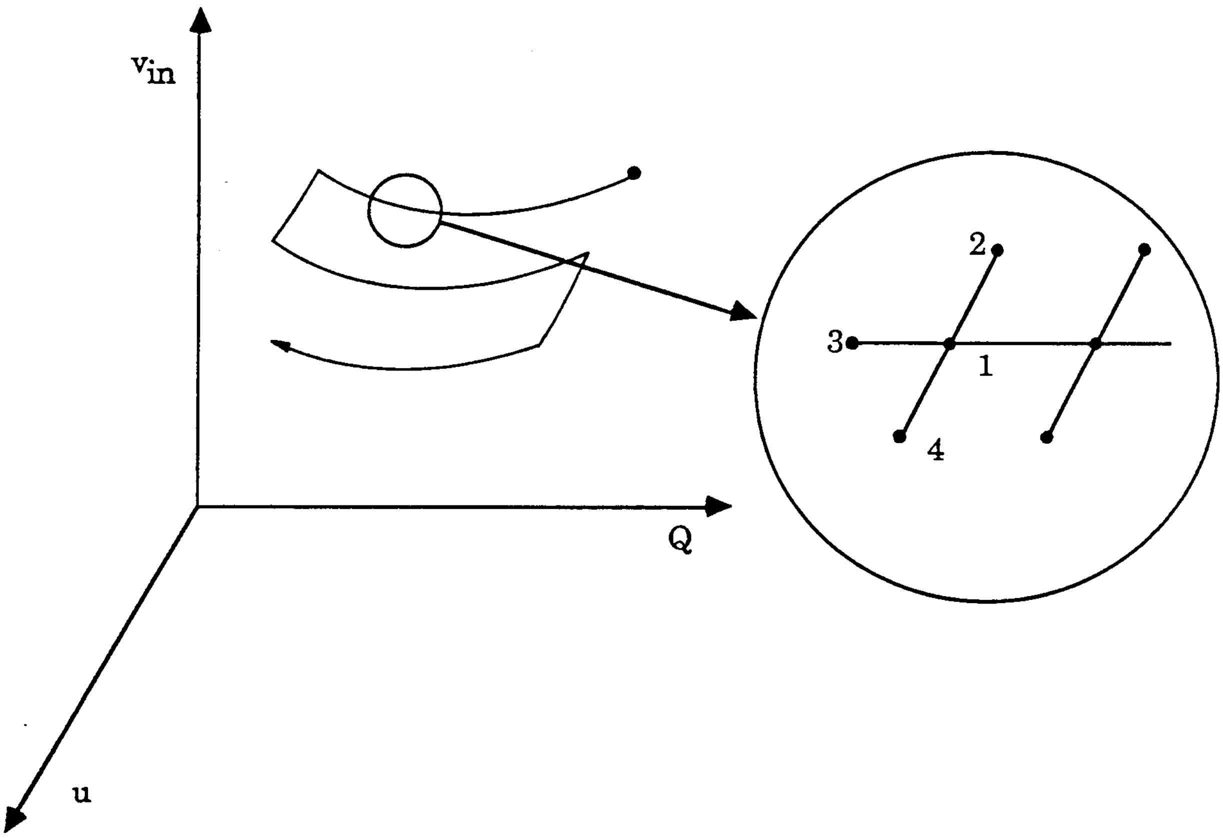


Figure 5. The classification diagram along with sample bifurcation diagrams illustrating the six bifurcation behaviors exhibited by the tubular CO oxidation reactor. Behavior VI contains no multiplicities.



Time →

CPU 1	TP / CM	Compute Satellite Point 2	I / O	MCL
" 2	"	" 3	Idle	"
" 3	"	" 4	"	"
" 4	"	Linear Stability of Point 1	"	"

Figure 6. The parallelized manifold construction scheme and its processor activity chart.

Computational Issues

A 2500 line FORTRAN program named BIFF has been written to do both the manifold construction and numerical singularity analysis tasks. The program is set up in a structure similar to an ODE solver—the user needs only to change the subroutine that contains the model and sections in the main program that pertain to problem-specific I/O demands and the like. All of the results presented in this paper have been computed by BIFF. The FORCE macros (IITMAX, 1988) make parallelization of the main DO-loop a simple task and also makes deparallelization of the code (so that it can be run on a serial machine) likewise simple.

The computer used was a six processor Encore Multimax with a Weitek floating-point accelerator. Each processor on this machine is capable of 2 MIPS. With a model consisting of more than 100 state variables and with all arithmetic done in double precision, a single separating line in the classification plane (Figure 5) took roughly 2 hours to compute and a manifold can take up to one week. When faced with the prospect of one CPU month of time required for the construction of a manifold on a serial machine, the benefits of parallel processing make the extra programming effort worthwhile.

Concluding Remarks

A FORTRAN program has been written that does both bifurcation diagram and classification diagram construction for nonlinear systems described by a large vector of nonlinear equations. The numerical technique developed has unique features that make for large computational efficiency increases even with coarse parallelization. Since the efficiency of our parallelization scheme will increase as the codimension of the singularity sought does, our future work in the development of a complete classification of all the bifurcation behaviors displayed by the tubular CO oxidation reactor will depend heavily on parallelized code.

Acknowledgements

The partial financial support of the Department of Energy (DE-FG02-87ER13696) is gratefully acknowledged.

References

Adomaitis, R. A., 1988, The bifurcation and modification of equilibrium states in chemical reactors. Ph.D. thesis, Illinois Institute of Technology, Chicago, Illinois.

Adomaitis, R. A., and A. Çinar, 1988, Numerical singularity analysis. Paper submitted to *Chem. Engng Sci.*

Adomaitis, R. A., and A. Çinar, 1987, The bifurcation behavior of feedback-controlled chemical reactors. 1987 A.I.Ch.E. Annual Meeting, New York, New York.

Golubitsky, M. and D. G. Schaeffer, 1985, *Singularities and Groups in Bifurcation Theory, Volume I.* Springer-Verlag, New York.

IITMAX, 1988, All of the documentation required for incorporating the FORCE macros was obtained from our Encore Multimax's on-line manuals.