A MODIFIED CELL-TO-CELL MAPPING METHOD
FOR NONLINEAR SYSTEMS*

Mo-Hong Chou
Institute of Mathematics, Academia Sinica
Nankang, Taipei 11529, Taiwan, R.O.C.

(Received June 1992; accepted September 1992)

Abstract—A variant of C. S. Hsu's cell-to-cell mapping method for nonlinear systems is proposed to cope with the situation in which the global analysis is started with a fairly coarse structure of state cells. The analysis is improved through an iterative cell refinement and cell lumping process. The efficacy of our approach is demonstrated by solving the van der Pol equation as well as some equations modeling the enzyme kinetics.

1. INTRODUCTION

When studying a nonlinear dynamical system, one is often interested in finding the asymptotically stable equilibrium states or periodic motions, and the global domains of attraction of these attractors, if any. This is the motif of Hsu [1] and his co-workers to develop the so-called cell-to-cell mapping methods. The basic idea behind these methods is to consider the state space not as a continuum but rather as a collection of a large number of state cells, with each cell taken as a state entity.

In the literature there are two types of mapping: simple cell mapping and generalized cell mapping. The accuracy or reliability of the simple cell mapping is, in general, limited by the associated cell size. Therefore, a huge number of cells are expected to accompany this simple algorithm. To alleviate this situation, generalized cell mapping is then introduced at the expense of more algorithmic complexity. In practice, these two mapping methods are employed together in the following way. A pilot simple cell mapping "compatible" with the pursued generalized cell mapping is used to reveal the general structure of the given dynamical system, such as the locations of periodic motion. These then form the rudimentary persistent groups in the context of generalized cell mapping.

Even with the generalized cell mapping, the structures of importance in the state space, like the persistent groups and the boundaries of domains of attraction, are often too crude and have to be improved through some iterative or adaptive refinement techniques. In Hsu's methods, a critical cell is refined by partitioning it into $K = \prod_{i=1}^{N} k_i$ subcells, where each $k_i$ is an odd integer, and $N$ is the order of the dynamical system in consideration. One reason for such an arrangement of these $k_i$'s is to maintain some sort of compatibility between successive levels of iteration. This requirement, however, will lead the computational effort, CPU time plus the data storage, covering the critical regions to increase at least $3^N$ times after each level of refinement. This is expected to be an expensive task as the iteration continues. Therefore, only one or two iterations are performed in practice, provided that the initial structure of state cells is fine enough.

In this work we propose a variant of Hsu's method, which bears no aforementioned interlevel compatibility, and therefore, significantly reduces the computational cost spent between successive iterations. The variation is schematically shown in Figure 1. We use such a variant to explore the situation when an iterative refinement process is started with a fairly coarse structure of state cells. To make our approach work, some modifications with respect to Hsu's original method are

---

*This research work was supported in part by the National Science Council under Grant NSC 82-0208-M-001-112.
made accordingly. These include the detection of singular cells, the detection of multidomicile transient cells, and the exploration of putting the iteration scheme into a cell addition/deletion fashion for controlling the expansion of cell number.

For simplicity of expression, our idea is explained in terms of dynamical systems of order two. Extending it to systems of higher order is straightforward. The efficacy of our approach is demonstrated by solving the van der Pol equation as well as some equations modeling the enzyme kinetics.

![Schema of Cell Refinement](image)

Figure 1. Schema of mark sampling and cell refinement: (a) Hsu's method; (b) present algorithm.

2. THE METHOD

Without loss of generality, the dynamical system we are concerned with is assumed to be governed by an autonomous system of nonlinear ordinary differential equations (ODE). According to the cell-to-cell mapping methods, the state space is considered as a collection of a large number of state cells, including a huge sink cell representing the domain outside of our interest.

In the context of simple cell mapping, each cell is associated with exactly one image cell which is determined by integrating the ODE over a prescribed time interval, with a particular point like the cell center as the initial value. Let the finite collection of state cells be numbered as 1, 2, ..., M, with the sink cell at the first place. Their image cells are stored in the target array \( T(i), \) \( i = 1, 2, ..., M \). Then, the dynamical system is simulated by the iterated mappings: \( T(i), T(T(i)), \ldots \), etc. In this context, one is primarily concerned with the structures of periodic motions.

The result of simple cell mapping is given by the array \( P(i) \) recording the associated periodic group of the \( i \)th cell, and by the array \( Q(i) \) recording the stages of the \( i \)th cell away from the associated periodic chain of cell mapping. Namely, the \( k \)th group of periodic motion is formed by those \( i \)'s such that \( P(i) = k \) and \( Q(i) = 0 \). Figure 2 presents a flowchart to construct \( P \) and \( Q \).

Clearly, the accuracy or reliability of such a simulation is limited by the individual cell size which justifies the representation of the associated image cell. To relax this limitation, generalized cell mapping is proposed: Each cell is allowed to have several image cells which are determined in the same way as shown in the context of simple cell mapping, but with suitably sampled initial values. The compatibility between these two types of cell mapping is defined by requiring the image cell determined by the simple mapping is also one of the sampling image cells associated
Figure 2. Flowchart for constructing arrays $P$ and $Q$ in connection with simple cell mapping.
with the generalized mapping. As far as sampling is concerned, we shall adopt the "interior and boundary" method as sketched in Figure 1.

The counterpart of periodic motions in the context of generalized cell mapping is the formation of persistent groups in which each member cell has probability one to revisit itself. In practice, these groups are formed through a pilot simple cell mapping compatible with the generalized one. Let there be $J$ groups of periodic motions associated with the selected simple cell mapping (SSCM), and $S(j), j = 1, 2, \ldots, J$ be an array containing a representative cell for each group. Then, due to compatibility, there exist at most $J$ persistent groups. The exact number of these groups wholly depends on the subsequent generalized mapping of these cells $S(j)$.

As before, let the state cells be numbered as $1, 2, \ldots, M$. But the target array $T(i), i = 1, 2, \ldots, M$ become an array of multi-component integer vectors taking account of the multiplicity of image cells. For example, each $T(i)$ is a nine-component vector in the present study, see Figure 1. When a particular image cell, say the $i^{th}$ one, is referred to, we will use the notation $T(i, i)$.

Starting with the array $T(i)$, the generalized cell mapping is recorded by an array $G(i)$ which tells us whether a cell is persistent or not. Namely, the $i^{th}$ cell is persistent if $G(i) > 0$; Otherwise, it is transient, with $G(i) = -2$ for the time being. Figure 3 presents a flowchart to construct this $G$. Note that, for clarity, these charts only present some basic parts of the general theory. For a more comprehensive version, we refer the reader to the book written by Hsu [1].

Figure 3. Flowchart for constructing array $G$ in connection with generalized cell mapping.
Figure 3. continued.
Since we are interested in starting the global analysis with a fairly coarse structure of state cells, the final stage of this structure will become rather irregular as the analysis gets improved through iterative cell refinement. Therefore, some modifications to Hsu's original method are necessary, and are listed as follows.

2.1. Basic Data Structure

To cope with the irregularity, the cell-to-cell mapping method adopted here is based on the following "mark and cell" data structure expressed in a C-like language [2].

```c
struct mark {
    float map[2];
    struct mark *dsn[2], *morn, *lsp, *nxp;
    struct cell *hom;
};
struct cell {
    float site[2];
    short gen, sgr, stg, ggr, dom, sig;
    struct mark *cntr, *vrtx[4], *side[4];
    struct cell *son[4], *par, *ist, *nxt;
};
```

In view of above expression and Figure 1, each state cell is identified by `site`: its center's coordinate, and by `cntr`, `vrtx`, `side`: its nine image marks' pointers. Each mark is represented by `map`: the outcome of a particularly sampled ODE integration, and by `horn`: the pointer of the current home cell containing the coordinate of `map`. The collection of state cells is organized into a tree structure through two pairs of pointers: (`son`, `par`) and (`dsn`, `morn`), and at the same time into a linked list through another two pairs of pointers: (`lst`, `nxt`) and (`lsp`, `nxp`).

The integral variables `gen`, `sgr`, `stg`, `ggr`, `dom` and `sig` respectively stand for the generation of a cell in the history of iteration, the periodic group number assigned by simple mapping, the stage(s) leading to a periodic group in the context of simple mapping, the persistent group detected by generalized mapping, the counter of domiciles of a cell in the context of generalized mapping, and the indicator of cell singularity. Associated with this data structure, the arrays `P`, `Q`, `G` mentioned previously in the simple/generalized cell mapping algorithm have been replaced componentwise by `sgr`, `stg`, `ggr` respectively.

Note that at the very beginning of the iteration process, the state cells usually form a rectangular array covering the domain of interest, with `gen = 1` for each cell. The tree and linked-list structures are primarily for coping with cell refinement and cell lumping.

2.2. Detection of Singular Cells

A cell is said to be singular if it is likely to contain an equilibrium point of the given dynamical system. In continuum cases, such a point can be detected by the orientation of the surrounding vector field. This idea is copied to the present discrete case. In Hsu's methods, one has to check the "triplets" formed by the mutually adjoining cells. Such an approach, however, is not convenient in connection with a rather irregular cell structure. Therefore the singularity of a cell is checked here in a self-contained way as follows.

Let `{mi | i = 1, ..., 9}` be the maps linked to a cell `C`, and `{si | i = 1, ..., 9}` be the pre-images of the corresponding maps, namely the initial values. The later information is provided by `site`, `gen` and the initial cell size. Then, `C` is claimed to be singular, with `sig ≠ 0`, if there exist `i, j, k` such that `1 ≤ i, j, k ≤ 9` and

\[(m_i - s_i) \cdot (m_k - s_k) ≤ 0\] and \[(m_i - s_i) \cdot (m_k - s_k)^\perp ≥ 0;\]
\[(m_j - s_j) \cdot (m_k - s_k) ≤ 0\] and \[(m_j - s_j) \cdot (m_k - s_k)^\perp ≤ 0.\]

where `(·)` stands for the inner product of two vectors, and `(·)\perp` means to rotate a vector counterclockwise by 90°. When the integration interval used to obtain maps is large enough, it
is possible to produce spurious singular cells by the above criterion. If this is unwanted as far as the cell refinement is concerned (see Section 2.4), our suggestion is to reduce the integration interval or to replace the quantities such as $m_i - s_i$ by $F(x = s_i)$ where $x = F(x)$ governs the dynamical system.

2.3. Detection of Multidomicile Transient Cells

A transient cell in the context of generalized mapping has multiple domiciles if it has the probability to be led to different persistent groups. These transient cells form the boundaries of domains of attraction. When the initial structure of state cells is coarse enough, some singular cells—although persistent analytically—will be detected as transient in the intermediate stages of the iteration process. This situation, in turn, will block the detection of some important multidomicile transient cells, such as those covering an unstable limit cycle. To remedy, we allow a transient cell to increase its counter of domiciles, $\text{dom}$, by one, when it has the probability to be led to a transient singular cell.

2.4. Cell Refinement

When the resolution of the persistent groups as well as the boundaries of domains of attraction are not satisfied, one has to refine the corresponding cells, namely cells with $\text{ggr} > 0$ or $\text{dom} > 1$. The refinement is sketched in Figure 1 in which each daughter cell increases the inherited $\text{gen}$ value by one. The refining procedure is extended to cover each singular cell and its neighboring cells, in order to resolve its persistency as encountered in Section 2.3, and to localize the related equilibrium point. For an individual cell, the refinement is terminated when its $\text{gen}$ exceeds some prescribed value.

The so-called interlevel compatibility presented in Hsu's methods is to ask if the marks of the parent cells are inherited by the daughter cells in a one-to-one fashion as indicated by Figure 1a. One can use such a compatibility to save the involved selected simple cell mapping. However, this condition is not observed in our scheme, as indicated by Figure 1b. In view of Figure 1, our alternative can save more significant computational effort between successive levels of iteration.

2.5. Cell Lumping

Given a cell $C$ in the current level of iteration. If $C$ and its sibling cells determined by the tree structure shown in Section 2.1 are all transient, non singular, and of single and common domicile, then $C$ and its sibling are lumped together. In other words, they will be replaced by their parent cell in the next level of iteration. This action brings about cell deletion, and takes place along with the cell addition procedure Section 2.4. The aim of such an addition/deletion approach is at controlling the expansion of cell number along the iteration process.

3. WORKED EXAMPLES AND DISCUSSION

We begin with a comparative study in connection with the van der Pol equation [1, 3]:

\[
\ddot{x} + \zeta (x^2 - 1) \dot{x} + x = 0.
\] (1)

To apply the cell-to-cell mapping methods, we put Equation (1) into a system of first order ordinary differential equations as follows.

\[
\dot{x} = -\zeta \left( \frac{x^3}{3 - x} \right) - y,
\] (2)

\[
\dot{y} = x.
\] (3)

With $\zeta = 1.0$, the domain of interest is $(x, y) \in [-3, 3] \times [-3, 3]$. We then partition this domain into $10 \times 10$ uniform cells, which are fairly coarse as compared to the $101 \times 101$ partitioning employed by Chiu and Hsu [4]. In both studies, the cell mapping is generated by Runge-Kutta integration [5] over 2.5 units of time duration.
Through seven levels of iterative refinement, the centers of persistent cells (marked by dots) and singular cell (marked by circle) in terms of \((x, \dot{x})\) coordinate are shown in Figure 4. The final result, Figure 4d, is fairly close to that obtained by Hsu’s algorithm with two levels of refinement [1, 4]. The total of state cells used in our scheme is 5467, while some 19425 (= 10201 + 8 x 296 + 8 x 882) cells are used in connection with Hsu’s algorithm.

The ratio of total (persistent resp.) cell count between successive iterations or generations is plotted in Figure 5a (b resp.). In view of these figures, the cell number is nearly doubled after each level of iteration, while it is nearly tripled in Chiu and Hsu’s persistent case.

Next, we study the following two-compartment enzyme model [6, 7]:

\[
\begin{align*}
\dot{s}_1 &= (s_0 - s_1) + (s_2 - s_1) - \rho R(s_1), \\
\dot{s}_2 &= (s_0 - s_2) + (s_1 - s_2) - \rho R(s_2).
\end{align*}
\]

(4) (5)

Where the reaction term

\[
R(s) = \frac{s}{1 + s + \kappa s^2}.
\]

(6)
ADAPTIVE C-C MAP FOR S-S ENZYME KINETICS

Figure 6. Seven-level iterative cell-to-cell mapping method for one-compartment enzyme model with \( \rho = 100, \kappa = 1.0, s_0 = 26 \): (a)–(c) the location of centers of singular cells (circles) as well as multidomicile transient cells (dots) at level 1, 3, 5, respectively; (d) successive ratio of total cell count along iteration: — with cell lumping; ······ without.

ADAPTIVE C-C MAP OF S-S ENZYME KINETICS, GEN = 7

Figure 7. Final (seventh level) result of one-compartment enzyme model. Shown are the equilibrium states marked by 1(sink), 2(source), 3(saddle), and their respective domains of attraction depicted by solid curves.
Figure 8. Seven-level iterative cell-to-cell mapping method for one-compartment activator-inhibitor model with $s_0 = 110, a_0 = 500, \alpha = 0.2, \kappa = 0.1$ and $\rho = 3.25$: (a)–(c) the location of centers of persistent, singular and multidomicile transient cells at level 3, 5, 7, respectively. The final depiction includes a stable equilibrium, an Unstable limit cycle, and a Stable limit cycle.

(d) Successive ratio of total cell count along iteration: --- with cell lumping; . . . . without.

By varying the parameters $\rho$, $\kappa$ and $s_0$, Equations (4) and (5) can exhibit a number of non-symmetric equilibrium states in addition to the symmetric ones, as demonstrated in [6] by continuation techniques. We are then interested in finding their respective domains of attraction. As an example, we set $\rho = 100$, $\kappa = 1.0$, $s_0 = 26$ and then investigate this system by the proposed cell-to-cell mapping method. In this case, the domain of interest is $(s_1, s_2) \in [0, 30] \times [0, 30]$; with a uniform $10 \times 10$ initial partition. The ODE integration is over 1.0 units of time duration. Figure 6 shows the intermediate results of the seven-level iterative refinement of the locations of the centers of multidomicile transient cells (marked by dots), and the singular cells (marked by circles) as well. Also shown in this figure is the ratio of total cell count between successive iterations. The computational cost is far less than that spent in the van der Pol case. Shown in Figure 7 is the final result where the equilibrium states are marked by 1(sink), 2(source), 3(saddle), respectively, and the boundaries of domains of attraction are depicted by solid lines.

Finally, we study the following one-compartment activator-inhibitor model [6, 7]:

$$\dot{s} = (s_0 - s) - \rho R(s, a), \quad \dot{a} = \alpha(a_0 - a) - \rho R(s, a). \quad (7)$$

Where the reaction rate

$$R(s, a) = \frac{sa}{1 + s + \kappa s^2}, \quad \kappa > 0. \quad (9)$$

By varying the parameters $s_0$, $a_0$, $\alpha$, $\kappa$ and $\rho$, Equations (7) and (8) can exhibit Hopf bifurcation with “jump” (see [6, pp. 497–499]). In general, this phenomenon is brought about by the coexistence of a stable equilibrium state, a stable limit cycle, and an unstable limit cycle, when the parameters are located in some critical region. To show this situation, let $s_0 = 110, a_0 = 500, \alpha = 0.2, \kappa = 0.1$ and $\rho = 3.25$. The domain of interest in this case is $(s, a) \in [0, 55] \times [0, 150]$, and is partitioned into $20 \times 20$ uniform cells initially. The ODE integration is over 8.0 units of time duration. Figures 8a to 8c show several stages of the seven-level iterative refinement of the location of the centers of persistent, singular, and multidomicile transient cells. At the end of the iteration, it comes out with a clear depiction of a stable equilibrium,
an unstable limit cycle, and a stable limit cycle. However, as shown in Figure 8d, the computational cost is the most expensive one among the studied cases, most efforts are spent in resolving the unstable limit cycle.

In view of these worked examples, we note that the mapping time interval as well as the numerical or arithmetic accuracy associated with the ODE integration should be compatible with the cell size occurring in the final stage of the iteration process. If the interval is too short or the accuracy is too low, the integration will be smeared out by the round-off effect inherent in the cell-to-cell mapping. This situation will have significant influence on resolving an unstable limit cycle, for example.

As a final remark, we shall discuss the cell deletion aspect of our scheme. In view of Figures 5, 6d, and 8d, the cell lumping as proposed in Section 2.5 in general yields a more favorable successive ratio of cell number along the iteration. The cell number reduced by this method is from below 4% in the van der Pol case to above 15% in the enzyme reaction cases. These preliminary results show that, to make it more promising, one needs a more sophisticated lumping process in which a state cell may have a polygonal shape.

REFERENCES