

A Support State Marker Method for Detecting Discontinuities in Simulation of Dynamic Systems

Qiong Yuan, Dalong Ding
 Computer and Science Institute,
 Wuhan Donghu University,
 Wuhan, 430212, China
 E-mail: yqszb@126.com

Guangming Dai
 Computer Institute,
 China University of Geosciences,
 Wuhan, 430074, China
 E-mail: gmdai@cug.edu.cn

Guohui Hong
 Computer Institute,
 Wuhan University,
 Wuhan, 430072, China
 E-mail: hghwx@126.com

Abstract--The design of discontinuity facilities is a critical factor in the success of many simulation efforts. This paper discusses the problems caused by detecting a discontinuity in terms of a change of sign in the value of a discontinuity function and describes an alternative which uses a change of the state marker value to suit the simulation of discontinuous systems in engineering applications. Subroutines which implement the method have been successfully built into integration package for stiff system with discontinuities.

Keywords--simulation; discontinuity; state marker

I. INTRODUCTION

The problem of dealing with discontinuities is often encountered in the simulation of mechanical, electrical, control and other dynamic systems. Such system are usually described by a set of ordinary differential equations of the form

$$\frac{dy_i}{dt} = f_{ij}(t, y_1, y_2, \dots, y_n)$$

$$i=1,2,\dots,n \quad j=1,2,\dots,m$$

with a set of discontinuity functions $\phi_k(t, y_1, y_2, \dots, y_n)$. The finding of discontinuities lies in two stages. Firstly, the presence of a discontinuity is detected in an integration step; secondly, this discontinuity is located as accurately as user specification demands. As was generally accepted, the presence of a discontinuity is detected by a change of sign in the value of a discontinuity function [1][2][3][4].

In this paper, the problems caused by this sign rule of detecting discontinuities for some application are discussed, and an alternative which uses a change in the value of a state marker is presented.

II. THE SIGN RULE OF DETECTING DISCONTINUITIES

Many techniques are available for handling discontinuities. Although methods of locating discontinuities are various, methods of detecting discontinuities are nearly

the same. Generally, steps of detecting discontinuities are as follows:

- A. The discontinuity functions ϕ_k are evaluated at the end of each integration step.
- B. A check is made to determine if change of sign in the value of any ϕ has happened, or a test is made for negative sign in the product $S = \phi_t \times \phi_{t+h}$.
- C. If the sign of a discontinuity function ϕ has changed or the value of S is found to be negative at end of a step, it indicates a discontinuity somewhere in that step.
- D. If a discontinuity is predicted, its time will be located as accurately as user specification demands. The integration will be adjusted to reach this and then restart. If none is found, the integration will continue normally.

In many cases, this sign rule of detecting discontinuities will be a successful prediction criterion. However, there are three sources which may affect the efficiency of the sign rule for some applications:

- 1) One of the conditions $\phi_k=0$ is necessary but not sufficient for the presence of a discontinuity. Note that a discontinuity location processes consequently as soon as any ϕ passes through zero, this may raise an extra location.
- 2) The sign of the zero value of discontinuity function is indeterminate, This problem is not trivial and is shown by an example.

Example:

$$\frac{dy}{dt} = \begin{cases} 0.5 & (0 \leq y < 0.055 \vee y \geq 0.325) \\ 3 & (0.055 \leq y < 0.325) \end{cases}$$

$$y(0) = 0, \quad t \in [0, 0.5]$$

Two discontinuity functions are defined as:

$$\phi_1 = y - 0.055$$

$$\phi_2 = y - 0.325$$

Two states are:

State 1 :

Condition $\phi_1 < 0 \vee \phi_2 \geq 0$ Equation $\frac{dy}{dt} = 0.5$

State 2:

Condition $\phi_1 \geq 0 \wedge \phi_2 < 0$ Equation $\frac{dy}{dt} = 3$

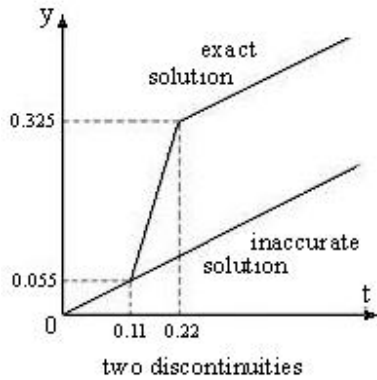


Figure1. Solution Curve

The result which is solved by routines with the sign rule of detecting discontinuities, for instance Hay's routines, is illustrated in Fig.1. However, when a few Changes which ought not to affect the solution take place:

Condition of State1: $\phi_1 \leq 0 \vee \phi_2 > 0$;

Condition of State2: $\phi_1 > 0 \wedge \phi_2 \leq 0$;

the result is grossly inaccurate (see Fig.1). This inaccurate result is due to inconsistency in the classification of the sign of the zero value of discontinuity functions. In Hay's discontinuity detection subroutine, the zero value of any ϕ is classed as positive. The value of discontinuity functions is distinguished into two classes: ≥ 0 and < 0 . User's subroutine which sets the system state marker must have same distinguishing feature.

Go further into a question that the zero value of a discontinuity function should be classed as positive or negative. An analysis to the question is shown in Table 1.

TABLE 1. CLASSIFICATION OF THE VALUE OF DISCONTINUITY FUNCTIONS

ϕ	classification	result
ϕ from < 0 to > 0 	Class zero under positive $\geq 0, < 0$	correct
	Class zero under negative $\leq 0, > 0$	mistake
ϕ from > 0 to < 0 	Class zero under positive $\geq 0, < 0$	mistake
	Class zero under negative $\leq 0, > 0$	correct

Table 1 shows the zero value of a discontinuity function ϕ should be classed as condition which determines a new

system state. If the point at which ϕ passing through zero is located accurately, it requires the behavior of ϕ to be known in advance.

3) In some applications, the zero value of a discontinuity function has to be distinguished separately; that is, the value of discontinuity functions is distinguished into five classes: $> 0, = 0, < 0, \geq 0$ and ≤ 0 .

When discontinuities are detected by the sign rule, these problems as above must be considered. However, it is difficult to do these for some applications. That is why the sign rule of detecting discontinuities can not be adapted for some discontinuous systems.

III. THE STATE MARKER METHOD OF DETECTING DISCONTINUITIES

To overcome the shortcomings of the sign rule, an alternative which uses a change in the value of a state marker is proposed. The philosophical point of this state marker method is that a discontinuity is straightforwardly detected by a change in the value of the system state marker ISTATE. Each value of ISTATE corresponds to a state of the system. The steps of this method are as follow:

- A. ϕk are evaluated at the end of each integration step.
- B. ISTATE is determined by ϕk .
- C. If a change of the value of ISTATE is not found, the integration proceeds normally.
- D. If the value of ISTATE change from ISTATELAST to ISTATENEW, a discontinuity is predicted. A discontinuity location will follow and then the integration will restart under a new state of the system.

Compared with the sign rule, this method can best tackle those problems as described in the previous section. If that one of ϕk passing through zero does not cause a change of ISTATE, the present method will not predict a discontinuity. This can avoid an extra location. In the present method, the problem of the zero value of discontinuity functions is solved by a procedure which sets the state marker value for the new system state. As soon as the point at which a discontinuity function ϕ passing through zero is located as accurately as user specification demands, the value of this ϕ is set zero and other discontinuity functions are evaluated at this point. Then the current value of ISTATE is determined. If the current value ISTATE is equal to either ISTATELAST or ISTATENEW, ISTATENEW indicates the state of the new system. If the current value ISTATE is equal to neither ISTATELAST nor ISTATENEW, the current value ISTATE indicates the state of the new system.

IV. PROGRAM MODIFICATIONS

Fig.2 indicates a general flow diagram of modified program. The discontinuity detection subroutine DET and the new state flag subroutine SETF simulate the method proposed in the previous section.

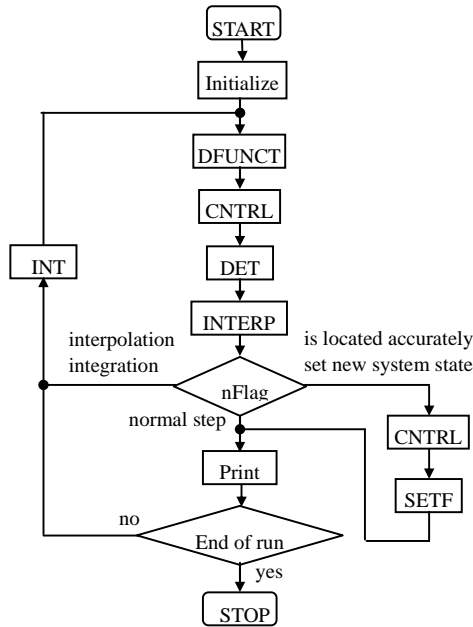


Figure2. General flow diagram for Modified program

List of program variables associated with DET and SETF:

- IFIRST is set to 0 to indicate first entry to DET.
- ISTATE system state marker, its value determines the current system equations integrated.
- ISTATELAST value of ISTATE at the start of the last integration step.
- ISTATENEW value of ISTATE at the restart of the integration when a discontinuity is located accurately.
- INDF is set by DET to:

=0 no discontinuity detected, the current T is at the normal step end.

=1 no discontinuity detected, the current T is at the interpolation step end, it forces INTERP to implement the forward interpolation and the inverse interpolation is by-passed.

=2 an interpolation step is required to locate a discontinuity.

NORM is set to 0 by INTERP to indicate the current T is at the normal step end.

A. DET: discontinuity detection subroutine

```

void DET(IFIRST, INDF, ISTATE, ISTATELAST, ISTATENEW)
{
  if (IFIRST==0)
  {
    ISTATELAST=ISTATE; ISTATENEW=ISTATE;
  }
  if (ISTATE==ISTATELAST)
  {
    INDF=0;
    if(NORM!=0) INDF=1;
  }
  else
  {
    INDF=2;
    ISTATENEW=ISTATE;
    ISTATE=ISTATELAST;
  }
}
  
```

B. SETF :set new state flag subroutine

```

void SETF(ISTATE, ISTATELAST, ISTATENEW)
{
  if( ISTATE==ISTATELAST||ISTATE==ISTATENEW)
  {
    ISTATE=ISTATENEW;
    ISTATELAST=ISTATE;
  }
}
  
```

To implement the present method, the small number of modifications to the interpolation subroutine INTERP of locating a discontinuity and the main program loop are required. The integration subroutine INT employs the Gear algorithm for stiff equation sets and the Runge-Kutta-Merson Algorithm. The subroutine DFUNCT which define discontinuity functions and the subroutine CNTRL which sets the system state marker are provided by user.

V. APPLICATIONS

The modified program with Runge-Kutta-Merson and Gear integration subroutines has been successfully applied to the simulation of some mechanical, electrical, control and hydraulic dynamic systems for which the original program is inefficient. The following example is one of the applications.

e.g. Hydraulic system

To simulate opening characteristics of the hydraulic two-stage relief valve, it is recognized that the equation system of the valve has four states which depend on whether the piloted valve, the main valve or both are open and the state conditions contain the following four discontinuity functions (DFUNCT):

$$\begin{aligned}
 \phi_1 &= y_1 \\
 \phi_2 &= y_2 \\
 \phi_3 &= A_1 p_1 - A_2 p_2 - F_{O1} \\
 \phi_4 &= A_d p_2 - F_{O2}
 \end{aligned}$$

y_1 is the opening amount of the main valve port. y_2 is the opening amount of the piloted valve port. p_1 is the pressure of the main valve port. p_2 is the pressure of the piloted valve port. ϕ_3 is the force of acting on the main valve. ϕ_4 is the force of acting on the piloted valve.

The four states are (CNTRL) :

State 1 :Both the main valve and the piloted valve are not open.

$$\text{Condition } (\phi_1=0 \wedge \phi_3 \leq 0) \wedge (\phi_2=0 \wedge \phi_4 \leq 0)$$

State 2: The piloted valve only is open .

$$\text{Condition } (\phi_2 > 0 \vee (\phi_2 = 0 \wedge \phi_4 > 0)) \wedge (\phi_1 = 0 \wedge \phi_3 \leq 0)$$

State 3: The main valve only is open

$$\text{Condition } ((\phi_1 > 0 \vee (\phi_1 = 0 \wedge \phi_3 > 0)) \wedge (\phi_2 = 0 \wedge \phi_4 \leq 0))$$

State 4 :Both the main valve and the piloted valve are open

$$\text{Condition } ((\phi_1 > 0 \vee (\phi_1 = 0 \wedge \phi_3 > 0)) \wedge (\phi_2 > 0 \vee (\phi_2 = 0 \wedge \phi_4 > 0)))$$

The simulated response of the relief valve is shown in Fig.3. This result is done with a Local truncation error tolerance of 10^{-4} .

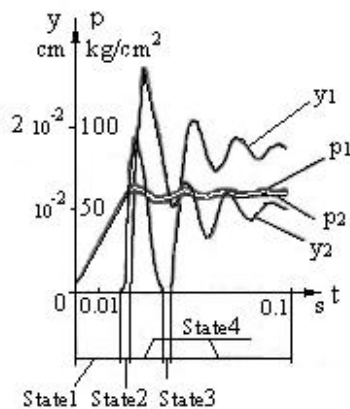


Figure3. Simulation of the relief valve

VI. CONCLUSIONS

While the sign rule of detecting discontinuities is usually reliable, it causes problem when discontinuity conditions are complicated. The present method which predicts a discontinuity in terms of a change of the state marker value might perform better. Required modifications to the original package are done without difficulty. Based on the same mathematical form, this method may be directly adapted for many integration packages for system with discontinuities.

REFERENCES

[1] G.L. Xiong, Digital Simulation Algorithms and Software(in Chinese). Beijing: Astronautic Publishing House, 1991

[2] D.G. Liu, and J.G Fei, Digital Simulation Algorithms for Dynamic Systems(in Chinese). Beijing: Science Press, 2000
 [3] T.Y. Xiao, W.H. Fan, Introduction to System Simulation, 2rd ed.(in Chinese). Beijing: Tsinghua University Press, 2010
 [4] D.Ellison, "Efficient Automatic Integration of Ordinary Differential Equations with Discontinuities", in Math. Comput. Simulation, vol. XXIII, March 1981, pp. 12–20
 [5] J.L.Hay, R.E.Crosbie, and R.I.Chaplin., "Integration Routines for System with Discontinuities Simulation",in Comput. J, vol.IVII, 1974, pp. 275-278
 [6] B. Das, S. Steinberg, D.J. Zhang, and T. Robey , "Comparisons of Numerical Solution Methods for Differential Equations with Discontinuous Coefficients", in Math. Comput. Simulation, vol. III VI, April 1994, pp. 57-75