Computers and Chemical Engineering 62 (2014) 139–160

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

# From discretization to regularization of composite discontinuous functions

Tareg M. Alsoudani*, I.D.L. Bogle

*Centre for Process Systems Engineering, University College London, Department of Chemical Engineering, Torrington Place, London, WC1E 7JE, U.K.*

ARTICLE INFO

*Article history:*
Received 1 October 2012
Received in revised form
24 November 2013
Accepted 30 November 2013
Available online 18 December 2013

*Keywords:*
Discontinuity detection
Discontinuity resolution
Solving differential equations.

ABSTRACT

Discontinuities between distinct regions, described by different equation sets, cause difficulties for PDE/ODE solvers. We present a new algorithm that eliminates integrator discontinuities through regularizing discontinuities. First, the algorithm determines the optimum switch point between two functions spanning adjacent or overlapping domains. The optimum switch point is determined by searching for a “jump point” that minimizes a discontinuity between adjacent/overlapping functions. Then, discontinuity is resolved using an interpolating polynomial that joins the two discontinuous functions.

This approach eliminates the need for conventional integrators to either discretize and then link discontinuities through generating interpolating polynomials based on state variables or to reinitialize state variables when discontinuities are detected in an ODE/DAE system. In contrast to conventional approaches that handle discontinuities at the state variable level only, the new approach tackles discontinuity at both state variable and the constitutive equations level. Thus, this approach eliminates errors associated with interpolating polynomials generated at a state variable level for discontinuities occurring in the constitutive equations.

Computer memory space requirements for this approach exponentially increase with the dimension of the discontinuous function hence there will be limitations for functions with relatively high dimensions. Memory availability continues to increase with price decreasing so this is not expected to be a major limitation.



## 1. Introduction

A process can be thought of as a complex system that is described by, mostly, continuous mathematical functions (algebraic or differential). Solution of these differential equations, usually through integration, brings an insight into the behaviour of the process under study. However, the continuity of these mathematical functions is sometimes broken by internal or external influences. Breakage of continuity occurs because of the tendency of scientists to treat each process condition with differing constitutive equations and/or boundary conditions. Once simulation shifts from one condition to another, the underlying equations change, usually with no reservation to mathematical continuity. A rapid phase shift or flow reversal represents an example of an internally generated discontinuity in ODE/DEA system whereas switching a pump on/off can be considered as an external influence that raises a mathematical discontinuity in the modelled system.

Handling discontinuity through ODE/DAE solvers is performed through two steps: discontinuity detection and discontinuity resolution; although some solvers combine the two steps (Mao & Petzold, 2002).

The literature refers to the problem of locating a discontinuity as discontinuity detection (Javey, 1988). Process simulators usually couple their integrators with the modelling language. This coupling eases detection of jump discontinuities.

Regardless of the form or source of discontinuity, it needs to be resolved either before starting to integrate the ODE/DAE system (if possible) or whenever it is encountered during the evolution of integration process. Methods for the resolution of discontinuities arising during integration of differential equations can be divided into two types:

1. Type I tries to handle discontinuities using methods that are usually integrated with the solver (integrator) of the ODE/DAE system. Those methods are usually generic, irrespective of the system to be modelled and handle discontinuities at the time they are encountered during integration (or simulation). Most literature on discontinuity detection and resolution covers this

* Corresponding author. Tel.: +966 50 634 7994; fax: +44 0 20 7383 2348.
*E-mail addresses:* t.alsoudani@live.ucl.ac.uk (T.M. Alsoudani), d.bogle@ucl.ac.uk (I.D.L. Bogle).

0098-1354/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compchemeng.2013.11.019

**Nomenclature**

| | |
|---|---|
| $a_p$ | specific area of the pellet |
| $C_T$ | dimensionless total concentration |
| $C_t^{ref}$ | total molar concentration |
| $D_z$ | axial thermal conductivity |
| $k_{gl}$ | overall mass transfer coefficient |
| $L$ | column length |
| $N_g^m$ | number of fluid film mass-transfer units = $\frac{1-\varepsilon}{\varepsilon}\frac{a_p k_{gl} L}{u_{ref}}$ |
| $Pe_m$ | mass Peclet number = $\frac{u_{ref} L}{D_z}$ |
| $\rho_s$ | solid density |
| $q_i^{ref}$ | maximum adsorbence of adsorbate $i$ in adsorbent pellet |
| $Q_i$ | dimensionless adsorbence of adsorbate $i$ in adsorbent pellet |
| $u_{ref}$ | reference velocity |
| $U$ | dimensionless velocity |
| $\vec{v}_i$ | vector dimension at time instant $i$ of simulation run |
| $x$ | dimensionless axial distance or $x$-dimension |
| $y$ | dimensionless concentration (mole fraction) or $y$-dimension |
| $\langle y \rangle$ | adsorbate dimensionless concentration (mole fraction) in solid phase |

*Greek Letters*

| | |
|---|---|
| $\varepsilon$ | void fraction |
| $\zeta_{m_i}$ | mass capacity factor = $\frac{1-\varepsilon}{\varepsilon}\frac{\rho_s q_i^{ref}}{C_t^{ref}}$ |
| $\tau$ | dimensionless time |

*Sub/superscripts:*

| | |
|---|---|
| $f$ | feed |
| $p$ | purge |
| $i$ | component index or simulation time instant |
| $m$ | mass |
| $s$ | solid |

class (eg. Ellison, 1981; Javey, 1988; Mao & Petzold, 2002; Park & Barton, 1996).

2. Type II handles discontinuities using knowledge about the process to be modelled. It remodels the ODE/DAE system in a way that eliminates discontinuities. Literature is very sparse in this area (e.g. Borst, 2008; Brackbill, Kothe, & Zemach, 1992; Carver, 1978; Helenbrook, Martnelli, & Law, 1999).

Borst (2008) refers to the two types as discretization and regularization, respectively (Fig. 1). He also points out that internal model discontinuities are better handled using type II methods

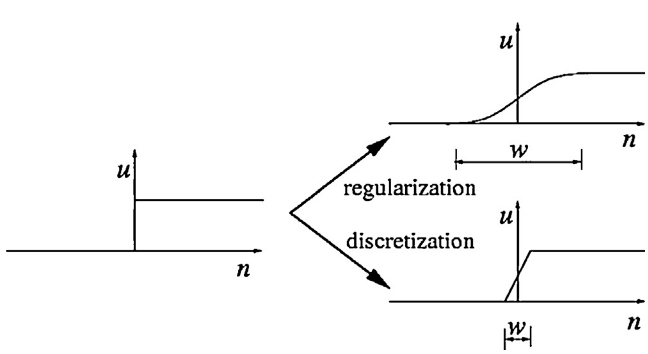


**Fig. 1.** Transformation of a discontinuity into a regularization or a discretization problem. (Borst, 2008).

irrespective of the solver integration routine. Surprisingly, both classes use some form of an interpolation to convert a discontinuous region into a continuous one when dealing with internally generated discontinuities. Externally generated discontinuities are usually handled by re-initialization of the model equations and their respective new initial and boundary conditions. In the foregoing discussion, we will briefly touch on recent literature covering each of the categories.

### *1.1. Type I—Integrator based discontinuity resolution*

Cellier (1979) demonstrated that the most efficient approach to locating a state event is through *discontinuity locking*. In discontinuity locking the system of ODE/DAE is locked until the end of the integration step regardless of the existence of a state event during the step. After completion of the integration step that involves a state event, the exact location of the state event is detected. Several event location algorithms that use discontinuity locking mechanism are reported and for a comprehensive review of state event detection algorithms the reader may refer to Park and Barton (Park & Barton, 1996). Mao and Petzold (2002) have introduced an event detection algorithm that is based on regulating the integration step size based on discontinuity functions that are appended to the DAE system. Recently, Archibald, Gelb, and Yoon (2008) introduced a state event detection algorithm that is based on polynomial annihilation techniques. Their method relies on the difference of the Taylor series expansions behaviour between continuous and non-continuous intervals of the tested function.

Once a discontinuity is detected, it needs to be resolved before the integrator passes it. Javey (1988) reports three methods for resolving discontinuities. In all methods, the integrator checks the sign change of a discontinuity-function after each integration step as indication of having located a discontinuity:

1. Once the discontinuity is located, the integrator switches modelling equations to those after the discontinuity and starts at the end of the current step. This procedure is inaccurate as it accumulates error each time a discontinuity is encountered. Mao and Petzold (2002) warn about mere stepping over discontinuities without carefully handling them with some rigour.
2. Once the discontinuity is located, the integrator halves the step and repeats the last integration step in a hope to resolve the discontinuity. Resolution is generally achieved if the function is continuous but the integrator may fail to resolve the discontinuity due to the use of a large integration step. Thus, repeating the integration step with smaller step sizes, where the discontinuity is detected should eventually reveal the continuity of the function. This solution, although better than the first one, is still considered inefficient because the integrator needs to iterate at the discontinuity until an acceptable error tolerance is achieved. If the acceptable error tolerance is not achieved after repeated step-halving (usually because of an instantaneous discontinuity), the integrator aborts integration. The method is then unable to resolve the discontinuity (Carver, 1978).
3. Once the discontinuity is located, the integrator reinitializes the differentiable variables using post discontinuity conditions after interpolating differential and algebraic variable at the discontinuity using a *discontinuity function* (an interpolating polynomial). It should be noted that this method implies mathematical continuity of differential equations through the discontinuity domain regardless of the validity of the resulting solution, as demonstrated by Cellier (1979). This method is the most commonly adopted in recent integration routines used for process simulation. The mismatch between the results obtained using the interpolating polynomial and those obtained when reinitializing the ODE/DAE system after crossing a