

Constructive and Non-constructive Asynchronous Qualitative Simulation

George M. Coghill
Centre for Intelligent Systems,
Department of Computer Science,
University of Wales,
Aberystwyth SY23 3DB.
gmc@aber.ac.uk.

Michael J. Chantler
Intelligent Systems Laboratory,
Department of Computing and Electrical Engineering,
Heriot-Watt University,
Edinburgh EH14 4AS.
mjc@cee.hw.ac.uk

Abstract

From its inception it was known that qualitative reasoning had a problem with the generation of spurious behaviours, and a great deal of effort has been expended in tackling this problem. Two approaches have been taken to constraint based qualitative simulation: *constructive* and *non-constructive*, though a thorough and objective examination of the operation and results of performing qualitative simulation by means of these two approaches has not been carried out. The context of the work presented here is fuzzy qualitative reasoning. The two systems FuSim and *Mycroft* are used to compare non-constructive, semi-constructive and constructive approaches; and the results of this are described and analysed. A number of conclusions are drawn, the most important of which is confirmation of the idea of "horses for courses". It is plain that a causally ordered model reveals the interactions of the relations in a straightforward manner and constructive qualitative analysis gives an efficient initialisation process, which may be useful for diagnosis or explanation. On the other hand, non-constructive approaches permit the simulation of systems which contain algebraic loops.

Introduction

One of the original motivations for developing qualitative simulation systems was to provide a means of permitting an expert system to reason from first principles. Indeed, as research progressed it became apparent that the ability to reason about the behaviour of physical systems without having to resort to numerical approaches had a great deal of potential for application in industrial contexts for predicting the behaviour of systems which were incompletely specified, but utilising such knowledge as was available.

Dynamic physical systems are most often represented by differential equations. Because of this the constraint based approach, which is an abstraction of differential equations (called *qualitative differential equations*) became the major focus of interest. And, because it was made freely available, QSIM became the system most commonly utilised.

Even from its inception it was known that qualitative reasoning had a problem with the generation of spurious behaviours, and a great deal of effort has been expended in tackling this problem (Kuipers 1994). However, from the point of view of those researchers with a background in traditional simulation techniques, the *non-constructive* al-

gorithm used in QSIM appeared somewhat wrong-headed. In fact it was suggested that the non-constructive nature of the algorithm was the cause of (at least some of) the spurious behaviours generated by QSIM; and this led to a further strand in qualitative reasoning research (Bousson & Trave-Massuyes 1994, Wiegand 1991). However, no one has thus far presented a thorough and objective examination of the operation and results of performing qualitative simulation constructively and non-constructively. That is the subject of this paper.

The context of the work presented here is fuzzy qualitative reasoning. The two systems FuSim (Shen & Leitch 1993) and *Mycroft* (Coghill 1996) are the systems used for the comparison. FuSim is an excellent representative of the non-constructive approach because it utilises the same algorithms as QSIM but permits the incorporation of semi-quantitative and temporal information in a seamless manner. *Mycroft* is also a fuzzy qualitative reasoning system; in fact it is a framework incorporating a number of synchronous (Scott & Coghill 1998) and asynchronous simulation and envisionment methods (Coghill 1996). However, for the purposes of this paper the focus will be on the semi-constructive and constructive asynchronous algorithms.

In the rest of the introduction the scene will be set by a description of the operation of constructive and non-constructive algorithms. This will be followed, in the following section, by a brief outline of fuzzy qualitative simulation, which will lead into a brief description of the *Mycroft* framework (because the details of its operation are less readily available than those of FuSim). The greater portion of the paper is then taken up with describing and analysing the results of simulating the behaviour of a benchmark example (the coupled tanks system) constructively and non-constructively. Finally, a number of conclusions are drawn in the light of these results which impact on the use and future directions of qualitative simulation research.

Constructive and Non-constructive Qualitative Simulation

The distinction between constructive and non-constructive algorithms was discussed in detail in the context of QR by Wiegand (Wiegand 1991). Prior to the inception of QR as a domain of research simula-

tion algorithms were epitomised by those used in numerical simulation of dynamic systems (Shoup 1979). These algorithms typically distinguished between two divisions of the process: numerical integration and the solving of the system equations.

A fully constructive simulator is defined (Wiegand 1991) as one that uses the system model to generate the values of the system variables directly; both solving the model equations and the integration of dynamic systems. Consider the following equation:

$$\dot{x} = f(x, u)$$

Those variables whose derivatives appear on the left hand side of this equation are the state variables. In the integration phase of numerical simulation the magnitude and the derivatives of the state variables at a particular time (t , say) are used to estimate the magnitudes of the state variables some short time in the future, δt . For example, consider the simplest form of numerical integration, the Euler integration formula:

$$x(t + \delta t) = x(t) + \dot{x}(t) \cdot \delta t$$

This formula is used to generate (or construct) the succeeding value that this particular state variable will have, making this integration formula constructive. After this the equations making up the system model are solved to supply the values of the other endogenous variables of the system. That is, the new values of the state variables and the given information concerning the values of any exogenous variables are used to solve the system equations and construct the values of the endogenous variables. Thus, this is a wholly constructive simulation algorithm (albeit synchronous); both with regard to the integration and equation solving phases of the algorithm. Translating this into the QR domain and utilising the terminology of Williams (Williams 1984) where Integration is named Transition Analysis (TA) and the equation solving phase is termed Qualitative Analysis (QA).

At the other end of the spectrum a non-constructive simulator would use the model equations only to filter out values of the complete set of variables of the system which had already been generated by the TA phase of the simulation. algorithm operating in this manner is QSIM, and the approach has more recently been employed in the Fuzzy Qualitative Simulation engine FuSim (Shen & Leitch 1993).

The QSIM algorithm (Kuipers 1986) epitomises, and in fact defines, the non-constructive approach. In QSIM the TA phase is applied first. Since all the model variables are represented a magnitude/derivative pairs, a set of possible future values may be generated for every variable individually. Only after this has been completed is the system model applied in the QA phase; and then only to filter out impossible assignments of values to variables. Two types of filter are used: a *constraint filter*, which checks the assignments with respect to each individual constraint, and a *pairwise consistency filter* which ensures the consistency of the assignments between the system constraints (and thus for the whole model). Global filters are also applied, but these are generally applicable to both constructive and non-constructive simulators.

The distinction between constructive and non-constructive algorithms is not binary but rather forms a continuum, with some algorithms not being completely one or the other, but the classification allows qualitative simulations and numerical simulation to be viewed by means of the same categories.

Spurious behaviour generation

The major problem of QR is the generation of spurious behaviours. These incorrect behaviours take a variety of forms with a commensurate number of causes. A detailed mathematical analysis of QR, which highlighted a number of these problems, was performed by Struss (Struss 1988). Perhaps the most well documented phenomenon within symbolic qualitative simulators is that of chattering (Kuipers & Chui 1987). This occurs because of insufficient derivative information being contained within the model, and consists of the highest derivatives of the system oscillating aimlessly around a critical point. A great deal of effort has been expended in generating indirect solutions to this problem in the form of additional filters applied to QR algorithms with some, but not total, success (e.g. see (Kuipers 1994)). Efforts in the direction of generating direct solutions to this problem have been the motivation for including *semi-quantitative information* (SQI) into QR algorithms (Berleant & Kuipers 1990). One such system is FuSim (Shen & Leitch 1993). Also, a major motivation for the constructive nature of the design of the *Predictive Algorithm* (PA) was to alleviate this problem. Here, one cause of spurious behaviours was attributed to the fact that QSIM had to use the model to remove states that had been generated independently in the TA phase. In the PA this was thought not to happen because these states are not created in the first place.

However, in this paper we are only concerned with these issues in so far as they relate to the central theme of comparing the results of performing the simulation constructively and non-constructively. That is, we will only address the issue of chattering in so far as it impinges on the motivation for developing constructive simulators.

The Mycroft Framework

The *Mycroft* framework (Coghill 1996) is a constraint-based fuzzy qualitative reasoning system comprising a number of simulation and envisionment algorithms. The development of this framework has permitted the suitability of different techniques to be examined in a number of contexts; and the comparison of different approaches to constraint-based fuzzy qualitative simulation to be made.

This section contains an outline description of asynchronous simulation in *Mycroft* which is based around the common division of the process (depicted in figure 1) into four parts: the model representation, the input data, the output behaviours and the inference engine. The model is a, possibly incomplete, representation of a physical system consisting of a number of variables and the relations between them. The variables of the system take values from a quantity space, which in this case consists of fuzzy values. The input

data is the assignment of values to the exogenous and state variables, and the behaviours output are trees or graphs of states; each state being a consistent assignment of values to all the variables in the model. In each aspect of the *Mycroft* framework the representation of the model structure and the variables remains the same, while the inference engine utilised for the reasoning process changes. In the rest of this section each of these components is described in turn. (The details of the operation of *Mycroft* can be found in (Coghill 1996)).

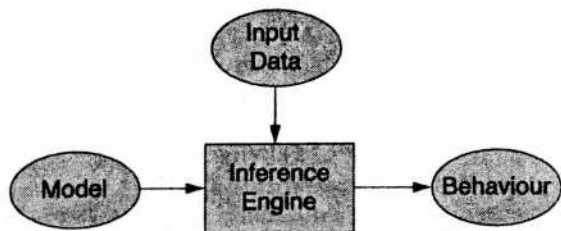


Figure 1: The Qualitative Reasoning Process

Fuzzy Sets and Quantity Spaces

Both qualitative reasoning and approximate reasoning (Zadeh 1965) have a common foundation, that of reasoning about systems that are incompletely specified. However, there is a strong distinction in that whereas in qualitative reasoning a knowledge of the structure of the system under consideration is assumed, in fuzzy systems it is merely an input/output representation that is utilised; albeit with the possibility of being empirically derived. Thus there are three advantages which ensue from the combination of fuzzy and qualitative approaches:

- the fact that the meaning of a qualitative value and its support set (the real number line here) are captured in a single representation,
- the ability to incorporate empirical knowledge into a model (which is also finer grained than the $M^{+/-}$ constraint in QSIM (Kuipers 1986)), and
- being able to include more detailed knowledge of the temporal behaviour of the variables in a model than the total ordering available within QSIM, which is essential for use in such applications as model-based diagnosis and control.

This was the motivation behind the development of FuSim (Shen & Leitch 1993), and its major contribution to Qualitative Reasoning. FuSim is the system which was the major influence on the development of *Mycroft*.

Fuzzy sets extend the ideas of traditional set theory to include the concept of partial (or graded) membership. It is assumed here that the ideas underlying fuzzy sets are known to the reader; however, a description of the domain and explanation of the concepts may be found in (Kosko 1992). In FuSim, for reasons of computational efficiency, trapezoidal fuzzy numbers and intervals are used. These are represented by the four tuple (a, b, α, β) which yields the interval shown in figure 2. The trapezoidal fuzzy numbers maintain

the essential features of fuzzy sets: graded membership and the use of α -cuts.

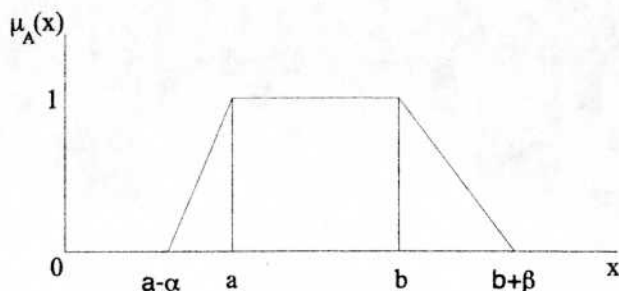


Figure 2: A Trapezoidal Fuzzy Number

The quantity space which is built from fuzzy numbers must be closed, continuous, finite and cover all values which a variable can take. An example of such a quantity space is shown in figure 3. In fuzzy qualitative simulation, unlike QSIM, the quantity space for the derivatives of a variables may also be dense (that is can have any number of divisions consistent with the definition of a quantity space).

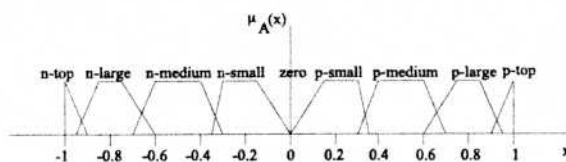


Figure 3: A Fuzzy Quantity Space

Model Representation

The *Mycroft* framework is a qualitative reasoning system within the so called constraint based ontology. The models used in the framework consist of sets of variables and the constraints that relate them. In fuzzy qualitative reasoning the operators utilised are the same as for its symbolic counterpart, though by the nature of the case there is a difference in the way they are implemented. The fuzzy case makes use of two concepts: the *extension* and *approximation* principles. The former of these extends the results of numerical and interval arithmetic to the fuzzy domain. It allows one to define a set of algebraic operations that may be applied to fuzzy sets and specifies the form that the fuzzy result will take. All the variables of the system take their values from a predefined fuzzy quantity space. In performing any operation on one or more fuzzy values it is unlikely that the resulting fuzzy number (the *propagated value*) will also be a member of the constrained variable's quantity space. Therefore, the latter principle provides a means of mapping the calculated value onto the appropriate values in the variable's quantity space (the *predicted value(s)*). There are two ways in which this can be done: in the simplest case, if there is any overlap between a member of the quantity space and the calculated fuzzy number, then that quantity is a valid approximation and may be assigned as a valid value for the variable. This is depicted in Figure 4. However, if there is more than one

possible approximation (as is usually the case) then one can use a distance metric to gain a measure of the degree of approximation of each qualitative value. In this way one can prioritise the values from "most likely" to "least likely" (Leitch & Shen 1993).

It is convenient to point out here that the distinction between the QA phase in constructive and non-constructive fuzzy qualitative simulation is that: in the non-constructive case the *propagated value* is used to eliminate possible predicted values if they do not overlap with said propagated value, whereas in the constructive case the propagated value is used to generate the predicted values by being mapped onto the quantity space of the variable under consideration.

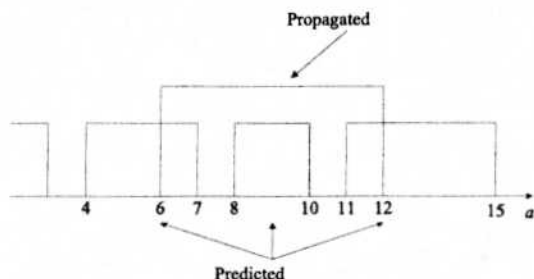


Figure 4: The Approximation Principle

In *Mycroft* the model constraints are *causally ordered* (Iwasaki 1988) and distributed over a number of differential planes as in the PA (Wiegand 1991). That is, the qualitative differential equation (*qde*) model is developed on plane-0, and the relationships between the higher derivatives of the system are obtained by differentiating the *qde* and representing the results as a *qde* on the so called higher differential planes. To illustrate, consider a single tank system. The quantitative equations describing this system are:

plane-0

$$\begin{aligned} q_o &= kV \\ V' &= q_i - q_o \end{aligned}$$

plane-1

$$\begin{aligned} q'_o &= kV' \\ V'' &= q'_i - q'_o \end{aligned}$$

where q_i is the flow of fluid into the tank, q_o is the corresponding flow of fluid leaving the tank, V is volume of fluid in the tank at any particular time, and k is a parameter representing the resistance to flow presented by the outlet pipe.

This system model is linear and it can be seen that the relations in plane-1 have the same form as those in plane-0; with the difference being that each variable is the next derivative of the variable in plane-0. The means by which models represented in *Mycroft* can be used to generate the system behaviours is described in the following section.

In *Mycroft* as in QSIM and FuSim variables are represented as vectors (Morgan 1988). However, whereas in those systems they are fixed two element vectors (consisting of the magnitude and first derivative) in *Mycroft* they can be of any length, including one. For example, a vector of length n will consist of the mag-

nitude and first $n - 1$ derivatives. Each member of the vector is called a *variable-vector element*.

Inference Engines

As stated the *Mycroft* framework contains a number of reasoning algorithms, all but one of which operate asynchronously. The distinction between synchronous and asynchronous simulation lies in the fact that in a synchronous simulator the TA phase (integration step) is driven by an external clock (Scott & Coghill 1998); whereas in the asynchronous case the simulation proceeds on the basis of the derivative information contained within the model. Synchronous simulation is useful in situations where the task involves tracking the behaviour of some physical plant, and asynchronous techniques are useful, in particular, for explaining the behaviour of systems.

In the *Mycroft* framework there are two asynchronous simulation algorithms: *semi-constructive* and *constructive*, each of which makes the simulation process more constructive than the non-constructive algorithm in QSIM and FuSim. These two algorithms are distinguished by the degree to which they make the simulation constructive.

In the semi-constructive approach the transition rules are applied only to the *magnitudes* of the state variables in order to *generate* successor values for those variable-vector elements. These values are then passed through the model constraints which, since they are causally ordered, can be used to *generate* the values for the other system variables directly (and their derivatives thanks to the differential plane structure of the model). Due to the ambiguity of the approximation principle it is possible that a calculated value for a variable-vector element may be discontinuous with its previous value; therefore, to ensure continuity the transition rules are applied to *filter* out any discontinuous assignments. In this way a behaviour tree is generated which it was hoped would contain fewer spurious behaviours than the non-constructive approach.

Constructive simulation operates in the same manner as semi-constructive simulation, with the exception that the application of the transition rules as generators is not restricted to the magnitudes of the state variables. Rather, they are applied to that derivative of a state variable which is estimated as most likely to transit first (on the basis of the available temporal information). The motive for this approach is to be able to select the "best" real behaviours of the system whilst eliminating spurious ones.

It may be noted from the foregoing summary description that the transition rules in *Mycroft* are applied as both filters and generators; whereas in non-constructive approaches they are applied exclusively as generators. Full details of the non-constructive algorithm FuSim may be found in (Shen & Leitch 1993) and of the asynchronous algorithms in *Mycroft* in (Coghill 1996). However, in order to better understand the operation of the *Mycroft* the transition rules are reproduced here.

Generation Rules (G1 and G2):

For a vector of length n , where the j th derivative, $d^j_{t_k}$, in the time interval t_k , has a value from its quan-

tity space, q_i , (i.e. $d_{t_k}^j := q_i$) the generation rules whereby the values which this derivative can take in the time interval t_{k+1} are generated, are as follows:
(G1) If there exists a derivative, $d_{t_k}^m$, such that $d_{t_k}^m$ is the first non-zero derivative which is a higher derivative than $d_{t_k}^j$, then

If $d_{t_k}^m >_\alpha 0$ ($d_{t_k}^m <_\alpha 0$)

If $q_i \in \mathbb{R}$, then

$$d_{t_{k+1}}^j = q_{i+1} \quad (d_{t_{k+1}}^j = q_{i-1})$$

else

$$d_{t_{k+1}}^j = [q_i \vee q_{i+1}] \quad (d_{t_{k+1}}^j = [q_i \vee q_{i-1}])$$

(G2) If all derivatives higher than $d_{t_k}^j$ are zero, or if $j = n$, then

$$d_{t_{k+1}}^j = [q_{i-1} \vee q_i \vee q_{i+1}]$$

In the above rules it is assumed that quantities are ordered from lowest to highest in the quantity space. Rule 2 reflects the fact that, although all the known higher derivatives are zero, the $(n+1)$ th derivative is unknown and so one has to assume that it could affect the j th derivative in any direction or none.

Filter Rules (F1 and F2):

The rules given above were for generating future values. If however, we wish to filter out values which have been generated by a constraint of the system, but which may not be continuous with the present value for the variable under consideration, then we may utilise the following filtering rules.

Consider a vector of length n , where the j th derivative, $d_{t_k}^j$, at time, t_k , has a value from its quantity space, q_i . If a value for the j th derivative in the succeeding interval, $d_{t_{k+1}}^j$, has been generated with a value from its quantity space of q_l , then the filter rules for ascertaining whether q_l is a valid value for $d_{t_{k+1}}^j$ to take are as follows:

(F1) If $l > i + 1$ ($l < i - 1$), then $\neg ok$ (the adjacency filter)

(F2) If there exists a derivative, $d_{t_k}^m$, such that $d_{t_k}^m$ is the first non-zero derivative which is a higher derivative than $d_{t_k}^j$, and $m < n$, then

If $d_{t_k}^m >_\alpha 0$ ($d_{t_k}^m <_\alpha 0$) then

If $q_i \in \mathbb{R}$, then

If $l \leq i$ ($l \geq i$) then $\neg ok$

else

If $l < i$ ($l > i$) then $\neg ok$

Experimental Comparisons

The experiments associated with the semi-constructive algorithm are designed to permit a direct comparison with the results of FuSim. This enables an assessment of the *pros* and *cons* of constructive and non-constructive approaches to qualitative simulation to be presented. The constructive simulation experiments aim to assess whether, and to what extent, it is possible to make a simulation fully constructive.

Experimental Testbed

The purpose of the experiments described here is to permit the direct comparison of different approaches

to and different aspects of qualitative simulation. To this end, the results presented in this section relate to the same benchmark example: the coupled tanks system. The simulators under discussion have been used to simulate the behaviour of a number of benchmark systems, but this particular system is of sufficient complexity to test the simulators thoroughly, whilst at the same time still being simple enough to be understood and analysed.

A schematic diagram of the coupled tanks system is shown in Figure 5

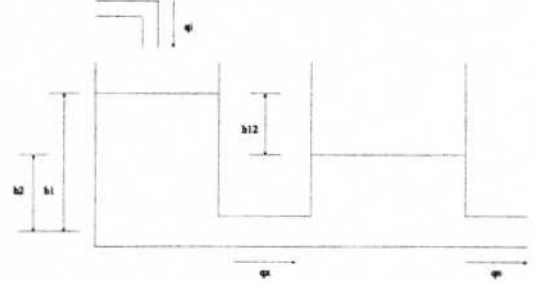


Figure 5: A Coupled Tanks System

The analytic equations describing a linear time-invariant model of this system are as follows:

$$\begin{aligned} q_x &= \frac{\rho g}{R_x} \cdot (h_1 - h_2) \\ q_o &= \frac{\rho g}{R_o} \cdot (h_2) \\ h_1' &= \frac{(q_i - q_x)}{A_1} \\ h_2' &= \frac{(q_x - q_o)}{A_2} \end{aligned} \quad (1)$$

where the symbols in the model are as given in the list of symbols. Note also that the form of the relations between the flows and the volumes of fluid in the tanks is linear to ease the explanation and comparisons.

In the experiments described in this section, all the variable-vector elements have been assigned the same quantity space. This is a normalised space consisting of nine values as shown below.

$Q_F =$	n-max	(nt)	[-1, -1, 0, 0.1]
	n-large	(nl)	[-0.9, -0.75, 0.05, 0.15]
	n-medium	(nm)	[-0.6, -0.4, 0.1, 0.1]
	n-small	(ns)	[-0.25, -0.15, 0.1, 0.15]
	zero	(z)	[0, 0, 0, 0]
	p-small	(ps)	[0.15, 0.25, 0.15, 0.1]
	p-medium	(pm)	[0.4, 0.6, 0.1, 0.1]
	p-large	(pl)	[0.75, 0.9, 0.15, 0.05]
	p-max	(pt)	[1, 1, 0.1, 0]

This quantity space was chosen for convenience in comparing the results of a *Mycroft* simulation with the same simulation performed using FuSim. It is the quantity space used in the description of FuSim given in (Shen & Leitch 1993). The terms in brackets after the name of the quantity are short names which will be used in the tables and diagrams of this section.

Semi-constructive Simulation versus FuSim

The primary purpose of the experiments described in this section is to allow the comparison of the results produced by a semi-constructive simulator with those of a non-constructive simulator. However, in the course of describing the results produced, there will be a discussion of why the results are as they are. The behaviours generated by the two simulators are discussed with respect to the two phases of the simulation: qualitative analysis (QA) and transition analysis (TA). To make the discussion easier to follow, the argument will initially focus on the QA phase as it is utilised in the initialisation part of the simulation. Once the issues involved there have been clarified the discussion will proceed to the TA as it is used to generate the possible future states of the system.

Experimental Results

Initialisation Consider the coupled tanks system described in the previous section. We will assume that the cross sectional area of both tanks is equal to 1 because this is the setup which is reported by Shen and Leitch (Shen & Leitch 1993). Also, because FuSim requires that all variables be represented as magnitude/derivative pairs, the *Mycroft* model will contain two differential planes.

The system is a single input, single output, second order system. The values for the input (which is steady) and state variable magnitudes for the simulation in this experiment are as follows:

Input: $q_i = < p - \text{small zero} >$

States: $\begin{cases} h_1 = p - \text{small} \\ h_2 = \text{zero} \end{cases}$

The simulation commences with these values. Because of qualitative ambiguities the output from the initialisation phase is a list of 13 possible initial states as shown in Table 1. (where the quantities for each variable-vector are represented by the abbreviations listed in Section). In this table the entries for each variable are listed as vectors with the first element being the magnitude of the variable. It can be seen that the state variables, h_1 and h_2 , are one element longer than the other endogenous variables, having three elements rather than two. This arises because in the model given by the equations in (1), the state variables are the only variables for which the magnitude and derivative explicitly appear in the model. Therefore, for a model with n differential planes, the variable-vectors for the state variables will be of length $n + 1$, whereas the variable-vectors for the other variables will be of length n .

The results of performing the same initialisation process with FuSim are shown in Table 2. In this table the state variables are $h1$ and $h2$, and the derivatives of the state variables are labelled $h1dash$ and $h2dash$. Since FuSim represents variables as magnitude derivative pairs, the single vectors generated by *Mycroft* for the state variables and shown in Table 1 are spread across two columns in Table 2. Also, to allow direct comparison between the two tables the equivalent state numbers generated by *Mycroft* are shown in brackets

State	h1	h2	h12	qx	qo
1	ps ps ns	z ps ps	ps ps	ps ps	z ps
2	ps z ps	z ps ns	ps ns	ps ns	z ps
3	ps ns ps	z ps ns	ps ns	ps ns	z ps
4	ps ps z	z ps ns	ps z	ps z	z ps
5	ps ps ps	z ps ns	ps ns	ps ns	z ps
6	ps ns pm	z ps nm	ps nm	ps nm	z ps
7	ps ps ns	z ps z	ps ps	ps ps	z ps
8	ps ps ns	z ps ns	ps ps	ps ps	z ps
9	ps z ps	z ps nm	ps ns	ps ns	z ps
10	ps ns ps	z ps nm	ps ns	ps ns	z ps
11	ps ps ps	z ps nm	ps ns	ps ns	z ps
12	ps ns pm	z ps nl	ps nm	ps nm	z ps
13	ps ns pm	z ps nt	ps nm	ps nm	z ps

Table 1: Initialisation of Coupled Tanks in *Mycroft*

in column 1 of Table 2. It can be seen from Table 1 and Table 2 that the results of performing the same initialisation process with both *Mycroft* and FuSim gives the same set of initial states.

Having ascertained, experimentally, that the results obtained for the initialisation phase of a simulation are identical regardless of whether the process is carried out constructively or non-constructively, the task is to discuss whether this ought to be the case or whether these results are merely fortuitous.

Consider a single constraint. To allow simple and direct comparison of the two methods, one must assume that the constraining variables are known. As an example consider a constraint such as:

$$a = b + c$$

This situation is as shown in Figure 4 and it is obvious that the condition of the propagated and predicted values overlapping is dyadic. That is, one will get exactly the same set regardless of whether one maps the propagated value directly onto the quantity space or eliminates those values from the quantity space which do not overlap with the propagated value. However, even at this stage one advantage of the constructive approach is becoming apparent: namely that it is more efficient for initialisation. This is because in the non-constructive version all of the predicted values are tested against the propagated value, whereas in the constructive version only those members of the quantity space which bound the propagated value need be found. This problem is exacerbated as the size of the model increases and more unknown variables appear in it.

As a rule models do not consist of just one constraint, but rather are made up of the conjunction of a number of constraints. Therefore, the next stage is to show that it follows that constructive and non-constructive initialisations will give the same results regardless of the number of constraints in the model.

In a non-constructive simulator such as FuSim the mechanism whereby the QA is carried out is a Waltz filter (Kuipers 1986). In this there are two separate operations: constraint filtering and pairwise filtering. The former of these operates by checking all the predicted values of each variable in a constraint against the predicted values of all the other variables in that

State	h1	h1dash	h2	h2dash	h12	qx	qo
1 (13)	ps ns	ns pm	z ps	ps nt	ps nm	ps nm	z ps
2 (12)	ps ns	ns pm	z ps	ps nl	ps nm	ps nm	z ps
3 (6)	ps ns	ns pm	z ps	ps nm	ps nm	ps nm	z ps
4 (10)	ps ns	ns ps	z ps	ps nm	ps ns	ps ns	z ps
5 (3)	ps ns	ns ps	z ps	ps ns	ps ns	ps ns	z ps
6 (9)	ps z	z ps	z ps	ps nm	ps ns	ps ns	z ps
7 (2)	ps z	z ps	z ps	ps ns	ps ns	ps ns	z ps
8 (11)	ps ps	ps ps	z ps	ps nm	ps ns	ps ns	z ps
9 (5)	ps ps	ps ps	z ps	ps ns	ps ns	ps ns	z ps
10 (4)	ps ps	ps z	z ps	ps ns	ps z	ps z	z ps
11 (8)	ps ps	ps ns	z ps	ps ns	ps ps	ps ps	z ps
12 (7)	ps ps	ps ns	z ps	ps z	ps ps	ps ps	z ps
13 (1)	ps ps	ps ns	z ps	ps ps	ps ps	ps ps	z ps

Table 2: Initialisation of Coupled Tanks in FuSim

constraint, while the latter ensures that variables appearing in more than one constraint have the same value in each of the constraints in which they appear, thus ensuring that the model remains a conjunction of constraints.

In the case of a constructive simulator, such as *Mycroft*, the QA operates as outlined in section . That the two approaches will produce the same results may be seen by showing that the two filters in the non-constructive approach have their counterparts in the constructive version. This can be shown by considering the constraints of the simple model shown in Figure 6.

In Figure 6 the constraints on the left are ordered, and those on the right are not. It can be seen that the set of unordered constraints have been constructed by simply re-arranging the ordered constraints; so it is obvious that the same model is being represented (since the model is a conjunction of constraints and logically $A \wedge B \equiv B \wedge A$).

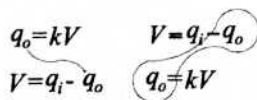


Figure 6: Ordered and un-ordered constraints

Consider now the pairwise filter. Its purpose is to ensure that the variables have the same values in every constraint in which they appear. This is represented in the figure by the variables inside the loop. In the constructive case the values of the variables on the left hand side of an expression are calculated from those on the right hand side. Because the constraints are causally ordered this can only happen once. These calculated values are then passed directly to any constraints which require them as constraining variables. This is shown in Figure 6 by the curved arrow. Thus the variable will have the same values in each constraint in which it appears. Hence the pairwise filters perform the same function in a non-constructive simulator that the ordering of the constraints performs in a constructive simulator.

The other filter is the constraint filter. It operates on

the individual constraints and has already been shown to give the same results for a single constraint as the constructive calculation of the constrained variable.

Thus, since the model is a conjunction of constraints, and the constraint filter and the calculation of a value give the same results, and the pairwise filter and causal ordering perform equivalent functions, we can say that it is right that one should expect the constructive and non-constructive simulators to give the same results for the initialisation phase of a simulation. The arguments given above also demonstrate that both the constructive and non-constructive initialisation phases are sound, (in QSIM terminology) since no possible predicted value will be missed by either algorithm (Kuipers 1986, Shen & Leitch 1993).

Step Ahead Simulation Having obtained a set of initial states one starting state is selected (based on priority or user choice) for the rest of the simulation. There are two options: to proceed to equilibrium or to only simulate one step ahead. The step ahead option is utilised in diagnosis where, before simulating further the results of the present step must be compared against the behaviour of the plant being diagnosed (Leitch *et al*). In the first instance the results of a single step ahead calculation will be shown, again for the purpose of comparison with FuSim. The state from which the simulation proceeds is state 1 in the initialisation. Table 3 shows the results of the step ahead simulation for the semi-constructive simulation in *Mycroft*, and Table 4 shows the results of the same process for FuSim. The form of the information contained in these tables is as described for Table 1 and Table 2. Once again examination of the entries in the tables shows that the results are identical. The comparison between the semi-constructive simulator and FuSim has been made for a number of systems; in each case the results for both simulators have been the same.

We have already discussed the reasons for the initialisation process yielding the same results for the constructive and non-constructive algorithms. The question of why this should be so for the step ahead simu-

State	h1	h1dash	h2	h2dash	h12	qx	qo
14 (22)	pm ps	ps ns	ps ps	ps z	ps ps	ps ps	ps ps
15 (19)	pm ps	ps ns	ps ps	ps ps	ps ps	ps ps	ps ps
16 (17)	ps ps	ps ns	ps ps	ps z	ps ps	ps ps	ps ps
17 (14)	ps ps	ps ns	ps ps	ps ps	ps ps	ps ps	ps ps

Table 4: Output of a step ahead simulation using FuSim

State	h1	h2	h12	qx	qo
14	ps ps ns	ps ps ps	ps ps	ps ps	ps ps
17	ps ps ns	ps ps z	ps ps	ps ps	ps ps
19	pm ps ns	ps ps ps	ps ps	ps ps	ps ps
22	pm ps ns	ps ps z	ps ps	ps ps	ps ps

Table 3: Output of a step ahead semi-constructive simulation

lation will now be discussed.

Consider the situation depicted in Figure 7, with the present value of an element of a variable-vector being *Medium*. The transition rules decree which value(s) that element may take in the next time period. Now, according to the rules only the present value or one of its neighbours can be the value in the succeeding time period. And this is the case regardless of whether the rules manifest themselves as filters or generators. That is, the result is the same regardless of whether the rule is of the form only generate a value which is the same as or adjacent to the present value, or of the form eliminate any value which is not the same as the present value or one of its neighbours.

The fact that QA gives the same results regardless of whether it is performed constructively or non-constructively has already been shown. Here we have demonstrated that transition rules will generate the same successor values whether they are utilised as filters or generators. Taking these two facts together indicates that one should expect that a simulation performed non-constructively will yield the same behaviour tree as one performed semi-constructively. And once again, since all possible transitions are captured, both the non-constructive and semi-constructive algorithms produce sound simulations (again in QSIM terms).

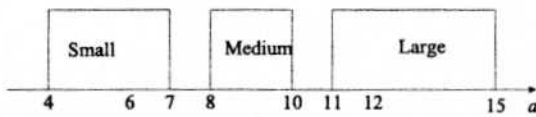


Figure 7: Transition Quantities

Generating a Complete Behaviour Tree In the preceding discussion we have concentrated on comparing different techniques and features for performing the simulation. However, a qualitative simulator must actually be able to generate a complete set of behaviours for the system of interest. In Figure 8 the behaviour tree is generated by letting the system discussed in the previous sections (with the same input

and initial conditions) run to completion; that is, till all the paths to equilibrium have been found or some resource limit has been exceeded. It is obvious from this tree that a substantial number of behaviours can be generated. This set will include a number of spurious behaviours (caused by the qualitative nature of the transition algorithm and the effects of error propagation), as well as all the real behaviours of the system. It is therefore desirable to have a means of reducing the size of the behaviour tree whilst, ideally, retaining only real behaviours. Hence in the next section an attempt to make asynchronous fuzzy qualitative simulation fully constructive is examined.

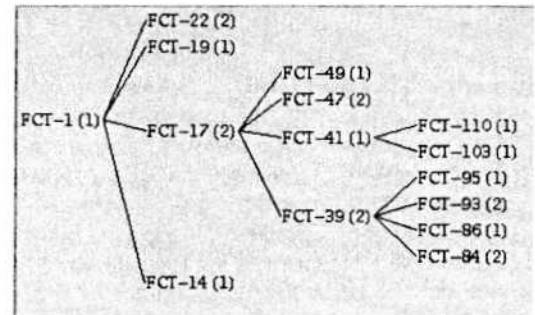


Figure 8: Behaviour tree for semi-constructive simulation

Conclusions The conclusions of this section are as follows:

1. Non-constructive and semi-constructive simulators will produce the same behaviour trees if the simulation is performed on the same model.
2. Constructive QA is more efficient than non-constructive for initialisation.
3. Semi-constructive transition analysis is exponential; therefore the efficiency advantage may be lost overall.

Constructive Simulation

The primary purpose of this section is to allow the comparison of the results of simulation utilising a constructive TA phase with the results generated by the semi-constructive simulator discussed in the previous section.

The main motivation for developing a constructive simulation engine was the belief that a constructive approach will prevent some spurious behaviours from being generated which it would not be possible to remove by filtering. In the case of the QA phase this approach has been shown not to yield results that differ from those produced via a non-constructive simulation

engine; thus negating that aspect of the motivation. However, it was argued that some gains were made from the improved efficiency of the causally ordered QA, though any advantage is lost due to the exponential nature of the TA phase of semi-constructive simulation. The above situation points to two motivations for developing a simulator with a constructive TA phase. First, even though the QA phase does not yield any advantage with respect to removing spurious behaviours, because constructive TA selects a single transition on the basis of a temporal criterion it will reduce the size of the behaviour tree, permitting experimentation to determine the types of behaviour selected. Also, because the transiting variable is selected on the basis of the information associated with the existing states of the system in a single pass, the TA is no longer exponential, which constitutes a major improvement in efficiency.

As with FuSim, *Mycroft* provides temporal information regarding the behaviour of the system under consideration. Because the quantities in the quantity space are fuzzy intervals, the times calculated will also be intervals. In fact there are two intervals associated with the transition of a variable between two states: the Departure time and the Arrival time. As stated, these are intervals with upper and lower bounds, which for the purposes of constructive simulation are named the: *possible departure time*, the *guaranteed departure time*, the *possible arrival time* and the *guaranteed arrival time*. The constructive TA was tested with each of these as the transition criterion and it was discovered that similar results were obtained regardless of the criterion used.

Experimental Results For this set of experiments we will consider the same system as for the semi-constructive version of the algorithm, with the same input and initial conditions. The initialisation phase is exactly the same for the semi-constructive and constructive versions of the *Mycroft* simulator. This is because the QA phase is identical in both versions: the only difference lies in the way that the TA is carried out. To illustrate the way in which a set of future states are generated, consider the initial state chosen in the previous section (state 1) with the guaranteed departure time as the selected transition criterion. In this case the guaranteed departure times of the state variables and their derivatives are as follows:

$$h_1 = 3.0, 3.0$$

$$h_2 = 0.0, 3.0$$

where in each case the first number is the time for the magnitude of the variable and the second number is the time for the first derivative (the second derivative, being a highest derivative of the system represented by three element vectors, has no transit time associated with it). From these figures it can be seen that the magnitude of h_2 has the earliest guaranteed departure time. This variable-vector element is therefore selected with the appropriate predicted next value and passed to the QA phase, as depicted in Figure 9.

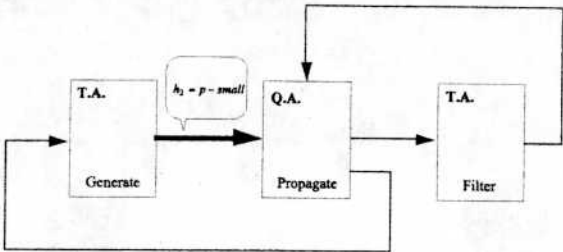


Figure 9: Information passed to the QA phase in constructive simulation

State	h1	h2	h12	qx	qo
14	ps ps ns	ps ps ps	ps ps	ps ps	ps ps
18	ps ps ns	ps ps z	ps ps	ps ps	ps ps

Table 5: Output of a step ahead constructive simulation

QA is carried out in the same manner as for the semi-constructive version with the exception that instead of having to start with the magnitudes of the state variables it may now start with one of their derivatives. The values of variable-vector elements calculated from a constraint higher up in the list of constraints remain the same as they were in the previous time period. The set of states thus generated are shown in Table 5. It can be seen that they form a subset of those states generated by the step ahead simulation. Indeed the whole behaviour tree, Figure 10, created by utilising a constructive TA phase is a sub-tree of that created by semi-constructive simulation, and finds a path to one of the steady state values generated by the semi-constructive version. That is, state FCT-31 in Figure 10 corresponds to state FCT-103 in Figure 8.

From these results it might appear that using the temporal boundaries is a useful criterion for assessing which branch of the complete behaviour tree to explore. Unfortunately, it transpires that these results are not typical. They do indicate that a temporal criterion can be used to find one path through the transient response to steady state for a dynamic system. However, in the majority of the examples examined they did not find any path to a steady state for the coupled tanks system. Why this is so will be demonstrated by means of another example.

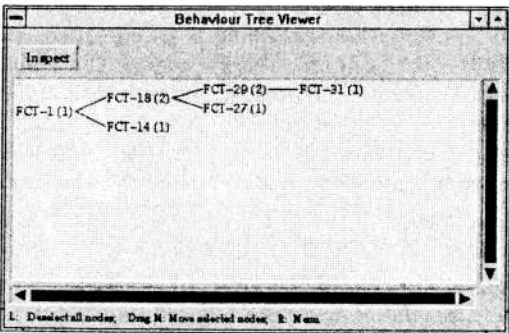


Figure 10: Constructive behaviour tree

Consider the coupled tanks system with a continuous input of [p - medium zero] and the same initial conditions as before. The behaviour tree generated contains no equilibrium states.

Table 6 shows one behaviour from the behaviour tree generated via the semi-constructive simulator. The variables values for this path from the initial state to an equilibrium state are also shown in Table 6. In Table 6, the state immediately preceding the equilibrium state is FCT-217; this matches a state with label FCT-140 in for this constructive simulation. The variable vectors for state FCT-140, with their associated guaranteed departure times, are as follows:

State FCT-140

$$h_1 = \langle pl \ ps \ ns \rangle, 5.476, 9.667$$

$$h_2 = \langle ps \ ps \ ns \rangle, 5.476, 9.667$$

From the guaranteed departure times for state FCT-140 it can be seen that the first derivative of h_2 has the earliest guaranteed departure time and will therefore be selected as the most likely transition. The value which it will take in the next time interval is zero. Unfortunately, there is no state reachable from state FCT-140 which can have the first derivative of h_2 equal to zero. To reach equilibrium, the first derivatives of both h_1 and h_2 ought to have transited together. This highlights a significant difficulty with this method of prioritising the transitions of variable-vector elements. The problem is that the information available relates to the boundaries of time intervals; and these are singular real numbers. Given the number of different transitions which can possibly occur and the different paths which a variable can take, it becomes extremely unlikely that any multiple variable-vector elements will have exactly the same time in accordance with the selected transition criterion.

Conclusion In this section the effect of making the simulation as constructive as possible has been explored. The motivation for this task is to reduce the number of spurious behaviours generated (Wiegand 1991) and it can be viewed as performing the role of a temporal prioritiser to be combined with a constraint prioritiser (Leitch & Shen 1993) to help select the most likely behaviour as the simulation. It was found that although it was possible to preferentially explore a subtree of the behaviour tree generated by the semi-constructive simulator, it was also possible that no real behaviour would be found. In fact, the unfortunate state of affairs is that the more common effect of constructive transiting is to eliminate all the possible real behaviours. Also, because the constructive algorithm only explores one part of the behaviour tree generated with the semi-constructive simulation it is no longer sound. This was demonstrated by the fact that in several cases no real behaviours were found.

General Discussion

The goal in this paper has been to examine the operation and results of constructive and non-constructive simulations. In comparing the non-constructive and semi-constructive approaches to fuzzy qualitative simulation it was shown that the results were the same

regardless of which approach was taken. That is, the semi-constructive approach does not remove any spurious states that would be predicted using a non-constructive simulation, and *vice versa*.

It could be suggested that the semi-constructive version is preferable in model-based diagnosis where one has to initialise a system, possibly several times, in the course of a diagnosis. However, the semi-constructive approach relies on the modeller being able to causally order the constraints of the system. In a large number of cases this is possible; but there are systems, the models of which contain algebraic loops and thus cannot be causally ordered. These arise when there are insufficient measurements available to resolve the relationships in the model. In this case a non-constructive approach which can handle loops would be the appropriate (in fact the only) choice.

While it is true that no clear winner emerges from the contest between the semi-constructive and non-constructive algorithms, the aspects of the models as they appear in *Mycroft*, namely the vector representation of the variables and the spread of the constraints across differential planes have the advantage of flexibility in allowing the model to be constructed according to the needs of the situation. These are independent of the form of the algorithm used. Therefore, if it were the case that algebraic loops did appear in a model, one could still use vectors for the variables and differential planes for the constraints along with a non-constructive algorithm (that is to say that neither of these features is dependent on the constraints of the model being causally ordered). In fact it is the case that this is currently being incorporated into the framework. The other two sets of simulation experiments described and discussed in this section both sought to reduce the behaviour tree in a rational manner, so as to retain some of the real behaviours whilst removing those which are spurious. It has been demonstrated that *Mycroft* is not very successful in this regard. However, it has also been shown that when it did find an expected set of behaviours it generated a much smaller behaviour tree and was thus more efficient; an important motivation if qualitative reasoning systems are going to provide useful tools for real-time applications. It is certain that the correct approach will involve some form of prioritisation of both the QA and the TA. The harder of the two will undoubtedly be that of the transitions because of the fact that the time intervals expand as the simulation proceeds and they are not constrained by a quantity space. In fact it is not obvious how one can decide on what qualifies as a valid temporal prioritisation scheme. A great deal of effort has been expended over the past few years on developing global filters for qualitative simulators. Since prioritisation operates on the assumption that the states being prioritised are real it should only be applied as an adjunct to those global filters whose aim is to remove spurious behaviours (after they have been applied).

Conclusions

In this paper we have examined the operation of the *Mycroft* fuzzy qualitative reasoning framework with

Path → Variables ↓	FCT-1	FCT-13	FCT-45	FCT-217	FCT-566
h_1	ps pm nm	ps ps ns	pm ps ns	pl ps ns	pt z z
h_2	z ps pm	ps ps ps	ps ps z	ps ps ns	pm z z

Table 6: A path to steady state

respect to the novel aspects of its design. The results obtained have been compared with those from other simulators and envisioners. It was found that semi-constructive and non-constructive simulations gave the same results when applied to the same model. Also, making the simulation constructive reduced the size of the behaviour tree and could select a real behaviour, but in most cases did not find any of the real behaviours.

From a comparison of the different sets of results and local conclusions the following general conclusions can be drawn:

1. Performing a simulation semi-constructively or non-constructively results in the same behaviour tree being generated. The most important aspect of this result is that the QA phase results in the same set of states being generated (whether in an initialisation or during a simulation). This shows that the main motivation for making the QA phase constructive has been falsified.
2. Performing the simulation constructively does produce a smaller behaviour tree than the semi-constructive approach, making it more efficient. A constructive simulation can also produce real behaviours, though a more common result in the experiments carried out was that no real behaviours were found. This result shows that while it is efficient, its utility is limited.
3. Probably the most important result of these experiments is the confirmation they give to the idea of "horses for courses". It is plain that a causally ordered model reveals the interactions of the relations in a straightforward manner and constructive QA gives an efficient initialisation process, which may be useful for diagnosis or explanation. On the other hand non-constructive approaches permit the simulation of systems which contain algebraic loops.

The *Mycroft* is still being developed with other aspects being incorporated and investigated (Scott & Coghill 1998). Currently non-constructive (synchronous and asynchronous) algorithms are being developed.

References

- Berleant D. & Kuipers B. J., Combined qualitative and numerical simulation with Q3, in Proceedings of the Fourth International Workshop on Qualitative Reasoning about Physical Systems, Pp 140-152, Lugano, Switzerland, 9-12 July, 1990.
- Bousson K & Trave-Massuyes L., "Putting more numbers in the Qualitative Simulator CA-EN" in Proceedings of the Second International Conference on Intelligent Systems Engineering, Pp 62-69, Hamburg, 5-9 September 1994.
- Coghill G. M. "Mycroft: A framework for constraint-based fuzzy qualitative reasoning", Ph.D. Thesis, Heriot-Watt University, September 1996.
- Iwasaki Y, "Causal ordering in a mixed structure", Proceedings AAAI-88, St Paul, Minn, Pp 313 - 318, 1988.
- Kosko B., Neural networks and fuzzy systems, Prentice-Hall International, 1992.
- Kuipers, B. J., "Qualitative simulation", Artificial Intelligence, 29, 289-338, 1986.
- Kuipers, B. J. *Qualitative Reasoning*, MIT Press 1994.
- Kuipers B. J. & Chui C., "Taming Intractable Branching in Qualitative Reasoning", in Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI-87, Pp 23-28, Milan, Italy, 1987.
- Leitch R. R. & Shen Q. "Prioritising behaviours in Qualitative Simulation", in Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI-93, Pp 1523 -1528, Chambourcy, France, 1993.
- Leitch R. R., Shen Q., Coghill G. M., Chantler M. J. & Slater A. F., "Qualitative model-based diagnosis of a continuous process", in Proceedings of the Eighth International Conference on the Applications of Artificial Intelligence in Engineering, vol. 2, Pp 291-309, Toulouse, France, 1993.
- Morgan A. J., "Qualitative Behaviour of Dynamic Physical Systems", Ph.D. Thesis, University of Cambridge, 1988.
- Scott J. A. & Coghill G. M., "Qualitative Euler Integration with Continuity" in Proceedings of the Twelfth International Workshop on Qualitative Reasoning, Cape Cod, Mass
- Shen Q. and Leitch R. R., "Fuzzy qualitative simulation". in IEEE Transactions on Systems, Man and Cybernetics, 23, 1038 - 1061, 1993.
- Shoup T. E., A Practical Guide to Computer Methods for Engineers, Prentice-Hall International, Englewood Cliffs, N.J., 1979.
- Struss P. "Mathematical Aspects of Qualitative Reasoning", Artificial Intelligence in Engineering, 3, -170, 1988.
- Wiegand M. E., Constructive Qualitative Simulation of Continuous Dynamic Systems, Ph.D. Thesis, Heriot-Watt University, May 1991.
- Williams B. C., "The use of continuity in a qualitative physics" in Proceedings AAAI-84, 350-354, Austin, Texas, 1984.
- Zadeh L. Fuzzy Sets, Information and Control, 8, 338-353, 1965.