ELSEVIER

Artificial Intelligence 95 (1997) 215-255

Artificial Intelligence

Qualitative and quantitative simulation: bridging the gap ☆

Daniel Berleant [a,*], Benjamin J. Kuipers [b,1]

[a] *Department of Computer Systems Engineering, University of Arkansas, Fayetteville, AR 72701, USA*

[b] *Department of Computer Sciences, University of Texas, Austin, TX 78712, USA*

Received January 1993; revised May 1997

Abstract

Shortcomings of qualitative simulation and of quantitative simulation motivate combining them to do simulations exhibiting strengths of both. The resulting class of techniques is called *semi-quantitative simulation*. One approach to semi-quantitative simulation is to use numeric intervals to represent incomplete quantitative information. In this research we demonstrate semi-quantitative simulation using intervals in an implemented semi-quantitative simulator called Q3. Q3 progressively refines a qualitative simulation, providing increasingly specific quantitative predictions which can converge to a numerical simulation in the limit while retaining important correctness guarantees from qualitative and interval simulation techniques.

Q3's simulations are based on a technique we call *step size refinement*. While a pure qualitative simulation has a very coarse step size, representing the state of a system trajectory at relatively few qualitatively distinct states, Q3 interpolates newly explicit states between distinct qualitative states, thereby representing more states which instantiate new constraints, leading to improved quantitative inferences.

Q3's techniques have been used for prediction, measurement interpretation, diagnosis, and even analysis of the probabilities of qualitative behaviors. Because Q3 shares important expressive and inferential properties of both qualitative and quantitative simulation, Q3 helps to bridge the gap between qualitative and quantitative simulation. © 1997 Elsevier Science B.V.

Keywords: Qualitative; Simulation; Semi-quantitative; Intervals; QSIM; Q3

☆ This research was supported in part by NSF grants IRI-8905494 and IRI-8904454 and NASA grant NAG 5207, and performed in part for the Jet Propulsion Laboratory, California Institute of Technology, sponsored by the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, or service does not constitute or imply its endorsement by any organization.

* Corresponding author. Email: djb@engr.uark.edu.

[1] Email: kuipers@cs.utexas.edu.

PII S0004-3702(97)00050-7

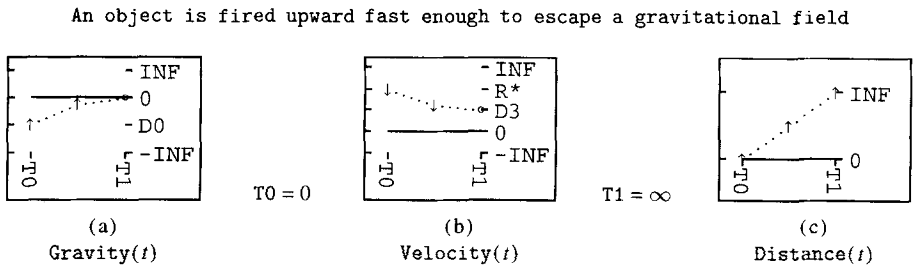


Fig. 1. Qualitative simulation of an object fired upward at greater than escape velocity shows that the gravitation experienced by the object produces a negative acceleration (a), reducing its velocity (b). As distance increases (c), gravitation decreases. Qualitative simulation also produces another behavior in which the object falls back to Earth (not shown).

1. Introduction

Systems that change over time are often so complex that analytical solutions, equations predicting future system states as a function of time, cannot be found. In those cases simulation is useful for prediction. Given a model of system structure and initial state, simulation determines the system's trajectory through its state space.

When accurate numerical information about structure and initial state is available, a large body of numerical simulation techniques is available. When only qualitative information about a model is available, a significant body of work describes methods for qualitative simulation. But what about the many cases in which accurate numerical information is unavailable, preventing traditional numerical simulation, yet incomplete numerical information is available, providing the potential for stronger predictions than pure qualitative simulation provides? That question motivates semi-quantitative simulation, in which numerical and qualitative techniques are combined to make more informative inferences than either alone would make.

As an example consider a nonlinear, second-order system, a rocket fired straight up in a gravitational field that decreases with height. Compared to the simple case of movement in a gravitational field that remains constant with height, this example is more interesting for qualitative simulation due to its nonlinear and second-order nature.

To illustrate the point that qualitative and numerical simulation have relative strengths and weaknesses, here is a limitation of each:

- Numerical simulation cannot infer that the final height could be infinite. More generally, unlike qualitative simulation (Fig. 1), numerical simulations cannot infer infinite values at all.
- Qualitative simulations cannot infer whether or not the rocket rises to infinity as shown in Fig. 1, or instead falls back to the ground. More generally, unlike numerical simulation, qualitative simulation cannot infer which qualitative behavior will be the one to actually occur in a given instance.

Semi-quantitative simulation combines both qualitative and quantitative simulations so as to compensate for weaknesses in each with strengths of the other. This leads to significant guarantees which semi-quantitative simulation can provide.

- All qualitative behaviors that are consistent with available quantitative information can be found.
- Each qualitative behavior can either be annotated with intervals providing quantitative bounds on system trajectories conforming to that qualitative behavior, or ruled out entirely.

In an earlier system, Q2 [59], we showed how a qualitative behavior's symbolic values can be annotated with intervals that bound their quantitative values. While Q2 provides useful results and has been used in other work [35, 36, 39, 53], it relies on simulations that contain only the few time points at which qualitatively significant events occur, which limits the quantitative inferences it can provide.

The present system, Q3, extends Q2 with step size refinement and auxiliary techniques. Step size refinement interpolates new states into an existing sequence of states in a simulation trajectory, adaptively reducing the size of the time steps in the simulation. Step size refinement constitutes a pragmatic contribution because it allows better inferences than either qualitative or quantitative simulation alone. From a theoretical perspective, it inherits important guarantees from both qualitative and quantitative simulations.

This paper significantly revises and expands a preliminary account [9], and provides a proof of convergence and stability for step size refinement.

2. Q3 and step size refinement

Q3 builds on the tree of qualitative simulation trajectories (behaviors) produced by Q2 [59]. That tree of qualitative simulation trajectories is annotated with intervals that constrain the values of model variables at qualitatively significant simulation time points in each trajectory. While these intervals allow better predictions about model trajectories than are possible with pure qualitative simulation, and better pruning of the tree of qualitative behaviors, Q2 often suffered from weak quantitative inferences in the form of very wide inferred intervals.

Somewhat better inferences might be obtained by augmenting Q2's simple constraint propagation with more sophisticated approaches, such as quantity lattices [78], Q1 [89] [2] or BOUNDER [71]. However while such sophisticated methods would help, a critical issue would still remain. This issue is the size of the time periods between explicitly represented time points in the simulation trajectories, and it is a variant of the well known issue of step size in numerical simulations of ordinary differential equations.

For typical numerical models, numerical simulation results are poor when step sizes are large, becoming progressively more accurate as the step size of the simulation becomes smaller. (A large step size means that qualitative features of a trajectory, such as slope of one or more model variables, change significantly from one time point to the next.) This basic characteristic of standard numerical simulation algorithms [46] is a

[2] The name Q1 has no connection with the names Q2 and Q3.

serious problem for numerically annotated qualitative simulations because the qualitative features of qualitative simulation trajectories do change significantly from one time point to the next. Therefore step sizes for qualitative simulations are large by definition and so numerical inferences on them tend to be weak.

Augmenting a Q2 simulation so that it has smaller step sizes can lead to greatly improved quantitative inferences, just as numerical simulations can be improved by reducing the step size (within limits imposed by the accuracy of floating point arithmetic). *Step size refinement* is our algorithm for doing this. Q3 augments Q2 with the theoretically and pragmatically significant capability of smaller step sizes.

Q3 first generates qualitative behaviors via QSIM which are annotated with quantitative information via Q2. Then, better inferences are obtained by progressively reducing the step size using step size refinement and auxiliary algorithms. Step size refinement is an *adaptive discretization* technique. Adaptive discretization techniques reduce numerical error in simulation methods that represent a continuous system at a finite number of discrete points, usually time points, by varying the step size depending on the current status of the simulation. Previously described adaptive discretization techniques include *adaptive step size control* [46, 70], and *multigrid* methods [16, 17].

Because the two main phases in the operation of Q3 are generating a simulation and refining it, we next describe each phase in turn.

2.1. Phase I: generating the simulation trace

Creating the simulation trace involves qualitative simulation in close coordination with quantitative inference. Q3 does this by calling Q2 [59] as a subroutine. The coordination consists of iterating over (a) using qualitative simulation to incrementally grow the behavior tree, then (b) performing quantitative inference on the incrementally extended tree. In more detail:

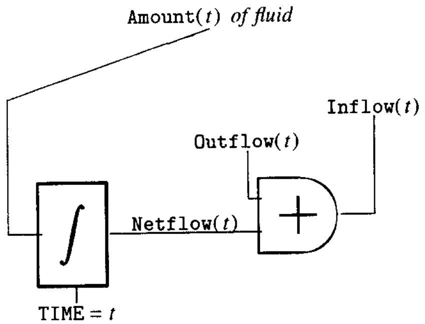
(1) *Qualitative simulation.* Incrementally grow a tree of qualitative behaviors.
The tree of behaviors is guaranteed to include the actual behavior of any real system conforming to a given qualitative model [58]. Q3 represents each behavior as a *constraint network* relating each model variable at the time value of each qualitative state in the simulation. This constraint network is annotated with intervals representing the quantitative information that is given or has been previously inferred about various landmark values of various model variables. Fig. 2 illustrates this constraint network concept with an elementary example.

(2) *Propagate quantitative information.* The new qualitative state creates new constraints which initiate new quantitative inferences from given and previously inferred intervals. The effects of these inferences ripple through the network, using constraint propagation (i.e. Waltz filtering on interval labels [27]). Propagation through an add constraint (Figs. 3 and A.1) exemplify this.

(3) *Iterate.* But if the qualitative simulation does not provide further incremental growth, go on to Phase II.

We now examine how quantitative information is propagated in detail.

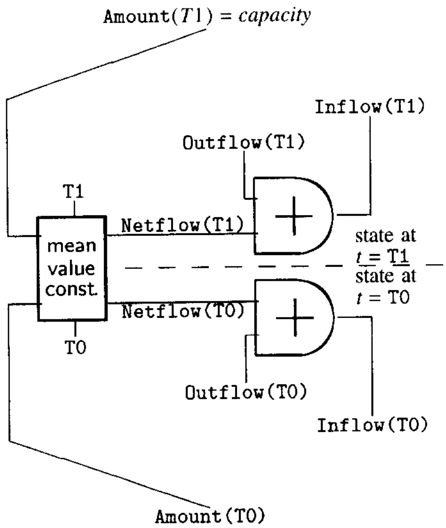
Model of a tank with fluid flow



Netflow(t) = Inflow(t) − Outflow(t)
Amount(t) = $\int_t$ Netflow(t)

(a) Diagram and equations of a *model*

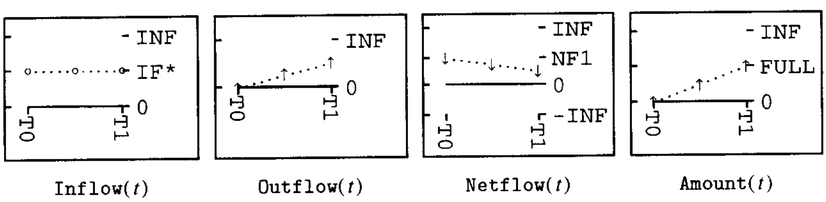
Overflow behavior of the tank



Netflow(T1) = Inflow(T1) − Outflow(T1)
Netflow(TO) = Inflow(TO) − Outflow(TO)

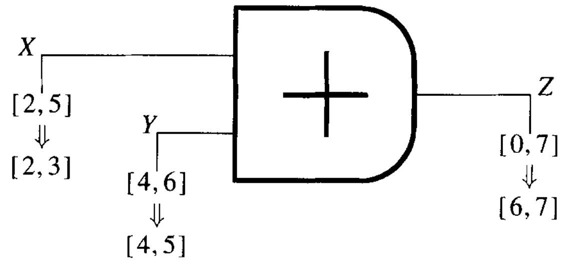
$\frac{\text{Amount(T1)} - \text{Amount(TO)}}{\text{T1} - \text{TO}} \subseteq$
$[\min(\underline{\text{Netflow(TO)}}, \underline{\text{Netflow(T1)}}),$
$\max(\overline{\text{Netflow(TO)}}, \overline{\text{Netflow(T1)}})]$

(b) Diagram and equations of a *behavior*



(c) Overflow behavior: graphical representation

Fig. 2. A simple tank model and its overflow behavior. The model consists of two constraint *templates*, shown in (a). Those templates hold at all times. *Instantiating* the templates at qualitatively distinctive time points TO and T1 leads to representations of its behaviors. A constraint network representation of the overflow behavior is shown in (b). A larger model or one with more time points would be unwieldy to depict graphically, but can easily exist as a data structure in memory. The same behavior represented graphically is shown in (c). Amount(TO) is the initial quantity of fluid in the tank. Overflow occurs at TIME = T1, when the value of Amount(T1) is the capacity of the tank. The mean value constraint (Section 2.1.3) relates six values, two each of Amount, Netflow, and TIME.



$$Z := Z \cap (X+Y) = [0,7] \cap ([2,5]+[4,6]) = [0,7] \cap [6,11] = [6,7]$$
$$X := X \cap (Z-Y) = [2,5] \cap ([6,7]-[4,6]) = [2,5] \cap [0,\ 3] = [2,3]$$
$$Y := Y \cap (Z-X) = [4,6] \cap ([6,7]-[2,3]) = [4,6] \cap [3,\ 5] = [4,5]$$

Fig. 3. *Constraint propagation* of interval labels through an add constraint. The interval at each terminal is narrowed by using the constraint to propagate the intervals currently at the other terminals, so an add constraint actually enforces three relations, one addition, $Z \subseteq X+Y$, and two subtractions, $X \subseteq Z-Y$ and $Y \subseteq Z-X$.

2.1.1. Propagating quantitative information

The constraint network composing each qualitative behavior consists of constraint *templates* (Fig. 2(a)) defining the model which are *instantiated* into actual constraints on the values of the model variables concerned at particular points in time (Fig. 2(b)). These constraints relate intervals that quantitatively bound the qualitative landmarks of model variables. The constraints often support narrowing of one or more intervals [27], where a narrower interval expresses less uncertainty about quantitative value. When an interval is narrowed (or assigned an initial value), the constraints directly affected can often narrow other interval(s) conected to them (Figs. 3 and A.1). Thus the effect of narrowing an interval can propagate, narrowing other intervals throughout the constraint network.

There are a number of different kinds of constraints that can be expressed in a model description.

- Arithmetic and monotonicity constraints among variables.
- Greater and less than constraints among the different qualitative landmark values of a given model variable.
- Mean value constraints relating values in states at adjacent time points in the simulation trajectory.

Propagation of intervals using some of these constraints is simple, but for others it is less so.

To illustrate the greater and less than constraints, suppose some landmark `topHeight` is greater than some other landmark `bottomHeight`, and we are given that `topHeight` $\in [100,200]$ and `bottomHeight` $\in [0,\infty)$. Then we can infer `bottomHeight` $\in [0,200]$.

To illustrate arithmetic constraints, inferencing is exemplified by Fig. 3 and is consistent with previous work [49]. Subtraction is modeled using the add (Fig. 3), and division relations are modeled analogously with the `mult` constraint.

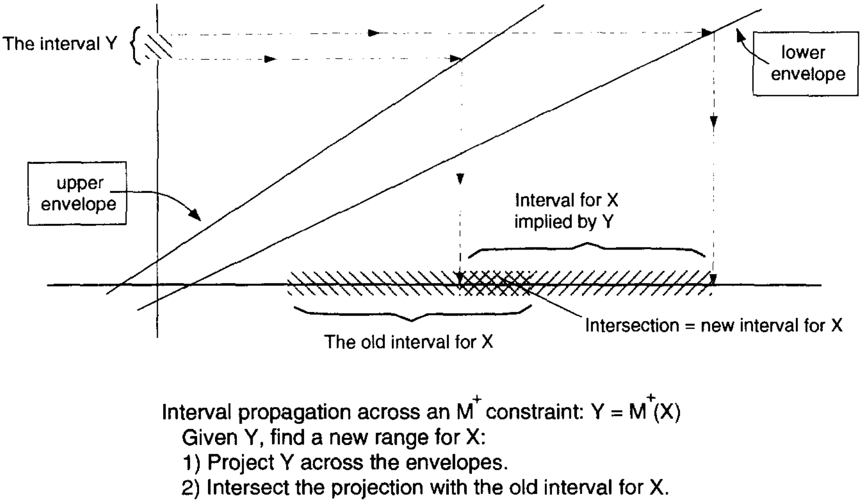


Fig. 4. Propagation through an M^+ (positive monotonic) function with quantitative envelopes. Hatched regions of the axes represent intervals, and the upper and lower envelopes bound a space of monotonic functions. Propagation shown is from an interval on the y-axis to the x-axis. Propagation the other way, from x to y is analogous.

The monotonicity and mean value constraints are more involved, and are explained next.

2.1.2. The monotonicity constraint

Monotonicity implies that a change in one variable leads to a change in the other variable, in the same direction for positive monotonicity and in the opposite direction for negative monotonicity. If two variables are monotonically related, the highest and lowest points of an interval on one of them imply highest and lowest points of the projection of that interval on the other.[3]

A *qualitative* monotonic function represents a large set of *quantitative* functions consisting of all those that are monotonic in the direction specified by the qualitative monotonic function. A middle ground between qualitative monotonic functions and specific numeric monotonic functions is upper and lower monotonic *envelopes* which bound a space of numeric monotonic functions. This is illustrated by Fig. 4, which shows a simple propagation method justified by the definition of monotonicity.

2.1.3. The mean value constraint

The mean value constraint is designed to allow propagation of quantitative information from one time point to another. It derives directly from the mean value theorem of elementary calculus, which states:

[3] See e.g. [49, pp. 89–90]. Nonmonotonic envelope functions and relations call for different analysis techniques [12].

$$\exists t^*: rate(t^*) = \frac{level(t_n) - level(t_{n-1})}{t_n - t_{n-1}}, \tag{1}$$

where time $t^* \in (t_{n-1}, t_n)$. *rate* designates a derivative quantity and *level* an integral quantity [45] related by

$$rate = \frac{d(level)}{dt}.$$

We do not know the value of t^*, but can conservatively infer that

$$rate(t^*) \in [\min(rate(t_{n-1}), rate(t_n)), \max(rate(t_{n-1}), rate(t_n))], \tag{2}$$

because $t^* \in (t_{n-1}, t_n)$, because a closed interval $I = [a, b]$ is a superset of the open interval (a, b), and because the qualitative simulation module (QSIM) ensures that $rate(t)$ and all other varying quantities are monotonic between states at adjacent time points t_{n-1} and t_n.

From (1) and (2),

$$\frac{level(t_n) - level(t_{n-1})}{t_n - t_{n-1}} \in [\min(rate(t_{n-1}), rate(t_n)), \max(rate(t_{n-1}), rate(t_n))]. \tag{3}$$

When quantities are known only to within intervals, this equation must be intervalized in the obvious ways. Real variables x_i are replaced by corresponding interval variables X_i, real functions $f(x_i)$ are replaced by corresponding interval functions $F(X_i)$, real arithmetic operations such as "−" are given the corresponding interval interpretations, the natural interval extension [66] of $\in$ is $\subseteq$, and functions min() and max() are applied respectively to the low and high bounds of intervals. Intervalizing (3) gives

$$\frac{LEVEL(T_n) - LEVEL(T_{n-1})}{T_n - T_{n-1}} \subseteq [\min(\underline{RATE(T_{n-1})}, \underline{RATE(T_n)}), \max(\overline{RATE(T_{n-1})}, \overline{RATE(T_n)})],$$

where the low bound of an interval X is denoted by $\underline{X}$ and the high bound by $\overline{X}$ [66].

The right-hand side simplifies to the convex hull of the set $RATE(T_{n-1}) \cup RATE(T_n)$, or $RATE(T_{n-1}) \underline{\cup} RATE(T_n)$, where the convex hull includes everything in either interval or between them. This results in the *mean value constraint*:

$$\frac{LEVEL(T_n) - LEVEL(T_{n-1})}{T_n - T_{n-1}} \subseteq RATE(T_{n-1}) \underline{\cup} RATE(T_n). \tag{4}$$

Eq. (4) can be solved algebraically for each variable on the left hand side. The resulting right-hand side can then be evaluated to give an interval, which is intersected with the quantity's current interval as in Fig. 3.

Quantitative inferences provided by the mean value constraint tend to be weak when the values T_{n-1} and T_n are widely separated, as is often the case with qualitatively distinct time points (as in Fig. 6(a)). Much better results are typically obtained from the mean value constraint after step size refinement, which makes adjacent time points closer together (as in Fig. 6(b)).

Alternatives to the mean value constraint. The mean value constraint is based on Euler's method. An obvious improvement over the relatively weak Euler's method is the Runge–Kutta method, a mainstay of numerical simulation. For interval problems, Lohner [60] shows that any 1-step method (such as Runge–Kutta) can be extended to interval simulation. Another direction would be to use an existence and convergence theorem for interval operator equations [66, Theorem 5.7]. Moore [66, pp. 94–97] also describes a Taylor series based method for interval simulation. Eijgenraam [37] and Lohner [60] describe other methods. Missier and Travé-Massuyès [64] demonstrate the feasibility of a Taylor series based approach in semi-quantitative simulation.

2.2. *Phase II: progressive refinement*

A quantitatively annotated qualitative simulation was generated in Phase I, and is now progressively refined in Phase II. The mainstay of this refinement process is the step size refinement algorithm, which gradually reduces the size of the time steps by interpolating new time points between existing ones (auxiliary techniques are behavior splitting, Section 3, and target interval splitting, Section 3 and Appendix A). Step size refinement is presented next.

2.2.1. *Step size refinement: overview*

Standard numerical simulation algorithms estimate system state at the next time point by extrapolating from current trends. It is better to extrapolate only a short distance along the system trajectory and then to reassess current trends before extrapolating further, than to extrapolate over a longer distance. This means keeping the step size of the simulation small and, intuitively, is why small step sizes typically lead to less error in the predicted trajectory of a numerical simulation.

For most interval generalizations of numerical simulation methods [60] smaller step sizes lead to narrower but correct interval predictions. This is the case for step size refinement. In step size refinement, the step sizes of the simulation are decreased gradually, leading to increasingly sharp predictions in the form of narrower intervals.

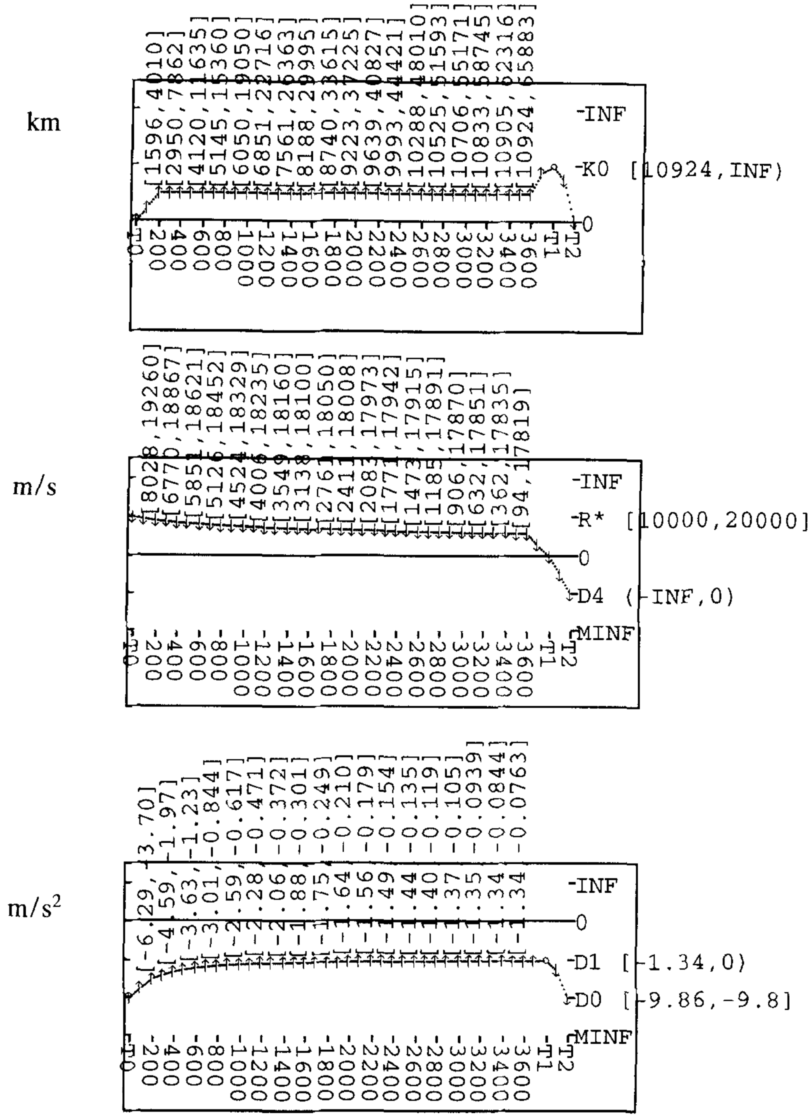
These intuitions about step size refinement are described precisely in Section 4.2 and formally proven in Appendix B. We now describe the algorithm.

2.2.2. *Step size refinement: algorithm*

We are given a finite ordered sequence `T0, T1, ..., Tn` of time points `Ti`, each with an associated interval range $[\underline{t_i}, \overline{t_i}]$ representing uncertainty about its value. The range for `T0` is $[0, 0]$. The ranges may overlap, and a range upper limit $\overline{t_i}$ may be infinite.

Iterating over the following three steps progressively sharpens the predictions of the simulation.

Example. Twenty-five iterations were done to produce Fig. 6(b) from Fig. 6(a).

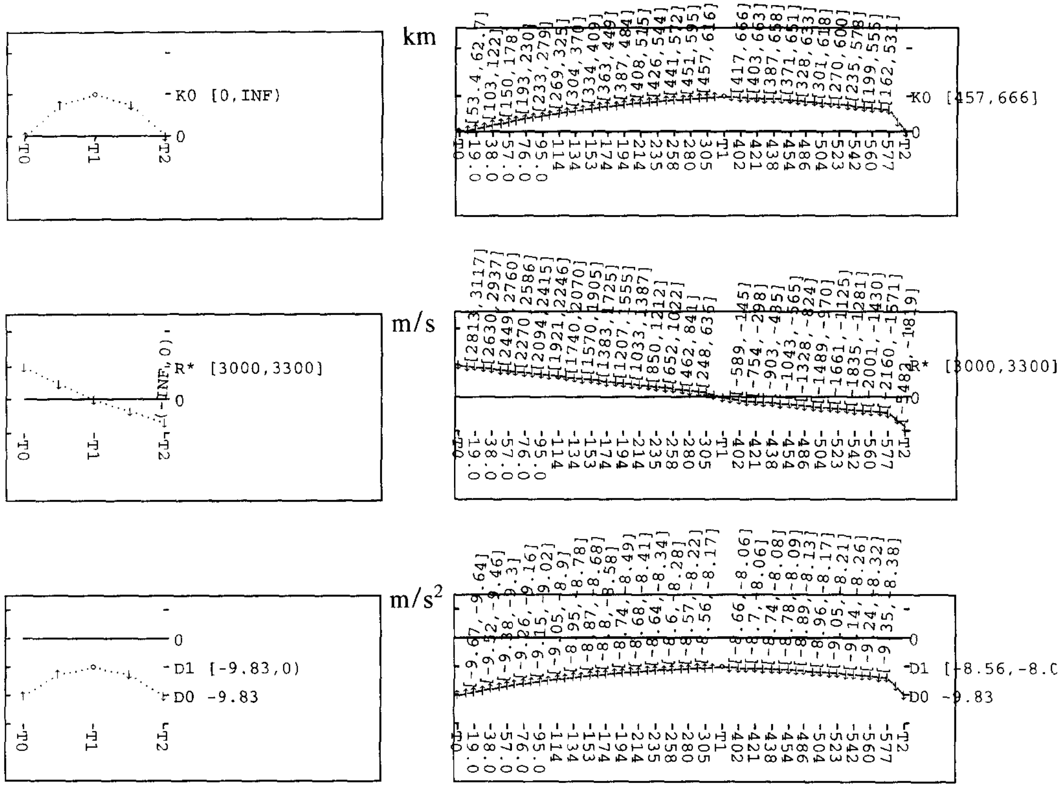


$T0 \in [0,0] = 0$

$T1 \in [3671, \infty)$

$T2 \in [3671, \infty)$

Fig. 5. A rocket is fired upward from the Earth's surface. Gravity decreases with the squared distance from the Earth's center (Fig. 8). Initial velocity is R* $\in$ [10000,20000] m/s, which contains escape velocity (approximately 11000 m/s). Q3 correctly predicts that the rocket could either return to Earth or escape. The return behavior is the one shown, with graphs for height, velocity, and acceleration versus time. Inferences about qualitative time points T0, T1, and T2 are shown at the bottom.



T0 ∈ [0,0]
T1 ∈ [305,∞)
T2 ∈ [305,∞)

(a)

T0 ∈ [0,0]
T1 ∈ [334,384]
T2 ∈ [607,915]

(b)

Fig. 6. A rocket is fired upward at [3000,3300] meters per second, less than escape velocity. The behavior in which the rocket falls back to the ground is shown in (a), which also shows weak quantitative inferences that were unable to prune any of the other two behaviors. In (b) the same behavior is shown, however 25 time points were interpolated. Consequently, quantitative inferences are much stronger—and the 2 impossible behaviors have been pruned. Table 1 summarizes key intermediate stages in the simulation.

1. *Locate a gap*. Find a time interval containing a *gap*. That is, find an i such that $\overline{t_i} < \underline{t_{i+1}}$ (see Fig. 7). If no gap exists, create one using an auxiliary method to be described later.
   Example: Fig. 6(a) indicates a gap with *width* = 305 between T0 ∈ [0,0] and T1 ∈ [305,∞).
   Example: Consider the step going from time 3600 to time [3671,∞) in Fig. 5. There is a gap between 3600 and T1 ∈ [3671,∞) from (3600,3671) where any interpolated state will certainly occur after 3600 but before T1.

Table 1
The returning rocket simulation at various stages in its refinement. Values of four model variables are each shown at TIME = 153 (the first time point interpolated by the step size refinement algorithm), and either T1 when the rocket is at its highest point or T2 when it hits the ground on the way down. Each technique provided by Q3 has been used in at least one stage. Note that interval results get narrower as the stages progress. (Rows (a), (b), (c), ..., are keyed to descriptions in the text.)

Variable:	Acceleration		Velocity		Height		Time		#beh's
TIME:	153	T1	153	T2	153	T1	T1	T2	
Stage:									
(a) Phase I		$[-9.83, 0)$		$(-\infty, 0)$		$(0, \infty)$	$[305, \infty)$	$[305, \infty)$	3
(b) 1 interp.	$[-9.16, -8.44]$	$[-9.16, 0)$	$[1496, 2009]$	$(-\infty, 0)$	$[229, 505]$	$[229, \infty)$	$[316, \infty)$	$[316, \infty)$	1
(c) 4 interp.	$[-8.99, -8.57]$	$[-8.77, 0)$	$[1535, 1948]$	$(-\infty, 0)$	$[289, 452]$	$[373, \infty)$	$[326, \infty)$	$[326, \infty)$	1
(d) TIS	$[-8.99, -8.57]$	$[-8.77, 0)$	$[1535, 1948]$	$(-\infty, 0)$	$[289, 452]$	$[373, \infty)$	$[326, \infty)$	$[519, \infty)$	1
(e) 7 interp.	$[-8.91, -8.64]$	$[-8.65, 0)$	$[1558, 1919]$	$(-\infty, 0)$	$[319, 424]$	$[422, \infty)$	$[331, \infty)$	$[519, \infty)$	1
(f) Beh. split	$[-8.91, -8.64]$	$[-8.65, -0.00052]$	$[1558, 1919]$	$(-\infty, 0)$	$[319, 424]$	$[422, 866624]$	$[331, 1e6)$	$[519, \infty)$	2
(g) 8 interp.	$[-8.91, -8.64]$	$[-8.63, -7.98]$	$[1558, 1919]$	$(-\infty, -1044]$	$[319, 424]$	$[428, 699]$	$[331, 388]$	$[519, \infty)$	1
(h) 25 interp.	$[-8.87, -8.68]$	$[-8.56, -8.06]$	$[1570, 1905]$	$[-5482, -1819]$	$[334, 409]$	$[457, 666]$	$[334, 384]$	$[607, 915]$	1
(i) 50 interp.	$[-8.87, -8.69]$	$[-8.53, -8.10]$	$[1572, 1902]$	$[-4653, -2137]$	$[337, 407]$	$[470, 646]$	$[335, 382]$	$[607, 835]$	1

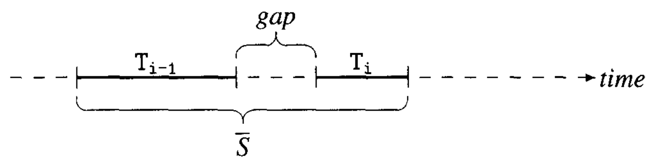


Fig. 7. T_{i-1} and T_i are time *points*, but their values are known only to within intervals, indicated by solid line segments. Between them is a gap. T_{i-1} and T_i are the endpoints of a time step whose size is in $[w(gap), \overline{S}]$, where $w(gap)$ is the width of the gap. $\overline{S}$ is the maximum possible step size from T_{i-1} to T_i.

2. *Interpolate a state.* Insert a new state with a time point Ta in the gap and assign it the zero-width interval $[t^*, t^*]$, where t^* is a number within the gap. This reduces the step size of the simulation. Q3 by default picks values of t^* rounded to the nearest integer, to communicate better with the viewer by avoiding visual clutter in the display [85] (see Fig. 6(b)). The multiple to which rounding is done is easily customizable and multiples of 200 were used in Fig. 5. The values of the other model variables in this newly created state must be between their values in the two adjacent previously existing states. This enables initializing the new state with qualitative and interval values.
3. *Propagate interval bounds.* The newly interpolated time-point T_a creates a new set of instantiations of the constraint templates defined by the model. Propagation of the new interval value $T_a = [t^*, t^*]$ and previously existing intervals through the newly expanded constraint network can narrow various interval bounds throughout the network [27].
   Example: Fig. 2 illustrates a model (a) and its instantiated constraints (b).
   Example: A detailed, step-by-step account of how propagation through the expanded constraint network resulting from an interpolated time point led to markedly improved quantitative inferences appears in [5].

3. Detailed example: a nonlinear second-order rocket

We now step through an example requiring step size refinement and the auxiliary techniques Q3 offers. Consider a rocket in a gravitational field which decreases with distance. This system is both second-order and nonlinear, and hence makes a useful demonstration example. The qualitative model appears in Fig. 8. The simulation for this example was initialized with parameters describing known quantitative data about the Earth (Fig. 9) and a velocity in [3000, 3300] m/s, less than the escape velocity of 11,000 m/s so it must fall back to Earth. [4]

To direct Q3's operation, we specify a goal.

> Goal: "*Prune as many behaviors as possible and numerically bound the remaining behavior(s).*"

[4] Rotation and atmospheric resistance are ignored.

```
(define-QDE escape-velocity
(text "Gravity decreases with height as r''=-GM/r^2")
; Define model variables and their qualitative values.
(quantity-spaces
  (r        (0    sea-level inf)  "meters from Earth's core")
  (r^2      (0    inf         )  "distance squared"        )
  (h        (0    inf         )  "meters above surface"    )
  (km       (0    inf         )  "kilometers above surface")
  (surface  (0    s*          )  "depth of Earth"          )
  (dr/dt    (minf 0 r*     inf)  "velocity, m/s"           )
  (d2r/dt2  (minf 0        inf)  "acceleration, m/s^2"     )
  (G        (0    G*          )  "Gravitational constant G")
  (Earth-M  (0    M*          )  "Mass of Earth M"         )
  (K        (0    K*          )  "K (=G*M)"                )
  (-K       (-K*  0           )  "-K"                      )
  (=1000    (0    1000        )  "Thousand"                ))
; The model defines these constraint templates
(constraints
  ((mult    km      =1000    h  ))
  ((add     surface h        r   ) (s* 0 sea-level))
  ((mult    r       r        r^2)                  )
  ((mult    G       Earth-M  K   ) (g* m* k*)      )
  ((minus   k       -k           ) (k* -k*)        )
  ((mult    d2r/dt2 r^2      -K  )                 )
  ((d/dt    r       dr/dt        ))
  ((d/dt    dr/dt   d2r/dt2      ))
  ((constant surface))
  ((constant G))
  ((constant Earth-M))
  ((constant k))
  ((constant -k))
  ((constant =1000))) )
```

Fig. 8. *Qualitative model of a second-order nonlinear system.* A list of `quantity-spaces` describes the qualitative values the various model variables can have, and a list of `constraints` describes the relationships among those model variables. This model describes an object in free fall in the Earth's gravitational field, such as a rocket or other projectile fired upward or an object falling downward. Gravity decreases with distance according to the standard nonlinear second-order differential equation $d^2r/dt^2 = -GM/r^2$.

To minimize the potential complexity of numerical inferences on multiple behaviors, the first subgoal to pursue, given a tree of behaviors is

Subgoal 1: "*Prune as many behaviors as practicable.*"

Phase I of a simulation with Q3 is to get qualitative behaviors, each annotated with rough bounds obtained by constraint propagation throughout its constraint network, of

```
(def-quantitative-info
  (name initial-velocity-about-3000)
  (quantitative-initializations
    ;gravitational constant
    (G       (G*         (6.67e-11     6.67e-11)))
    ;Earth's mass
    (Earth-M (M*         (5.98e24      5.98e24 )))
    ;radius of Earth
    (r       (sea-level  (6.37e6       6.37e6  )))
    ;Initial condition, less than escape velocity
    (dr/dt   (r*         (3000         3300    ))))
  (envelopes ()))
```

Fig. 9. Quantitative data describing known facts about the Earth, as well as the incompletely specified initial velocity dr/dt of a rocket (Fig. 8) fired upward from the Earth's surface.

the given quantitative information (Section 2.1). At this point, both return to Earth and the two escape behaviors[5] appear plausible. Fig. 6(a) showed what is known at this stage about the return behavior and is summarized in Table 1, row (a). The simulation has not yet ruled out the escape behaviors, though escape is in fact impossible for the given initial velocity of [3000, 3300] m/s.

Proceeding to Phase II, step size refinement can be applied because of the gap from 0 to 305 between $T0 \in [0,0]$ and $T1 \in [305, \infty)$ for the return behavior (Fig. 6(a) and Table 1, row (a)). There are also gaps between T0 and T1 for the escape behaviors, which will be refuted later (Table 1 last column and Table 2). Step size refinement interpolated one new state between T0 and T1 for each behavior. For the return behavior this was at time 153. For the escape behaviors, which have $T1 = \infty$, it occurred at time 1000 (Section 4.2.1). Constraint propagation on the resulting constraint network for each behavior pruned the escape behaviors and improved the characterization of the return behavior somewhat. The pruning of an escape behavior is described in Table 2, and the improved characterization of the return behavior was described in Table 1, row (b).

Subgoal 1 has been fully satisfied but the overall Goal is still only partially satisfied, because we still know little about how high the rocket goes or how long it takes to return. Thus we wish to narrow existing intervals, and infer new intervals for values of model variable at more time points in the return behavior. This requires satisfying two additional subgoals:

Subgoal 2: "*Infer the system trajectory between* T0 *and* T1."

and

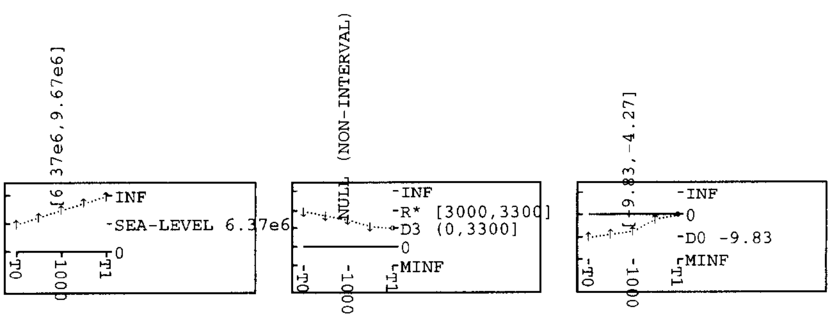
Subgoal 3: "*Infer the system trajectory between* T1 *and* T2."

[5] In one escape behavior the velocity of the rocket decreases asymptotically to zero, and in the other to a positive value.

Table 2
(a) Excerpt of a trace showing how constraint propagation eliminated a behavior of the rocket, after a state was interpolated at *time* = 1000. This trace occurred for both escape behaviors. (b) A plot for a pruned escape behavior. Note the inconsistency detected in velocity at time 1000.

Constraint	*Inference*		*Reason*
Initial conditions	R_{T0} = SEA-LEVEL = 6.37e6 meters $(DR/DT)_{T0} \in [3000, 3300]$ m/s	(i) (ii)	Given
Previously inferred	-K* = −3.99e14 $D2R/DT2)_{T0} = -9.83\,m/s^2$	(iii) (iv)	From G* and M* in Phase I. From $(R\hat{}2)_{T0}$ and -K in Phase I.
Mean value *RATE* variable: DR/DT (velocity, m/s upward)	*RATE*(T0) $\underline{\cup}$ *RATE*(1000) $= (-\infty, \underline{3300}]$; Decrease $\overline{R_{1000}}$; $R_{1000} \in (-\infty, 9.67e6]$	(v)	Qualitative behavior has DR/DT decreasing; use that and (ii). Solve Eq. (4) for *LEVEL*(T_n) using (i), (ii), and (v).
LEVEL variable: R (radius, meters from the Earth's center) *TIME*: T_{n-1} = T0 = 0, T_n = 1000	Increase $\underline{(DR/DT)_{1000}}$; $(DR/DT)_{1000} \in (0, \infty)$ Increase $\underline{R_{1000}}$; $R_{1000} \in \overline{(6.37e6}, 9.67e6]$	(vi) (vii)	Qualitative behavior has $R_{1000} > R_{T0}$, so Eq. (4) implies that DR/DT is positive. Solve Eq. (4) for *LEVEL*(T_n) using (i), (ii), and (vi).
Multiplication R × R = R^2, so $R_{1000} \times R_{1000} = (R\hat{}2)_{1000}$	Increase $\underline{(R\hat{}2)_{1000}}$; Decrease $\overline{(R\hat{}2)_{1000}}$; $(R\hat{}2)_{1000} \in [4.06e13, 9.35e13]$	(viii)	Square of R_{1000} from (vii).
Multiplication (D2R/DT2) × (R^2) = -K, so $(D2R/DT2)_{1000} \times (R\hat{}2)_{1000}$ $= -K_{1000}$ = -K*	Increase $\underline{(D2R/DT2)_{1000}}$; Decrease $\overline{(D2R/DT2)_{1000}}$; $(D2R/DT2)_{1000} \in [-9.83, -4.27]$	(ix)	Divide -K*, from (iii), by $(R\hat{}2)_{1000}$, from (viii).
Mean value *RATE* variable: D2R/DT2 (acceleration of gravity, m/s^2) *LEVEL* variable: DR/DT *TIME*: T_{n-1} = T0 = 0, T_n = 1000	Prune behavior; $(DR/DT)_{1000} \in (0, \infty)$ and $(DR/DT)_{1000} \in [-6830, -965.5]$ have a null intersection; inconsistency detected.		(ii), (iv), (vi), and (ix) are inconsistent with Eq. (4).

(a) Trace ↗



(b) Plot ↗

Radius=r(t) Velocity=dr/dt(t) Acceleration=d2r/dt2(t)

Step size refinement can be applied between T0 and T1 to address Subgoal 2. However step size refinement cannot yet be applied between T1 $\in [316, \infty)$ and T2 $\in [316, \infty)$ to address Subgoal 3 because there is no gap between T1 and T2. Addressing Subgoal 3 thus requires first satisfying a subsidiary subgoal:

Subgoal 3a: "*Create a gap between* T1 *and* T2."

It is possible that inferences arising from addressing Subgoal 2 will result in a gap between T1 and T2, satisfying Subgoal 3a as well. Thus step size refinement is applied between T0 and T1 to satisfy Subgoal 2 and perhaps Subgoal 3a.

States were interpolated between T0 and $\underline{\text{T1}}$ three times. The simulation at that point is summarized in Table 1, row (c). Observe that little progress has been made toward creating a gap between T1 and T2. To get this gap it is necessary to infer a decrease in $\overline{\text{T1}}$ sufficient to produce a gap. Q3 provides two auxiliary techniques for creating gaps: target interval splitting (Appendix A), and behavior splitting. Target interval splitting should always be tried before behavior splitting because behavior splitting creates new branches in the behavior tree and hence can lead to high computational complexity in subsequent simulation refinement. Thus, target interval splitting ("TIS") is used next, to try raising $\underline{\text{T2}}$ and lowering $\overline{\text{T1}}$.

Starting with the knowledge that T2 $\in [326, \infty]$, Q3's implementation of target interval splitting assumed T2 $\in [326, 434]$, let the simulation settle via constraint propagation as usual, and discovered that settling led to an inconsistency (Table 2 exemplifies detecting an inconsistency). Therefore T2 $\notin [326, 434]$, so $[326, 434]$ was trimmed from the interval for T2, giving T2 $\in [434, \infty]$. (Before successfully ruling out $[326, 434]$, target interval splitting unsuccessfully tried to rule out the larger $[326, 651]$. Subsequent to ruling out $[326, 434]$, target interval splitting unsuccessfully tried ruling out the adjacent interval $[434, 579]$, successfully ruled out $[434, 482]$, unsuccessfully tried ruling out $[482, 536]$, successfully ruled out $[482, 500]$, then $[500, 519]$, and finally unsuccessfully tried ruling out $[519, 525]$. Target interval splitting was not able to reduce $\overline{\text{T1}}$ from ∞ although it tried ruling out a sub-interval $[x, \infty)$ with x very large. More details on target interval splitting appear in Appendix A.)

The results at this stage are summarized by Table 1, row (d). Observe that there is still no gap between T1 and T2, because target interval splitting did not reduce $\overline{\text{T1}}$ from ∞. Nevertheless, now $\underline{\text{T2}} > \underline{\text{T1}}$, so significant progress has been made in creating the desired gap. Since the choice now is between step size refinement, Q3's main technique, and behavior splitting, Q3's least favored auxiliary technique, step size refinement is invoked at this point.

Three more states were interpolated via step size refinement, summarized by Table 1, row (e). Observe that while $\underline{\text{T2}}$ is rising, $\overline{\text{T1}}$ is not falling, which it must if a gap is to be created. Therefore behavior splitting, the remaining technique available to Q3, should be invoked.

Behavior splitting involves copying a behavior to produce a pair of independently represented, qualitatively identical behaviors, and replacing in each an interval that is to be split by a separate sub-interval of it. Each behavior is subsequently processed independently. In this example, T1 $\in [331, \infty)$ was split into the separate sub-intervals

$[331, 10^6]$ and $[10^6, \infty)$, here 10^6 was an arbitrary high number providing a high but finite bound to one of the sub-intervals. (High but finite bounds are significantly more useful than infinite bounds because infinite values inhibit inferences, since the magnitude of an infinite bound is not reduced by subtracting or dividing it by any real number.) Each sub-interval is associated with T1 in one of the otherwise identical copies of the original behavior. For each copy, the effects of the new sub-interval for T1 propagate throughout the copy's constraint network representation (Table 1, row (f)).

After interpolating just one more state in each behavior, the behavior for which $T1 \in [10^6, \infty]$ is refuted, and the one for which $T1 \in [331, 10^6]$ now has a gap between T1 and T2 (Table 1, row (g)). The new gap satisfies Subgoal 3a, enabling step size refinement in support of Subgoal 3. There is now nothing to prevent step size refinement from continuing to refine the quality of the simulation for the system's entire trajectory.

After each new interpolation, better numerical bounds are inferred. After a total of twenty-five states have been interpolated, the results are summarized by Table 1, row (h) (and were shown in more detail back in Fig. 6(b)). Further step size refinement causes further incremental improvement; row (i) summarizes the simulation after a total of 50 interpolations.

4. Correctness, convergence, stability, and termination

Correctness here means that each interval describing a trajectory bounds the range of values that could be exhibited by any actual system conforming to the model and its initial conditions. *Convergence* means that with continued step size refinement, the inferred intervals will become narrower, approaching point values in the limit if the model is specified with real valued initial conditions and model parameters. When the model is specified imprecisely with one or more intervals we are interested in *stability*, which intuitively means that if system specifications are weakened, the widths of result intervals will be wider but only to a limited degree. We first discuss correctness, followed by convergence, stability and finally termination for step size refinement.

4.1. Correctness

Numerical methods estimate answers, and interval methods bound them. Correctness here implies that the bounds safely contain the space of possible answers (e.g. [37, p. 65]). Bounds may also include extraneous values, which may occur for the following reasons.

- *Excess width.* This is a well-known problem in evaluating many interval expressions (see [66], [1, pp. 84–88] and [78, Section 2.2.4]). The simplest such expression is $(X - X)$. Its value is obviously 0, but naive evaluation can give a weaker answer. For example, given $X \in [1,2]$ then straightforward calculation gives $X - X =$

[1,2] − [1,2] = [−1,1]. Excess width can occur in evaluating some expressions containing subtractions or divisions in which an interval valued symbol appears more than once. Eliminating excess width in the general case is non-trivial. Existing algorithms are either limited to linear problems and rather convoluted [56] or computationally complex [4], although progress on optimizing an apparent tradeoff between computational and conceptual complexity is reported by Cornelius and Lohner [23] and Hyvönen [49].

- *Impossible values between possible values.* Values in the middle of an interval may be impossible while values nearer to the endpoints are possible. This problem could be solved by allowing value description as disjoint sets of intervals, rather than a single interval. This strategy is central in Hyvönen [49].

Q3 uses constraint propagation on interval labels, which is correct because no interval will be narrowed too much [27]. However, a full accounting of correctness in Q3 also requires that the imprecise nature of machine arithmetic does not introduce incorrectness through round-off error.

4.1.1. Machine round-off error

Inaccurate arithmetic can obviously impact correctness. The finite precision of floating point calculations often introduces inaccuracy, called *round-off error*. For example, while 1/2 can be represented precisely in floating point format, 1/3 cannot. One solution is to use a language such as Pascal-SC [11] which supports interval operations that are correct (have guaranteed inclusion) despite round-off error. This is achieved by rounding low bounds downward and high bounds upward. COMMON LISP offers another solution by supporting rational arithmetic, which is completely accurate, but both slow and not closed under common transcendental functions. Q3 is written in COMMON LISP and will work with rationals if a switch is set, but defaults to *rounded interval arithmetic* [66, p. 15], which increments the high bound of each calculated interval by a small proportion of its value and decrements the low bound analogously. Provided this proportion is large enough to compensate or overcompensate for any inaccuracies introduced by round-off error, inclusion and hence correctness are maintained.

4.2. Convergence

For numerical simulation, convergence means improving point predictions all the way to full accuracy [46]. For interval simulations, convergence means narrowing interval enclosures all the way to correct point predictions (see [37, p. 57], [66, pp. 96-97] and [60, p. 261]). Both senses apply in the limit as the step size of the simulation approaches zero.

As the step size decreases, the total number of steps increases. The computational complexity of simulations containing a large number of steps, together with round-off error intrinsic to floating point arithmetic or the compensating extra width added intentionally in rounded interval arithmetic, restricts convergence in practice. Nevertheless, the concept is a central tool in validating simulation algorithms.

Our analysis builds on traditional analyses of convergence of numerical simulation methods such as Euler's method (see [46]; also see basic texts such as [31]). Theorem B.2 (Appendix B) states:

Let $Y' = F(Y)$ be a system of first-order differential equations,[6] *where F is a vector of interval valued functions of Y. We consider some bounded subset $[lo, hi]$ of the reals such that for each component $Y_{(j)}$ of vector Y, $Y_{(j)}(t) \subseteq [lo, hi]$.*[7] *We assume that $F(Y)$ is defined when each $Y_{(j)} \subseteq [lo, hi]$, and that each F_i in vector F is the natural interval extension of a real rational function $f_i(y)$.*[8]

Let h be the maximum step size, let $\|Y_{t=b}\|$ represent the amount of uncertainty in the simulated estimate of Y at interpolated time point $t = b$ as measured by its vector norm,[9] *and let $\|Y_0\|$ represent the amount of uncertainty in the initial conditions. Then there are constants K_1 and K_2 such that*

$$\|Y_{t=b}\| \leqslant K_1 \|Y_0\| + K_2 h. \tag{5}$$

Given precise initial conditions, $\|Y_0\| = 0$. Then for any fixed time $t = b$, Eq. (5) implies that $\|Y_n\| \to 0$ as $h \to 0$. This constitutes *convergence*, and assumes that the maximum step size can be reduced arbitrarily close to zero. In Q3, satisfying this assumption requires having a gap starting at time $t = 0$. A gap may have been created by qualitative simulation plus constraint propagation of quantitative information (Phase I of simulation refinement). If not, it will need to be created (Section 4.3.1). Within the gap, there is nothing to prevent continued interpolation, thus allowing convergence within that region.

This convergence result may be generalized to gaps starting at arbitrary times by observing that any state that is fully specified (e.g. by a measurement vector) can be considered an initial state with precise initial conditions.

Example. Fig. 10 illustrates convergence for a system which plots x versus time according to

$$\frac{dx}{dt} = \frac{x+1}{x}.$$

The predictions become much narrower as additional times are interpolated, and would result in full convergence given an infinitesimal step size and perfect machine arithmetic. Note that this particular system suffers from excess width. For example, as Davis [27] points out,

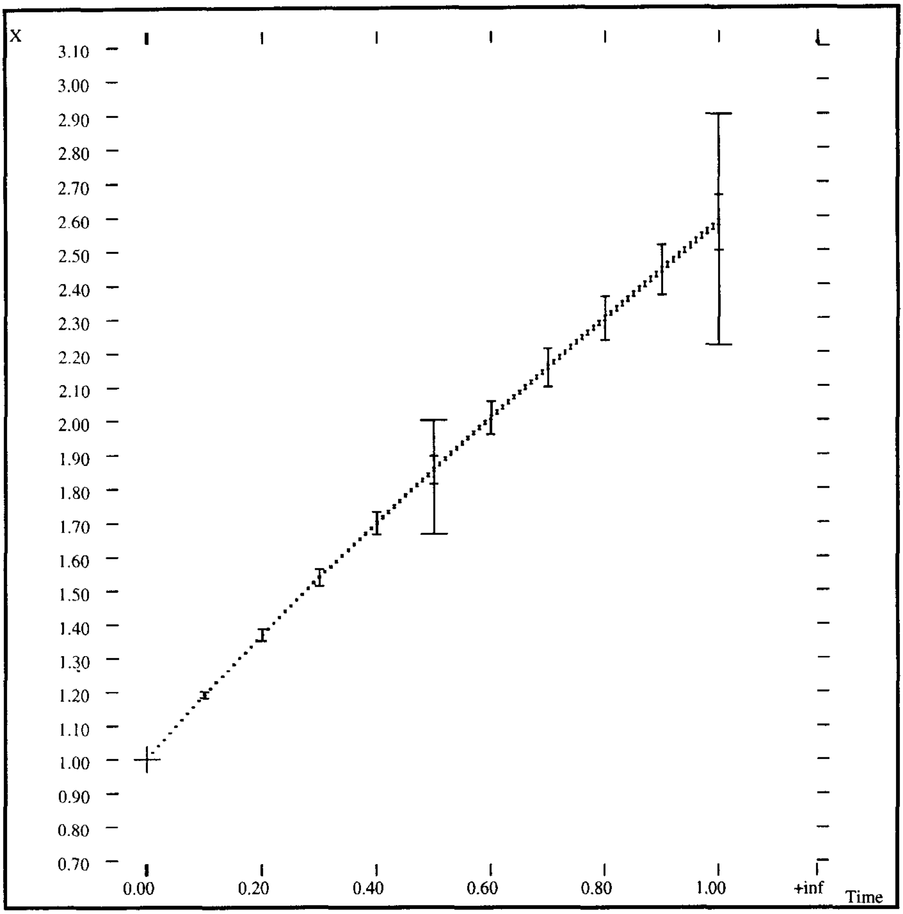
$$x \in [1, 2] \quad \text{implies} \quad \frac{x+1}{x} \in \left[\frac{3}{2}, 2\right]$$

[6] However, Q3 uses constraint network representations. Appendix B discusses equivalence of equations and constraint networks.

[7] $Y \subseteq [lo, hi]$ is not a major restriction because physical variables in real systems remain finite. However, in the case of a model where a variable diverges to infinity, the theorem only applies within a bounded region.

[8] Common notation is to use upper case for interval valued variables and functions and lower case for real valued variables and functions.

[9] Vector norm here means the width of the widest interval in the vector.



$$\frac{\mathrm{d}x}{\mathrm{d}t} = \frac{x+1}{x}, \quad x(0) = 1$$

Fig. 10. Example of convergence even though the interval calculations produce excess width. A simulation for $\mathrm{d}x/\mathrm{d}t = x + 1/x$ reveals a slightly concave down curve. Before step size refinement there are time points at 0 and ∞. After refining the simulation by interpolating two new states at time values $t = 0.50$ and $t = 1.00$, uncertainty in x increases rapidly, as shown by the two tall interval delimiters at $t = 0.50$ and $t = 1.00$. After refinement with ten interpolations, the trajectory is significantly more constrained, as shown by the ten interval delimiters of intermediate height at times 0.10, 0.20, etc. Refinement with 100 interpolations leads to much better results as shown by the 100 much shorter interval delimiters.

but straightforward calculation (e.g. by hand or in Q3), gives

$$\frac{x+1}{x} \in \frac{[1,2]+1}{[1,2]} = \frac{[2,3]}{[1,2]} = \left[\frac{2}{2}, \frac{3}{1}\right] = [1,3].$$

Thus this example demonstrates convergence despite excess width.

4.2.1. The infinitesimal step size assumption

Convergence as a theoretical property (both in numerical simulation and in the present case) assumes that the step size can be made infinitesimally small. We discuss this issue in the bullets below.

- If interpolation of each new time point can be done so as to reduce the size of the largest step S in the region in which convergence is desired, then continued interpolation will lead to a strictly monotonic decrease in the maximum step size in the region of convergence.
  Example: If the region of convergence is $[0,1]$ successive interpolation points of 0.5, 0.75, and 0.875 would not be allowed because a time point must be interpolated in the gap $(0,0.5)$ before the gap $(0.75,1)$ if a strictly monotonic decrease in maximum step size is to be achieved.
- The decrease in maximum step size within the region of convergence should not only be strictly monotonic, but also an interpolated time point should divide the enclosing step into two smaller steps such that the width of each is smaller than $P(\mathit{width}(S))$, where P is some predefined constant in $(0.5,1)$.
  Example: If as before the region of convergence is $[0,1]$, successive interpolation points of $0.1, 0.11, 0.111, \ldots$ would lead to strictly decreasing maximum step size but not to convergence.
- If there is no gap, as might occur when initial conditions are weak, step size refinement can be run only after a gap is created. This may be done using the techniques of Section 4.3.1.
- The region of convergence starts at $\mathit{time} = 0$ but may have a finite width, if the upper bound of the gap is finite.
  Example: In Fig. 5, the region of convergence is some finite value $\mathit{time} > 3671$ because while the low bound of T1 is 3671 (shown below the plots) it rises with continued interpolation, because the interpolated states lead to better knowledge of T1's possible values.
- In a few cases, as when $\mathtt{T1} = \infty$, the gap has infinite width. In such cases, the value of the first interpolated time point can be any number. This first interpolated point will divide the gap into a finite region starting at 0, and an infinite region. Subsequent interpolations can be in either the finite or the infinite region.
  Example: The behavior shown in Fig. 10 has qualitative time points $\mathtt{T0} = 0$ and $\mathtt{T1} = \infty$. The first interpolated time point was at 1.0, which therefore divided the initial time step between T0 and T1 into two steps, one with $\mathit{width} = 1.0$, and the other with $\mathit{width} = \infty$. In Fig. 10 further interpolations were in the region $(0,1)$. However, any time point $t_k > 1$ could be interpolated later, increasing the region of convergence to t_k.

While convergence is universally recognized as an important theoretical property of simulation methods for continuous systems, it should be noted that pragmatically oriented uses of time point interpolation have not had convergence as a goal (see [36,52] and Section 5—especially Section 5.3.2—of this paper). Pragmatically oriented work shows that even one interpolation can lead to significantly improved quantitative bounds on model trajectories (Berleant [5] provides a simple, detailed example).

4.3. Stability

In numerical simulation stability is, intuitively, the desirable characteristic that "...a change in the starting values by a fixed amount produces a bounded change in the numerical solution..." given a well-posed problem and sufficiently small step sizes [46, p.9]).

Gear [46, p.56] defines stability more formally as

$$\|y_n - \tilde{y}_n\| \leqslant K\|y_0 - \tilde{y}_0\|, \tag{6}$$

where y_0 and $\tilde{y}_0$ are two sets of initial conditions, y_n and $\tilde{y}_n$ are the corresponding results of numerical simulation after n steps with a one-step method, $\|\ \|$ is the *norm* operator which here is a vector generalization of absolute value, the numerical simulation is of a set of Lipschitz differential equations containing the variables in vector y, and Eq. (6) holds for all step sizes in $[0, h_0]$ for some positive constants h_0 and K.

We adapt this notion of stability to interval simulation by replacing the idea of the *difference between two solutions* with the idea of a *single interval valued solution with a width*. Observe that while the difference of two solutions concept of stability provides no correctness guarantee, the interval approach does. [10]

Thus, a reasonable stability criterion for a correct interval valued simulator is:

$$\|Y_n\| \leqslant K\|Y_0\|.$$

We show that this stability property holds in the limit as $h \to 0$ (see Appendix B). Therefore, employing the concept of $h \to 0$ stability described by Henrici [48] and Young and Gregory [92] and named by Young [91] we have that, given interval initial conditions, step size refinement possesses $h \to 0$ stability. [11]

The pragmatic implications of this stability property are twofold.

(1) Simulation results benefit from step size refinement, even when initial conditions and model parameters are only incompletely specified via intervals. (Fig. 5 illustrated how significant inferences result after step size refinement reduces the maximum step size h sufficiently, even though initial velocity was only weakly specified.)

(2) More precise initial conditions lead to more precise predictions.

4.3.1. Gap existence and creation

While step size refinement is stable, convergent, and correct, it can only run within a gap. The most common and important case is a gap starting at T0 $\in [0, 0]$. In particular:

[10] For example, both numerical solutions could underestimate the value of some model variables leading to a genuine solution $y(t_n)$ not between numerical solutions y_n and $\tilde{y}_n$.

[11] This may seem like a weak stability criterion but is reasonable because, due to the $K_2 h$ term of Theorem B.2, the predicted intervals contain the *discretization error* of the simulation algorithm. The discretization error of commonly used one-step numerical simulation methods, such as Runge–Kutta, can be accounted for explicitly by representing the nominal predictions $\pm$ calculated error bounds, yielding intervals. Then such numerical methods possess no better than $h \to 0$ stability even though their real valued results have nominally stronger stability properties.

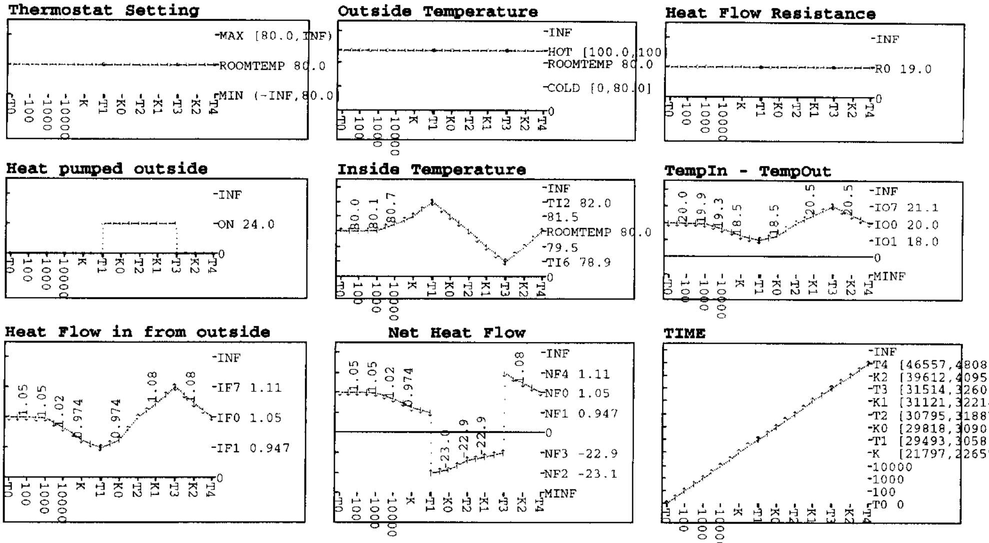


Fig. 11. Piecewise continuous simulation of an air conditioned dwelling. The time points in this simulation come from three sources: (1) qualitative simulation, which created time points T0, T1, T2, T3 and T4; (2) interpolations in the gap between T0 and T1, which created time points 100, 1000, and 10000; and (3) interpolations in gaps of model variable Inside Temperature, which created time points K, K0, K1 and K2 at temperatures 79.5 and 81.5. Discontinuities visible in some of the plots are caused by transitions between models.

- When the behavior has two qualitative time points $T0 = 0$ and $T1 = \infty$ the gap between T0 and T1 includes all positive finite values, allowing states to be interpolated at arbitrary times, so step size refinement is unimpeded.
- When $0 < \underline{T1} < \infty$, the first gap is the open interval $(0, \underline{T1})$, and step size refinement is unimpeded for time values in that interval. $\underline{T1}$ may also increase as the simulation becomes more refined, increasing the size of the gap. (This occurred in Table 1.)

While often the requisite gaps will exist prior to step size refinement due to propagation of intervals in Phase I of the progressive simulation refinement process (Section 2.1), sometimes they may not, due to weak initial conditions. Q3 provides ways to deal with lack of a gap.

- Use *target interval splitting*. See Appendix A and Section 3 for details.
- Use *behavior splitting* to force a place to interpolate. See Section 3.
- Use another time step that does have a gap.
- Interpolate using a gap in a model variable other than TIME that has a gap. *Example*: step size refinement using a gap in Inside Temperature instead of TIME occurs in Fig. 11.

4.3.2. Termination

Constraint propagation is guaranteed to terminate when the label sets containing candidate values have a finite number of elements [61]. However, in the case of intervals or other label sets containing an infinite number of elements, settling may be asymptotic and termination may not occur. In the case of floating point (not real) quantities, there are a finite but large number of them, and termination can be impractically slow unless measures are taken to speed it up. The measure taken by Q3 is to increase a lower interval bound or decrease an upper interval bound during constraint propagation only if the bound will change by a proportion of its value greater than some constant ε. This ensures termination because a bound can change by a factor of ε only a finite number of times before it must cross the other bound,[12] which if it happens means the qualitative behavior can be pruned[13] (as we have seen).

5. Applications

Techniques first developed in Q3 have been applied not only to improving simulation predictions but also to diagnosis, measurement interpretation, and bounding the probabilities of qualitative behaviors, as described next.

5.1. Improved predictions

By making quantitative inferences, semi-quantitative simulation can often prune qualitative behaviors that are plausible from a purely qualitative standpoint. A behavior is pruned when quantitative inference reveals that no interval is possible for some model variable at some time point (as we saw in Table 2). Dalle Molle [24] and Dalle Molle and Edgar [25] used phase I of Q3 (Q2) for this purpose with two models of chemical engineering systems, the relatively simple but useful difference of two parallel first-order chemical processes, and the less simple adiabatic continuous stirred tank reactor.

Farquhar and Brajnik [39] used phase I of Q3 in a system called SQPC ("Semi-Quantitative Physics Compiler"). They generated semi-quantitative models automatically and ran them. They were able to model and simulate a real hydroelectric dam, predicting power outputs and water levels for different water control scenarios.

5.2. Diagnosis

Semi-quantitative simulation can help diagnose which fault model explains observed faulty behavior. Models for which all behaviors are inconsistent with observation are ruled out, ideally leaving just one remaining fault model [57,59]. MIMIC in its more recent version [36] used time point interpolation to help diagnose fault models.

[12] Except with 0 or $\pm\infty$ bounds. When a bound is moving asymptotically toward zero, termination occurs from arithmetic underflow. If toward $\pm\infty$, termination occurs from overflow when the bound is changing geometrically, and from failure to change by a proportion greater than ε if the bound is changing linearly. We have experienced these termination cases only in examples designed specifically to create them.

[13] A similar approach was taken by Siskind and McAllester [82].

5.3. Measurement interpretation

The concept of interpolating a state extends naturally to measurement interpretation, because a measurement partially specifies a new state, which can often be interpolated. We illustrate the power of this concept with a familiar example, then briefly review some related work.

5.3.1. An illustrative example

Suppose the height of the rocket (Fig. 5) is measured to be within [12000, 12500] km at time $t = 3375$. Clearly any state whose time value is 3375 could be interpolated between time points 3200 and 3400. This interpolated state would be further defined with the measured value for height. That measured height could narrow the heights in the neighboring states, all the way from the previous [10833, 58745] km (Fig. 5) down to [10833, 12500] km for time 3200. This is because the maximum possible height at time 3200 is bounded from above by the measured height at $t = 3375$. The effects of the measurement are then propagated, leading to better predictions for various model variables at various time points.

Results of an experiment are summarized in Table 3.

5.3.2. Related work

MIMIC [36] does diagnosis by interpolating states containing the measured values, just as in the example above. The measured quantitative information is propagated and used to rule out alternative models. In MIMIC, the foundation of monitoring and diagnosis is measurement interpretation, and the foundation of measurement interpretation is interpolation and propagation.

Measurements in MIMIC and Q3 lead to new inferences for two reasons.

(1) Average step size is decreased.

(2) Interpolated measurements incorporate newly asserted quantitative information.

The interpolation method of measurement interpretation contrasts with DeCoste's DATMI system [29] and its precursor ATMI [42]. A significant difference is that DATMI abstracts measurements into qualitative categories before using them, whereas MIMIC and Q3 use the actual measured quantitative information. Hence DATMI loses quantitative information retained and used by MIMIC and Q3.

DATMI is intended for handling large numbers of measurements. The unmodified Q3 approach is unwieldy for large numbers of measurements, but can be modified to circumvent this shortcoming by propagating forward but not backward in time, and propagating forward only as far as needed. This was the approach taken by MIMIC.

5.4. Bounding the probabilities of qualitative behaviors

Qualitative simulation alone can find all possible behaviors of a system but not their probabilities. Adding quantitative information can help. Q3 was part of a system that inferred probabilities for the qualitative behaviors of a fault tolerant system [8]. Probability density functions (pdfs) were used instead of intervals to describe model input values. Pdfs are more informative than intervals. An interval represents a set of

Table 3
Effects of different measurement strengths on predictions for Velocity, Height, and Acceleration of the rocket, at time points 400, 3000, and T1. The intervals for the "no measurement" condition are the same as in Fig. 5. The effects of interpolating a state with a weakly constraining measurement condition are shown in the middle rows. A strong measurement condition is shown in the last rows. Notice how predicted intervals tend to narrow as stronger measurement conditions introduce stronger quantitative information into the simulation

Variable:	Velocity			Height			Acceleration		
Time point:	400	3000	T1	400	3000	T1	400	3000	T1
No measurement:	$[6770, 18867]$	$[906, 17870]$	$[0, 0]$	$[2950, 7862]$	$[10706, 55171]$	$[10924, \infty)$	$[-4.59, -1.97]$	$[-1.37, -0.105]$	$[-1.34, 0)$
Weak measurement: *Height* $\in [12000, 12500]$ at *time* = 3375	$[6770, 18463]$	$[906, 14295]$	$[0, 0]$	$[2950, 4558]$	$[10706, 12314]$	$[12025, \infty)$	$[-4.59, -3.34]$	$[-1.37, -1.14]$	$[-1.18, 0)$
Strong measurement: *Height* = 12000 at *time* = 3375	$[6770, 18293]$	$[906, 13844]$	$[0, 0]$	$[2950, 4058]$	$[10706, 11814]$	$[12025, \infty)$	$[-4.59, -3.67]$	$[-1.37, -1.2]$	$[-1.18, 0)$

pdfs containing all pdfs with heights of zero beyond the interval endpoints, hence is a weaker description of value.

The pdfs were first discretized using histograms. Note that each bar of a histogram spans an interval. Thus problems represented using pdfs are decomposed into subproblems represented using intervals and solvable using Q3. While discretization often leads to approximation, in our approach the discretization leads instead to inferring probabilities of behaviors expressed as ranges within which the actual probabilities of the respective behaviors must reside [6,8].

6. Other related work

Considerable work related to semi-quantitative simulation has been reported in addition to the works discussed in foregoing sections, including spatial reasoning [18,21,62, 80], temporal ordering (e.g. [3,22,28,51,63,87]), interval reasoning (e.g. [2,27,37, 49,62,66,69,88,89]), digital circuits (e.g. [19,20,47,74,83]), phase space mapping (e.g. [54,67,72,73,90,93]), discrete event simulation [65,77,79–81], and difference equations [55].

A review of the aforementioned work is left to the interested reader. Here we review domain independent work that addresses the general problem of increasing the power of qualitative simulation with intervals, numbers, or fuzzy values.

6.1. Interval work

One of the earliest works in qualitative reasoning was de Kleer [30], which recognized the advantages of using intervals to represent incomplete quantitative information, and like the present work chose intervals for that purpose [30, pp. 76–77].

ATMI [42] and DATMI [29] used intervals describing measurements to find the qualitative path of an evolving system, as discussed earlier.

NSIM [53] and SQSIM [52] were developed in part to alleviate the wide bounds that Q3's predecessor Q2 often infers. While NSIM sometimes provides better bounds than Q2 [53,57], sometimes NSIM's results are poorer than Q2's, a result which led to SQSIM which combines features of both NSIM and Q2. Kay [52] describes SQSIM in detail but no comparison of its inferential ability to that of Q3 exists.

Vescovi, Farquhar and Iwasaki [86] describe a method of numerical simulation using intervals instead of numbers. Their approach, like other interval simulators (e.g. [37, 60]) does not distinguish among different qualitative behaviors. However they were able to solve a complex practical problem involving process monitoring of sintering at a steel plant.

6.2. Numerical work

Forbus and Falkenhainer [43,44] combined numerical and qualitative simulations in the SIMGEN (SIMulator GENeration) system, building on qualitative process theory [41]. SIMGEN displays notable advantages.

(1) Use of qualitatively inferred model transitions (e.g. when water temperature ceases rising and boiling commences) enabling automating simulations beyond that of ordinary numerical simulations.
(2) Causal ordering applied to qualitative models to enable automatic explanation generation.

Limitations of SIMGEN include (1) the requirement for a comprehensive domain model and (2) the need for precise numerical information, which like ordinary numerical simulation results in approximate outputs and often unsupported precision in specifications of initial conditions.

While SIMGEN used qualitative simulation to control numerical simulation, Bonarini and Maniezzo [13] used numerical simulation to control qualitative simulation. They pruned qualitative behaviors as they became inconsistent with a numerical simulation by matching an evolving qualitative simulation against an evolving numerical simulation.

Bonarini and Maniezzo's system, in contrast with SIMGEN, has the advantage of not requiring comprehensive domain models, but the disadvantage of not addressing sophisticated model switching and explanation. (Q3 like QSIM addresses model switching though not as comprehensively as SIMGEN, Fig. 11 showing a typical example.)

6.3. Fuzzy mathematics work

Extension of qualitative simulation with fuzzy mathematics was first published by D'Ambrosio [26], further discussed by Nordvik, Smets and Magrez [68], and developed and fully implemented by Shen and Leitch [75,76]. Shen and Leitch used trapezoidally shaped fuzzy intervals (Fig. 12) with arithmetic operations as defined by Tong and Bonissone [84], Bonissone [14], Bonissone and Decker [15], and DiCesare, Sahnoun and Bonissone [32]. DiCesare et al. [32] claim without explanation that these operations are consistent with Dubois and Prade's [34] more general and rigorously developed account. If so, their multiplication (and hence division) operation is more closely related to Dubois and Prade's relatively easily computed approximation [34, Eq. 3] than to the exact method [34, p. 620]. These approximations are claimed to produce "very good" results [15, p. 230] with "very little error" [14]. Exact multiplication of trapezoidal fuzzy intervals is computationally more complex, typically yielding a product with curved sides [50, Fig. 1.12] which is therefore not trapezoidal. Since computationally practical multiplication and division of fuzzy intervals requires approximate formulae, the quality of such formulae is critical.

6.3.1. As with standard intervals, operations on fuzzy intervals can produce excess width

Fig. 12 contains a very simple example of how the excess width problem in calculations on intervals has similar manifestations in calculations on fuzzy intervals. Values of x in the interval $[104, 106]$ are full members of fuzzy interval Z, and those in the sloping areas are possible members. Subtraction would give the region of full membership in the difference $Z - Z$ as $[104, 106] - [104, 106] = [-2, 2]$, the region of non-zero membership as $[102, 107] - [102, 107] = [-5, 5]$, and fuzzy edges of constant slope. However, Z is perfectly correlated with itself, so $Z - Z$ actually has full membership at

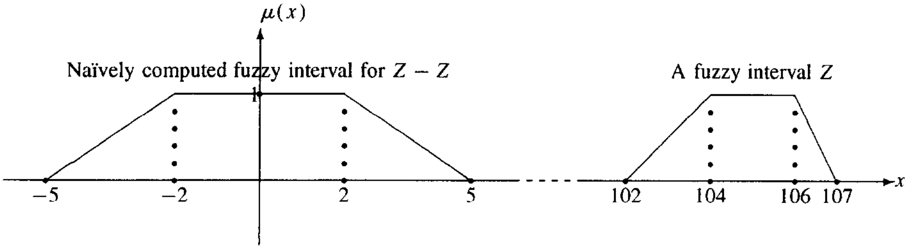


Fig. 12. Fuzzy intervals. Sloping line segments indicate fuzzy regions. The lower the value of membership function $\mu(x)$, the less the degree of membership for x in the fuzzy interval.

0 and zero membership everywhere else. While this particular example is trivial, such situations can be arbitrarily complex, just as with ordinary intervals.

Considering $\mu(x)$ as an upper bound on the membership, rather than the actual membership, addresses the membership over-estimation problem. This provides a sensible interpretation for results containing excess width, such as $Z - Z$ in Fig. 12.

Correlated fuzzy simulation [40] eliminates the excess width problem, but at the cost of assuming all operands are fully correlated. Correlated fuzzy simulation reduces to a special case of Monte Carlo simulation.

Fuzzy *values* generalize fuzzy intervals, and have also been suggested for qualitative reasoning [33]. The excess width problem in fuzzy and non-fuzzy interval calculations is a special case of membership over-estimation, which can occur in constraint propagation of fuzzy values [33, Eq. 6].

7. Conclusion

We have presented a semi-quantitative approach to simulation based *step size refinement*, an implementation, Q3, and work by ourselves and others employing that technique. The implementation provides much better predictions than its subset and predecessor Q2, by employing strengths of both qualitative and interval reasoning algorithms such as the following.

- *From qualitative simulation*: the guarantee that all qualitative behaviors will be found.
- *From interval simulation*: the guarantee that the trajectory of any real system conforming to an incompletely specified model is enclosed by one of the predicted semi-quantitative behavior descriptions.
- *From interval simulation*: $h \to 0$ stability.
- *From interval simulation*: convergence as uncertainty in the quantitative specifications, and maximum step size, both approach zero.
- *From qualitative and interval representations*: the ability to express and make predictions from partial knowledge.

The capabilities of Q3 rely mostly on the following.

- *Step size refinement*, for adaptive reduction in step size by introducing newly explicit intermediate time points into a predicted behavior.
- *Propagation of interval labels* in constraint network representations of behaviors.

Examples of graphical output from Q3 were provided and varied applications were reviewed, involving the domains of prediction, diagnosis, monitoring, measurement interpretation, and probabilities of qualitative behaviors.

The significance of Q3 to qualitative reasoning is both pragmatic and theoretical. Pragmatic, because Q3 demonstrates an effective method of obtaining better quantitative bounds on semi-quantitative simulation trajectories, step size refinement, which often leads to significant improvement in quantitative inferences after interpolating only one state. Theoretical, because step size refinement has the important theoretical guarantees of (1) convergence, (2) stability, and (3) correctness.

Acknowledgments

This paper is dedicated to Herbert Kay, Ph.D., 1959-1997.

Many useful comments and discussions contributed, directly and indirectly, to this work. Jo Decuyper, Keith Downing, Akira Hayashi, Herbert Kay, Raman Rajagopalan, Claire Rollwage, and anonymous referees are only a partial list. Several other people critiqued small yet important sections. We are grateful to all of these individuals.

Appendix A. Brief overview of Target Interval Splitting (TIS)

Target interval splitting, or TIS [7], narrows a *target* interval by ruling out subintervals of it. The method is described by the example of Figs. A.1 and A.2. A related technique called `divide-and-conquer-force` is claimed (albeit with insufficient support) to allow inferential "completeness" [82].

Appendix B. Proof of convergence and stability for step size refinement

This proof takes a system described as a set of first-order differential equations, which is the standard form for convergence proofs [46]. Q3, however, utilizes a constraint network. Fortunately, a network of arithmetic constraints is easily transformed into a set of equations [49, p. 81]. Each binary constraint `(op a b)` may be expressed as an equation $\mathtt{op}(a) = b$ and each ternary constraint `(op a b c)` as an equation $a \,\mathrm{op}\, b = c$. Constraints over more than three quantities are expressed as a longer equation (as in the mean value constraint, Section 2.1.3).

Transformation in the other direction, from equations to constraint networks, is also easily done (see [49, p. 76]). For example, the equation $y' = x'(y + x')$ becomes the constraints `(add y dxdt w)`, `(mult dxdt w dydt)`, `(d/dt x dxdt)`, and `(d/dt y dydt)`. Unary operations are expressed as binary constraints and binary operations as ternary constraints. Thus equations and networks of algebraic constraints can be interchangeable.

Target Interval Splitting (TIS): Outline

Given: • $Y = X^2 - X$, and $X \in [0,1]$
Therefore: • $Y \in [-1,1]$ by constraint propagation (shown in this figure).
Objective: • Narrow Y (the target) further, by testing and ruling out pieces of its current interval as in Fig. A.2.

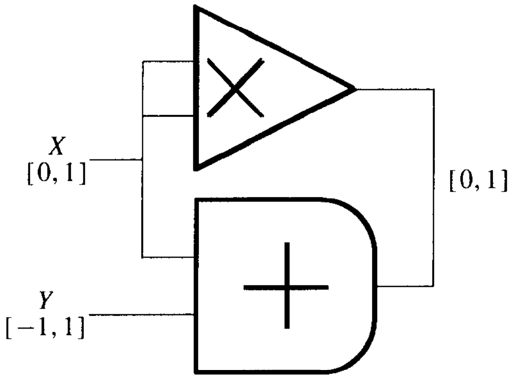


Fig. A.1. A constraint network for the equation $Y = X^2 - X$. Given $X \in [0,1]$, constraint propagation concludes $Y \in [-1,1]$. This conclusion is correct, but excessively weak, and is strengthened in Fig. A.2.

Lemma B.1 (Bounded uncertainty). *Let $Y' = F(Y)$ be a first-order differential equation, where F is an interval valued function of Y. We consider some bounded subset $[lo, hi]$ of the reals such that $Y(t) \subseteq [lo, hi]$.*[14] *We assume $F(Y)$ is defined when $Y \subseteq [lo, hi]$,*[15] *and $F(Y)$ is the natural interval extension*[16] *of a real rational function $f(y)$.*[17]

Let h be the step size, and let $w(Y_{t=b})$ be the width of the simulated estimate for Y at time point $t = b$. Let $w(Y_0)$ represent the width of the initial condition. Then there are constants K_1 and K_2 such that

$$w(Y_{t=b}) \leqslant K_1 w(Y_0) + K_2 h. \tag{A.1}$$

[14] See Footnote 7.

[15] In cases where F specifies division, there is the possibility of F being undefined in cases where the divisor is an interval having 0 as a member or endpoint. In such cases f does not satisfy the Lipschitz property over intervals containing zero, putting it outside the scope of this proof. Qualitative constraints deriving from QSIM quantity spaces will disambiguate the sign of a quantity so that division by intervals containing zero does not occur in practice. However, an interval divisor can still have 0 as an endpoint, in which case intervals with infinite endpoints can result, F may not be Lipschitz, and convergence is not guaranteed. Fortunately, such convergence problems can often be handled by behavior splitting, as exemplified by the rocket of Section 3, because a behavior split can always put an infinite endpoint in one behavior, usually making the other one Lipschitz.

[16] The term *natural interval extension* was defined in Section 2.1.3.

[17] See Footnote 8.

	TIS tests low bounds ...	
Iteration	Interval(s)	Reason
1	$Y \in [-1, 1]$	Constraint propagation
	$[-1, 0]$ $[0, 1]$	Split and test
2	$[-1, -0.5]$ $[-0.5, 1]$	Re-split and test
	X	Rule out
	$[-0.5, 1]$	Narrowed interval
3	$[-0.5, +0.25]$ $[+0.25, 1]$	Split and test
4	$[-0.5, -0.125]$ $[-0.125, 1]$	Re-split and test
5	$[-0.5, -0.3125]$ $[-0.3125, 1]$	Re-split and test
	X	Rule out
	$[-0.3125, 1]$	Narrowed interval
6	$[-0.3125, +0.3438]$ $[+0.3438, 1]$	Split and test
7	$[-0.3125, +0.0158]$ $[+0.0158, 1]$	Split and test
.	.	.
.	.	.
.	.	.
	$[-0.25 - \varepsilon_1, 1]$	Conclude
	... then high bounds	
N	$[-0.25 - \varepsilon_1, 1]$	Given
	$[-0.25 - \varepsilon_1, +0.375 - \varepsilon_1/2]$ $[+0.375 - \varepsilon_1/2, 1]$	Split and test
.	.	.
.	.	.
.	.	.
	$[-0.25 - \varepsilon_1, 0 + \varepsilon_2]$	Conclude

Fig. A.2. *Target interval splitting narrows a target interval by ruling out pieces of it.* The constraint network for $Y = X^2 - X$ was shown in Fig. A.1. Target interval splitting first tests the lower half of a target interval, $Y \in [-1, 1]$ in this example, by setting Y to $[-1, 0]$, then propagating. If the network settles successfully (i.e. has a solution), then it tests the lower quarter, $[-1, -0.5]$ in this case, the lower eighth if necessary, etc., until a sub-interval is found for which the network has no solution. That sub-interval is therefore inconsistent, and ruled out. In the example, the lowest quarter, $Y \in [-1, -0.5]$, was the first inconsistent sub-interval found. Repeat the process for the *highest* half, quarter, etc. of the target. Inconsistent sub-intervals are marked with an "X" above. For $Y = X^2 - X$, target interval splitting gradually narrows Y toward, yet never quite reaching, $[-0.25, 0]$. Termination is ensured by testing a sub-interval only if its width exceeds some ε.

Proof. The proof has similarities with standard proofs of Euler's method [31,46] and is also influenced by Moore [66]. We will abbreviate $Y_{t=(n-1)h}$ as Y_{n-1} and $Y_{t=nh}$ as Y_n.

(1) The inference method used by Q3 to propagate quantitative information from one time step to the next is the mean value constraint. This and the other constraints representing the states at times t_{n-1} and t_n are applied according to

$$Y_n := Y_n \cap (Y_{n-1} + h[F(Y_{n-1}) \underline{\cup} F(Y_n)]), \tag{A.2}$$

where ":=" signifies assignment and F is the natural interval extension of f. F is determined by the constraint model of the system of interest. Y_n has an initial interval value provided when the state containing it was interpolated (Section 2.2.2).

(2) Constraint propagation means Eq. (A.2) is applied iteratively, until a fixed point is reached with no further changes to Y_n. Iteration will eventually terminate (Section 4.1), with

$$Y_n = Y_{n-1} + h[F(Y_{n-1}) \underline{\cup} F(Y_n)]. \tag{A.3}$$

Y_n will be consistent with the direction of change of Y specified by some qualitative behavior(s) because Q3 uses QSIM for behavior generation, and QSIM generates all possible behaviors [58].

(3) Since F is a natural interval extension, F is *inclusion monotonic*, meaning $F(A) \subseteq F(B)$ if $A \subseteq B$ [66, Section 3.2]. We know that Y_n and Y_{n-1} are subsets of $(Y_{n-1} \underline{\cup} Y_n)$, so

$$\begin{array}{lrcl} \text{because} & F(Y_{n-1}) & \subseteq & F(Y_{n-1} \underline{\cup} Y_n) \\ \text{and} & F(Y_n) & \subseteq & F(Y_{n-1} \underline{\cup} Y_n), \\ \text{consequently} & F(Y_{n-1}) \underline{\cup} F(Y_n) & \subseteq & F(Y_{n-1} \underline{\cup} Y_n). \end{array}$$

From this and (A.3) we conclude

$$Y_n \subseteq Y_{n-1} + hF(Y_{n-1} \underline{\cup} Y_n). \tag{A.4}$$

(4) We now shift our concern from intervals to *widths* of intervals:

$$\begin{aligned} w(Y_n) &\leqslant w(Y_{n-1} + hF(Y_{n-1} \underline{\cup} Y_n)) \\ &\qquad \text{Width of an interval} \leqslant \text{width of a superset.} \\ &\leqslant w(Y_{n-1}) + hw(F(Y_{n-1} \underline{\cup} Y_n)) \\ &\qquad \text{Evaluating terms separately may lead to excess width.} \end{aligned} \tag{A.5}$$

(5) Since F is a natural interval extension defined for $Y_0 = Y(0)$, it satisfies the Lipschitz property for interval functions: [18]

$$w(F(Y_{n-1} \underline{\cup} Y_n)) \leqslant Lw(Y_{n-1} \underline{\cup} Y_n),$$

where L is the Lipschitz constant for F. Substituting into (A.5), we have

$$w(Y_n) \leqslant w(Y_{n-1}) + hLw(Y_{n-1} \underline{\cup} Y_n). \tag{A.6}$$

(6) We get Y_n out of the right-hand side as follows:

[18] A function f satisfies the Lipschitz condition if there is a constant L such that $|f(y_1) - f(y_2)| \leqslant L|y_1 - y_2|$ for all y_1, y_2 in the domain of f. This property extends straightforwardly to interval extensions of functions. $F(Y)$ will be Lipschitz if it is a real rational function defined for any $Y_0 \subseteq Y$ [66, p. 34, Definition and Lemma 4.1].

$$
\begin{array}{rrcl}
 & Y_i & \subseteq & M, \text{ all } i, \quad \text{where } M \equiv [lo, hi] \text{ (see lemma statement)} \\
\therefore & F(Y_i) & \subseteq & F(M) \quad \text{as } F \text{ is inclusion monotonic (this proof, step (3))} \\
\therefore & Y_n & \subseteq & Y_{n-1} + h[F(M) \underline{\cup} F(M)] \quad \text{(from Eq. (A.3))} \\
\therefore & Y_n & \subseteq & Y_{n-1} + hF(M).
\end{array}
$$

This and equation (A.6) justify

$$w(Y_n) \leqslant w(Y_{n-1}) + hLw(Y_{n-1} \underline{\cup} (Y_{n-1} + hF(M))).$$

(7) Since F is Lipschitz and a natural interval extension, $F(M)$ is bounded. Let $m = |F(M)|$, where the absolute value of an interval is the maximum of the absolute values of its endpoints. Then

$$
\begin{aligned}
w(Y_n) &\leqslant w(Y_{n-1}) + hLw(Y_{n-1} \underline{\cup} (Y_{n-1} + h[-m, m])) \\
&= w(Y_{n-1}) + hLw(Y_{n-1} + h[-m, m]) \\
&= w(Y_{n-1}) + hL(w(Y_{n-1}) + 2hm) \\
&= w(Y_{n-1}) + hLw(Y_{n-1}) + 2h^2Lm \\
&= (1 + hL)w(Y_{n-1}) + 2h^2Lm. \qquad \text{(A.7)}
\end{aligned}
$$

(8) Eq. (A.7) describes the width $w(Y_n)$ for the “worst” case, that is, the widest possible Y’s, and is a first-order linear difference equation which can be solved by applying a standard formula [31, Section 2.6]:

$$
\begin{aligned}
w(Y_n) &= w(Y_0)\left(\prod_{k=0}^{n-1}(1 + hL)\right) + \sum_{k=0}^{n-1}\left(\prod_{j=k+1}^{n-1}(1 + hL)\right)2h^2Lm \\
&= w(Y_0)(1 + hL)^n \\
&\quad + 2h^2Lm[(1 + hL)^{n-1} + (1 + hL)^{n-2} + \cdots + (1 + hL)^0].
\end{aligned}
$$

We eliminate intermediate powers of $(1 + hL)$ by multiplying both sides by $(1 + hL)$ and then subtracting.

$$
\begin{array}{rl}
(1 + hL)w(Y_n) &= w(Y_0)(1 + hL)^{n+1} + 2h^2Lm((1 + hL)^n + \cdots + (1 + hL)^1) \\
-w(Y_n) &= -w(Y_0)(1 + hL)^n - 2h^2Lm((1 + hL)^{n-1} + \cdots + (1 + hL)^0) \\
\hline
hLw(Y_n) &= w(Y_0)hL(1 + hL)^n + 2h^2Lm((1 + hL)^n - 1)
\end{array}
$$

(9) From elementary calculus texts, $e^{hL} = 1 + hL + (hL)^2/2! + (hL)^3/3! + \cdots$. Hence $e^{hL} \geqslant 1 + hL$, so

$$
\begin{aligned}
hLw(Y_n) &\leqslant hL(e^{hL})^n w(Y_0) + 2h^2Lm((e^{hL})^n - 1) \\
&= hLe^{nhL}w(Y_0) + 2h^2Lm(e^{nhL} - 1).
\end{aligned}
$$

(10) Recall that Y_n is the interval calculated to contain y at time $t = nh$, and let b be time nh. Then

$$hLw(Y_n) \leqslant hLe^{bL}w(Y_0) + 2h^2Lm(e^{bL} - 1),$$
$$w(Y_n) \leqslant e^{bL}w(Y_0) + 2hm(e^{bL} - 1).$$

(11) Let $K_1 = e^{bL}$ and $K_2 = 2m(e^{bL} - 1)$. For any *fixed* time $b = nh$, K_1 and K_2 are constant regardless of changes to the value of h, since n can vary to compensate. Then,

$$w(Y_{t=b}) \leqslant K_1 w(Y_0) + K_2 h. \tag{A.8}$$

Recall $w(Y_0)$ represents the amount of uncertainty in the initial condition of state variable Y. As this uncertainty approaches zero, $w(Y_0) \to 0$. When in addition $h \to 0$, then $w(Y_n)$, the uncertainty in Y_n, also approaches 0. This constitutes *convergence.* Eq. (A.8) also shows *stability*: The uncertainty in the predictions is bounded by the uncertainty in the initial conditions times constant K_1, as $h \to 0$.

Theorem B.2. *Let $\boldsymbol{Y}' = \boldsymbol{F}(\boldsymbol{Y})$ be a system of first-order differential equations, where $\boldsymbol{F}$ is a vector of interval valued functions of $\boldsymbol{Y}$. We consider some bounded subset $[lo, hi]$ of the reals such that for each component $Y_{(j)}$ of vector $\boldsymbol{Y}$, $Y_{(j)}(t) \subseteq [lo, hi]$. We assume that $\boldsymbol{F}(\boldsymbol{Y})$ is defined when each $Y_{(j)} \subseteq [lo, hi]$, and that each F_i in vector $\boldsymbol{F}$ is the natural interval extension of a real rational function f_i.* [19]

Let h be the maximum step size, let $\|\boldsymbol{Y}_{t=b}\|$ represent the amount of uncertainty in the simulated estimate of $\boldsymbol{Y}$ at interpolated time point $t = b$ as measured by its vector norm, [20] *and let $\|\boldsymbol{Y}_0\|$ represent the amount of uncertainty in the initial conditions. Then there are constants K_1 and K_2 such that*

$$\|\boldsymbol{Y}_{t=b}\| \leqslant K_1 \|\boldsymbol{Y}_0\| + K_2 h.$$

Higher-order systems

The proof of Lemma B.1 extends to higher-order systems as follows. First, we describe the higher-order system as a system of first-order equations [46, Section 3.2]. [21]

Each individual equation F_i in the system $\boldsymbol{F}$ may be a function of several variables and interval valued constants $Y_{(j)}$, each a component of vector $\boldsymbol{Y}$. We must push through the proof of Lemma B.1 for $\boldsymbol{Y}$ and $\boldsymbol{F}$ in place of Y and F.

Steps (1) and (2) of the proof of Lemma B.1 now involve a mean value constraint for *each* component F_i of vector $\boldsymbol{F}$. The algorithm now needs to terminate for a fixed point to be reached. Termination was discussed in Section 4.3.2. Step (3) generalizes to $\boldsymbol{F}$ because it applies to each component F_i. Step (4) uses interval widths, which do not apply to *vectors* of intervals. Norms may be used instead, defining the norm of a vector as the width of its widest component interval. Since Moore [66] proves his

[19] See Footnote 8.

[20] See Footnote 9.

[21] Gear assumes that the higher-order equation can be algebraically manipulated such that the highest-order derivative is on the left of the "=" and everything else is on the right. While Gear makes this assumption without further comment, there are instances in which a higher-order equation cannot be so manipulated, for example the nonseparable equation $x''(x')^2 = 0$. Although Q3 can simulate this equation, in such cases the theorem reduces to a conjecture.

Theorem 4.1 and Lemma 4.2 for interval vectors, and both widths and norms are single numbers, the remaining steps work without modification.

Variable h

In step size refinement, step sizes can vary from one step to the next, yet Lemma B.1 assumes all steps have the same size h. Lemma B.1 extends to variable step sizes by taking h as the size of the largest step instead of the size of each step. Then, most of the proof of Lemma B.1 becomes conservative since each step size is $\leqslant h$.

Theorem B.2 does not apply to models containing monotonic function ($\mathtt{M}^+$ and $\mathtt{M}^-$) constraints bounded by envelopes (Section 2.1.2), because two separate functions defining upper and lower envelopes are not a single natural interval extension (Section 2.1.3), as specified in the theorem statement and required by step (1) of the proof of Lemma B.1. For example, propagation of a number through non-identical envelopes results in an interval, yet natural interval extensions, which are generalizations of real valued functions, produce real results when passed real arguments. Thus, we wish to extend our results for convergence and stability to models containing envelopes.

Corollary B.3. *Theorem* B.2 *applies to systems containing monotonic function constraints if for each monotonic function constraint, the space of plausible monotonic functions can be bounded by two envelope functions that differ only in the values of one or more constants c_i. For such systems we define $\|Y_t^*\|$, the norm at time point $t = b$, as the greater of $\|Y_t\|$ from Theorem* B.2 *and the largest amount by which any pair of envelope functions differs in its values for any c_i. Then,*

$$\|Y_{t=b}^*\| \leqslant K_1 \|Y_0^*\| + K_2 h.$$

Proof. A function E defined by an algebraic expression can be rewritten in a generalized form G, such that all of the constants in E are arguments of G. Then G is equivalent to E when passed values corresponding to the values of the constants in E.

We apply this generalization idea to monotonic function spaces which are bounded by two explicitly specified envelope functions $E_{lowerEnvelope}$ and $E_{upperEnvelope}$, with generalized forms $G_{lowerEnvelope}$ and $G_{upperEnvelope}$. We distinguish two cases:

Case 1: $E_{lowerEnvelope}$ and $E_{upperEnvelope}$ differ only in the values of some constants. Then, $G_{lowerEnvelope} \equiv G_{upperEnvelope}$. Call this function G.

Consider each constant c_i whose value differs between $E_{lowerEnvelope}$ and $E_{upperEnvelope}$. Let $\underline{c_i}$ be the lower of the values and $\overline{c_i}$ the higher one. We define an interval $C_i = [\underline{c_i}, \overline{c_i}]$, and pass C_i to G instead of $\underline{c_i}$ or $\overline{c_i}$. By inclusion monotonicity,

$$G(\ldots, C_i, \ldots) \supseteq G(\ldots, \underline{c_i}, \ldots), \qquad G(\ldots, C_i, \ldots) \supseteq G(\ldots, \overline{c_i}, \ldots).$$

The norm $\|G\|$ is defined as $\max_{r=i,j,\ldots} w(C_r)$. If we substitute $G(\ldots, C_i, \ldots, C_j, \ldots)$ for $E_{lowerEnvelope}$ and $E_{upperEnvelope}$, the two envelopes are now described by a single function and the system is now subject to Theorem B.2, but with its norm $\|Y_n^*\|$ at each time point defined by $\max(\|G\|, \|Y_n\|)$.

This reasoning is extended to systems containing more than one pair of envelopes by using $\|Y_n^*\| = \max(\max_{\text{all } G}(\|G\|), \|Y_n\|)$. The envelopes have now been accounted for.

As a special case, if the system is fully specified then for each $E_{lowerEnvelope} \equiv E_{upperEnvelope}$, $\|G\| = 0$, so $\|Y_0\| = \|Y_0^*\| = 0$, and convergence applies (Section 4.2).

Case 2: The argument above applies only if, for the envelopes in each pair, both have the same generalized form, i.e. $G_{lowerEnvelope} \equiv G_{upperEnvelope}$. When is this not true?

- When some pair of envelopes differ in more ways than just values of constants.
- When an envelope has no natural interval extension. For example, it might be a function defined using a lookup table.

In such cases, we push the proof through by enclosing $E_{lowerEnvelope}$ and $E_{upperEnvelope}$ with less constraining envelopes that do have the same generalized form G. Then we can apply Case 1 above. Since the corollary now applies to a less constraining version of the original system, the actual situation is at least as good.

References

[1] H. Abelson and G.J. Sussman, *Structure and Interpretation of Computer Programs* (MIT Press, Cambridge, MA, 1985).

[2] G. Alefeld and J. Herzberger, *Introduction to Interval Computations* (Academic Press, San Diego, CA, 1983).

[3] J.F. Allen, Maintaining knowledge about temporal intervals, *Comm. ACM* 26 (11) (1983) 832-843.

[4] N.S. Asaithambi, S. Zuhe and R.E. Moore, On computing the range of values, *Computing* 28 (1982) 225-237.

[5] D. Berleant, A unified representation for numerical and qualitative simulations, *Appl. Computing Review* 3 (1) (1995) 23-26.

[6] D. Berleant, Automatically verified reasoning with both intervals and probability density functions, *Interval Computations* 2 (1993) 48-70.

[7] D. Berleant, The use of partial quantitative information with qualitative reasoning, Ph.D. Thesis, Tech. Rept. AI90-163, Artificial Intelligence Laboratory, University of Texas, Austin, TX (1991).

[8] D. Berleant, C. Chandra, K. Bognæs, C. Liaw, L. Sheng and J. Ch'ng, Probabilities of qualitative behaviors for dependability analysis of a fault tolerance model, in: *Conference Proceedings 1992 Symposium on Applied Computing*, Kansas City, MO (ACM Press, New York, 1992) 883-889; preliminary results in: *Workshop Notes AAAI-91 Model Based Reasoning Workshop*, Anaheim, CA (1991).

[9] D. Berleant and B.J. Kuipers, Qualitative-numeric simulation with *Q3*, in: B. Faltings and P. Struss, *Recent Advances in Qualitative Physics* (MIT Press, Cambridge, MA, 1992) 3-16.

[10] D.G. Bobrow, ed., *Qualitative Reasoning about Physical Systems* (MIT Press, Cambridge, MA, 1985); reprinted from: *Artificial Intelligence* 24 (1984).

[11] G. Bohlender, C. Ullrich, J. von Gudenberg and L.B. Rall, *Pascal-SC: A Computer Language for Scientific Computation* (Academic Press, San Diego, CA, 1987).

[12] V.C. Boltyanskii, *Envelopes* (Macmillan, New York, 1964).

[13] A. Bonarini and V. Maniezzo, Integrating qualitative and quantitative modeling, *Internat. J. Expert Systems* 4 (1) (1991) 51-70; preliminary report in: *Proceedings 2nd Annual Conference on AI, Simulation and Planning in High Autonomy Systems*, Cocoa Beach, FL (IEEE Computer Society Press, Los Alamitos, CA, 1991) 290-297; also in: *Internal Working Proceedings 1st European Workshop on Qualitative Reasoning about Physical Systems*, Genova, Italy (1991).

[14] P.P. Bonissone, A fuzzy sets based linguistic approach: theory and applications, in: *Proceedings 1980 Winter Simulation Conference*, Orlando, FL (IEEE, New York, 1980) 99-111; also in: M.M. Gupta and E. Sanchez, eds., *Approximate Reasoning in Decision Analysis* (North-Holland, Amsterdam, 1982).

[15] P.P. Bonissone and K.S. Decker, Selecting uncertainty calculi and granularity: an experiment in trading off precision and complexity, in: L.N. Kanal and J.F. Lemmer, eds., *Uncertainty in Artificial Intelligence* (Elsevier, Amsterdam, 1986) 217–247.

[16] A. Brandt, Multi-level adaptive solutions to boundary-value problems, *Math. Comput.* 31 (138) (1977) 333–390.

[17] W.L. Briggs, *A Multigrid Tutorial* (SIAM, Philadelphia, PA, 1987).

[18] R.A. Brooks, Symbolic reasoning among 3-D models and 2-D images, *Artificial Intelligence* 17 (1981) 285–348.

[19] R.E. Bryant, MOSSIM: a switch-level simulator for MOS LSI, in: *Proceedings 18th Design Automation Conference* (1981) 786–790.

[20] B.R. Chawla, H.K. Gummel and P. Kozak, MOTIS—an MOS timing simulator, *IEEE Trans. Circuits and Systems* 22 (12) (1975) 901–910.

[21] P.R. Cohen and E.A. Feigenbaum, eds., *Handbook of Artificial Intelligence* 3 (William Kaufmann Inc., Los Altos, CA, 1982) 313–321.

[22] E.W. Coiera, Qualitative superposition, *Artificial Intelligence* 56 (2–3) (1992) 171–196.

[23] H. Cornelius and R. Lohner, Computing the range of values of real functions with accuracy higher than second order, *Computing* 33 (1984) 331–347.

[24] D.T. Dalle Molle, Qualitative simulation of dynamic chemical processes, Ph.D. Thesis, Chemical Engineering Department, University of Texas, Austin, TX (1989); reprinted as: Tech. Rept. AI89-107, Artificial Intelligence Laboratory, University of Texas, Austin, TX (1989).

[25] D.T. Dalle Molle and T.F. Edgar, Qualitative modeling of chemical reaction systems, in: M.L. Mavrovouniotis, ed., *Artificial Intelligence in Process Engineering* (Academic Press, San Diego, CA, 1991).

[26] B. D'Ambrosio, Extending the mathematics in qualitative process theory, in: *Proceedings AAAI-87*, Seattle, WA (MIT Press, Cambridge, MA, 1987) 595–599.

[27] E. Davis, Constraint propagation with interval labels, *Artificial Intelligence* 32 (3) (1987) 281–331.

[28] T.L. Dean and M. Boddy, Reasoning about partially ordered events, *Artificial Intelligence* 36 (3) (1988) 375–399.

[29] D. DeCoste, Dynamic across-time measurement interpretation, *Artificial Intelligence* 51 (1–3) (1991) 273–341; also in: *Proceedings AAAI-90*, Boston, MA (MIT Press, Cambridge, MA, 1990) 373–379.

[30] J. de Kleer, Qualitative and quantitative knowledge in classical mechanics, Tech. Rept. AI-TR-352, MIT, Artificial Intelligence Laboratory, Cambridge, MA (1975).

[31] W. Derrick and S. Grossman, *Elementary Differential Equations with Applications* (Addison-Wesley, Reading, MA, 1976).

[32] F. DiCesare, Z. Sahnoun and P.P. Bonissone, Linguistic summarization of fuzzy data, *Inform. Sci.* 52 (1990) 141–152.

[33] D. Dubois and H. Prade, Fuzzy arithmetic in qualitative reasoning, in: *Ensembles Flous 87: Algèbre, Intégrations, Calcul d'Intervalles et Approximations*, Rapport L.S.I. No. 302, Langages et Systèmes Informatiques, ENSEEIHT, Toulouse, France (1988).

[34] D. Dubois and H. Prade, Operations on fuzzy numbers, *Internat. J. Systems Sci.* 9 (6) (1978).

[35] D. Dvorak, Model-based monitoring of dynamic systems, in: *Proceedings IJCAI-89*, Detroit, MI (1989) 1238–1243.

[36] D. Dvorak, Monitoring and diagnosis of continuous dynamic systems using semiquantitative simulation, Ph.D. Thesis, Tech. Rept. AI92-170, Artificial Intelligence Laboratory, University of Texas, Austin, TX (1992).

[37] P. Eijgenraam, *The Solution of Initial Value Problems Using Interval Arithmetic: Formulation and Analysis of an Algorithm* (Mathematisch Centrum, Amsterdam, 1981).

[38] B. Faltings and P. Struss, *Recent Advances in Qualitative Physics* (MIT Press, Cambridge, MA, 1992).

[39] A. Farquhar and G. Brajnik, A Semi-Quantitative Physics Compiler, in: *Proceedings 8th International Workshop on Qualitative Reasoning*, 1994.

[40] P.A. Fishwick, Fuzzy simulation: specifying and identifying qualitative models, *Internat. J. General Systems* 19 (1991) 295–316.

[41] K.D. Forbus, Qualitative process theory, *Artificial Intelligence* 24 (1984); reprinted in: D.G. Bobrow, ed., *Qualitative Reasoning about Physical Systems* (MIT Press, Cambridge, MA, 1985).

[42] K.D. Forbus, Interpreting measurements of physical systems, in: *Proceedings AAAI-86*, Philadelphia, PA (MIT Press, Cambridge, MA, 1986) 113–117.
[43] K.D. Forbus and B. Falkenhainer, Self-explanatory simulations: an integration of qualitative and quantitative knowledge, in: B. Faltings and P. Struss, *Recent Advances in Qualitative Physics* (MIT Press, Cambridge, MA, 1992) 49–66; also in: *Proceedings AAAI-90*, Boston, MA (MIT Press, Cambridge, MA, 1990) 380–387.
[44] K.D. Forbus and B. Falkenhainer, Self-explanatory simulations: scaling up to large models, in: *Proceedings AAAI-92*, San Jose, CA (MIT Press, Cambridge, MA, 1992).
[45] J.W. Forrester, *Industrial Dynamics* (MIT Press, Cambridge, MA, 1961).
[46] W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations* (Prentice-Hall, Englewood Cliffs, NJ, 1971).
[47] W. Hamscher, Modelling digital circuits for troubleshooting, *Artificial Intelligence* 51 (1–3) (1991) 223–271.
[48] P. Henrici, *Discrete Variable Methods for Ordinary Differential Equations* (John Wiley and Sons, New York, 1962).
[49] E. Hyvönen, Constraint reasoning based on interval arithmetic: the tolerance propagation approach, *Artificial Intelligence* 58 (1–3) (1992) 71–112.
[50] A. Kaufmann and M. Gupta, *Introduction to Fuzzy Arithmetic* (Van Nostrand Reinhold Co., New York, 1985).
[51] H.A. Kautz and P.B. Ladkin, Integrating metric and qualitative temporal reasoning, in: *Proceedings AAAI-91*, Anaheim, CA (MIT Press, Cambridge, MA, 1991) 241–246.
[52] H. Kay, Refining imprecise models and their behaviors, Ph.D. Thesis, Department of Computer Science, University of Texas, Austin, TX (1996).
[53] H. Kay and B.J. Kuipers, Numerical behavior envelopes for qualitative models, in: *Proceedings AAAI-93*, Washington, DC (MIT Press, Cambridge, MA, 1993) 606–613.
[54] G.T. Kelley, J.I. Glasgow and R.E. Ellis, Extracting qualitative descriptions from quantitative models, in: *Proceedings 2nd Annual Conference on AI, Simulation and Planning in High Autonomy Systems*, Cocoa Beach, FL (IEEE Computer Society Press, Los Alamitos, CA, 1991) 298–305.
[55] Y. Kiang, Exploration of qualitative reasoning and its applications to management, Ph.D. Thesis, University of Texas at Austin, Austin, TX (1991).
[56] L.V. Kolev, V.M. Mladenov and S.S. Vladov, Interval mathematics algorithms for tolerance analysis, *IEEE Trans. Circuits and Systems* 35 (8) (1988) 967–975.
[57] B.J. Kuipers, *Qualitative Reasoning* (MIT Press, Cambridge, MA, 1994).
[58] B.J. Kuipers, Qualitative simulation, *Artificial Intelligence* 32 (1986) 57–95.
[59] B.J. Kuipers and D. Berleant, Using incomplete quantitative knowledge in qualitative reasoning, in: *Proceedings AAAI-88*, St. Paul, MN (MIT Press, Cambridge, MA, 1988) 324–329.
[60] R.J. Lohner, Enclosing the solutions of ordinary initial and boundary value problems, in: E. Kaucher, U. Kulisch and C. Ullrich, eds., *Computerarithmetic: Scientific Computation and Programming Languages* (Teubner, Stuttgart, 1987) 255–286.
[61] A.K. Mackworth and E.C. Freuder, The complexity of some polynomial network consistency algorithms for constraint satisfaction problems, *Artificial Intelligence* 25 (1985) 65–74.
[62] D. McDermott and E. Davis, Planning routes through uncertain territory, *Artificial Intelligence* 22 (1984) 107–156.
[63] I. Meiri, Combining qualitative and quantitative constraints in temporal reasoning, in: *Proceedings AAAI-91*, Anaheim, CA (MIT Press, Cambridge, MA, 1991) 260–267.
[64] A. Missier and L. Travé-Massuyès, Temporal information in qualitative simulation, in: *Proceedings 2nd Annual Conference on AI, Simulation and Planning in High Autonomy Systems*, Cocoa Beach, FL (IEEE Computer Society Press, Los Alamitos, CA, 1991) 298–305.
[65] J. Mohammed and R. Simmons, Qualitative simulation of semiconductor fabrication, in: *Proceedings AAAI-86*, Philadelphia, PA (MIT Press, Cambridge, MA, 1986) 794–799.
[66] R.E. Moore, *Methods and Applications of Interval Arithmetic* (SIAM, Philadelphia, PA, 1979).
[67] T. Nishida and S. Doshita, Reasoning about global behavior of ordinary differential equations by combining qualitative and quantitative analysis, in: *Proceedings 2nd Annual Conference on AI, Simulation*

and Planning in High Autonomy Systems, Cocoa Beach, FL (IEEE Computer Society Press, Los Alamitos, CA, 1991) 98-107.
[68] J.-P. Nordvik, P. Smets and P. Magrez, Fuzzy qualitative modeling, in: B. Bouchon, L. Saitta and R.R. Yager, eds., *Uncertainty and Intelligent Systems* (Springer, New York, 1988).
[69] W. Older and A. Vellino, Constraint arithmetic on real intervals, in: F. Benhamou and A. Colmerauer, eds., *Constraint Logic Programming, Collected Papers* (MIT Press, Cambridge, MA).
[70] W. Press, B. Flannery, S. Teukolsky and W. Vetterling, *Numerical Recipes* (Cambridge University Press, Cambridge, MA, 1986).
[71] E. Sacks, Hierarchical reasoning about inequalities, in: *Proceedings AAAI-87*, Seattle, WA (MIT Press, Cambridge, MA, 1987) 649-654.
[72] E. Sacks, Automatic analysis of one-parameter planar ordinary differential equations by intelligent numerical simulation, *Artificial Intelligence* 48 (1991) 27-56.
[73] E. Sacks, Review of K. Yip, *KAM: A System for Intelligently Guided Numerical Experimentation by Computer* (MIT Press, Cambridge, MA, 1991), *SIGART Bull.* 3 (3) (1992) 18-19.
[74] R.A. Saleh and A.R. Newton, *Mixed-Mode Simulation* (Kluwer, Boston, MA, 1990).
[75] Q. Shen and R. Leitch, Fuzzy qualitative simulation, *IEEE Trans. Systems Man Cybernet.* 23 (1993) 1038-1061.
[76] Q. Shen and R. Leitch, Integrating common-sense and qualitative simulation by the use of fuzzy sets, in: B. Faltings and P. Struss, *Recent Advances in Qualitative Physics* (MIT Press, Cambridge, MA, 1992) 83-100.
[77] R. Simmons, The use of qualitative and quantitative simulations, in: *Proceedings AAAI-83*, Washington, DC (MIT Press, Cambridge, MA, 1983) 364-368.
[78] R. Simmons, "Commonsense" arithmetic reasoning, in: *Proceedings AAAI-86*, Philadelphia, PA (MIT Press, Cambridge, MA, 1986) 118-124.
[79] R. Simmons, A theory of debugging plans and interpretations, in: *Proceedings AAAI-88*, St. Paul, MN (MIT Press, Cambridge, MA, 1988) 94-99.
[80] R. Simmons, The roles of associational and causal reasoning in problem solving, *Artificial Intelligence* 53 (1-2) (1992) 159-207.
[81] R. Simmons and R. Davis, Generate, test and debug: combining associational rules and causal models, in: *Proceedings IJCAI-87*, Milan, Italy (1987) 1071-1078.
[82] J.M. Siskind and D.A. McAllester, Nondeterministic Lisp as a substrate for constraint logic programming, in: *Proceedings AAAI-93*, Washington, DC (MIT Press, Cambridge, MA, 1993).
[83] C.J. Terman, RSIM—a logic-level timing simulator, in: *Proceedings IEEE International Conference on Computer Design* (1983) 437-440.
[84] R.M. Tong and P.P. Bonissone, Linguistic decision analysis using fuzzy sets, Memorandum No. UCB/ERL M79/72, Electronics Research Laboratory, University of California, Berkeley, CA (1979).
[85] E.R. Tufte, *The Visual Display of Quantitative Information* (Graphics Press, Cheshire, CT, 1983).
[86] M. Vescovi, A. Farquhar and Y. Iwasaki, Numerical interval simulation: bounding behaviors of non-monotonic systems, in: *Proceedings IJCAI-95*, Montreal, Que. (1995).
[87] B.C. Williams, Doing time: putting qualitative reasoning on firmer ground, in: *Proceedings AAAI-86*, Philadelphia, PA (MIT Press, Cambridge, MA, 1986) 105-112.
[88] B.C. Williams, MINIMA: a symbolic approach to qualitative algebraic reasoning, in: *Proceedings AAAI-88*, St. Paul, MN (MIT Press, Cambridge, MA, 1988) 264-269.
[89] B.C. Williams, A theory of interactions: unifying qualitative and quantitative algebraic reasoning, *Artificial Intelligence* 51 (1-3) (1991) 39-94.
[90] K. Yip, *KAM: A System for Intelligently Guided Numerical Experimentation by Computer* (MIT Press, Cambridge, MA, 1991).
[91] D.M. Young, Stability ($h \rightarrow 0$ and $h > 0$) for systems, Course Notes Supplement 78.6, revises: D.M. Young and R.T. Gregory, *A Survey of Numerical Mathematics* II (Addison-Wesley, Reading, MA, 1972) 566-573, Center for Numerical Analysis, University of Texas at Austin (1978).
[92] D.M. Young and R.T. Gregory, *A Survey of Numerical Mathematics* II (Addison-Wesley, Reading, MA, 1972).
[93] F. Zhao, Extracting and representing qualitative behaviors of complex systems in phase spaces, *Artificial Intelligence* 69 (1994) 51-92; also in: *Proceedings IJCAI-91*, Sidney, Australia (Morgan Kaufmann, Los Altos, CA, 1991).