Flow*: An Analyzer for Non-Linear Hybrid Systems

Xin Chen[1], Erika Ábrahám[1], and Sriram Sankaranarayanan[2] ⋆

1. RWTH Aachen University, Germany. {xin.chen,abraham}@cs.rwth-aachen.de
2. University of Colorado, Boulder, CO. srirams@colorado.edu

Abstract. The tool FLOW* performs Taylor model-based flowpipe construction for non-linear (polynomial) hybrid systems. FLOW* combines well-known Taylor model arithmetic techniques for guaranteed approximations of the continuous dynamics in each mode with a combination of approaches for handling mode invariants and discrete transitions. FLOW* supports a wide variety of optimizations including adaptive step sizes, adaptive selection of approximation orders and the heuristic selection of template directions for aggregating flowpipes. This paper describes FLOW* and demonstrates its performance on a series of non-linear continuous and hybrid system benchmarks. Our comparisons show that FLOW* is competitive with other tools.

1 Overview of FLOW*

In this paper, we present the FLOW* tool to generate flowpipes for non-linear hybrid systems using *Taylor Models* (TMs). TMs were originally proposed by Berz and Makino [1] to represent functions by means of higher-order Taylor polynomial expansions, bloated by an interval to represent the approximation error. TMs support functional operations such as addition, multiplication, division, derivation and anti-derivation. Guaranteed integration techniques can utilize TMs to provide tight flowpipe over-approximations to non-linear ODEs, with each flowpipe segment represented by a TM [2]. However, these techniques do not naturally extend to *non-linear* hybrid systems consisting of multiple modes and discrete transitions (jumps).

Figure 1 presents a schematic diagram of the major components of FLOW*. FLOW* accepts (i) A hybrid system model file which describes the modes, the polynomial dynamics associated with each mode and the transitions between modes; (ii) A specification file includes TM flowpipes with the state space and unsafe set specifications. For a model file, FLOW* performs a flowpipe construction for a specified time horizon $[0, T]$ and a maximum jump depth J such that the flowpipe set is an over-approximation of the states which can be reached in $[0, T]$ with at most J jumps. FLOW* also checks whether the flowpipe intersects the unsafe set and outputs a visualization of the set of reachable states using polyhedral over-approximations of the computed TM flowpipes. FLOW* is extensible in quite simple ways. Our TM output can be parsed in by other tools, including FLOW* itself to check multiple properties incrementally. This can help

⋆ Sankaranarayanan's research is supported by the US National Science Foundation (NSF) under grant numbers CNS-0953941 and CPS-1035845. All opinions expressed are those of the authors and not necessarily of the NSF.

support future advances such as checking for MTL property satisfaction for flowpipes, finding limit cycles, inferring likely invariants or Lyapunov functions from flowpipes and extending robustness metric computations for entire flowpipes [3]. The FLOW* tool with its source code and the set of non-linear benchmarks used in this paper, as well as our original work [4] are available on-line [1].

TM Integrator: The Taylor model integrator implements a guaranteed integration scheme using higher-order TMs, along the lines of previous work described in [1, 2, 5, 6]. Our implementation includes numerous enhancements to the existing TM integration techniques including the adaptive adjustment of the orders of the TMs for better control of integration error and better handling of flowpipe guard intersections.

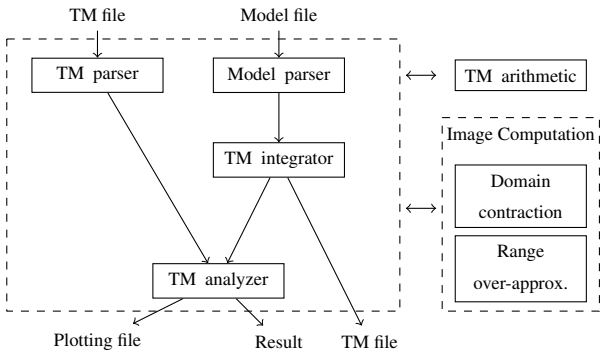

Fig. 1. Structure of FLOW*

Image Computation for Discrete Transitions: The implementation of image computation is based on the techniques described in our previous work for approximating the intersection of TMs with guard sets of the transitions [4]. This approximation is achieved by two complementary techniques: (a) the domain contraction technique computes a smaller TM by shrinking the initial condition and the time interval for which an intersection with the guard is possible; (b) the range over-approximation technique that converts TMs into representations such as template polyhedron or zonotopes over which guard intersection can be computed efficiently. Range over-approximation also includes the conversion of the result back to a TM. FLOW* implements a combination of (a) and (b) to achieve a better accuracy than using either of them.

The verification techniques developed for hybrid systems over the last few decades have resulted in many tools for linear hybrid systems analysis including HyTech, Checkmate, d/dt, Ariadne, HySAT/iSAT, RealPaver, PHAVer, SpaceEx [7]. However, few tools exist for non-linear systems. A few notable non-linear analysis tools include KeYmaera [8], Ariadne [9] and HySAT/iSAT [10].

2 Novel features in FLOW*

We discuss features of the FLOW* which have not been included in our earlier work [4]. These features improve the efficiency of our overall approach and allow the automatic

[1] http://systems.cs.colorado.edu/research/cyberphysical/taylormodels/

Name	Description
Remainder Interval (I_e)	Maximum remainder interval I_e for each integration step.
	Adapt step sizes and TM orders to ensure error within I_e.
Adaptive Step Sizes	Range $[\alpha, \beta]$ for possible step sizes.
Adaptive TM Orders	Change TM orders on-the-fly.
	Allow state variables to have different orders.
Preconditioning	Change of basis for better flow approximation.
Template Directions	Aggregating flowpipe segments by a template polyhedron.

Fig. 2. Basic parameters for controlling FLOW* algorithm.

selection of parameters. In our experience, the integration of non-linear systems is often quite *fragile*. There are many parameters to adjust, as outlined in Figure 2. It is necessary to choose them judiciously to ensure that the desired flowpipe accuracy is maintained without expending too much resources. Automating the choice of some of these parameters on-the-fly seems to be the only possible solution to this problem. The new features added to FLOW* automate this to a large extent by allowing the user to specify a flexible range of parameters and adapting the flowpipe construction on the fly within this range to trade off precision of the result against the running time and memory consumption.

Specifying different orders over dimensions The performance and accuracy of TM integration depends critically on the order of the TM approximation chosen. However, existing approaches specify a single fixed order for each state variable. Nevertheless, it is clear that for a complex system, different state variables grow at varying rates. A key feature of FLOW* allows us to perform integration while specifying orders for the TM representation of each state variable independently during the integration process. For instance, state variables that represent timers can be specified to have order 1 TMs, whereas fast varying variables can be represented by higher order TMs at the same time.

Adaptive techniques FLOW* supports adaptive integration time step and adaptive TM order selection for each state variable using specially designed schemes.

Adaptive step sizing. Adaptive step sizing is a standard feature in many flowpipe tools including SpaceEx [7]. A range $[\alpha, \beta]$ wherein $0 < \alpha \leq \beta$ is specified by the user for the time step size. The purpose of adaptive step sizing is to choose a step size $\delta \in [\alpha, \beta]$ such that the Picard operator[2] on the current flowpipe yields a TM with remainder interval that is contained in I_e. Our approach starts from $\delta = \beta$ and as long as the Picard operator fails to yield a remainder inside I_e, it updates δ by $\delta' := \lambda\delta$ using a discount factor λ which is set to 0.5 in our implementation. If $\delta < \alpha$, a diagnostic message is printed asking the user to either (a) decrease the lower bound on the adaptive time step α, (b) enlarge the interval I_e or (c) increase the approximation order. Note that either (a) or (c) slows down the overall computation, but is ultimately unavoidable if the dynamics are hard to approximate.

[2] Given an ODE $\frac{d\boldsymbol{x}}{dt} = F(\boldsymbol{x}, t)$, the Picard operator on a function $g(\boldsymbol{x}_0, t)$ is given by $\mathbb{P}_F(g)(\boldsymbol{x}_0, t) = \boldsymbol{x}_0 + \int_0^t F(g(\boldsymbol{x}_0, s), s)\ ds$. If the Picard operator is contractive on g with some t, then g is an over-approximation of the flow at time t.

Adaptive TM orders. Adaptive choice of TM orders also seeks to bound the error within I_e. Our technique first concentrates on the state variables for which the interval error estimate is breached. The orders of these state variables are increased by 1. If the technique fails to achieve the interval I_e, the orders of all the remaining variables are increased as well. This process continues until the upper limit specified by the user is breached. On the other hand, if the interval I_e is achieved, our approach starts to decrease the TM orders to find the smallest order for which the flowpipe's remainder is contained in I_e. If adapting the TM orders fails, the tool prints a diagnostic.

Currently, the techniques of adaptive orders and step sizes operate independently of each other. The tool fixes the step sizes and performs adaptive TM orders; or fixes the TM orders and adapts the step sizes. The simultaneous adaptation of both step sizes and TM orders is not supported. This is also a challenging problem, since many optimal points of tradeoff may exist. Exploring these choices systematically in a time-efficient manner will be part of our future work.

FLOW* also provides various options for aggregating flowpipe/guard intersections. Users are allowed to partially or fully specify a parallelotopic template for the union.

3 Experimental Evaluation

We now provide an experimental evaluation of FLOW*. The evaluation includes a comparison against VNODE-LP [3] by Nedialkov et al. [11], a state-of-the art guaranteed integration tool for continuous systems. Next, we consider the evaluation for hybrid systems. Our previous work demonstrated a comparison against the Ariadne and the HySAT/iSAT tool. We have been unable to obtain tools from other papers to enable a meaningful comparison. Therefore, we restrict ourselves to showcasing performance on a new class of benchmarks and comparison of our new features against the earlier prototype. All experiments were performed on a i7-860 2.8GHz CPU with 4GB RAM running Ubuntu Linux. The benchmarks can be downloaded as a part of our release.

Continuous systems: Table 1 shows a comparison with the VNODE-LP tool. The VNODE-LP tool often fails to integrate these benchmarks when the initial set is too large ("`VNODE - LP could not reach t = [T,T]`"). Therefore, to ensure a fair comparison, we subdivide the initial set into smaller intervals and integrate each separately. The G.S. column denotes the *grid sizes* used for the initial sets. Our approach subdivides the initial sets uniformly. We manually find the largest grid size for which VNODE-LP does not fail. This setting is chosen for the experiments. For experiments $\#5, 6, 7$, VNODE-LP failed or could not complete within the set time out of an hour. In contrast, FLOW* performs well on all the benchmark examples, often finishing well within the one hour timeout interval. The parameters used for FLOW* are provided for reference in Table 1.

The precision comparisons are quite tricky. FLOW* computes TM flowpipe segments whereas VNODE-LP computes boxes. Converting TMs into boxes can be quite expensive. Our approach uses a coarse computation using interval arithmetic. Nevertheless, the comparison in terms of the *max. widths* of the intervals (i.e, the maximum

[3] `http://www.cas.mcmaster.ca/~nedialk/Software/VNODE/VNODE.shtml`

Table 1. Comparisons on continuous benchmarks. All times are in seconds. Legends: var: number of the variables, T: time horizon is [0,T], orders: the adaption range of the TM orders, R.E.: remainder estimation, R.W.: remainder width, G.S.: grid size, W.S.: width of the solution interval for $\boldsymbol{x}(T)$, T.O.: $>$ 1 hour.

			FLOW*							VNODE-LP		
ID	Benchmarks	var	T	step	orders	R.E.	time	R.W.	W.S.	G.S.	time	W.S.
1	Brusselator	2	10	0.02	3∼5	[-1e-4,1e-4]	8.2	<**1e-5**	**3e-2**	2e-2	**1.9**	3.3e-2
2	Lorentz	3	1	0.01	3∼6	[-1e-4,1e-4]	**15**	<**5e-4**	1.08	1e-2	64	**1.06**
3	Rössler	3	5	0.02	4∼6	[-5e-4,5e-4]	**30**	<**5e-4**	2.60	2e-2	50	**1.95**
4	6-D sys. [12]	6	1	0.01	3∼6	[-1e-2,1e-2]	**102**	<**1e-3**	**0.37**	2.5e-2	180	0.40
5	6-D sys. [12]	6	2	0.01	3∼6	[-1e-2,1e-2]	**213**	<**1e-3**	**0.37**	1.5e-2	T.O.	-
6	Reaction [13]	7	1e-3	2e-6	3∼5	[-1e-2,1e-2]	**469**	<**1e-1**	**15.91**	2e-2	Fail	-
7	Phosphorelay [14]	7	2	2e-3	3∼5	[-1e-3,1e-3]	**882**	<**1e-4**	**0.17**	1.5e-3	Fail	-
8	Phosphorelay [14]	7	3	2e-3	3∼5	[-1e-3,1e-3]	**836**	<**1e-6**	**3e-2**	1e-3	3156	3.5e-2
9	Bio [15]	9	0.1	1e-3	3∼5	[-1e-2,1e-2]	**121**	<**1e-1**	**1.42**	1e-2	318	1.58
10	Bio [15]	9	0.1	1e-3	3∼5	[-1e-2,1e-2]	**153**	<**1e-1**	**2.11**	1e-2	T.O.	-

interval width along any of the dimensions) is quite similar. Overall when successful, VNODE-LP's flowpipe was quite similar to that of FLOW*. We are investigating better box and octagon approximation schemes for TMs to enable a better comparison.

Hybrid systems: We demonstrate our approach on a series of non-linear navigation benchmarks representing a vessel moving through a fluid. We assume a velocity dependent drag force along each direction $F_x : -k \cdot v_x^3$ and $F_y : -k \cdot v_y^3$ to the velocities v_x, v_y in each cell with $k = 0.1$. The other parameters are the almost the same as the linear benchmarks [16] with a few exceptions: the initial values for v_x in NAV05 lie in the range $[0.8, 0.1]$, and for NAV09, $x(0) \in [3.1, 3.5], v_y(0) \in [-0.8, -0.5]$. Table 2 summarizes the performance on a set of hybrid system benchmarks, comparing the effect of adaptive step sizes and adaptive orders. Interestingly, our results indicate that adapting the orders is more advantageous than step sizes. Adapting the TM orders seems to have a pronounced effect on the efficiency of the flowpipe guard intersection procedure. We also include the artificial pancreas (AP) models described in [4] with modified safety specifications. We created new benchmarks instances Diabetic 3 & 4 by adding timing delays between controller mode changes in Diabetic 1 & 2.

References

1. M. Berz. *Modern Map Methods in Particle Beam Physics*, volume 108 of *Advances in Imaging and Electron Physics*. Academic Press, 1999.
2. M. Berz and K. Makino. Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models. *Reliable Computing*, 4:361–369, 1998.
3. Georgios Fainekos and George J. Pappas. Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science*, 410:4262–4291, 2009.
4. Xin Chen, Erika Ábrahám, and Sriram Sankaranarayanan. Taylor model flowpipe construction for non-linear hybrid systems. In *Proc. RTSS'12*, pages 183–192. IEEE, 2012.

Table 2. Hybrid benchmarks. Legends: var: # of variables, loc: # of locations, T: time horizon [0,T], jps: max jump depth, δ: step sizes, t: time cost (s), MUL: multiple TM orders. The safety properties are all proved.

					fixed steps&orders			adaptive steps			adaptive orders		
benchmarks	var	loc	T	jps	δ	order	t	δ	order	t	δ	order	t
n.-l. NAV04	4	7	30	8	0.01	3	138	[0.01,0.1]	3	57	0.05	3∼6	**32**
n.-l. NAV05	4	7	30	8	0.02	4	168	[0.01,0.1]	4	69	0.05	3∼6	**38**
n.-l. NAV06	4	7	30	8	0.01	3	162	[0.01,0.1]	3	70	0.05	3∼6	**40**
n.-l. NAV07	4	14	30	10	0.01	4	567	[0.01,0.1]	4	117	0.05	3∼6	**61**
n.-l. NAV08	4	14	30	10	0.01	4	545	[0.01,0.1]	4	122	0.05	3∼6	**60**
n.-l. NAV09	4	14	30	10	0.01	4	222	[0.01,0.02]	4	117	0.02	3∼6	**47**
Diabetic 1	4	9	360	6	0.02	4	1655	[0.01,0.1]	4	382	0.04	MUL	**212**
Diabetic 2	4	6	360	4	0.02	4	1023	[0.01,0.1]	4	229	0.04	MUL	**142**
Diabetic 3	5	9	360	6	0.02	4	852	[0.01,0.1]	4	191	0.04	MUL	**102**
Diabetic 4	5	6	360	4	0.02	4	502	[0.01,0.1]	4	106	0.04	MUL	**77**
Water tank	5	2	300	10	0.02	3	828	[0.01,0.1]	3	255	0.05	MUL	**218**

5. K. Makino and M. Berz. Taylor models and other validated functional inclusion methods. *J. Pure and Applied Mathematics*, 4(4):379–456, 2003.
6. M. Neher, K. R. Jackson, and N. S. Nedialkov. On Taylor model based integration of ODEs. *SIAM Journal on Numerical Analysis*, 45:236–262, 2006.
7. G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. SpaceEx: Scalable verification of hybrid systems. In *Proc. CAV'11*, volume 6806 of *LNCS*, pages 379–395. Springer, 2011.
8. A. Platzer. *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics*. Springer, 2010.
9. Luca Benvenuti, Davide Bresolin, Alberto Casagrande, Pieter Collins, Alberto Ferrari, Emanuele Mazzi, Alberto Sangiovanni-Vincentelli, and Riziano Villa. Reachability computation for hybrid systems with Ariadne. In *Proc. the 17th IFAC World Congress*. IFAC Papers-OnLine, 2008.
10. M. Fränzle, C. Herde, T. Teige, S. Ratschan, and T. Schubert. Efficient solving of large non-linear arithmetic constraint systems with complex Boolean structure. *Journal on Satisfiability, Boolean Modeling and Computation*, 1:209–236, 2007.
11. N. S. Nedialkov. Implementing a rigorous ode solver through literate programming. In *Modeling, Design, and Simulation of Systems with Uncertainties Mathematical Engineering*, volume 3, pages 3–19. Springer, 2011.
12. Z. She, B. Xue, and Z. Zheng. Algebraic analysis on asymptotic stability of continuous dynamical systems. In *Proc. ISSAC'11*, pages 313–320. ACM, 2011.
13. G. Craciun, Y. Tang, and M. Feinberg. Understanding bistability in complex enzyme-driven reaction networks. *Proc. of the National Academy of Sciences*, 103(23):8697–8702, 2006.
14. E. Klipp, R. Herwig, A. Kowald, C. Wierling, and H. Lehrach. *Systems Biology in Practice: Concepts, Implementation and Application*. Wiley-Blackwell, 2005.
15. J. M. G. Vilar, H. Y. Kueh, N. Barkai, and S. Leibler. Mechanisms of noise-resistance in genetic oscillators. *Proc. of the National Academy of Sciences*, 99(9):5988–5992, 2002.
16. Ansgar Fehnker and Franjo Ivančić. Benchmarks for hybrid systems verification. In *Proc. HSCC'04*, volume 2993 of *LNCS*, pages 326–341. Springer, 2004.