Multi-Rate Integration Algorithms: a Path Towards Efficient Simulation of Object-Oriented Models of Very Large Systems

[Work in Progress]

Akshay Ranade
Dipartimento di Matematica
Politecnico di Milano
Piazza Leonardo da Vinci, 32
20133 Milano - Italy
akshay.ranade@polimi.it

Francesco Casella*
Dip. di Elettronica, Informazione e Bioingegneria
Politecnico di Milano
Piazza Leonardo da Vinci, 32
20133 Milano - Italy
francesco.casella@polimi.it

ABSTRACT

Object-oriented modelling languages allow to build models of large, loosely coupled systems, as well as of multi-domain systems with fast and slow sub-systems easily. State-of-the-art simulation tools employ sophisticated techniques to efficiently turn the system DAEs into ODEs, but then rely on standard single-rate algorithms for the simulation of ODEs. These algorithms perform very poorly in the two above-mentioned classes of systems as their size grows, up to the point of making their simulation practically unfeasible. The goal of this paper is to introduce multi-rate algorithms with error control to the EOOLT community, showing through an exemplary case study the potential they have for the simulation of such large-scale systems.

Keywords

G.1.7 [**Ordinary Differential Equations**]: Initial Value Problems; G.4 [**Mathematical Software**]: Algorithm Design and Analysis

1. INTRODUCTION AND MOTIVATION

Equation-based, object-oriented (O-O) languages and tools have now become well-established for system-level modelling of engineering systems, in particular for those spanning multiple physical domains. Sophisticated symbolic and numerical methods (see, e.g., [3] for a review) are used in object-oriented simulation tools to transform the large, possibly high-index systems of Differential Algebraic Equations (DAEs) into state-space form:

$$\mathbf{y}' = f(t,\mathbf{y}),\ \ \mathbf{y}(t_0) = \mathbf{y_0}$$
$$\mathbf{v} = g(t,\mathbf{y})$$

where $\mathbf{y}$ is the state variable vector and $\mathbf{v}$ the algebraic variable vector. The last step in the simulation is to numerically integrate the ODEs for which there are well-established, general-purpose ordinary differential equation (ODE) solvers (e.g., DASSL [2] or the Sundials suite [6]).

The standard ODE solvers have advanced features such as adaptive step-size and adaptive order for error control. All of them are, however, single-rate algorithms: the solution $\mathbf{y}(t)$ is computed at certain time steps t_k, and in order to do that the *entire* vector function $f(t,\mathbf{y})$ is evaluated once at every time step and possibly at other intermediate points. There are two classes of systems where this approach becomes prohibitively expensive in terms of computational load as the size and complexity of the system model grows.

The first class is given by models of distributed systems having a large number of sub-units, interacting with each other through a network connection. When some kind of local activity is triggered on a single sub-unit, shorter time steps are required in order to keep the integration errors within the specified bounds; however, this local activity does not significantly affect other distant sub-units. Therefore, computing the derivatives for the entire system is unnecessarily wasteful, when only those belonging to the sub-unit in question are changing significantly. Notable examples in this class are models of smart grids, district heating networks, power transmission systems, etc.

The second class is given by models of multi-domain systems, where a slower sub-system interacts with a faster sub-system. Single-rate algorithms will choose a short system-wide time-step, due to the faster sub-system. Most of the computation will be unnecessary in this case, because the states of the slower sub-system will hardly change over these steps. A notable example is the model of a thermal power generation plant, with a slow (and computationally intensive) model of boiler and turbine coupled to a fast, but lighter, model of the electrical generation and trasmission equipment.

Due to this limitation, state-of-the-art O-O solvers scale up badly in terms of computation times, making simulations

---

*Corresponding author

EOOLT 2014, Oct 10-10 2014, Berlin, Germany
ACM 978-1-4503-2953-8/14/10.
http://dx.doi.org/10.1145/2666202.2666214.

infeasible for larger models. To overcome this limitation, fixed time step multirate algorithms are often used for real-time simulation, using co-simulation schemes, see, e.g., [13]. However, the partitioning of the system and the choice of the time steps is entirely up to the modeller, there is no guaranteed error bound, and numerical instabilities might occur. In the spirit of declarative modelling and simulation, the goal of this research is to look for multi-rate integration methods that do not require intervention by the modeller, but yet guarantee rigorous error bound.

The main idea behind multi-rate integration algorithms is to integrate different components with different time-steps, depending on their individual dynamical properties. The faster components are integrated with smaller time-steps than the slower ones. Coupling between fast and slow components is taken care of by interpolation or extrapolation. In this way, the number of evaluations of derivatives $f(t, \mathbf{y})$ for the slower components is greatly reduced. The idea was first introduced in 1960 [10]. Several others, based on different types of methods, have been subsequently introduced[4, 5, 8, 12]. Some theoretical analysis of stability issues exists [1, 7, 11], though only for a restricted class of problems.

The contribution of this paper is to introduce multi-rate algorithms to the EOOLT community, demonstrate their potential on a simple case study, and motivate further research on the topic. The paper is structured as follows: in the next section, the general multirate method is reviewed. Section 3 introduces the model which is used as a test case. Results are discussed in Section 4, which is followed by conclusions and proposal for future research in Section 5.

2. THE MULTI-RATE ALGORITHM

This section is organised as follows: (2.1) gives a general introduction to multirate methods and (2.2) describes the self-adjusting multirate scheme we have used.

2.1 Introduction

Consider the initial value problem

$$\mathbf{y}' = f(t, \mathbf{y}),\ \mathbf{y}(t_0) = \mathbf{y_0}$$

where $\mathbf{y} \in \mathbb{R}^n$. We require f to be continuous and satisfy the Lipschitz condition w.r.t $\mathbf{y}$ for the entire region $\mathbb{R} \times \mathbb{R}^n$

In a general multi-rate scheme, one considers a partitioned system consisting of y_a-the *active* components, and y_l-the *latent* ones

$$\mathbf{y_a}' = f_a(t, \mathbf{y_a}, \mathbf{y_l})$$
$$\mathbf{y_l}' = f_l(t, \mathbf{y_a}, \mathbf{y_l})$$

where $\mathbf{y_a} \in \mathbb{R}^{n_a}$, $\mathbf{y_l} \in \mathbb{R}^{n_l}$ and $n_a + n_l = n$.

The partitioned system is then integrated with a smaller time-step h_a for the active part and a larger time-step h_l for the latent part. Coupling between the two sets of components is taken care of by interpolation/extrapolation.

If the dynamical properties of the system are well known, then the system can be partitioned *a priori*. However, such information is not always available. Moreover the system may be such that components go from being active to latent, and vice-versa, during the course of a simulation. Therefore, methods with an automatic mechanism to partition the system would be desirable.

2.2 A Self-adjusting multirate scheme

Self-adjusting multi-rate integration schemes partition the system based on the local truncation error estimates. Numerical schemes which have lower order methods embedded in them are the ideal choice as a basis for a multi-rate scheme. The error estimate can be computed without additional cost by comparing the lower order method with the higher order one.

A tentative solution for the so-called *global* time-step t_{n-1} to t_n, is computed. Components with error exceeding the tolerance are classified as active. The active components are 'refined' using two steps of size $\frac{\Delta t_n}{2}$, using interpolated values for the so-called *latent* components. There may be components which still have error. The refinement step is repeated, using interpolation for the components which are not recomputed but needed for the refinement. This can be implemented efficiently as a recursive algorithm.

For this study, we have used the recursive multirate Rosenbrock 2nd order scheme proposed in [12]. The Rosenbrock 2nd order numerical method is given by

$$y_n = y_{n-1} + \frac{3}{2}k_1 + \frac{1}{2}k_2$$
$$(I - \gamma hJ)k_1 = hf(t_{n-1}, y_{n-1}) + \gamma h^2 f_t(t_{n-1}, y_{n-1})$$
$$(I - \gamma hJ)k_2 = hf(t_{n-1}, y_{n-1} + k_1) - \gamma h^2 f_t(t_{n-1}, y_{n-1}) - 2k_1$$

where J is the Jacobian and γ is a parameter of the method. This method has the embedded first-order method

$$\bar{y}_n = y_{n-1} + k_1,$$

which is used to estimate the error. This error estimator is then used to partition the system at the end of each global step.

3. A TEST CASE

The test problem we consider is a simplified, lumped-parameter model of a heating system with a central heater supplying heat to several users through a distribution network. The temperature of the distribution network is controlled using P control and that of each user by an on-off controller.

The system is oversimplified and we do not suggest that it has any practical application in the design or analysis of a heating system. However, the resulting system of equations is very stiff and has localised activity. Thus, it is a good test problem to study the applicability of multirate methods to the first class of problems described in the introduction.

The central heating unit consists of a source supplying heat to the distribution network fluid, which is represented by a single temperature and has a large heat capacity, thus also acting as a buffer. A P temperature controller modulates the heat source appropriately.

The model parameters are the maximum heat supply rate Q_{max}, the proportional gain of the controller K_p. The heat supply rate Q of the heater can vary between 0 and Q_{max}. This is modeled by a smooth saturation function, to ensure that all functions are continuously differentiable and thus the (local) existence of solutions can be guaranteed.

$$Q_{sat} = \frac{Q_{max}}{2} \cdot tanh(\frac{2Q}{Q_{max}} - 1) + \frac{Q_{max}}{2} \qquad (2)$$

The model for each user consists of heated unit which has an on-off controller with hysteresis, which provides very fast localized action. The model parameters are the temperature set point T_{SP}, the hysteresis width of the controller T_ϵ, the heat conductance of the units G_h, the heat conductance of the heated units to the atmosphere G_u, and the heat capacity of the heated unit C. The model variables are the temperature T and the state of the on-off controller x.

The heater tries to maintain the temperature close to a set-point T_{SP}. The on-off controller is modelled by a smooth nonlinear system:

$$\dot{x} = -50 \cdot \left(\frac{(x-0.5)\cdot(x+0.5)\cdot x}{0.0474} \cdot T_\epsilon \right) + (T_{SP} - T) \quad (3)$$

which shows on-off behaviour due to a bifurcation triggered by T: as T rises above or falls below T_{SP}, x very rapidly goes to the corresponding equilibrium point. With appropriate tuning, the system behaves like an on-off controller w.r.t the controlled variable, though the mathematical model is continuously differentiable and needs no event handling.

The controller equations are very stiff and very nonlinear. On the one hand, this helps testing the proposed integration algorithm with a stiff problem, as it is often the case with O-O models. On the other hand, accurately describing the local fast (but continuous) transition between the on and the off state requires a lot of time steps, making multi rate-algorithms particularly attractive.

Each heated unit receives heat from the distribution network. The ambient temperature T_0 varies sinusoidally with a period of 86,400 seconds which corresponds to the diurnal temperature variation. The heat received from the network is being regulated by the on-off controller by changing the thermal conductance to the heat source via a nonlinear function $f(x)$, to closely resemble on-off operation. The energy balance of this model can be written as

$$C \cdot \dot{T} = G_h \cdot (T_d - T) \cdot f(x) - G_u \cdot (T - T_0) \quad (4)$$

The heat capacities of the units are set to only slightly differ from each other, so that the period between the on-off transitions of the different units are not the same. In this way, local activity in each unit will take place at different points in time, to the advantage of the multi-rate algorithm.

As to the distribution network, we assume that its average temperature is governed by

$$C_d \cdot \dot{T_d} = Q_{heater} + \sum_{i}^{N} Q_i, \quad (5)$$

where N is the number of users connected to the network. The heat capacity C_d is taken very large in comparison to the heat capacity of an individual unit and scaled up proportionally to N.

3.1 Explicit ODE form of the system

The above subsections described the salient features of the models in a declarative way. In this subsection we show the actual explicit ODEs that describe the system dynamics. For N number of users there are $2 \times N + 1$ differential variables viz. the temperature of the distribution network fluid, the temperatures of the N units and the states of the N on-off controllers. The equations that arise are

$$\dot{T_d} = \frac{1}{C_d} \left\{ tanh\left(2\frac{K_p(T_{d0} - T_d)}{Q_{max}} - 1\right) \cdot \frac{Q_{max}}{2} + \frac{Q_{max}}{2} + \sum_{i=1}^{N} G_h(T_d - T_i)\left(\frac{1}{2}tanh\left(2x_i - 1\right) + \frac{1}{2}\right) \right\}$$

$$\dot{T_i} = \frac{1}{C_i} \left\{ G_h(T_d - T_i)\left(\frac{1}{2}tanh\left(2x_i - 1\right) + \frac{1}{2}\right) - G_{ui}\left(T_i - \left(278.15 + 8sin\left(\frac{2\pi t}{86400}\right)\right)\right) \right\}$$

$$\dot{x_i} = -50 \cdot \left(\frac{(x-0.5)\cdot(x+0.5)\cdot x_i}{0.0474} \cdot T_\epsilon \right) + (T_{SP} - T_i)$$

4. RESULTS AND DISCUSSION

The performance of the multi-rate method was compared against that of the corresponding single-rate method for the model described in section 3. The performance of the codes with increasing system size is quantified by two parameters, namely the number of individual derivative evaluations and computational time. In all the runs the simulation parameters were $T = 5\times10^4$ and $tol = 10^{-4}$. The obtained solutions were verified by comparing them with those obtained using standard solvers (DASSL in OpenModelica).

4.1 Number of individual derivative evaluations

Figure 1 shows the number of evaluations of individual components of the derivative vector by the multi-rate and single-rate methods with varying system size. It can be seen that the performance of the multi-rate method as regards evaluation of the derivatives, scales up quadratically as opposed to the single-rate method which scales up cubically with the system size. This could be particularly useful when the evaluation of derivatives of the slow components is computationally expensive, e.g., if it involves calculating fluid properties with sophisticated models.

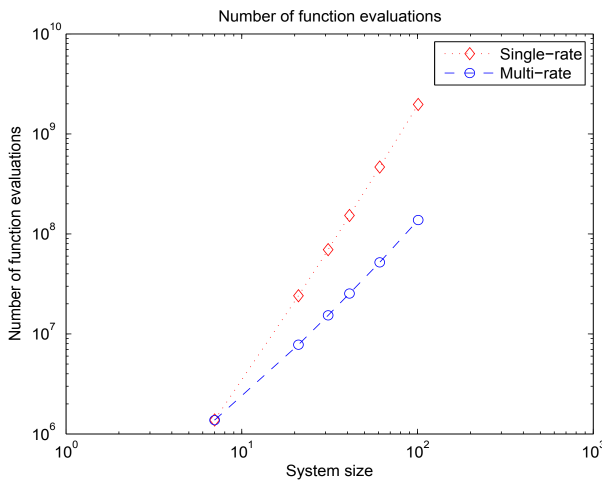

Figure 1: Number of function evaluations vs System size

4.2 CPU time

Figure 2 shows the CPU time taken by both methods for systems of different sizes. In the multi-rate method, partitioning of the system causes some overhead. For very small systems, the time required to solve implicit systems of equations is not very significant, so the multi-rate is slower on the whole. However, as the system size increases, the cost to solve implicit systems of equations scales up in general as $O(N^3)$. The multi-rate method outperforms the single-rate counterpart because it has a much smaller implicit system to solve. Again one can see a quadratic scaling for the multi-rate and cubic for the single-rate.

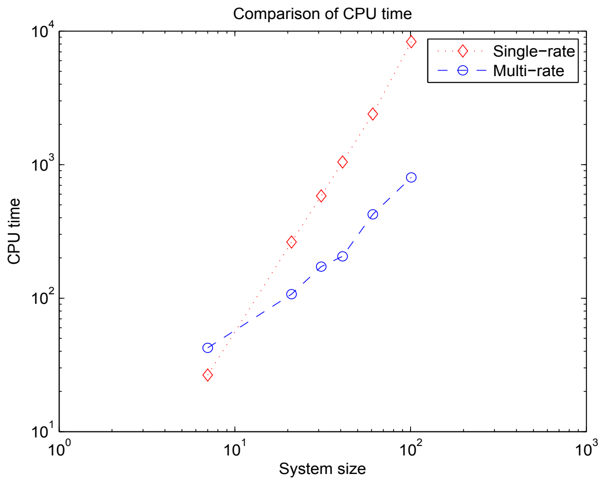

Figure 2: CPU time for simulation run with varying system size

5. CONCLUSIONS AND FUTURE WORK

A multi-rate integration algorithm based on the 2^{nd} order Rosenbrock method was applied to the simulation of an exemplary model of thermal distribution system, that can be easily scaled up to arbitrarily large size. The system is characterized by many sub-units with fast changes happening locally at uncorrelated time instant, which are weakly coupled by a thermal distribution network with a large inertia. The results presented in the paper show very clearly that multi-rate algorithms scale up much better than single-rate ones, overcoming any overhead for large enough sizes. It is possible to conclude that this kind of algorithms have a huge potential for the simulation of large, distributed systems described by EOOOLs, and this motivates further research in this direction.

It would be interesting to study the behavior of other multi-rate methods on this as well as other test problems. The multi-rate method used here has a self-adjusting mechanism based on the error estimates. Other ways of dynamically partitioning the system, similar to cycle analysis and dynamic decoupling [9] are being studied.

In this paper, the model, which contains no algebraic loops, was turned into explicit ODE form manually. The next interesting question to be addressed is then how to efficiently compute the required sub-sets of the derivative vector, from the original DAE formulation of an O-O model. Other questions to be addressed in the future regarding multi-rate algorithms applied to O-O models involve the handling of events, the efficient computation of Jacobians and the possible parallelization of the integration algorithm.

6. REFERENCES

[1] J. Andrus. Stability of a multi-rate method for numerical integration of ode's. *Computers Math. Applic.*, 25:3–14, 1993.

[2] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential–Algebraic Equations*. SIAM, 1996.

[3] F. Cellier and E. Kofman. *Continuous System Simulation*. Springer-Verlag, 2006.

[4] C. Engstler and C. Lubich. Multirate extrapolation methods for differential equations with different time scales. *Computing*, 58(2):173–185, June 1997.

[5] C. Gear and D. Wells. Multirate linear multistep methods. *BIT 24*, pages 484–502, 1984.

[6] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward. Sundials: Suite of nonlinear and differential/algebraic equation solvers. *ACM Trans. Math. Softw.*, 31(3):363–396, Sept. 2005.

[7] A. Kværnø. Stability of multirate runge-kutta schemes. 2000.

[8] A. Kvaernø and P. Rentrop. Low order multirate runge-kutta methods in electric circuit simulation. 1999.

[9] A. V. Papadopoulos and A. Leva. Automating dynamic decoupling in object-oriented modelling and simulation tools. In *5th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools*, pages 37–44, 2013.

[10] J. R. Rice. Split runge-kutta method for simultaneous equations. *JOURNAL OF RESEARCH of the National Bureau of Standards-B. Mathematics and Mathematical Physics*, 64B(5):151–170, July-September 1960.

[11] V. Savcenco. Comparison of the asymptotic stability properties for two multirate strategies. *Journal of Computational and Applied Mathematics*, 220(1-2):508–524, October 2008.

[12] V. Savcenco, W. Hundsdorfer, and J. Verwer. A multirate timesteooing strategy for stiff ordinary differential equations. *BIT: Numerical Mathematics*, 47(1):137–155, March 2007.

[13] B. Thiele, M. Otter, and S.-E. Mattsson. Modular multi-rate and multi-method real time simulation. In *Proc. 10th International Modelica Conference*, pages 381–393, Lund, Sweden, Mar 10-12 2014.