An efficient method for the numerical integration of measured variable dependent ordinary differential equations

C. Rodrigues de Azevedo a, J. Peres b, M. von Stosch a, *

a REQUIMTE, Departamento de Química, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
b LEPABE, Departamento de Engenharia Química, Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal

ARTICLE INFO

Article history:
Received 31 May 2014
Received in revised form 7 October 2014
Accepted 14 October 2014

Keywords:
Dynamic neural networks
Hybrid semi-parametric modeling
Numerical integration
Ordinary differential equations
Differential equations

ABSTRACT

The Ordinary Differential Equations (ODEs) of dynamic models that are used in process monitoring, control or optimization, are not only functions of time and states, but also of measured variables. So far two possibilities for the numerical integration of such ODEs were given: (i) a fixed step size integration schema could be applied, matching the step size to the time instances of the measurements; or (ii) using an adaptive step size method while interpolating the measurements. While fixed step size methods are computationally expensive, the repetitive interpolation of measurements for the application of adaptive step size methods is prone to errors and time prohibitive, especially for great numbers of measured variables.

In this paper, an adaptive step size numerical integration method is proposed and evaluated for dynamic neural network/hybrid semi-parametric models. The method evaluates the ODEs only at time instances at which online measurements are available and adapts the step size according to those time instances. The numerical solution of the ODEs is provided at time instances which are specified by the user, i.e. at time instances of offline measured states. The rationale behind the proposed method is carefully analyzed, and it is demonstrated that its application along with a hybrid model/dynamic neural network model can result into a significant reduction of number of function evaluations, in the studied cases about 50%, while adhering user specified error tolerances for the numerical integration. In addition, the mutual interference between step-size adaption, parameter identification, coping of the neural network and model performance is studied, a fact that other studies have paid little to no attention.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

In the area of chemical and biochemical engineering the list of quantities that can be measured increased significantly in the last decade (Schuegerl, 2001). This development gave rise to the application of data-driven techniques for process modeling. In particular Neural Networks (NNs) have found wide application, since they can easily be applied for fast nonlinear process model development. Most (bio)chemical processes are dynamic, wherefore the standard, rather static, concept of NNs has been extended in many different ways in order to yield a dynamic model (Sinha et al., 2000). One peculiar way is modeling the time varying functions of Ordinary Differential Equations (ODEs) by NNs, resulting in dynamic neural networks (DNN) (Petre et al., 2010). This approach preserves some structural resemblance to the modeling of dynamic systems by first-principles, i.e. a set of ODEs. To an even bigger extend hybrid semi-parametric modeling exploits the available first-principles knowledge, represented in from of parametric models, by combining it with nonparametric models that are identified from data (Thompson and Kramer, 1994, von Stosch et al., 2014). In general, the model structure of DNNs or dynamic hybrid semi-parametric models can be expressed as a set of Ordinary Differential Equations (ODEs), i.e.,

\[
\frac{dx}{dt} = f(x, z, t, w), \quad x(t_0) = x_0,
\]

where \( x \) is a vector of state variables, \( z \) is a vector of measured variables, \( t \) is the time, \( w \) represents all model parameters and \( f \) is a vector of rate functions describing how the states change along with time. In case of the DNN models, these functions, \( f \), are modeled by NNs. In case of hybrid semi-parametric models, these functions are combinations of NNs with fundamental knowledge, as e.g. presented in Oliveira (2004).

Depending on the arguments of \( f \), the model structure can either be a one-step ahead predictor, namely if measured variables, \( z \), are
incorporated or a multi-step ahead predictor if \( f = f(x, t, w) \). In any case, Eq. \((1)\) needs to be integrated to obtain the state estimations \( x \), which is usually done numerically. Whereas in the case of multi-step ahead predictors traditional numerical integration schemes can readily be applied (e.g. the ode45 of the MATLAB toolbox), this is not the case for one-step ahead prediction structures, since the value of the \( z \) might not be measured/available at the time instances at which \( f \) is evaluated by the traditional schemes. Thus, for their integration either a fixed-step size schema, such as the well known Euler or Heun is adopted, fitting the step size to the sampling frequency of \( z \), or some approach is adapted to represent \( z \) as a function of time, e.g. spline or polynomial approximations, and then standard adaptive integration schema is applied, as for multi-step ahead predictors. Both approaches are computationally inefficient. Fixed step size methods allocate the function evaluations on a rigid grid, completely disregarding \( f \)’s curvature. Interpolation becomes computational expensive when repetitively carried out as e.g. for network structure discrimination and network training. Moreover for other types of measured variables, such as those obtained from spectroscopic devices (e.g. Near InfraRed (NIR)), the application of interpolation techniques might not even be appropriate, because (i) the error introduced by the interpolation technique can be large; and (ii) the number of measured variables is large, wherefore interpolations come with a significant time and computational burden. In order to avoid (repetitive) interpolations, fixed step size methodologies find application in which the integration step-size is determined by the sampling frequency of the measured variables \( z \) (Schubert et al., 1994; van Can et al., 1996; von Stosch et al., 2012). However, it is well known that fixed step size methodologies are, likewise, not computationally efficient.

The sampling rate of online measured variables, \( z \), is in most (bio)chemical processes both time constant and frequent. In contrast, the proportions of the divisions made for each step in most standard numerical integration schema, such as Runge–Kutta Fehlberg (4–5) or Dormand–Price (4–5) (Gladwell et al., 2003, Ashino and Vaillancourt, 2009) are irregular. Therefore their direct application along with the frequently sampled online measurements is hindered. In the following the intermeshing use of four integration methods, all of which having different but regular step proportions, is proposed. Since their application comes with different requirements on two consecutive steps, the time interval of the integration is subdivided into different sections, as explained in the following. For each of these sections a different integration schema is applied. A schema for the control of the integration error is proposed, in which the adaptation of the step size is carried out with respect to the given sampling time instances of the measurements.

### 2. Methodology

Two conditions arise for the numerical integration:

#### 2.1. Condition 1 (C1)

For parameter identification or for model performance assessment, Eq. \((1)\) needs to be solved at those time instances at which state measurements were made. Most numerical integration methods are capable of this. The methods usually either integrate from time instance to time instance or they integrate over the whole time interval and interpolate thereafter (Shampine, 1987), the post-interpolation usually being computationally less expensive.

![Fig. 1.](image)

#### 2.2. Condition 2 (C2)

For the numerical integration of Eq. \((1)\), the measured variables, \( z \), needs to be provided at those time instances \( t_z \) at which the function \( f(x(t_z), z(t_z), t_z, w) \) is evaluated. Even though \( z \) is measured with high frequency (implied through its choice as input) it might not be available at \( t_z \). To obtain \( z(t_z) \), Hermite cubic interpolations or smoothing spline interpolation can be applied for some types of measured variables, but during the network structure discrimination, the network training or process optimization the repetitive interpolation becomes computationally expensive. For other types of measured variables, such as those obtained from spectroscopic devices (e.g. Near InfraRed (NIR)), the application of interpolation techniques might not even be appropriate, because (i) the error introduced by the interpolation technique can be large; and (ii) the number of measured variables is large, wherefore interpolations come with a significant time and computational burden. In order to avoid (repetitive) interpolations, fixed step size methodologies find application in which the integration step-size is determined by the sampling frequency of the measured variables \( z \) (Schubert et al., 1994; van Can et al., 1996; von Stosch et al., 2012). However, it is well known that fixed step size methodologies are, likewise, not computationally efficient.

The sampling rate of online measured variables, \( z \), is in most (bio)chemical processes both time constant and frequent. In contrast, the proportions of the divisions made for each step in most standard numerical integration schema, such as Runge–Kutta Fehlberg (4–5) or Dormand–Price (4–5) (Gladwell et al., 2003, Ashino and Vaillancourt, 2009) are irregular. Therefore their direct application along with the frequently sampled online measurements is hindered. In the following the intermeshing use of four integration methods, all of which having different but regular step proportions, is proposed. Since their application comes with different requirements on two consecutive steps, the time interval of the integration is subdivided into different sections, as explained in the following. For each of these sections a different integration schema is applied. A schema for the control of the integration error is proposed, in which the adaptation of the step size is carried out with respect to the given sampling time instances of the measurements.

#### 2.3. Time instance analysis and classification of equal property sections

The proposed numerical integration method has to obey to two timelines, \((1)\) the timeline at which the solution of the ODE is desired, \( t_{x \text{mes}} \), i.e. the time instances at which infrequent offline measurements have been obtained and at which the solution of the ODE is compared to the measurements (e.g. for parameter identification see Section 3.1), see Fig. 1A; and \((2)\) the timeline of the frequent online measurements, \( t_z \), whose entries determine the minimal possible step sizes that can be chosen, see Fig. 1B. The sampling frequency of \( z(t_z) \) is typically constant throughout the experiment, such that the increments between the time instances contained in \( t_z \) do all have the same size. However, online measurements might not exist at the time instances of the offline measurements at which the solution is desired, i.e. the time instance elements of \( t_{x \text{mes}} \) do not have to be contained in \( t_z \), see Fig. 1. In order to obey to these given conditions, it is proposed to divide the timeline into distinct sections such that for the numerical integration of each section a different methodology can be applied. In particular, the time vector \( t_{\text{mix}} \) (which contains all the sampling times of \( t_{x \text{mes}} \) & \( t_z \)) is divided into the following categories, which are represented in Fig. 1:

- **S1** Sections with (at least) two equal sequential steps; (equal in the context of numerics meaning up to eight digits behind the point, \( 10^{-8} \)).
S2) Sections in which (at least) two sequential steps are not equal; and
S3) Steps for which no online measurements are available at the end of the step.

For each section different numerical integration methods can be used, as described in the following.

2.4. The integration methods for sections with more than two equal sequential steps (S1)

For sections of the integration interval with more than two equal sequential steps, a Runge–Kutta method of order 4 is applied, named here RK4. We reformulated the classic Runge–Kutta method of order 4 in such way that it does not require the calculation of the ODEs at the middle of each step, but instead two sequential steps are performed at once, i.e. using the relation \( h = 2 \cdot h_i \). The RK4 scheme then reads

\[
\begin{align*}
k_1 &= h_i \cdot f(t_n, x_n, z(t_n)) \\
k_2 &= h_i \cdot f(t_n + h_i, x_n + k_1, z(t_n + h_i)) \\
k_3 &= 2 \cdot h_i \cdot f(t_n + h_i, x_n + k_2, z(t_n + h_i)) \\
k_4 &= 2 \cdot h_i \cdot f(t_n + 2 \cdot h_i, x_n + k_3, z(t_n + 2 \cdot h_i)) \\
x_{n+1} &= x_n + 1/6 \cdot (k_1 + 4 \cdot k_2 + 2 \cdot k_3 + k_4)
\end{align*}
\]  

(2)

where \( h_i \) is the step size, \( n \) is the current time instance and \( k_i, i = 1...4 \) are the incremental coefficients.

For the estimation of the local truncation error associated with the solution of the RK4 schema, a Runge–Kutta method of third order (called here RK3) was adopted, since only one additional function evaluation per step is required. Utilizing the same step size relation \( h = 2 \cdot h_i \), the equation of the obtained local truncation error estimate (LEE) reads

\[
LEE_{RK3,RK4} = (-4 \cdot k_2 + 2 \cdot k_3 + k_4 - k_3^2)/6,
\]

(3)

where \( k_3^2 = 2 \cdot h_i \cdot f(t_n + 2 \cdot h_i, x_n - 2 \cdot k_1 + 4 \cdot k_2, z(t_n + 2 \cdot h_i)) \).

2.5. The integration method for sections with two unequal sequential steps (S2)

In case that two sequential steps do not have equal lengths, the RK4 method cannot be applied. Instead it is proposed to employ the Heun numerical integration method (Ashino and Vaillancourt, 2009), also known as the Euler modified method, since the proportions of two sequential steps have no relevance for this method and since the method is of second order. The local truncation error of the Heun scheme is approximated by the calculation of a LEE utilizing the Euler forward method, resulting in

\[
LEE_{Heun, forw} = (k_1 - k_2)/2.
\]

(4)

where \( k_1 \) and \( k_2 \) are the incremental coefficients, which are identical to the ones of the RK4 method, see eq. (2).

Since the Heun method makes two function evaluations per step (thus for two sequential steps four function evaluations are performed) the number of function evaluations for two consecutive steps are the same for the Heun and the RK4 method. Thus the question arises whether the solution after two consecutive steps is generally better using the RK4 or the Heun method. All in all it can be expected that higher order methods (RK4) provide a solution with a lower integration error than lower order methods (Heun or Euler forward), but since for the considered scenario the smallest step size that can be used by the RK4 method is two times the smallest possible step size of the Heun method, this expectation will in limit not hold true. Hence, the solutions obtained with the RK4 method were compared to those of the Heun method studying different fixed step size values for \( h_i \).

2.6. Steps for which no online measurements are available at the end of the step (S3)

In case that the function evaluation of \( k_2 \) cannot be accomplished because no online measured data are available at the respective time instance, an Euler forward method is adopted for this step. Since the error which is due to the numerical integration can be assumed to be greater for the Euler forward method than the one obtained with other methods, the solution is not used to continue the numerical integration but instead the numerical integration is continued using the starting point of this step and the endpoint of the following step. A LEE is not calculated in this case.

2.7. Restricted step size control

The integration can be expedited by augmentation of the step size in areas in which the time slope is relatively constant, i.e. the curvature is low. Typically, LEEs are employed to control the step size, where the step-size is some function of the LEEs and some user-defined tolerances (Shampine, 2005), e.g.:

\[
|LEEmethod| \leq \max(\text{RelTol} \times |x_n+1(t_n+1)|), \text{AbsTol},
\]

(5)

where \( \text{AbsTol} \) is an user given scalar absolute tolerance value, \( \text{RelTol} \) is an user given vector of relative tolerance values and \( \times \) signifies an element wise multiplication. It has been shown that by using such
an approach for step-size control, the numerical integration is typically stable (Lapidus and Seinfeld, 1971). For the presented scenario the step size cannot be arbitrarily augmented or shortened since it has to comply with the given time instances. The time instances divide the timeline into time increments. Thus instead of adapting the step size instances. The time instances divide the timeline into time increments. 

\[ \Delta h \]

varying the number of time increments, \( \Delta h \) (the relation between \( h_1 \) and \( \Delta h \) is presented in Fig. 2a for an example). By nature the smallest possible \( \Delta h \) is one, while the largest \( \Delta h \) can be chosen by the user (10 in this study). The schema displayed in Fig. 2b is applied to control the step-size \( h_1 \) when the steps comprised by \( \Delta h \) are not of same size. At any given step, the LEE is calculated and compared to the specified tolerances. If the value of the LEE is lower than the tolerances then the step-size of this step is accepted, i.e. the ODE is integrated with the Heun method, and for the subsequent step the step-size is enlarged. In case that the steps have the same size the schema presented in Fig. 2c is applied to control the step-size \( h_1 \). At any given step, the LEEs for both methods are calculated with the same step-size \( h_1 \) and then compared, i.e.: \( LEE_{\text{Heun}} < LEE_{\text{RK4}} \), see Fig. 2c. This condition has been introduced to reduce the numerical integration error when the smallest possible step size (defined by the sampling frequency of the online data) is used and it is a result of the observation made in Sections 2.6. and 4.1.1. If the \( LEE_{\text{RK4}} \) is lower, then the following routine is similar to the one described above except that the integration is carried out with the RK4 method. In case the \( LEE_{\text{RK4}} \) is greater, the step-size is reduced as far as possible and the LEEs are calculated for the new step-size. If the lowest possible step-size is reached, the integration is carried out with the Heun method and the routine described in Fig. 2b is applied.

Whenever \( \Delta h \) reaches out to a region of another section, then \( \Delta h \) is reduced in such a way as to provide the solution at the boarder of the sections. In case that the smallest possible step size is reached and that the LEE exceeds the user defined tolerances, a warning message is printed suggesting the user to lower the sampling frequency in upcoming experiments. Note that for the Euler forward schema \( \Delta h = 1 \).

2.8. Stability and truncation errors

Per se, the employed Runge Kutta, Heun and Euler forward integration methods are stable provided that the step-size is sufficiently small (Lapidus and Seinfeld, 1971). The step-size is, as described in the section before, controlled by keeping the values of the LEEs below those of the absolute and relative tolerances, as normally done (Lapidus and Seinfeld, 1971; Shampine, 2005). Stability is such indirectly controlled by the user through the choice of absolute and relative tolerance values. However, in the method proposed here the step-size has to comply with the timeline also given by the user, i.e. \( \epsilon_{\text{max}} \), as described before. The step-size can therefore not be lower than the difference between two neighboring time instances. This restriction might result in the exceeding of the tolerance values and subsequently into a loss of stability. As mentioned before, the user is informed if the tolerance values are exceeded due to the step-size restrictions.

To better understand the impact of the step-size on the stability and integration error, the linear ODE \( \frac{dy}{dt} = \lambda \cdot y \) is used. This linear system can be adequate to analyze also the stability of nonlinear systems, given that the \( k_i \)'s converge to some constant values as the step-size decreases (Lapidus and Seinfeld, 1971). For one step with the step size \( h_1 \) (i.e. the total length of the step is \( 2 \cdot h_1 \)) the RK4 method yields

\[
y_{\text{RK4}}(t_0 + 2 \cdot h_1) = y(t_0) \cdot \left( 2/3 \cdot h_1^2 \cdot \lambda^3 + 4/3 \cdot h_1^3 \cdot \lambda^3 + 2 \cdot h_1^2 \cdot \lambda^2 + 2 \cdot h_1 \cdot \lambda + 1 \right).
\]

Fig. 3. (A) The characteristic root over \( \lambda \cdot h_1 \) for one step with step-size \( h_1 \) of the RK4 method (dark-gray continuous line); for two steps with step-size \( h_1 \) of the Heun method (gray dashed line); for one step with step-size \( h_1 \) of the Heun method (light gray continuous line); (B) Same content plotted as in A, but zoom on the y-axis. (C) Local truncation error over \( \lambda \cdot h_1 \) for one step with step-size \( h_1 \) of the RK4 method (dark-gray continuous line); for two steps with step-size \( h_1 \) of the Heun method (gray dashed line); for one step with step-size \( h_1 \) of the Heun method (light gray continuous line); (D) Local truncation error and estimates over \( \lambda \cdot h_1 \) for one step with step-size \( h_1 \) of the RK4 method (dark-gray continuous line); the respective LEE (dark-gray dashed-dotted line); for one step with step-size \( h_1 \) of the Heun method (light gray continuous line) and the respective LEE (light gray dashed-dotted line).
whereas two steps are needed with the Heun method to reach the same point, i.e.,
\[ y_{\text{Heun}}(t_0 + 2 \cdot h) = y(t_0) \left( 1 + \frac{1}{4} h^2 \cdot \lambda^2 + \frac{1}{3} h^2 \cdot \lambda^3 + 2 \cdot h^2 \cdot \lambda^3 + 2 \cdot h \cdot \lambda + 1 \right), \]
(7)
where \( t_0 \) is the starting time point of the integration. Absolute stability of the methods is given when the absolute value of the terms in the brackets of Eqs. (6) and (7), i.e., the characteristic root, are smaller than one. For different combinations of \( \lambda \cdot h_i \) values the stability is displayed in Fig. 3 A and B. The impact of the system of ODEs on the stability is given by \( \lambda \). It can be seen that the Heun method for two steps (with step-size \( h_1 \)) or one step with step size \( h_1 \) is stable for a wider range of \( \lambda \cdot h_1 \) values than the RK4 method for one step with the same step size. This could be expected since lowest possible step size of the Heun is half that of the RK4 method. Further, it can be observed that the values of the characteristic roots for positive \( \lambda \cdot h_1 \) values increase faster than in the case of negative values. This is because ODEs in which the real part of the eigenvalues is smaller than zero are inherently stable (Lapidus and Seinfeld, 1971).

The integration error can be obtained subtracting Eqs. (6) and (7) from the analytic solution, \( y(t_0 + 2 \cdot h) = y(t_0) \cdot e^{\lambda(2 \cdot h)}, \) which yields in case of the RK4 method for one step
\[ E_{\text{RK4,1step}} = y(t_0) \cdot \left( \frac{e^{\lambda(2 \cdot h_1)}}{2/3 \cdot h_1^2 \cdot \lambda^4 + 4/3 \cdot h_1^3 \cdot \lambda^3 + 2 \cdot h_1^2 \cdot \lambda^2 + 2 \cdot h_1 \cdot \lambda + 1} \right), \]
in case of the Heun method for two steps
\[ E_{\text{Heun,2steps}} = y(t_0) \cdot \left( \frac{e^{\lambda(2 \cdot h_1)}}{\frac{1}{1/4} \cdot h_1^2 \cdot \lambda^4 + h_1^3 \cdot \lambda^3 + 2 \cdot h_1^2 \cdot \lambda^2 + 2 \cdot h_1 \cdot \lambda + 1} \right), \]
and in case of the Heun method for one step
\[ E_{\text{Heun,1step}} = y(t_0) \cdot \left( \frac{e^{\lambda(h_1)}}{\frac{1}{1/2} \cdot h_1^2 \cdot \lambda^2 + h_1 \cdot \lambda + 1} \right). \]
(8)

The integration error is thus amplified or damped by the value of \( y(t_0) \). The differences for different values of \( \lambda \cdot h_1 \) are shown in Fig. 3C. It can be seen that for positive \( \lambda \cdot h_1 \) values the RK4 method produces lower integration errors, whereas in the case of negative values from approximately –0.93 on the Heun method produces lower integration errors. These errors are approximated by the respective LEE methods, shown in Fig. 3D. It can be seen that the LEEs are upper bounds for the local truncation errors in the region where the integration methods are stable. The truncation error and the LEE values of the Heun method are for small step-size values always above those of the RK4 method. It can further be seen that for positive \( \lambda \cdot h_1 \) values, which are in the stable range, the RK4 method produces lower truncation errors and LEEs as the Heun method. In case, the \( \lambda \cdot h_1 \) values are lower than –0.93, the truncation error and LEE of the Heun method are lower than those of the RK4 method. Thus, given the case that the step-size cannot be reduced further and that the LEE of the Heun methods is lower than that of the RK4 method, then the integration with the Heun method produces lower truncation errors. Therefore, in this case the integration should be carried out with the Heun method, as proposed in the schema shown in Fig. 2.

2.9. Evaluation criteria

The solutions obtained through the numerical integration methodologies are compared using the following criteria, which address the accuracy and the related computational cost.

The True Local Error (TLE) can be calculated when the analytical solution of the studied ODEs system is known
\[ \text{TLE}_i(t) = x_{\text{analytical},i}(t) - x_{\text{analytical},i}(t), \]
(11)
where the index \( i \) refers to each state and analytical/numerical specifies how the solution was obtained. In order to have a straightforward criteria for tabular comparison, a scalar True Global Error (TGE) can be calculated given the analytical solution of the studied ODEs system is known
\[ \text{TGE}_i = \sum (x_{\text{analytical}}(t_{\text{offline}}) - x_{\text{analytical}}(t_{\text{offline}}))^2. \]
(12)

Similarly, a straightforward criteria for tabular comparison of the estimated numerical integration error is provided through the Global Error Estimation (GEE) which is calculated applying a sum of squares to the LEE_{RK4,3k} or LEE_{Heun,1step}, i.e., Eqs. (3) and (4), respectively, resulting in the following error criteria:
\[ \text{GEE}_i = \sum (\text{LEE}_i(t_{\text{offline}}))^2. \]
(13)

The second criteria type, accounting for the computational cost, reflects how efficient a method is. Maybe the most accurate manner to evaluate the computational cost of an integration is through the number of function evaluations/function calls that a scheme made during the integration (Krogh, 1973), because the number of function calls is independent of the used computer and eventual parallel process which might run on the machine.

3. DNN/hybrid modeling structure, parameter identification and model discrimination methods

The set of ODEs, Eq. (1), describes the general form of dynamic hybrid semi-parametric models, where \( f \) is a vector of rate functions. In case that no knowledge about the rate functions is available, the functions \( f \) can be modeled using Neural Networks and in this particular case the hybrid model structure is identical to structure of the DNN model described by Rovithakis and Christodoulou (1995) and Petre et al. (2010). Whenever knowledge about the rate functions is available, this knowledge can be incorporated and only the remaining unknown variables are estimated by the NN as e.g. described in detail in Oliveira (2004). In order to train the neural network and to discriminate the best performing network topology the procedure described in the following can be applied.

3.1. Training – parameter identification

For the training/parameter identification the available data are split into two sets, a training and a validation set. Using the training set the weights are identified minimizing a weighted least square function of the residual between the simulated offline measured state variables \( x_{\text{mes}} \) and the model estimates weighted by the respective variance \( \sigma_x \), i.e.,
\[ \text{min} \left( E \right) = \text{min}_w \left( \frac{1}{2} \cdot \sum_{i=1}^{n} \frac{(x_{\text{mes},i} - x_{\text{est},i})^2}{\sigma_x} \right). \]
(14)
for all \( n \) instances measured. A gradient based method is used for the identification, namely a Levenberg–Marquardt method, i.e. the lsqnonlin function of the Matlab Optimization toolbox was used. The gradients are derived using the sensitivities approach (Oliveira, 2004; von Stosch et al., 2011), i.e.
\[ \frac{d}{dt} \frac{dx}{dw} = \frac{dx}{dt} \frac{df}{dw}, \]
(15)
where \( x \) and \( t \) are independent of \( w \). Since also \( x_0 \) is independent of \( w \), \( dx_0/dw = 0 \). Eq. (15) is integrated along with Eq. (1). The training is stopped when the weighted least square error calculated for the validation set starts to increase, i.e. early-stopping is used to avoid over-fitting (Simutis and Luebbert, 1997). In order to escape from local minima during the identification, several random weight initializations (10 in the cases presented below) are
Table 1: True Global Errors and Global Error Estimations values calculated for the solutions at $t_{off} = [0.0, 0.1, 0.2, ..., 1.9, 2]$. Analytically and with fixed step size methods using a step size of $h = 10^{-4}$.

<table>
<thead>
<tr>
<th>$(h = 10^{-4})$</th>
<th>TGE Forward</th>
<th>TGE Heun</th>
<th>GEE$_{Forward}$</th>
<th>TGE RK4</th>
<th>GEE$_{RK4}$</th>
<th>GEE$_{BM}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_1$</td>
<td>4.31e−012</td>
<td>3.61e−018</td>
<td>4.31e−012</td>
<td>1.98e−027</td>
<td>3.65e−022</td>
<td></td>
</tr>
<tr>
<td>$y_2$</td>
<td>1.26e−012</td>
<td>3.61e−018</td>
<td>1.27e−012</td>
<td>1.46e−027</td>
<td>3.65e−022</td>
<td></td>
</tr>
<tr>
<td>$y_3$</td>
<td>1.26e−012</td>
<td>3.61e−018</td>
<td>1.27e−012</td>
<td>1.46e−027</td>
<td>3.65e−022</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: True Global Errors and Global Error Estimations values calculated for the solutions at $t_{off} = [0.0, 0.1, 0.2, ..., 1.9, 2]$ analytically and with fixed step size methods using a step size of $h = 10^{-3}$.

<table>
<thead>
<tr>
<th>$(h = 10^{-3})$</th>
<th>TGE Forward</th>
<th>TGE Heun</th>
<th>GEE$_{Forward}$</th>
<th>TGE RK4</th>
<th>GEE$_{RK4}$</th>
<th>GEE$_{BM}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_1$</td>
<td>4.09e−010</td>
<td>3.38e−014</td>
<td>4.14e−014</td>
<td>1.69e−019</td>
<td>4.37e−016</td>
<td></td>
</tr>
<tr>
<td>$y_2$</td>
<td>1.07e−010</td>
<td>3.87e−014</td>
<td>1.11e−010</td>
<td>1.69e−019</td>
<td>4.37e−016</td>
<td></td>
</tr>
<tr>
<td>$y_3$</td>
<td>1.07e−010</td>
<td>3.87e−014</td>
<td>1.11e−010</td>
<td>1.69e−019</td>
<td>4.37e−016</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: True Global Errors and Global Error Estimations values calculated for the solutions at $t_{off} = [0.0, 0.1, 0.2, ..., 1.9, 2]$ analytically and with fixed step size methods using a step size of $h = 10^{-2}$.

<table>
<thead>
<tr>
<th>$(h = 10^{-2})$</th>
<th>TGE Forward</th>
<th>TGE Heun</th>
<th>GEE$_{Forward}$</th>
<th>TGE RK4</th>
<th>GEE$_{RK4}$</th>
<th>GEE$_{BM}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_1$</td>
<td>3.11e−008</td>
<td>1.39e−009</td>
<td>3.65e−008</td>
<td>9.20e−011</td>
<td>1.45e−009</td>
<td></td>
</tr>
<tr>
<td>$y_2$</td>
<td>1.98e−009</td>
<td>1.39e−009</td>
<td>6.69e−009</td>
<td>9.20e−011</td>
<td>1.45e−009</td>
<td></td>
</tr>
<tr>
<td>$y_3$</td>
<td>1.98e−009</td>
<td>1.55e−009</td>
<td>7.03e−009</td>
<td>8.74e−009</td>
<td>2.04e−002</td>
<td></td>
</tr>
</tbody>
</table>

performed and the best parameters are chosen using cross-validation (Simutis and Luebbert, 1997, Oliveira, 2004, von Stosch et al., 2011).

3.2. Model discrimination

Several network topologies are typically tested, varying the number of nodes in the hidden layer of the neural network, as well as the node transfer functions and, if necessary, the number of hidden layers. If available, the performance of the different model structures is tested on a third data set, the test set. Else the structures are evaluated employing evaluation criteria. In the cases presented in the following, the best performing model structure was chosen using the Bayesian information criteria (BIC), which balances the fit against the complexity of the model (Peres et al., 2008). This criteria is calculated for the training and validation set.

4. Results and discussion

4.1. Case study I – a simulated system

A mildly stiff ODEs system, presented in the following, was chosen because its analytical solution is known, wherefore the numerical solution can be assessed in comparison to the analytical one. This system of linear first order ODEs, has been proposed by Lapidus and Seinfeld (1971)

$$\frac{dy}{dt} = \begin{bmatrix} -0.1 & -49.9 & 0 \\ 0 & -50 & 0 \\ 0 & 70 & -120 \end{bmatrix} \cdot y, \quad (16)$$

where $0 < t < 2$ and $y_0 = [2, 1, 2]^T$. For a generic initial condition $y_{\text{generic}}(t=0) = [y_{0.1}, y_{0.2}, y_{0.3}]^T$, the analytical solution is

$$y(t) = \left[ \begin{array}{c} (y_{0.1} - y_{0.2}) \cdot e^{-0.1 \cdot t} + y_{0.2} \cdot e^{-50 \cdot t} \\ y_{0.2} \cdot e^{-50 \cdot t} - (y_{0.2} - y_{0.3}) \cdot e^{-120 \cdot t} \end{array} \right]. \quad (17)$$

4.1.1. Analysis of the fixed step-size integration methods and local/ global error estimations

The integration methods Euler forward, Heun and RK4 were separately analyzed (in “fixed step size mode”) in order to assess the properties of the proposed numerical integration methodology in relation to different values of $h_1$. In Tables 1–4 the estimated errors (GEE) and the true errors (TGE) are presented for different values of $h_1$. It can be observed (Tables and 2) that for sufficiently small step sizes ($h = 10^{-2}$ or $h = 10^{-3}$) the RK4 performs better than the Heun method, even though the step size of the RK4 is two times the step size of the Heun method. This could be expected because the RK4 method is of 4th order, producing local truncation errors of the 5th order ($O(h^5)$), while the Heun method is of order two (local truncation errors of order $O(h^3)$). Thus, if the step size is sufficiently small, then the RK4 method is preferable since with the same number of function evaluations the solution is corrupted with a much lower numerical integration error, see e.g. Section 2.6. However, when increasing the step size (Tables 1–3) it can be observed that the differences between the errors of the RK4 and of the Heun method diminish. For $h_1 = 10^{-2}$, it can even be observed that for the third state, $y_3$, lower associated GEE values are obtained for the Heun method than for the RK4, see Table 3. This observation is visually confirmed when looking at Fig. 4, wherein the time courses of the local errors (TLE and LEE) are plotted for $y_3$. Therefore it is concluded that the RK4 method performs poorer than the Heun method when a
certain threshold step size value is reached, which is in accordance with the findings made in section 2.6. Hence the proposed numerical methodology, switches the numerical integration schema in case that the step size cannot be further lowered and that the error estimate of the Heun method is better than the one of the RK4. In case that a too large step size is chosen both methods are no longer stable (Table 4), as expected (Lapidus and Seinfeld, 1971).

4.1.2. The DNN/hybrid model structure

Hybrid models in general find application if the system of ODEs is not completely known. Such a scenario is simulated by choosing a very simple structured hybrid model, which as a matter of fact is identical to a DNN model structure (Rovithakis and Christodoulou, 1995; Petre et al., 2010), i.e.

\[
\frac{dy_{HM}}{dt} = \begin{bmatrix} r_1(y_{online,mes}, W) \\ r_2(y_{online,mes}, W) \\ r_3(y_{online,mes}, W) \end{bmatrix},
\]

where \( r_1, r_2 \) and \( r_3 \) are all functions of the inputs \( y_{online,mes} \), which simulate online measurements of the state variables (Note that the vector that contains the functions \( r_1, r_2 \) and \( r_3 \) is represented simply by \( f \) in Eq. (1)). Since it is assumed that the underlying system is unknown, \( r_1, r_2 \), and \( r_3 \) are represented by Artificial Neural Networks (ANNs) wherein the parameters \( w \) represent the weights of the connections between the three layers of the ANN. For the identification of the weights, Eq. (18) is integrated with the proposed “variable step size” method and the parameter identification methodology described above was employed. The training and validation data were generated with the analytical solution, Eq. (17), by (i) varying the values of \( y_0 \) by 5%; (ii) recording simulated offline measured state variables, \( y_{offline,mes} \), using a constant sampling frequency of \( 10^{-1} \), and (iii) recording simulated online measured variables \( y_{online,mes} \) using three constant sampling frequencies \( 10^{-1}, 10^{-2} \) and \( 10^{-3} \). An additional test data set is generated with \( y_0 \) using the same sample frequencies for \( y_{offline,mes} \) and \( y_{online,mes} \) such that the hybrid model performance can be analyzed with respect to the observations made before. The best performing model structure was discriminated with the above described methodology. The three nodes of the input and output layer have linear transfer functions and for the hidden layer a single hyperbolic tangent transfer function node was chosen.

4.1.3. Results obtained for the integration of the DNN/hybrid model

In Tables 5–7 global errors are presented for the test data, which were obtained integrating Eq. (18) using the proposed integration method (relative tolerance \( 10^{-5} \) and absolute tolerance \( 10^{-6} \)) with the different underlying online sampling frequencies \( 10^{-1}, 10^{-2} \) and \( 10^{-3} \) respectively. In case of the lowest sampling frequency (Table 5), which is identical to the lowest step size studied before (Table 4), the results in terms of TGE obtained for the integrated hybrid model ODE system (Table 5) do not show the same loss of stability. What is the reason? Obviously, the ODEs are different, i.e. Eq. 18 differs from Eq. 16. However, ultimately the ODE system should approximate the original system with desired accuracy within certain bounds. To approximate the system, the model weights, \( w \), are changed such that the residual between the experimental value and the model estimate is minimized, Eq. (14). The residual depends on the model estimations \( x(t_{end}) \), which are obtained by integrating Eq. (18) with the proposed method. A loss in stability caused by too great step-sizes results in oscillating/increasing values of \( x(t_{end}) \) and thus great residual values. The residual is used to identify the parameter values, and thus the parameters are identified such that they can account for unsuitable great step-sizes. On one hand this means that the numerical integration error can partially be accounted for by the ANN, but on the other hand the underlying “true” ODE system, i.e. Eq. (16), is no longer mimicked by Eq. (18). The parameter identification is also the reason why the TGE values obtained for a sampling frequency of \( 10^{-1} \) are lower than those obtained for a sampling frequency of \( 10^{-2} \). Even with a sufficiently small sampling frequency \( 10^{-3} \) Table 7 it can be observed that the TGE values are significantly higher than the GEE values, which is contrary to the observations made in Section 4.1.1. The trend that would be expected can be observed for the GEE, for which the values are lower for a decreasing sampling frequency (Tables 5–7). The GEE provide an upper bound to the error resulting from the numerical integration, wherefore the remaining error must stem from the parameter/model mismatch. Thus, it follows that the error which is due to the model mismatch, in case that the sampling frequency is sufficiently small, is greater than the error which is due to the numerical integration.

A fact that was expected but also strikes, is that the number of function evaluations for the proposed method is only lower than the number of function evaluations for the Heun fixed step size method when the sampling frequency is sufficiently small (see headers of Tables 5–7). Obviously the number of function evaluations in case of the proposed method depends on many factors, but as a rule of thumb it might be stated that the sampling frequency is sufficiently small if the number of function evaluations with a Heun fixed step size method is significantly greater than the number obtained with the proposed method (in which case the error due to the numerical integration is much lower than the error due to model parameter or structural mismatch).

However, a sufficiently small sampling frequency in this sense does not guarantee that the user specified integration tolerances are met. For instance it can be seen in Fig. 5, that both error tolerances

| Table 6 |
| States over Global error values calculated as the difference between the solutions of \( y_{true}(t_{fin}) \) and \( y_{ANN}(t_{fin}) \) at \( t_{fin} = [0, 0.1, 0.2, ... 1.9, 2] \). Number of function evaluations made 533 in relation to 402 required for a fixed step size Heun method; Online sampling frequency \( h = 10^{-2} \).
<table>
<thead>
<tr>
<th>( (h = 10^{-2}) )</th>
<th>TGE</th>
<th>GEE_{true, faw}</th>
<th>GEE_{REA, RET1}</th>
</tr>
</thead>
<tbody>
<tr>
<td>y1</td>
<td>1.01E-003</td>
<td>8.69E-003</td>
<td>-</td>
</tr>
<tr>
<td>y2</td>
<td>1.41E-003</td>
<td>8.72E-003</td>
<td>-</td>
</tr>
<tr>
<td>y3</td>
<td>7.59E-003</td>
<td>3.33E-002</td>
<td>-</td>
</tr>
</tbody>
</table>

| Table 7 |
| States over Global error values calculated as the difference between the solutions of \( y_{true}(t_{fin}) \) and \( y_{ANN}(t_{fin}) \) at \( t_{fin} = [0, 0.1, 0.2, ... 1.9, 2] \). Number of function evaluations made 1443 compared to 402 required for a fixed step size Heun method; Online sampling frequency \( h = 10^{-3} \).
<table>
<thead>
<tr>
<th>( (h = 10^{-3}) )</th>
<th>TGE</th>
<th>GEE_{true, faw}</th>
<th>GEE_{REA, RET1}</th>
</tr>
</thead>
<tbody>
<tr>
<td>y1</td>
<td>5.58E-005</td>
<td>2.47E-008</td>
<td>1.852E-011</td>
</tr>
<tr>
<td>y2</td>
<td>3.28E-005</td>
<td>2.46E-008</td>
<td>1.841E-011</td>
</tr>
<tr>
<td>y3</td>
<td>2.97E-003</td>
<td>9.91E-008</td>
<td>7.430E-011</td>
</tr>
</tbody>
</table>
are violated around a time of 0.1, since the step size cannot be reduced below the sampling frequency of $10^{-3}$. The user must be warned about this circumstance, so that he can take action if desired. Else it can be seen that the step size is reduced when the curvature is large and when approaching the time events at which the solution has to be provided (for comparison with the offline measured states). At sections where the curvature is low, the step size is successfully enlarged.

4.2. Case study II – batch fermentations of Bordetella pertussis

The objective in this case is to demonstrate that the proposed methodology reduces the number of function evaluations, when applied for the development of DNNs/hybrid models using typical experimental bioreactor data. The experimental study published by Soons et al. (2008) provides the bases. Eight cultivations were carried out in batch mode and variations to the process conditions, such as in pH, temperature and dissolved oxygen were applied, as reported in Soons et al. (2008). The process data consist of infrequent, sparse measured concentrations of lactate, glutamate and biomass, respectively and $r_1$–$r_3$ are outputs of the neural network (Note that the vector that contains the functions $r_1$, $r_2$ and $r_3$ is represented simply by $f(\cdot)$ in Eq. (1)). The inputs of the neural network comprise the estimated concentrations and the online measured variables, i.e.: $r_i = r_i(L_{\text{C}}^{\text{DNN}}, G_{\text{U}}^{\text{DNN}}, X^{\text{DNN}}, pH, Temp, DO)\forall i = 1,\ldots,3$. The model estimates are obtained by integrating Eq. (19) either with the proposed “variable step size” method or the traditional fixed step-size method. The parameter identification and model discrimination was carried out as described in Section 3 above. The required sets of training and validation data consist of data of 5 and 2 batches, respectively. A test set that comprised data of 1 batch was used in addition to assess the generalization properties of the models. Network structures with one to three nodes in the hidden layer were studied. The best performing model comprised a network with one node and the transfer functions of the nodes in the hidden layer were tangential hyperbolic.

4.2.2. Results

A significant decrease of the number of function evaluations when integrating Eq. (19) with the proposed integration method can be observed, see Table 8. On average the number of function evaluations is 48% lower than the number of fixed-step size evaluations. During the identification of the network weights the sensitivities equations are integrated along with the model equations, and a decrease in the number of function evaluations due to the proposed integration method is not always observable, see e.g. Fig. 6. Therein it can be seen that the number of function evaluations during the first iterations shows a sharp peak, whereupon it drops below the number required for a fixed step-size Euler forward integration. Then again it rises and stabilizes around 3300 function evaluation per iteration, which is about 1000 evaluations more than in the case of the fixed step-size Euler forward method. The greater number of function

**Fig. 5.** Upper plot: Time courses of the “measured” $y_{\text{ref,lin,mm,3}}$, squares, and the estimated $y_{\text{HM,3}}$, crosses, (online sampling frequency of $10^{-3}$, integration tolerances $\text{AbsTol} = 10^{-6}$, RelTol $= 10^{-5}$). Right corner, a zoom of the area with the highest curvature. Lower plot: The respective local errors of $y_{\text{HM,3}}$, AbsTol and RelTol $\times y_3$ over time.

**Table 8**
The percentage of reduction in number of function evaluations when integrating Eq. 19 with the variable step size method in relation to its integration with a fixed-step size Euler forward method (online sampling frequency of $10^{-3}$, integration tolerances $\text{AbsTol} = 10^{-6}$, RelTol $= 10^{-5}$).

<table>
<thead>
<tr>
<th>Set</th>
<th>Training</th>
<th>Validation</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN</td>
<td>42.15%</td>
<td>46.22%</td>
<td>51.70%</td>
</tr>
<tr>
<td></td>
<td>44.64%</td>
<td>47.79%</td>
<td>48.33%</td>
</tr>
<tr>
<td></td>
<td>53.90%</td>
<td>44.41%</td>
<td></td>
</tr>
</tbody>
</table>
evaluations in case of the proposed method is owed to the sensitivity of the sensitivities equations with respect to changes in the step-size length and the strict integration tolerances (for greater tolerance limits the number of function evaluations generally decreases during the final stage). Nevertheless, it is remarkable that the number of function evaluations is still significantly lower as when applying a method to ensure similar numerical integration errors, such as e.g. a fixed step-size Heun method (4600 function evaluations per iteration). The profile of the number of function evaluations over the iterations is due to the parameter identification (gradient based), during which the parameters change the most in the initial stage of the identification followed by small parameter changes close to a minimum.

5. Conclusions

An adaptive step size method for the integration of measured variable dependent ODEs systems was proposed, which avoids the computational expensive repetitive interpolation of measurements. Using a first case study, the properties of the method are rigorously analyzed and the method is applied in combination with a hybrid semi-parametric model that in this particular case is identical to a dynamic neural network. In a second experimental case study the performance of the method was assessed for the development of a hybrid model with typical bioreactor experiment data. The following can be concluded:

i) For sufficiently small step sizes the Heun method produces greater integration errors than the RK4 method, even though the step size of the latter is two times the one of the Heun method, wherefore both methods evaluate the ODE system equally often.

ii) When increasing the step size, the differences between the integration errors of the Heun and the RK4 method become lower, ultimately resulting in a better performance of the Heun method.

iii) In case the proposed integration method is applied with the hybrid/DNN model it can be observed that errors that are due to the model mismatch are generally greater than those that are due to the numerical integration.

iv) The error due to the numerical integration can partially be accounted for by the incorporated Artificial Neural Network, which is however not desirable for reasons addressed above in Section 4.1.3.

v) In case of model simulation the proposed integration method can significantly reduce the number of function evaluations as in comparison to a fixed step-size method, by adapting the step size according to the curvature while adhering to the sampling frequency of the incorporated measurements. In the worst case, the proposed method will perform similar to an equivalent fixed-step size method, but the user is informed by the method if the desired error tolerances are exceeded. In the best case, a several fold speed up for the integration can be yielded (here about 10 times depending on the greatest number of $\Delta h$). The proposed method also avoids the repetitive interpolation of the online measurements, which would be required if traditional adaptive step-size methods would be used. This is of increasing importance the greater the number of repetitive interpolations and/or the greater the number of comprised online measured variables that need to interpolated.

vi) In case of parameter identification the number of function evaluations is higher than for a simple fixed-step Euler forward method, because (1) the sensitivities equations are integrated along with the model equations and so the curvature at least for a subset of equations is high and (2) because both of the combined methods, Heun and RK4, require twice as many evaluations for the same step as the forward method.

vii) Since the sampling frequency determines the smallest possible step size, the error tolerances can only be ensured to the degree where the smallest possible steps equal the sampling frequency.

viii) The sampling frequency is not required to be constant, but in general a constant frequency allows the application of the RK4 method, which obtains a significantly reduced integration error with the same number of function evaluations as the Heun method.

Acknowledgments

The author MVS acknowledges financial support through the Fundação para a Ciência e a Tecnologia (SFRH/BPD/84573/2012 and SFRH/BD/36990/2007). The authors thank Zita I.T.A Soons, Mathieu Streefland and the Netherlands Vaccine Institute for providing the data.

References


