

Research

Open Access

## Dynamic sensitivity analysis of biological systems

Wu Hsiung Wu<sup>1</sup>, Feng Sheng Wang<sup>\*2</sup> and Maw Shang Chang<sup>1</sup>

Address: <sup>1</sup>Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 62102, Taiwan and <sup>2</sup>Department of Chemical Engineering, National Chung Cheng University, Chiayi 62102, Taiwan

Email: Wu Hsiung Wu - [wwh@cs.ccu.edu.tw](mailto:wwh@cs.ccu.edu.tw); Feng Sheng Wang\* - [chmfsw@ccu.edu.tw](mailto:chmfsw@ccu.edu.tw); Maw Shang Chang - [mschang@cs.ccu.edu.tw](mailto:mschang@cs.ccu.edu.tw)

\* Corresponding author

from Asia Pacific Bioinformatics Network (APBioNet) Seventh International Conference on Bioinformatics (InCoB2008) Taipei, Taiwan. 20–23 October 2008

Published: 12 December 2008

*BMC Bioinformatics* 2008, **9**(Suppl 12):S17 doi:10.1186/1471-2105-9-S12-S17

This article is available from: <http://www.biomedcentral.com/1471-2105/9/S12/S17>

© 2008 Wu et al; licensee BioMed Central Ltd.

This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/2.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

### Abstract

**Background:** A mathematical model to understand, predict, control, or even design a real biological system is a central theme in systems biology. A dynamic biological system is always modeled as a nonlinear ordinary differential equation (ODE) system. How to simulate the dynamic behavior and dynamic parameter sensitivities of systems described by ODEs efficiently and accurately is a critical job. In many practical applications, e.g., the fed-batch fermentation systems, the system admissible input (corresponding to independent variables of the system) can be time-dependent. The main difficulty for investigating the dynamic log gains of these systems is the infinite dimension due to the time-dependent input. The classical dynamic sensitivity analysis does not take into account this case for the dynamic log gains.

**Results:** We present an algorithm with an adaptive step size control that can be used for computing the solution and dynamic sensitivities of an autonomous ODE system simultaneously. Although our algorithm is one of the decouple direct methods in computing dynamic sensitivities of an ODE system, the step size determined by model equations can be used on the computations of the time profile and dynamic sensitivities with moderate accuracy even when sensitivity equations are more stiff than model equations. To show this algorithm can perform the dynamic sensitivity analysis on very stiff ODE systems with moderate accuracy, it is implemented and applied to two sets of chemical reactions: pyrolysis of ethane and oxidation of formaldehyde. The accuracy of this algorithm is demonstrated by comparing the dynamic parameter sensitivities obtained from this new algorithm and from the direct method with Rosenbrock stiff integrator based on the indirect method. The same dynamic sensitivity analysis was performed on an ethanol fed-batch fermentation system with a time-varying feed rate to evaluate the applicability of the algorithm to realistic models with time-dependent admissible input.

**Conclusion:** By combining the accuracy we show with the efficiency of being a decouple direct method, our algorithm is an excellent method for computing dynamic parameter sensitivities in stiff problems. We extend the scope of classical dynamic sensitivity analysis to the investigation of dynamic log gains of models with time-dependent admissible input.

## Background

A mathematical model to understand, predict, control, or even design a real biological system is a central theme in systems biology. The most often used mathematical models for dynamic biological systems are formulated as nonlinear ordinary differential equations (ODEs). The critical challenges to get an ODE model are structure identification and parameter estimation of the model. To identify the structure and parameters of a dynamic model, the most important and essential job is to find the solution of ODEs efficiently and accurately. This job can be treated by analytical and numerical methods. Analytical methods are limited to certain special forms of ODEs, but numerical methods have no such limitations. There are several numerical methods can be used to solve ODEs [1-3], e.g., Taylor-series methods, modified Euler methods, and Runge-Kutta methods with variable step size control. The Taylor-series method takes more computation time to solve ODEs if the various derivatives are complicated, and the error is difficult to determine for arbitrary functions. The modified Euler method is a special case of a second-order Runge-Kutta method, and is more efficient compared to the Taylor-series method [4]. Fourth-order Runge-Kutta method is the most widely used ODE solver to meet requirements on both efficiency and accuracy. Collocation methods [5] are another common algorithms for solving ODEs and have been used for more than forty years. Wang [6] proposed a modified collocation method to transform ODEs into algebraic equations, and solved them by the Newton-Raphson method or an iteration method with step length restriction. The restricted step size is fixed and computed by trial and error. To overcome this drawback, we propose an adaptive step size control approach based on the Banach fixed point theorem for the modified collocation method in this paper.

There are different types of gains and dynamic sensitivities defined for sensitivity analysis [7-9]. The relative change of a dependent variable in response to a relative change in an independent variable is called a *logarithmic gain*, or a *log gain*. Log gains describe the change of dependent variables due to an environment change and are very useful for the assessment of the robustness and parameter estimation of a model. The change of a dependent variable in response to a change in a parameter is called a *parameter sensitivity*. In contrast to log gains, parameter sensitivities are the change of dependent variables correspond to a structure change in the model. The Biochemical Systems Theory (BST) [10] and Metabolic Control Analysis (MCA) [11-13] have achieved a great success in addressing the sensitivities at a steady state. However, the transient or periodic behavior is the primary interest in many systems (e.g., oscillation systems and fermentation systems that do not have a steady state). In these systems, the parameter sensitivities and log gains change with time, therefore the cal-

ulation methods for the steady state responses can not function. Dynamic sensitivity analysis is used in studying time-varying sensitivities in dynamic biological systems. A dynamic biological system can be characterized using logarithmic gains, sensitivities with respect to parameters and initial conditions. Several methods have been published to evaluate dynamic sensitivities [14-24]. They can be divided into the indirect methods (IDMs) and the direct methods (DMs). In the IDMs, the value of one dedicated parameter is varied infinitesimally while the values of other parameters are fixed. The model equations are solved anew for these sets of values of the parameters that differ in the value of the dedicated parameter only. The sensitivity of each variable with respect to this dedicated parameter is computed using the difference between the solutions of that variable for the two sets of values of the parameters divided by the infinitesimal difference of the dedicated parameter. In the DMs, using an ODE solver to solve the model equations and sensitivity equations simultaneously is the most used method for computing dynamic sensitivities. Shiraishi et al. [25] published a variable-order, variable-step Taylor-series method that can be used as an ODE solver providing a highly accurate calculation to compute dynamic sensitivities. This method is limited to general mass action (GMA) models described by power-law differential equations. Runge-Kutta methods with variable step size control can be used to compute dynamic sensitivities for most of the nonlinear differential equations, but is inefficient to determine the step size in a large dimensional system including the model differential equations and sensitivity differential equations. Due to the efficiency, Dunker [15] proposed the decoupled direct method (DDM), in which the sensitivity equations are solved separately from the model equations. He said: "the decoupled method has advantages in simplicity, stability, accuracy, efficiency, storage requirements, and program size over other methods". Although the DDM approach has so many advantages, the step size for the time profile determined by the error control of model equations is unable to be used for the sensitivities when the sensitivity equations are more stiff than the model equations and will generate inaccurate results.

Dynamic sensitivity analysis evaluates the influences on dependent variables due to variations of parameters, initial conditions and independent variables. In many practical applications, e.g., the fed-batch fermentation systems, the system admissible input (corresponding to independent variables of the system) can be time-dependent. The main difficulty for investigating the dynamic log gains of these systems is the infinite dimension due to the time-dependent input. Shiraishi et al. [26] proposed an efficient method, based on a combination of the recasting technique and the Taylor-series method, for calculating the time courses of log gains to investigate the dynamic

behavior of log gains for oscillation models with a limit cycle. The method is limited to the computations of dynamic log gains with respect to constant independent variables. We extend the computations of dynamic log gains to a model with continuous time-varying admissible input based on the finite parameterization method (PM). The classical PM was created for numerical solutions of optimal control problems [27]. The central idea of the method relies on a simple approximation mechanism. The whole time domain of a continuous admissible input is partitioned into several subintervals, and the input for each subinterval is approximated by a piecewise constant function. The dynamic log gains with respect to the continuous admissible input can be computed based on the partial derivations of dependent variables with respect to the piecewise constant input [28-30].

In this paper, we present an algorithm with an adaptive step size control that can be used for computing the solution and dynamic sensitivities of an autonomous ODE system simultaneously. This algorithm is the modified collocation method, proposed by Wang [6], with an adaptive step size control approach. Although our algorithm is one of the decouple direct methods in computing dynamic sensitivities of an ODE system, the step size determined by model equations can be used on the computations of the time profile and dynamic sensitivities with moderate accuracy even when sensitivity equations are more stiff than model equations. In the algorithm, the modified collocation method is used to transform model and sensitivity equations into algebraic equations, and the approximated solution is solved by an iteration method. This algorithm can be extended easily to solve problems of mixed differential and algebraic equations (DAEs) by combing algebraic equations with that transferred from differential equations. In our algorithm for computing dynamic sensitivities of an ODE system, the model equations and sensitivity equations are solved alternatively in two stages. First, the model equations are advanced from  $t_i$  to  $t_i + \eta$  using the iteration method, where  $\eta$  is the step size decided by model equations based on the fixed-point theorem. Second, the solution of model equations at  $t_i + \eta$  and the same step size are used to propagate the sensitivity equations from  $t_i$  to  $t_i + \eta$ . For dynamic systems with continuous time-dependent admissible input, the dynamic log gains are computed based on the parameterization techniques. The PM is used to approximate the original infinite-dimensional problem by a finite dimensional one with piecewise constant input. The dynamic log gain for this approximation problem is defined as the percentage change of a dependent variable in response to an infinitesimal percentage change for each piecewise constant input.

To show this algorithm can perform dynamic sensitivity analysis on very stiff ODE systems with moderate accuracy, it is implemented and applied to two sets of chemical reactions: pyrolysis of ethane and oxidation of formaldehyde. The accuracy of this algorithm is demonstrated by comparing the dynamic parameter sensitivities from this new method and that from the direct method with Rosenbrock stiff integrator based on the indirect method. The same dynamic sensitivity analysis is performed on an ethanol fed-batch fermentation system with a time-varying feed rate to evaluate the applicability of the algorithm to realistic models with time-dependent admissible input.

## Results and discussion

To illustrate the accuracy of our algorithm, it is implemented and applied to stiff chemical mechanisms for the pyrolysis of ethane as well as the oxidation of formaldehyde. These systems have been shown to be unstable using both the DM and the Green's function method [15]. The same dynamic sensitivity analysis is performed on an ethanol fed-batch fermentation system with a time-varying feed rate to evaluate the applicability of the algorithm to realistic models with time-dependent admissible input.

### Pyrolysis of ethane

The chemical mechanism for the pyrolysis of ethane is a very stiff system and consists of seven species in five reactions. The chemical reactions and rate constants are shown in Table 1 and are described by GMA model equations as follows:

$$\begin{aligned}\frac{d[CH_3]}{dt} &= 2k_1[C_2H_6] - k_2[CH_3][C_2H_6] \\ \frac{d[CH_4]}{dt} &= k_2[CH_3][C_2H_6] \\ \frac{d[C_2H_4]}{dt} &= k_3[C_2H_5] \\ \frac{d[C_2H_5]}{dt} &= k_2[CH_3][C_2H_6] + k_4[C_2H_6][H] - k_3[C_2H_5] \\ \frac{d[C_2H_6]}{dt} &= -k_1[C_2H_6] - k_2[CH_3][C_2H_6] - k_4[C_2H_6][H] \\ \frac{d[H]}{dt} &= k_3[C_2H_5] - k_4[C_2H_6][H] - 2k_5[H]^2 \\ \frac{d[H_2]}{dt} &= k_4[C_2H_6][H] + k_5[H]^2\end{aligned}$$

where  $[x]$  is the concentration of species  $x$  and  $k_i$  is the rate constant. The initial concentration of  $C_2H_6$  is  $5.951 \times 10^{-6}$  mol/cm<sup>3</sup> and all other initial concentrations are zeros. All sets of sensitivity coefficients with respect to all rate constants and initial conditions are computed simultaneously without any difficulty using our algorithm with a tolerance of  $10^{-7}$ . The normalized sensitivity coefficients for the pyrolysis of ethane at 1 s and 20 s calculated by our

algorithm are shown in Table 2. The results obtained by the indirect method (IDM) according to the finite difference approximation and the direct method with Rosenbrock stiff integrator (R/DM) [24] are also given in Table 2 for comparison. The results of our algorithm are of equal accuracy to R/DM in comparison to the IDM and the maximum relative error is 0.58%.

**Oxidation of formaldehyde**

The formaldehyde oxidation mechanism is a larger system, involves 15 species in 25 reactions. The chemical reactions and rate constants are shown in Table 3 and are described by GMA model equations as follows:

$$\begin{aligned} \frac{d[HCO]}{dt} &= k_2[HO_2][CH_2O] + k_4[CH_2O][OH] + k_{11}[CH_2O][H] + k_{16}[CH_2O][O] \\ &\quad + k_{22}[O_2][CH_2O] - k_1[HCO][O_2] - k_{20}[HCO] \\ \frac{d[O_2]}{dt} &= k_8[HO_2]^2 + k_{14}[HO_2][M] + k_{25}[HO_2][H] \\ &\quad - k_1[HCO][O_2] - k_{12}[O_2][H] - k_{13}[O_2][M][H] - k_{22}[O_2][CH_2O] \\ \frac{d[HO_2]}{dt} &= k_1[HCO][O_2] + k_5[H_2O_2][OH] + k_{13}[O_2][M][H] + k_{17}[H_2O_2][H] \\ &\quad + k_{19}[H_2O_2][O] + k_{22}[O_2][CH_2O] - k_2[HO_2][CH_2O] - k_7[HO_2] \\ &\quad - 2k_8[HO_2]^2 - k_{10}[HO_2][CO] - k_{14}[HO_2][M] - (k_{23} + k_{24} + k_{25})[HO_2][H] \\ \frac{d[CO]}{dt} &= k_1[HCO][O_2] + k_{20}[HCO] - k_9[CO][OH] - k_{10}[HO_2][CO] \\ \frac{d[CH_2O]}{dt} &= -k_2[HO_2][CH_2O] - k_4[CH_2O][OH] - k_{11}[CH_2O][H] - k_{16}[CH_2O][O] \\ &\quad - k_{22}[O_2][CH_2O] \\ \frac{d[H_2O_2]}{dt} &= k_2[HO_2][CH_2O] + k_8[HO_2]^2 - k_3[H_2O_2][M] - k_5[H_2O_2][OH] \\ &\quad - k_6[H_2O_2] - (k_{17} + k_{18})[H_2O_2][H] - k_{19}[H_2O_2][O] \\ \frac{d[M]}{dt} &= 0 \\ \frac{d[OH]}{dt} &= 2k_3[H_2O_2][M] + k_{10}[HO_2][CO] + k_{12}[O_2][H] + k_{15}[H_2][O] \\ &\quad + k_{16}[CH_2O][O] + k_{18}[H_2O_2][H] + k_{19}[H_2O_2][O] + 2k_{23}[HO_2][H] \\ &\quad - k_4[CH_2O][OH] - k_5[H_2O_2][OH] - k_9[CO][OH] - k_{21}[OH][H_2] \\ \frac{d[H_2O]}{dt} &= k_4[CH_2O][OH] + k_5[H_2O_2][OH] + k_{18}[H_2O_2][H] + k_{21}[OH][H_2] \\ &\quad + k_{24}[HO_2][H] \\ \frac{d[H_2O_2(wall)]}{dt} &= k_6[H_2O_2] \\ \frac{d[HO_2(wall)]}{dt} &= k_7[HO_2] \\ \frac{d[CO_2]}{dt} &= k_9[CO][OH] + k_{10}[HO_2][CO] \\ \frac{d[H]}{dt} &= k_9[CO][OH] + k_{14}[HO_2][M] + k_{15}[H_2][O] + k_{20}[HCO] \\ &\quad + k_{21}[OH][H_2] - k_{11}[CH_2O][H] - k_{12}[O_2][H] - k_{13}[O_2][M][H] \\ &\quad - (k_{17} + k_{18})[H_2O_2][H] - (k_{23} + k_{24} + k_{25})[HO_2][H] \\ \frac{d[H_2]}{dt} &= k_{11}[CH_2O][H] + k_{17}[H_2O_2][H] + k_{25}[HO_2][H] - k_{15}[H_2][O] \\ &\quad - k_{21}[OH][H_2] \\ \frac{d[O]}{dt} &= k_{12}[O_2][H] + k_{24}[HO_2][H] - k_{13}[H_2][O] - k_{16}[CH_2O][O] \\ &\quad - k_{19}[H_2O_2][O] \end{aligned}$$

where [x] is the concentration of species x and  $k_i$  is the rate constant. The initial concentrations in mol/cm<sup>3</sup> are [CH<sub>2</sub>O] = 1.124 × 10<sup>-7</sup>, [O<sub>2</sub>] = 2.109 × 10<sup>-6</sup>, [CO] = 4.699 × 10<sup>-6</sup>, [M] = 1.1772 × 10<sup>-5</sup>, and all other initial concentrations are zeros. Sensitivity coefficients with respect to all rate constants and initial conditions are computed successfully using our algorithm with a tolerance of 10<sup>-9</sup>. The normalized sensitivity coefficients for O and H<sub>2</sub>O at 0.005 s calculated by our algorithm are presented in Table 4. The

**Table 1: The mechanism for ethane pyrolysis.**

Reaction	Rate constants
$C_2H_6 \rightarrow CH_3 + CH_3$	$1.14 \times 10^{-2}$
$CH_3 + C_2H_6 \rightarrow CH_4 + C_2H_5$	$1.19 \times 10^6$
$C_2H_5 \rightarrow C_2H_4 + H$	$1.57 \times 10^3$
$H + C_2H_6 \rightarrow H_2 + C_2H_5$	$9.72 \times 10^8$
$H + H \rightarrow H_2$	$6.99 \times 10^{13}$

The initial concentration of C<sub>2</sub>H<sub>6</sub> is 5.951 × 10<sup>-6</sup> mol/cm<sup>3</sup> and all other initial concentrations are zeros. Units for rate constants are mol, cm, s and the temperature is 923 K.

results obtained by IDM and the direct method with Rosenbrock stiff integrator (R/DM) are also given in Table 4 for comparison. Our results are in good agreement with the R/DM in comparison to the IDM, and the maximum relative error is 0.25%. The discrepancies between the results of our algorithm and the R/DM method are sufficiently small to prove that this new method is capable of performing dynamic sensitivity analysis for stiff differential equations as accurate as direct methods.

**Ethanol fed-batch fermentation**

The dynamic sensitivity analysis of an ethanol fed-batch fermentation process, a real dynamic biological system never reaching a steady state, is used to elucidate the applicability of our algorithm. Wang et al. [31] built a mathematical kinetic model of fermentation for ethanol and glycerol production using *Saccharomyces diastaticus* LORRE 316, which is a high ethanol tolerance yeast. The mathematical kinetic model for the fed-batch process consists of the dynamic behavior of biomass, glucose, ethanol and glycerol, and its dynamic mass balance equations are expressed as follows:

$$\begin{aligned} \frac{dx}{d\tau} &= t_f \left( \mu x - \frac{F}{V} x \right), \\ \frac{ds}{d\tau} &= t_f \left[ - \left( \frac{q_{p1}}{Y_{p1/s}} + \frac{q_{p2}}{Y_{p2/s}} \right) x + \frac{F}{V} (s_F - s) \right], \\ \frac{dp_1}{d\tau} &= t_f \left( q_{p1} x - \frac{F}{V} p_1 \right), \\ \frac{dp_2}{d\tau} &= t_f \left( q_{p2} x - \frac{F}{V} p_2 \right), \\ \frac{dV}{d\tau} &= t_f F, \end{aligned}$$

where x is the concentration of cell mass, s is the concentration of glucose, p<sub>1</sub> is the concentration of ethanol, p<sub>2</sub> is the concentration of glycerol, V is the working volume of the fermenter, t<sub>f</sub> is the final fermentation time, τ = t/t<sub>f</sub> is the

**Table 2: Sensitivity coefficients for ethane pyrolysis.**

Species $x_i$	$\partial \ln [x_i]/\partial k_1$ at 1 s			$\partial \ln [x_i]/\partial k_1$ at 20 s		
	IDM	R/DM	our algorithm	IDM	R/DM	our algorithm
$CH_3$	1.000	1.000	0.99986	1.000	1.000	1.00000
$CH_4$	0.976	0.976	0.97625	0.644	0.644	0.64350
$C_2H_4$	0.680	0.680	0.68039	0.324	0.323	0.32348
$C_2H_5$	0.662	0.662	0.66149	-0.209	-0.210	-0.20950
$C_2H_6$	-0.044	-0.044	-0.04425	-0.819	-0.819	-0.81896
$H$	0.478	0.478	0.47783	0.091	0.091	0.09053
$H_2$	0.602	0.602	0.60214	0.221	0.221	0.22098

A tolerance of  $10^{-8}$  is used for R/DM. A fourth-order adaptive Rosenbrock algorithm is used for integration and  $\pm 5\%$  variations of  $k_1$  are used in the computation of the sensitivities w.r.t.  $k_1$  using indirect method.

**Table 3: The mechanism for formaldehyde oxidation.**

Reaction	Rate constants
$HCO + O_2 \rightarrow HO_2 + CO$	$6.02 \times 10^{10}$
$HO_2 + CH_2O \rightarrow H_2O_2 + HCO$	$3.43 \times 10^{10}$
$H_2O_2 + M \rightarrow 2OH + M$	$4.01 \times 10^6$
$OH + CH_2O \rightarrow H_2O + HCO$	$9.64 \times 10^{13}$
$OH + H_2O_2 \rightarrow H_2O + HO_2$	$3.07 \times 10^{12}$
$H_2O_2 \rightarrow H_2O_2(\text{wall})$	$1.05 \times 10^2$
$HO_2 \rightarrow HO_2(\text{wall})$	$1.05 \times 10^1$
$HO_2 + HO_2 \rightarrow H_2O_2 + O_2$	$1.81 \times 10^{12}$
$OH + CO \rightarrow CO_2 + H$	$1.99 \times 10^{11}$
$HO_2 + CO \rightarrow CO_2 + OH$	$7.23 \times 10^8$
$H + CH_2O \rightarrow H_2 + HCO$	$1.63 \times 10^{12}$
$H + O_2 \rightarrow OH + O$	$3.32 \times 10^{10}$
$H + O_2 + M \rightarrow HO_2 + M$	$3.63 \times 10^{15}$
$HO_2 + M \rightarrow H + O_2 + M$	$2.83 \times 10^5$
$O + H_2 \rightarrow OH + H$	$1.82 \times 10^{11}$
$O + CH_2O \rightarrow OH + HCO$	$6.02 \times 10^{13}$
$H + H_2O_2 \rightarrow HO_2 + H_2$	$7.83 \times 10^{11}$
$H + H_2O_2 \rightarrow H_2O + OH$	$3.55 \times 10^{12}$
$O + H_2O_2 \rightarrow OH + HO_2$	$6.02 \times 10^{10}$
$HCO \rightarrow H + CO$	$4.60 \times 10^{-12}$
$OH + H_2 \rightarrow H_2O + H$	$6.02 \times 10^{12}$
$CH_2O + O_2 \rightarrow HCO + HO_2$	$1.75 \times 10^4$
$H + HO_2 \rightarrow 2OH$	$3.01 \times 10^{12}$
$H + HO_2 \rightarrow H_2O + O$	$3.01 \times 10^{13}$
$H + HO_2 \rightarrow H_2 + O_2$	$2.71 \times 10^{13}$

The initial concentrations in mol/cm<sup>3</sup> are  $[CH_2O] = 1.124 \times 10^{-7}$ ,  $[O_2] = 2.109 \times 10^{-6}$ ,  $[CO] = 4.699 \times 10^{-6}$ ,  $[M] = 1.1772 \times 10^{-5}$  and all other initial concentrations are zeros. Units for rate constants are mol, cm, s and the temperature is 952 K.

**Table 4: Sensitivity coefficients for formaldehyde oxidation.**

Rate constant	$\partial \ln [HO_2]/\partial k_i$			$\partial \ln [O]/\partial k_i$		
	IDM	R/DM	our algorithm	IDM	R/DM	our algorithm
$k_2$	0.683	0.683	0.68255	0.827	0.827	0.82719
$k_3$	0.700	0.700	0.69986	0.835	0.835	0.83486
$k_4$	-0.210	-0.209	-0.20917	-1.160	-1.156	-1.15579
$k_8$	-0.306	-0.306	-0.30569	-0.296	-0.296	-0.29599
$k_9$	0.210	0.210	0.20962	1.156	1.156	1.15628
$k_{10}$	0.164	0.164	0.16373	1.031	1.031	1.03065
$k_{11}$	-0.121	-0.121	-0.12087	-0.660	-0.659	-0.65906
$k_{12}$	0.188	0.188	0.18848	0.979	0.979	0.97926
$k_{13}$				-0.327	-0.327	-0.32713
$k_{16}$				-1.002	-1.000	-0.99990
$k_{22}$	0.686	0.685	0.68536	0.742	0.742	0.74169

A tolerance of  $10^{-15}$  is used for R/DM. A fourth-order adaptive Rosenbrock algorithm is used for integration and  $\pm 5\%$  variations of  $k_i$  are used in the computation of the sensitivities w.r.t.  $k_i$  using indirect method.

normalized fermentation time,  $s_F$  is the feed concentration of glucose,  $F$  is the feed rate,  $Y_{p_1/s}$  is the ethanol yield factor, and  $Y_{p_2/s}$  is the glycerol yield factor. The unstructured kinetic models for the specific cell growth and product formation are respectively expressed as follows:

$$\mu = \frac{\mu_m s}{K_s + s + s^2 / K_{sI}} \frac{K_{p1}}{K_{p1} + p_1 + p_1^2 / K_{p1I}} \frac{K_{p2}}{K_{p2} + p_2 + p_2^2 / K_{p2I}},$$

$$q_{p_1} = \frac{v_{p1} s}{K'_s + s + s^2 / K'_{sI}} \frac{K'_{p1}}{K'_{p1} + p_1 + p_1^2 / K'_{p1I}},$$

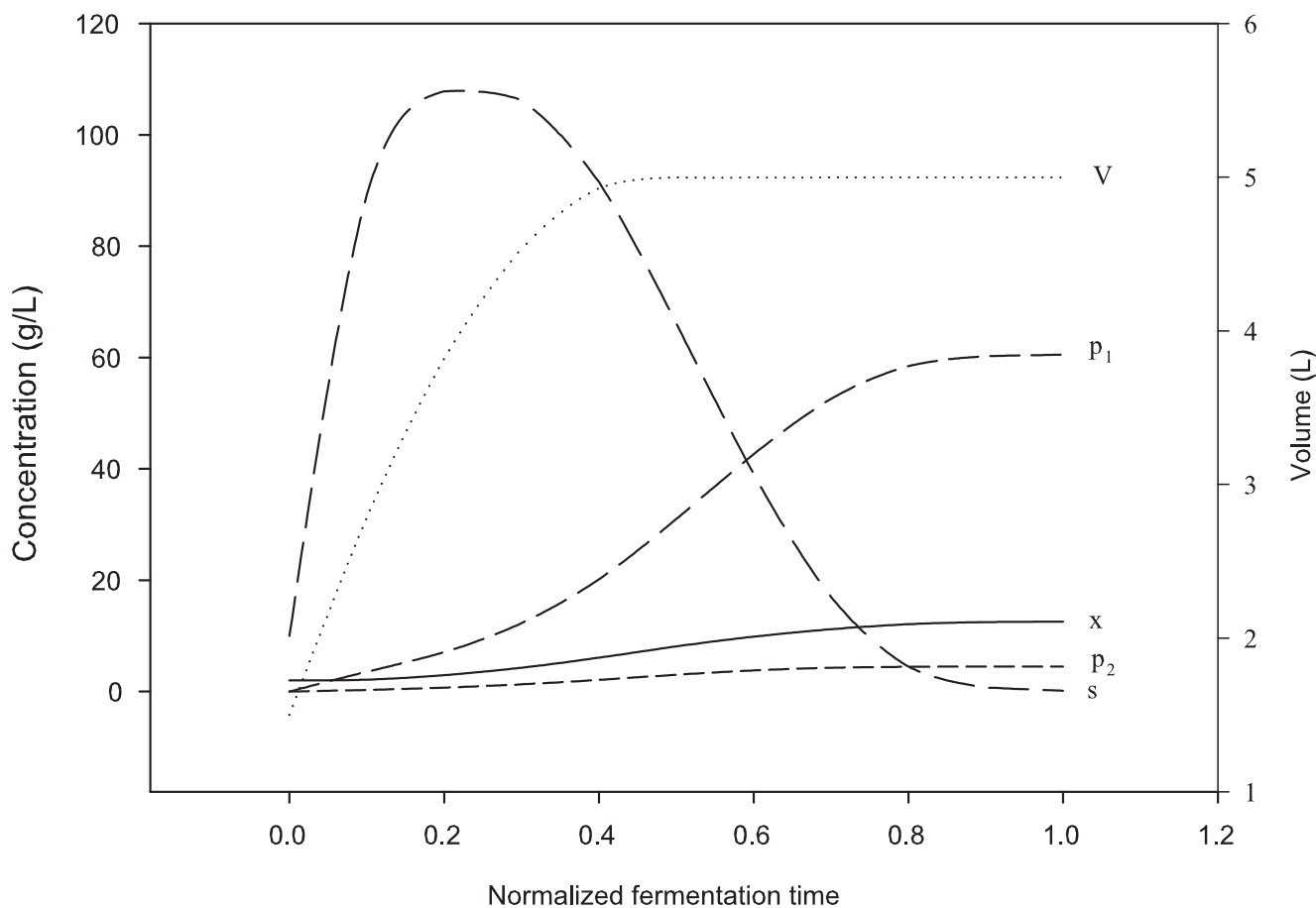
$$q_{p_2} = \frac{v_{p2} s}{K''_s + s + s^2 / K''_{sI}} \frac{K'_{p2}}{K'_{p2} + p_2 + p_2^2 / K'_{p2I}}.$$

Using a batch fermentation model, Wang et al. obtained the optimal values of 19 parameters [31]. The initial and feed concentrations of glucose are set to 10 and 200 g/L, the initial concentration of biomass is set to 2 g/L, and the starting working volume is set to 1.5 L in the computations of optimal feed rate and optimal fermentation time to maximize the ethanol production rate  $J = p_1 V / t_f$  under some physical constraints, e.g., the residual glucose restriction  $s(t_f) \leq s_r$  for reducing the separation cost,  $s_r$  is the concentration of the desired residual glucose. The optimal final fermentation time is 12.836 hours and the optimal feed rate  $F^*$  for the fed-batch fermentation model [31] is as follows:

$$F^* = \begin{cases} 1.0 & \text{for } 0 \leq \tau \leq 0.1 \\ 0.8067 & \text{for } 0.1 < \tau \leq 0.2 \\ 0.5567 & \text{for } 0.2 < \tau \leq 0.3 \\ 0.3067 & \text{for } 0.3 < \tau \leq 0.4 \\ 0.0557 & \text{for } 0.4 < \tau \leq 0.5 \\ 1.3687 \times 10^{-18} & \text{for } 0.5 < \tau \leq 0.6 \\ 6.2749 \times 10^{-18} & \text{for } 0.6 < \tau \leq 0.7 \\ 0.0 & \text{for } 0.7 < \tau \leq 0.8 \\ 0.0 & \text{for } 0.8 < \tau \leq 0.9 \\ 2.1883 \times 10^{-20} & \text{for } 0.9 < \tau \leq 1 \end{cases} \quad (1)$$

Figure 1 shows the computational time profile of the ethanol fed-batch fermentation model with the optimal feed rate.

Our algorithm is applied to the ethanol fed-batch fermentation model using the initial conditions as described above. All dynamic sensitivities with respect to 22 parameters (including  $s_F$ ,  $F$  and  $t_f$ ) and initial conditions, and the dynamic log gains with respect to time-varying feed rate are computed simultaneously without any difficulty. Figure 2 shows the dynamic relative sensitivities with respect to  $\mu_m$ ,  $Y_{p_1/s}$ ,  $Y_{p_2/s}$ , and  $s_F$ . When the maximum specific growth rate  $\mu_m$  is increasing, the rate of consuming glucose is increasing such that the concentration of residue glucose is decreasing. This situation is compatible with the trend in Figure 2(a). The increases in the ethanol and glycerol yield factor cause the increases in the production of ethanol and glycerol, and more glucose remains at the final time. As Figures 2(b) and 2(c) show, to increase the



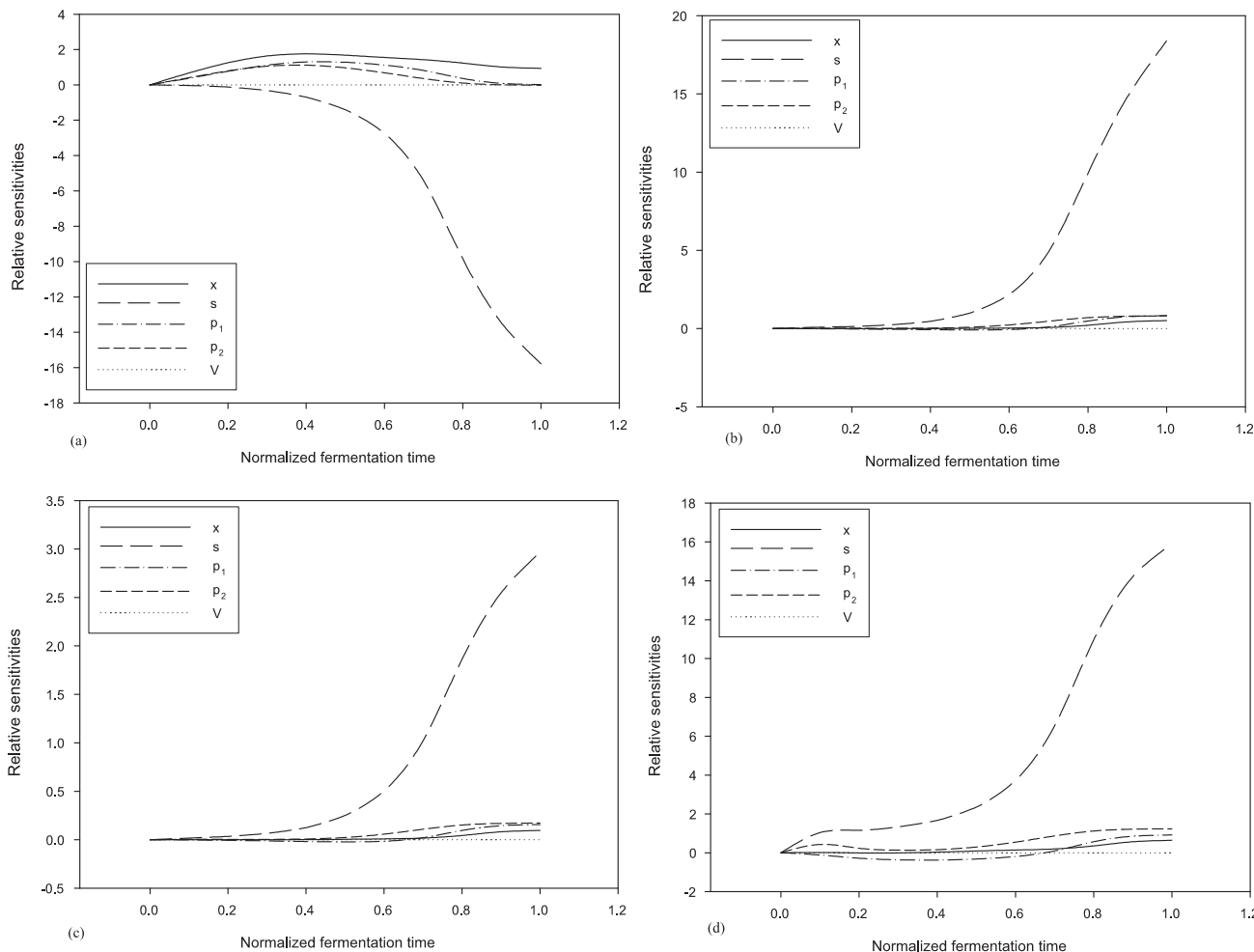
**Figure 1**  
**Dynamic behavior of the ethanol fed-batch fermentation model.** Time profiles of cell mass ( $x$ ), glucose ( $s$ ), ethanol ( $p_1$ ), glycerol ( $p_2$ ), and the working volume ( $V$ ) for the ethanol fed-batch fermentation computed using the optimal feed rate  $F^*$  and the feed glucose of 200 g/L. The horizontal scale is in normalized fermentation time ( $t/t_f$ ).

production of ethanol and glycerol by improving the ethanol yield factor is better than by increasing the glycerol yield factor. Figure 2(d) shows that if the feed concentration of glucose is increasing, the cell growth and the production of ethanol and glycerol are increasing. Under this condition, *S. diastaticus* LORRE 316 is unable to completely consume glucose to produce ethanol during the fermentation time and more glucose remains at the final time.

The relative sensitivity with respect to  $t_f$  is shown in Figure 3(a). As expected, an increase in  $t_f$  causes a low relative increase in the concentration of cell mass and a high relative decrease in the concentration of residue glucose. Figures 3(b), 3(c) and 3(d) show the dynamic relative sensitivities with respect to the initial conditions  $x$ ,  $s$ , and

$V$ . When the initial concentration of cell mass increases, the residue glucose decreases, and the production of ethanol will increase a little, but the production of glycerol will decrease a little at the final fermentation time. Starting the fermentation process with more glucose will cause more glucose to remain and the production of ethanol and glycerol to increase a little at the final fermentation time as shown in Figure 3(c). Figure 3(d) shows that if the initial working volume is increasing, all concentrations of cell mass, glucose, ethanol, and glycerol are decreasing at the final fermentation time.

We are interested in the ethanol production rate  $J$  in the fermentation process. The effects on  $J$  with respect to  $\mu_m$ ,  $Y_{p_1/s}$ ,  $Y_{p_2/s}$ ,  $S_{Fv}$  and  $t_f$  are shown in Figure 4. To increase



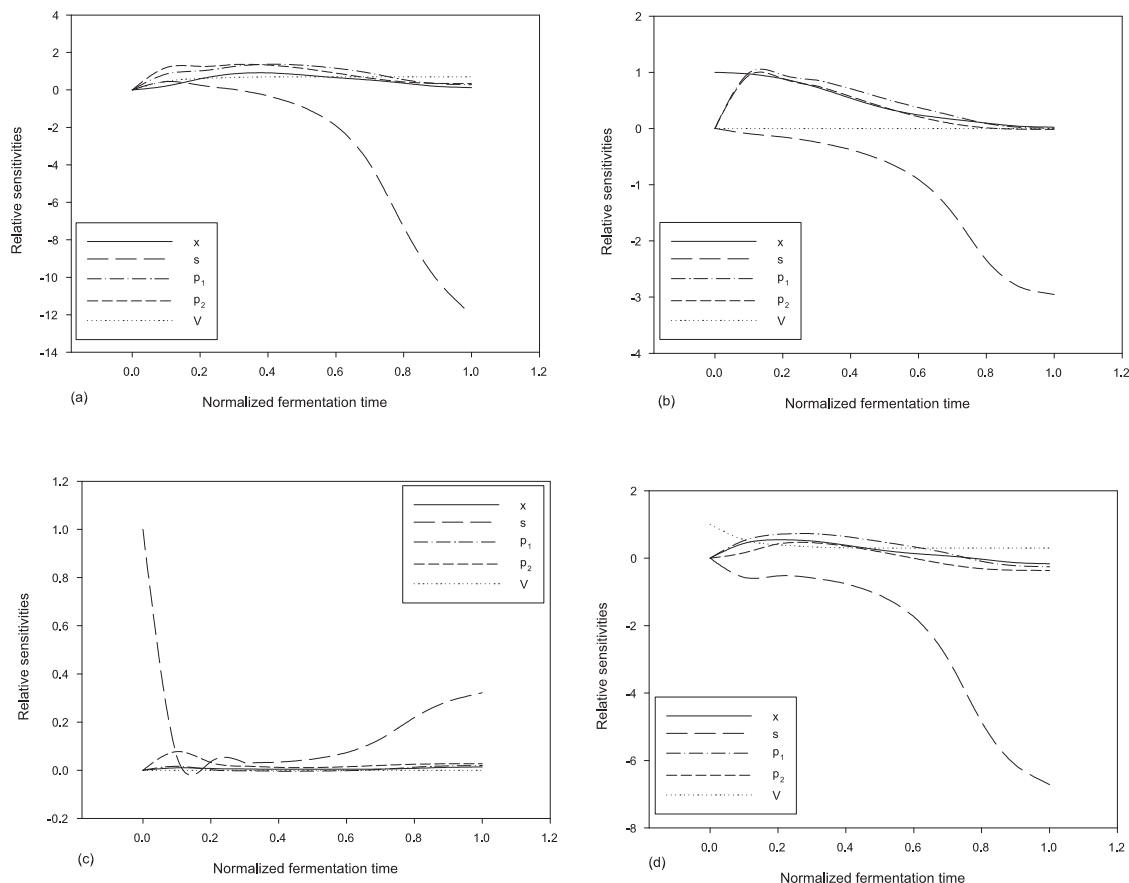
**Figure 2**  
**Dynamic relative sensitivities with respect to  $\mu_m$ ,  $Y_{p_1/s}$ ,  $Y_{p_2/s}$ , and  $s_F$ .** (a) relative sensitivities with respect to  $\mu_m$ ; (b) relative sensitivities with respect to  $Y_{p_1/s}$ ; (c) relative sensitivities with respect to  $Y_{p_2/s}$ ; (d) relative sensitivities with respect to  $s_F$ . The horizontal scale is in normalized fermentation time ( $t/t_f$ ).

$J$ , it is clear that an increase in  $Y_{p_1/s}$  or  $s_F$  will have more impact than an equal relative increase in  $Y_{p_2/s}$  or  $\mu_m$ . The negative value of relative sensitivity for  $J$  with respect to  $t_f$  means a decrease in the fermentation time will get a higher  $J$  at the expense of more residual glucose. Though the relative sensitivity of  $J$  with respect to  $s_F$  is higher than that with respect to  $Y_{p_1/s}$  at the final fermentation time, by increasing  $s_F$  to increase  $J$  will cause more glucose left at the final time and increase the cost to separate the residue glucose and the ethanol product. We can make a conclu-

sion that to increase  $J$  by increasing  $Y_{p_1/s}$  will be a better choice than increasing  $s_F$  or  $Y_{p_2/s}$ , and decreasing the fermentation time.

The feed rate  $F(t)$  of the fed-batch fermentation model is a time-dependent input control variable, so that the computation of the effect on  $J$  with respect to  $F(t)$  is an infinite dimensional problem. The fermentation time is divided into ten equal time partitions, and the optimal feed rate  $F^*$  for the fed-batch fermentation model [31] is approximated by ten piecewise constant functions. The ten input control parameters, denoted by  $F_i$ ,  $i = 1, \dots, 10$ , are shown





**Figure 3**  
**Dynamic relative sensitivities with respect to  $t_f$ ,  $x(0)$ ,  $s(0)$ , and  $V(0)$ .** (a) relative sensitivities with respect to  $t_f$ ; (b) relative sensitivities with respect to the initial value of  $x$ ; (c) relative sensitivities with respect to the initial value of  $s$ ; (d) relative sensitivities with respect to the initial value of  $V$ . The horizontal scale is in normalized fermentation time ( $t/t_f$ ).

in equation (1). The proposed algorithm computes the dynamic log gains based on the parameterization method. All dynamic log gains with respect to  $F_i$ ,  $i = 1, \dots, 10$ , are computed (data not shown here) with the parameter sensitivities simultaneously. The dynamic log gains of  $J$  with respect to  $F_i$ ,  $i = 1, \dots, 10$ , are computed by

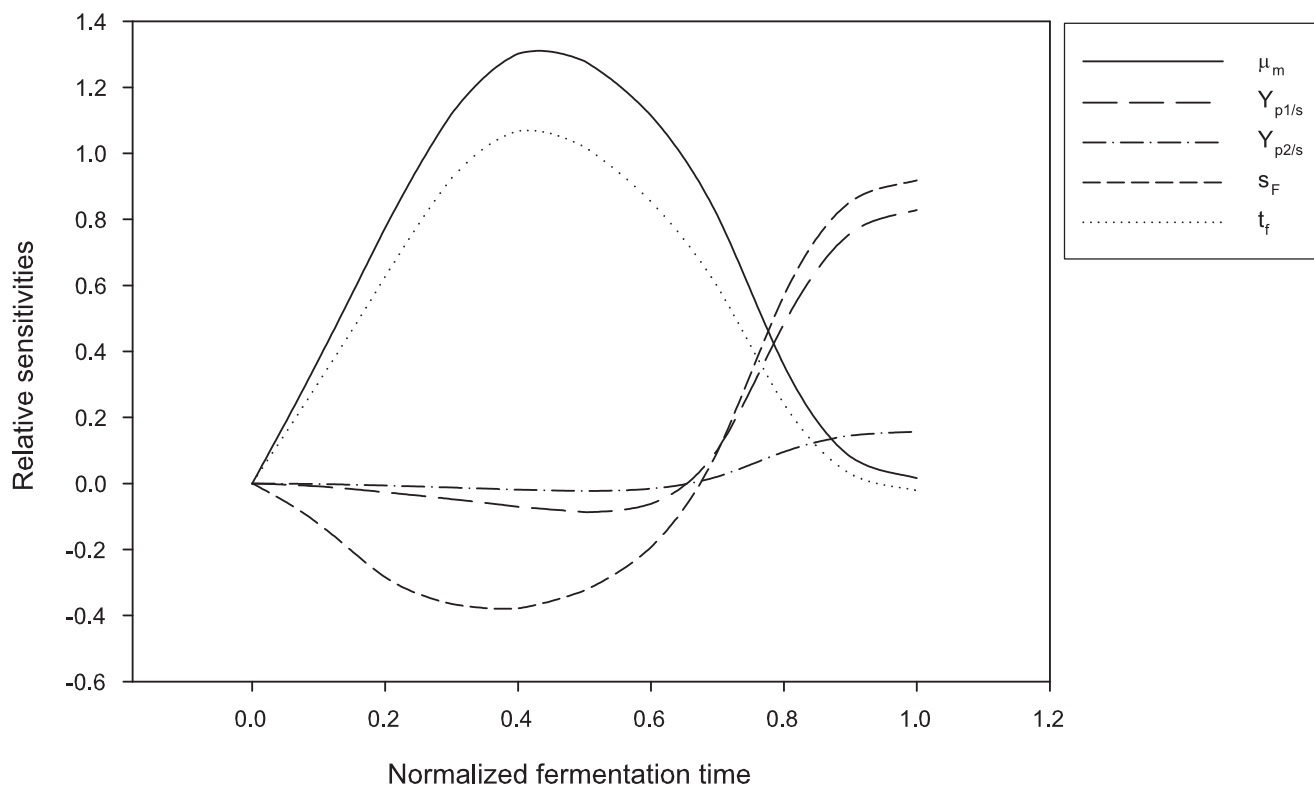
$$\frac{\partial \ln J}{\partial \ln F_i} = \frac{\partial \ln p_1}{\partial \ln F_i} + \frac{\partial \ln V}{\partial \ln F_i}.$$

Due to the optimal values of  $F_i$ ,  $i = 6, \dots, 10$  are equal to or very close to 0, The dynamic log gains of  $J$  with respect to them are small and can be ignored. The dynamic log gains of  $J$  with respect to  $F_i$ ,  $i = 1, \dots, 5$ , are shown in Figure 5. The effects on  $J$  are in decreasing order from  $F_1$  to  $F_5$ . Increasing the feed rate at an early stage will get a higher  $J$  at the

final fermentation time than that at a later stage without considering the residual glucose.

**Conclusion**

To deeply study the dynamic behavior of a biological system, one of the methods is to model it as a mathematical model. The most used mathematical model for simulating biological systems is the ODE model. The essential task for modeling and simulating a biological system is to find the solution of an ODE model efficiently and accurately. We present an algorithm with an adaptive step size control that can be used for computing the solution and dynamic sensitivities of an ODE system simultaneously. Instead of using error control to decide the step size in solving the model equations, our algorithm computes the step size based on the fixed-point theorem and the same step size can be used in solving the sensitivity equations.



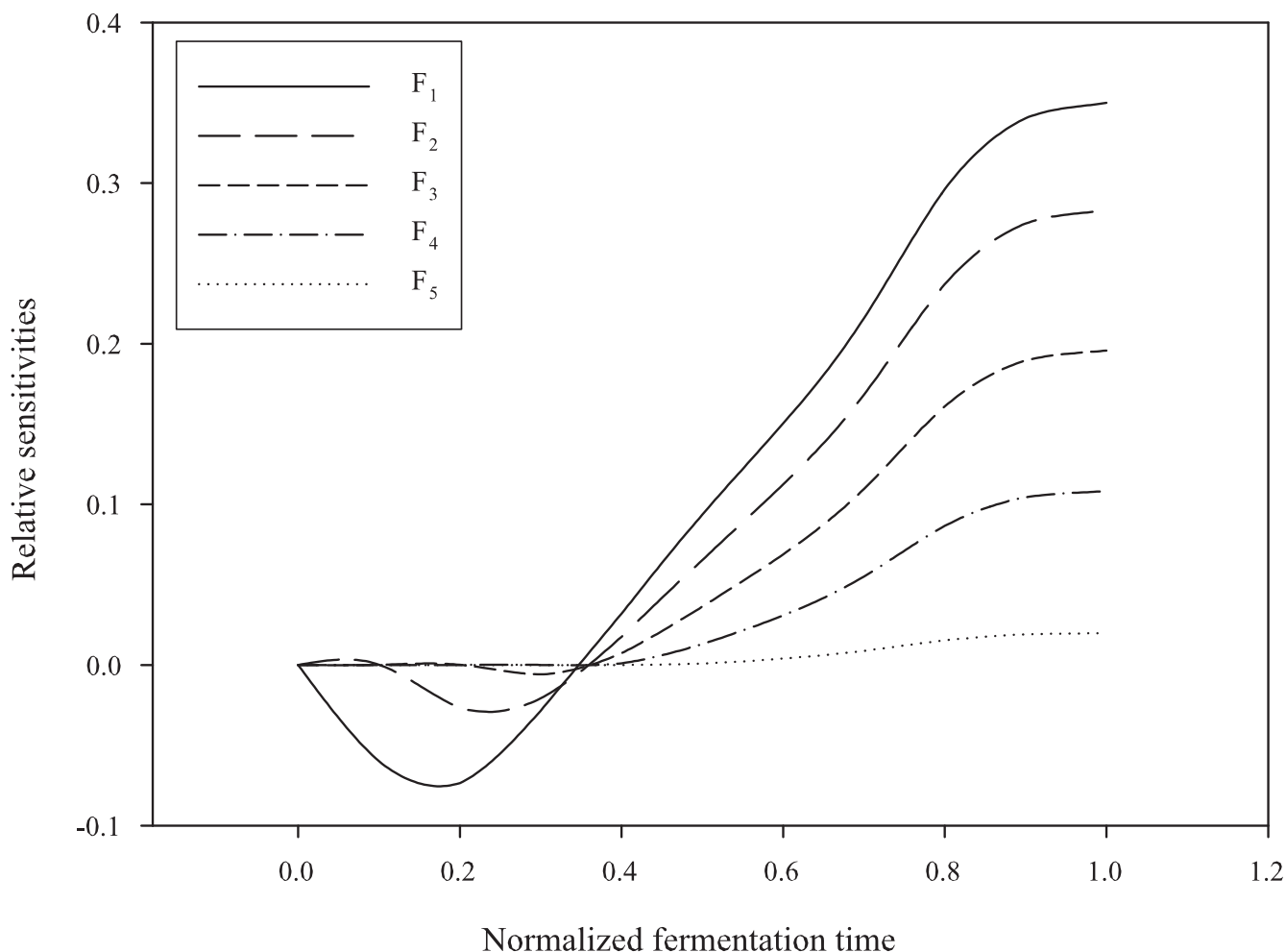
**Figure 4**

**Dynamic relative sensitivities of  $J$ .** The relative sensitivities of ethanol production rate with respect to  $\mu_m$ ,  $Y_{p1/s}$ ,  $Y_{p2/s}$ ,  $s_F$ , and  $t_f$ . The horizontal scale is in normalized fermentation time ( $t/t_f$ ).

Dynamic sensitivity analysis is a useful tool to investigate the behavior of dynamic systems. In the direct methods for solving the dynamic sensitivities, sensitivity equations and model equations are coupled and solved together at the expense of more computation time. In contrast, sensitivity equations and model equations are solved separately in the decouple direct methods. The DDMs are more efficient than the DMs due to the dimension of ODEs. The chief disadvantage of DDMs is the requirement of error control on both model equations and sensitivity equations. Our algorithm with an efficient step control approach based on the fixed-point theorem is used to address the disadvantage of DDMs. Analogous to the DMs, the same step size obtained by model equations is used on both model and sensitivity equations. It has been implemented and applied to well-known stiff problems with the same accuracy compared to the direct method with Rosenbrock stiff integrator (R/DM). As our algorithm is one of the DDMs, it has the efficiency of the DDMs and the same accuracy of the DMs as presented in

the section describing the results. By combining the efficiency and accuracy, our algorithm is an excellent method for computing dynamic parameter sensitivities in stiff problems.

We extend the scope of classical dynamic sensitivity analysis to the investigation of dynamic log gains of models with time-dependent admissible input. The parameterization method is used to approximate the infinite-dimensional computation problem for dynamic log gains in models with time-dependent admissible input by a classical finite-dimensional computation problem of dynamic log gains. Then, all dynamic log gains and parameter sensitivities can be obtained simultaneously from our algorithm. Appropriate parameterization allows one to obtain a more efficient way to compute the dynamic log gains with respect to a continuous time-dependent input than that by finite difference approximation. Finally, the new proposed algorithm is applied to the ethanol fed-batch fermentation system, a real dynamic biological system



**Figure 5**  
**Dynamic log gains of  $J$  w.r.t.  $F(t)$ .** The dynamic log gains of ethanol production rate with respect to  $F_i, i = 1, \dots, 5$ . The horizontal scale is in normalized fermentation time ( $t/t_f$ ).

which never reaches a steady state, with a time-varying feed rate for elucidating the applicability to realistic models with time-dependent admissible input. Through the dynamic sensitivity analysis of the ethanol fed-batch fermentation model, we conclude that to get a higher ethanol production rate by increasing the ethanol yield factor is a good choice.

**Methods**

A dynamic biological system is always modeled as a non-linear ODE system:

$$\frac{dx}{dt} = Nv(x(t), y(t); \theta), x(0) = x_0, y(0) = y_0, \quad (2)$$

where  $x(t) \in \mathbb{R}^n$  is a vector of dependent variables,  $y(t) \in \mathbb{R}^m$  is a vector of independent variables,  $\theta \in \mathbb{R}^p$  is a vector of parameters,  $v \in \mathbb{R}^q$  is a vector of fluxes between the var-

iables,  $N \in \mathbb{R}^{(n+m) \times q}$  is the stoichiometric matrix describing the interconnecting fluxes,  $x_0$  and  $y_0$  are initial concentrations of  $x$  and  $y$  respectively. Using a different kinetic description for  $v$  results in a different mathematical model. In the Michaelis-Menten model, each element  $v_i$  of  $v$  is of the form

$$v_i = \frac{V_i^{max} x_j}{K_i + x_j}, i = 1, \dots, q, \quad (3)$$

where  $x_j$  is the substrate,  $V_i^{max}$  is the maximum flux for  $v_i$ , and  $K_i$  is the half-saturated flux for  $v_i$ . For the GMA systems, the kinetic equation for  $v_i$  is expressed as a power-law function

$$v_i = \gamma_i \prod_{j=1}^{n+m} x_j^{g_{ij}}, i = 1, \dots, q. \tag{4}$$

where  $x_j \in \mathbf{x}$  for  $j \leq n$ ,  $x_j \in \mathbf{y}$  for  $j > n$ ,  $\gamma_i$  is the rate constant, and  $g_{ij}$  is the kinetic order for each  $x_j$ . Equation (2) can be expressed concisely as:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}(t), \mathbf{y}(t); \theta), \mathbf{x}(0) = \mathbf{x}_0, \mathbf{y}(0) = \mathbf{y}_0, \tag{5}$$

where the function  $\mathbf{f}$  is assumed to be continuous and differentiable in all its arguments  $\mathbf{x}$ ,  $\mathbf{y}$ , and  $\theta$ . This assumption on  $\mathbf{f}$  is satisfied for both of equations (3) and (4).

For a model described by a nonautonomous dynamic system as follows:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}(t), \mathbf{y}(t), t; \theta), \mathbf{x}(0) = \mathbf{x}_0, \mathbf{y}(0) = \mathbf{y}_0. \tag{6}$$

we let  $x_{n+1} = t$  and  $dx_{n+1}/dt = 1$ , equation (6) can be rewritten as equation (5) with  $\mathbf{x}(t) \in \mathbb{R}^{n+1}$ . This is an  $(n + 1)$ -dimensional autonomous dynamic system. Similarly, an  $n$ -dimensional time dependent equation is a special case of an  $(n+ 1)$ -dimensional autonomous dynamic system. Using this trick, we can always remove any time dependence by adding an extra dimension to the system. Thus, without loss of generality, we will consider the autonomous dynamic systems expressed by equation (5) unless stated otherwise.

**ODE solver**

Given a set of ODEs expressed as equation (5) and a set of time points  $T = \{t_i | i = 1, \dots, k\}$ . An ODE solver is to find the value of  $\mathbf{x}(t_i)$ ,  $t_i \in T$  for a given  $\theta$ . Many ODE solvers with variable step size control can be used to solve equation (5). Wang [6] proposed a modified collocation method with Lagrange polynomials as shape functions to transform ODEs into algebraic equations. The whole time domain of the problem is divided into a number of non-overlapping intervals  $[t_{i-1}, t_i]$ ,  $i = 1, \dots, k$ . The unified formulas of the modified collocation method for the subinterval  $[t_{j-1}, t_j]$ ,  $t_{i-1} \leq t_{j-1} < t_j \leq t_i$  can be expressed as

$$\mathbf{x}(t_j) = \mathbf{I}\mathbf{x}(t_{j-1}) + \eta_j \{ \mathbf{D}\mathbf{f}(\mathbf{x}(t_j), \mathbf{y}(t_j); \theta) + \mathbf{D}\mathbf{f}(\mathbf{x}(t_{j-1}), \mathbf{y}(t_{j-1}); \theta) \}, \tag{7}$$

where  $\eta_j$  is the step size in time  $t_j$ ,  $\mathbf{I}$  is an identity-like matrix, and the coefficient matrices  $\mathbf{D}$  and  $\hat{\mathbf{D}}$  depend on the shape functions. The accuracy and efficiency of collocation methods depend largely on the degree of shape functions. The modified collocated equations with piece-

wise linear polynomials transformed from equation (5) for each subinterval  $[t_{j-1}, t_j]$ ,  $t_{i-1} \leq t_{j-1} < t_j \leq t_i$  have the same formulas as the modified Euler method:

$$\mathbf{x}(t_j) = \mathbf{x}(t_{j-1}) + \frac{1}{2} \eta_j \{ \mathbf{f}(\mathbf{x}(t_j), \mathbf{y}(t_j); \theta) + \mathbf{f}(\mathbf{x}(t_{j-1}), \mathbf{y}(t_{j-1}); \theta) \}. \tag{8}$$

Instead of solving the ODEs in equation (5) directly, we find the solution of algebraic equations in equation (8) step-by-step for each time interval  $[t_{i-1}, t_i]$ ,  $i = 1, \dots, k$ . The solution obtained from equation (8) is a good approximation solution of ODEs in equation (5) when the step size  $\eta_j$  is small enough. How to decide the step size is an important problem for all ODE solvers. A larger step size can cause the solution to be inaccurate and divergent, and a smaller step size is inefficient for the computations. Wang [6] uses the Newton-Raphson method with step length restriction to solve equation (8). The restricted step size is fixed and computed by trial and error. To overcome this drawback, we propose an adaptive step size control approach based on the Banach fixed point theorem for the modified collocation method in this paper. We will show that this approach can determine the step size automatically and efficiently when computing the solutions and dynamic sensitivities of equation (5) simultaneously.

The Banach fixed point theorem and some terminologies for describing the theorem are defined below.

**Definition 1.** Metric space [32]

A metric space  $(X, d)$  is a set  $X$  where a notion of distance  $d$  (called a metric) between elements of the set is defined.

**Definition 2.** Cauchy sequence [32]

A sequence  $(x_n)$  in a metric space  $(X, d)$  is said to be Cauchy if for every  $\varepsilon > 0$  there is a positive integer  $N$  such that for all natural numbers  $m, n > N$ , the distance  $d(x_m, x_n)$  is less than  $\varepsilon$ .

**Definition 3.** Complete metric space [32]

A metric space  $(X, d)$  in which every Cauchy sequence has a limit in  $X$  is called complete.

**Theorem 4.** Banach fixed point theorem [32]

Let  $(X, d)$  be a non-empty complete metric space. A mapping  $\psi: X \rightarrow X$  is called a contraction on  $X$  if there is a nonnegative real number  $q < 1$  such that for all  $a, b$  in  $X$

$$d(\psi(a), \psi(b)) \leq q \cdot d(a, b). \tag{9}$$

The contraction  $\psi$  on  $X$  admits one and only one fixed point  $x^* \in X$  such that  $\omega(x^*) = x^*$ .

We now show how to decide the step size  $\eta_j$  in equation (8) based on the Banach fixed point theorem. Equation (8) is an implicit expression of  $\mathbf{x}$ , and it can be rewritten as

$$\begin{aligned} \mathbf{x}(t_j) &= \frac{1}{2} \eta_j \mathbf{f}(\mathbf{x}(t_j), \mathbf{y}(t_j); \theta) + \mathbf{c} \\ &= \mathbf{g}(\mathbf{x}(t_j), \mathbf{y}(t_j); \theta), \end{aligned} \tag{10}$$

where  $\mathbf{c} = \mathbf{x}(t_{j-1}) + 0.5 \eta_j \mathbf{f}(\mathbf{x}(t_{j-1}), \mathbf{y}(t_{j-1}); \theta)$  is a constant vector. When the values of independent variable  $\mathbf{y}$  and  $\theta$  are given, the solution of  $\mathbf{x} = \mathbf{g}(\mathbf{x})$  can be found by an iteration process. A sequence of values of  $\mathbf{x}(t_j)$  is obtained using the iterative rule. If  $\mathbf{x}^i(t_j)$  tends to a limit  $\mathbf{x}^*(t_j)$  when  $i \rightarrow \infty$ , it is the answer of  $\mathbf{x} = \mathbf{g}(\mathbf{x})$  and called a fixed point of the function  $\mathbf{g}(\mathbf{x})$ . Let  $X$  be the set of  $\mathbf{x}^i(t_j)$ ,  $i = 1, \dots, \infty$  and  $d(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\|_p$  where  $\mathbf{a}$  and  $\mathbf{b}$  are arbitrary  $\mathbf{x}^i(t_j)$  and  $\|\cdot\|_p$  is the  $p$ -norm. Then  $(X, d)$  forms a non-empty complete metric space. If  $\mathbf{g}$  is a contraction on  $X$ , the Banach fixed point theorem guarantees the existence of a fixed point and the convergence of the iteration process to that fixed point. By the equation (9) and the definition of distance function  $d$ , for  $\mathbf{a}, \mathbf{b} \in X$  we obtain

$$\|\mathbf{g}(\mathbf{a}) - \mathbf{g}(\mathbf{b})\|_p \leq q \|\mathbf{a} - \mathbf{b}\|_p, \quad q < 1. \tag{11}$$

We suppose that  $\mathbf{g}(\mathbf{x})$  is a continuous and differentiable function on  $X$ . By the generalized mean value theorem and the definition of matrix norm, we have

$$\|\mathbf{g}(\mathbf{a}) - \mathbf{g}(\mathbf{b})\|_p \leq \left\| \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right\|_p \|\mathbf{a} - \mathbf{b}\|_p. \tag{12}$$

Comparing equations (11) with (12), we obtain

$$\left\| \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right\|_p < 1, \tag{13}$$

where  $\|\cdot\|_p$  is the  $p$ -norm of a matrix. By substitution of the term on the right of the equal sign in equation (10) for  $\mathbf{g}$ , equation (13) can be rewritten as

$$\frac{1}{2} \eta_j \left\| \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{y}; \theta)}{\partial \mathbf{x}} \right\|_p < 1. \tag{14}$$

This equation is used to compute the maximum  $\eta_j$  with  $p = 2$  when the process of finding the solution of equation (8) is in progress. The Jacobian matrix  $\partial \mathbf{f} / \partial \mathbf{x}$  in equation (14) must be evaluated at each time  $t_j$ . The computation of the Jacobian matrix can be done by evaluating the analytic formula of the partial derivative of  $\mathbf{f}$  with respect to  $\mathbf{x}$  which is user-provided, or by the finite difference approximation. For the GMA systems, the model equations are

expressed in power-law and the value of the Jacobian matrix can be straightforwardly obtained using the analytic formula as follows:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = N \begin{bmatrix} g_{11} x_1^{-1} v_1 & g_{12} x_2^{-1} v_1 & \cdots & g_{1n} x_n^{-1} v_1 \\ g_{21} x_1^{-1} v_2 & g_{22} x_2^{-1} v_2 & \cdots & g_{2n} x_n^{-1} v_2 \\ \vdots & \vdots & \ddots & \vdots \\ g_{q1} x_1^{-1} v_q & g_{q2} x_2^{-1} v_q & \cdots & g_{qn} x_n^{-1} v_q \end{bmatrix}.$$

where  $N$  is the stoichiometric matrix,  $v_i$  is the  $i^{\text{th}}$  element of  $\mathbf{v} \in \mathbb{R}^q$  and  $g_{ij}$  is the kinetic order for each  $x_j \in \mathbf{x}$  in  $v_i$ . For efficiency, we approximate the value of 2-norm of the Jacobian matrix  $\partial \mathbf{f} / \partial \mathbf{x}$  by  $\|\partial \mathbf{f} / \partial \mathbf{x}\|_{\Delta}$ , where  $n$  is the dimension of  $\mathbf{x}$  and  $\|\partial \mathbf{f} / \partial \mathbf{x}\|_{\Delta}$  is the maximum absolute value of the element of the Jacobian matrix. The proposed algorithm AMCM, Adaptive Modified Collocation Method, is shown as follows

**Algorithm AMCM**

**Input:**

1. A set of  $n$  ordinary differential equations  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{y})$  with  $n$  dependent variables  $x_i$ ,  $i = 1, \dots, n$  and  $m$  independent variables  $y_i$ ,  $i = 1, \dots, m$ .
2. Two order sets  $\mathbf{x}_0 = \{x_i(t_0) | i = 1, \dots, n\}$  of initial values of  $\mathbf{x}$  and  $\mathbf{y}_0 = \{y_i(t_0) | i = 1, \dots, m\}$  of initial values of  $\mathbf{y}$ .
3. An order set  $T = \{t_1, \dots, t_k\}$  of sampling points,  $t_i$ ,  $1 \leq i \leq k$  is the sampling time of the solution of each ODE,  $k$  is the number of sampling points.
4. A tolerance  $\varepsilon$

**Output:** The set of solutions of dependent variables at each sampling time.

• For each sampling time  $t_i$  in  $T$ .

1.  $\eta_j \leftarrow t_i - t_{i-1}$ ,  $d_t \leftarrow 0$ ,  $\mathbf{x}^c \leftarrow \mathbf{x}(t_{i-1})$ ,  $\mathbf{y}^c \leftarrow \mathbf{y}(t_{i-1})$
2. Repeat the following steps until  $d_t = t_i - t_{i-1}$ .

(a)  $\mathbf{x}^p \leftarrow \mathbf{x}^c$ ,  $\mathbf{y}^p \leftarrow \mathbf{y}^c$

(b) Evaluate the Jacobin matrix  $\mathbf{A} \leftarrow \frac{\partial \mathbf{f}(\mathbf{x}^p, \mathbf{y}^p)}{\partial \mathbf{x}}$

(c) Compute the upper bound  $\mu$  of the value of  $\|\mathbf{A}\|_2$  by  $\sqrt{n(m+n)} \|\mathbf{A}\|_{\Delta}$ ,  $\|\mathbf{A}\|_{\Delta} \equiv \max_{i,j} |a_{ij}|$

(d) If  $\mu * \varepsilon \geq 1$ , it means the ODEs are stiff, then exit this algorithm.

(e) If  $\mu * \eta_j > 1$ , then update  $\eta_j$  with  $0.9/\mu$

(f) Call iteration algorithm to compute the value of  $\mathbf{x}^c$  stepped forward  $\eta_j$  from  $\mathbf{x}^p$ .

(g) If the iteration algorithm succeeds in computing  $\mathbf{x}^c$ , then  $d_t \leftarrow d_t + \eta_j$  and  $\eta_j \leftarrow t_i - t_{i-1} - d_t$ , otherwise exist this algorithm.

3.  $\mathbf{x}(t_i) \leftarrow \mathbf{x}^c, \mathbf{y}(t_i) \leftarrow \mathbf{y}^c$ .

• return  $\mathbf{x}(t_i), i = 1, \dots, k$ .

**End of Algorithm AMCM**

**Algorithm Iteration**

**Input:**

1. A set of  $n$  ordinary differential equations  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{y})$ .

2.  $\mathbf{x}(t), \mathbf{y}(t), \eta_j$  and the iteration limitation.

**Output:**  $\mathbf{x}(t + \eta_j)$ .

1. Evaluate the value of  $\mathbf{f}(\mathbf{x}(t), \mathbf{y}(t))$ .
2.  $\mathbf{x}(t + \eta_j) \leftarrow \mathbf{x}(t) + \mathbf{f}(\mathbf{x}(t), \mathbf{y}(t)) * \eta_j$ .
3.  $\mathbf{y}(t + \eta_j) \leftarrow \mathbf{y}(t) + \mathbf{f}(\mathbf{x}(t), \mathbf{y}(t)) * \eta_j$ .
4. Repeat the following steps until the iteration limitation is reached or the value of  $\mathbf{x}(t + \eta_j)$  converges.

(a) Evaluate the value of  $\mathbf{f}(\mathbf{x}(t + \eta_j), \mathbf{y}(t + \eta_j))$ .

(b)  $\mathbf{x}(t + \eta_j) \leftarrow \mathbf{x}(t) + 0.5 * \eta_j * (\mathbf{f}(\mathbf{x}(t), \mathbf{y}(t)) + \mathbf{f}(\mathbf{x}(t + \eta_j), \mathbf{y}(t + \eta_j)))$

5. If the iteration limitation is reached, then exit this algorithm; otherwise, return  $\mathbf{x}(t + \eta_j)$ .

**End of Algorithm Iteration**

**Dynamic sensitivity Solver**

For a model described by equation (5), the absolute parameter sensitivity  $s(x_i, \theta_j)$  of dependent variable  $x_i \in \mathbf{x}$  with respect to parameter  $\theta_j \in \theta$  is defined as

$$s(x_i, \theta_j) = \lim_{\Delta\theta_j \rightarrow 0} \frac{x_i(t; \theta_j + \Delta\theta_j) - x_i(t; \theta_j)}{\Delta\theta_j}, \quad (15)$$

where  $x_i(t; \theta_j + \Delta\theta_j)$  is the  $i^{th}$  component of the solution of equation (5) with an increment  $\Delta\theta_j$  on the  $j^{th}$  parameter. The function  $x_i(t; \theta_j + \Delta\theta_j)$  can be expanded into a Taylor series as follows:

$$x_i(t; \theta_j + \Delta\theta_j) = x_i(t; \theta_j) + \frac{\partial x_i(t; \theta_j)}{\partial \theta_j} \Delta\theta_j + \frac{\partial^2 x_i(t; \theta_j + \xi \Delta\theta_j)}{\partial \theta_j^2} \frac{\Delta\theta_j^2}{2}, \quad (16)$$

where  $0 < \xi < 1$ . If  $\Delta\theta_j$  is sufficiently small, the last term of equation (16) can be truncated, leading to a linear approximation of  $x_i(t; \theta_j + \Delta\theta_j)$ . Substituting the linear approximation of  $x_i(t; \theta_j + \Delta\theta_j)$  into equation (15) leads to

$$s(x_i, \theta_j) = \frac{\partial x_i(t; \theta_j)}{\partial \theta_j}.$$

This is defined as the first-order local sensitivity of  $x_i$  with respect to  $\theta_j$  [33]. The relative parameter sensitivity  $S(x_i, \theta_j)$  of  $x_i$  with respect to  $\theta_j$  is defined as

$$S(x_i, \theta_j) = \frac{\partial \ln x_i(t)}{\partial \ln \theta_j} = \frac{\theta_j}{x_i} s(x_i, \theta_j).$$

Similar to the parameter sensitivity, the absolute log gain  $l(x_i, \gamma_j)$  and log gain  $L(x_i, \gamma_j)$  of  $x_i \in \mathbf{x}$  with respect to  $\gamma_j \in \mathbf{y}$  are expressed respectively as follows:

$$l(x_i, \gamma_j) = \frac{\partial x_i(t)}{\partial \gamma_j},$$

$$L(x_i, \gamma_j) = \frac{\partial \ln x_i(t)}{\partial \ln \gamma_j} = \frac{\gamma_j}{x_i} l(x_i, \gamma_j).$$

Once the local sensitivity is known, the calculation of the relative sensitivity is straightforward. So, for briefing, we limit our explanation on the absolute sensitivity only below.

The absolute dynamic sensitivity of  $x_i$  with respect to  $\theta_j$  is given as

$$\frac{ds(x_i, \theta_j)}{dt} = \sum_{k=1}^n \frac{\partial f_i}{\partial x_k} s(x_k, \theta_j) + \frac{\partial f_i}{\partial \theta_j}, \quad (17)$$

where  $f_i$  is the  $i^{th}$  element of  $\mathbf{f}$  [34]. The absolute dynamic log gain of  $x_i$  with respect to  $\gamma_j$  is similar to equation (17) by replacing  $\theta_j, s(x_i, \theta_j)$  with  $\gamma_j, l(x_i, \gamma_j)$  respectively when  $\gamma_j(t)$  is a constant. In the case which  $\gamma_j(t)$  is a continuous time-dependent function, the whole time domain of  $\gamma_j(t)$  is partitioned into  $N_u$  time intervals  $(t_{k-1}, t_k), k = 1, \dots, N_u$ . Function  $\gamma_j(t)$  is parameterized by the piecewise constant functions  $\omega_k(t), k = 1, \dots, N_u$ , as follows:

$$\gamma_j(t) \approx \sum_{k=1}^{N_u} u_k \omega_k(t), \quad (18)$$

where  $u_k, k = 1, \dots, N_u$  are constant input control parameters and

$$\omega_k(t) = \begin{cases} 1, & t_{k-1} \leq t < t_k \\ 0, & \text{otherwise.} \end{cases}$$

The continuous time-dependent function  $\gamma_j(t)$  is approximated by equation (18). The dynamic log gain  $l(x_i, \gamma_j)$  of infinite dimension is transferred into  $N_u$  dynamic log gains of dimension one with respect to the input control parameters, and expressed as.

$$\frac{dl(x_i, u_j)}{dt} = \begin{cases} \sum_{k=1}^n \frac{\partial f_i}{\partial x_k} l(x_k, u_j) + \frac{\partial f_i}{\partial u_j}, & t_{j-1} \leq t < t_j \\ \sum_{k=1}^n \frac{\partial f_i}{\partial x_k} l(x_k, u_j), & \text{otherwise,} \end{cases}$$

where  $j = 1, \dots, N_u$ .

We extended the proposed algorithm AMCM to compute the dynamic sensitivities. The dynamic sensitivities with respect to parameters (include the initial conditions) and absolute dynamic log gains can be computed simultaneously. Let  $\mathbf{u}$  be the vector of input control parameters and  $\mathbf{r}$  be an  $N_r$  dimensional vector of constants which contains constant independent variables in  $\mathbf{y}$  and the input control parameters in  $\mathbf{u}$ . When all components of  $\mathbf{y}$  are constant, the vector  $\mathbf{r}$  is equal to  $\mathbf{y}$ . Let  $\mathbf{z}$  be an  $n + p + N_r$  dimensional vector which contains model parameters  $\theta$ , initial conditions of dependent variables  $\mathbf{x}_0$  and constant input in  $\mathbf{r}$ .  $\Phi$  indicates a matrix of size  $n \times (n + p + N_r)$  which contains the absolute sensitivities with respect to model parameters and initial conditions, and the absolute log gains with respect to constant independent variables and input control parameters.  $\mathbf{z}$  and  $\Phi$  have the following form:

$$\mathbf{z} = \begin{bmatrix} \theta \\ \mathbf{x}_0 \\ \mathbf{r} \end{bmatrix}, \quad \Phi(t) = \frac{\partial \mathbf{x}(t)}{\partial \mathbf{z}} = \left[ \frac{\partial \mathbf{x}(t)}{\partial \theta}, \frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}_0}, \frac{\partial \mathbf{x}(t)}{\partial \mathbf{r}} \right].$$

The model equations are rewritten as

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}(t), \mathbf{r}; \theta), \mathbf{x}(0) = \mathbf{x}_0.$$

The sensitivity equations in matrix form can be derived by applying the chain rule to the derivative of  $\Phi$ :

$$\frac{d\Phi}{dt} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Phi(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{z}}, \quad (19)$$

$$\Phi(0) = \begin{bmatrix} \mathbf{0}_{n \times p}, \mathbf{I}_{n \times n}, \mathbf{0}_{n \times N_r} \end{bmatrix}.$$

Let  $\phi_i$  be the  $i^{\text{th}}$  column vector of  $\Phi$  and  $z_i$  be the  $i^{\text{th}}$  element of  $\mathbf{z}$ . The matrix sensitivity equations in equation (19) are rearranged into a vector of linear ODEs:

$$\frac{d\phi}{dt} = \mathbf{M}\phi(t) + \mathbf{w},$$

$$\phi(0) = \begin{bmatrix} \phi_1(0) \\ \phi_2(0) \\ \vdots \\ \phi_{n+p+N_r}(0) \end{bmatrix}, \quad (20)$$

Where

$$\phi = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{n+p+N_r} \end{bmatrix}_{n(n+p+N_r) \times 1},$$

$$\mathbf{w} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial z_1} \\ \frac{\partial \mathbf{f}}{\partial z_2} \\ \vdots \\ \frac{\partial \mathbf{f}}{\partial z_{n+p+N_r}} \end{bmatrix}_{n(n+p+N_r) \times 1},$$

$$\mathbf{M} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial \mathbf{x}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \end{bmatrix}_{n(n+p+N_r) \times n(n+p+N_r)}.$$

In order to find the solution of equation (20), the whole time domain is divided into a number of non-overlapping time intervals  $[t_{i-1}, t_i], i = 1, \dots, k$ . The sensitivity equations in equation (20) are transformed to algebraic equations using modified collocation method with piecewise linear polynomials as shape functions for each subinterval  $[t_{j-1}, t_j], t_{i-1} \leq t_{j-1} < t_j \leq t_i$ :

$$\phi(t_j) = \phi(t_{j-1}) + \frac{1}{2} \eta_j \{ \mathbf{M}(t_j) \phi(t_j) + \mathbf{w}(t_j) + \mathbf{M}(t_{j-1}) \phi(t_{j-1}) + \mathbf{w}(t_{j-1}) \}, \quad (21)$$

where  $\eta_j$  is the step size in time  $t_j$ . This equation can be rewritten as:

$$\phi(t_j) = \frac{1}{2}\eta_j \mathbf{M}(t_j)\phi(t_j) + \mathbf{c} = \mathbf{h}(\phi(t_j)), \quad (22)$$

where the constant vector

$$\mathbf{c} = \phi(t_{j-1}) + \frac{1}{2}\eta_j\{\mathbf{w}(t_j) + \mathbf{M}(t_{j-1})\phi(t_{j-1}) + \mathbf{w}(t_{j-1})\}.$$

The value of  $\mathbf{w}(t_j)$  can be evaluated if the values of  $\mathbf{x}(t_j)$  have been computed when solving equation (8) by the ODE solver. The value of the  $\mathbf{M}(t_j)$  in equation (22) must be calculated before the solution of equation (21) can be obtained by an iteration method. When the value of the Jacobian matrix  $\partial f/\partial \mathbf{x}$  in time  $t_j$  is known, the value of the  $\mathbf{M}(t_j)$  can be obtained straightforwardly. There is no requirement for computing the value of the Jacobian matrix here due to it has been computed for step size control when solving equation (8) by the ODE solver.

Equation (22) is similar to equation (10), we can apply the fixed point theorem to  $\phi = \mathbf{h}(\phi)$  to get the maximum  $\eta_j$  satisfying the following condition

$$\left\| \frac{\partial \mathbf{h}(\phi)}{\partial \phi} \right\|_p = \frac{1}{2}\eta_j \|\mathbf{M}(t_j)\|_p < 1.$$

According to the definition of the matrix norm, it is easy to verify that  $\|\mathbf{M}\|_p = \|\partial f/\partial \mathbf{x}\|_p$  and we have

$$\left\| \frac{\partial \mathbf{h}(\phi)}{\partial \phi} \right\|_p = \frac{1}{2}\eta_j \left\| \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{r}; \theta)}{\partial \mathbf{x}} \right\|_p < 1. \quad (23)$$

Equation (23) is the same as equation (14), so the same criterion is used to determine the step size  $\eta_j$  for computing the time course and sensitivity profile with the guarantee of convergence and existence of a fixed point. The dynamic sensitivities can be obtained directly by solving equation (21) after the time profile of  $\mathbf{x}$  has been computed and the step size  $\eta_j$  has been decided by the ODE solver.

### Competing interests

The authors declare that they have no competing interests.

### Authors' contributions

WHW developed and implemented the algorithm and drafted the manuscript. FSW conceived of the study, participated in its design and coordination, and helped to draft the manuscript. MSC assisted in developing the algorithm and finalizing the manuscript. All authors read and approved the final manuscript.

### Acknowledgements

The financial support from the National Science Council, Taiwan, ROC (Grant NSC97-2221-E-194-010-MY3 and NSC97-2627-B-194-001), is highly appreciated.

This article has been published as part of *BMC Bioinformatics* Volume 9 Supplement 12, 2008: Asia Pacific Bioinformatics Network (APBioNet) Seventh International Conference on Bioinformatics (InCoB2008). The full contents of the supplement are available online at <http://www.biomedcentral.com/1471-2105/9?issue=S12>.

### References

- Albrecht P: **A New Theoretical Approach to Runge-Kutta Methods.** *SIAM Journal on Numerical Analysis* 1987, **24**:391-406.
- Albrecht P: **The Runge-Kutta Theory in a Nutshell.** *SIAM Journal on Numerical Analysis* 1996, **33**:1712-1735.
- Inc M, Bildik N, Bulut H: **A comparison of numerical ODE solvers based on Euler methods.** *Math Comput Appl* 1998, **3**:153-159.
- Gerald CF, Wheatley PO: *Applied numerical analysis* Addison-Wesley; 1994.
- Villadsen JV, Stewart WE: **Solution of boundary-value problems by orthogonal collocation.** *Chem Eng Sci* 1967, **22**:1483-1501.
- Wang FS: **A modified collocation method for solving differential-algebraic equations.** *Applied Mathematics and Computation* 2000, **116**:257-278.
- Savageau MA: **Parameter sensitivity as a criterion for evaluating and comparing the performance of biochemical systems.** *Nature* 1971, **229**:542-544.
- Savageau MA: **The behavior of intact biochemical control systems.** *Current Topics in Cellular Regulation* 1972, **6**:63-129.
- Voit EO: *Computational analysis of biochemical systems: a practical guide for biochemists and molecular biologists* Cambridge, UK: Cambridge University Press; 2000.
- Savageau M: *Biochemical Systems Analysis. A Study of Function and Design in Molecular Biology* Reading, MA: Addison-Wesley; 1976.
- Kacser H, Burns JA: **The control of flux.** *Symp Soc Exp Biol* 1973, **27**:65-104.
- Fell DA: **Metabolic control analysis: a survey of its theoretical and experimental development.** *Biochem J* 1992, **286**:313-330.
- Heinrich R, Schuster S: *The Regulation of Cellular Systems* New York: Chapman & Hall; 1996.
- Caracotsios M, Stewart WE: **Sensitivity Analysis of Initial Value Problems with Mixed Ode's and Algebraic Equations.** *Comput Chem Eng* 1985, **9**:359-365.
- Dunker AM: **The decoupled direct method for calculating sensitivities coefficients in chemical kinetics.** *J Chem Phys* 1984, **81**:2385-2393.
- Dougherty EP, Hwang JT, Rabitz H: **Further developments and applications of the Green's function method of sensitivity analysis in chemical kinetics.** *J Chem Phys* 1979, **71**:1794-1808.
- Dickinson RP, Gelinas RJ: **Sensitivity analysis of ordinary differential equation systems. A direct method.** *J Comput Phys* 1976, **21**:123-143.
- Leis JR, Kramer MA: **The simultaneous solution and sensitivity analysis of systems described by ordinary differential equations.** *ACM Trans Math Softw* 1988, **14**:45-60.
- Mauch K, Arnold S, Reuss M: **Dynamic sensitivity analysis for metabolic systems.** *Chemical Engineering Science* 1997, **52**:2589-2598.
- Ghosh A, Miller D, Zou R, Pais H, Sokhansanj B, Kriete A: **Integrated Spatio-temporal Model of Cell Signaling.** *FOSSBE* 2005.
- Hwang JT, Dougherty EP, Rabitz S, Rabitz H: **The Green's function method of sensitivity analysis in chemical kinetics.** *J Chem Phys* 1978, **69**:5180-5191.
- Ingalls BP, Sauro HM: **Sensitivity analysis of stoichiometric networks: an extension of metabolic control analysis to non-steady state trajectories.** *Journal of Theoretical Biology* 2003, **222**:23-36.
- Shen J: **A direct method of calculating sensitivity coefficients of chemical kinetics.** *J Chem Phys* 1999, **111**:7209-7214.
- Zou R, Ghosh A: **Automated sensitivity analysis of stiff biochemical systems using a fourth-order adaptive step size**



- Rosenbrok integration method.** *AIEE Proc Sys Biol* 2006, **153**:79-90.
25. Shiraishi F, Takeuchi H, Hasegawa T, Nagasue H: **A Taylor-series solution in Cartesian space to GMA-system equations and its application to initial-value problems.** *Applied Mathematics and Computation* 2002, **127**:103-123.
  26. Shiraishi F, Hatoh Y, Irie T: **An efficient method for calculation of dynamic logarithmic gains in biochemical systems theory.** *Journal of Theoretical Biology* 2005, **234**:79-85.
  27. Gorbunov VK: **The parameterization method for optimal control problems.** *Comput Math Math Phys* 1979, **19**:212-224.
  28. Gorbunov VK, Lutoshkin IV: **The parameterization method in optimal control problems and differential-algebraic equations.** *Journal of Computational and Applied Mathematics* 2006, **185**:377-390.
  29. Li R, Teo KL, Wong KH, Duan GR: **Control parameterization enhancing transform for optimal control of switched systems.** *Mathematical and Computer Modelling* 2006, **43**:1393-1403.
  30. Teo KL, Jennings LS, Lee HW, Rehbock V: **The control parameterization enhancing transform for constrained optimal control problems.** *J Austral Math Soc Ser B* 1999, **40**:314-335.
  31. Wang FS, Su TL, Jang HJ: **Hybrid differential evolution for problems of kinetic parameter estimation and dynamic optimization of an ethanol fermentation process.** *Ind Eng Chem Res* 2001, **40**:2876-2885.
  32. Kreyszig E: *Introductory Functional Analysis with Applications* Wiley; 1989.
  33. Varma A, Morbidelli M, Wu H: *Parameter sensitivity in chemical systems* Cambridge: Cambridge University Press; 1999.
  34. Tomovic R, Vukobratovic M: *General Sensitivity Theory* New York: American Elsevier; 1972.

Publish with **BioMed Central** and every scientist can read your work free of charge

"BioMed Central will be the most significant development for disseminating the results of biomedical research in our lifetime."

Sir Paul Nurse, Cancer Research UK

Your research papers will be:

- available free of charge to the entire biomedical community
- peer reviewed and published immediately upon acceptance
- cited in PubMed and archived on PubMed Central
- yours — you keep the copyright

Submit your manuscript here:  
[http://www.biomedcentral.com/info/publishing\\_adv.asp](http://www.biomedcentral.com/info/publishing_adv.asp)

