Avoiding the Finite Difference Sensitivity Analysis Deathtrap by Using the Complex-step Derivative Approximation Technique

D.J.W. De Pauw [a], P.A. Vanrolleghem[a b]

[a]BIOMATH, Department of Applied Mathematics, Biometrics and Process Control
Ghent University, Coupure Links 653, 9000 Gent, Belgium

[b]model*EAU*, Département de Génie Civil
Pavillon Pouliot, Université Laval, Québec, G1K 7P4, QC, Canada

e-mail: dirk.depauw@ugent.be

Abstract: In this paper, the complex-step derivative approximation technique will be used for calculating local sensitivity functions. This technique is compared to the finite difference approximation, probably the most used local sensitivity analysis technique. For this comparison, 4 biotechnological models with varying model complexity were used. A well known problem of the finite difference approximation is the choice of a suitable perturbation factor in order to avoid non-linear model effects or numerical errors due to the subtraction of almost equal numbers. The main advantage of the complex-step derivative approximation technique is that it is not susceptible to errors introduced by small perturbation factors, ruling out the entire search for optimal perturbation factors. However, the main disadvantage is an important execution time increase for large models.

Keywords: Local sensitivity analysis, complex-step, finite difference, numerical methods, biotechnology.

1. INTRODUCTION

Sensitivity analysis studies the ”sensitivity” of the outputs of a system to changes in the parameters, inputs or initial conditions. Sensitivity analysis can be divided into two large categories: local and global sensitivity analysis. Local sensitivity analysis methods refer to small changes of parameters, while global methods refer to the effect of simultaneous, possibly orders-of-magnitude parameter changes. The main focus of this paper is on local sensitivity analysis techniques.

The general equation of the systems described here is given by (1)-(2):

$$\begin{aligned} dx/dt &= f\left(x,\theta,u,t\right), \quad x\left(t_0\right)=x_0 & (1)\\ y &= g\left(x,\theta,u,t\right) & (2)\end{aligned}$$

where x is a vector of state variables, θ a vector of parameters, y a vector of outputs, u a vector of inputs and t the independent variable.

The sensitivity of a state variable y to a parameter θ can be expressed as a sensitivity function (3). The state variable y is called sensitive to θ if small changes in θ produce significant changes in y.

$$S(t) = \partial y(t)/\partial\theta \qquad (3)$$

This partial derivative can be analytically solved if the analytical solution of (1)-(2) is known. Unfortunately, this is rarely the case and numerical methods have to be used in order to approximate the sensitivity function (3). Local sensitivity analysis techniques evaluate this partial derivative at one specific set of parameter values, also called the nominal parameter set.

2. SIMULATION ENVIRONMENT AND MODELS

Throughout this paper several models related to biological wastewater treatment will be used for illustrative purposes. These models have been implemented in the modelling and simulation software WEST (World wide Engine for Simulation, Training and automation) [Vanhooren et al., 2003] which was also used to perform the sensitivity analysis.

First, a simple biotechnological model will be used in which microbial biomass (X) is growing on a limiting substrate (S). The rate at which the bacteria grow, is modelled by the well known Michaelis-Menten or Monod kinetics [Holmberg, 1982]. This model consists of 2 differential equations, given by (4)-(5) and 4 parameters: the maximum specific growth rate (μ_{max}), the yield coefficient (Y), the decay coefficient (K_d) and the half saturation concentration for the substrate at which the biomass grows at half the growth rate (K_S).

$$\frac{dS}{dt} = -\frac{1}{Y}\frac{\mu_{max}S}{K_S+S}X \tag{4}$$

$$\frac{dX}{dt} = \frac{\mu_{max}S}{K_S+S}X - K_d X \tag{5}$$

A more complex anaerobic digestion model was also considered [Bernard et al., 2001]. This model is a two-step (acidogenesis-methanisation) mass-balance model describing the growth of two biomass species: acidogenic bacteria (X_1) and methanogenic bacteria (X_2). In a first step, the acidogenic bacteria consume organic substrate (S_1) and produce CO_2 and volatile fatty acids (S_2). The methanogenic bacteria use the volatile fatty acids in a second step for growth and produce carbon dioxide (CO_2) and methane (CH_4). Alkalinity and pH are also modelled because of their importance in anaerobic digestion processes. Overall this model consists of 6 differential equations and 19 parameters.

The third model used, was a sequencing batch reactor (SBR) model describing nitrogen and phosphorous removal [Sin et al., 2004]. This activated sludge model was built on the basis of Activated Sludge Model No. 1 and 2d [Henze et al., 2000]. Nitrogen transformations were incorporated as an integral module following an approach similar to ASM1. The particulate nitrogen is first hydrolyzed to soluble organic nitrogen and then ammonified to ammonia by heterotrophic biomass. The model consists of 89 parameters and 22 differential equations.

Finally, the relatively complex EU COST (European Co-operation in the Field of Scientific and Technical Research) Simulation Benchmark model [Copp, 2001] was used. This is a wastewater treatment model that was designed to provide an unbiased basis for comparison of control strategies. It was also successfully used for comparing different simulation packages in the wastewater community. The Simulation Benchmark has five biological tanks in series and a secondary settling tank. The biological tanks are modelled by the Activated Sludge Model No. 1 (ASM1) [Henze et al., 2000]. ASM1 has 13 components and 8 processes describing growth and decay of biomass, hydrolysis of organic compounds and ammonification. The secondary settler is modelled using the 1D settling model of Takacs, Patry and Nolasco [Takacs et al., 1991]. The model consists of 145 (5x13+80) differential equations.

3. FINITE DIFFERENCE APPROXIMATION

3.1 Theory and Implementation

The simplest way of calculating local sensitivities is to use the finite difference approximation. This technique is also called the brute force method or indirect method. It is very easy to implement because it requires no extra code beyond the original model solver. The partial derivative defined in (3) can be mathematically formulated by the equation given below (forward difference).

$$\frac{\partial y_i}{\partial \theta_j} = \lim_{\Delta\theta_j \to 0} \frac{y_i(t,\theta_j+\Delta\theta_j) - y_i(t,\theta_j)}{\Delta\theta_j} \tag{6}$$

This equation is only valid if we consider an infinitesimal variation (perturbation) of the parameters, inputs or initial conditions θ $(\Delta\theta_j \to 0)$. Equation (6) shows that the application of the finite difference method requires the solution of the model (1)-(2) using the nominal value of the parameters $y_i(t,\theta_j)$ and p solutions of the equations using perturbed parameters $y_i(t,\theta_j+\Delta\theta_j)$, where p is the number of parameters involved in the sensitivity analysis. It should be noted that only one parameter is perturbed at a time while all others are kept at their nominal value. The sensitivities obtained, actually belong to the $(\theta + \Delta\theta/2)$ parameter set because (6) can also be seen as the average of the sensitivities of the model output y_i at θ_j and $\theta_j + \Delta\theta_j$. If the sensitivities are desired to belong to the nominal values θ_j, (6) should be modified into the central difference formula (7) which requires $2p$ solutions.

$$\frac{\partial y_i}{\partial \theta_j} \approx \frac{y_i(t,\theta_j+\Delta\theta_j) - y_i(t,\theta_j-\Delta\theta_j)}{2\Delta\theta_j} \tag{7}$$

3.2 Problems Choosing a Correct Perturbation Factor

Practically, $\Delta\theta_j$ of (6) and (7) was implemented as the nominal parameter value θ_j multiplied by a user defined perturbation factor ξ. As will be shown below, the choice of this perturbation factor will determine the quality of the sensitivity function. The finite difference approximation of the partial derivative (3) is only valid if the perturbation factor approaches 0. From a theoretical point of view this is correct, but numerically this can never be achieved because of the limited precision of the calculations. If the perturbation factor is taken too small it will result in numerical inaccuracies. On the other hand, $\xi\theta_j$ should not become too large because then the nonlinearity of the model will start to play an important role in the sensitivity calculations. This dilemma is illustrated in Figures 1 and 2 using the sensitivity of the Benchmark autotrophic biomass $(X_{B,A})$ to the maximum autotrophic growth rate (μ_{mA}) for perturbation factors $1E{-}02$ and $1E{-}06$. In each figure the sensitivity function calculated with the optimal perturbation factor $1E{-}04$ is also shown. This perturbation factor was determined using the method developed by De Pauw and Vanrolleghem [2003] which tries to find a balance between numerical inaccuracies and the nonlinearity of the model.

The problem of selecting a suitable perturbation factor was thoroughly investigated by De Pauw and Vanrolleghem [2003]. In their research, it was found that the perturbation factor is parameter dependent and to a lesser extent variable dependent. It was also found that the search for the optimal perturbation factors becomes more difficult with increasing numbers of variables and parameters involved in the sensitivity analysis. Hence, to perform an accurate sensitivity analysis may require extensive tuning of perturbation factors, one at a time.

4. COMPLEX-STEP DERIVATIVE APPROXIMATION METHOD

4.1 Theoretical Background

A possible solution for the difficulties encountered with the selection of perturbation factors in the finite difference method is the use of the complex-step derivative approximation method. Computing sensitivity derivatives using complex variables was first suggested by Lyness and Moler [1967]. For some reasons, such as the inability of compilers to deal with complex arithmetic, this technique was not exploited much until Squire and Trapp [1998] reintroduced it. Since then, the technique has been applied several times. Some fields of application are: aeronautics and astronautics [Martins et al., 2000, 2001, 2003], mechanical engineering [Perez-Foguet et al., 2000; Vatsa, 2000] and chemical engineering Butuk and Pemba [2003]. In this publication, the technique is applied for another purpose: the calculation of dynamic local sensitivity functions of models consisting of algebraic and differential equations.

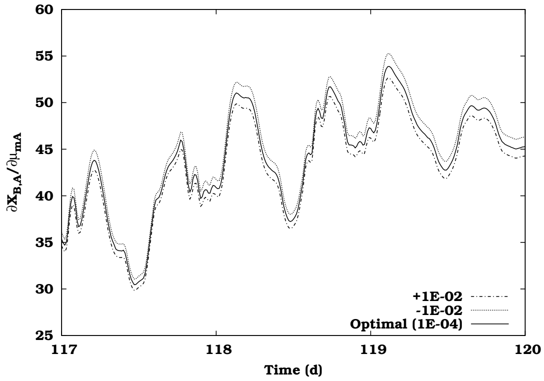

Figure 1. Nonlinear effect: sensitivity of the Benchmark autotrophic biomass $(X_{B,A})$ to the maximum autotrophic growth rate (μ_{mA}) calculated with perturbation factors $\pm 1E{-}02$ and $1E{-}04$.

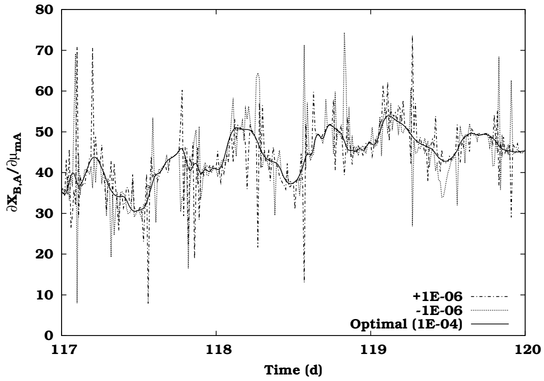

Figure 2. Numerical error effect: sensitivity of the Benchmark autotrophic biomass $(X_{B,A})$ to the maximum autotrophic growth rate (μ_{mA}) calculated with perturbation factors $\pm 1E{-}06$ and $1E{-}04$.

The basic equation for the complex-step derivative approximation method can be derived using the Taylor series expansion of a function in terms of complex variables:

$$y(\theta + i\Delta\theta) = y(\theta) + i\Delta\theta\frac{dy}{d\theta} - \frac{\Delta\theta^2}{2}\frac{d^2y}{d\theta^2} - \frac{i\Delta\theta^3}{6}\frac{d^3y}{d\theta^3} + hot \quad (8)$$

where hot are the higher order terms of the expansion. Isolating the imaginary part of this equation leads to:

$$Im\left[y\left(\theta+i\Delta\theta\right)\right]=\Delta\theta\frac{dy}{d\theta}-\frac{\Delta\theta^3}{6}\frac{d^3y}{d\theta^3}+hot \quad (9)$$

For very small imaginary steps $i\Delta\theta$, the third and higher order terms of the expansion become negligible and the derivative can be written as:

$$\frac{dy}{d\theta}\approx\frac{Im\left[y\left(\theta+i\Delta\theta\right)\right]}{\Delta\theta} \quad (10)$$

Notice that this equation avoids the subtraction of nearly equal numbers, as is the case for the standard finite differencing methods, and therefore does not exhibit the accuracy problems associated with small step sizes. In order for (10) to be valid, the complex function y needs to be analytic which means that the function must be complex differentiable. Some complex functions may fail to be analytic due to the presence of singularities or discontinuities. Martins et al. [2001] studied this problem and found that the complex-step derivative approximation method still produced accurate derivatives in the neighborhood of the singularities and discontinuities.

4.2 Implementation Issues

Implementation of the complex-step derivative approximation method involves the conversion of floating point-value functions (model equations) into their complex equivalent, i.e. the functions need to be modified in such a way that they can accept complex arguments. To be practically usable, this process should be as automatic as possible because manually changing the source code not only is a tedious task, but may also introduce coding errors. The most elegant way of automatic implementation is the use of derived data types and operator overloading. With this technique the data type of the variables in the functions is replaced by a new complex data type and all operators and arithmetic functions are redefined for this new data type. Fortunately, FORTRAN and C++ natively support these functionalities making this technique easily implementable. Martins et al. [2003] provides a single C++ file (complexify.h) that needs to be included with the model source code and contains the required complex data type and arithmetic function redefinitions. Because no further changes to the source code need to be made, this technique is almost as straightforward to implement as the finite difference method. Besides these minor changes, the solvers (integrated in WEST) which integrate the differential equations of the model were also adapted to use the new complex data type.

Performing a sensitivity analysis using this technique in WEST can be summarized as follows:

1. Select variables (y) and parameters (θ) and compose the sensitivity functions.
2. Apply a small perturbation $\Delta\theta$ (e.g. $1E{-}20$) to the imaginary part of one parameter.
3. Run the simulation and retain the trajectories of the imaginary part of all required variables.
4. The sensitivity functions are then calculated by dividing each point of these trajectories by the perturbation $\Delta\theta$.
5. Repeat steps 2-4 for all parameters.

Although this technique only requires p simulations compared to $p{+}1$ simulations required for a forward difference and $2p$ simulations for a central difference, its major drawback is an increased simulation time due to the complex arithmetic. A comparison of the computational cost between the finite difference and the complex-step derivative approximation technique will be given below.
An additional drawback of the complex-step derivative approximation method is that the amount of memory used is approximately doubled due to the introduction of the new complex data type. For the biotechnological models discussed in this paper, this poses no problems since most large models never grow beyond a few MB.

4.3 Numerical Example

In this section the influence of different perturbation factors on the accuracy of the sensitivity calculations will be studied for the complex-step derivative approximation method and compared to the forward and central difference technique. The sensitivity function that will be used for this purpose is the sensitivity of the volatile fatty acids concentration (S_2) of the anaerobic digestion model to the half saturation constant of the volatile fatty acids (K_{S_2}). This sensitivity function was calculated for different perturbation factors, ranging from 1 to $1E{-}17$, using the complex-step derivative approximation method, the forward difference and the central difference technique. The model equations were solved using the Runge Kutta 4 Adaptive Step size Control integration algorithm (RK4ASC) [Press et al., 1992] using an accuracy of $1E{-}06$. In order to quantify the calculation accuracy of each calculated sensitivity function, the sum of relative errors with respect to a reference sensitivity function was calculated. The reference used for this purpose was the sensitivity function obtained from the complex-step derivative approximation method using a perturbation factor of $1E{-}30$.

In Figure 3 the calculated sum of relative errors as a function of the different perturbation factors is shown for the complex-step derivative approximation method, forward difference and central difference sensitivity calculations. As expected, the error of the finite difference methods initially decreases (for perturbation factors 1 to $1E{-}03$) to reach a minimum value for a perturbation factor of $1E{-}03$. For smaller perturbation factors, the error increases again due to the subtractive cancellation errors. Also notice that the central difference reaches a lower error than the forward difference.

For the complex-step derivative approximation method it is clear that the errors decrease with decreasing perturbation factor. At a perturbation factor of $1E{-}07$ the error stabilizes at a value of $\pm 1E{-}15$ which corresponds to the accuracy of a double precision floating point number on a Intel32 architecture. This value is maintained for perturbation factors as low as $1E{-}307$ ($1E{-}308$ being the smallest possible non-zero value available on an Intel32 architecture).

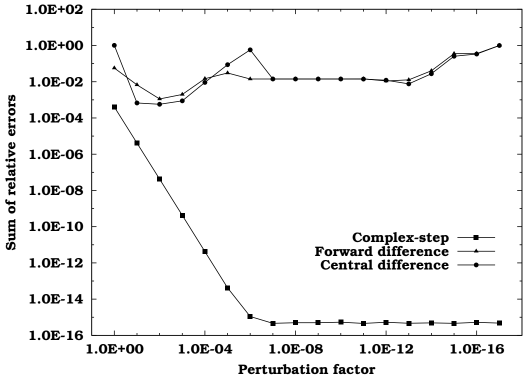

Figure 3. Sum of relative errors of the sensitivity function $\partial S_2/\partial K_{S_2}$ of the anaerobic digestion model for the complex-step derivative approximation method, forward difference and central difference.

4.4 Computational Comparison

As already mentioned, the major drawback of the complex-step derivative approximation method is an increase in the computational cost due to the complex arithmetic of the model. To gain more insight into this problem, the four models used in this paper were compared for the time needed to run a single simulation. All simulations were performed on an Intel PIII 1GHz system running Linux (kernel 2.4.22). Again, the integration algorithm used, was RK4ASC with an accuracy of $1E{-}06$.

Table 1 lists the ratios between the execution times of a single simulation for the complex model (with complex arithmetic) and the normal model (without complex arithmetic). For simple models, like the Monod and the anaerobic model, no noticeable differences in execution speed were detected. However, for more complex models (SBR and Benchmark) large differences in simulation time exist. On average, the execution of the complex model is 30 times slower than the model without complex arithmetic which is much more than the ratios between 1 and 3 reported in literature [Martins et al., 2000; Vatsa, 2000; Burg and Newman III, 2003]. These reported ratios, however, were all related to implementations of the complex-step derivative approximation method on FORTRAN codes as opposed to C++ in this case study. Therefore, inefficient C++ complex arithmetics might explain part of the speed difference, but more research to pinpoint the exact cause is certainly needed.

Table 1. Ratios between the execution times of a single simulation for the complex model (with complex arithmetic) and the normal model (without complex arithmetic).

Model	Ratio
Monod	1
Anaerobic	1
SBR	36
Benchmark	29

The conclusions derived from the simulation speed analysis have obvious implications for the execution speed of a complete sensitivity analysis, which requires the execution of several simulations. However, the increased computational time can be compensated by the fact that only p simulations are required for the complex-step derivative approximation method compared to $p + 1$ simulations for a forward difference and $2p$ for a central difference. Besides this, it has to be mentioned that in a real finite difference application much more simulations are required because of the search for the optimal perturbation factors. This can be very cumbersome and time consuming (especially if many parameters are involved), and requires the execution of many finite difference sensitivity analyses. Even then, it is not guaranteed that an adequate perturbation factor can always be found. This requirement of additional simulations for the finite difference technique makes the complex-step derivative approximation method certainly attractive even with the associated simulation speed increase for large models.

5. CONCLUSIONS

In this paper, two techniques for local sensitivity analysis, the finite difference approximation and the complex-step derivative approximation technique were described and compared. A well-known problem of the finite difference approximation is the choice of a suitable perturbation factor in order to avoid non-linear model effects or numerical errors due to the subtraction of almost equal numbers. The main advantage of the complex-step derivative approximation technique is that it is not susceptible to errors introduced by small perturbation factors, ruling out the entire search for optimal perturbation factors. However, the main disadvantage is an important execution time increase for large models.

6. REFERENCES

Bernard, O., Z. Hadj-Sadok, D. Dochain, A. Genovesi, and J. Steyer. Dynamical model development and parameter identification for an anaerobic wastewater treatment process. *Biotechnology and Bioengineering*, 75(4):424–438, 2001.

Burg, C. and J. Newman III. Computationally efficient, numerically exact design space derivatives via the complex Taylor's series expansion method. *Computers and Fluids*, 32(3):373–383, 2003.

Butuk, N. and J. Pemba. Computing CHEMKIN sensitivities using complex variable. *Journal of Engineering for Gas Turbine and Power - Transactions of the ASME*, 125(3):854–858, 2003.

Copp, J. *The COST Simulation Benchmark: Description and Simulator Manual*. Office for Official Publications of the European Community, Luxembourg, Luxembourg, 2001.

De Pauw, D. and P. Vanrolleghem. Practical aspects of sensitivity analysis for dynamic models. In *Proceedings IMACS 4th MATHMOD Conference*, Vienna, Austria, February 5-7, 2003, 2003.

Henze, M., W. Gujer, T. Matsuo, and M. Van Loosdrecht. *Activated sludge models ASM1, ASM2, ASM2d and ASM3, IWA scientific and technical report no.9*. IWA publishing, London, UK, 2000.

Holmberg, A. On the practical identifiability of microbial growth models incorporating Michaelis-Menten type nonlinearities. *Mathematical Biosciences*, 62(1):23–43, 1982.

Lyness, J. and C. Moler. Numerical differentiation of analytic functions. *SIAM Journal on Numerical Analysis*, 4(2):202–210, 1967.

Martins, J. R. R. A., P. Sturdza, and J. Alonso. The connection between the complex-step derivative approximation and algorithmic differentiation. In *Proceedings of the 39th Aerospace Sciences Meeting*, Reno, USA, 2001.

Martins, J., I. Kroo, and J. Alonso. An automated method for sensitivity analysis using complex variables. In *Proceedings of the 38th Aerospace Sciences Meeting*, Reno, USA, 2000.

Martins, J., P. Sturdza, and J. Alonso. The complex-step derivative approximation. *ACM Transactions on Mathematical Software*, 29(3):245–262, 2003.

Perez-Foguet, A., A. Rodriguez-Ferran, and A. Huerta. Numerical differentiation for local and global tangent operators in computational plasticity. *Computer Methods in Applied Mechanics and Engineering*, 189(1):277–296, 2000.

Press, W., S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical recipes in C*. Cambridge University Press, Cambridge, UK, 1992.

Sin, G., G. Insel, D. Lee, and P. Vanrolleghem. Optimal but robust N and P removal in SBRs: A systematic study of operating scenarios. *Water Science and Technology*, 50(10):97–105, 2004.

Squire, W. and G. Trapp. Using complex variables to estimate derivatives of real functions. *SIAM Review*, 40(1):110–112, 1998.

Takacs, I., G. Patry, and D. Nolasco. A dynamic model of the clarification thickening process. *Water Research*, 25(10):1263–1271, 1991.

Vanhooren, H., J. Meirlaen, Y. Amerlinck, F. Claeys, H. Vangheluwe, and P. Vanrolleghem. WEST: modelling biological wastewater treatment. *Journal of Hydroinformatics*, 5(1):27–50, 2003.

Vatsa, V. Computation of sensitivity derivatives of Navier-Stokes equations using complex variables. *Advances in Engineering Software*, 31(8-9):655–659, 2000.