EURASIAN JOURNAL OF MATHEMATICAL AND COMPUTER APPLICATIONS
ISSN 2306–6172
Volume 3, Issue 3 (2015) 16– 68

THE COMPLEX-STEP METHOD FOR SENSITIVITY ANALYSIS OF NON-SMOOTH PROBLEMS ARISING IN BIOLOGY

H.Banks, K. Bekele-Maxwell, L. Bociu,M.Noorman, K. Tillman

Abstract We investigate the complex-step method as applied to compute sensitivities with respect to model "parameters" for several types of examples. We first consider time delayed differential equations (DDEs) whose sensitivities are known to have lack of smoothness or even discontinuities with respect to parameters such as the delays. The second type of "parameter sensitivity" we consider is that of solutions to partial differential equations (PDEs) with respect to boundary conditions which again may not possess smoothness. These sensitivities are fundamental in any type of boundary control formulation such as those we motivate in Section 4 below. Our main focus here is to evaluate the so-called "complex-step methods" for computing such sensitivities. This is of interest since the complex-step method was derived based on the Cauchy-Riemann equations for analytic complex functions. Our computational findings are compared to those using the standard chain rule-based sensitivity differential equations which can be rigorously developed even for derivatives possessing much less regularity than analyticity. Our findings suggest that the complex-step methods are in very good agreement with the usual sensitivity equation results up to some critical step size we call h_{crit}. They can offer significant savings in computational costs for problems driven by complicated dynamical systems with reasonable parameter size.

Key words: complex-step method, sensitivity analysis, sensitivity equations, delayed differential equations.

AMS Mathematics Subject Classification: 49Q12, 93B35, 34K06, 30D05

1 Introduction

Sensitivity analysis continues to play a fundamental role in modeling, mathematics, statistics and engineering for an incredibly wide range of applications. This topic plays a critical role in widely diverse topics including:

- parameter selection and identifiability [11] and many references therein, [6, 9] in inverse problem formulations;

- statistics and Fisher Information Matrix [13, 25, 34, 35, 62], and in particular asymptotic theories for large sample size problems;

- data set information content studies [12];

- optimal design of experiments [22, 23];

- PDEs and boundary term sensitivity [19, 21] as potential control formulations in biomedical studies–see, for example, recent references on modeling pathologies such as glaucoma in the human eye [26, 32, 37, 42]; such studies are precursors to general boundary control design which are also often studied in a wide range of engineering problems;
- non-smooth problems where one does not have analyticity of response with respect to the "parameters", including delay systems with respect to delays [24], where the usual situation is represented by a lack of smoothness, let alone analyticity; and PDE problems with respect to boundary parameters such as those in aggregate data problems where estimated parameters are probability measures [13, Chapter 6] and the examples and references therein.

In usual sensitivity analyses, one studies how the output of a model is affected by a wide variety of inputs. In these efforts we are concerned with calculating the rates of change in the output variables (solutions or observables) of a system which result from small perturbations in the problem parameters.

We outline an ODE inverse problem framework to illustrate the basic ideas even though we will be concerned with more complicated systems and sensitivities to delays in a DDE and to boundary inputs in PDE formulations. Consider an n-dimensional vector system

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= \mathbf{g}(t, \mathbf{x}(t), \mathbf{q}) \\ \mathbf{x}(t_0) &= \mathbf{x}_0 \end{aligned} \tag{1.1}$$

with observation process (we assume without loss of generality a scalar observation process)

$$f(t; \boldsymbol{\theta}) = C\mathbf{x}(t; \boldsymbol{\theta}),$$

where $\boldsymbol{\theta} = [\mathbf{q}^{\mathsf{T}}, \mathbf{x_0}^{\mathsf{T}}]^{\mathsf{T}}$, and $\mathbf{q}$ is a vector of length p so that $\boldsymbol{\theta}$ is a vector of length $\kappa_{\boldsymbol{\theta}} = p+n$ (we could consider the case where all or part of the initial conditions $\mathbf{x}_0$ are known–for simplicity in notation we do not-see [13, 25] for such more general formulations). The inverse problem is to determine $\boldsymbol{\theta}$ using given observations over time. Using an ordinary least-squares method (OLS) for estimation (again, more general formulations can be found in [13, 25]), we wish to find

$$\hat{\boldsymbol{\theta}}_{\text{OLS}} = \arg\min_{\boldsymbol{\theta} \in \Omega_{\boldsymbol{\theta}}} \sum_{j=1}^{n} (y_j - f(t_j; \boldsymbol{\theta}))^2$$

where y_j is the data at time t_j and $\Omega_{\boldsymbol{\theta}}$ is the admissible set for the parameters.We assume that $\hat{\boldsymbol{\theta}}_{\text{OLS}}$ follows a normal distribution with mean $\boldsymbol{\theta}_0$ and covariance matrix $\Sigma_0 \approx \sigma_0^2[\chi^T\chi]^{-1}$ where

$$\chi_{jk} = \frac{\partial f(t_j; \boldsymbol{\theta})}{\partial \theta_k} \tag{1.2}$$

The matrix χ is called the *sensitivity matrix.* Our goal is to compute $\frac{\partial f(t_j;\boldsymbol{\theta})}{\partial \theta_k}$ as efficiently and accurately as possible. There are various ways of approximating these derivatives. The most common ones (see [18, 13, 25] and the references therein) are analytic methods, finite difference approximation, automatic differentiation and the use of sensitivity equations. Analytic methods are accurate but they require the derivation and development of a program that is specific to each problem, hence inefficient. The finite difference method, on the other hand, is relatively easy and efficient to implement but it suffers from cancellation error. Sensitivity equations are known to be accurate and computationally inexpensive for reasonably small systems.

More recently, a method referred to as *the complex-step* has been used to calculate sensitivities ([55, 56]). The idea of using complex variables to estimate derivatives originated with the work of Lyness and Moler [52] and Lyness[51]. After its rediscovery by Squire and Trapp in 1998, it has become quite popular in aerodynamic optimization [1, 2, 55, 56, 65]. The complex-step estimate is suitable for use in numerical computing and shown to be very accurate, extremely robust while retaining a reasonable computational cost. Martins, et al., in [55] have shown the method to have implementation advantages over automatic differentiation and computational advantages over finite-differencing. Although a step size parameter is required, in most cases the numerical derivatives are not subject to subtractive cancellation errors as in the use of finite differences, (see [56] for exceptions and remedies). Therefore, the method exhibits true second-order accuracy as the step size is reduced. In addition, the procedure is easily implemented into existing programs. The only requirements are that the floating point variables be declared as complex and that a complex perturbation be added to the variable of interest.

In this report, we apply these techniques for various types of problems including time-delayed ODEs as well as boundary sensitivity for PDEs and compare the results with solutions of sensitivity equations. In the first and second sections of the report, we provide brief summaries of the complex-step method and the method of sensitivity equations. We provide various numerical examples in the later section, and we conclude with our findings and comments in the last section.

2 The complex-step method

In this section, following [55], we summarize the complex-step method. Given (1.1), let $z = x + iy,\ x,\ y \in \mathbb{R}$ be a complex number and $f(z) = f(x, y) = u(x, y) + iv(x, y)$ be a function of a complex variable. If f is analytic, we have the following Cauchy-Riemann equations that establish the relationship between the real and imaginary parts of the function.

$$\frac{\partial u}{\partial x}(x,y) = \frac{\partial v}{\partial y}(x,y), \qquad \frac{\partial u}{\partial y}(x,y) = -\frac{\partial v}{\partial x}(x,y) \tag{2.1}$$

For a given step size h we can derive a finite difference-like first derivative estimate for real functions using complex calculus. From the first equation in (2.1), and the definition of derivatives we have

$$\frac{\partial u}{\partial x}(x,y) = \lim_{h\to 0} \frac{v(x,y+h)-v(x,y)}{h} \tag{2.2}$$

$$= \lim_{h\to 0} \frac{\mathrm{Im}[f(x+i(y+h))]-\mathrm{Im}[f(x+iy)]}{h}. \tag{2.3}$$

If f takes a real-valued input (as in the case where, for example, f represents the solution to a differential equation model), then $y=0$, $f(x)=u(x,0)$, and $v(x,0)=\mathrm{Im}[f(x)]=0$. Thus (2.2) becomes

$$\frac{\partial f}{\partial x} = \lim_{h\to 0} \frac{\mathrm{Im}[f(x+ih)]}{h}. \tag{2.4}$$

Therefore, for small h, we have *the complex-step* derivative approximation

$$\frac{\partial f}{\partial x} \approx \frac{\mathrm{Im}[f(x+ih)]}{h}. \tag{2.5}$$

This formula can also be obtained by approximating an analytic function f with a complex variable using a Taylor series expansion:

$$f(x+ih) \approx f(x) + ihf'(x) - \frac{h^2}{2!}f''(x) - i\frac{h^3}{3!}f^{(3)}(x) + \frac{h^4}{4!}f^{(4)}(x) + \cdots$$

Taking the imaginary parts of both sides and dividing by h gives

$$f'(x) \approx \frac{\mathrm{Im}[f(x+ih)]}{h} + O(h^2).$$

Terms of order h^2 and higher can be ignored because the step size h can be chosen up to machine precision. Thus the complex-step derivative is given by (2.5) with a truncation error $E_t(h) = \frac{h^2}{6}f^{(3)}(x)$. The method is accurate down to a specific step size we call h_{crit}. Below h_{crit}, underflow occurs and the approximation becomes useless. It is important to note that all our computations were done on machines with the limits of machine precision 10^{-324}.

The derivative estimate (2.5) constitutes a big advantage over the finite-difference approach. This is because the finite-difference approximation is subject to subtractive error due to the differencing operation. On the other hand, the accuracy of the complex-step estimates is only limited by the numerical precision of the algorithm that evaluates the function f.

Note that the complex-step approximation formula (2.5) is derived based on the Cauchy-Riemann formula for analytic functions. Therefore, it is important that the function f be analytic. In the case when the functions have singularities or branch cuts where they are not analytic, the complex-step method provides a correct derivative approximation up to the point of discontinuity. In addition, we get accurate approximations of one-side derivatives if we uniquely define the function at that point. The method also provides an accurate first order derivative when a function has jumps in its higher order derivatives (see examples 4.2-4.4).

2.1 Implementation

2.1.1 Complex function definitions

The complex-step approximation formula (2.5) is derived based on the Cauchy-Riemann formula assuming the function f is analytic. Then it is important to investigate to what extent this assumption holds when the value of the function is calculated by a numerical algorithm. In addition we need to extend functions of real variables to complex in such a way that the Cauchy-Riemann equations are satisfied. Even though our discussion below is for implementation of the complex-step in Matlab, the principles are the same for other programming languages. For further discussion and Fortran implementation, we refer the reader to [55] and [56] .

When we convert a 'real' algorithm to complex, we are mainly concerned with two types of operations:

1. Relational operators
   Relational logic operators like "greater than" and "less than" are defined in Matlab to compare only the real parts of a complex number. These operators are usually used in '*if*' statements to redirect the execution thread. The original algorithm and its complex version must follow the same execution thread. Therefore, the Matlab definition of these operators is the correct one.

2. Arithmetic functions and operators
   In Matlab, complex numbers are a standard data type and many functions have complex counterparts. Functions that choose one argument like *max* and *min* are based on relational operators. Therefore, one would assume that they are defined based on their real parts in Matlab. Unfortunately, that is not the case. In Matlab, *min* and *max* functions compare the radii of two complex numbers. Hence, they need to be redefined to compare only the real parts.

   Another function that we need to give attention to is the absolute value (*abs*) function. *abs* in Matlab returns the absolute value or modulus of a complex number. Since this definition was not derived assuming the Cauchy-Riemann equations, the complex-step method does not give a correct derivative approximation. Therefore, we need to define '*abs*' so that it satisfies the Cauchy-Riemann equations. Since we know what the value of the derivative must be, we can write

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} = \begin{cases} -1, & x < 0 \\ +1, & x > 0 \end{cases} \tag{2.6}$$

   From (2.1), since $\partial v/\partial x = 0$ on the real axis, we get $\partial u/\partial y = 0$ on the same axis, so the real part of the result must be independent of the imaginary part of the variable. Therefore, the new sign of the imaginary part depends only on the sign of the real part of the complex number, and an analytic absolute value function can be defined as:

$$abs(x + iy) = \begin{cases} -x - iy & x < 0 \\ x + iy & x \geq 0 \end{cases} . \tag{2.7}$$

This function is not analytic at $x = 0$. But, as was mentioned earlier, the complex-step approximation yields an accurate derivative up to the discontinuity. In addition, since we have defined the function at $x = 0$ as for $x > 0$, the method gives the correct right-hand side derivative at $x = 0$.

Similarly, for every real valued function, by requiring that its complex extension satisfies the Cauchy-Riemann equations, and that the real and the complex have the same properties, we can obtain a unique complex function definition either by adjusting the definition of custom functions, or ensuring that Matlab's complex analog function is correctly defined.

Many of the arithmetic operators including addition, multiplication, and trigonometric function have standard complex definitions that are analytic almost everywhere.

The transpose ($'$) operator in Matlab gives the complex conjugate of the matrix. So we need to use the non-conjugate transpose ($\cdot'$) instead.

We remark that before implementing the complex-step method, one always needs to check whether the functions and operators in the algorithm need re-definition and do so accordingly.

2.1.2 Implementation Procedure

Given an analytic function f, the following is an outline of the general steps for implementing the complex-step method for computing the first derivative, df/dx.

1. Define all functions and operators that are not defined for complex arguments such as for example *max, min* and *abs.*
2. Add a small complex step ih to the desired variable 'x', run the algorithm that evaluates f.
3. Compute df/dx using (2.5).

2.2 Generalization for vector-valued functions

Given a step size h, and the vector valued function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ which takes the vector $\mathbf{x} \in \mathbb{R}^n$ and produces the vector $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^m$, the complex-step approximation of the Jacobian matrix is given by [50]

$$J \approx \frac{1}{h}\text{Im} \begin{bmatrix} f_1(\mathbf{x}+ih\mathbf{e}_1) & \cdots & f_1(\mathbf{x}+ih\mathbf{e}_j) & \cdots & f_1(\mathbf{x}+ih\mathbf{e}_n) \\ \vdots & & \vdots & & \vdots \\ f_p(\mathbf{x}+ih\mathbf{e}_1) & \cdots & f_p(\mathbf{x}+ih\mathbf{e}_j) & \cdots & f_p(\mathbf{x}+ih\mathbf{e}_n) \\ \vdots & & \vdots & & \vdots \\ f_m(\mathbf{x}+ih\mathbf{e}_1) & \cdots & f_m(\mathbf{x}+ih\mathbf{e}_j) & \cdots & f_m(\mathbf{x}+ih\mathbf{e}_n) \end{bmatrix} \tag{2.8}$$

where f_p is the $p-$th component of $\mathbf{f}$ and $\mathbf{e}_j$ is the $j-$th unit vector in $\mathbb{R}^n$.

3 Sensitivity Equations

In this section, we briefly summarize the derivation of sensitivity equations for the ODE model (1.1) which can be extended for PDEs in a similar way. For extensive discussion of the method, see [13] and [18] and references therein.

Given (1.1), first we derive sensitivity equations for the parameter $\mathbf{q}$. Let $\mathbf{s} = (\mathbf{s}_{q_1}, \cdots, \mathbf{s}_{q_k})$, where

$$\mathbf{s}_{q_k}(t) = \frac{\partial \mathbf{x}}{\partial q_k}(t; \boldsymbol{\theta}), \quad k = 1, \cdots, p. \tag{3.1}$$

Taking the derivative of equation (1.1) with respect to $\mathbf{q}$, we obtain

$$\begin{aligned} \frac{d\mathbf{s}(t)}{dt} &= \frac{\partial \mathbf{g}}{\partial \mathbf{x}}\mathbf{s}(t) + \frac{\partial \mathbf{g}}{\partial \mathbf{q}} \\ \mathbf{s}(t_0) &= \mathbf{0}_{n\times p}. \end{aligned} \tag{3.2}$$

For the initial condition $\mathbf{x}_0$, let $\mathbf{r} = (\mathbf{r}_{x_{0_1}}, \cdots, \mathbf{r}_{x_{0_n}})$ where

$$\mathbf{r}_{x_{0_j}}(t) = \frac{\partial \mathbf{x}}{\partial x_{0_j}}(t; \boldsymbol{\theta}), \quad j = 1, \cdots, n. \tag{3.3}$$

Then we obtain the system

$$\begin{aligned} \frac{d\mathbf{r}(t)}{dt} &= \frac{\partial \mathbf{g}}{d\mathbf{x}}\mathbf{r}(t) + \frac{\partial \mathbf{g}}{\partial \mathbf{x}_0} \\ &= \frac{\partial \mathbf{g}}{\partial \mathbf{x}}\mathbf{r}(t), \\ \mathbf{r}(t_0) &= \mathbf{I}_{n\times n}. \end{aligned} \tag{3.4}$$

To find sensitivities, we solve equations (3.2) and (3.4) together with (1.1). Namely, we solve

$$\begin{aligned} \frac{d\mathbf{x}(t)}{dt} &= \mathbf{g}(t, \mathbf{x}(t), \mathbf{q}), \\ \frac{d\mathbf{s}(t)}{dt} &= \frac{\partial \mathbf{g}}{\partial \mathbf{x}}\mathbf{s}(t) + \frac{\partial \mathbf{g}}{\partial \mathbf{q}}, \\ \frac{d\mathbf{r}(t)}{dt} &= \frac{\partial \mathbf{g}}{\partial \mathbf{x}}\mathbf{r}(t), \\ \mathbf{x}(t_0) &= \mathbf{x}_0, \\ \mathbf{s}(t_0) &= \mathbf{0}_{n\times p}, \\ \mathbf{r}(t_0) &= \mathbf{I}_{n\times n}. \end{aligned}$$

4 Numerical Examples

In this section, we apply the complex-step method to compute sensitivities for various kinds of examples and we compare the results with solutions of sensitivity equations (or analytic solutions when available). The first example is the logistic model for population growth which is a simple enough ODE model yet a widely applied example in biological applications [48, 60]. This simple example can be used to calibrate our computational times.

We then turn to two classes of examples where the underlying foundations for the complex step method do not hold. The first of these is *delay differential* (also called *functional differential*) *equations* where sensitivity with respect to the delays or hysteresis kernels do not in general satisfy the analyticity requirements for use of the Cauchy-Riemann equations. Delay equations have been used in a wide variety of biological applications as well as in many engineering problems –see for examples the references [3, 4, 7, 8, 27, 28, 29, 30, 33, 36, 38, 39, 40, 41, 43, 45, 49, 53, 54, 61]. Very early interest in the 1940's focused around the studies of mechanical systems by Minorsky [57, 58, 59] and slightly later those of Hutchinson [46, 47] in biology. Both of these authors argued that time delays in dynamical systems can produce oscillatory phenomena in an otherwise non-oscillatory system.

In 1948 Hutchinson [46] developed a delay differential equation model, subsequently known as *Hutchinson's equation*, which is the *delayed logistic equation*, to describe the dynamics of a circular causal system. An example of an ecological circular causal system is a parasite-host interaction where a parasite completes its life cycle without killing the host or drastically altering the growth of the host population. The host population can then continue to exist [46, 49]. The delay in this model can represent various naturally occurring phenomena such as the gestation period in a growing population, the life cycle of a parasite, cell cycle delays, etc. Hutchinson's equation (to be used along with Minorsky's harmonic oscillator with a delayed damping in the numerical illustrations below), its variations and other delay systems have also been used to model physiological control systems as well as numerous other biological processes.

Delay differential equations (DDEs) are particularly interesting because the derivatives of their solutions often have discontinuities (see [24] for a theoretical treatment and discussions). This is generally true because the first derivative of a non-constant history function at zero is almost always different from the right derivative of the solution at the initial point. As we shall see below, in addition to the discontinuity at the initial point, discontinuities in derivatives of the initial function tend to be propagated with one degree of smoothness added per time delay interval. Since the complex-step method is derived assuming analyticity, one would expect for the method to fail when it comes to computing the sensitivity with respect to the time lag τ. But the results for these examples show that the complex-step method approximates the sensitivities accurately up to h_{crit} even in the presence of discontinuities in the solutions.

The second class of examples we investigate involves sensitivities of boundary control problems governed by partial differential equations (PDEs) where one is interested in sensitivities of the solutions with respect to the boundary terms (the controls) themselves. In particular we are motivated by problems of control of [32, 37, 42] poroelastic models for the perfusion of the lamina cribrosa in the optic nerve head. Retinal hemodynamics plays a crucial role in the pathophysiology of several ocular diseases, including glaucoma, age-related macula degeneration and diabetic retinopathy. The retinal vascular bed nourishes the retinal ganglion cells that are responsible for the transmission of visual information from the retina to the brain, via the optic nerve. Blood is supplied by the central retinal artery (CRA), and drained by the central retinal vein. The central retinal vessels, i.e., central retinal artery and vein, run through the optic nerve canal, where they are exposed to the retrolaminar tissue pressure (RLTp). They also enter inside the eye globe, where they are exposed to the intraocular pressure (IOP). The pressure difference between the RLTp in the optic nerve tissue (baseline value 7-10 mmHg) and the IOP inside the eye globe (baseline value 12-15 mmHg) is maintained by the lamina cribrosa, a collagen structure that is pierced by the central retinal vessels approximately in its center. There are clear evidences that the CRA hemodynamics is strongly affected by the level of IOP inside the eye globe. Control of the IOP leads to a natural optimization problem involving boundary control of a PDE system (a linear poroelastic system). Again in general one does not expect analyticity of the Dirichlet or Neumann BC map to the solution. Here we investigate performance of the complex-step method in computing the relevant sensitivities. Again, in comparing the sensitivity equations results to those using the complex-step method, we find good agreement up to some h_{crit} which suggests that the complex-step method performs well even when the model at hand is more sophisticated and lacks the appropriate analyticity requirements of the boundary-to-solution maps.

4.1 The logistic model for population dynamics

In traditional population growth model, as developed by Malthus (1798), the growth rate is proportional to the size of population:

$$\frac{dx(t)}{dt} = rx(t)$$

where r is the intrinsic growth rate of the population and $x(t)$ represents the population size at time t. As a result, the population grows exponentially:

$$x(t) = x_0 e^{rt},$$

where $x_0 = x(0)$ is the initial population. In reality, this model may only be valid for a short time period because the environment imposes limitations to population growth. A more accurate model by Verhulst (1836) postulates that the relative growth rate approaches the *carrying capacity* K of the environment. The corresponding equation is called the *logistic differential equation*:

$$\frac{dx(t)}{dt} = rx(t)\left(1 - \frac{x(t)}{K}\right),\ 0 < t \leq T \tag{4.1}$$

$$x(0) = x_0, \tag{4.2}$$

with analytic solution

$$x(t) = \frac{K}{1 + \left(\frac{K}{x_0} - 1\right)e^{-rt}}.$$

We compute the sensitivities of the solution x with respect to the parameters $r,\ K,\ x_0$ using both the complex-step method and sensitivity equations.

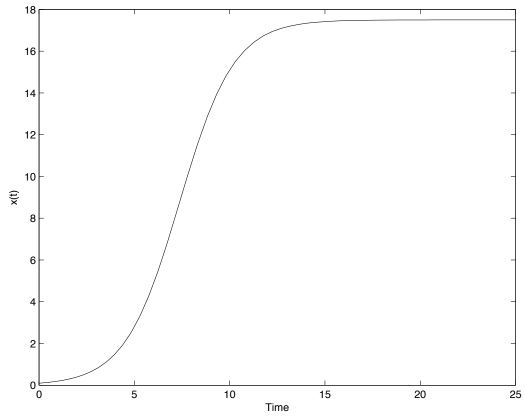


Figure 1: Solution function of the logistic equation at $r = 0.7,\ K = 17.5, x_0 = 0.1$.

To derive the sensitivity equations, we take the derivative of the equation with respect to each parameter. The sensitivity equations are, thus, given by

$$\begin{aligned}
\frac{dx(t)}{dt} &= rx(t)\left(1-\frac{x(t)}{K}\right),\\
\frac{ds_r(t)}{dt} &= \left(r-\frac{2r}{K}x(t)\right)s_r(t)+x(t)-\frac{1}{K}x^2(t),\\
\frac{ds_K(t)}{dt} &= \left(r-\frac{2r}{K}x(t)\right)s_K(t)+\frac{r}{K^2}x^2(t),\\
\frac{ds_{x_0}}{dt} &= \left(r-\frac{2r}{K}x(t)\right)s_{x_0}(t),
\end{aligned}$$

with initial conditions

$$x(0) = x_0,\ s_r(0) = s_K(0) = 0,\ s_{x_0}(0) = 1.$$

where $s_K(t) = \frac{\partial x(t)}{\partial K},\ s_r(t) = \frac{\partial x(t)}{\partial r},\ s_{x_0}(t) = \frac{\partial x(t)}{\partial x_0}$.

To compute the sensitivities using the complex-step method, we add ih to each parameter separately and solve the original ODE system. Then use the formula (2.5) to compute the derivatives. In this case, we need run the program solving the ODE system three times because we have three parameters.

In Figure 2 below, we compare the sensitivity functions computed by solving the above sensitivity equations and the complex-step method. As it is shown, the accuracy of the complex-step approximation starts to decline at $h_{crit} = 10^{-320}$. We have displayed the sensitivities at $h = 10^{-40}$ and at $h = 10^{-321}$ side by side. (Recall our machine accuracy is 10^{-324}).

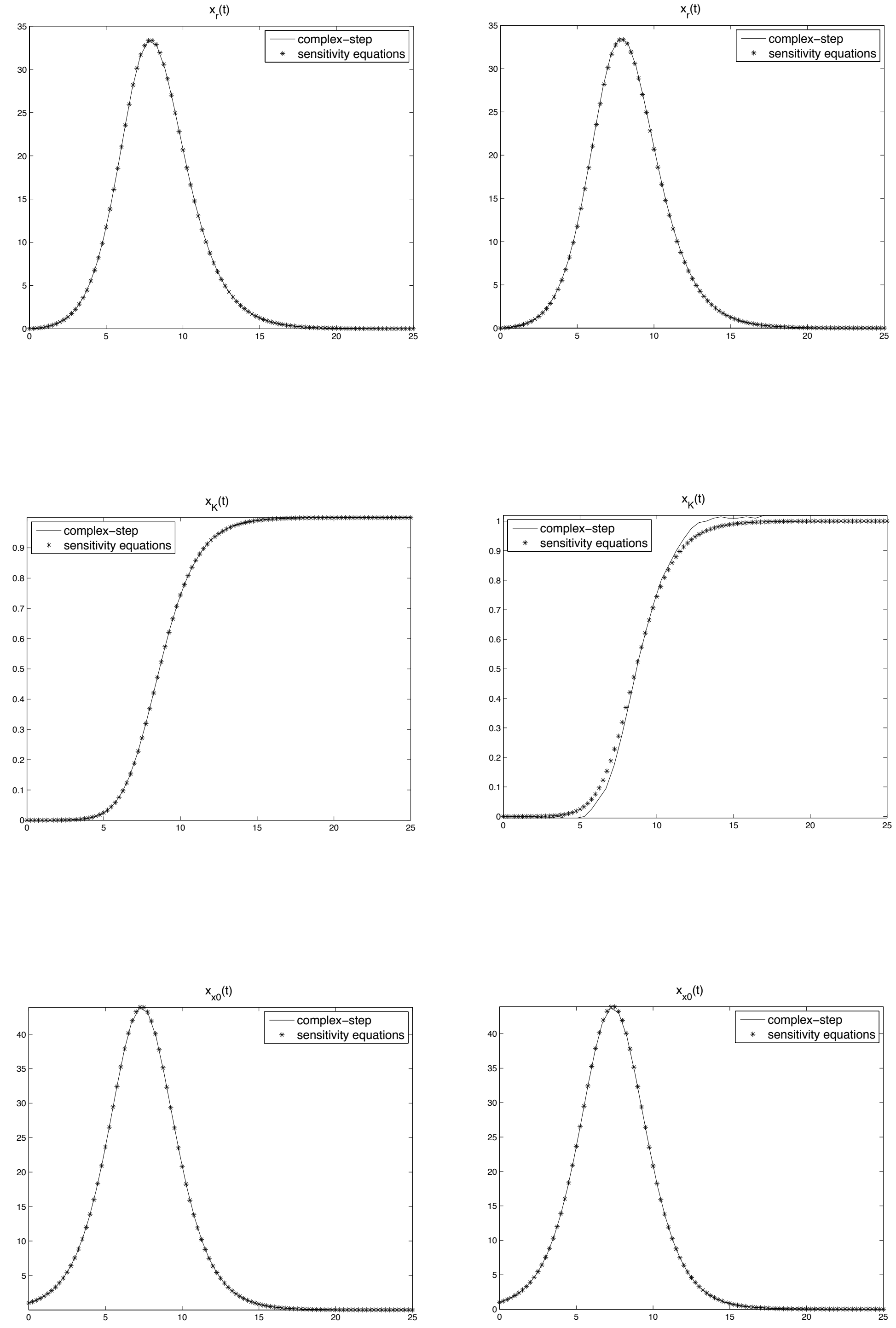


Figure 2: Comparison of sensitivity functions for the logistic equation with respect to growth rate r, carrying capacity K, constant initial state x_0, each evaluated at $(r, K, x_0) = (0.7, 17.5, 0.1)$. The step size for the complex-step approximation is taken to be $h = 10^{-40}$ (left) and $h = 10^{-321}$ (right).

4.2 Hutchinson equation

In the previous logistic model it is assumed that the growth rate of a population at any time t depends on the relative number of individuals at that time. In practice, the process of reproduction is not instantaneous. For example, in a Daphnia population, a large clutch presumably is determined not by the concentration of unconsumed food available when the eggs hatch, but by the amount of food available when the eggs were forming, some time before they pass into the brood pouch. Between the determination and the time of hatching many newly hatched Daphnias may have been liberated from the brood pouches of other Daphnia in the culture, which increases the population. Hutchinson (1948) assumed egg formation to occur τ units of time before hatching and proposed the following more realistic delayed logistic equation

$$\frac{dx(t)}{dt} = rx(t)\left(1 - \frac{x(t-\tau)}{K}\right) \tag{4.3}$$

where r and K have the same meaning as the logistic equation (4.1) and $\tau > 0$ is a constant. This equation is referred to as *Hutchinson's equation* or the *delayed logistic equation*. This is a delayed differential equation which is an infinite dimensional equation which is properly posed in a function space requiring an initial function on $-\tau \leq t \leq 0$ to generate a forward solution [24]. Several solutions are depicted in Figure 3.

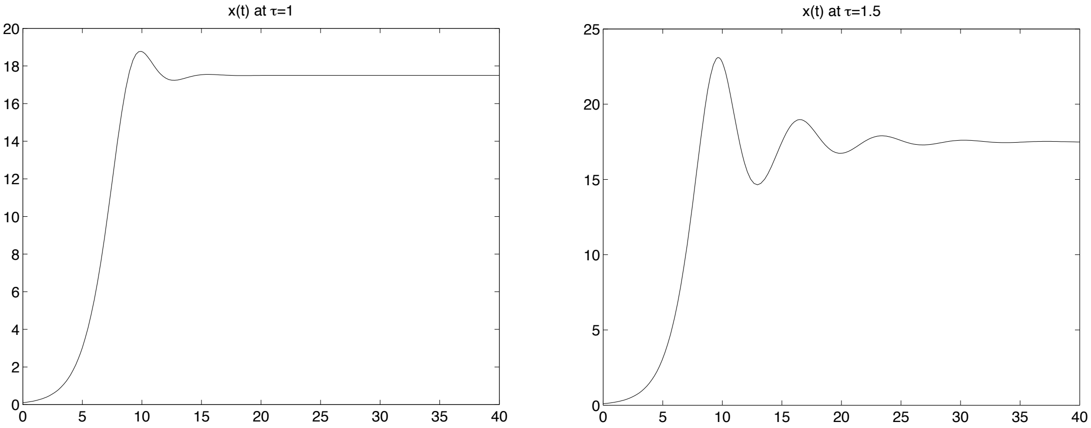


Figure 3: Solution functions of the Hutchinson equation with a small delay $\tau = 1$ (left) and a moderate delay $\tau = 1.5$. The initial condition $z_0 = (x_0, x(\theta)) = (0.1, 0.1)$ and parameter values are $r = 0.7$ and $K = 17.5$.

There are several options for a "state space" to develop theoretical foundations for existence, uniqueness, differentiability with repect to parameters, etc. One obvious choice [44] for an n-dimensional vector delay differential equation is $\mathrm{C}(-\tau, 0; \mathbb{R}^n)$, however, the more common choice for computational efforts is a state space that treat the current state in a Euclidean sense while treating the history part in an L_2 sense. That is, $Z = \mathbb{R}^n \times L_2(-\tau, 0; \mathbb{R}^n)$, with elements $z(t) = (x(t), x(t+\theta))$ where $-\tau \leq \theta \leq 0$. Thus to generate forward solutions for $t > 0$ we must specify $x(0)$ in the Euclidean sense and $x(\theta)$, $-\tau \leq \theta < 0$ in the L_2 sense. Note that the initial function need not be continuous at $t = 0$, and even if it is left continuous at $t = 0$, the solution itself need not be continuously differentiable, let alone analytic at $t = 0$. Moreover, non-analyticity in initial data can be propagated in solutions in time.

The sensitivity equations are derived in similar fashion as in the previous example and have the form (in general the sensitivity equations are also DDEs)

$$\frac{dx(t)}{dt} = rx(t)(1 - \frac{x(t-\tau)}{K}) \tag{4.4}$$

$$\frac{ds_r(t)}{dt} = r\left[1 - \frac{x(t-\tau)}{K}\right] s_r(t) - r\frac{x(t)}{K} s_r(t-\tau) + x(t)\left[1 - \frac{x(t-\tau)}{K}\right] \tag{4.5}$$

$$\frac{ds_K(t)}{dt} = r\left[1 - \frac{x(t-\tau)}{K}\right] s_K(t) - r\frac{x(t)}{K} s_K(t-\tau) + rx(t)\left[\frac{x(t-\tau)}{K^2}\right] \tag{4.6}$$

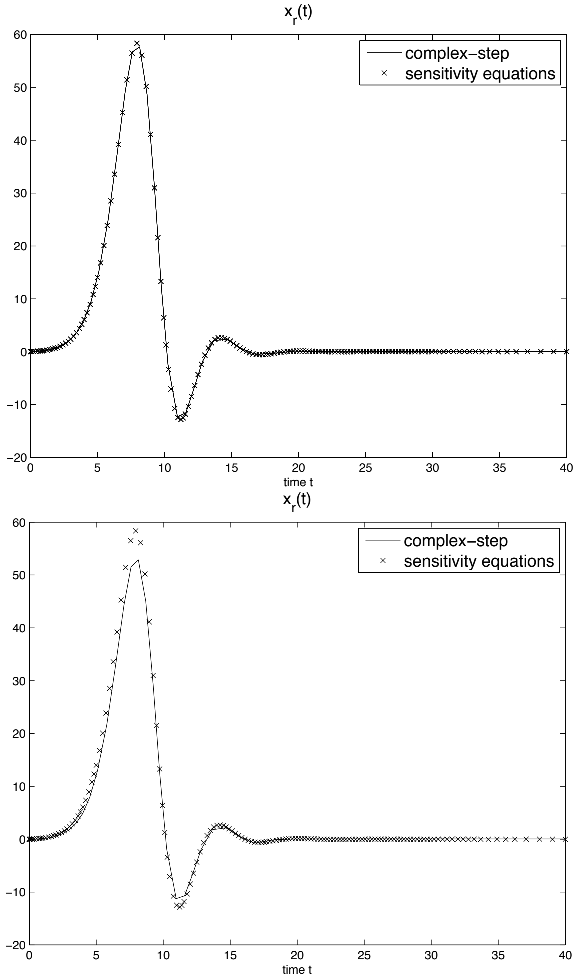
$$\frac{ds_{x_0}(t)}{dt} = r\left[1 - \frac{x(t-\tau)}{K}\right] s_{x_0}(t) - r\frac{x(t)}{K} s_r(t-\tau) \tag{4.7}$$

$$\frac{ds_\tau(t)}{dt} = r\left[1 - \frac{x(t-\tau)}{K}\right] s_\tau(t) - r\frac{x(t)}{K}[s_\tau(t-\tau) - \dot{x}(t-\tau)] \tag{4.8}$$

with initial conditions

$$x(\theta) = x_0,\ s_r(\theta) = s_K(\theta) = 0, -\tau \leq \theta \leq 0,\ s_{x_0}(0) = 1.$$

Sensitivities using the two methods are compared below in Figure 4 and Figure 5. For this example, $h_{crit} = 10^{-321}$. We observe a slight decline right below this value as shown in the figure and underflow occurs for smaller values. As mentioned in the beginning of the this section, the Hutchinson equation solution is not analytic with respect to τ. But the result in Figure 4 and Figure 5 reveal that the complex-step method gives a very good approximation of the derivatives with respect to τ up to $h_{crit} = 10^{-321}$.


$x_r(t)$
complex−step
sensitivity equations
60
50
40
30
20
10
0
−10
−20
0
5
10
15
20
25
30
35
40
time t
$x_r(t)$
complex−step
sensitivity equations
60
50
40
30
20
10
0
−10
−20
0
5
10
15
20
25
30
35
40
time t

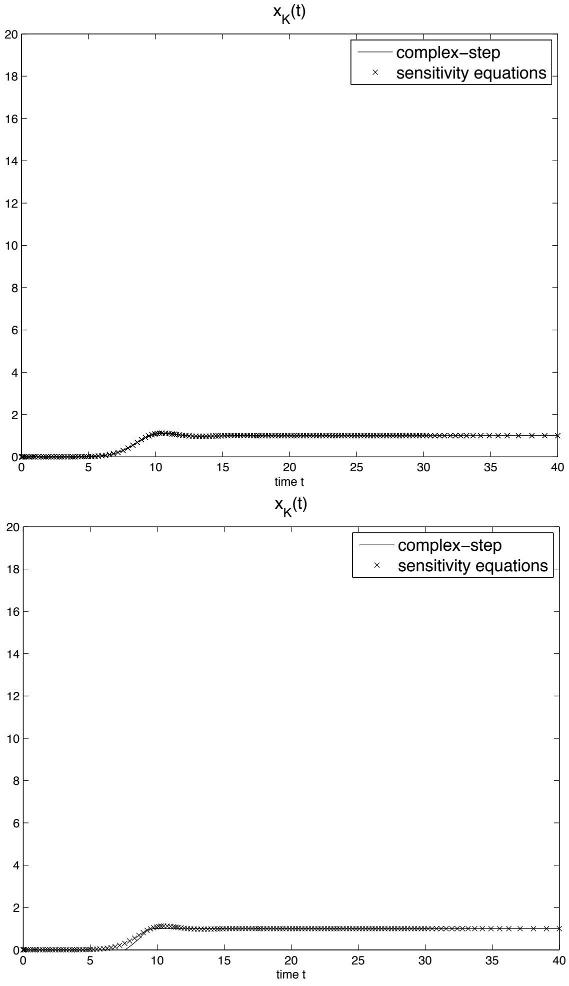
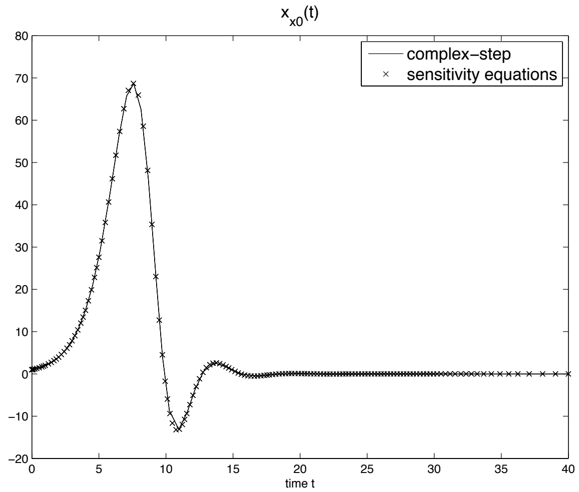
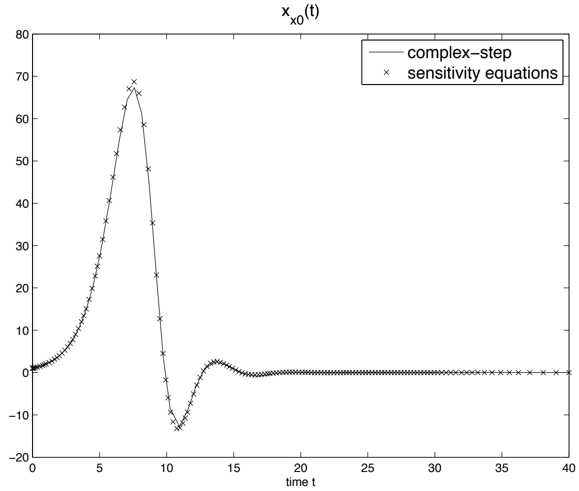


Figure 4: Comparison of sensitivity functions for the Hutchinson equation with respect to growth rate r, and carrying capacity K, each evaluated at $(r, K, x_0, \tau) = (0.7, 17.5, 0.1, 1)$. The step size for the complex-step approximation is taken to be $h = 10^{-40}$ (left) and $h = 10^{-322}$ (right).


$x_{x0}(t)$
complex−step
sensitivity equations
80
70
60
50
40
30
20
10
0
−10
−20
0
5
10
15
20
25
30
35
40
time t


$x_{x0}(t)$
complex−step
sensitivity equations
80
70
60
50
40
30
20
10
0
−10
−20
0
5
10
15
20
25
30
35
40
time t

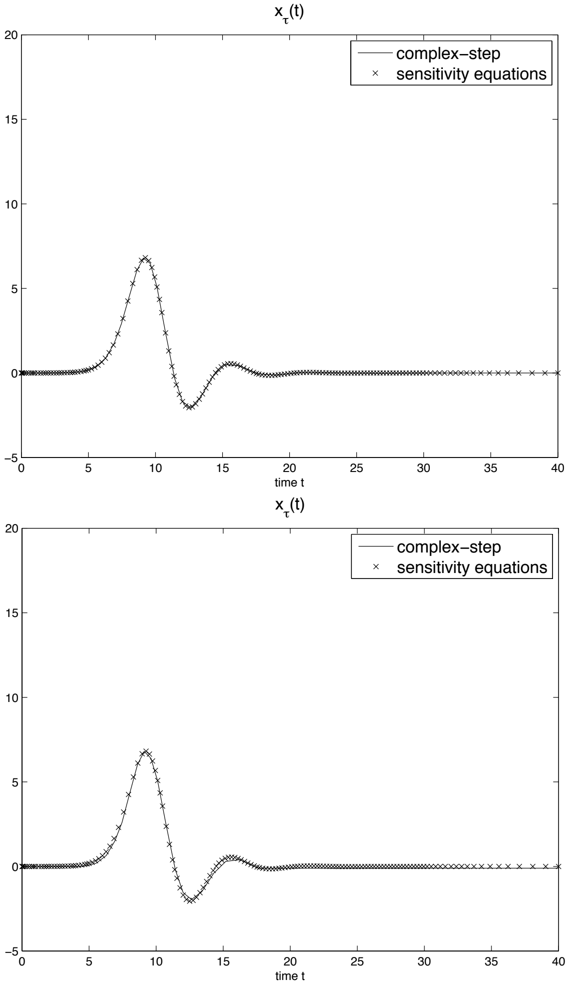


Figure 5: Comparison of sensitivity functions for the Hutchinson equation with respect to constant initial state $x(\theta) = x_0 \ \ -\tau \leq \theta \leq 0$, and delay τ, each evaluated at $(r, K, x_0, \tau) = (0.7, 17.5, 0.1, 1)$. The step size for the complex-step approximation is taken to be $h = 10^{-40}$ (left) and $h = 10^{-322}$ (right).

4.3 Harmonic oscillator with delayed damping

As a third example we present the Minorsky harmonic oscillator with a delayed damping. This example arises in many physical applications where oscillatory phenomena are important and delayed damping is relevant. The equation has the form

$$\frac{d^2x(t)}{dt^2} + K\frac{dx(t-\tau)}{dt} + bx(t) = g(t) \tag{4.9}$$

where K and b are spring and damping constants. We use the complex-step method and sensitivity equations to determine regions of sensitivity for model parameters K, b and the time delay τ. To derive the sensitivity equations, we take the derivative of the equation with respect to each parameter. First let $x_1 = x(t)$ and $x_2 = \dot{x}(t)$ and rewrite equation (4.9) as a first order system

$$\frac{dx_1(t)}{dt} = x_1(t) \tag{4.10}$$

$$\frac{dx_2(t)}{dt} = g(t) - bx_1(t) - Kx_2(t-\tau). \tag{4.11}$$

Letting $s_1(t) = \frac{\partial x_1(t)}{\partial K}$, $s_2(t) = \frac{\partial x_1(t)}{\partial b}$, $s_3(t) = \frac{\partial x_1(t)}{\partial \tau}$, $s_4(t) = \frac{\partial x_2(t)}{\partial K}$, $s_5(t) = \frac{\partial x_2(t)}{\partial b}$ and $s_6(t) = \frac{\partial x_2(t)}{\partial \tau}$, the sensitivity equations are given by

$$\frac{ds_1(t)}{dt} = s_4(t), \tag{4.12}$$

$$\frac{ds_2(t)}{dt} = s_5(t), \tag{4.13}$$

$$\frac{ds_3(t)}{dt} = s_6(t), \tag{4.14}$$

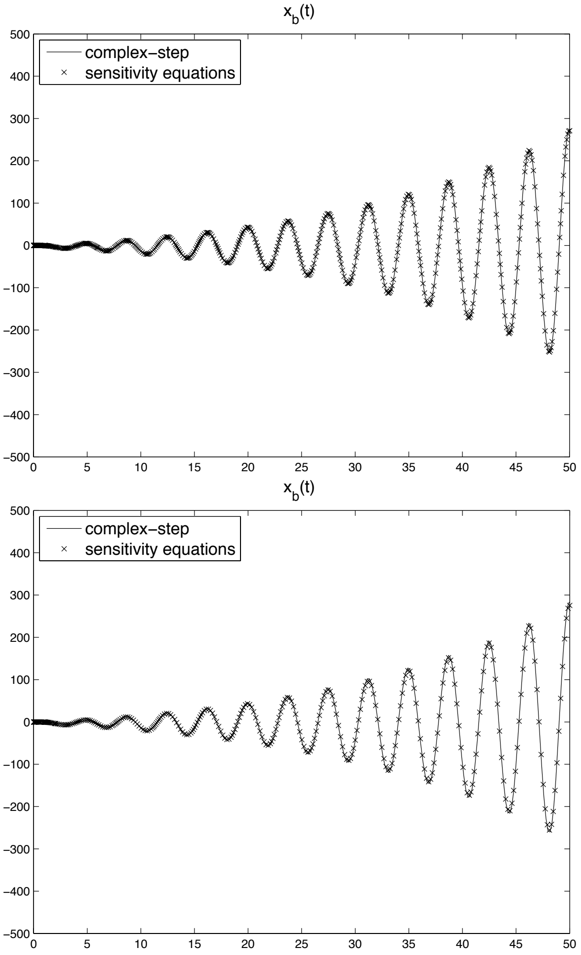
$$\frac{ds_4(t)}{dt} = -bs_1(t) - Ks_4(t-\tau) - x_2(t-\tau), \tag{4.15}$$

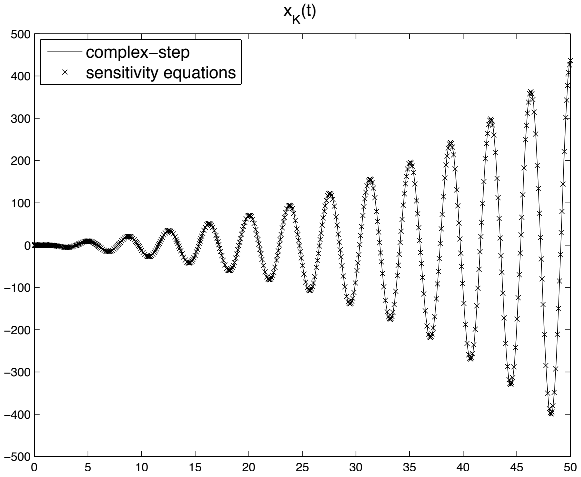
$$\frac{ds_5(t)}{dt} = -bs_2(t) - Ks_5(t-\tau) - x_1(t), \tag{4.16}$$

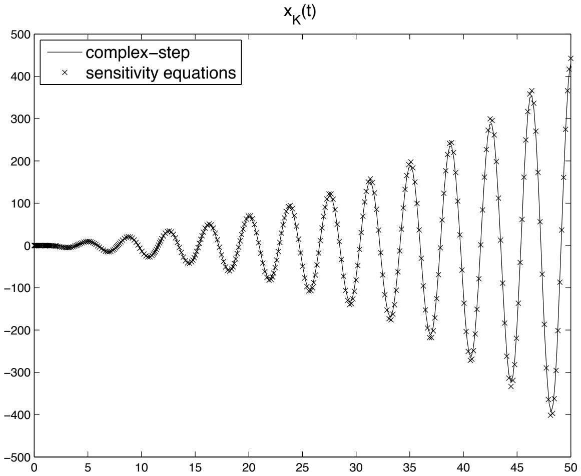
$$\frac{ds_6(t)}{dt} = -bs_3(t) - Ks_6(t-\tau) + K\dot{x}_2(t-\tau), \tag{4.17}$$

with initial conditions $x(\theta) = x_0, \ -\tau \leq \theta \leq 0, \ s_i(0) = 0, \quad i = 1,2,3,5, \ s_i(\theta) = 0, \quad i = 4,6.$

In Figure 6 the solutions of the sensitivity equations are shown for parameter values $K = 0.5, \ b = 2, \ g(t) = 10$, and delay $\tau = 1$, along with the sensitivities with respect to the parameters b, K and τ using the complex-step method. The results are identical for all the complex step size h up to the critical step size $h_{crit} = 10^{-322}$.


$x_b(t)$
complex−step
sensitivity equations
500
400
300
200
100
0
−100
−200
−300
−400
−500
0
5
10
15
20
25
30
35
40
45
50
$x_b(t)$
complex−step
sensitivity equations
500
400
300
200
100
0
−100
−200
−300
−400
−500
0
5
10
15
20
25
30
35
40
45
50


$x_K(t)$
complex−step
sensitivity equations
500
400
300
200
100
0
−100
−200
−300
−400
−500
0
5
10
15
20
25
30
35
40
45
50


$x_K(t)$
complex−step
sensitivity equations
500
400
300
200
100
0
−100
−200
−300
−400
−500
0
5
10
15
20
25
30
35
40
45
50

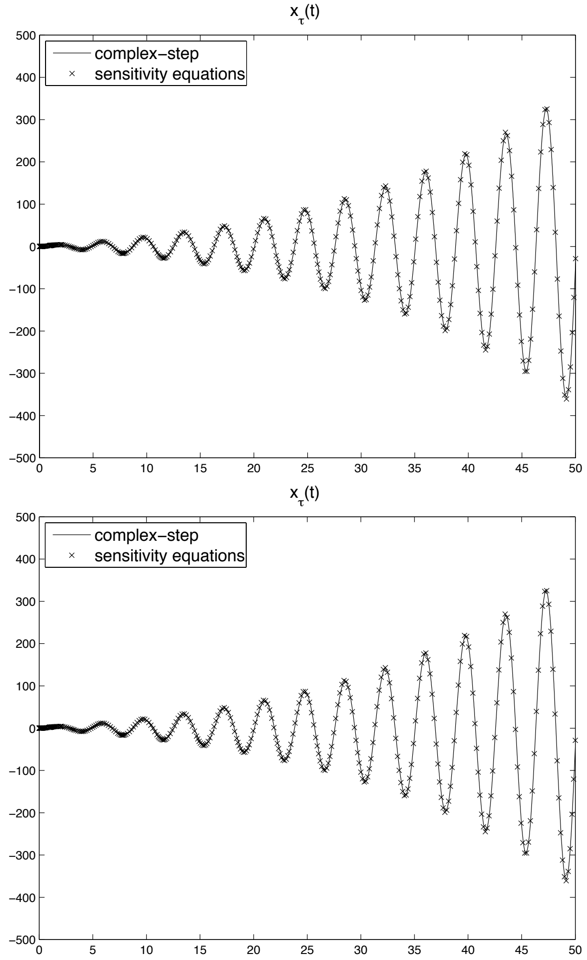


Figure 6: Comparison of sensitivity functions for the harmonic oscillator with delayed damping with respect to the restoring rate b, damping coefficient K, and delay τ, each evaluated at $(b, K, x_0, \tau) = (2, 0.5, 0.1, 1)$. The step size for the complex-step approximation is taken to be $h = 10^{-40}$ (left) and $h_{crit} = 10^{-322}$ (right).

4.4 Time delayed differential equations with non-smooth history functions

In this section we test the complex-step method for DDEs with non C^1 history function. Such systems arise in a wide class of applications, for example, when one has a discontinuous initial function such as $x(\theta) = 0$ for $-\tau \leq \theta < 0$, $x(0) = x_0 \neq 0$ [24]. We note that discontinuities in the history propagate to the solution in such a way that there is a jump in subsequent higher derivatives.

1. The first example is the Hutchinson equation (see Example 4.2) with a history function that has a jump in its first derivative at one point. We are interested primarily in the sensitivity with respect to the delay τ, since the solution is continuous in r and K.

$$\dot{x}(t) = rx(1 - \frac{x(t-\tau)}{K}), \quad 0 \leq t \leq 2\tau \tag{4.18}$$

$$x_0(t) = \phi(t) = \min(-1.8t + 0.1, 2t + 2), \quad -\tau \leq t \leq 0 \tag{4.19}$$

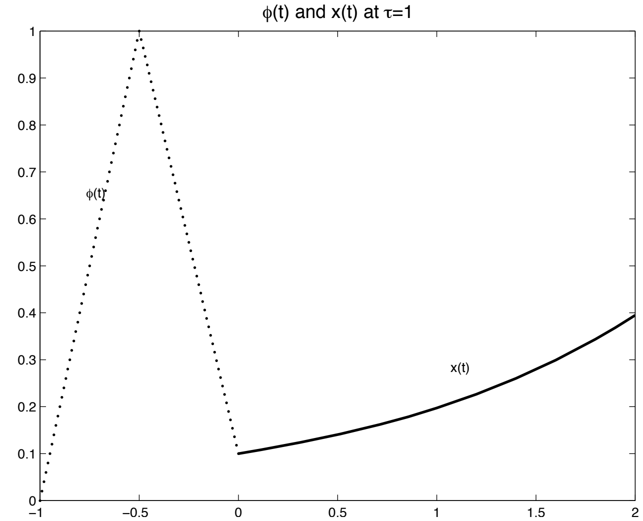


Figure 7: History and solution functions

Let $s_\tau = \frac{\partial x(t)}{\partial \tau}$, the sensitivity equation with respect to the delay is given by

$$\frac{ds_\tau(t)}{dt} = r\left[1 - \frac{x(t-\tau)}{K}\right] s_\tau(t) - r\frac{x(t)}{K}[s_\tau(t-\tau) - \dot{x}(t-\tau)] \tag{4.20}$$

$$s_\tau(\theta) = 0, \ -\tau \leq \theta \leq 0 \tag{4.21}$$

where

$$\dot{x}(t-\tau) = \begin{cases} 2 & 0 \leq t \leq \tau/2 \\ -1.8 & \tau/2 \leq t \leq \tau \end{cases} \tag{4.22}$$

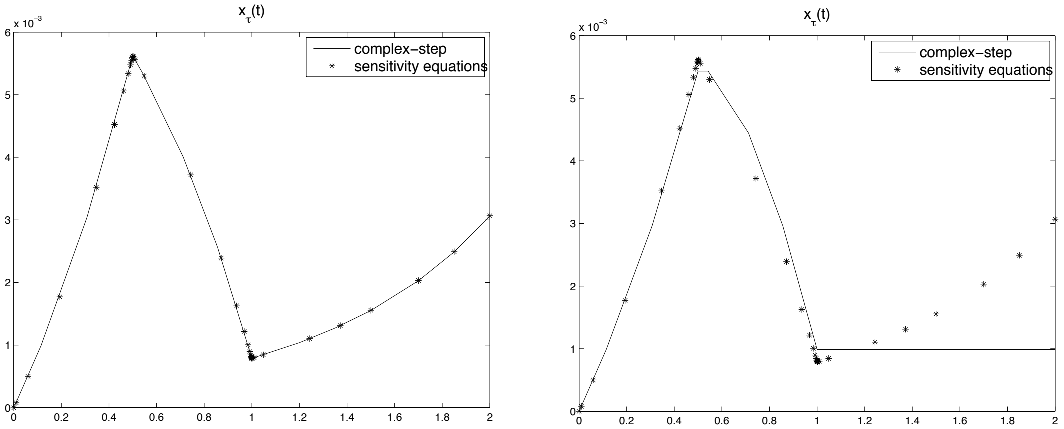


Figure 8: Complex-step approximation of $\frac{\partial x}{\partial \tau}$ overlaid with solution of the sensitivity equations for Hutchinson's equations for $t \in [0, 2\tau]$, $\tau = 1$. The step size for the complex-step is taken to be $h = 10^{-40}$ (left), $h = 10^{-320}$ (right, where h is taken too small).

2. Next we compute $\frac{\partial x}{\partial \tau}$ for the DDE:

$$\begin{aligned} \frac{dx}{dt} &= x(t-\tau), \qquad (4.23) \\ x_0(t) = \phi(t) &= \min\{2t+2, -2t\}, \; -\tau \le t \le 0. \end{aligned}$$

The analytic solution for this problem is given by

$$x(t;\tau) = \begin{cases} (t-\tau)^2 + 2t - \tau^2 & \text{for} \quad 0 \le t \le \tau/2, \\ \\ -(t-\tau)^2 - 1/2(\tau-1)^2 + 1/2 & \text{for } \tau/2 < t \le \tau. \end{cases} \qquad (4.24)$$

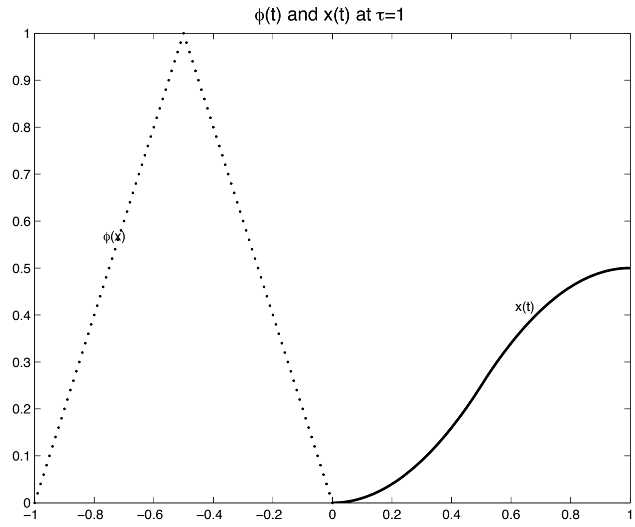


Figure 9: History and solution functions of equation (4.23) at $\tau = 1$.

The sensitivity equation with respect to τ is given by

$$\frac{ds_\tau}{\partial t} = -\dot{x}(t-\tau) + s_\tau(t-\tau),\ t > 0, \tag{4.25}$$

$$s_\tau(\theta) = 0,\ -\tau \le \theta \le 0. \tag{4.26}$$

where $s_\tau = \frac{\partial x(t)}{\partial \tau}$ and

$$\dot{x}(t-\tau) = \begin{cases} 2 & \text{for } 0 \le t \le \tau/2, \\ \\ -2 & \text{for } \tau/2 < t \le \tau. \end{cases} \tag{4.27}$$

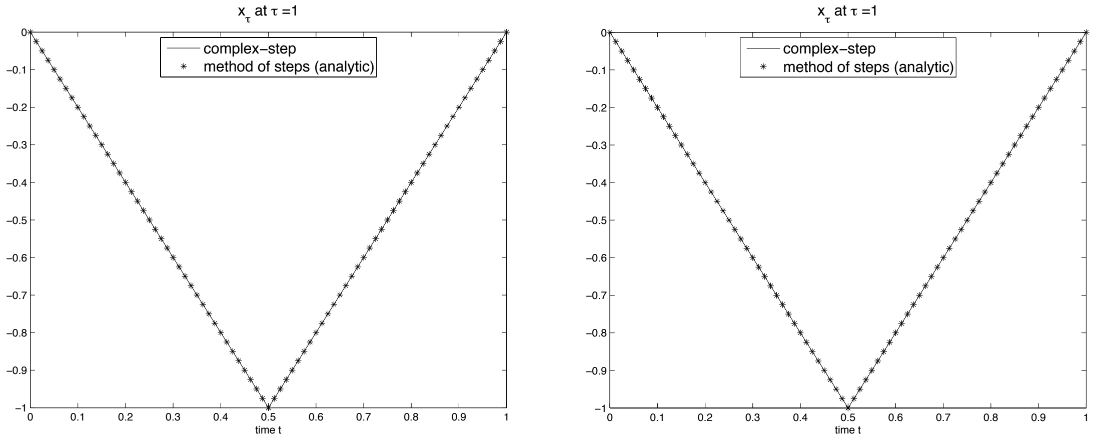


Figure 10: Sensitivity of the solution with respect to τ of equation (4.23) at $\tau = 1$ using the complex-step method. The step size is taken to be $h = 10^{-40}$ (Left) and $h_{crit} = 10^{-322}$.

Recall the complex-step formula (2.5) is derived using the fact that the function f is analytic. Once again these examples demonstrate that the complex-step method provides accurate approximation of the sensitivities when higher derivatives may have discontinuities. From this and other examples it is very clear that analyticity of the solution functions is *sufficient* but *NOT necessary* for the complex-step method to be effective.

4.5 Elasticity Equations

As a final example, we compute the sensitivities of the solution u of the linear elasticity model with respect to the parameters μ, λ and a constant boundary condition g. Given a domain $\Omega \subset \mathbb{R}^2$ with boundary $\partial\Omega = \Gamma_0 \cup \Gamma_1$, the linear elasticity equations are given by the system of PDEs:

$$-\text{div}(\sigma u) = f, \qquad \text{in } \Omega \tag{4.28}$$

$$u = 0, \qquad \text{on } \Gamma_0 \tag{4.29}$$

$$\sigma(u) \cdot \bar{n} = g, \qquad \text{on } \Gamma_1 \tag{4.30}$$

Here the stress tensor $\sigma(u)$ is given in terms of the strain $\epsilon(u)$ by the constitutive law:

$$\sigma(u) = \lambda \text{Tr}(\epsilon(u))I + 2\mu\epsilon(u),$$

where $\epsilon(u) = \frac{1}{2}(Du + (Du)^T)$, and λ and μ are the Lamé parameters.

In the following, we look at the cases when we have mixed Dirichlet-Neumann boundary and pure Dirichlet boundary conditions. We have documented a detailed explanation of the finite element method implementation for solving the linear elasticity equations in [16]. To compute the sensitivities using the complex-step method, we only add the increment ih to the desired sensitivity variable into the existing finite element program that solves (4.28), that is, we solve for $u(\lambda+ih)$, $u(\mu+ih)$ and $u(g_k+ih_k)$ and compute $\frac{\partial u_j}{\partial \lambda}$, $\frac{\partial u_j}{\partial \mu}$, $\frac{\partial u_j}{\partial g_k}$, $j, k = 1, 2$, respectively using equation (2.5). The sensitivity equations to compute sensitivities with respect to parameters μ and λ (Refer to [16] for derivation) are given by

$$\begin{cases} -\text{div}(\sigma(u)) = f, & \text{in } \Omega \\ -\text{div}(\sigma(s_1)) - \nabla(\text{div}u) = \frac{\partial f}{\partial \lambda} & \text{in } \Omega \\ -\text{div}(\sigma(s_2)) - \Delta u - \nabla(\text{div}u) = \frac{\partial f}{\partial \mu} & \text{in } \Omega \\ \sigma(u)\bar{n} = g, & \text{on } \Gamma_1 \\ (\sigma(s_1) + (\text{Tr}(\varepsilon(u)))I)\bar{n} = \frac{\partial g}{\partial \lambda} & \text{on } \Gamma_1 \\ (\sigma(s_2) + 2\varepsilon(u))\bar{n} = \frac{\partial g}{\partial \mu} & \text{on } \Gamma_1 \\ u = 0,\ s_1 = 0,\ s_2 = 0 & \text{on } \Gamma_0 \end{cases} \tag{4.31}$$

and the system for the sensitivity with respect to the boundary g reads:

$$\begin{cases} -\text{div}(\sigma u) = f, & \text{in } \Omega \\ -\text{div}(\sigma r_1) = 0, & \text{in } \Omega \\ -\text{div}(\sigma r_2) = 0, & \text{in } \Omega \\ u = 0, & \text{on } \Gamma_0 \\ r_1 = 0, & \text{on } \Gamma_0 \\ r_2 = 0, & \text{on } \Gamma_0 \\ \sigma(u) \cdot \bar{n} = g, & \text{on } \Gamma_1 \\ \sigma(r_1) \cdot \bar{n} = (1, 0), & \text{on } \Gamma_1 \\ \sigma(r_2) \cdot \bar{n} = (0, 1), & \text{on } \Gamma_1, \end{cases} \tag{4.32}$$

where $s_1 = \frac{\partial u}{\partial \lambda}$, $s_2 = \frac{\partial u}{\partial \mu}$, $r_1 = \frac{\partial u}{\partial g_1}$, and $r_2 = \frac{\partial u}{\partial g_2}$.

The above system is solved using the finite element method (see [16] for details). In all of our computations, we use P-2 finite elements for spatial discretization, i.e., the solution u is approximated using polynomials of degree two. The mesh step size for the finite element discretization is taken to be $k = 1/30$.

As shown below in the figures, the complex-step approximations consistently agree with the ones obtained by deriving sensitivity equations up to some step size h_{crit}, where the complex-method starts to break down due to numerical underflow.

4.5.1 Elasticity system with mixed Neumann-Dirichlet boundary condition

In this first case, we consider the mixed Neumann-Dirichlet boundary problem. We take the domain Ω to be the unit square $[0,1] \times [0,1]$ with boundary $\partial\Omega = \Gamma_0 \cup \Gamma_1$ where $\Gamma_0 = [0,1] \times \{0\}$ and $\Gamma_1 = \partial\Omega \setminus \Gamma_0$.

1. **Sensitivities with respect to lamé coefficients**

   We compute the sensitivities with respect to the coefficients μ and λ for the elasticity system at $\mu = \lambda = 1$ with the right hand side and boundary data:

$$f = \begin{pmatrix} 2\mu y + 4(\mu + \lambda)xy - 2(2\mu + \lambda) \\ 2\mu y^2 + 2(2\mu + \lambda)x^2 - 2\mu x - 3\mu - \lambda \end{pmatrix}$$

$$g = \begin{pmatrix} \mu(x^2 - y^2 + 2xy^2 - 3x + 1) \\ 2(2\mu + \lambda)x^2 y - 4\mu xy - \lambda(2x + y - 1) \end{pmatrix}$$

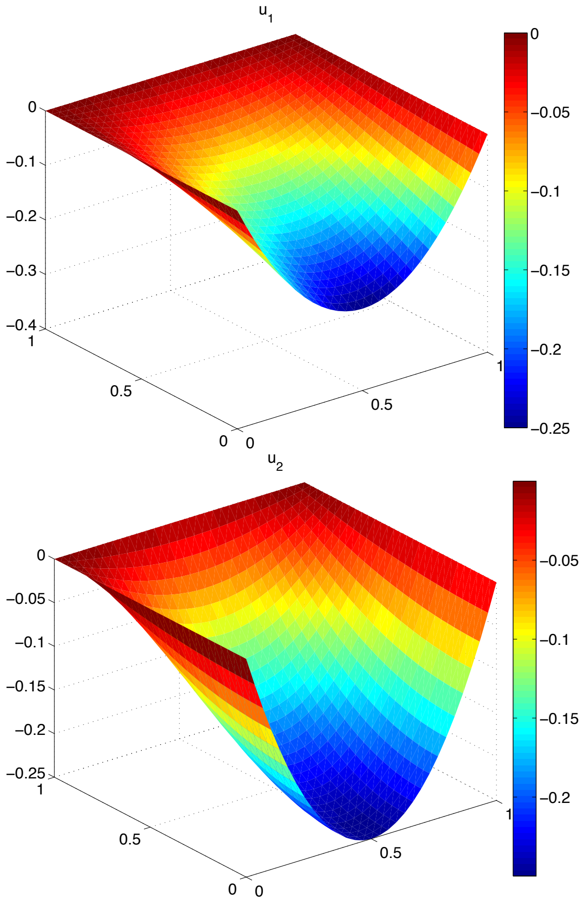
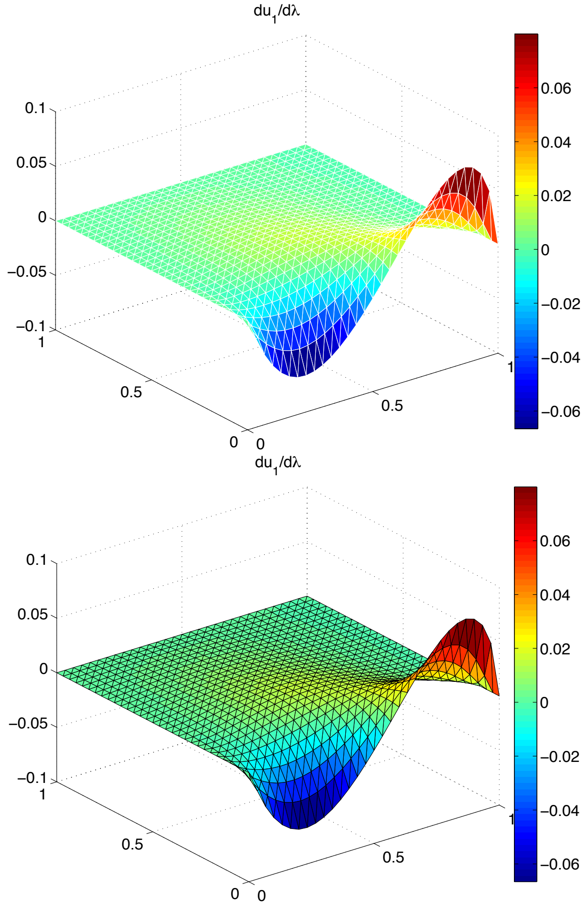


Figure 11: Finite element solution $u = (u_1, u_2)$.

In Figures 12 and 13 we present a side by side comparison of the approximation of the sensitivity functions $\frac{du_1}{d\lambda}$, $\frac{du_2}{d\lambda}$, $\frac{du_1}{d\mu}$ and $\frac{du_2}{d\mu}$ computed using the complex-step and sensitivity equations methods.


du$_1$/dλ
0.1
0.05
0
−0.05
−0.1
1
0.5
0
0
0.5
1
0.06
0.04
0.02
0
−0.02
−0.04
−0.06
du$_1$/dλ
0.1
0.05
0
−0.05
−0.1
1
0.5
0
0
0.5
1
0.06
0.04
0.02
0
−0.02
−0.04
−0.06

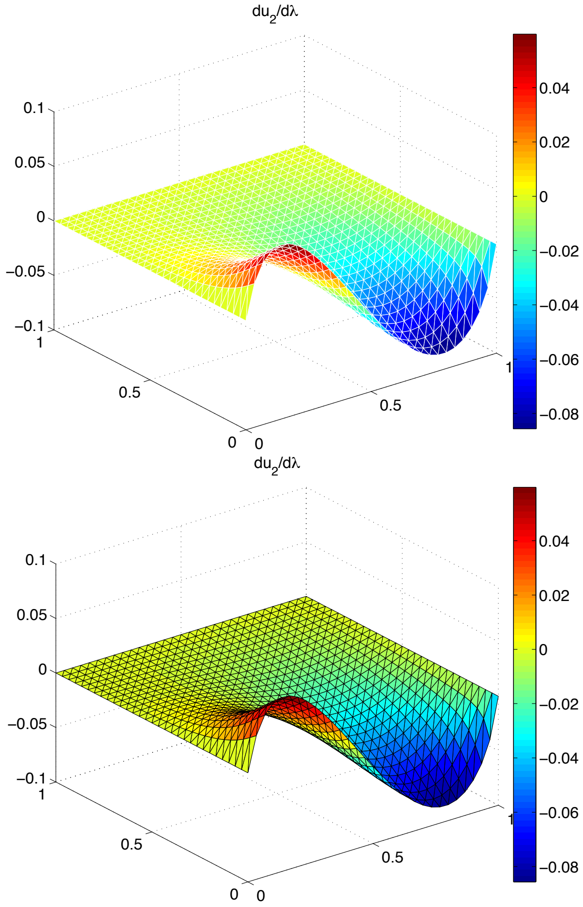
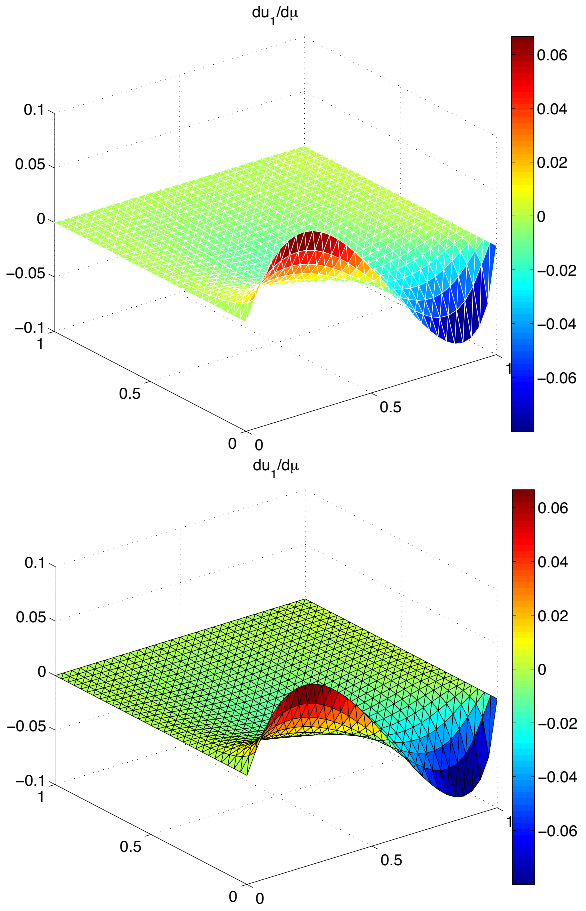


Figure 12: Sensitivities of the solution $u = (u_1, u_2)$ with respect to λ at $\lambda = 1$ computed using the complex-step method (left) and the method of sensitivity equations (right). The step size for the complex-step is taken to be $h = 10^{-40}$.


du$_1$/dμ
0.1
0.05
0
−0.05
−0.1
1
0.5
0
0
0.5
1
0.06
0.04
0.02
0
−0.02
−0.04
−0.06
du$_1$/dμ
0.1
0.05
0
−0.05
−0.1
1
0.5
0
0
0.5
1
0.06
0.04
0.02
0
−0.02
−0.04
−0.06

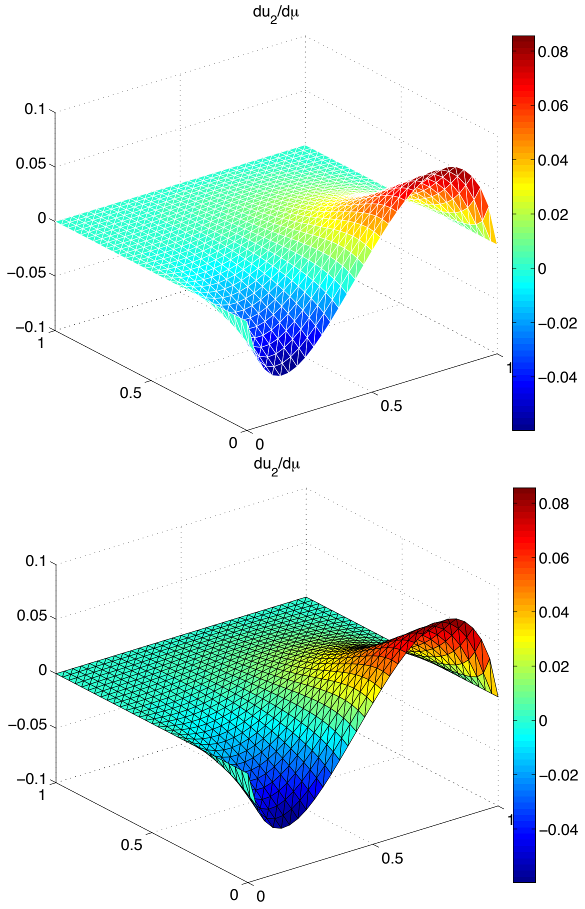
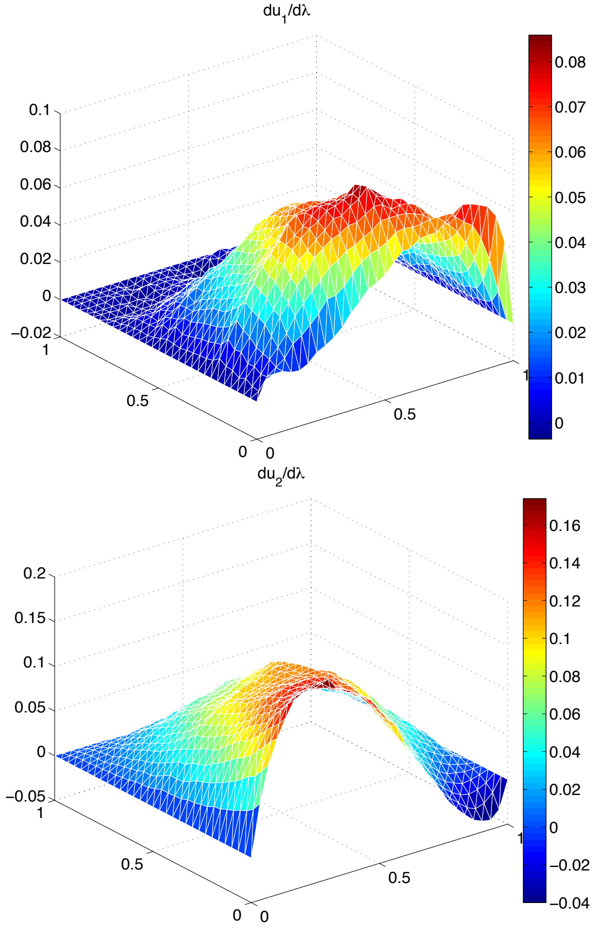


Figure 13: Sensitivities of the solution $u = (u_1, u_2)$ with respect to μ at $\mu = 1$ computed using the complex-step method (left) and the method of sensitivity equations (right). The step size for the complex-step is taken to be $h = 10^{-40}$.

The sensitivity function approximation with the complex-step method starts to decline after the critical step $h_{crit} = 10^{-318}$ and it becomes meaningless for smaller values of h than that as seen in Figure 14.



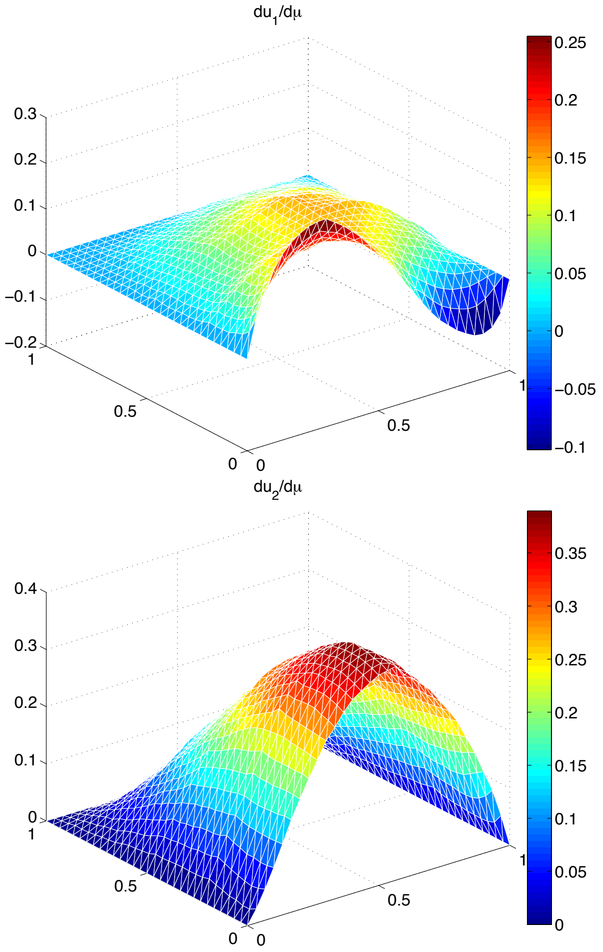


Figure 14: Inaccurate approximations of the sensitivities by the complex-step method at $h = 10^{-319}$.

2. **Sensitivity with respect to a constant boundary function**

We compute the sensitivity of the solution to the elasticity system with respect to a constant boundary function g with the right hand side data:

$$f = \begin{pmatrix} 2\mu y + 4(\mu+\lambda)xy - 2(2\mu+\lambda) \\ 2\mu y^2 + 2(2\mu+\lambda)x^2 - 2\mu x - 3\mu - \lambda \end{pmatrix}$$

$$g = (g_1, g_2) = (1, 1).$$

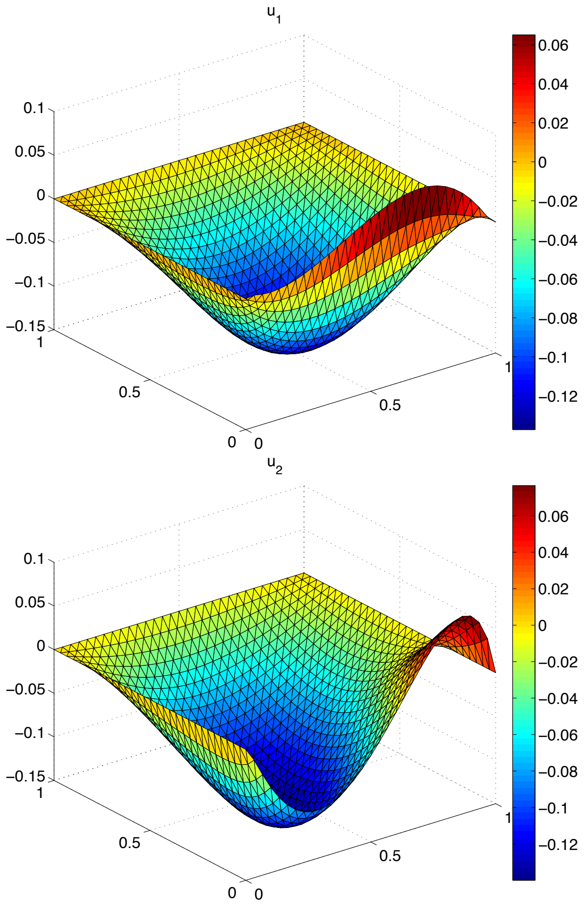
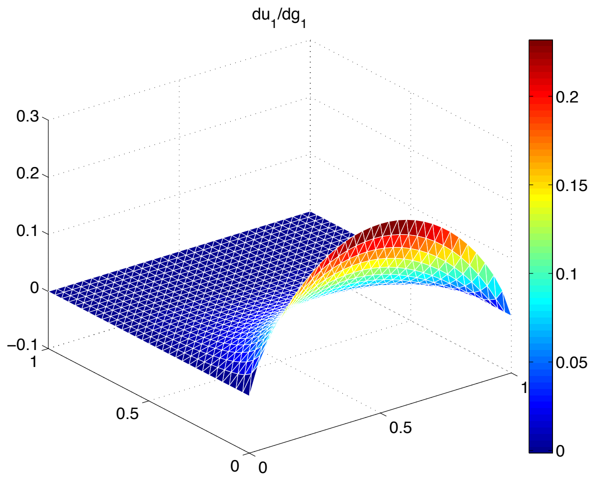
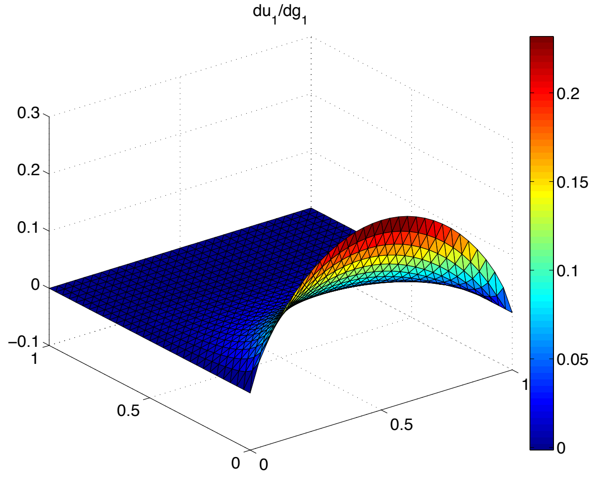


Figure 15: Finite element solution $u = (u_1, u_2)$.


du_1/dg_1
0.3
0.2
0.1
0
−0.1
1
0.5
0
0
0.5
1
0.2
0.15
0.1
0.05
0


du_1/dg_1
0.3
0.2
0.1
0
−0.1
1
0.5
0
0
0.5
1
0.2
0.15
0.1
0.05
0

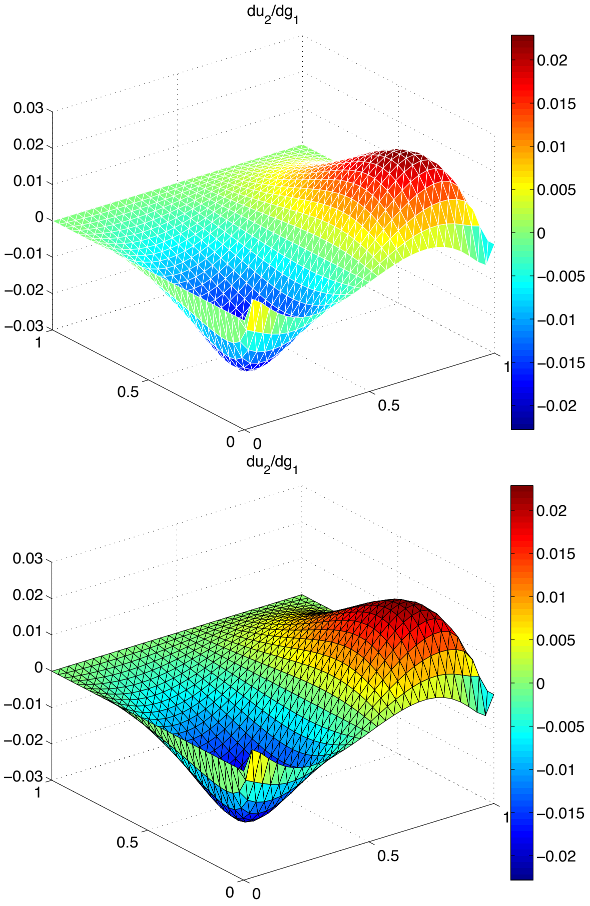
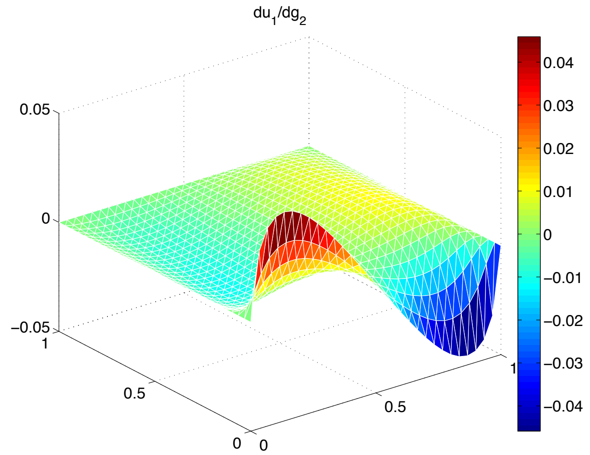
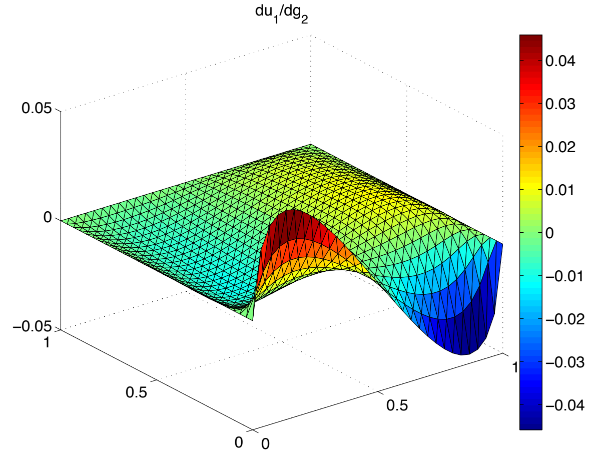


Figure 16: Sensitivity of the solution $u = (u_1, u_2)$ with respect to a constant boundary function g_1 for $g = (1, 1)$ computed using the complex-step method (left) and the method of sensitivity equations (right). The step size for the complex-step is taken to be $h = 10^{-40}$.


du$_1$/dg$_2$
0.05
0
−0.05
1
0.5
0
0
0.5
1
0.04
0.03
0.02
0.01
0
−0.01
−0.02
−0.03
−0.04


du$_1$/dg$_2$
0.05
0
−0.05
1
0.5
0
0
0.5
1
0.04
0.03
0.02
0.01
0
−0.01
−0.02
−0.03
−0.04

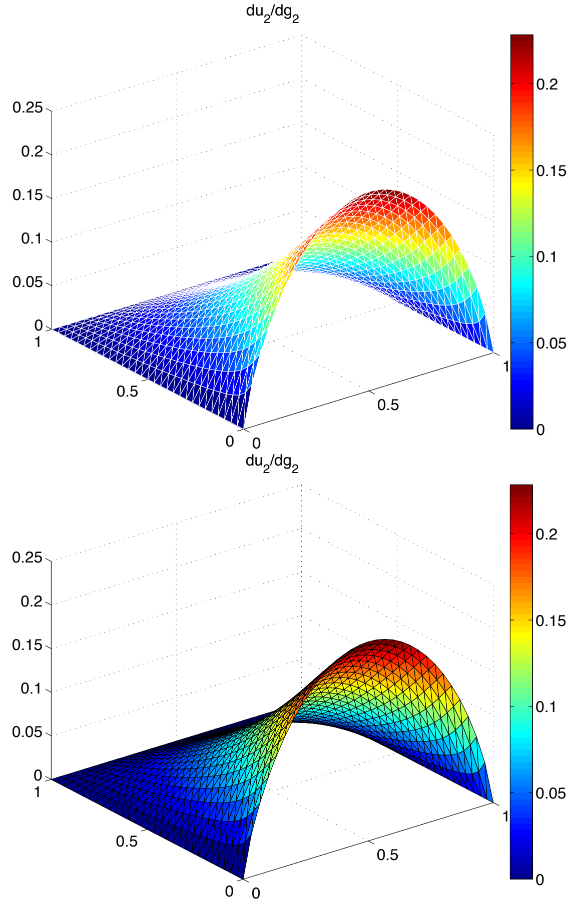


Figure 17: Sensitivity of the solution $u = (u_1, u_2)$ with respect to a constant boundary function g_2 for $g = (1, 1)$ computed using the complex-step method (left) and the method of sensitivity equations (right). The step size for the complex-step is taken to be $h = 10^{-40}$.

4.5.2 Elasticity system with homogeneous Dirichlet boundary condition

In this example, we solve the elasticity system with homogeneous Dirichlet boundary condition. Again we take Ω to be the square domain $[0,1]\times[0,1]$ with boundary $\partial\Omega=\Gamma_0$. The right had data for the system is given by:

$$f=\begin{pmatrix} -2(\lambda+2\mu)(y^2-y)-(\lambda+\mu)(2x-1)(3y^2-2y)-2\mu(x^2-x) \\ -2(\lambda+2\mu)(x^2-x)(3y-1)-(\lambda+\mu)(2x-1)(2y-1)-2\mu(y^3-y^2) \end{pmatrix}$$

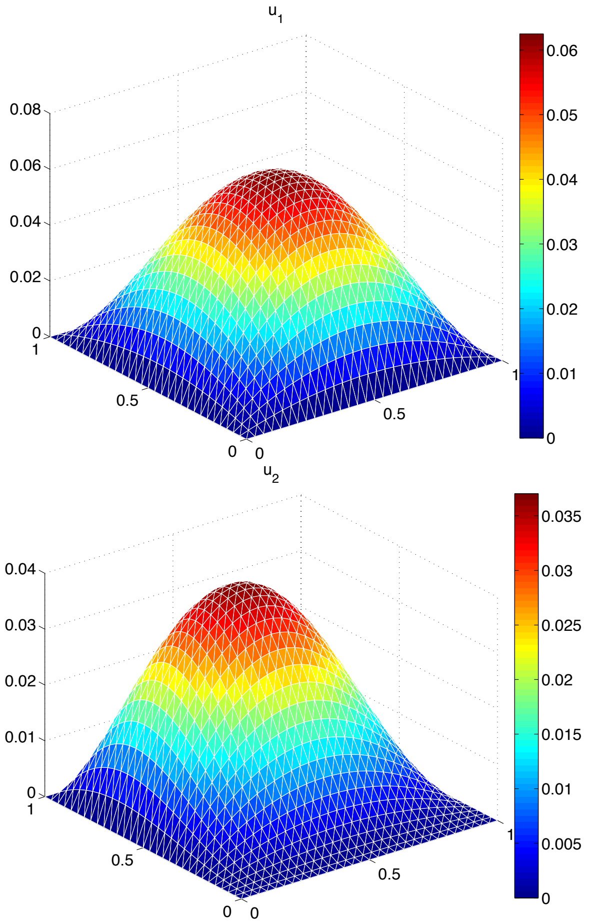
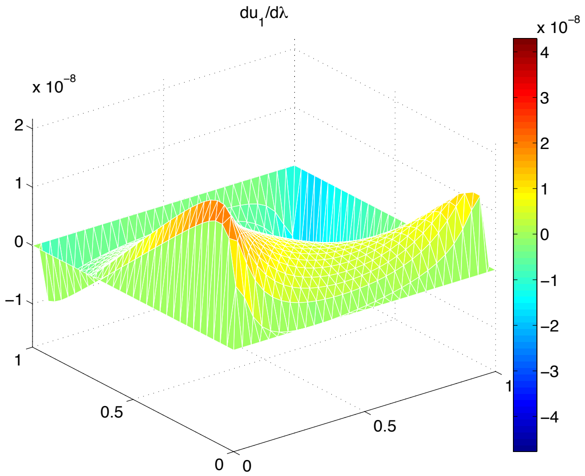
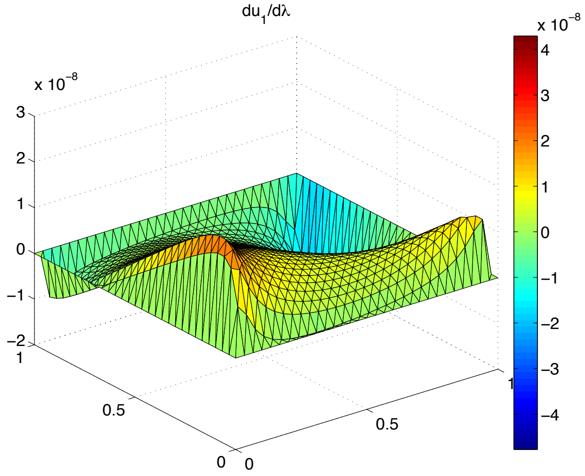


Figure 18: Solution components of the elasticity system with Dirichlet boundary condition


du$_1$/dλ
x 10^{-8}
2
1
0
−1
1
0.5
0
0
0.5
1
x 10^{-8}
4
3
2
1
0
−1
−2
−3
−4


du$_1$/dλ
x 10^{-8}
3
2
1
0
−1
−2
1
0.5
0
0
0.5
1
x 10^{-8}
4
3
2
1
0
−1
−2
−3
−4

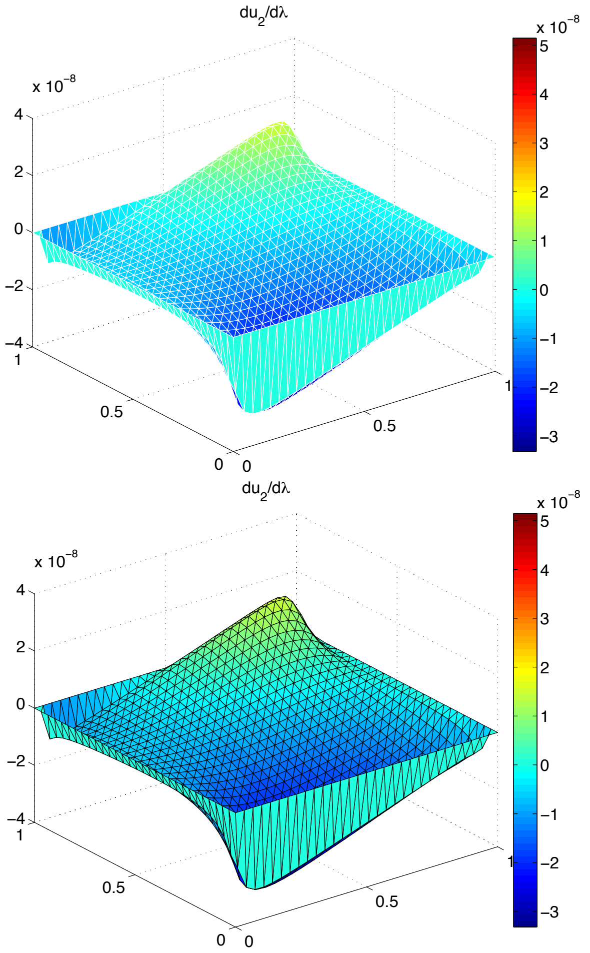
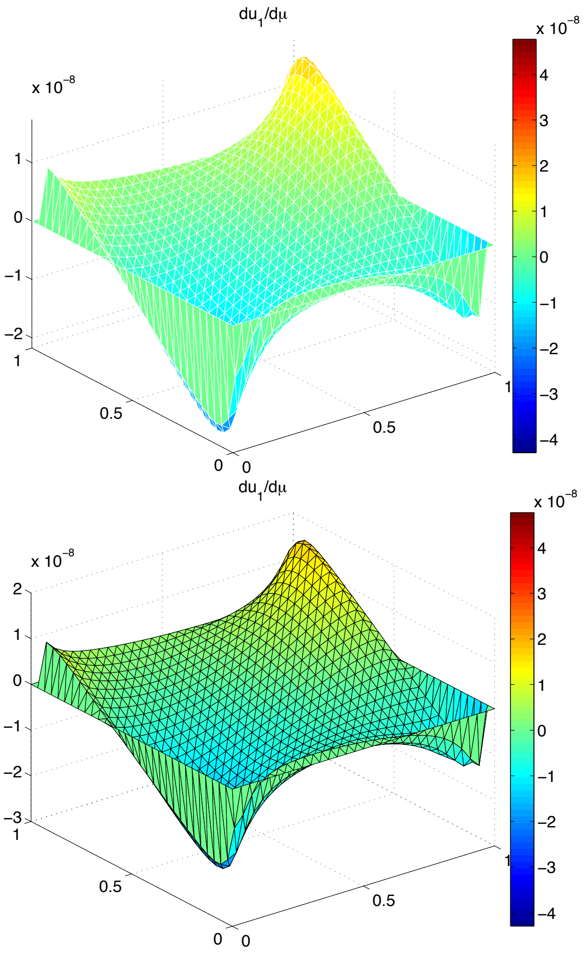


Figure 19: Sensitivities of the solution $u = (u_1, u_2)$ with respect to λ at $\lambda = 1$ computed using the complex-step method (left) and the method of sensitivity equations (right). The step size for the complex-step is taken to be $h = 10^{-40}$.


du$_1$/dμ
x 10^{-8}
1
0
−1
−2
1
0.5
0
0
0.5
1
x 10^{-8}
4
3
2
1
0
−1
−2
−3
−4
du$_1$/dμ
x 10^{-8}
2
1
0
−1
−2
−3
1
0.5
0
0
0.5
1
x 10^{-8}
4
3
2
1
0
−1
−2
−3
−4

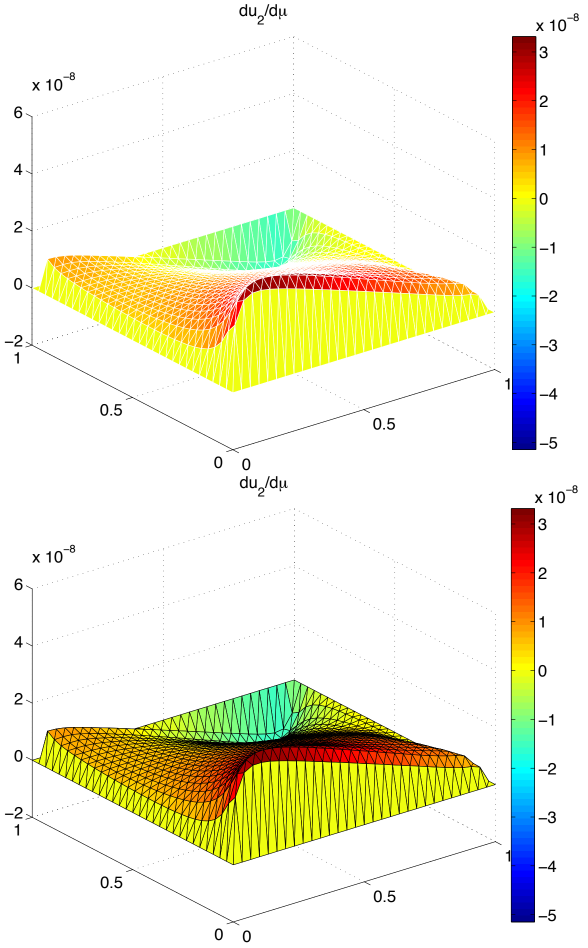
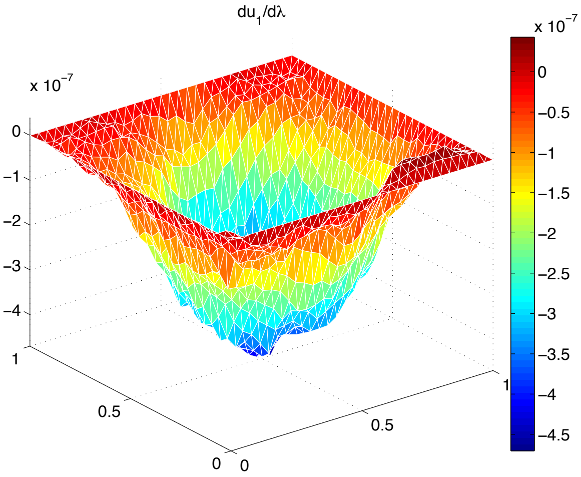
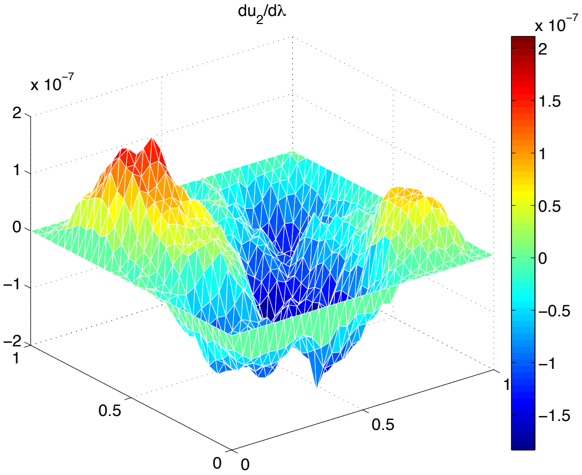


Figure 20: Sensitivities of the solution $u = (u_1, u_2)$ with respect to μ at $\mu = 1$ computed using the complex-step method (left) and the method of sensitivity equations (right). The step size for the complex-step $h = 10^{-40}$.

The sensitivity function approximation with the complex-step method starts to decline after the critical step $h_{crit} = 10^{-314}$ and it becomes meaningless for smaller values of h than that as seen in the figures below.





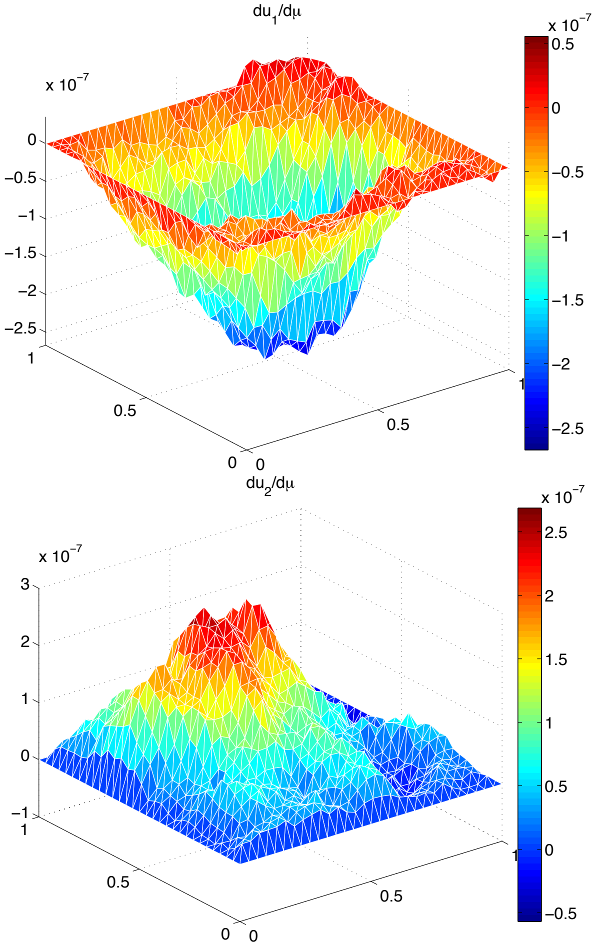


Figure 21: Very inaccurate approximations of the sensitivities by the complex-step method at $h = 10^{-314}$.

For all of our computations, we used Matlab version 2014a on a Red Hat Linux Workstation featuring Intel(R) Core(TM) i7-2600 CPU - 3.40 GHz. For solving the DDEs, we modified the Matlab solver dde23 [63] to work for complex variables. For solving the elasticity equations in the last example, the complex-step is built into a finite-element solver. In the Table 1, we report on computation time for the examples presented.

Example No.	**Complex-Step**	**Sensitivity Equations**
4.1	0.29s	0.06s
4.2	0.43s	0.15s
4.3	7.59s	0.35s
4.4 (no.1)	0.04s	0.05s
4.4 (no.2)	0.03s	0.11s
4.5.1 (no.1)	5.21s	5.49s
4.5.1 (no.2)	2.92s	3.00s
4.5.2	4.81s	4.39s

Table 1: Computation times for the complex-step method and sensitivity equations in seconds.

From the table above we see that the complex-step method provides less computational time when the parameter size is less than the dimension of the problem (Examples 4.4, 4.5.1), and runtime is large compared to solving the sensitivity equations when we have bigger number of parameters (Examples: 4.1, 4.2, 4.3 and 4.52). The reason is that to compute sensitivity to each parameter, the increament ih is added to the parameter while the rest are kept constant. Therefore, we need to run the program as many times as the parameter size resulting in additional computation time. The sensitivity equations give sensitivities with respect to all the parameters at once as a vector, which explains the decrease in computation time in some of the examples.

5 Conclusions

We have demonstrated use of the complex-step method for computing sensitivities. The method is applied to examples of various complexity and the results are compared with solutions of traditional sensitivity equations. Even though there is a step size parameter h to be chosen, the method gives consistently second order accurate approximation of the derivative starting from h as large as 10^{-2} up to $h = h_{crit}$ which is on the order of 10^{-300} (approximately our machine precision at $h = 10^{-324}$) with a true second order accuracy. Even though the complex-step formula is derived assuming analyticity of the sensitivity function, we observed that the approximation provides accurate one-sided derivatives for functions with far less smoothness. Thus our conclusion is that analyticity of the solution functions is *sufficient* but *NOT necessary* for the complex-step method to be effective.

One disadvantage of the complex-step approximation is when the number of sensitivity parameters to be considered is very large. In this case the method becomes computationally inefficient because to compute sensitivity with respect to each parameter, the increment ih has to be added separately to the parameter of interest while the rest are kept constant. Therefore, the program needs to be run as many times as the number of the parameters being investigated. On the other hand the sensitivity equations give the sensitivities with respect to all the parameters at once as a vector, which can provide a big savings in computational times (see Table 1 for comparison of computation times). The complex-step approximation is still more robust when the dimension of the underlying dynamics is large compared to the number of parameters, this could be a big advantage. Even with this additional computational cost for problems with many parameters, its ease of implementation makes it a very good alternative to methods using the sensitivity equations. In light of the amount of effort involved in deriving sensitivity equations for complicated dynamical problems, the complex-step method could be a reasonable choice. Some of the pros and cons of the complex-step method are summarized below:
Pros

- Easy to implement, no derivation of sensitivity equations required.
- The complexity of the algorithm is the same as the complexity of the algorithm evaluating the function f (In our case, the algorithm solving the differential equations).
- Less computation time is needed if the number of parameters is not large compared to the dimension of the problem.

Cons

- A step size parameter is required.
- Need to iterate the complex-step program for as many times as the number of parameters, which would mean increased computation times when the number of parameters is large.

For a problem of dimension n and m parameters, comparison of the complex-step approximation with the use of sensitivity equations is summarized below.

Complex-Step	**Sensitivity Equations**
Add a perturbation on existing algorithm	New program to solve new set of equations.
Equations of size n to solve	$n + nm$ equations to solve
Has step size	No step size
Need to run the program m times for each parameter	Program runs only once

Table 2: Comparison of the complex-step method and method of sensitivity equations.

Acknowledgments

This research was supported in part by the Air Force Office of Scientific Research under grant numbers AFOSR FA9550-12-1-0188 and AFOSR FA9550-15-1-0298, and in part by the National Science Foundation under grant number NSF DMS-1312801.

References

[1] W. Kyle Anderson and Eric J. Nielsen, Sensitivity analysis for navier– stokes equations on unstructured meshes using complex variables, *AIAA Journal,* **39** (2001).

[2] W. Kyle Anderson, Eric J. Nielsen, and D.L. Whitfield, Multidisciplinary sensitivity derivatives using complex variables, Technical report, Engineering Research Center Report, Missisipi State University, mSSU-COE-ERC-98-08, July, 1998.

[3] J. Arino, L. Wang and G. Wolkowicz, An alternative formulation for a delayed logistic equation, *J. Theo. Bio.*, **241** (2006), 109–118.

[4] H.T. Banks, Delay systems in biological models: approximation techniques, *Nonlinear Systems and Applications* (V. Lakshmikantham, ed.), Academic Press, New York, 1977, 21–38.

[5] H.T. Banks, J.E. Banks, J. Rosenheim, and K. Tillman, Modeling populations of *Lygus Hesperus* on cotton fields in the San Joaquin Valley of California: The importance of statistical and mathematical model choice, CRSC-TR15-04, Center for Research in Scientific Computation, N. C. State University, Raleigh, NC, May, 2015; *J. Biological Dynamics*, to appear.

[6] H.T. Banks, R. Baraldi, K. Cross, K. Flores, C. McChesney, L. Poag, and E. Thorpe, Uncertainty quantification in modeling HIV viral mechanics, CRSC-TR13-16, N. C. State University, Raleigh, NC, December, 2013; *Math. Biosciences and Engr.*, **12** (2015), 937–964.

[7] H.T. Banks and D. M. Bortz, A parameter sensitivity methodology in the context of HIV delay equation models, *J. Math. Biol.*, **50** (2005), 607–625.

[8] H. T. Banks, D. M. Bortz and S. E. Holte, Incorporation of variability into the mathematical modeling of viral delays in HIV infection dynamics, *Math. Biosciences*, **183** (2003), 63–91.

[9] H.T. Banks, A. Cintron-Arias, A. Capaldi and A. L. Lloyd, A sensitivity matrix based methodology for inverse problem formulation, CRSC-TR09-09, April, 2009; *J. Inverse and Ill-posed Problems*, **17** (2009), 545–564.

[10] H.T. Banks, J. Catenacci and S. Hu, Asymptotic properties of probability measure estimators in a nonparametric model, *SIAM/ASA Journal on Uncertainty Quantification*, to appear.

[11] H.T. Banks, A. Cintron-Arias and F. Kappel, Parameter selection methods in inverse problem formulation, CRSC-TR10-03, N.C. State University, February, 2010, Revised, November, 2010; in *Mathematical Modeling and Validation in Physiology: Application to the Cardiovascular and Respiratory Systems*,(J. J. Batzel, M. Bachar, and F. Kappel, eds.), pp. 43 - 73, Lecture Notes in Mathematics Vol. 2064, Springer-Verlag, Berlin 2013.

[12] H.T. Banks, Marie Doumic, Carola Kruse, Stephanie Prigent, and Human Rezaei, Information content in data sets for a nucleated-polymerization model, CRSC-TR14-15, N. C. State University, Raleigh, NC, November, 2014; *J. Biological Dynamics*, (2015), DOI:10.1080/17513758.2015.1050465

[13] H.T. Banks, S. Hu, and W.C. Thompson, *Modeling and Inverse Problems in the Presence of Uncertainty*, Chapman & Hall/CRC Press, Boca Raton, FL, 2014.

[14] H.T. Banks, J.E. Banks, K. Link, J.A. Rosenheim, C. Ross, and K.A. Tillman, Model comparison tests to determine data information content, CRSC-TR14-13, N.C. State University, Raleigh, NC; *Applied Mathematical Letters*, **43** (2015), 10–18.

[15] J.E. Banks, H.T. Banks, K. Rinnovatore, C.M. Jackson, Optimal sampling frequency and timing of threatened tropical bird populations: a Modeling approach, *Ecological Modeling*,**303** (2014),70–77.

[16] H.T. Banks, K. Bekele-Maxwell, L. Bociu, M. Noorman and K. Tillman, Sensitivity analysis for elasticity equations. In preparation.

[17] H.T. Banks, Stacey L. Ernstberger and Sarah L. Grove, Standard errors and confidence intervals in inverse problems: Sensitivity and associated pitfalls , CRSC-TR06-10, March, 2006; *J. Inverse and Ill-posed Problems*, **15** (2007), 1–18.

[18] H.T. Banks, S. Dediu, and S.E. Ernstberger, Sensitivity functions and their uses in inverse problem, *J. Inverse and Ill-posed Problems,* **15** (2006), 683—708.

[19] H.T. Banks, S. Dediu and H.K. Nguyen, Sensitivity of dynamical systems to parameters in a convex subset of a topological vector space , CRSC-TR06-25, September, 2006; *Math. Biosci. and Engr.*, **4** (2007), 403–430.

[20] H.T. Banks, S.L. Ernstberger and Shuhua Hu, Sensitivity equations for a size-structured population model, CRSC-TR07-18, September, 2007; *Quarterly of Applied Mathematics*, **LXVII** (2009), 627–660.

[21] H.T. Banks and H.K. Nguyen, Sensitivity of dynamical systems to Banach space parameters, CRSC-TR05-13, February, 2005; *J. Math. Analysis and Applications*, **323** (2006), 146–161.

[22] H.T. Banks and Keri L. Rehm, Experimental design for vector output systems, CRSC-TR12-11, N. C. State University, Raleigh, NC, April, 2012; *Inverse Problems in Sci. and Engr.*, **22** (2014) 557–590. doi:10.1080/17415977.2013.797973.

[23] H.T. Banks and K.L. Rehm, Parameter estimation in distributed systems: Optimal design, CRSC TR14-06, N. C. State University, Raleigh, NC, May, 2014; *Eurasian Journal of Mathematical and Computer Applications*, **2** (2014), 70-79.

[24] H.T. Banks, Danielle Robbins and Karyn L. Sutton, Theoretical foundations for traditional and generalized sensitivity functions for nonlinear delay differential equations CRSC-TR12-14, N. C. State University, Raleigh, NC, July, 2012; *Math. Biosci. Engr.*, **10** (2013), 1301–1333; http://dx.doi.org/10.3934/mbe.2013.10.1301

[25] H.T. Banks and H.T. Tran, *Mathematical and Experimental Modeling of Physical and Biological Processes*, CRC Press, Boca Raton, FL, July, 2008, 308pp. published, January 2, 2009.

[26] L. Bociu, G. Guidoboni, R. Sacco, and J.T. Webster, Analysis of nonlinear poro-elastic and poro-visco-elastic models, *Archive of Rational Mechanics and Analysis,* 2015, submitted.

[27] J.A. Burns, E.M. Cliff and S.E. Doughty, Sensitivity analysis and parameter estimation for a model of Chlamydia Trachomatis infection, *J. Inverse Ill-Posed Problems*, **15** (2007), 19–32.

[28] S.N. Busenberg and K.L. Cooke, eds., *Differential Equations and Applications in Ecology, Epidemics, and Population Problems*, Academic Press, New York, 1981.

[29] V. Capasso, E. Grosso and S.L. Paveri-Fontana, eds., *Mathematics in Biology and Medicine*, Lecture Notes in Biomath., Vol. **57**, Springer-Verlag, Berlin, Heidelberg, New York, 1985.

[30] J. Caperon, Time lag in population growth response of *Isochrysis Galbana* to a variable nitrate environment, *Ecology*, **50** (1969), 188–192.

[31] A. Casal and A. Somolinos, Forced oscillations for the sunflower equation, entrainment, *Nonlinear Analysis: Theory, Methods and Applications*, **6** (1982), 397–414.

[32] Paola Causin, Giovanna Guidoboni, Alon Harris, Daniele Prada, Riccardo Sacco, and Samuele Terragni, A poroelastic model for the perfusion of the lamina cribrosa in the optic nerve head, *Mathematical Biosciences,* (2014), March 11, submitted.

[33] J.M. Cushing, *Integrodifferential Equations and Delay Models in Population Dynamics*, Lec. Notes in Biomath., Vol. **20**, Springer-Verlag, Berlin and New York, 1977.

[34] M. Davidian and D.M. Giltinan, *Nonlinear Models for Repeated Measurement Data*, Chapman and Hall, London, 2000.

[35] A. Ronald Gallant, *Nonlinear Statistical Models*, John Wiley and Sons, New York, 1987.

[36] U. Forys and A. Marciniak-Czochra, Logistic equations in tumor growth modelling, *Int.J. Appl. Math. Comput. Sci.*, **13** (2003), 317–325.

[37] Giovanna Guidoboni, Alon Harris, Lucia Carichino, Yoel Arieli and Brent A. Siesky, Effect of intraocular pressure on the hemodynamics of the central retinal artery: A mathematical model, *Math. Biosci. Engr.*, **11** 2014, 523–546. doi:10.3934/mbe.2014.11.523

[38] L. Glass and M. C. Mackey, Pathological conditions resulting from instabilities in physiological control systems, *Ann. N. Y. Acad. Sci.* **316** (1979), 214–235.

[39] B.C. Goodwin, *Temporal Organization in Cells: a dynamic theory of cellular control processes*, Academic Press, London, 1963.

[40] B.C. Goodwin, Oscillatory behavior in enzymatic control processes, *Adv. Enzyme Reg.*, **3** (1965) 425–439.

[41] K. Gopalsamy, *Stability and Oscillations in Delay Differential Equations of Population Dynamics*, Kluwer, Dordrecht, 1992.

[42] Rafael Grytz, Günther Meschke and Jost B. Jonas, The collagen fibril architecture in the lamina cribrosa and peripapillary sclera predicted by a computational remodeling approach, *Biomech Model Mechanobiol.*, **10** (2011), 371—382 DOI 10.1007/s10237-010-0240-8.

[43] K. P. Hadeler, Delay equations in biology, in *Functional differential equations and approximation of fixed points* (H. O. Peitgen and H. O. Walther, eds.), Lect. Notes Math. **730** Springer-Verlag Berlin, 1979, 136–156.

[44] J. Hale, *Theory of Functional Differential Equations*, Springer-Verlag, New York, 1977.

[45] F.C. Hoppensteadt, ed., *Mathematical Aspects of Physiology*, Lectures in Applied Math, Vol. 19, American Mathematical Society, Providence, 1981.

[46] G. E. Hutchinson, Circular causal systems in ecology, *Annals of the NY Academy of Sciences*, **50** (1948), 221–246.

[47] G. E. Hutchinson, *An Introduction to Population Ecology*, Yale University, New Haven, 1978.

[48] Mark Kot, *Elements of Mathematical Ecology,* Cambridge University Press, Cambridge, 2001.

[49] Y. Kuang, *Delay Differential Equations With Applications in Population Dynamics*, Vol. **191** of Mathematics in Science and Engineering, Academic Press, New York, NY, 1993.

[50] Kok-Lam Lai, Generalizations of the complex-step derivative approximation, PhD thesis, University at Buffalo, Buffalo, NY, 2006.

[51] J. N. Lyness, Numerical algorithms based on the theory of complex variables, *Proc. ACM 22nd Nat. Conf.,* **4** (1967), 124—134.

[52] J. N. Lyness and C. B. Moler, Numerical differation of analytic functions, *SIAM J. Numer. Anal.,* **4** (1967), 202—210.

[53] N. MacDonald, Time lag in a model of a biochemical reaction sequence with end-product inhibition, *J. Theor. Biol.*, **67** (1977), 727–734.

[54] N. MacDonald, *Time Lags in Biological Models*, Lecture Notes in Biomathematics **27**, Springer, Berlin, 1978.

[55] Joaquim R. R. A. Martins, Ilan M. Kroo, and Juan J. Alonso. An automated method for sensitivity analysis using complex variables. AIAA Paper 2000-0689 (Jan.), 2000.

[56] Joaquim R. R. A. Martins, Peter Sturdza, and Juan J. Alonso. The complex-step deriva- tive approximation. Journal ACM Transactions on Mathematical Software (TOMS), 2003.

[57] N. Minorsky, Self-excited oscillations in dynamical systems possessing retarded actions, *Journal of Applied Mechanics*, **9** (1942) A65–A71.

[58] N. Minorsky, On non-linear phenomenon of self-rolling, *Proceedings of the National Academy of Sciences*, **31** (1945), 346–349.

[59] N. Minorsky, *Nonlinear Oscillations*, Van Nostrand, New York, 1962.

[60] S. I Rubinow, *Introduction to Mathematical Biology,* J. Wiley, New York, 1975.

[61] R. Schuster and H. Schuster, Reconstruction models for the Ehrlich Ascites Tumor of the mouse, in: *Mathematical Population Dynamics, Vol. 2* (edited by O. Arino, D Axelrod, M. Kimmel), Wuertz, Winnipeg, 1995, 335–348.

[62] G.A.F. Seber and C.J. Wild, *Nonlinear Regression*, J. Wiley & Sons, Hoboken, NJ, 2003.

[63] Lawrence F. Shampine and Skip Thompson, Solving ddes in matlab, *Applied Numerical Mathematics*, **37**, No. 4 (2001), 441–458.

[64] F. R. Sharpe and A. J. Lotka, Contribution to the analysis of malaria epidemiology IV: Incubation lag, supplement to *Amer. J. Hygiene* **3** (1923), 96–112.

[65] W. Squire and G. Trapp, Using complex variables to estimate derivatives of real functions, *SIAM Review,* **10** (1998), 100—112.

H.T. Banks
Center for Research in Scientific Computation
Department of Mathematics
North Carolina State University
Raleigh, NC 27695-8212
htbanks@ncsu.edu
Kidist Bekele-Maxwell
Center for Research in Scientific Computation
Department of Mathematics
North Carolina State University
Raleigh, NC 27695-8212
ktzeleke@ncsu.edu
Lorena Bociu
Center for Research in Scientific Computation
Department of Mathematics
North Carolina State University
Raleigh, NC 27695-8212
lvbociu@ncsu.edu
Marcella Noorman
Center for Research in Scientific Computation
Department of Mathematics
North Carolina State University
Raleigh, NC 27695-8212
mjnoorma@ncsu.edu
Kristen Tillman
Center for Research in Scientific Computation
Department of Mathematics
North Carolina State University
Raleigh, NC 27695-8212
kristentillman@ncsu.edu

Received 10.09.2015, Accepted 28.09.2015