COMPLEX-STEP GRADIENT APPROXIMATION FOR ROBUSTNESS ANALYSIS OF NONLINEAR SYSTEMS

Jongrae Kim, Declan G. Bates, Ian Postlethwaite

Control & Instrumentation Research Group
Department of Engineering
University of Leicester
Leicester LE1 7RH, UK
Email: jrk7, dgb3, ixp@le.ac.uk

Abstract: In this paper, the complex-step perturbation method is extended to the setting of optimisation problems involving dynamical systems modelled as nonlinear differential equations. The main advantage of the complex-step method for gradient approximation is that it entails no subtraction cancellation error, and therefore the truncation error can be made arbitrarily small. The method is applied to two robust performance analysis problems and is shown to provide more accurate solutions and improved convergence times when compared with standard finite difference-based approaches.

Keywords: Robustness Analysis, Optimisation, Nonlinear Systems

1. INTRODUCTION

Derivative approximation using complex variables was first presented in Lyness, *et al.* (1967) and Lyness (1967). The usefulness of this formulation for approximating derivatives of real-valued functions was highlighted by Squire and Trapp (1998). The main advantages of the approach are its simplicity, since it is just as simple as finite difference-based methods if all operations and functions are properly defined in the complex domain, and its accuracy, due to the absence of subtraction cancellation error. Martins, *et al.* uses the complex-step method to estimate the sensitivity for structural finite element methods and computational fluid dynamics code in (Martins *et al.*, 2000). In (Martins *et al.*, 2001) Martins, *et al.* showed the connection between the complex-step method and algorithmic differentiation and also gave some suggestions for ways to further improve the accuracy of the complex-step method. In another new application of this method, Cerviño, *et al.* applied the complex-step approach to pseudo-spectral algorithms (Cerviño and Bewley, 2003).

In this paper, we show how to extend the complex-step perturbation method to the setting of optimisation problems involving dynamical systems modelled as nonlinear differential equations. In particular, we consider the problem of worst-case performance analysis of systems subject to parametric uncertainty. We show that, for such problems, the complex-step perturbation method can provide more accurate solutions, in a faster time, than are obtained using the standard finite difference-based approximations which are commonly used in optimisation algorithms such as "fmincon" in the MATLAB optimization toolbox, (MathWorks, 2003). In addition, the proposed approach is shown to be easier to implement and automate, since, unlike finite difference-based approaches, it does not require the calculation of an "optimal" step size for each problem.

2. COMPLEX STEP PERTURBATION METHOD

In this section we briefly describe the complex-step perturbation method for approximating the

derivative of a real-valued scalar function. Subsequently, we extend this approach to the setting of optimisation problems involving dynamical systems modelled as nonlinear differential equations. Finally, we introduce the two robustness analysis problems considered in this study.

2.1 Gradient Approximation using Complex Steps

Consider

$$J = J(x_\delta) \tag{1}$$

where $J(\cdot)$ is a real-valued function, i.e., $J \in \mathbb{R}$, x_δ is a parameter vector in Δ, which is a subset of $\mathbb{R}^p$, and p is a positive integer. Define the k^{th} element perturbation of x_δ at $x_\delta = x_\delta^*$ as follows:

$$\alpha_\delta^{k\pm} = \begin{cases} x_{\delta_j}^* \pm h, & \text{for } j = k \\ x_{\delta_j}^*, & \text{for } j \neq k \end{cases} \tag{2}$$

where h is a positive real number. One of the standard ways to obtain the approximate slope at x_δ^* towards $x_{\delta_k}^*$ is to use a finite difference formula such as (Squire and Trapp, 1998):

$$\left.\frac{\partial J(x_\delta)}{\partial x_{\delta_k}}\right|_{x_\delta = x_\delta^*} \approx \frac{J(\alpha_\delta^{k+}) - J(\alpha_\delta^{k-})}{2h} \tag{3}$$

Note that the truncation error for the above central difference formulation is $O(h^2)$, and therefore, to reduce the truncation error h should be chosen to be as small as possible. However, when h is below a certain number [1], the subtraction cancellation error will become dominant in the numerator of the above expression. Hence, when the finite difference method is used to approximate the derivative, (a) there is a limit to the accuracy that can be achieved, and (b) the optimal value of h has to be computed for each problem.

To avoid the above difficulties, the complex perturbation method can be used, since, in this method, there is no subtraction cancellation error. Consider the complex perturbation of the k-th element of x_δ at $x_\delta = x_\delta^*$ as follows:

$$z_\delta^k = \begin{cases} x_{\delta_j}^* + ih, & \text{for } j = k \\ x_{\delta_j}^*, & \text{for } j \neq k \end{cases} \tag{4}$$

where i is the imaginary number, i.e. $i = \sqrt{-1}$. The function $J(z_\delta^k)$ that is perturbed in the complex direction can be expanded in a Taylor series as follows: (Squire and Trapp, 1998):

$$\begin{aligned} J(z_\delta^k) = J(x_\delta^*) &+ ih \left.\frac{\partial J(x_\delta)}{\partial x_{\delta_k}}\right|_{x_\delta = x_\delta^*} \\ &- \frac{h^2}{2!} \left.\frac{\partial^2 J(x_\delta)}{\partial x_{\delta_k}^2}\right|_{x_\delta = x_\delta^*} \\ &- \frac{ih^3}{3!} \left.\frac{\partial^3 J(x_\delta)}{\partial x_{\delta_k}^3}\right|_{x_\delta = x_\delta^*} + \dots \end{aligned} \tag{5}$$

[1] The certain number varies from problem to problem.

Taking imaginary parts from both sides and dividing by h, the approximation of the derivative is given by (Squire and Trapp, 1998)

$$\left.\frac{\partial J(x_\delta)}{\partial x_{\delta_k}}\right|_{x_\delta = x_\delta^*} \approx \frac{\operatorname{Im}\left[J(z_\delta^k)\right]}{h} \tag{6}$$

where $\operatorname{Im}(\cdot)$ is the imaginary part of the argument. The approximation error is $O(h^2)$, which is the same as for the central difference method. Note, however, that since for this formulation there is no subtraction cancellation error, h can now be made arbitrarily small (as long as it remains inside the numerical range for real numbers of the computer). The gradient of $J(x_\delta)$ can be approximated by applying the above operation p-times for each k.

2.2 Extension To Dynamical Systems

In this section the complex perturbation method is applied to approximate the gradient of a cost function $J(x_\delta)$ for a dynamical system, which is modelled as a nonlinear differential equation:

$$\dot{x} = f(x_\delta,\, x,\, t) \tag{7}$$

where t is in $[t_0,\, \infty)$, the initial condition is given by $x(t_0)$, and x is the state in $\mathbb{R}^n$. Now, x_δ can be interpreted as a vector of uncertain parameters in $\Delta \subset \mathbb{R}^p$. Each element of x_δ represents a possible value for a real physical uncertain parameter in the system model (7). In this paper, $f(x_\delta,\, x,\, t)$ is assumed to be a piecewise continuous function in $\mathbb{R}^n$, and the conditions for the existence and the uniqueness of the solution are also assumed.

The gradient of the cost function can now be approximated by (6). The value of J for a complex perturbed x_δ can easily be calculated if the cost function is given in an explicit closed form involving x_δ. [2] If this is not the case, an implicit method to calculate the value of the complex perturbed cost function has to be derived, as follows.

By substituting x_δ in (7) by z_δ^k, (7) becomes

$$\dot{x}(z_\delta^k,\, t) = f(z_\delta^k,\, x,\, t) \tag{8}$$

Let $x(z_\delta^k,\, t)$ be the solution of the above differential equation. $x(z_\delta^k,\, t)$ can be divided into two parts i.e., real and imaginary parts:

$$x(z_\delta^k,\, t) = x_R(z_\delta^k,\, t) + i x_I(z_\delta^k,\, t) \tag{9}$$

where

$$x_R(z_\delta^k,\, t) = \operatorname{Re}[x(z_\delta^k,\, t)] \tag{10a}$$

$$x_I(z_\delta^k,\, t) = \operatorname{Im}[x(z_\delta^k,\, t)] \tag{10b}$$

and $\operatorname{Re}(\cdot)$ is the real part of the argument. Substituting (9) into (8), the following differential equations are obtained:

[2] Sometimes, the system includes a noise model and the optimisation problem then becomes a stochastic one. For simplicity, the stochastic case is not considered in this paper.

$$\dot{x}_R(z_\delta^k,\, t) = \mathrm{Re}[f(z_\delta^k,\, x,\, t)] \tag{11a}$$
$$\dot{x}_I(z_\delta^k,\, t) = \mathrm{Im}[f(z_\delta^k,\, x,\, t)] \tag{11b}$$

Note that the above operation, i.e., calculating the real and the imaginary parts of $f(z_\delta^k,\, x,\, t)$, is very easy in most cases. Of course, all of the calculations and operations inside $f(z_\delta^k,\, x,\, t)$ are normally defined in the real number domain, and therefore they must be appropriately transformed into the complex number domain. Details of how some standard operations should be defined in the complex domain can be found in (Martins *et al.*, 2000). Also, note that because the dimension of differential equation is doubled, this may cause some numerical problems when n is large number and this increases the calculation cost. Since the original initial condition is given by $x(t_0)$, the complex initial condition has to satisfy:

$$x(t_0) = x_R(t_0) + i x_I(t_0) \tag{12}$$

The initial condition for each differential equation is therefore given by

$$x_R(t_0) = x(t_0) \tag{13a}$$
$$x_I(t_0) = 0 \tag{13b}$$

Finally, the solution, (9), is obtained by solving the differential equation, (11), with the initial condition, (13).

2.3 Robust Performance Analysis

In the field of robustness analysis, a typical robust performance analysis problem is to find the value of x_δ that maximizes the following cost function (Tierno *et al.*, 1995):

$$\max_{x_\delta \in \Delta} J(x_\delta) = \frac{1}{2} \int_{t_0}^{t_f} x^T(t)\, x(t)\, dt \tag{14}$$

which is a finite L_2 gain (assuming the system remains stable for the defined level of uncertainty), where t_f is the final time and is greater than t_0. For this problem, the term inside the integration with the solution from the complex perturbed differential equation becomes

$$\begin{aligned} x^T(z_\delta^k,\, t)\, x(z_\delta^k,\, t) &= (x_R^T + i x_I^T)(x_R + i x_I) \\ &= (x_R^T\, x_R - x_I^T\, x_I) + i\, 2\, x_R^T\, x_I \end{aligned} \tag{15}$$

Note that the transpose operation is not changed to complex conjugate transpose but remains the same as the original transpose for both real and imaginary parts. Otherwise, the above has no imaginary part and $\mathrm{Im}(J)$ is always equal to zero. Substituting (15) into (14) and taking imaginary parts

$$\mathrm{Im}\,[J(z_\delta^*)] = \int_{t_0}^{t_f} x_R^T(t)\, x_I(t) dt \tag{16}$$

the derivative is approximated by

$$\left.\frac{\partial J(x_\delta)}{\partial x_{\delta_k}}\right|_{x_\delta = x_\delta^*} \approx \frac{1}{h} \int_{t_0}^{t_f} x_R^T(t)\, x_I(t) dt \tag{17}$$

By repeating the above calculation p-times for each x_{δ_k}, the gradient is obtained.

Another typical cost function frequently used in the robustness analysis of flight control systems, (Fielding *et al.*, 2002), has the following form :

$$\max_{x_\delta \in \Delta} J(x_\delta) = \max_{t \in [t_0,\, t_f]} x(t) \tag{18}$$

where, for example, $x(t)$ could be the angle of attack of the aircraft, and the problem is to find the uncertain parameter combination that gives the maximum angle of attack for a given pilot stick input. The "max" function is redefined in (Martins *et al.*, 2000) as follows:

$$\max(z_1,\, z_2) = \begin{cases} z_1, & \text{for } a_1 \geq a_2 \\ z_2, & \text{for } a_1 < a_2 \end{cases} \tag{19}$$

where

$$z_1 = a_1 + i b_1 \tag{20a}$$
$$z_2 = a_2 + i b_2 \tag{20b}$$

As a result, the new "max" function still compares only the real parts of the arguments and returns the complex number with largest real part. Therefore, the following has to be returned for the complex perturbed cost function:

$$\mathrm{Im}\,[J(z_\delta^*)] = \mathrm{Im}\left[x_R(\tilde{t}) + i x_I(\tilde{t})\right] = x_I(\tilde{t}) \tag{21}$$

where $\tilde{t}$ is defined by

$$\tilde{t} = \arg \max_{t \in [t_0,\, t_f]} x_R(t) \tag{22}$$

i.e., $\tilde{t}$ is the instant in $[t_0,\, t_f]$ when $x_R(\tilde{t})$ is greater than and equal to any other $x_R(t)$ in $[t_0,\, t_f]$. Finally, in this case the derivative is approximated by

$$\left.\frac{\partial J(x_\delta)}{\partial x_{\delta_k}}\right|_{x_\delta = x_\delta^*} \approx \frac{x_I(\tilde{t})}{h} \tag{23}$$

Note, finally, that both of the robustness analysis problems defined above may result in non-convex optimisation problems in general, and therefore the search for the worst-case combination of uncertain parameters corresponds to the computation of lower bounds on worst-case performance, (Tierno *et al.*, 1995).

3. EXAMPLES

In this section the complex-step method described in the previous section is applied to two robust performance problems: a first-order linear system with a finite L_2 gain cost function and a second-order nonlinear system with a cost function corresponding to the maximum overshoot in response to a unit step reference demand. The optimisation algorithm used in all cases is "fmincon" in MATLAB (MathWorks, 2003), with the gradient for the optimisation being supplied using both central finite difference and complex-step approximations.

Table 1 (Example 1) No. of cost function evaluations & convergence results for FD (central finite difference) and CS (complex-step)

Method	h	Number of cost function evaluations	Convergent point $[x_{\delta_1}, x_{\delta_2}, \ldots, x_{\delta_{10}}]$
FD	10^{-1}	14784	[0.267, 0.073, 0.267, 0.073, 0.587, 0.347, 0.587, 0.347, 0.587, 0.347]
FD	10^{-6}	9009	[1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000]
FD	10^{-16}	4158	[−1.200, 1.000, −1.200, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000]
CS	10^{-1}	2662	[−0.953, 0.913, −0.953, 0.913, 1.028, 1.056, 1.028, 1.056, 1.028, 1.056]
CS	10^{-6}	4356	[1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000]
CS	10^{-50}	4598	[1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000]

3.1 First-Order System With Finite L_2 Gain Cost Function

Consider the following first-order linear system:

$$\dot{x} = -\eta(x_\delta)x \tag{24}$$

where $x(0) = 1$, x_δ is in Δ, which is a hyperbox in $\mathbb{R}^{10}$, each of whose edges is bounded by $-4 \leq x_{\delta_j} \leq 4$ for $j = 1, 2, \ldots, 10$, and $\eta(x_\delta)$ is the ten-dimensional Rosenbrock's function given by (Spall, 2003) [3]

$$\eta\left(x_\delta\right) = \sum_{i=1}^{5}\left[100\left(x_{\delta_{2i}} - x_{\delta_{2i-1}}^2\right)^2 + \left(1 - x_{\delta_{2i-1}}\right)^2\right] \tag{25}$$

The cost function is a finite L_2 gain as follows:

$$\max_{x_\delta \in \Delta} J(x_\delta) = \frac{1}{2}\int_0^1 x^2(t)dt \tag{26}$$

Since (25) has a global minimum, 0, at $x_\delta = (1, 1, \ldots, 1)$, the maximum of the cost function occurs at the same location of x_δ. The optimal cost is 1/2.

The initial value of x_δ input to the optimisation routine is $(-1.2, 1, -1.2, 1, \ldots, 1)$ (Spall, 2003). The errors between the approximated gradient and the analytical gradient for each iteration are shown for each method in Figures 1 and 2. The errors are defined as:

$$\delta g = |g_t - g_a| \tag{27a}$$

$$\delta\theta = \arccos\left(g_t \cdot g_a\right)/(|g_t|\,|g_a|)180/\pi \tag{27b}$$

where g_t, the true (analytical) gradient, is given by

$$g_t = \begin{cases} \dfrac{[-1 + (1+2\eta)e^{-2\eta}]}{4\eta^2}\dfrac{\partial\eta}{\partial x_\delta}, & \text{for } \eta(x_\delta) \neq 0 \\ [0, 0, \ldots, 0], & \text{for } \eta(x_\delta) = 0 \end{cases} \tag{28}$$

and g_a is the approximate gradient either from the finite difference method or the complex-step method.

Figure 1 shows the gradient approximation errors of the central difference method for different sizes of the step, h. For h equal to 10^{-1} the truncation error is not small enough to give a reasonable gradient approximation. As shown in Table 1, the optimisation routine takes a long time to converge and finally doesn't converge to the correct solution. When h is equal to 10^{-6}, a reasonable approximation of the gradient is produced since the truncation error becomes small. However, if we try to further improve the accuracy of the approximation by decreasing h to, for example, $h = 10^{-16}$, the subtraction cancellation error becomes dominant causing the approximation to be always equal to zero and the algorithm to fail to converge to the correct solution.

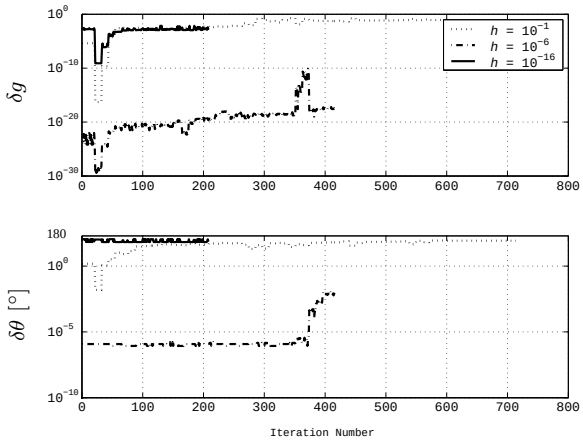

Fig. 1. Gradient approximation error of central finite difference method

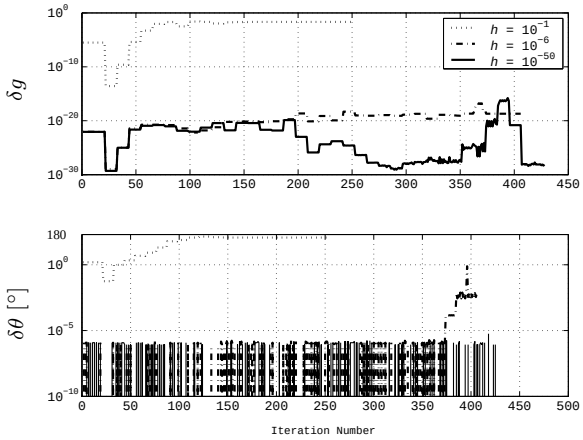

Fig. 2. Gradient approximation error of complex-step method

[3] See Example 2.5 in (Spall, 2003)

For the complex-step approximation, it can be seen in Figure 2 that when h is equal to 10^{-1}, the gradient is inaccurate in both magnitude and direction since the truncation error is too large, and in fact the algorithm fails to converge to the correct solution (Table 1). However, for h equal to 10^{-6} the approximation is very close to the analytical one. Moreover, if h is further decreased to 10^{-50}, an even better approximation of the gradient is produced, due to the absence of any subtraction cancellation error. The resulting approximation of the gradient is extremely good - the error for $h = 10^{-50}$ is bounded by 10^{-15} in magnitude and 10^{-5} degrees in direction.

For each value of h for each method, the number of function evaluations required to converge to a solution are given in Table 1. From the table, it can be seen that, as well as being more accurate, the complex-step method is also significantly faster. This is not surprising, since the central difference method requires two function evaluations to approximate the slope, as compared to one for the complex-step approach.

3.2 Second-Order System With Maximum-Value Cost Function

Consider the following second order nonlinear system taken from (Slotine and Li, 1991): [4]

$$\dot{x}_1 = x_2 \tag{29a}$$

$$\dot{x}_2 = \frac{1}{x_{\delta_1}} \left[-x_{\delta_2} x_1 |x_1| + u \right] \tag{29b}$$

where the bound for each uncertain parameter is as follows: $1 \leq x_{\delta_1} \leq 20$ and $-5 \leq x_{\delta_2} \leq 5$. The control input u is given by(Slotine and Li, 1991) [5]

$$u = -10\sqrt{5}(x_1 - 1) + x_2|x_2| - k\,\mathrm{sat}(s) \tag{30}$$

where $\mathrm{sat}(s)$ is given by

$$\mathrm{sat}(s) = \mathrm{sign}(s)\min(|s|, 1), \tag{31}$$

s is the sliding surface that is given by $s = x_2 + 10(x_1 - 1)$ and

$$k = \left(2 + \frac{\sqrt{5}}{10}\right) + 10\sqrt{5}\left(\sqrt{5} - 1\right)|x_1 - 1| \tag{32}$$

Note that, in order to use the complex-step method to generate a gradient approximation, all of the operations defined above must be implemented as shown in (Martins *et al.*, 2000) and (Martins *et al.*, 2001). For example,

[4] See Examples 7.1 and 7.4. Our example is slightly modified from the original: the bounds for the two uncertain parameters are changed so that the response has some overshoot.

[5] $\mathrm{sat}(\cdot)$ is used to avoid chattering in the control signal and the original reference trajectory was a time-varying signal but it is changed in our example to the unit step.

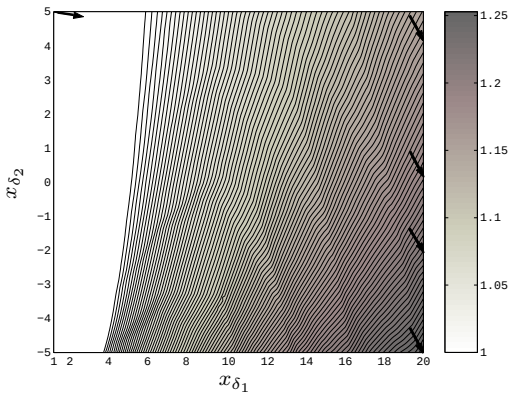

Fig. 3. The contour lines show the overshoot values with respect to x_δ. The arrows show the directions of the approximate gradients calculated by the complex-step method for each optimisation step starting from (1, 5). It converges to the solution in four steps

$$|z_1| = \begin{cases} z_1, & \text{for } a_1 \geq 0 \\ -z_1, & \text{for } a_1 < 0 \end{cases} \tag{33a}$$

$$\min(z_1, z_2) = \begin{cases} z_1, & \text{for } a_1 \leq a_2 \\ z_2, & \text{for } a_1 > a_2 \end{cases} \tag{33b}$$

$$\mathrm{sign}(z_1) = \begin{cases} 1, & \text{for } a_1 \geq 0 \\ -1, & \text{for } a_1 < 0 \end{cases} \tag{33c}$$

The cost function is given as

$$\max_{x_\delta \in \Delta} J(x_\delta) = \max_{t \in [0,\,10]} x_1(t) \tag{34}$$

Hence, the optimisation problem is to find the values of $(x_{\delta_1}, x_{\delta_2})$ where the overshoot of x_1 is a maximum. For the purposes of comparison, an exhaustive grid-based search was used to find an approximate global maximum for this problem. A gridding of 200 points for x_{δ_1} and 100 points for x_{δ_2} was used so that the interval magnitude for both parameters was 0.1. The total number of function evaluations for the grid-based search was therefore 20,000 and the global maximum was found to be $(x_{\delta_1}, x_{\delta_2}) = (20, -5)$.

The initial values of $(x_{\delta_1}, x_{\delta_2})$ input to the optimisation algorithm corresponded to the farthest point from the global solution, i.e., (1, 5). Using the complex-step method with the step size h equal to 10^{-50}, the optimisation converged to the solution (20.00, -5.00), exactly to the global solution. The number of function evaluations required was 63. Since the gradient around the initial guess is almost flat (see Figure 3), the optimisation algorithm using the finite difference-based gradient approximation failed to converge to the global solution, when using the best value for the step size h chosen by the function "fmincon" (in fact, it was immediately trapped at its initial value). As can be seen from Figure 3, however, the complex-step method is able to calculate a good approximation of the gradient even at the initial point of the optimisation, and is thus able to converge to the correct solution. Figure 4 shows the number of function evaluations for the two methods with

Table 2 (Example 2) No. of cost function evaluations & convergence results for FD (central finite difference), and CS (complex-step) with 220 different starting points

Method	Minimum	Maximum	Average	Standard Deviation	Success Rate
FD	15	1290	135.41	146.25	98.18%
CS	9	144	59.07	18.43	100%

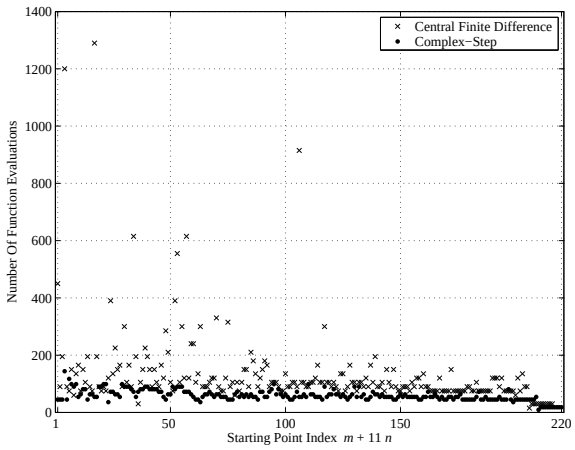

Fig. 4. The number of cost function evaluations for different starting points. The starting point index, $m + 11n$, corresponds to the starting point, $(x_{\delta_1}, x_{\delta_2})$ being equal to $(-5+m, 1+n)$, where $m = 1, 2, \ldots, 11$ and $n = 0, 1, \ldots, 19$. The total number of points is 220.

220 different starting points, where the incorrect convergent cases are removed and the step size h for the central finite difference is equal to 10^{-5}, which is chosen to improve the success rate and reduce the number of function evaluations. Clearly, the upper bound of the complex-step method is significantly less than that of the finite difference method. Table 2 summarizes the simulation results. In terms of the average number of function evaluations and the success rate, the complex-step method gives efficiency improvements of 56% and 2%, respectively, compared to the finite difference method.

4. CONCLUSIONS

In this paper, the complex-step perturbation method was extended to the setting of optimisation problems involving dynamical systems modelled as nonlinear differential equations. The method was applied to two robust performance analysis problems, and was shown to provide more accurate solutions and improved convergence times when compared with standard finite difference-based gradient approximation methods.

ACKNOWLEDGMENTS

This work was carried out under EPSRC research grant GR/S61874/01. The authors are grateful to Dr. Andres Marcos of the Control and Instrumentation Group at the University of Leicester for bringing the complex-step gradient approximation method to our attention.

REFERENCES

Cerviño, Laura I. and Thomas R. Bewley (2003). On the extension of the complex-step derivative technique to pseudospectral algorithms. *Journal Of Computational Physics* **187**, 544–549.

Fielding, Christopher, Andras Varga, Samir Bennani and Michiel Selier (Eds.) (2002). *Advanced Techniques for Clearance of Flight Control Laws*. Springer, Chapter 10.

Lyness, J. N. (1967). Numerical algorithm based on the theory of complex variables. In: *Proceeding ACM 22nd National Conference*. Thomson Book Co.. Washington DC.

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

Martins, Joaquim R. R. A., Ilan M. Kroo and Juan J. Alonso (2000). An automated method for sensitivity analysis using complex variables. In: *AIAA-2000-0689*. AIAA Aerospace Sciences Meeting & Exhibit. Reno, NV, USA.

Martins, Joaquim R. R. A., Peter Sturdza and Juan J. Alonso (2001). The connection between the complex-step derivative approximation and algorithmic differentiation. In: *AIAA-2001-0921*. AIAA Aerospace Sciences Meeting & Exhibit. Reno, NV, USA.

MathWorks (2003). *Optimization Toolbox (Version 2) For Use With MATLAB*. The MathWorks, Inc.

Slotine, Jean-Jacques E. and Weiping Li (1991). *Applied Nonlinear Control*. Prentice Hall, Inc., pp. 288-299.

Spall, James C. (2003). *Introduction To Stochastic Search And Optimization : Estimation, Simulation, and Control*. John Willey & Sun, 49-50.

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

Tierno, Jorge E., Richard M. Murray and John C. Doyle (1995). An efficient algorithm for performance analysis of nonlinear control systems. In: *American Control Conference*. Seattle, WA.