Complex-Step Derivative Approximations for Sensitivity Analysis

Ralph C. Smith
Department of Mathematics
North Carolina State University
Local Sensitivity Sensitivity Analysis

Motivation:

- Ascertain whether the model is robust or overly fragile with regard to various parameters;
- Determine whether the model can be simplified by fixing insensitive parameters;
- Specify regimes in the parameter space that optimally impact responses or their uncertainties;
- Guide experimental design to determine measurement regimes that have the greatest impact on parameter or response sensitivity.

Models:

\[ y = f(\theta) \]

\[ y = f(t, \theta) \]

\[ y_i = f(t_i, \theta) \]

\[ y_i = f(t_i, \theta) + \varepsilon_i \]

Notation:

- \( y \): Vector or scalar-valued response
- \( \theta \): Inputs; e.g., parameters, IC, BC
- \( t \): Independent variable; e.g., time
- \( \varepsilon_i \): Observation errors
**Complex-Step Derivative Approximation**

**Initial Approach:** Consider complex variable \( z = x + iy \) and function 
\[
f(z) = u(x, y) + iv(x, y). \]
For **analytic** \( f \), consider Cauchy-Riemann equations
\[
\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}
\]

For real \( h \),
\[
\frac{\partial u}{\partial x} = \lim_{h \to 0} \frac{v(x, y + h) - v(x, y)}{h}
\]
\[
= \lim_{h \to 0} \frac{\text{Im}[f(x + i(y + h))] - \text{Im}[f(x + iy)]}{h}
\]

Note: For real-valued \( f \),
\[
y = 0, \quad f(x) = u(x, 0), \quad v(x, 0) = \text{Im}[f(x)] = 0
\]

**Complex-Step Approximation:**
\[
f'(x) \approx \frac{\text{Im}[f(x + ih)]}{h}
\]
Complex-Step Derivative Approximation

Complex-Step Approximation:
\[ f'(x) \approx \frac{\text{Im}[f(x + ih)]}{h} \]

A Bit of History: Discussed in


Big Problem: Assumption of analyticity overly restrictive for simulation codes

Solution: For sufficiently smooth \( f \), consider
\[ f(x + ih) = f(x) + ihf'(x) - \frac{h^2}{2!} f''(x) - i \frac{h^3}{3!} f^{(3)}(x) + \mathcal{O}(h^4) \]

Complex-Step Approximation:
\[ f'(x) = \frac{\text{Im}[f(x + ih)]}{h} + \mathcal{O}(h^2) \]

Note:
- Reduces smoothness requirement
- Numerically demonstrated to be accurate up to points of discontinuity
Complex-Step Derivative Approximation

**Complex-Step Approximation:**

\[
f'(x) = \frac{\text{Im}[f(x + ih)]}{h} + \mathcal{O}(h^2)
\]

**A Bit More History:**

- Used to compute sensitivities for 3-D aero-structural models – verified using adjoints: [Martins et al, 2003].
- Noted by Tim Kelley in his green book.

**Notes:**

- Avoids subtractive cancellation and relatively insensitive to stepsize.
- May require modification of MATLAB functions such as `abs`, `min`, `max`.
- Structure very similar to forward-mode AD.
- Surprisingly robust and difficult to break!
Complex-Step Derivative Approximation

**Analytic Example:** Based on example in J.N. Lyness and C.B. Moler, 1967.

\[ f(x) = \frac{e^{2x}}{\sqrt{\sin^2 x + \cos^2(3x)}} \]

**Relative Errors:** Complex-step and finite-difference

\[ e_{rel} = \frac{|f' - f'_{exact}|}{|f'_{exact}|} \]
Complex-Step Derivative Approximation

**Example:** 3-parameter SIR model

\[
\frac{dS}{dt} = \delta (N - S) - \gamma IS, \quad S(0) = S_0
\]

\[
\frac{dl}{dt} = \gamma IS - (r + \delta)l, \quad l(0) = l_0,
\]

**Sensitivity Equations:**

\[
\frac{ds(t)}{dt} = \frac{\partial g}{\partial u} s(t) + \frac{\partial g}{\partial q}
\]

\[
s(t) = \left[ \frac{\partial S}{\partial \delta}, \frac{\partial l}{\partial \delta}, \frac{\partial S}{\partial \gamma}, \frac{\partial l}{\partial \gamma}, \frac{\partial S}{\partial r}, \frac{\partial l}{\partial r} \right]^T
\]

\[
s(t_0) = 0_{N \cdot p}
\]

\[
\frac{\partial g}{\partial u} = \begin{bmatrix} J & 0 & 0 \\ 0 & J & 0 \\ 0 & 0 & J \end{bmatrix}, \quad J = \begin{bmatrix} -\delta - \gamma l & -\gamma S \\ -\gamma l & \gamma S - (r + \delta) \end{bmatrix}
\]

\[
\frac{\partial g}{\partial q} = [(N - S), -l, -lS, lS, 0, -l]^T.
\]
Complex-Step Derivative Approximation

**Example:** 3-parameter SIR model

\[
\frac{dS}{dt} = \delta(N - S) - \gamma IS, \quad S(0) = S_0
\]

\[
\frac{dl}{dt} = \gamma IS - (r + \delta)l, \quad l(0) = l_0,
\]

**Results:** Representative parameter sensitivities
Complex-Step Derivative Approximation

**Results:** Representative initial condition sensitivities

$$\frac{\partial S}{\partial S_0}(t)$$

Notes:

- Complex-step required modification of two lines of code:
  
  $$S0\_complex = \text{complex}(S0,h); S\_S0 = \text{imag}(Y(:,1))/h;$$

- Accuracy dictated by accuracy of ODE solver.

- Significantly easier to code than sensitivity equations!
Complex-Step Derivative Approximation

**Euler-Bernoulli Model**: $0 < x < L$, $t > 0$

\[ \rho(x) \frac{\partial^2 w}{\partial t^2} + \gamma \frac{\partial w}{\partial t} + \frac{\partial^2 M}{\partial x^2} = f(t, x) \]

\[ M(t, x) = Yl(x) \frac{\partial^2 w}{\partial x^2} + cl(x) \frac{\partial^3 w}{\partial x^2 \partial t} \]

Here

\[ \rho(x) = \rho h b + \rho \rho h b \rho \chi p(x), \]

\[ Yl(x) = Yl_b + Yl_p \chi p(x) , \]

\[ cl(x) = cl_b + cl_p \chi p(x) \]

**Observation Model**:

\[ y_i = f(t_i, \theta) + \varepsilon_i \]

\[ f(t_i, \theta) = w(t_i, \bar{x}, \theta) \]

**Parameters**:

\[ \theta = [Yl_b, cl_b, \gamma, k_p, \rho_p, Yl_p, cl_p] \]
Complex-Step Derivative Approximation

**Goal:** Approximate $s_{CS}(t) = \frac{\partial w(t)}{\partial Y_l}$ Stiffness coefficient

Finite Difference: $h = 1 \times 10^{-4}$
Complex-Step: $h = 1 \times 10^{-16}$

**Note:**
- Complex-step accuracy relatively insensitive to stepsize.
- Difficult to compute reasonable finite-difference sensitivity.
Complex-Step Derivative Approximation

**Goal:** Approximate $s_{CS}(t) = \frac{\partial w(t)}{\partial \gamma}$ Air damping

**Notes:**
- Most affected by primary mode
- Could not get reasonable approximation using finite-differences
Automatic Differentiation

**Approach:** Apply chain rule to elementary arithmetic operations and functions

**Example:** Differentiate $f(x_1, x_2) = x_1 x_2^2$ with respect to $x_2$

<table>
<thead>
<tr>
<th>Forward AD</th>
<th>Complex-Step Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta x_1 = 0$</td>
<td>$h_1 = 0$</td>
</tr>
<tr>
<td>$\Delta x_2 = 1$</td>
<td>$h_2 = 10^{-16}$</td>
</tr>
<tr>
<td>$f(x_1, x_2) = x_1 x_2^2$</td>
<td>$f = (x_1 + i h_1) (x_2 + i h_2)^2$</td>
</tr>
<tr>
<td>$\Delta f = \Delta x_1 x_2^2 + 2 x_1 x_2 \Delta x_2$</td>
<td>$f = [x_1 (x_2 - h_2^2) - 2 x_2 h_1 h_2] + i [h_1 (x_2^2 - h_2^2) + 2 x_1 x_2 h_2]$</td>
</tr>
<tr>
<td>$\frac{\partial f}{\partial x_2} = \Delta f = 2 x_1 x_2$</td>
<td>$\frac{\partial f}{\partial x_2} = \frac{\text{Im}[f]}{h_2} = 2 x_1 x_2$</td>
</tr>
</tbody>
</table>
Concluding Remarks

Advantages:

• Easy to code.
• Provides second-order accuracy with one function evaluation and relatively insensitive to stepsize.
• Avoids solving coupled sensitivity equations for evolution models.
• Numerical tests demonstrate accuracy up to discontinuities for several applications.
• Close ties to forward AD but does not require AD architectures.

Disadvantages:

• Can fail without numerical warning for problems with insufficient regularity.
• Requires p model evaluations for p inputs. Can be a problem for high-dimensional problems.
• Requires modification of certain functions such as \( \text{abs}, \text{min}, \text{max} \).
• Does not run automatically in MATLAB pde toolbox.