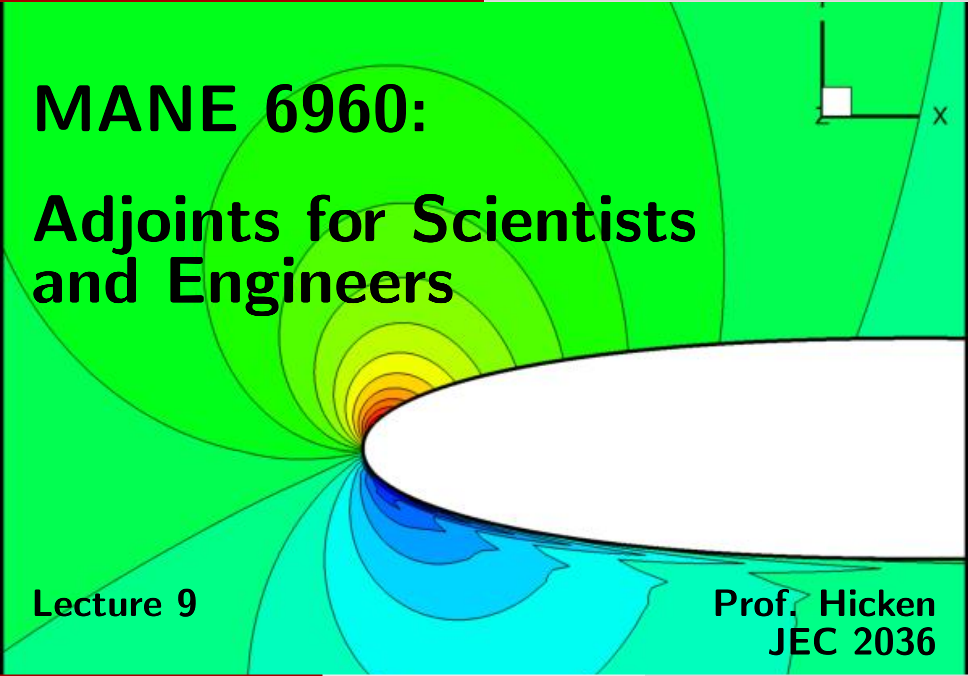


MANE 6960:

Adjoints for Scientists and Engineers

Lecture 9

Prof. Hicken
JEC 2036

Computing Derivatives:

Finite-Difference Approximations

Forward-Difference Approximation

The class of finite-difference approximations use Taylor's theorem to construct difference formulae that approximation the derivative.

- The forward-difference approximation is the simplest and most commonly used finite-difference method in optimization

Forward-Difference Approximation (cont.)

The forward-difference approximation is easy to derive.

- suppose we want the j^{th} partial derivative of f, that is $\partial f / \partial x_j$
- apply Taylor's theorem at an arbitrary point $x \in \mathbb{R}^n$ in the direction e_j.
- e_j is the (unit) basis vector for the j^{th} variable, which picks out the partial derivative we want; it is given by

$$e_j \equiv \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}^T$$

Forward-Difference Approximation (cont.)

We get

$$f(x + he_j) = f(x) + h\frac{\partial f}{\partial x_j} + \frac{h^2}{2}\frac{\partial^2 f}{\partial x_j^2}(x + \bar{h}e_j)$$

- $h > 0$ denotes the step size we have control of
- $\bar{h} \in (0, h)$ is required by Taylor's theorem

Rearranging the above expression we obtain the (first-order) forward-difference approximation.

Forward-Difference Approximation (cont.)

Definition: Forward-Difference Approximation

The forward-difference approximation of the partial derivative $\partial f/\partial x_j$ of the function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is given by

$$\frac{f(x + he_j) - f(x)}{h} = \frac{\partial f}{\partial x_j} + \underbrace{L_{\mathrm{FD}} h}_{\text{error}}$$

where $h > 0$ is the step size, e_j is the j^{th} Cartesian basis vector, and L_{FD} is a constant that does not depend on h.

- Note that the error is proportional to h.

Exercise

Compute the partial derivative $\partial f / \partial x_1$ of

$$f(x) = \frac{3}{2}x_1^2 + x_2^2 - 2x_1x_2 + 2x_1^3 + \frac{1}{2}x_1^4$$

at $x = (-1, -1)^T$ using the forward-difference approximation.

- Try step sizes $h = 10^{-2}$, $h = 10^{-7}$, and $h = 10^{-20}$
- Compare your estimates to the true derivative

The Effect of Round-off Errors

The error in the forward-difference approximation, $L_{\mathrm{FD}} h$, suggests that we make h small, but this failed in the previous exercise.

- This failure is not unique
- The problem is round-off error

Computers can only represent a finite number of digits of $f(x)$ accurately.

- For IEEE double precision arithmetic, the relative error in approximating $f(x)$ is $\epsilon_{\mathrm{mach}} \approx 10^{-16}$.

The Effect of Round-off Errors (cont.)

Let $\tilde{f}(x)$ denote the computed value of $f(x)$ (i.e. the exact value) and $\tilde{f}(x + he_j)$ the computed value of $f(x + he_j)$. Then[1]

$$\begin{aligned} & |\tilde{f}(x) - f(x)| \leq \epsilon_{\mathrm{mach}} L_f \\ \text{and} \qquad & |\tilde{f}(x + he_j) - f(x + he_j)| \leq \epsilon_{\mathrm{mach}} L_f \end{aligned}$$

where $|f(x)| \leq L_f$ for all x in the region of interest.

If we substitute these bounds into the forward-difference formula, we get the following error estimate:

$$\mathrm{Error}_{\mathrm{FD}} \approx L_{\mathrm{FD}} h + \frac{2\epsilon_{\mathrm{mach}} L_f}{h}$$

[1] This analysis is based on [NW06, pg. 196]

The Effect of Round-off Errors (cont.)

We want the value of h that minimizes this error; we know how to do that!

Taking the derivative of our estimate of Error_{FD} with respect to h and setting the result to zero, we find

$$h^* = \sqrt{\frac{2L_f \epsilon_{\text{mach}}}{L_{\text{FD}}}} \approx \sqrt{\epsilon_{\text{mach}}}$$

- For double precision floating point numbers, $h^* \approx 10^{-8}$.

The Effect of Round-off Errors (cont.)

Thus, the error in the finite-difference approximation cannot be made zero.

This can be partially ameliorated by using a more accurate finite-difference approximation...

Central-Difference Approximation

Definition: Central-Difference Approximation

The central-difference approximation of the partial derivative $\partial f / \partial x_j$ of the function $f : \mathbb{R}^n \to \mathbb{R}$ is given by

$$\frac{f(x + he_j) - f(x - he_j)}{2h} = \frac{\partial f}{\partial x_j} + \underbrace{L_{\text{CD}} h^2}_{\text{error}}$$

where $h > 0$ is the step size, e_j is the j^{th} Cartesian basis vector, and L_{CD} is a constant that does not depend on h.

- The optimal step size in this case is $h^* \approx \sqrt[3]{\epsilon_{\text{mach}}}$.

Computational Cost

In addition to accuracy issues, finite-difference methods have a significant computational cost: to evaluate ∇f when $x \in \mathbb{R}^n$,

- the forward-difference approximation requires n function evaluations, not including the unperturbed value $f(x)$;
- the central-difference approximation requires $2n$ function evaluations.

Pros & Cons of Finite-Difference Approximations

- ✓ easy to implement
- ✓ can be applied to almost any “black-box” function
- ✗ accuracy can be an issue, especially for badly scaled problems; also, choosing h^* may be difficult for multiple inputs
- ✗ computational cost scales with the $\#$ of inputs

Computing Derivatives: Complex-Step Approximation

Refresher: Complex/Imaginary Numbers

Before we start, here's a (very) brief refresher on complex numbers

- $i = \sqrt{-1}$.
- A general complex number can be written as $z = x + iy$, where $x, y \in \mathbb{R}$.
- A complex number can be thought of as a point in the plane.

The Complex-Step Method

If $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a function of real variables, and it is complex differentiable — which implies the function is complex analytic — on the domain of interest, then

$$\begin{aligned} f(x + ihe_j) &= f(x) + ih\frac{\partial f}{\partial x_j} + \frac{i^2 h^2}{2}\frac{\partial^2 f}{\partial x_j^2} + \frac{i^3 h^3}{6}\frac{\partial^3 f}{\partial x_j^3} + \cdots \\ &= f(x) + ih\frac{\partial f}{\partial x_j} - \frac{h^2}{2}\frac{\partial^2 f}{\partial x_j^2} - \frac{ih^3}{6}\frac{\partial^3 f}{\partial x_j^3} + \cdots \end{aligned}$$

Taking the imaginary part of this equation, and rearranging gives...

The Complex-Step Method (cont.)

Definition: Complex-Step Approximation [ST98]

The complex-step approximation of the partial derivative $\partial f/\partial x_j$ of the function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is given by

$$\frac{\Im[f(x + ihe_j)]}{h} = \frac{\partial f}{\partial x_j} + \underbrace{L_{\text{CS}} h^2}_{\text{error}}$$

where $h > 0$ is the step size, e_j is the j^{th} Cartesian basis vector, and L_{CS} is a constant that does not depend on h.

- $\Im$ means "get the coefficient mutiplying i"

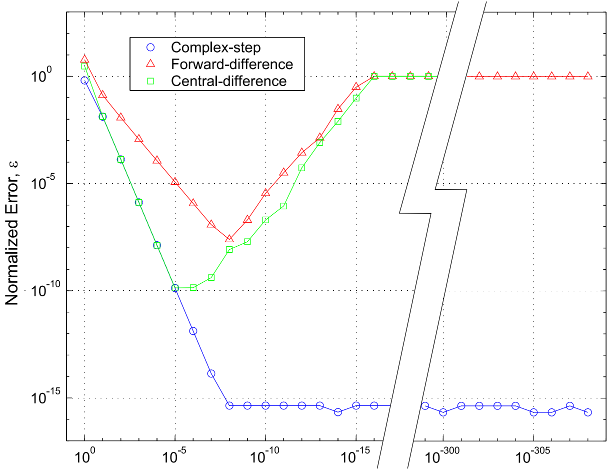
The Complex-Step Method (cont.)

This is cool for the following reason:

- the complex-step formula does not involve differences of function values;
- therefore, there is no subtractive cancellation due to round-off;
- therefore, we can make h as small as we want and make the error disappear!

OK, there is a limit to how small we can make h, because there is a smallest exponent allowed on computers, namely $h > 2.22 \times 10^{-308}$.

The Complex-Step Method (cont.)



Relative error in the derivative vs. decreasing step size

Complex-Step Pitfalls

While the complex-step method is straightforward to apply, there are a couple things to look out for [MSA03]:

- need to redefine `min`, `max`, and `abs`
- some trig and inverse trig functions may need to be redefined
- in Matlab and Julia, use the `transpose` function or the `.'` operator for transposing vectors and matrices, otherwise you get the conjugate transpose.

Let's look at the `abs` function as an example.

Complex-Step Pitfalls (cont.)

In many programming languages, abs is defined as the modulus of the complex number, i.e. $|z| = \sqrt{x^2 + y^2}$.

- This is not what we want.

Instead, we can define a new function:

```
function cabs(z)
  % complexified version of the absolute value
  if real(z) >= 0
    return z
  else
    return -z
  end
end
```

Pros & Cons of the Complex-Step Approximation

- ✓ easy to implement
- ✓ can be applied to almost any "black-box" function that accepts complex variables
- ✓ many open-source codes can be easily adapted to use complex-step.
- ✗ computational cost still scales with the # of input variables

Exercise

Compute the partial derivative $\partial f/\partial x_1$ of

$$f(x) = \frac{3}{2}x_1^2 + x_2^2 - 2x_1x_2 + 2x_1^3 + \frac{1}{2}x_1^4$$

at $x = (-1, -1)^T$ using the complex-step approximation.

- Try step sizes $h = 10^{-2}$, $h = 10^{-7}$, and $h = 10^{-20}$
- Compare your estimates to the true derivative

References

[MSA03] Joaquim R. R. A. Martins, Peter Sturdza, and Juan J. Alonso, *The complex-step derivative approximation*, ACM Transactions on Mathematical Software **29** (2003), no. 3, 245–262.

[NW06] J. Nocedal and S. J. Wright, *Numerical Optimization*, second ed., Springer–Verlag, Berlin, Germany, 2006.

[ST98] William Squire and George Trapp, *Using complex variables to estimate derivatives of real functions*, SIAM Review **40** (1998), no. 1, 110–112.