Lecture 1

Yuichiro Waki

July 24, 2007

5 Steps for Program Development

(1) State the problem clearly
(2) Inputs/outputs
(3) Design the algorithm
(4) Write a program
(5) Test the program

Problem

- Consider a partial derivative of a function $f : R \rightarrow R$ at point $x \in R$.
- Approximate it by $\frac{f(x+h)-f(x)}{h}$ for some given h. ("One-sided derivatives")
- h is called a "step-size".
- Cf. $\frac{f(x+h)-f(x-f)}{2h}$ is called "two-sided derivatives."
- Write a subroutine that implements this.

Inputs/outputs

- Inputs: function name, x, h
- Output: $\frac{f(x+h)-f(x)}{h}$

Algorithm

You need to evaluate a given function at a given point. You can use a built-in function `feval`.

Write a program: deriv.m

```
function fprime = deriv(fname, x, h)

% This subroutine calculates one-sided derivative
%of a given function at given point x, given a
%step-size h.
%
% fname : name of the function (character strings)
% x     : point of evaluation
% h     : step-size
%
% fprime : one-sided approximation of derivative

fprime = (feval(fname, x+h) - feval(fname, x))/h;
```

Test the program

Create a *test function*. One of the obvious choice is a linear function.

```
function y = test1(x)

y = x;
```

Test the program

Examine the following:

- `deriv('test1',1,10^-5)`
- `deriv('test1',1,10^-10)`
- `deriv('test1',1,10^-20)`

Test the program

- What happens when $h = 10^{-20}$???
- To understand this, you need to know the computer arithmetics and roundoff errors.

Number representation in MATLAB

- 64 binary digits
- $sc_1c_2...c_{11}f_1f_2...f_{52}$
- $s, c_1, c_2, ..., c_{11}, f_1, f_2, ..., f_{52} \in \{0, 1\}$
- Each number in the computer has the following representation: $x = (-1)^s 2^{c-1023}(1+f)$, where $c = \sum_{i=1}^{11} c_i 2^{11-i}$ and $f = \sum_{i=1}^{52} f_i (\frac{1}{2})^i$
- c is called the characteristic/exponent
- f is called the mantissa

Roudoff errors

Imagine (hypothetically) numbers in the computer have the following representation:
$x = \pm 0.d_1 d_2 d_2 ... d_K \times 10^n$, where $d_i \in \{0, 1, ..., 9\}$ and $n \in \{-N, ..., N\}$.
Suppose you want to add h to x. If $x/h > 10^K$, h is too small to change the value of x in the computer when added.
Even when h is not that small relative to x, some digits are lost when added. Errors due to this type of loss of information is called the roundoff errors.

Back to numerical derivatives

- What is often done is to set the actual step-size to $h \max(1, |x|)$. Also, don't set h too small.
- Incorporate such an adjustment into your subroutine and test the program again.