Introduction to MATLAB Software (2)

FuSen Lin

Department of Computer Science and Engineering
National Taiwan Ocean University

Scientific Computing, Fall 2011

§ 1.4 Errors

Absolute and Relative Error

- *Definition*: If $\tilde{x}$ approximates a scalar x, then the **absolute error** is given by

$$\text{Abs.Err.} = |\tilde{x} - x|$$

- The **relative error** is given by

$$\text{Rel.Err.} = \frac{|\tilde{x} - x|}{|x|}, \quad x \neq 0$$

- An Example: The **Stirling Formula**:

$$S_n = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n, \quad e = exp(1).$$

Talor Approximation

- The partial sum of the exponential function $\exp(x)$ satisfy

$$e^x = \sum_{k=0}^{n} \frac{x^k}{k!} + \frac{e^\eta}{(n+1)!} x^{n+1}$$

for some η between 0 and x. This is the Talor polynomial of $\exp(x)$ about 0.

Rounding Errors

- The Rounding Errors arise by the computer arithmetic, which is called floating point arithmetic.
- Numerical computation involves working with an inexact computer arithmetic system.
- An example: to compute the values of the polynomial

$$p(x) = x^6 - 6x^5 + 15x^4 - 20x^3 + 15x^2 - 6x + 1$$

  for the smaller neighborhoods around $x = 1$.
- *Algorithms that are equivalent mathematically may behave very differently numerically.*

A Floating Point System

- The numbers in a **floating point system** are defined by a base β, a mantissa length t, and exponent range $[L, U]$.
- A nonzero floating point number has the form

$$x = .b_1 b_2 \cdots b_t \times \beta^e.$$

  Here $.b_1 b_2 \cdots b_t$ is the mantissa and e is the exponent, which satisfies $L < e \leq U$. The b_i are base-β digits and satisfy $0 \leq b_i \leq \beta - 1$. It is **normalized** if $b_1 \neq 0$.
- The set of floating point numbers is *finite* and their spacing is *not uniform*.

IEEE Standard 754-1985

- MATLAB adopted The **IEEE Standard Binary Floating Point Arithmetic**–*double precision.*
- The normalized double precision numbers require a 64-bits representation:

s(sign):1 bit	c(exponent):11 bits	f(normalized mantissa):52 bits

- They have the form

$$(-1)^s \times 2^{c-1023} \times (1+f)_2$$

§ 1.5 Designing Functions

The general structure of a MATLAB function

- **function**[*Output Parameter*] $=<$ *Name of Function* $>$ $(<$ *Input Parameters* $>)$
- %
  % $<$ Comment that completely specify the function$>$
  %
- $$< Function \quad Body >$$
- Example: Write a MATLAB *function* to compute the approximation of $ln(a)$ by the Taylor series of $ln(1+x)$ (Hint: the input $x = 1 - a$).

Matlab codes for Taylor Series of *ln*(*a*)

```
function y = logsrs1(x, n)

% Date: 3/12/2001,  Fusen F. Lin
% This function computes log(a) by the series,
% log(1+x)= x-x^2/2+x^3/3-x^4/4+x^5/5-..., for n terms.
% Input : real x and integer n (x=1-a for log(a)).
% Output: the desired value log(1+x).

tn = x;                          % The first term.
sn = tn;                         % The n-th partial sum.
for k = 1:1:n-1,                 % To sum the series.
   tn = -tn*x*k/(k+1);           % compute it recursively.
   sn = sn + tn;
end
y = sn;                          % Output the final sum.
```

Computing the Exponential of a Vector

- Consider once again the Talor approximation

$$e^x \approx T_n(x) = \sum_{k=0}^{n} \frac{x^k}{k!}$$

  to the Exponential $\exp(x)$ or e^x.
- Use function call with *scalar level* and *vector level* to approximate the values of e^x.

Numerical Differentiation

- Suppose $f(x)$ is a differentiable function we wish to approximate whose derivative at $x = a$.
- A Taylor series expansion about this point says that

$$f(a+h) = f(a) + f'(a)h + \frac{f''(\eta)}{2}h^2$$

for some $\eta \in [a, a+h]$. Thus,

$$D_h = \frac{f(a+h) - f(a)}{h}$$

provides increasingly good approximations as h gets small.
- For example, if $f(x) = \sin(x)$, to find the derivative of sine at $a = 1$.

The Loss of Accuracy

- The **error bound** is

$$|D_h - f'(x)| \leq \frac{h}{2}|f''(\eta)|$$

- The error in the computation of the numerator of D_h is magnified by $1/h$.
- A heuristic bound is

$$|D_h - f'(x)| \approx \frac{h}{2}|f''(\eta)| \pm \frac{2\text{eps}}{h},$$

which are the **truncation error** due to calculus and the computation error due to **roundoff error**.
- This quantity is ***minimized*** when $h = 2\sqrt{\text{eps}/|f''(\eta)|}$.

Write a MATLAB Function

- Write a **function** to do numerical differentiation: The input parameters will include:
  - The name of the function f that is to be differentiated
  - The point of differentiation a
  - Information about $|f''(\eta)|$
  - Information about the accuracy of the computed f-evaluations
- Suppose that $|f''(\eta)| \leq M_2$ and the *absolute error* in a computed function evaluation is bounded by δ. Then the best choice of h is $h = 2\sqrt{\delta/M_2}$.

Homework 1

- Work on the problems: P.1.2.7, P.1.3.1, P.1.4.2, P.1.5.2, and P.1.5.3

§ 1.6 Structure Arrays and Cell Arrays

- 1.6.1 Three-Digit Arithmetic
- 1.6.2 Pade Approximants

Structure Arrays and Cell Arrays

- To use appropriate *data structures* is very important for programmers.
- Two ways for **Advanced data structures** in MATLAB: *Structure Arrays* and *Cell Arrays*.
- A *structure array* has fields and values (see **struture.m**).
- An example: a **geodesy application** where latitudes and longitudes are measured in degree, minutes, seconds. The field values are accessed with **'dot'** notation.
- A **cell array** is basically a matrix in which a given entry can be a matrix, a structure array, or cell array.
- If m and n are positive integers, then

$$C = \text{cell}(m, n)$$

Design Three-Digit Arithmetic

- **Structures and Strings** are nicely reviewed by developing a three-digit, base-10 *floating point arithmetic simulation* package.
- Assuming that the exponents range is $[-9, 9]$ and use a 4-field structure to represent each floating point number (see **Represent.m**).
- Need to *convert* the operands to conventional form, do the *arithmetic operations*, and then *represent the result* in 3-digit form.
- An example for estimating the Euler constant.

Pade Approximants

- A useful class of Approximation methods for exponential function e^z are the **Pade functions** defined by

$$R_{pq}(z) = \left(\sum_{k=0}^{p} \frac{(p+q-k)!p!}{(p+q)!k!(p-k)!} z^k \right) / \left(\sum_{k=0}^{q} \frac{(p+q-k)!q!}{(p+q)!k!(q-k)!} (-z)^k \right).$$

§ 1.7 More Refined Graphics

Fonts

- A font has a name, a size, a style.
- MATLAB's fonts has Time-Roman, AvantGarde, Bookman, Courier, Helvetica, Helvetica-Narrow, NewCenturySchlbk, Palatino, Zapfchancery.
- It is better to use **title**, **xlabel**, and **ylabel** with proper fonts.

Mathematical Typesetting

- It is possible to specify subscripts, superscripts, Greek letters, and various mathematical symbols in the strings that are passed to **title, xlabel , ylabel, and text**.

Text Placement

- Using **HorizontalAlignment** and **VerticalAlignment** with suitable modifiers.

Line Width and Axes

- An example:
  h = plot(x, y);
  set(h, 'LineWidth', 3)
  (see p.67).
- Legends and Color: see p. 69.
- Any operations of refined graphs can be done in figure windows.