FF505

Computational Science

Miscellanea

Marco Chiarandini (marco@imada.sdu.dk)

Department of Mathematics and Computer Science (IMADA)
University of Southern Denmark

Outline

Outline

Demos

Try!

```
demo 'matlab'
```

Order of Operations in MatLab

1. parenthesis, from innermost
2. exponentiation, from left to right
3. multiplication and division with equal precedence, from left to right
4. addition and subtraction with equal precedence, from left to right

```
>>4^2-12-8/4*2
ans =
    0
>>4^2-12-8/(4*2)
ans =
    3
>> 3*4^2 + 5
ans =
    53
>>(3*4)^2 + 5
ans =
    149
```

```
>>27^(1/3) + 32^(0.2)
ans =
    5
>>27^(1/3) + 32^0.2
ans =
    5
>>27^1/3 + 32^0.2
ans =
    11
```

Useful Functions

MATLAB -> functions

```
% max (or min)
a = [1 15 2 0.5]
val = max(a)
[val,ind] = max(a)

% find
find(a < 3)
A = magic(3) %N-by-N matrix
      constructed from the integers 1
      through N^2 with equal row, column,
       and diagonal sums.
[r,c] = find(A>=7)

% sum, prod
sum(a)
prod(a)
floor(a) % or ceil(a)
max(rand(3),rand(3))
max(A,[],1)
min(A,[],2)
A = magic(9)
sum(A,1)
sum(A,2)
```

```
% pseudo-inverse
pinv(A) % inv(A'*A)*A'

% check empty e=[]
isempty(e)
numel(A)
size(A)
prod(size(A))
```

```
sort(4:-1:1)
sort(A) % sorts the columns
```

Useful Functions

Working with polynomials:

$$f(x) = a_1 x^n + a_2 x^{n-1} + a_3 x^{n-2} + \ldots + a_{n-1} x^2 + a_n x + a_{n+1}$$

is represented in MATLAB by the vector

$$[a_1, a_2, a_3, \ldots, a_{n-1}, a_n, a_{n+1}]$$

```
help polyfun
r=roots([1,-7,40,-34]) % x^3-7x^2+40x-34
poly(r) % returns the polynomial whose roots are r
roots(poly(1:20))
poly(A) % coefficients of the characteristic polynomial, det(lambda*EYE(SIZE(A)) - A)
```

Common Mathematical Functions

Exponential	
`exp(x)`	Exponential; e^x.
`sqrt(x)`	Square root; $\sqrt{x}$.
Logarithmic	
`log(x)`	Natural logarithm; $\ln x$.
`log10(x)`	Common (base-10) logarithm; $\log x = \log_{10} x$.
Complex	
`abs(x)`	Absolute value; x.
`angle(x)`	Angle of a complex number x.
`conj(x)`	Complex conjugate.
`imag(x)`	Imaginary part of a complex number x.
`real(x)`	Real part of a complex number x.
Numeric	
`ceil(x)`	Round to the nearest integer toward ∞.
`fix(x)`	Round to the nearest integer toward zero.
`floor(x)`	Round to the nearest integer toward $-\infty$.
`round(x)`	Round toward the nearest integer.
`sign(x)`	Signum function: $+1$ if $x > 0$; 0 if $x = 0$; -1 if $x < 0$.

Common Mathematical Functions

Trigonometric*	
`cos(x)`	Cosine; $\cos x$.
`cot(x)`	Cotangent; $\cot x$.
`csc(x)`	Cosecant; $\csc x$.
`sec(x)`	Secant; $\sec x$.
`sin(x)`	Sine; $\sin x$.
`tan(x)`	Tangent; $\tan x$.
Inverse trigonometric†	
`acos(x)`	Inverse cosine; $\text{arccos}\, x = \cos^{-1} x$.
`acot(x)`	Inverse cotangent; $\text{arccot}\, x = \cot^{-1} x$.
`acsc(x)`	Inverse cosecant; $\text{arccsc}\, x = \csc^{-1} x$.
`asec(x)`	Inverse secant; $\text{arcsec}\, x = \sec^{-1} x$.
`asin(x)`	Inverse sine; $\arcsin x = \sin^{-1} x$.
`atan(x)`	Inverse tangent; $\arctan x = \tan^{-1} x$.
`atan2(y,x)`	Four-quadrant inverse tangent.

*These functions accept x in radians.

†These functions return a value in radians.

Common Mathematical Functions

Hyperbolic	
`cosh(x)`	Hyperbolic cosine; $\cosh x = (e^x + e^{-x})/2$.
`coth(x)`	Hyperbolic cotangent; $\cosh x/\sinh x$.
`csch(x)`	Hyperbolic cosecant; $1/\sinh x$.
`sech(x)`	Hyperbolic secant; $1/\cosh x$.
`sinh(x)`	Hyperbolic sine; $\sinh x = (e^x - e^{-x})/2$.
`tanh(x)`	Hyperbolic tangent; $\sinh x/\cosh x$.
Inverse hyperbolic	
`acosh(x)`	Inverse hyperbolic cosine
`acoth(x)`	Inverse hyperbolic cotangent
`acsch(x)`	Inverse hyperbolic cosecant
`asech(x)`	Inverse hyperbolic secant
`asinh(x)`	Inverse hyperbolic sine
`atanh(x)`	Inverse hyperbolic tangent

Multidimensional Arrays

Consist of two-dimensional matrices layered to produce a third dimension. Each layer is called a page.

```
cat(2,A,B) % is the same as [A,B].
cat(1,A,B) % is the same as [A;B].
```

```
>> A = magic(3); B = pascal(3);
>> C = cat(4,A,B) %concatenate matrices along DIM

C(:,:,1,1) =

    8 1 6
    3 5 7
    4 9 2

C(:,:,1,2) =
    1 1 1
    1 2 3
    1 3 6
```

Outline

Debugging

Two types of errors: (i) syntax errors (ii) runtime errors

- test on small examples whose result can be verified by hand
- display intermediate calculations
- use the debugger (F12 sets a breakpoint, F5, F10 to continue)
- modus tollens/pones (remove/add pieces of code)

```
a = magic(3);
%{
sum(a)
diag(a)
sum(diag(a))
%}
sum(diag(fliplr(a)))
```

Programming Style

- Document your scripts:
  - author and date of creation
  - what the script is doing
  - which input data is required
  - the function that the user has to call
  - definitions of variables used in the calculations and units of measurement for all input and all output variables!
- Organize your script as follows:
  1. input section (input data and/or input functions)
     Eg: `x=input("give me a number")`, `input("enter a key",'s')`
  2. calculation section
  3. output section (functions for displaying the output on the screen or files)
     Eg: `display(A)`, `display("text")`

Example


```
% Program M3eP32.m
% Program Falling_Speed.m: plots speed of a falling object.
% Created on March 1, 2009 by W. Palm III
%
% Input Variable:
% tfinal = final time (in seconds)
%
% Output Variables:
% t = array of times at which speed is computed (seconds)
% v = array of speeds (meters/second)
%
% Parameter Value:
g = 9.81; % Acceleration in SI units
%
% Input section:
tfinal = input('Enter the final time in seconds:');
%
% Calculation section:
dt = tfinal/500;
t = 0:dt:tfinal; % Creates an array of 501 time values.
v = g*t;
%
% Output section:
plot(t,v),xlabel('Time (seconds)'),ylabel('Speed (meters/second)')
```

Recall: Getting Help

- `help funcname`: Displays in the Command window a description of the specified function `funcname`.
- `lookfor topic`: Looks for the string topic in the first comment line (the H1 line) of the HELP text of all M-files found on MATLABPATH (including private directories), and displays the H1 line for all files in which a match occurs.
  Try: `lookfor imaginary`
- `doc funcname`: Opens the Help Browser to the reference page for the specified function `funcname`, providing a description, additional remarks, and examples.

Documentation

Effective documentation can be accomplished with the use of

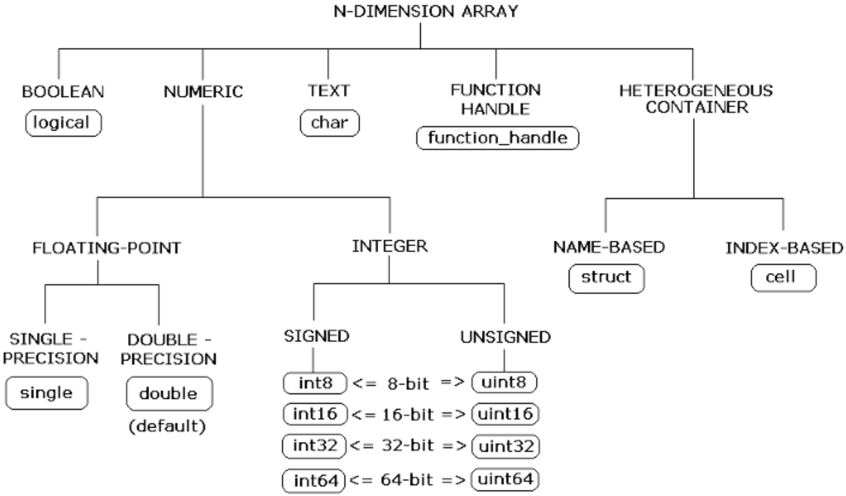
- Proper selection of variable names to reflect the quantities they represent.
- Use of comments within the program.
- Use of structure charts.
- Use of flowcharts.
- A verbal description of the program, often in pseudocode (uses structural conventions of a programming language, but is intended for human reading rather than machine reading. Pseudocode typically omits details, replacing them with natural language)

More Guidelines on Style

More https://sites.google.com/site/matlabstyleguidelines

Outline

Data Types


N-DIMENSION ARRAY
BOOLEAN
logical
NUMERIC
TEXT
char
FUNCTION HANDLE
function_handle
HETEROGENEOUS CONTAINER
FLOATING-POINT
INTEGER
NAME-BASED
struct
INDEX-BASED
cell
SINGLE - PRECISION
single
DOUBLE - PRECISION
double
(default)
SIGNED
UNSIGNED
int8 <= 8-bit => uint8
int16 <= 16-bit => uint16
int32 <= 32-bit => uint32
int64 <= 64-bit => uint64

Further Functions

- Numerical integration: `trapz` (trapezoidal integration), `polyint`
- Numerical Differentiation: `diff(y)./diff(x)`, `polyder`, `gradient`
- Ordinary Differential Equations: `ode45`, `ode15s`
- Fourier Transform: `fft`. `fftgui`

Outline

Random Generators

How can we generate random numbers in computers?
We cannot, we generate pseudo-random numbers.

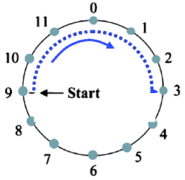
Modular Arithmetic

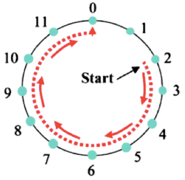
Modulo function: returns the reminder of division between two natural numbers: $x = a \cdot m + r$

Examples:

$$9 \bmod 8 = 1$$
$$16 \bmod 8 = 0$$
$$(9+6) \bmod 12 = 15 \bmod 12 = 3$$
$$(6 \cdot 2 + 15) \bmod 12 = 27 \bmod 12 = 3$$
$$1143 \bmod 1000 = 143$$

```
>> mod(9,8)
>> mod(16,8)
>> mod(9+6,12)
>> mod(6*2+15,12)
>> mod(27,12)
>> mod(1143,1000)
```





Given a, c, m and x_0:

$$
\begin{aligned}
x_1 &:= (a \cdot x_0 + c) \bmod m, \\
x_2 &:= (a \cdot x_1 + c) \bmod m, \\
x_3 &:= (a \cdot x_2 + c) \bmod m, \\
x_4 &:= (a \cdot x_3 + c) \bmod m, \\
&\vdots \\
x_{i+1} &:= (a \cdot x_i + c) \bmod m
\end{aligned}
$$

Setting $a = 5, c = 1, m = 16$ and the seed $x_0 = 1$

$$
\begin{aligned}
x1 &:= (5 \cdot 1 + 1) \bmod 16 = 6, \\
x2 &:= (5 \cdot 6 + 1) \bmod 16 = 15, \\
x3 &:= (5 \cdot 15 + 1) \bmod 16 = 12, \\
x4 &:= (5 \cdot 12 + 1) \bmod 16 = 13, \\
x5 &:= (5 \cdot 13 + 1) \bmod 16 = 2, \\
x6 &:= (5 \cdot 2 + 1) \bmod 16 = 11, \\
&\vdots
\end{aligned}
$$

- Sequence of numbers that approximates the properties of random numbers
- Deterministic and periodic behaviour
- sequence not truly random, but completely determined by a relatively small set of initial values, called the PRNG's state, which includes a truly random seed
- By varying a, c, m we can reach a full period of length m.

Characteristics of good generators:

- long period
- uniform unbiased distribution
- uncorrelated (time series analysis)
- efficient

Mersenne Twister is the default algorithm

search "seed" in the Help. Changing random number generator syntax

Summary

Topics common to main computational environments for Math, Physics and Engineering:

- Matrix calculations
- Graphics to visualize data
- Programming main structures (control flow, functions)

Take away messages:

- A computational environment like Matlab may be very helpful. Keep it alive, use it, experiment, try things.
- Develop skills to search things by your own. No way to remember everything. But helpful to know about existence. Look at User's Guide. Remember how things are called.

Outline

Floating-Point Numbers

A floating-point number in base b is a number of the form

$$\pm\left(\frac{d_1}{b}+\frac{d_2}{b^2}+\ldots+\frac{d_t}{b^t}\right)\times b^e$$

where $t, d_1, d_2, \ldots, d_t, b, e$ are all integers and

$$0 \le d_i \le b-1 \quad i=1,\ldots,t$$

- t refers to the number of digits and depends on the word length of the computer.
- e is restricted within bound $L \le e \le U$
- b is typically 2 or 10

Example: (5-digit, base 10)

$$0.53216 \times 10^{-4} \qquad 0.00112 \times 10^{8}$$

$$-0.81724 \times 10^{21} \qquad 0.11200 \times 10^{6}$$

Roundoff error

Most real numbers have to be rounded off to be represented in t-digit floating-point numbers.

Definition

If x is a real number and x' is its floating-point approximation, then the difference $x' - x$ is called the absolute error and the quotient $(x' - x)/x$ is called the relative error.

Real number x	4-digit decimal x'	Relative error
62.133	0.6213×10^5	$\frac{-3}{62.133} \approx -4.8 \times 10^{-5}$
0.12658	0.1266×10^0	$\frac{2\times10^{-5}}{0.12658}$
47.213	0.4721×10^2	$\frac{-0.003}{47.213} \approx -6.4 \times 10^5$
π	0.3142×10^1	$\frac{3.142-\pi}{\pi} \approx 1.3 \times 10^{-4}$

With arithmetic operations additional roundoff errors occur:

Example: $a' = 0.263 \times 10^4$, $b' = 0.466 \times 10^1$:

$$a' + b' = 0.263446 \times 10^4$$

but in 3-digit floating point the sum is: 0.263×10^4.

Relative error:

$$\frac{fl(a' + b') - (a' + b')}{a' + b'} = \frac{-4.46}{0.263446 \times 10^4} \approx -0.17 \times 10^2$$

Machine Precision

Relative error:

$$\delta = \frac{(x' - x)}{x} \qquad \text{or} \qquad x' = x(1 + \delta)$$

$|\delta|$ can be bounded by a positive constant ϵ, called machine precision.

Machine precision is the smallest floating-point number ϵ for which

$$fl(1 + \epsilon) > 1$$

Example: (3-digit, decimal basis)

$$fl(1 + 0.499 \times 10^{-2}) = 1$$

while

$$fl(1 + 0.500 \times 10^{-2}) = 1.01$$

The machine ϵ would be 0.500×10^{2}