

Introduction to Scientific Computing

ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยเชียงใหม่

1

- การสร้างโมเดลทางคณิตศาสตร์ และจำลองโมเดลโดยอาศัยคอมพิวเตอร์ เพื่อการศึกษาทางวิทยาศาสตร์
- Keywords
 - Modelling – Mathematics, Statistics
 - Simulation – Computer Science
 - Scientific problem – Physics, Biology, Chemistry, Geology

2

Why Scientific computing ?

- ข้อมูลทางด้านวิทยาศาสตร์มีจำนวนมากขึ้น การคำนวณด้วยมือยากลำบาก
- การทดลองโดยตรงอาจมีความอันตราย หรือ ใช้งบประมาณสูง
- มีความสะดวกและยืดหยุ่นกว่าในการทำการทดลอง สามารถทำการศึกษาดทดลองเวลาไหนก็ได้
- ข้อจำกัด
 - โมเดลทางคณิตศาสตร์ต้องมีความแม่นยำ (ทำได้ยาก)

3

Application (1)

- Numerical simulations
 - การจำลองเหตุการณ์ที่สามารถอธิบายด้วยวิทยาศาสตร์ได้แล้ว
 - เช่น การจำลองการเกิดแผ่นดินไหว จำลองการเกิดชีนามิ
 - หรือใช้ในการคาดเดาเหตุการณ์ล่วงหน้า
 - เช่น การพยากรณ์อากาศ การทำนายตลาดหลักทรัพย์

4

Application (2)

- Model fitting and data analysis

- การสร้างโมเดลขึ้นมาเพื่ออธิบายข้อมูลที่เก็บรวบรวมไว้
- เช่น การสร้างแผนที่ได้นำจากสัญญาณโซนาร์ การสร้างโมเดลทางภาษาจากข้อมูลทางภาษา (เอกสาร ประโยค วลี)
- การใช้กราฟในการโมเดลเนทเวิร์คต่างๆ
- เช่น ระบบสมอง ระบบการทำงานของเซลล์ ข้อมูลจาก Social Network

5

Application (3)

- Computational optimization

- การหาค่าพารามิเตอร์ที่ทำให้ ฟังก์ชันเป้าหมายมีค่าน้อยที่สุด
- เช่น การหาวิธีในการผลิตที่ใช้ต้นทุนต่ำที่สุด
- การหาวิธีในการเดินทางให้ครบทุกจุดหมายโดยใช้ต้นทุนน้อยที่สุด (Travelling salesman problem)

- Parameter

$$X_1, X_2, \dots X_n$$

- Objective

$$f(X_1, X_2, \dots X_n)$$

6

Software for Scientific Computing

- Spreadsheet: Excel
- Scientific computing language
 - เขียนคำสั่งเพื่อกำหนดขั้นตอนการคำนวณ
 - ต้องเรียนรู้ภาษาในการเขียนคำสั่ง
 - MATLAB, R, SciLab

- โปรแกรมเหล่านี้สามารถ

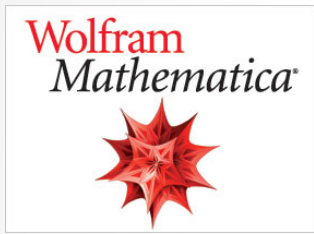
- ทำการคำนวณทางคณิตศาสตร์ได้
- แสดงผลออกมาในรูปแบบต่าง เช่น Graph, Chart, Histogram

7

Software



8



- เป็นภาษาคอมพิวเตอร์ (Programming language) ที่มาพร้อมกับฟังก์ชันการคำนวณทางวิทยาศาสตร์มากมาย
- ง่ายต่อการใช้งาน และเขียน ทำให้เหมาะแก่การสร้างต้นแบบโปรแกรม
- เป็นโปรแกรมแบบโอเพนซอส (Open source) สามารถดาวน์โหลดมาใช้งานได้โดยไม่ต้องจ่ายค่าลิขสิทธิ์
- รองรับระบบปฏิบัติการ Windows, Mac, Linux

9

10

SciLab is capable of doing

- Matrix manipulation
- Linear algebra
- Function optimization
- Differential equation
- Statistics
- Regression
- Signal processing
- Creating 2D and 3D plots

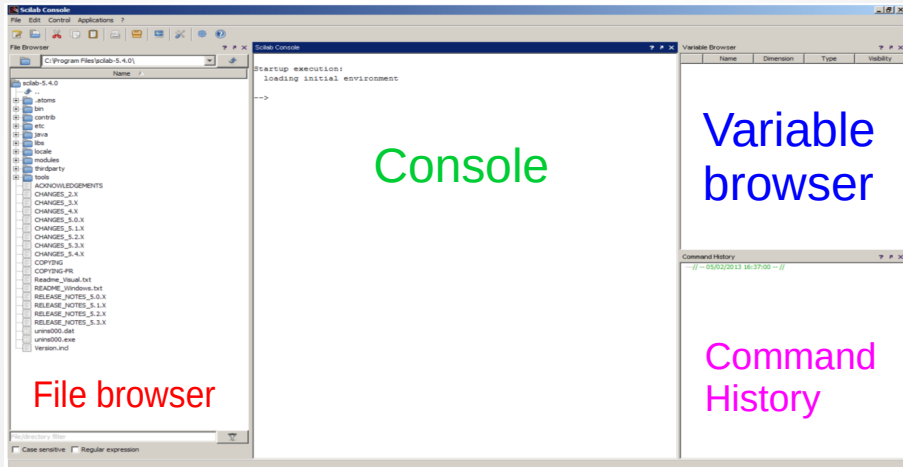
11

How to get SciLab

- สามารถดาวน์โหลดได้จาก
 - <http://www.scilab.org/download>
- มีทั้งเวอร์ชัน 32 และ 64 บิต
- สามารถศึกษาข้อมูลการติดตั้งและการใช้งานได้จากเว็บนั้นเช่นกัน

12

- The main window



13

- พิมพ์คำสั่งหลังเครื่องหมาย '-->' ในหน้าต่างคอนโซล
 - > 57/4
 - ans =
14.25
 - > (2+9)^5
 - ans =
161051.
- สามารถเรียกดูคำสั่งเดิมที่พิมพ์ไปแล้วด้วยลูกศรขึ้นลง

14

How to get help

- สามารถเรียกดูวิธีใช้งานฟังก์ชันได้โดยการพิมพ์ help ลงบนคอนโซล แล้วตามด้วยชื่อฟังก์ชัน เช่น
 - help norm
 - เป็นการเรียกดูวิธีการใช้งานฟังก์ชัน norm
 - help
 - เป็นการเปิดหน้าต่างวิธีใช้งานฟังก์ชันทั้งหมด (ไม่ระบุฟังก์ชัน)

15

การคำนวณพื้นฐาน

- คำสั่งคำนวณต่างๆ
 - การบวก ลบ คูณ หาร ใช้เครื่องหมาย + - * / ตามลำดับ
 - ยกกำลังให้ใช้เครื่องหมาย ^
 - ลองพิมพ์คำสั่งต่อไปนี้
 - > 2 + 5.3
- Scilab เป็นโปรแกรมแบบ case sensitive
 - sqrt() จะไม่เท่ากับ SQRT()
 - เช่น

--> sqrt(9)	--> SQRT(9)
ans =	! -- error 4
3	Undefined variable: SQRT

16

ฟังก์ชันใน Scilab

- ฟังก์ชันสามารถเรียกใช้โดย เรียกชื่อฟังก์ชันพร้อมใส่พารามิเตอร์ให้ถูกต้อง
- เช่น
 - sqrt(9)
 - sqrt คือชื่อฟังก์ชัน ซึ่งจะตามด้วยเครื่องหมาย ()
 - เลข 9 ในวงเล็บคือพารามิเตอร์ที่จะส่งไปยังฟังก์ชัน
 - หากไม่แน่ใจว่าฟังก์ชันนี้เรียกใช้อย่างไรให้หาคำช่วยเหลือ
--> help sqrt

17

ฟังก์ชันที่มีประโยชน์

- rand()
- sqrt(), log(), exp()
- ones()
- zeros()
- abs()
- sin(), cos(), tan(), asin()...
- max(), min(), ceil(), floor()...
- mean(), mode(), median(), stdev()

18

ค่าคงตัว (constant) ต่างๆ

- ใน Scilab จะมีการบัญญัติค่าคงตัวที่ใช้บ่อยไว้แล้ว ยกตัวอย่างเช่น ค่า e และค่า pi
- การเรียกใช้ ให้ใช้สัญลักษณ์ % นำหน้า
- เช่น
 - >%pi
 - %pi =
 - 3.1415927

19

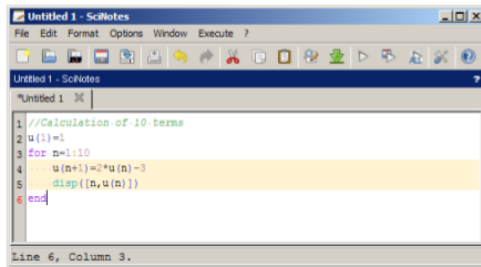
การไม่แสดงผลลัพธ์

- ปกติแล้วหากเราพิมพ์คำสั่งบน console จะมีการแสดงผลลัพธ์เสมอ
 - >%pi
 - %pi =
 - 3.1415927
- หากไม่ต้องการแสดงผลลัพธ์เราสามารถใช้เครื่องหมาย ; เพื่อปิดการแสดงผล
 - >%pi;

20

การพิมพ์คำสั่งหลายๆคำสั่งเก็บไว้ในไฟล์

- การใช้งาน Scilab แบบโต้ตอบมีข้อดีคือรวดเร็ว หากเป็นการคำนวณแบบสั้น
- แต่ถ้าวานยาวขึ้น การเขียนขั้นตอนการคำนวณ (code) ไว้ก่อนจะสะดวกกว่า และสามารถกลับมาแก้ไขได้
- การเขียนโค้ดสามารถทำได้บน Editor ของ scilab
- เครื่องหมาย // คือ comment ซึ่งจะไม่ถูกแปลผลโดย scilab



21

Programming with Scilab

- ตัวแปร (Variable)
 - เป็นเสมือนที่เก็บค่าที่เราต้องการ
 - ตัวแปรใน Scilab จะแสดงเป็น เมทริกซ์ ทั้งหมด (ตัวเลข คือ 1x1 เมทริกซ์)
 - ตัวแปรจะต้องมีการกำหนดค่าก่อนที่จะเรียกใช้
 - การประกาศตัวแปรไม่จำเป็นต้องระบุชนิดของตัวแปร
 - ชื่อตัวแปรต้องไม่ซ้ำกับค่าคงที่ที่กำหนดไว้แล้วโดยระบบ
 - ชื่อตัวแปรต้องเป็นอักษรภาษาอังกฤษหรือตัวเลขหรือ underscore '_' เท่านั้น
 - ชื่อตัวแปรต้องไม่ขึ้นต้นด้วยตัวเลข และ อักษรตัวเล็กตัวใหญ่มีผล (case sensitive)
 - ผลลัพธ์จากการคำนวณที่ไม่ได้ถูกเก็บไว้ในตัวแปร จะถูกเก็บไว้ในตัวแปรที่ชื่อว่า ans

22

Matrices (1)

- เราสามารถสร้างเมทริกซ์ได้หลายวิธี
 - การใช้ []
 - A = [1 2 3] คือการสร้าง 1x3 เมทริกซ์ A
 - การใช้ ; ร่วมกับ [] เพื่อการขึ้นแถวใหม่
 - B = [1; 2; 3] คือการสร้าง 3x1 เมทริกซ์ B
 - การสร้างเมทริกซ์ 3x3
 - C = [1 2 3; 4 5 6; 7 8 9]
 - การใช้เครื่องหมาย ' เพื่อทำการ Transpose
 - A จะเท่ากับ B'

23

Matrices (2)

- การเรียกดูขนาดของ matrix
 - size(variable)

```
-->a = [1 2; 3 4]
a =
```

```
1.  2.
3.  4.
```

```
-->size(a)
ans =
```

```
2.  2.
```

```
-->[nr nc] = size(a)
nc =
```

```
2.
nr =
```

```
2.
```

24

Matrices (3)

- การเรียกดูขนาดแบบอื่นๆ
- `size(a, "sel")`
 - `sel=1` --> เรียกดูจำนวนแถวของเมทริกซ์
 - `sel=2` --> เรียกดูจำนวนหลักของเมทริกซ์
 - `sel="*"` --> เรียกดูจำนวนสมาชิกของเมทริกซ์

```
-->a = ones(2,3)
a =

    1.    1.    1.
    1.    1.    1.

-->size(a,"*")
ans =

    6.
```

25

ตัวดำเนินการทางคณิตศาสตร์

การบวก

```
-->X = [1 2 3]; Y=[5 4 3];
-->X+Y
ans =

    6.    6.    6.
```

การลบ

```
-->X = [1 2 3]; Y=[5 4 3];
-->X-Y
ans =

   -4.   -2.    0.
```

การคูณ

```
-->X = [1 2 3]; Y=[5 4 3];
-->X*Y
!--error 10
Inconsistent multiplication.
```

```
-->X*Y'
ans =

    22.

-->X'*Y
ans =

     5.     4.     3.
    10.     8.     6.
    15.    12.     9.
```

การคูณที่ละตัว

```
-->X = [1 2 3]; Y=[5 4 3];
-->X.*Y
ans =

     5.     8.     9.
```

การหารที่ละตัว

```
-->a = 1:3
a =

     1.     2.     3.

-->1.0./a
ans =

     1.     0.5    0.33333333
```

การกำหนดค่าต่อเนื่อง

- สามารถกำหนดเวกเตอร์ที่ประกอบด้วยค่าเริ่มต้น ถึงค่าสุดท้าย โดยค่าจะเพิ่มขึ้นทีละเท่าๆกันโดย
- `A = start:step:stop`
 - Start คือ ค่าเริ่มต้น
 - Step คือ จำนวนที่ต้องการให้เพิ่มขึ้น
 - Stop คือ ค่าสุดท้าย
 - เช่น `a = 1:3:10`, จะได้ `a = [1 4 7 10]`
- หากต้องการให้ค่าเพิ่มทีละ 1 ไม่ต้องระบุค่า step ก็ได้
 - เช่น `a = 1:5`, จะได้ `a = [1 2 3 4 5]`
- สามารถกำหนด step เป็นค่าลบได้ด้วย เช่น `a=3:-1:1` จะได้ `a=[3 2 1]`

27

การเรียกดูค่าของ matrix

- เรียกดูค่าในแถวที่ *i* หลักที่ *j* ของเมทริกซ์ *A* ได้จาก `A(i,j)`
- นอกจากนั้นยังมีวิธี

<code>A</code>	the whole matrix
<code>A(:, :)</code>	the whole matrix
<code>A(i : j, k)</code>	the elements at rows from <i>i</i> to <i>j</i> , at column <i>k</i>
<code>A(i, j : k)</code>	the elements at row <i>i</i> , at columns from <i>j</i> to <i>k</i>
<code>A(i, :)</code>	the row <i>i</i>
<code>A(:, j)</code>	the column <i>j</i>

28

การแสดงผลบนหน้าจอ

- 1.สามารถแสดงค่าด้วยการไม่ปิดท้ายคำสั่งด้วย ;
- 2.การใช้ฟังก์ชัน disp()

```
-->disp("I like scilab")
I like scilab
```

- หากปิดคำสั่ง disp ช่างต้นด้วย ; โปรแกรมจะแสดงผลหรือไม่ ?

29

การทดสอบทางตรรกศาสตร์

Equal	Different	Less	Greater	Less or equal	Greater or equal
==	<>	<	>	<=	>=
True	False	And	Or	No	
%T	%F	&		~	

```
-->5 < 10
ans =
T
```

```
-->(10 == 9) == %T
ans =
F
```

```
-->%T & %F
ans =
F
```

30

การเปรียบเทียบ matrix

Using logical operator

```
-->X = [1 2 3]; Y=[5 4 3];
-->X==Y
ans =
F F T
```

Using isequal() function

```
-->X = [1 2 3]; Y=[5 4 3];
-->isequal(X,Y)
ans =
F
-->~isequal(X,Y)
ans =
T
```

31

Exercise

Exercise 4.1 (Plus one) Create the vector $(x_1 + 1, x_2 + 1, x_3 + 1, x_4 + 1)$ with the following x .

```
x = 1:4;
```

Exercise 4.2 (Vectorized multiplication) Create the vector $(x_1 y_1, x_2 y_2, x_3 y_3, x_4 y_4)$ with the following x and y .

```
x = 1:4;
y = 5:8;
```

Exercise 4.3 (Vectorized invert) Create the vector $(\frac{1}{x_1}, \frac{1}{x_2}, \frac{1}{x_3}, \frac{1}{x_4})$ with the following x .

```
x = 1:4;
```

Exercise 4.4 (Vectorized division) Create the vector $(\frac{x_1}{y_1}, \frac{x_2}{y_2}, \frac{x_3}{y_3}, \frac{x_4}{y_4})$ with the following x and y .

```
x = 12*(6:9);
y = 1:4;
```

Exercise 4.5 (Vectorized squaring) Create the vector $(x_1^2, x_2^2, x_3^2, x_4^2)$ with $x = 1, 2, 3, 4$.

Exercise 4.6 (Vectorized sinus) Create the vector $(\sin(x_1), \sin(x_2), \dots, \sin(x_{10}))$ with x is a vector of 10 values linearly chosen in the interval $[0, \pi]$.

Exercise 4.7 (Vectorized function) Compute the $y = f(x)$ values of the function f defined by the equation

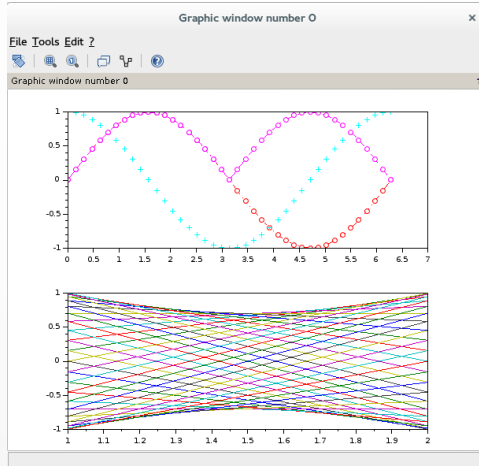
$$f(x) = \log_{10}(r/10^x + 10^x) \quad (1)$$

with $r = 2.220 \cdot 10^{-16}$ and x a vector of 100 values linearly chosen in the interval $[-16, 0]$.

32

Plotting

- Scilab มีฟังก์ชันสนับสนุนการพลอตกราฟที่ดีเยี่ยม
- ลองพิมพ์
--> plot



33

Basic plotting

- ใช้ฟังก์ชัน plot ในการพลอตกราฟ
- แนะนำให้ศึกษา help ของ plot
- การเรียกใช้อยู่ในรูป

`plot(x, y, "appearance");`

หมายเหตุ

- x และ y ต้องมีขนาดเท่ากัน
- appearance อยู่ในรูป "sc"
- s คือ style ของจุด c คือสีของจุด

34

Plotting appearance

- Colors

"b" = blue (by default), "k" = black, "r" = red, "g" = green, "c" = cyan, "m" = magenta, "y" = yellow, "w" = white.

- Point styles

Joined (by default), or ".", "+", "o", "x", "*".

Other options are available with: "s", "d", "v", "<", and ">".

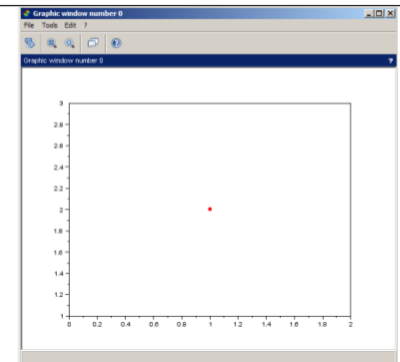
Plotting example

Plot the point A(1 ; 2) with a red point.

Scilab Editor

```
plot(1,2,".r")
```

Graphics Window



35

36

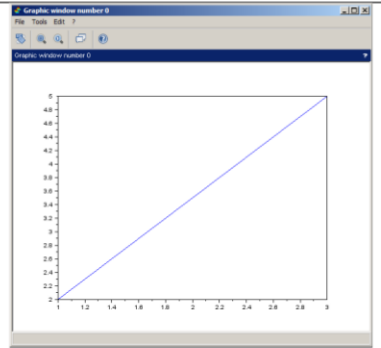
Plotting a segment

Plot the segment [AB] in blue (by default) with A(1 ; 2) and B(3 ; 5).

Scilab Editor

Graphics Window

```
plot([1,3],[2,5])
```



Default color=blue, Default style=connected line

37

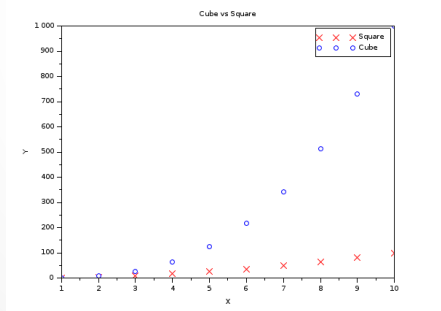
การ plot แบบอื่นๆ

- surf() สร้าง plot สามมิติ surf(a,b,rand(10,10))
- pie() สร้าง pie chart
- histplot() สร้าง histogram
- bar() สร้าง bar chart

38

การใส่ชื่อกราฟ, แกน x, แกน y

- title("ชื่อกราฟ")
- xtitle("ชื่อกราฟ", "ชื่อแกน x", "ชื่อแกน y")
- legend("ข้อมูลชุดที่ 1", "ข้อมูลชุดที่ 2", ...)



39

References

- SciLab for beginner
 - www.scilab.org/content/download/849/7901/file/Scilab_beginners.pdf
- Longer version of the above
 - www.scilab.org/content/download/247/1702/file/introscilab.pdf

40