

198 343 Principles of Digital Communications Laboratory

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

DTC-01: การคำนวณเชิงพีชคณิตด้วยเซจแมท (Sagemath)

1 วัตถุประสงค์

1. เพื่อศึกษาการประยุกต์โปรแกรมเซจไพธอน สำหรับการคำนวณในรูปสัญลักษณ์ หรือการคำนวณนิพจน์ที่คล้ายกับการคำนวณด้วยตนเอง
2. เพื่อศึกษาวิธีการใช้โปรแกรมเพื่อแก้ปัญหาทางแคลคูลัสและพีชคณิตเชิงเส้น

2 การใช้คำสั่งของเซจ

2.1 คำสั่งเกี่ยวข้องกับโน้ตบุ๊ก

ในหัวข้อนี้ แนะนำคำสั่งพื้นฐานเกี่ยวข้องกับโน้ตบุ๊ก ดังปรากฏในตารางที่ 1

ตารางที่ 1 คำสั่งเกี่ยวข้องกับโน้ตบุ๊ก

Evaluate cell: <shift-enter> <i>com</i> <tab> tries to complete <i>command</i> <i>command?</i> <tab> shows documentation <i>command??</i> <tab> shows source <i>a.</i> <tab> shows all methods for object <i>a</i> <code>lprint()</code> toggle LaTeX output mode <code>version()</code> print version of Sage Insert cell: click on blue line between cells Delete cell: delete content then backspace (more: <code>dir(a)</code>)
Numerical types Integers: $\mathbb{Z} = \mathbb{ZZ}$ e.g. -2 -1 0 1 10^{100} Rationals: $\mathbb{Q} = \mathbb{QQ}$ e.g. $1/2$ $1/1000$ $314/100$ -42 Decimals: $\mathbb{R} \approx \mathbb{RR}$ e.g. .5 0.001 3.14 -42. Complex: $\mathbb{C} \approx \mathbb{CC}$ e.g. $1+i$ $2.5-3*i$

ตัวอย่าง 1. ศึกษาคำสั่งบางส่วนที่เกี่ยวข้องกับโน้ตบุ๊ก

`sin?<tab>`

sin?

```
File: /ext/sage/sage-8.1/local/lib/python2.7/site-packages/sage/functions/trig.py
Signature : sin(self, coerce=True, hold=False, dont_call_method_on_arg=False, *args)
Docstring :
The sine function.
```

EXAMPLES:

```
sage: sin(0)
0
sage: sin(x).subs(x==0)
0
sage: sin(2).n(100)
0.90929742682568169539601986591
```

รูปที่ 1 เอกสารที่เกี่ยวข้องกับคำสั่ง sin

แสดงเอกสารที่เกี่ยวข้องกับคำสั่ง sin ได้ดังรูปที่ 1

```
sin??<tab>
```

แสดงเอกสารที่เกี่ยวข้องกับรหัสแหล่งที่มา (source code) ของ sin ได้ดังรูปที่ 2

sin??

```
File: /ext/sage/sage-8.1/local/lib/python2.7/site-packages/sage/functions/trig.py
Source:
class Function_sin(GinacFunction):
    def __init__(self):
        """
        The sine function.
```

รูปที่ 2 รหัสแหล่งที่มาของคำสั่ง sin

#

2.2 คำสั่งเกี่ยวข้องกับค่าคงที่และฟังก์ชัน

ในหัวข้อนี้ แนะนำคำสั่งพื้นฐานเกี่ยวข้องกับค่าคงที่และฟังก์ชัน ดังปรากฏในตารางที่ 2

ตารางที่ 2 คำสั่งเกี่ยวข้องกับค่าคงที่และฟังก์ชัน

Constants: $\pi = \text{pi}$ $e = e$ $i = i$ $\infty = \text{oo}$
Approximate: $\text{pi.n(digits=18)} = 3.14159265358979324$
Functions: \sin \cos \tan \sec \csc \cot \sinh \cosh \tanh sech csch coth \log \ln \exp
$ab = a*b$ $\frac{a}{b} = a/b$ $a^b = a^b$ $\sqrt{x} = \text{sqrt}(x)$ $\sqrt[n]{x} = x^{(1/n)}$ $ x = \text{abs}(x)$ $\log_b(x) = \log(x,b)$
Symbolic variables: e.g. $t, u, v, y, z = \text{var}('t u v y z')$
Define function: e.g. $f(x) = x^2$ $f(x) = x^2$
or $f = \text{lambda } x: x^2$ or $\text{def } f(x): \text{return } x^2$

ตัวอย่าง 2. ศึกษาคำสั่งในการสร้างฟังก์ชัน $f(x) = x^2$ ด้วยคำสั่ง

```
f(x)=x^2
```

```
f(1)
```

```
f(2)
```

```
1
```

```
4
```

หรือ

```
f=lambda x: x^2
```

```
f(1)
```

```
f(2)
```

```
1
```

```
4
```

หรือ

```
def f(x): return x^2
```

```
f(1)
```

```
f(2)
```

```
1
```

```
4
```

#

ตัวอย่าง 3. ศึกษาคำสั่งในการคำนวณ $x + y$ ด้วยคำสั่ง

```
x,y=var("x y")
```

```
x+y
```

```
x+y
```

#

2.3 คำสั่งเกี่ยวข้องกับการดำเนินการบนนิพจน์

ในหัวข้อนี้ แนะนำคำสั่งพื้นฐานเกี่ยวข้องกับการดำเนินการบนนิพจน์ ดังปรากฏในตารางที่ 3

ตารางที่ 3 คำสั่งเกี่ยวข้องกับการดำเนินการบนนิพจน์

```
factor(...)    expand(...)    (...).simplify_...  
Symbolic equations: f(x)==g(x)  
_ is previous output  
_+a  _-a  _*a  _/a  manipulates equation  
Solve  $f(x) = g(x)$ : solve(f(x)==g(x), x)  
solve([f(x,y)==0, g(x,y)==0], x, y)  
find_root(f(x), a, b) find  $x \in [a, b]$  s.t.  $f(x) \approx 0$   
 $\sum_{i=k}^n f(i) = \text{sum}([f(i) \text{ for } i \text{ in } [k..n]])$   
 $\prod_{i=k}^n f(i) = \text{prod}([f(i) \text{ for } i \text{ in } [k..n]])$ 
```

ตัวอย่าง 4. ศึกษาคำสั่งในการกระจายพหุนาม ด้วยคำสั่ง

```

R=QQ
R.<x>=PolynomialRing(R)
expand((1/2)*(x-2))
expand((x+3)*(x-2))
1/2*x - 1
x^2 + x - 6

```

#

ตัวอย่าง 5. ศึกษาคำสั่งในการหาคำตอบสมการเชิงพหุนาม ด้วยคำสั่ง

```

f(x)=x^2-1
solve(f(x)==0,x)
f(x,y)=2*x+3*y-2
g(x,y)=x-2*y-1
solve([f(x,y)==0,g(x,y)==0],x,y)
[x == -1, x == 1]
[[x == 1, y == 0]]

```

#

2.4 คำสั่งทางแคลคูลัส

ในหัวข้อนี้ แนะนำคำสั่งพื้นฐานเกี่ยวข้องกับแคลคูลัส ดังปรากฏในตารางที่ 4

ตารางที่ 4 คำสั่งพื้นฐานเกี่ยวข้องกับแคลคูลัส

```

lim f(x) = limit(f(x), x=a)
lim f(x) = limit(f(x), x=a, dir='minus')
lim f(x) = limit(f(x), x=a, dir='plus')
d/dx(f(x)) = diff(f(x), x)
d/dx(f(x,y)) = diff(f(x,y), x)
diff = differentiate = derivative
∫ f(x)dx = integral(f(x), x)
integral = integrate
∫ab f(x)dx = integral(f(x), x, a, b)
Taylor polynomial, deg n about a: taylor(f(x), x, a, n)
Summing infinite series
∑n=1∞ 1/n2 = π2/6
s = 'sum (1/n2,n,1,inf), simpsum'
SR(sage.calculus.calculus.maxima(s))

```

ตัวอย่าง 6. ศึกษาคำสั่งในการหาคำตอบปัญหาทางแคลคูลัส ด้วยคำสั่ง

```

f(x)=sin(pi*x)/(pi*x)
g(x)=sin(x)
limit(f(x),x=0)
diff(f(x),x)
integral(g(x),x)
integral(g(x),x,-pi,pi)

```

```

1
cos(pi*x)/x - sin(pi*x)/(pi*x^2)
-cos(x)
0

```

#

2.5 คำสั่งการพล็อตกราฟ 2 มิติ

ในหัวข้อนี้ แนะนำคำสั่งพื้นฐานเกี่ยวกับการพล็อตกราฟ 2 มิติ ดังปรากฏในตารางที่ 5

ตารางที่ 5 คำสั่งพื้นฐานเกี่ยวกับการพล็อตกราฟ 2 มิติ

```

line([ (x1,y1), ..., (xn,yn) ], options)
polygon([ (x1,y1), ..., (xn,yn) ], options)
circle((x,y),r,options)
text("txt", (x,y), options)
options as in plot.options, e.g. thickness=pixel,
rgbcolor=(r,g,b), hue=h where 0 ≤ r,b,g,h ≤ 1
use option figsize=[w,h] to adjust aspect ratio
plot(f(x), xmin, xmax, options)
parametric_plot((f(t),g(t)), tmin, tmax, options)
polar_plot(f(t), tmin, tmax, options)
combine graphs: circle((1,1),1)+line([(0,0),(2,2)])
animate(list of graphics objects, options).show(delay=20)

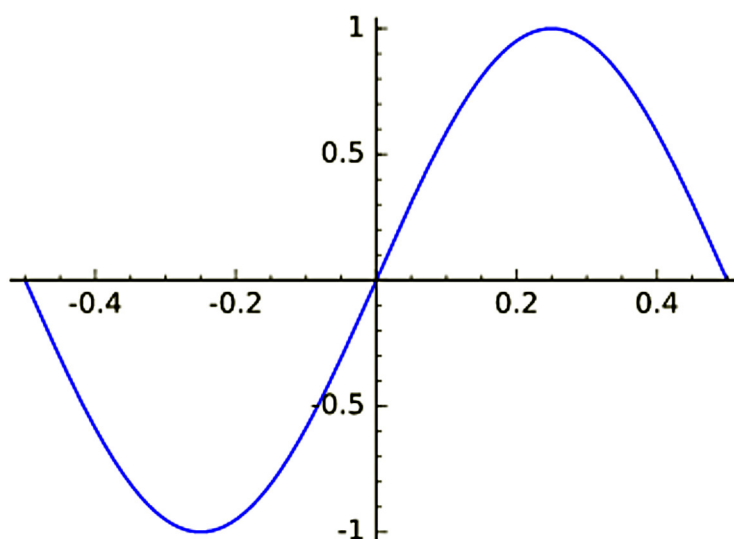
```

ตัวอย่าง 7. ศึกษาคำสั่งในการพล็อตกราฟ 2 มิติ ด้วยคำสั่ง

```

f(x)=sin(2*pi*x)
plot(f(x),x,-1/2,1/2,figsize=(4,3))

```



รูปที่ 3 การพล็อตกราฟ $\sin(2\pi x)$

#

2.6 คำสั่งพื้นฐานเกี่ยวข้องกับวิยุตคณิต

ในหัวข้อนี้ แนะนำคำสั่งพื้นฐานเกี่ยวข้องกับวิยุตคณิต ดังปรากฏในตารางที่ 6

ตารางที่ 6 คำสั่งพื้นฐานเกี่ยวข้องกับวิยุตคณิต

```
[x] = floor(x)    [x] = ceil(x)
Remainder of n divided by k = n%k    k|n iff n%k==0
n! = factorial(n)    (x/m) = binomial(x,m)
phi = golden_ratio    phi(n) = euler_phi(n)
Strings: e.g. s = 'Hello' = "Hello" = ""+"He"+"llo"
s[0]='H'    s[-1]='o'    s[1:3]='el'    s[3:]='lo'
Lists: e.g. [1,'Hello',x] = []+[1,'Hello']+ [x]
Tuples: e.g. (1,'Hello',x)    (immutable)
Sets: e.g. {1,2,1,a} = Set([1,2,1,'a']) (= {1,2,a})
List comprehension ≈ set builder notation, e.g.
{f(x) : x in X, x > 0} = Set([f(x) for x in X if x>0])
```

2.7 คำสั่งพื้นฐานเกี่ยวข้องกับพีชคณิตเชิงเส้น

ในหัวข้อนี้ แนะนำคำสั่งพื้นฐานเกี่ยวข้องกับพีชคณิตเชิงเส้น ดังปรากฏในตารางที่ 7

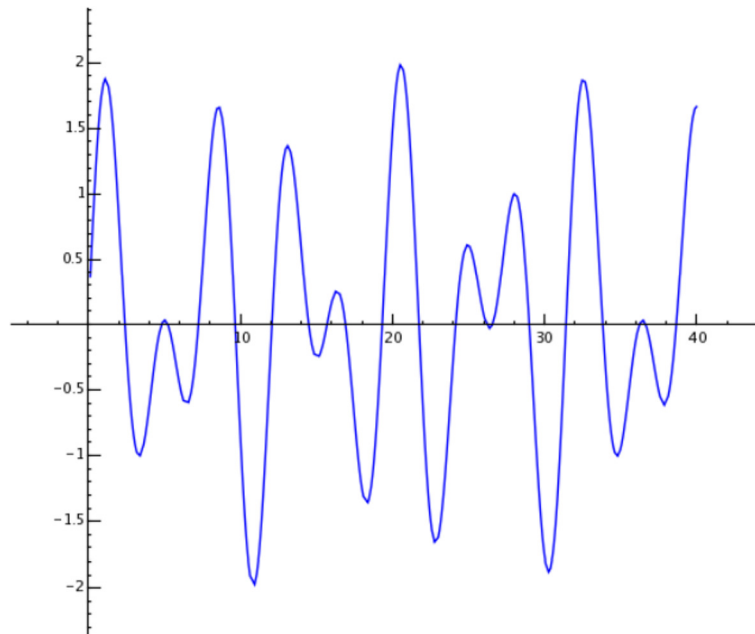
ตารางที่ 7 คำสั่งพื้นฐานเกี่ยวข้องกับพีชคณิตเชิงเส้น

```
(1)
(2) = vector([1,2])
(1 2)
(3 4) = matrix([[1,2],[3,4]])
| 1 2 |
| 3 4 | = det(matrix([[1,2],[3,4]]))
Av = A*v    A^-1 = A^-1    A^t = A.transpose()
methods: nrows() ncols() nullity() rank() trace()...
```

ตัวอย่าง 8.

```
sage: 3 + 5
8
sage: 57.1 ^ 100
4.60904368661396e175
sage: matrix([[1,2], [3,4]])^(-1)
[ -2      1]
[ 3/2 -1/2]
sage: x = var('x') # create a symbolic variable
sage: integrate(sqrt(x)*sqrt(1+x), x)
1/4*((x + 1)^(3/2)/x^(3/2) + sqrt(x + 1)/sqrt(x))/((x + 1)^2/x^2 - 2*(x + 1)/x + 1) -
1/8*log(sqrt(x + 1)/sqrt(x) + 1) + 1/8*log(sqrt(x + 1)/sqrt(x) - 1)
sage: a = var('a')
sage: S = solve(x^2 + x == a, x); S
```

```
[x == -1/2*sqrt(4*a + 1) - 1/2, x == 1/2*sqrt(4*a + 1) - 1/2]
sage: S[0].rhs()
-1/2*sqrt(4*a + 1) - 1/2
sage: show(plot(sin(x) + sin(1.6*x), 0, 40))
```



2.8 โครงสร้างข้อมูล

ในที่นี้แนะนำรายการ (list) และพจนานุกรม (dictionary) ที่ถูกใช้งานมาก รายการเป็นโครงสร้างข้อมูลที่กำหนดค่าเป็นกลุ่ม การสร้างทำได้โดยการใช้วงเล็บก้ามปู [และ มักใช้ร่วมกับฟังก์ชัน range() ดังเช่น

```
[ <expr> for <name> in <iterable> (if <condition>) ]
```

ตัวอย่าง 9. สร้างรายการของตัวเลขคู่ตั้งแต่ 0 ถึง 9 ยกกำลังสอง ด้วยโปรแกรมเซกแมท ได้ดังนี้

```
[ i^2 for i in range(10) if i % 2 == 0 ]
[0, 4, 16, 36, 64]
```

#

พจนานุกรมเป็นตารางความสัมพันธ์ระหว่างคีย์กับค่า ด้วยรูปแบบคำสั่งเป็น

```
{key1 : value1, key2 : value2 ...}
```

ตัวอย่าง 10. สร้างพจนานุกรมเป็นตารางความสัมพันธ์ระหว่างชื่อเด็กกับอายุได้ดังนี้

```
age={"cat":1,"mike":2,"john":3}
age["mike"]
```

2

#

2.9 การปฏิสัมพันธ์กับเซจแมท

ปฏิสัมพันธ์กับเซจโน้ตบุ๊กเซลล์สามารถกระทำด้วยวิธี ตัวเลื่อน กล่องข้อความ ปุ่ม เช็กล่อง เป็นต้น การสร้างปฏิสัมพันธ์กับเซจโน้ตบุ๊กเซลล์ จะเริ่มต้นด้วย การใส่คำสั่ง `@interact` ในบรรทัด ก่อนหน้า ฟังก์ชันที่นิยามขึ้น ดังตัวอย่างต่อไปนี้

ตัวอย่าง 11. ปฏิสัมพันธ์กับเซจโน้ตบุ๊กเซลล์ด้วยวิธี กล่องข้อความ และตัวเลื่อน เพื่อหาค่า $a + y$

```
@interact
def _(a=5, y=[0..10]):
    print a+y
```

ได้ผลรันดังรูปที่ 4

a

y

5

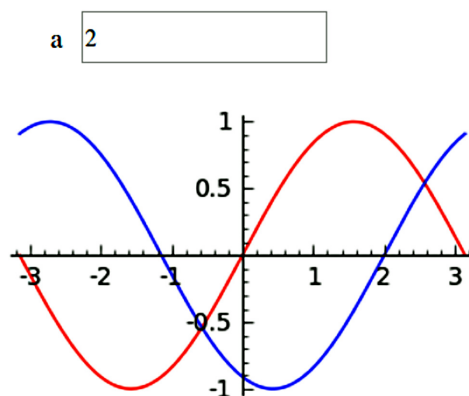
รูปที่ 4 ปฏิสัมพันธ์กับเซจโน้ตบุ๊กเซลล์ด้วยวิธี กล่องข้อความ และตัวเลื่อน

#

ตัวอย่าง 12. ปฏิสัมพันธ์กับเซจโน้ตบุ๊กเซลล์ด้วยวิธี กล่องข้อความ เพื่อพล็อตกราฟ $\sin(x), \sin(x - a)$

```
f(x)=sin(x)
@interact
def _(a=0):
    P1=plot(f(x), x, -pi, pi, color="red", figsize=(3,2))
    P2=plot(f(x-a), x, -pi, pi, color="blue", figsize=(3,2))
    show(P1+P2)
```

ได้ผลรันดังรูปที่ 5



รูปที่ 5 ปฏิสัมพันธ์กับเซจโน้ตบุ๊กเซลล์ด้วยวิธี กล่องข้อความ เพื่อพล็อตกราฟ $\sin(x), \sin(x - a)$

#

2.10 โครงสร้างการควบคุม

คำสั่งเชิงเงื่อนไข

```
if <condition>:
    <instruction sequence>
[elif <condition>:
    <instruction sequence>]*
[else:
    <instruction sequence>]
```

คำสั่งทำซ้ำ

```
for <name> in <iterable>:
    <instruction sequence>
[else:
    <instruction sequence>]
```

และ

```
while <condition>:
    <instruction sequence>
[else:
    <instruction sequence>]
```

ตัวอย่าง 13. นิยามฟังก์ชัน $f(x)$ ดังสมการ

$$f(x) = \begin{cases} \tan(\pi x/4), & \text{ถ้า } |x| < 1, \\ x, & \text{ถ้า } |x| \geq 1. \end{cases}$$

ด้วยเซกเมนต์เป็น

```
def f(x):
    if abs(x)<1:
        return tan(pi*x/4)
    else:
        return x

print f(2)
print f(-0.15).n(100)
2
-0.11835779964076783095861912482
```

#

ตัวอย่าง 14. พิมพ์จำนวนเต็มตั้งแต่ 10 ถึง 15 ด้วยโปรแกรมเซกเมนต์ ได้ดังนี้

```
for x in [10..15]:
    print x
```

```
10
11
12
13
14
15
```

#

ตัวอย่าง 15. พิมพ์จำนวนเต็มคู่ ที่มีค่ามากกว่าหรือเท่ากับ 10 ถึง 15 ด้วยโปรแกรมเซจแมท ได้ดังนี้

```
y=10
while y<= 15:
    if y%2==1:
        print y
    y+=1
```

11

13

15

#

3 ใบงาน

- บทนิยาม 1. a) ถ้า a, n เป็นจำนวนเต็มบวก แล้ว $a \bmod n$ หมายถึง เศษเหลือจากการหาร a ด้วย n
 b) รหัสคำนวณ x_0 หมายถึง ตัวเลขสองหลักสุดท้ายโดยไม่นับ – ตัวอย่างเช่น รหัส 583030452-4 ได้ $x_0 = 24$
 c) ให้ $g = 3$ และให้ $\gamma = g^{x_0} \bmod 7$ และ $\beta = g^{x_0+2} \bmod 7$ แล้ว คำสั่งในเซกเมนต์สำหรับหาค่า γ คือ

$x_0=24$

$g=3$

$\gamma = \text{power_mod}(g, x_0, 7)$

γ

1

ข้อกำหนด 1. ในโจทย์แต่ละข้อ จะต้องแทนค่า γ ที่ได้จากการคำนวณในบทนิยามที่ 1 ในสมการของโจทย์ก่อน แล้วจึงเขียนโปรแกรม

- จงเขียนโปรแกรมเพื่อป้อนเข้า (Enter) ฟังก์ชัน $f(x) = \frac{x^2 - x + 7}{x - \gamma}$ เข้าไปในเซกเมนต์ มา 2 วิธี

วิธีที่ 1.

Source code คือ

หาผลรันได้

$$f(x) =$$

$$f(10) =$$

วิธีที่ 2.

Source code คือ

หาผลรันได้

$$f(x) =$$

$$f(10) =$$

2. จงเขียนโปรแกรมเพื่อป้อนค่า (Enter) ฟังก์ชันต่อไปนี้ เข้าไปในเซจแมท

$$f(x) = \begin{cases} \gamma x^2, & x \leq 0 \\ 0, & \text{อื่นๆ} \end{cases}$$

Source code คือ

หาผลรันได้

$$f(2) =$$

$$f(-2) =$$

Source code สำหรับพล็อตกราฟเมื่อกำหนด $x \in [-10, 10]$ และ `figsize= (3, 4)` คือ

ผลรันได้กราฟเมื่อ $x \in [-10, 10]$ และ `figsize= (3, 4)` เป็น

3. จงเขียนโปรแกรมสำหรับปฏิสัมพันธ์กับเชอโนตบุ๊กเซลล์ด้วยวิธี กล่องข้อความ เพื่อหา $g(x) = f'(x)$ เมื่อ

$$f(x) = \sqrt{1 + \sqrt{x + \gamma}}$$

โดยให้ค่าเริ่มต้นของกล่องข้อความ มี $x = 5$

Source code คือ

หาผลรัน เมื่อ $x = 9$ ได้

$$g(9) =$$

4. จงเขียนโปรแกรมสำหรับปฏิสัมพันธ์กับเชอโนตบุ๊กเซลล์ด้วยวิธี ตัวเลื่อนค่า a เพื่อพล็อตกราฟของฟังก์ชัน $f(x)$ และ $g(x) = f(x - a)$ ในรูปเดียวกัน เมื่อ

$$f(x) = \frac{\cos(x)/\gamma}{2x^2 + 1}$$

และจำนวนเต็ม $a \in [-3, 3]$

Source code เมื่อ $x \in [-8, 8]$ คือ

บันทึกรูปภาพเมื่อ $a = -2$

บันทึกรูปภาพเมื่อ $a = 2$