



รายงาน

เรื่อง โปรแกรมคำนวณการขนส่งโดยใช้ Gauss-Jordan

จัดทำโดย

62015003 นายกฤษณพงศ์ บัวทองจันทร์

62015019 นายฉัตรชัย นพพลั้ง

เสนอ

รศ.ดร.อรฉัตร จิตต์โสภาคย์

รายงานนี้เป็นส่วนหนึ่งของรายวิชา

01076032 สมการอนุพันธ์และพีชคณิตเชิงเส้นพื้นฐาน

(ELEMENTARY DIFFERENTIAL EQUATIONS AND LINEAR ALGEBRA)

คณะ วิศวกรรมศาสตร์ สาขาวิชา วิศวกรรมคอมพิวเตอร์ (ต่อเนื่อง)

ประจำปีการศึกษา 2/2563

คำนำ

รายงานนี้เป็นส่วนหนึ่งของรายวิชา ELEMENTARY DIFFERENTIAL EQUATIONS AND LINEAR ALGEBRA (รหัสวิชา 01076032) สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มีจุดประสงค์เพื่อศึกษาความรู้เกี่ยวกับกระบวนการนำเรื่องที่ศึกษาเล่าเรียนไปประยุกต์ใช้งานให้เกิดประโยชน์ในชีวิตประจำวัน โดยรายงานฉบับนี้ทางคณะผู้จัดทำหวังเป็นอย่างยิ่งว่าจะเป็นประโยชน์แก่ผู้สนใจที่จะศึกษาเกี่ยวกับการประยุกต์ใช้งานในชีวิตประจำวันของเนื้อหาในวิชา ELEMENTARY DIFFERENTIAL EQUATIONS AND LINEAR ALGEBRA ที่ไม่มากนักน้อย หากมีข้อผิดพลาดประการใด ทางคณะผู้จัดทำขออภัยมา ณ ที่นี้

คณะผู้จัดทำ

สารบัญ

เรื่อง	หน้า
1. จุดประสงค์การใช้ประยุกต์ใช้งาน	1
2. ภาพรวมขั้นตอนการทำงาน	1
3. รายละเอียดข้อมูลที่เกี่ยวข้อง	2
4. รายละเอียดขั้นตอนการคำนวณ พร้อมคำอธิบาย	4
5. แสดงผลลัพธ์การทำงาน ที่ได้จากการคำนวณ	8
6. สรุปความน่าสนใจของการประยุกต์ใช้งานที่นำเสนอ	8
7. แหล่งอ้างอิงข้อมูล	9

1. จุดประสงค์การใช้ประยุกต์ใช้งาน

โปรแกรมคำนวณการขนส่งนี้จะเป็นการคำนวณการขนส่งกล้วยไม้ระหว่าง 2 ฟาร์ม ไปยังอีก 2 ตัวแทนจำหน่ายโดยใช้หลักการของ Gauss-Jordan ในการคำนวณโดยมีการคำนวณดังนี้ มีฟาร์ม 1 กับ 2 ทั้ง 2 ฟาร์มมีจำนวนกล้วยไม้ที่สามารถส่งออกได้ไม่เท่ากันโดยรวมทั้ง 2 ฟาร์มมีทั้งหมด 36 ต้น ต้องการส่งไปยังตัวแทนจำหน่าย A กับ B ทั้ง 2 รายก็มีความต้องการกล้วยไม้ไม่เท่ากันแต่รวมกันเท่ากับ 36 ต้น เช่นกัน โดยมีราคาในการขนส่งระหว่างแต่ละสถานที่ไม่เท่ากันโดยเริ่มจาก 1ไปA=220, 1ไปB=300, 2ไปA=400, 2ไปB=180 โดยจุดประสงค์ของการประยุกต์ใช้งานนี้คือการที่ฟาร์ม 1 กับ 2 สามารถส่งกล้วยไม้ไปยังตัวแทนจำหน่าย A กับ B ได้ครบทั้ง 36 ต้นโดยตัวแทนจำหน่ายทั้ง 2 ได้รับกล้วยไม้ทั้งหมดครบตามที่ต้องการและค่าขนส่งทั้งหมดไม่เกิน 10640 บาท

2. ภาพรวมขั้นตอนการทำงาน

- เริ่มต้นจะเป็นการกำหนดจำนวนของกล้วยไม้ในฟาร์ม 1 และ 2 โดยพื้ลงไปในตอนเริ่มต้นของโปรแกรมดังรูป
- ฟาร์ม 1 มีกล้วยไม้ทั้งหมด 28 ต้น
- ฟาร์ม 2 มีกล้วยไม้ทั้งหมด 8 ต้น
- โดยทั้งสองฟาร์มรวมกันจะมีทั้งหมด $28+8=36$ ต้น
- หลังจากนั้นก็ทำการกำหนดความต้องการของตัวแทนจำหน่าย A และ B
- ตัวแทนจำหน่าย A ต้องการกล้วยไม้ทั้งหมด 15 ต้น
- ตัวแทนจำหน่าย B ต้องการกล้วยไม้ทั้งหมด 21 ต้น

โดยทั้งสองตัวแทนจำหน่ายก็ต้องการกล้วยไม้ทั้งหมด $15+21=36$ ต้น เช่นกัน

```

1  import numpy as np
2
3  F1 = int(input('Enter total orchid in farm 1: '))
4  F2 = int(input('Enter total orchid in farm 2: '))
5  DA = int(input('Enter total need of dealer A: '))
6  DB = int(input('Enter total need of dealer B: '))
7

```

PROBLEMS	OUTPUT	TERMINAL	DEBUG CONSOLE
	Enter total orchid in farm 1: 20		
	Enter total orchid in farm 2: 16		
	Enter total need of dealer A: 18		
	Enter total need of dealer B: 18		

หลังจากนั้นโปรแกรมก็จะคำนวณโดยใช้ Gauss Jordan Elimination ว่าควรส่งกล้วยไม้จากฟาร์ม 1 และ 2 ไปยังตัวแทนจำหน่าย A และ B อย่างละกี่ต้นให้หมดทั้ง 36 ต้นโดยที่ไม่เกินค่าขนส่ง 10640 บาท ดังรูป

```

Gauss-Jordan-Transportation.py X
C: > Users > Mero Asebi > Desktop > Gauss-Jordan-Transportation.py > ...

52 print ('Farm 1 has a total of',P1,'orchids to send.')
53 print ('Send to dealer A amount',r[0][4],'orchids.')
54 print ('Send to dealer B amount',r[1][4],'orchids.\n')
55
56 print ('Farm 2 has a total of',P2,'orchids to send.')
57 print ('Send to dealer A amount',r[2][4],'orchids.')
58 print ('Send to dealer B amount',r[3][4],'orchids.\n')
59
60 print ('Dealer A gets total orchid is',r[0][4]+r[2][4])
61 print ('Dealer B gets total orchid is',r[1][4]+r[3][4],'\n')

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Farm 1 has a total of 28 orchids to send.
Send to dealer A amount 8 orchids.
Send to dealer B amount 20 orchids.

Farm 2 has a total of 8 orchids to send.
Send to dealer A amount 7 orchids.
Send to dealer B amount 1 orchids.

Dealer A gets total orchid is 15
Dealer B gets total orchid is 21

PS C:\Users\Mero Asebi>

```

3. รายละเอียดข้อมูลที่เกี่ยวข้อง

- เมทริกซ์ คือกลุ่มของจำนวนหรือสมาชิกของริงใดๆ เขียนเรียงกันเป็นรูปสี่เหลี่ยมผืนผ้าหรือจัตุรัส กล่าวคือเรียงเป็นแถวในแนวนอน และเรียงเป็นแถวในแนวตั้ง เรามักเขียนเมทริกซ์เป็นตารางที่ไม่มีเส้นแบ่งและเขียนวงเล็บคร่อมตารางไว้ (ไม่ว่าจะเป็นวงเล็บโค้งหรือวงเล็บเหลี่ยม) เช่น

$$\begin{bmatrix} 1 & 56 & 3 \\ 0 & 15 & 4 \\ 5 & -31 & -4 \end{bmatrix}$$

เราเรียกแถวในแนวนอนของเมทริกซ์ว่า แถว เรียกแถวในแนวตั้งของเมทริกซ์ว่า หลัก และเรียกจำนวนแต่ละจำนวนในเมทริกซ์ว่า สมาชิก ของเมทริกซ์ การกล่าวถึงสมาชิกของเมทริกซ์ จะต้องระบุตำแหน่งให้ถูกต้อง เช่น จากตัวอย่างข้างบน

สมาชิกที่อยู่ในแถวที่ 2 หลักที่ 3 คือเลข 4

สมาชิกที่อยู่ในแถวที่ 2 หลักที่ 2 คือเลข 15

สมาชิกที่อยู่ในแถวที่ 3 หลักที่ 1 คือเลข 5

เราเรียกเมทริกซ์ที่มี m แถวและ n หลักเรียกว่า เมทริกซ์ $m \times n$ เราเรียกจำนวน m และ n ว่ามิติหรือขนาด ของเมทริกซ์

เราใช้สัญลักษณ์ $A = (a_{i,j})_{m \times n}$ เพื่อหมายถึง เมทริกซ์ A ซึ่งมี m แถว และ n หลัก โดยที่ $a_{i,j}$ (หรือ a_{ij}) หมายถึง สมาชิกที่อยู่ในตำแหน่ง แถว i และ หลัก j ของเมทริกซ์

$$A = A_{m \times n} = \begin{bmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & \cdots & a_{2n} \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & \cdots & a_{mn} \end{bmatrix}$$

- Gauss Jordan Elimination การจัดรูปเมทริกซ์ให้อยู่ในรูปแบบขั้นบันไดลดรูป ซึ่งวิธีนี้ประกอบด้วย 2 ส่วน ส่วนแรกคือการทำให้สมาชิกนำมีค่าเป็น 1 และเลขที่อยู่ใต้สมาชิกนำมีค่าเป็นศูนย์ทั้งหมด และส่วนที่สองคือการทำให้เลขที่อยู่เหนือสมาชิกนำมีค่าเป็นศูนย์ทั้งหมด

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Python ภาษาโปรแกรม Python คือภาษาโปรแกรมคอมพิวเตอร์ระดับสูง โดยถูกออกแบบมาให้เป็นภาษาสคริปต์ที่อ่านง่าย โดยตัดความซับซ้อนของโครงสร้างและไวยากรณ์ของภาษาออกไป ในส่วนของการแปลงชุดคำสั่งที่เราเขียนให้เป็นภาษาเครื่อง Python มีการทำงานแบบ Interpreter คือเป็นการแปลชุดคำสั่งทีละบรรทัด เพื่อป้อนเข้าสู่หน่วยประมวลผลให้คอมพิวเตอร์ทำงานตามที่เราต้องการ นอกจากนั้นภาษาโปรแกรม Python ยังสามารถนำไปใช้ในการเขียนโปรแกรมได้หลากหลายประเภท โดยไม่ได้จำกัดอยู่ที่งานเฉพาะทางใดทางหนึ่ง (General-purpose language) จึงทำให้มีการนำไปใช้กันแพร่หลาย
- Numpy (Numeric Python) เป็นโมดูลส่วนเสริมของ Python ที่มีฟังก์ชันเกี่ยวกับคณิตศาสตร์และการคำนวณต่างๆ มาให้ใช้งาน โดยทั่วไปจะเกี่ยวกับการจัดการข้อมูลชุด (Array) ขนาดใหญ่และเมทริกซ์ NumPy นี้ครอบคลุมการคำนวณมากมายสามารถทำงานได้ใกล้เคียงกับ commercial software เช่น MatLab เลยทีเดียว

4. รายละเอียดขั้นตอนการคำนวณ (สามารถแสดงในรูปของการเขียนโปรแกรมแสดงการคำนวณ หรือการด้วย excel) พร้อมคำอธิบาย

เริ่มต้นจะขออธิบายการคำนวณโดยแยกเป็นตัวแปรดังนี้

x1 คือ จำนวนของกล้วยไม้ในฟาร์ม 1 ที่จะส่งไปยังตัวแทนจำหน่าย A

x2 คือ จำนวนของกล้วยไม้ในฟาร์ม 1 ที่จะส่งไปยังตัวแทนจำหน่าย B

x3 คือ จำนวนของกล้วยไม้ในฟาร์ม 2 ที่จะส่งไปยังตัวแทนจำหน่าย A

x4 คือ จำนวนของกล้วยไม้ในฟาร์ม 2 ที่จะส่งไปยังตัวแทนจำหน่าย B

r1 คือ $x_1 + x_2$ ก็คือจำนวนของกล้วยไม้ทั้งหมดที่ฟาร์ม 1 มี

r2 คือ $x_3 + x_4$ ก็คือจำนวนของกล้วยไม้ทั้งหมดที่ฟาร์ม 2 มี

r3 คือ $x_1 + x_3$ ก็คือจำนวนของกล้วยไม้ทั้งหมดที่ตัวแทนจำหน่าย A ต้องการ

r4 คือ $x_2 + x_4$ ก็คือจำนวนของกล้วยไม้ทั้งหมดที่ตัวแทนจำหน่าย B ต้องการ

r5 คือ ราคาค่าขนส่งของแต่ละสถานที่ $220x_1 + 300x_2 + 400x_3 + 180x_4 = 10640$

โดยที่ค่าขนส่งถูกจำกัดอยู่ที่ 10640 ไม่สามารถเปลี่ยนแปลงได้

เมื่อเอาข้อมูลข้างต้นมาทำเป็น Matrix ก็จะเป็นดังรูป

	x1	x2	x3	x4		
[1	1	0	0	F1]	r1
[0	0	1	1	F2]	r2
[1	0	1	0	DA]	r3
[0	1	0	1	DB]	r4
[220	300	400	180	10640]	r5

เนื่องจากโปรแกรมนี้มีบางส่วนที่ผู้ใช้สามารถกำหนดเองได้จึงมีบางส่วนที่เป็นตัวแปรเพิ่มขึ้นมาดังนี้

F1 คือจำนวนของกล้วยไม้ทั้งหมดที่ฟาร์ม 1 มี

F2 คือจำนวนของกล้วยไม้ทั้งหมดที่ฟาร์ม 2 มี

DA คือจำนวนของกล้วยไม้ทั้งหมดที่ตัวแทนจำหน่าย A ต้องการ

DB คือจำนวนของกล้วยไม้ทั้งหมดที่ตัวแทนจำหน่าย B ต้องการ

เมื่อนำ Matrix ข้างต้นมาเขียนเป็นโค้ดก็จะมีลักษณะดังรูป

```
import numpy as np

F1 = int(input('Enter total orchid in farm 1: '))
F2 = int(input('Enter total orchid in farm 2: '))
DA = int(input('Enter total need of dealer A: '))
DB = int(input('Enter total need of dealer B: '))

r = np.array([
    [1,1,0,0,F1],
    [0,0,1,1,F2],
    [1,0,1,0,DA],
    [0,1,0,1,DB],
    [220,300,400,180,10640]
])
```

เริ่มการคำนวณโดยในตัวอย่างนี้จะกรอกข้อมูลดังรูป

```
Enter total orchid in farm 1: 28
Enter total orchid in farm 2: 8
Enter total need of dealer A: 15
Enter total need of dealer B: 21
```

ฟาร์ม 1 มีกล้วยไม้ทั้งหมด 28 ต้น

ฟาร์ม 2 มีกล้วยไม้ทั้งหมด 8 ต้น

ตัวแทน A ต้องการกล้วยไม้ทั้งหมด 15 ต้น

ตัวแทน B ต้องการกล้วยไม้ทั้งหมด 21 ต้น

เมื่อกรอกข้อมูลครบแล้วในโปรแกรมก็จะแสดงการคำนวณดังนี้

$$1. r[2] = (-1*r[0]+r[2])$$

Result = [0 -1 1 0 -13]
 [[1 1 0 0 28]
 [0 0 1 1 8]
 [0 -1 1 0 -13]
 [0 1 0 1 21]
 [220 300 400 180 10640]]

$$1. (-1 * r1 + r3) \rightarrow r3$$

เอา -1 คูณแถว 1 บวกกับแถว 3 แล้วเอาผลลัพธ์ที่ไปแทนในแถว 3

$$2. r[4] = (-220*r[0]+r[4])$$

Result = [0 80 400 180 4480]
 [[1 1 0 0 28]
 [0 0 1 1 8]
 [0 -1 1 0 -13]
 [0 1 0 1 21]
 [0 80 400 180 4480]]

$$2. (-220*r1 + r5) \rightarrow r5$$

เอา -220 คูณแถว 1 บวกกับแถว 5 แล้วเอาผลลัพธ์ที่ไปแทนในแถว 5

$$3. r[[1, 3]] = r[[3, 1]]$$

Result = Swap r1 and r3
 [[1 1 0 0 28]
 [0 1 0 1 21]
 [0 -1 1 0 -13]
 [0 0 1 1 8]
 [0 80 400 180 4480]]

$$3. r2 \leftrightarrow r4$$

อันนี้คือการสลับแถวระหว่างแถว 2 กับแถว 4

$$4. r[2] = (r[1]+r[2])$$

Result = [0 0 1 1 8]
 [[1 1 0 0 28]
 [0 1 0 1 21]
 [0 0 1 1 8]
 [0 0 1 1 8]
 [0 80 400 180 4480]]

$$4. (r2 + r3) \rightarrow r3$$

เอา r2 บวกกับแถว 3 แล้วเอาผลลัพธ์ที่ไปแทนในแถว 3

$$5. r[4] = (-80*r[1]+r[4])$$

Result = [0 0 400 100 2800]
 [[1 1 0 0 28]
 [0 1 0 1 21]
 [0 0 1 1 8]
 [0 0 1 1 8]
 [0 0 400 100 2800]]

$$5. (-80*r2 + r5) \rightarrow r5$$

เอา -80 คูณแถว 2 บวกกับแถว 5 แล้วเอาผลลัพธ์ที่ไปแทนในแถว 5

$$6. r[3] = (-1*r[2]+r[3])$$

Result = [0 0 0 0 0]
 [[1 1 0 0 28]
 [0 1 0 1 21]
 [0 0 1 1 8]
 [0 0 0 0 0]
 [0 0 400 100 2800]]

$$6. (-1*r3 + r4) \rightarrow r4$$

เอา -220 คูณแถว 1 บวกกับแถว 5 แล้วเอาผลลัพธ์ที่ไปแทนในแถว 4

```

7. r[4] = (-400*r[2]+r[4])
Result = [ 0 0 0 -300 -400]
[[ 1 1 0 0 28]
 [ 0 1 0 1 21]
 [ 0 0 1 1 8]
 [ 0 0 0 0 0]
 [ 0 0 0 -300 -400]]

```

```

8. r[[3, 4]] = r[[4, 3]]
Result = Swap r3 and r4
[[ 1 1 0 0 28]
 [ 0 1 0 1 21]
 [ 0 0 1 1 8]
 [ 0 0 0 -300 -400]
 [ 0 0 0 0 0]]

```

```

9. r[3] = (-1/300*r[3])
Result = [0 0 0 1 1]
[[ 1 1 0 0 28]
 [ 0 1 0 1 21]
 [ 0 0 1 1 8]
 [ 0 0 0 1 1]
 [ 0 0 0 0 0]]

```

```

10. r[2] = (-1*r[3]+r[2])
Result = [0 0 1 0 7]
[[ 1 1 0 0 28]
 [ 0 1 0 1 21]
 [ 0 0 1 0 7]
 [ 0 0 0 1 1]
 [ 0 0 0 0 0]]

```

```

11. r[1] = (-1*r[3]+r[1])
Result = [ 0 1 0 0 20]
[[ 1 1 0 0 28]
 [ 0 1 0 0 20]
 [ 0 0 1 0 7]
 [ 0 0 0 1 1]
 [ 0 0 0 0 0]]

```

```

12. r[0] = (-1*r[1]+r[0])
Result = [1 0 0 0 8]
[[ 1 0 0 0 8]
 [ 0 1 0 0 20]
 [ 0 0 1 0 7]
 [ 0 0 0 1 1]
 [ 0 0 0 0 0]]

```

7. $(-400*r_3 + r_5) \rightarrow r_5$

เอา -400 คูณแถว 3 บวกกับแถว 5 แล้วเอาผลลัพธ์ไปแทนในแถว 5

8. $r_4 \leftrightarrow r_5$

อันนี้คือการสลับแถวระหว่างแถว 4 กับแถว 5

9. $(-1/300*r_4) \rightarrow r_4$

เอา -1/300 คูณแถว 4 แล้วเอาผลลัพธ์ไปแทนในแถว 4

10. $(-1*r_4 + r_3) \rightarrow r_3$

เอา -1 คูณแถว 4 บวกกับแถว 3 แล้วเอาผลลัพธ์ไปแทนในแถว 3

11. $(-1*r_4 + r_2) \rightarrow r_2$

เอา -1 คูณแถว 4 บวกกับแถว 2 แล้วเอาผลลัพธ์ไปแทนในแถว 2

12. $(-1*r_2 + r_1) \rightarrow r_1$

เอา -1 คูณแถว 2 บวกกับแถว 1 แล้วเอาผลลัพธ์ไปแทนในแถว 1

5. แสดงผลลัพธ์การทำงาน ที่ได้จากการคำนวณ ตาม ทฤษฎีคณิตศาสตร์ ที่เกี่ยวข้องในแต่ละขั้นตอน และอธิบายความหมายของผลลัพธ์ที่ได้

```
Farm 1 has a total of 28 orchids to send.
Send to dealer A amount 8 orchids.
Send to dealer B amount 20 orchids.
```

ฟาร์มที่ 1 มีกล้วยไม้ 28 ต้นให้จัดส่ง
ส่งให้ตัวแทนจำหน่าย A 8 ต้น
ส่งให้ตัวแทนจำหน่าย B 20 ต้น

```
Farm 2 has a total of 8 orchids to send.
Send to dealer A amount 7 orchids.
Send to dealer B amount 1 orchids.
```

ฟาร์มที่ 2 มีกล้วยไม้ 8 ต้นให้จัดส่ง
ส่งให้ตัวแทนจำหน่าย A 7 ต้น
ส่งให้ตัวแทนจำหน่าย B 1 ต้น

```
Dealer A gets total orchid is 15
Dealer B gets total orchid is 21
```

ตัวแทน A ได้รับต้นกล้วยไม้ไปจำหน่ายทั้งหมด 18 ต้น
ตัวแทน B ได้รับต้นกล้วยไม้ไปจำหน่ายทั้งหมด 18 ต้น

6. สรุปความน่าสนใจของการประยุกต์ใช้งานที่น่าเสนอ

จากที่ได้เรียนเกี่ยวกับ Gauss Jordan Elimination ก็รู้สึกว่าเป็นเรื่องที่น่าสนใจในการนำไปใช้งาน เนื่องจาก Gauss Jordan นั้นเป็นส่วนหนึ่งของการคำนวณต่างๆอยู่เสมอ จึงทำให้พวกเราอยากศึกษาหาข้อมูลว่า Gauss Jordan นั้นถูกเอาไปใช้ทำอะไรในชีวิตจริงบ้างทำให้พวกเราได้พบว่า Gauss Jordan นั้นสามารถเอามาใช้ในเรื่องของการขนส่งได้ ซึ่งจากตัวอย่างที่พวกเราเจอนั้นก็ค่อนข้างเป็นโจทย์ที่ทำให้สับสนได้ไม่น้อยเลยทีเดียวแต่เราสามารถใช่ Gauss Jordan เข้ามาแก้ปัญหานี้ได้อย่างง่ายดาย

7. แหล่งอ้างอิงข้อมูล

- Gauss-Jordan Application
<https://youtu.be/-0BkP5hhi0U>
- Numpy and Matplotlib
[https://www2.cs.science.cmu.ac.th/courses/204101/lib/exe/fetch.php?media=w13-
lec_-numpy.pdf](https://www2.cs.science.cmu.ac.th/courses/204101/lib/exe/fetch.php?media=w13-lec_-numpy.pdf)
- เมทริกซ์ (คณิตศาสตร์)
[https://th.wikipedia.org/wiki/เมทริกซ์_\(คณิตศาสตร์\)](https://th.wikipedia.org/wiki/เมทริกซ์_(คณิตศาสตร์))