การแก้สมการ LAPLACE ด้วย APPLE II

ชัยศิลป์ ชินพรเจริญพงศ์
อาจารย์ประจำภาควิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น

บทคัดย่อ

การแก้สมการ 2 มิติของ Laplace โดยใช้วิธี Iteration ของ Liebmanns มีตัวอย่างซึ่งประกอบด้วย Program Applesoft basic แก้ปัญหา steady state heat flow ชนิดง่าย

1. บทนำ

ในการจำลองปัญหาทางวิทยาศาสตร์ และวิศวกรรมเป็นไปได้ที่จะอยู่ในรูป Partial differential equation และหากวิเคราะห์ปัญหาดังกล่าวที่ steady state แล้ว สมการก็มักจะเหลือเป็นสมการ Laplace ต่อไปนี้จะเป็นวิธีแก้สมการ Laplace วิธีหนึ่งที่น่าสนใจ

2. สมการ DIFFERENCE[1]

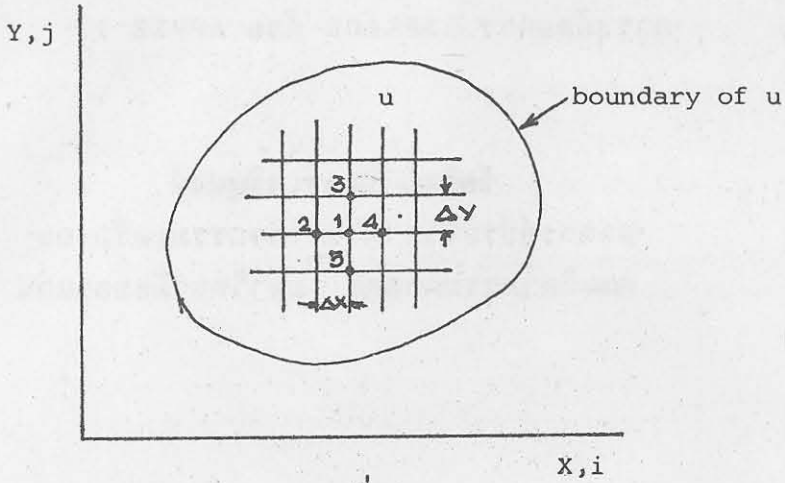
สมการ Laplace ซึ่งมี dependent variable, u และ independent variable x,y มีรูปเป็น

$$(\partial^2 u)/(\partial x^2) + (\partial^2 u)/(\partial y^2) = \Delta^2 u = 0 \quad \text{..........(1)}$$

สมการข้างบนสามารถเขียนอีกแบบหนึ่งเรียกว่าสมการ difference ดังนี้

$$\Delta^2 u_{i,j} = 1/h^2 \left| u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} \right| = 0 \quad \text{...........(2)}$$

ซึ่งได้มาจาก การแบ่งพื้นที่ใน boundary ของ u ออกเป็นช่อง ๆ เท่ากัน คือ $\Delta x = \Delta y = h$



รูปที่ 1

i และ j เป็นตัวเลขจำนวนเต็ม ใช้แสดงตำแหน่งของจุดตัด เช่น ถ้าจุด 1 คือ $u_{i,j}$ จุดอื่น ๆ จะเป็นดังนี้

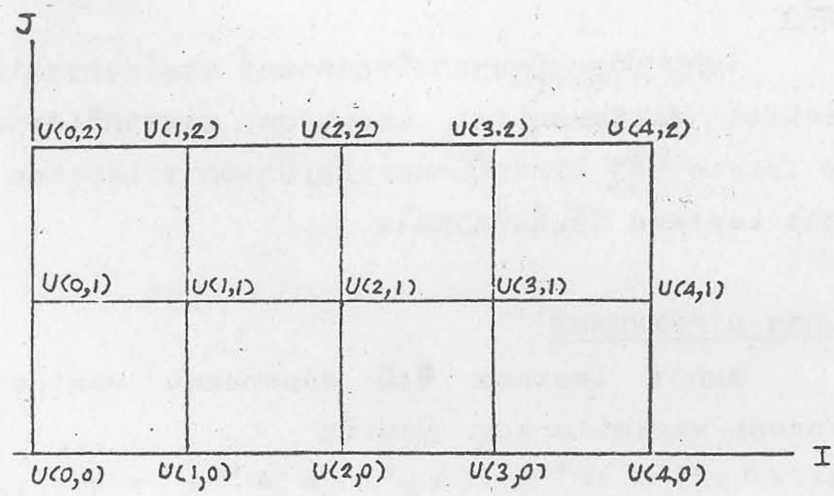
จุด 2 คือ $u_{i-1,j}$

จุด 3 คือ $u_{i+1,j}$

จุด 4 คือ $u_{i,j+1}$

จุด 5 คือ $u_{i,j-1}$

3 ITERATIVE METHOD



รูปที่ 2

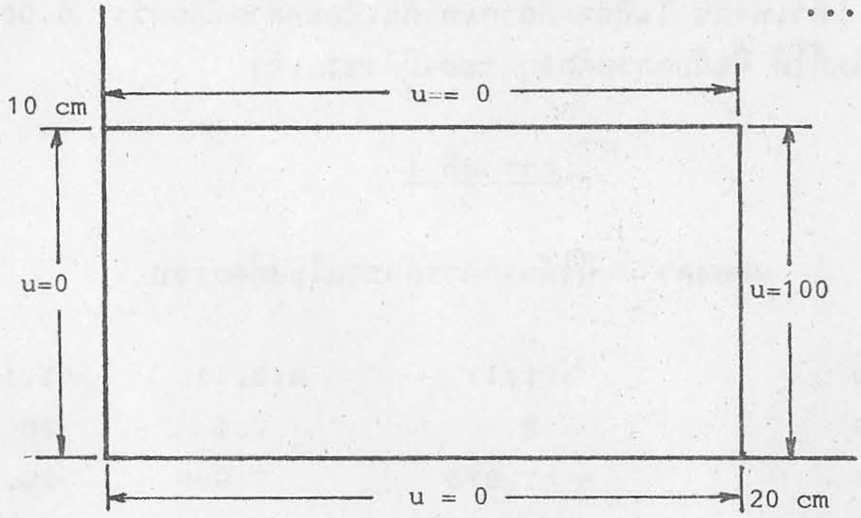
เราจะลองดูปัญหาของ Steady state heat flow ในแผ่นโลหะบางรูปสี่เหลี่ยมผืนผ้า ขนาด 10 ซม. x 20 ซม. ตามรูปที่ 2 ถ้าขอบด้าน 10 ซม. ด้านหนึ่งมีอุณหภูมิ 100 °C ส่วนด้านอื่น ๆ มีอุณภูมิ 0 °C สมมุติว่ามีการถ่ายเทความร้อนเข้าหรือออกทางขอบทั้ง 2 ด้าน ทั้งข้างบนและด้านข้างใต้ เมื่อ u แทนอุณหภูมิ

ที่จุดใด ๆ จะได้สมการดังนี้

$$(\partial^2 u)/(\partial x^2) + (\partial^2 u)/(\partial y^2) = 0$$

ก็คือสมการ Laplace นั่นเองและเพื่อความสะดวกจะเขียนสมการ (2) ใหม่เป็น

$$u(I,J) = 1/4 \, |u(I+1,J) + u(I-1,J) + u(I,J+1) + u(I,J-1)| \qquad \ldots\ldots\ldots\ldots(3)$$



รูปที่ 3

แบ่งแผ่นโลหะออกด้วย h=5 ซม. ตามรูปที่ 3 u ที่อยู่บนเส้น boundary นั้น รู้ค่าแล้วส่วน u อื่น ๆ ต้องกำหนดค่าเองเพื่อใช้เป็นค่าเริ่มต้นของการ iteration กำหนดให้

$$u(1,1) = 2 \quad u(2,1) = 7.5 \quad u(3,1) = 30$$

แทนค่า boundary condition ในสมการที่ 3

$$u(1,1) = u(2,2)/4$$

$$u(2,1) = \{u(1,1) + u(3,2)\}/4$$

$$u(3,1) = \{u(2,2) + 100\}/4$$

โดยวิธีของ Liebmanns จะเริ่มดังนี้

<u>รอบที่ 1</u>

$$u(1,1) = 7.5/4 = 1.875$$

$$u(2,1) = (1.875 + 30)/4 = 7.969$$

$$u(3,1) = (7.969 + 100)/4 = 26.992$$

<u>รอบที่ 2</u>

$$u(1,1) = 7.969/4 = 1.992$$

$$u(2,1) = (1.992 + 26.992)/4 = 7.246$$

$$u(3,1) = (7.246 + 100)/4 = 26.812$$

น่าสังเกตว่าในการคำนวณ u(2,1) จะเอา u(1,1)=1.875 ที่เพิ่ง

คำนวณได้มาใช้เลย และในการคำนวณ u(3,1) ก็จะใช้ u(2,1) ตัวใหม่คือ 7.969 เช่นกัน

เมื่อคำนวณผ่านไป 1 รอบ ได้ค่า u มาชุดหนึ่งคือ 1.875, 7969, 26.992 ก็จะนำค่า เหล่านี้ไปคำนวณในรอบที่ 2 และต่อไปในทำนองเดียวกัน จนกระทั่งถึงรอบที่ n ซึ่งจะหยุดการคำนวรเมื่อค่าของ u ที่แต่ละจุดในรอบที่ n และรอบที่ n-1 มีค่าเท่ากัน ในที่นี้จะถือว่าเท่ากันเมื่อผลต่างน้อยกว่า 0.0011 ผลต่างนี้จะใช้เท่าใดก็ได้ ขึ้นกับความต้องการของผู้คำนวณเอง

ตารางที่ 1

แสดงค่า u ที่ได้จากการคำนวณในแต่ละรอบ

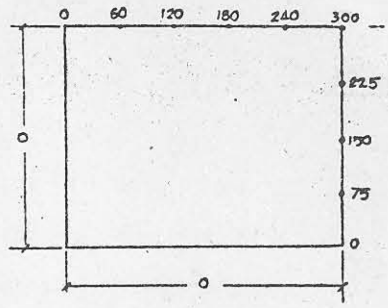
n	u(1,1)	u(2,1)	u(3,1)
0	2	7.5	30
1	1.875	7.969	26.992
2	1.992	7.246	26.812
3	1.812	7.156	26.789
4	1.789	7.144	26.786
5	1.786	7.143	26.786
6	1.786	7.143	26.786

ตารางที่ 1 แสดงค่า u ที่คำนวณได้ เห็นได้ว่ามีการคำนวณ 6 รอบ แต่ถ้ากำหนดค่าเริ่มต้น และค่าสุดท้ายห่างไกลกันกว่านี้ หรือ ถ้าแบ่งจุดให้มีจำนวนมาก ๆ จำนวนรอบที่ใช้คำนวณก็จะมากขึ้น อย่างไรก็ดีเราสามารถใช้คอมพิวเตอร์คำนวณเช่นนี้ได้ ในที่นี้จะใช้ Apple II ทำงานเหล่านี้แทน

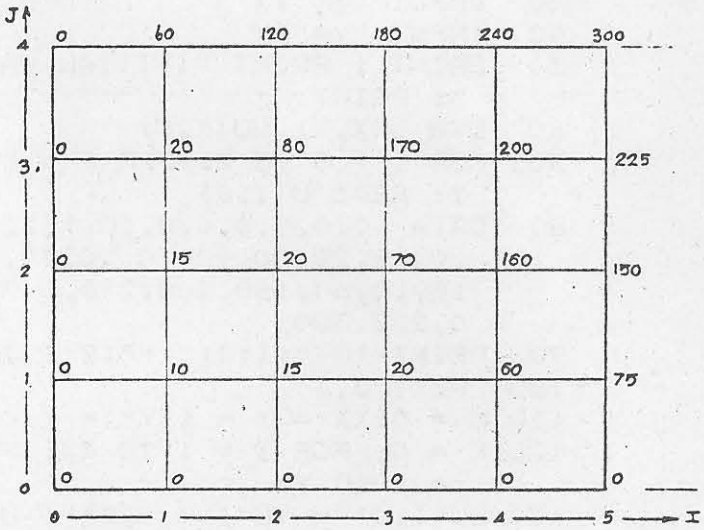
ถ้ามีแผ่นสี่เหลี่ยมขนาด 12x15 นิ้ว อุณหภูมิที่ขอบแผ่นเป็นดังกำหนดในรูปที่ 4 ต้องการหาอุณหภูมิที่จุดตัดต่าง ๆ โดยใช้ $\Delta x = \Delta y = 3$ นิ้ว

เริ่มแรก กำหนดค่าเริ่มต้นสำหรับจุดต่าง ในรูปที่ 5 จากนั้นก็ลำดับอุณหภูมิของจุดต่าง ๆ เข้าในโปรแกรมบรรทัดที่ 80

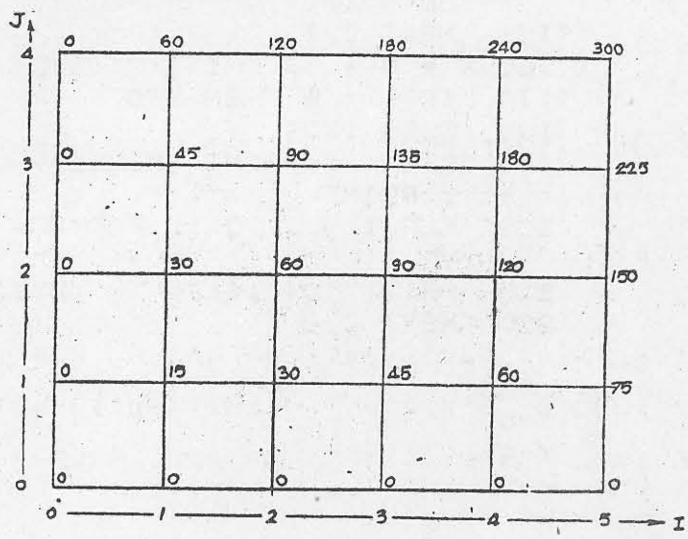
เมื่อเริ่มวิ่งโปรแกรมเครื่องจะถามค่า x และ y ซึ่งเป็นจำนวนจุดบนแกน x และ y ในที่นี้ x= 5 และ y = 4 เพราะนับเริ่มต้นจาก 0 เครื่องจะใช้เวลาประมาณครึ่งนาที คำนวณ 21 รอบ ได้ผลออกมาตามตารางที่ 2 และรูปที่ 6



รูปที่ 4 แผ่นสี่เหลี่ยมขนาด 12 นิ้ว x 15 นิ้ว



รูปที่ 5 กำหนด initial values



รูปที่ 6 แสดงค่า final values ที่จุดต่าง ๆ

4) <u>โปรแกรม Applesoft</u>

```
]LIST

10  REM  A program to solve LAPLA
     CES eqn. on rectangular regi
     on by LIEBMANNS METHOD
20  REM  X Number of point in x-d
     irection          Y Number of
     point in y-direction
     N Number of iterations
30  INPUT "X=";X
40  INPUT "Y=";Y
50  PRINT : PRINT "INITIAL VALUES
     ": PRINT
60  DIM U(X,Y),UU(X,Y)
70  FOR I = 0 TO X: FOR J = 0 TO
     Y: READ U(I,J)
80  DATA  0,0,0,0,0,0,10,15,20,60
     ,0,15,20,80,120,0,20,70,170,
     180,0,60,160,200,240,0,75,15
     0,225,300
90  PRINT "U(";I;J;")=";U(I,J)
100  NEXT J,I
110 N = 0:XX = X - 1:YY = Y - 1
120 K = 0: FOR I = 1 TO XX: FOR J
     = 1 TO YY
130 UU(I,J) = (U(I + 1,J) + U(I -
     1,J) + U(I,J + 1) + U(I,J -
     1)) / 4
140  IF  ABS (U(I,J) - UU(I,J)) >
     .0001 THEN K = K + 1:U(I,J) =
     UU(I,J)
150  NEXT J,I
160 N = N + 1: PRINT N; SPC( 2);
170  IF K = 0 THEN 190
180  GOTO 120
190  PRINT : PRINT "FINAL VALUES"
     : PRINT
200  FOR I = 0 TO X: FOR J = 0 TO
     Y
210  PRINT "U(";I;J;")=";U(I,J)
220  NEXT J,I

]RUN
X=5
Y=4
```

ตารางที่ 2

แสดงค่า intital values และ final values

INITIAL VALUES	FINAL VALUES
U(00)=0	U(00)=0
U(01)=0	U(01)=0
U(02)=0	U(02)=0
U(03)=0	U(03)=0
U(04)=0	U(04)=0
U(10)=0	U(10)=0
U(11)=10	U(11)=15.0001466
U(12)=15	U(12)=30.0001572
U(13)=20	U(13)=45.0001466
U(14)=60	U(14)=60
U(20)=0	U(20)=0
U(21)=15	U(21)=30.0001799
U(22)=20	U(22)=60.0001929
U(23)=80	U(23)=90.0001799
U(24)=120	U(24)=120
U(30)=0	U(30)=0
U(31)=20	U(31)=45.0002373
U(32)=70	U(32)=90.0002544
U(33)=170	U(33)=135.000237
U(34)=180	U(34)=180
U(40)=0	U(40)=0
U(41)=60	U(41)=60.0001935
U(42)=160	U(42)=120.000207
U(43)=200	U(43)=180.000194
U(44)=240	U(44)=240
U(50)=0	U(50)=0
U(51)=75	U(51)=75
U(52)=150	U(52)=150
U(53)=225	U(53)=225
U(54)=300	U(54)=300

5) สรุปและวิจารณ์

หลักสำคัญของ Liebmanns method คือในรอบการคำนวณเดียวกันจะเอาค่า u ที่คำนวณก่อนไปใช้คำนวณ u ตัดถัดไป เป็นการเพิ่มความเร็วสู่ค่าสุดท้าย

กรณีที่ใช้ค่า h เล็กลง จำนวนจุดและเวลาที่ใช้คำนวณจะมากขึ้น วิธีการหรือโปรแกรมที่มีนี้อาจจะไม่เหมาะสม การลดเวลาทางหนึ่งทำได้โดยเพิ่มขนาดของผลต่างให้ใหญ่ขึ้น เช่น 0.01

6) เอกสารอ้างอิง

1. M.L. James, G.M. 20 Smith, and J.C. Wolford, "Applied Numerical for Digital Computation", 2nd Ed., Harper and Row, Publishers, Inc, pp 525-537, 1977