

บทที่ 3

แบบจำลองการไหลของน้ำใต้ดินที่ได้รับการพัฒนาขึ้น

3.1 ทฤษฎี Finite Difference ที่ใช้กับแบบจำลอง

จากสมการการไหลของน้ำใต้ดินแบบไม่คงตัวในสองมิติ ในชั้นอุ้มน้ำแบบปิดและรั่วซึม มี Source และ Sink (สมการ 2.18) คือ

$$\nabla \cdot (T \nabla h) + \frac{Q_o}{a} - R - \left[\frac{K'}{b'} \right] (h - h') = S \frac{\partial h}{\partial t} \quad (3.1)$$

ในการประมาณค่า Finite Difference ให้กับสมการ Partial Differential Equation ในการศึกษา ใช้การประมาณโดยการอินทิเกรตสมการ Partial Differential ของการไหลของน้ำใต้ดินแบบไม่คงตัวในสองมิติ คือ สมการ(3.1) โดยรอบ Control Volume ซึ่งแสดงดังเส้นประในภาพที่ 3.2

จากสมการ (3.1) โดยการ Integrate ทั้งสองข้างโดยรอบ Control Volume ได้เป็น

$$\int_a \nabla \cdot (T \nabla h) da + \int_a \left[\frac{Q_o}{a} - R - \frac{K'}{b'} (h - h') \right] da = \int_a \left(S \frac{\partial h}{\partial t} \right) da \quad (3.2)$$

โดยที่ a คือพื้นที่หน้าตัดของ Control Volume

จากทฤษฎีไดเวอร์เจนซ์ของเกาส์ (Divergence theorem of Gauss) จะได้ว่า (O' Neil, 1993, p.929)

$$\int_a \nabla \cdot (T \nabla h) da = \int_{\Gamma} (T \nabla h) \cdot \hat{n} dc \quad (3.3)$$

เมื่อ Γ คือ เส้นขอบของ Control Volume
 $\hat{n}$ คือ unit vector ที่ตั้งฉากกับเส้นขอบ แสดงตัวอย่าง ในภาพที่ 3.3
 c คือ segment ของเส้นขอบ

ตามคำจำกัดความของ Direction Derivative (O' Neil, 1993, p.859) จะได้ว่า

$$(T \nabla h) \cdot \hat{n} = T \frac{dh}{dn} \quad (3.4)$$

จากสมการ (3.3) และ (3.4) จะได้

$$\int_a^b \nabla \cdot (T \nabla h) da = \int_a^b T \frac{dh}{dn} dc \quad (3.5)$$

จากสมการ (3.5) แทนในสมการ (3.2) จะได้เป็น

$$\int_a^b T \frac{dh}{dn} dc + \int_a^b \left[\frac{Q_o}{a} - R - \frac{K'}{b'} (h - h') \right] da = \int_a^b \left(S \frac{\partial h}{\partial t} \right) da \quad (3.6)$$

จากสมการ (3.6) สามารถแสดงค่าโดยประมาณในเทอมของตัวแปรจากจุด 0 ถึง 4 ดังภาพที่ 3.2 ได้ดังนี้

$$\sum_{i=1}^4 \frac{1}{2} (T_o + T_i) \frac{(h_i - h_o)}{\Delta} \Delta - \left[\left(\frac{K'}{b'} \right) (h_o - h_o') - R \right] \Delta^2 + Q_o = S \Delta^2 \frac{dh_o(t)}{dt} \quad (3.7)$$

โดยที่ Δ คือ ระยะห่างระหว่าง Nodes และ $dc \approx dn \approx \Delta$

จากสมการ (3.7) จัดรูปสมการให้อยู่ในรูปแบบตัวแปร h ที่ง่ายและเหมาะสำหรับการคำนวณโดยวิธี Finite Difference ได้ดังนี้

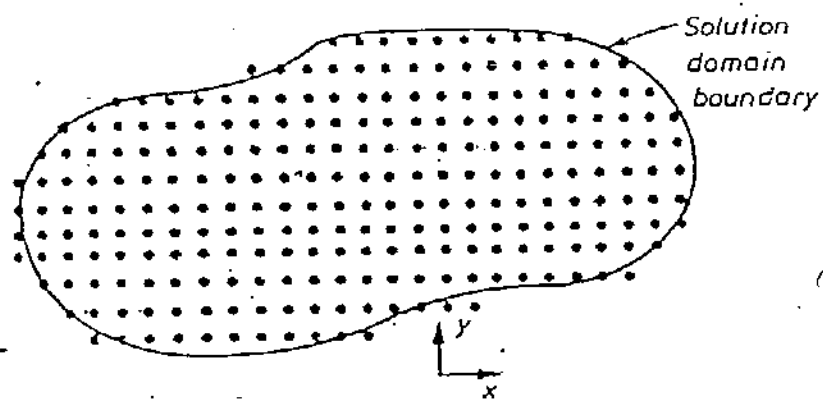
$$A_1 h_1 + A_2 h_2 + A_3 h_3 + A_4 h_4 - A_0 h_0 + B_0 = \frac{S \Delta^2 h_o(t) - h_o(t - \Delta t)}{\Delta t} \quad (3.8)$$

$$\text{เมื่อ } A_i = \frac{1}{2} (T_o + T_i) \text{ for } i=1,2,3 \text{ และ } 4 \quad (3.9a)$$

$$A_o = A_1 + A_2 + A_3 + A_4 + \left(\frac{K'}{b'} \right) \Delta^2 \quad (3.9b)$$

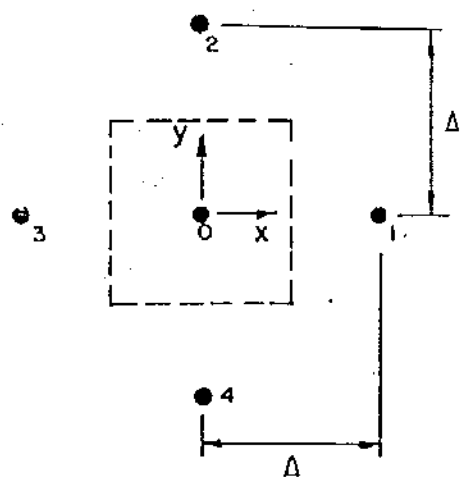
$$B_o = \left[\left(\frac{K'}{b'} \right) h_o' + R \right] \Delta^2 - Q_o \quad (3.9c)$$

ซึ่งสมการ (3.8) คือสมการ Finite Difference ที่จะนำไปแทนลงในแต่ละจุดหรือ Nodes ของ Finite Difference Grid ที่อยู่ภายใน Problem Domain (Internal Nodes) เพื่อทำการแก้สมการหาค่าตัวแปร h โดยแต่ละ Node จะมีสมการ Finite Difference ดังกล่าว 1 สมการ และมีความสัมพันธ์ต่อเนื่องกันภายในแบบ 5 Nodes ดังภาพที่ 3.2 ซึ่งได้มาจากการแบ่งกริดย่อยของพื้นที่ จากภาพที่ 3.1



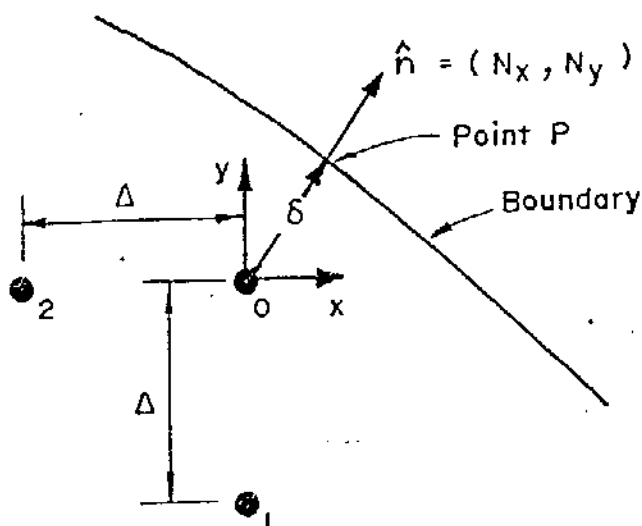
QW
TC
K286

ภาพที่ 3.1 แสดงการแบ่ง Finite Difference Grid



ภาพที่ 3.2 แสดงความสัมพันธ์ของ Internal Nodes

สำหรับ Nodes ที่อยู่บนเส้นหรือใกล้เส้นขอบเขตของ Solution Domain ซึ่งเรียกว่า Boundary Nodes นั้น จะมีความสัมพันธ์ต่างจาก Internal Nodes ตรงที่ไม่เป็นไปตามความสัมพันธ์ของแบบ 5 Nodes แต่จะมีความสัมพันธ์กันแบบ 3 Nodes แสดงดังภาพที่ 3.3



ภาพที่ 3.3 แสดงความสัมพันธ์ของ Boundary Nodes

สำหรับ Boundary Nodes นี้ Hunt (1978) ได้เสนอสมการเพื่อใช้ในการประมาณค่าของ Boundary Conditions ไว้ดังนี้

$$\alpha \frac{dh}{dn} + \beta h = F \quad (3.10)$$

ซึ่ง α, β และ F เป็น Discontinuous Function ที่มีค่าขึ้นอยู่กับชนิดของ Boundary เช่น Boundary เป็นชั้นดินที่ไม่ยอมให้น้ำซึมผ่าน (Impervious Layer) ซึ่ง $\frac{dh}{dn} = 0$ จะทำให้ α มีค่า = 1, $\beta = 0$ และ $F = 0$

ถ้า Boundary เป็นขอบเขตที่สามารถรู้ค่าเฮดชลศาสตร์ เช่น h จะทำให้ $\alpha = 0$, $\beta = 1$ และ $F = h$

สมการ (3.10) คือสมการที่จะใช้แทนค่าลงไปบนแต่ละจุดที่ต่อเนื่องกันของ Boundary Nodes ซึ่งการประมาณค่าสมการ จะสามารถหาได้จากความสัมพันธ์กันแบบ 3 จุดของ Boundary Nodes ตามภาพที่ 3.3 โดยใช้สมการการกระจายแบบ First-Degree Polynomial ดังนี้ (Hunt, 1978)

$$h(x, y) = h_o + \frac{h_o - h_2}{\Delta} x + \frac{h_o - h_1}{\Delta} y + O(\Delta^2) \quad (3.11)$$

การหาค่าสัมประสิทธิ์ในสมการ (3.11) สามารถหาได้โดยการแทนค่าสมการ Polynomial ที่จุด 0, 1 และ 2 ซึ่งสามารถตรวจสอบผลได้ โดยการกำหนดพิกัด (X,Y) เท่ากับ (0,0), (0, -Δ) และ (-Δ,0) ตามลำดับ ให้ P เป็นจุดอยู่บนเส้นขอบเขตที่อยู่ใกล้จุด 0 มากที่สุด $\hat{n} = (N_x, N_y)$ เป็นเวกเตอร์ 1 หน่วยที่ตั้งฉากกับ Boundary และ δ เป็นระยะทางระหว่างจุด 0 กับ P ซึ่งจะมีค่าเป็นบวกหรือลบขึ้นอยู่กับว่าจุด 0 จะอยู่ภายใน หรือภายนอก Solution Domain ดังนั้นจากสมการ (3.11)

$$\left[\frac{dh}{dn} \right]_P = \left[\nabla h \cdot \hat{n} \right]_P = \frac{h_0 - h_2}{\Delta} N_x + \frac{h_0 - h_1}{\Delta} N_y \quad (3.12)$$

$$(h)_P = h_0 + \frac{h_0 - h_2}{\Delta} N_x \delta + \frac{h_0 - h_1}{\Delta} N_y \delta \quad (3.13)$$

โดยที่ N_x คือ องค์ประกอบของ Unit Vector $\hat{n}$ ในแนวแกน x

N_y คือ องค์ประกอบของ Unit Vector $\hat{n}$ ในแนวแกน y

จากนั้นแทนสมการ (3.12) และ (3.13) ลงในสมการ (3.10) จะได้

$$A_0 h_0 - A_1 h_1 - A_2 h_2 = F_P \quad (3.14)$$

ซึ่ง ค่า F_P คือ ค่าของ F ที่จุด P และ

$$A_1 = (\alpha + \beta \delta) \frac{N_y}{\Delta} \quad (3.15a)$$

$$A_2 = (\alpha + \beta \delta) \frac{N_x}{\Delta} \quad (3.15b)$$

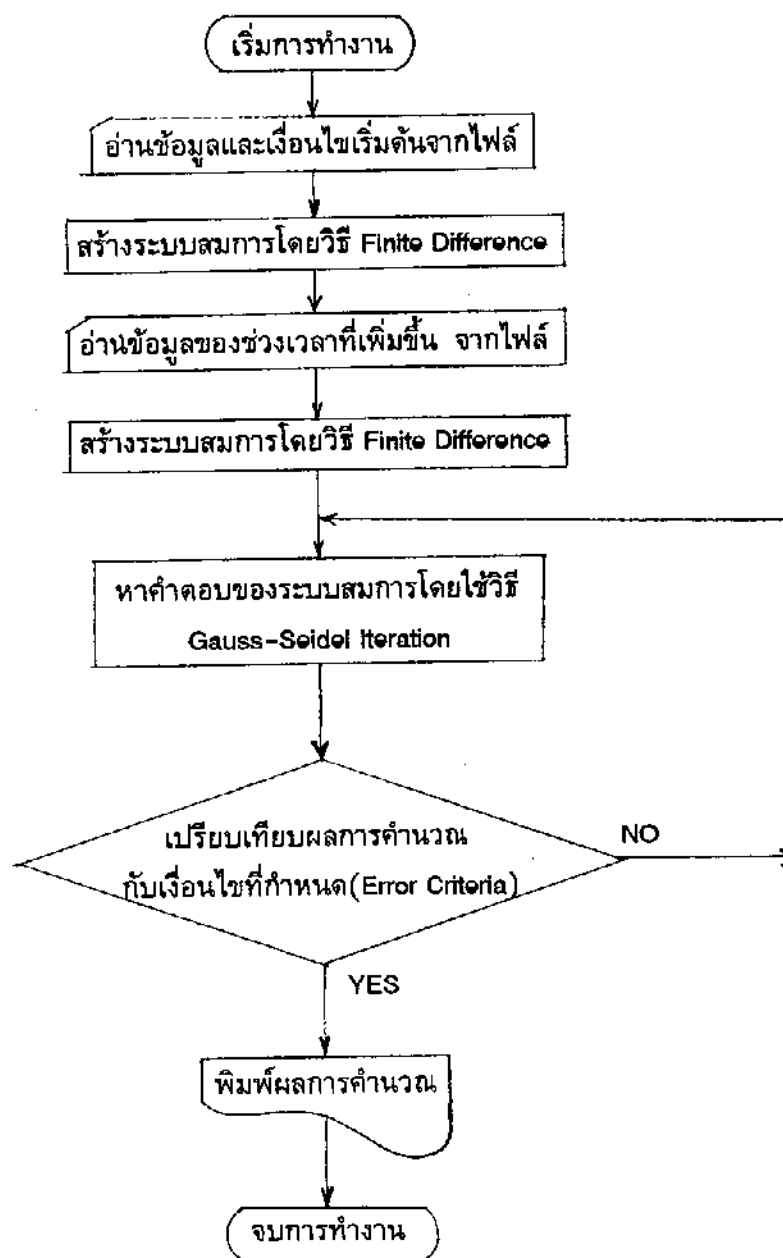
$$A_0 = A_1 + A_2 + \beta \quad (3.15c)$$

ซึ่งสมการ (3.14) คือสมการ Finite Difference ที่ใช้สำหรับประมาณค่าของตัวแปร h ของแต่ละ Nodes ทุกๆ Nodes บน Boundary Nodes

ในการหาผลเฉลยของสมการ Finite Difference ทั้ง Internal Nodes (สมการ 3.8) และ Boundary Nodes (สมการ 3.14) โดยมีค่า head ในแต่ละ Nodes เป็น Unknown เนื่องจากจำนวนสมการและจำนวน Unknown มีจำนวนมาก ดังนั้น วิธีที่เหมาะสมที่สุดคือใช้วิธีการแก้สมการแบบ Iterative Method ซึ่งในการศึกษานี้ได้ใช้วิธีการของ Gauss-Seidel Iteration โดยเขียนโปรแกรมการคำนวณในภาษา FORTRAN 77 เพื่อใช้ในการแก้สมการดังกล่าว

3.2 โปรแกรมคอมพิวเตอร์ (Computer Program)

ในการแก้สมการ Finite Difference Equation ของสมการ(3.8) และสมการ(3.14) เพื่อหาค่าตัวแปร h โดยวิธี Gauss-Seidel Iteration ได้เขียนเป็นโปรแกรมคอมพิวเตอร์ในภาษาFORTRAN77 โดยมีขั้นตอนการคำนวณ ดังภาพที่ 3.4



ภาพที่ 3.4 แสดงขั้นตอนการคำนวณในโปรแกรมคอมพิวเตอร์

3.2.1 สัญลักษณ์ที่ใช้ในโปรแกรม (Computer Code)

ในการกำหนดใน Computer Code เพื่อให้สะดวกต่อ Nodes แบบต่างๆ จะกำหนด ดังนี้

1. จุดที่เป็น Boundary Nodes ให้มีค่าตั้งแต่ 1 ถึง NB และสำหรับ Internal Nodes ให้มีค่าตั้งแต่ NB+1 ถึง N

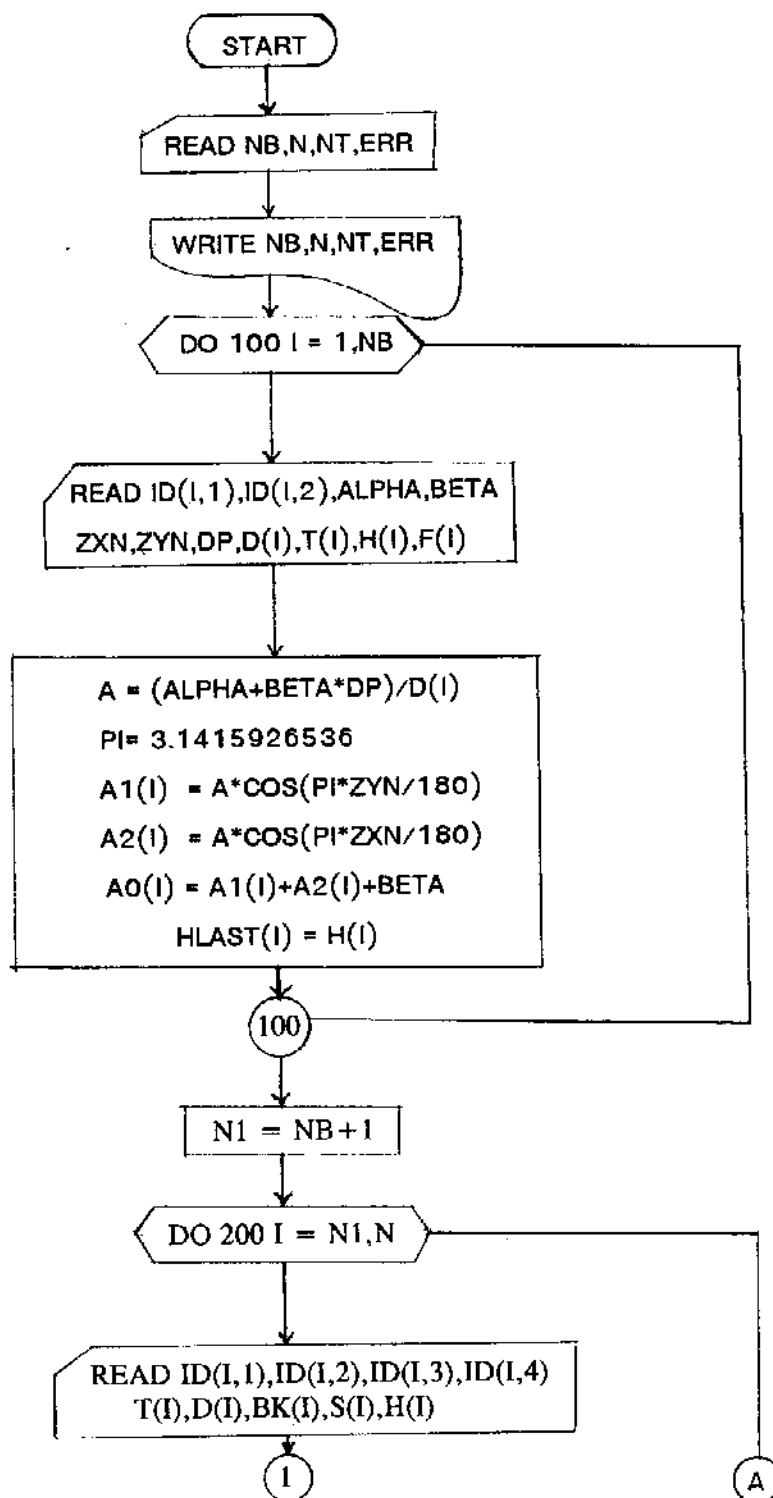
2. กำหนดจุดเป็นตัวเลขจำนวนเต็มบวกถ้าเป็น Internal Nodes ก็จะมี 4 Integers อันล้อมรอบซึ่งเรียกว่า ID (I,J) โดยที่ J จะนับจาก 1 ถึง 4 ถ้าเป็น Boundary Node จะมี 2 Integers ล้อมรอบ J จะนับจาก 1 ถึง 2

สัญลักษณ์ที่ใช้ในโปรแกรม มีดังนี้

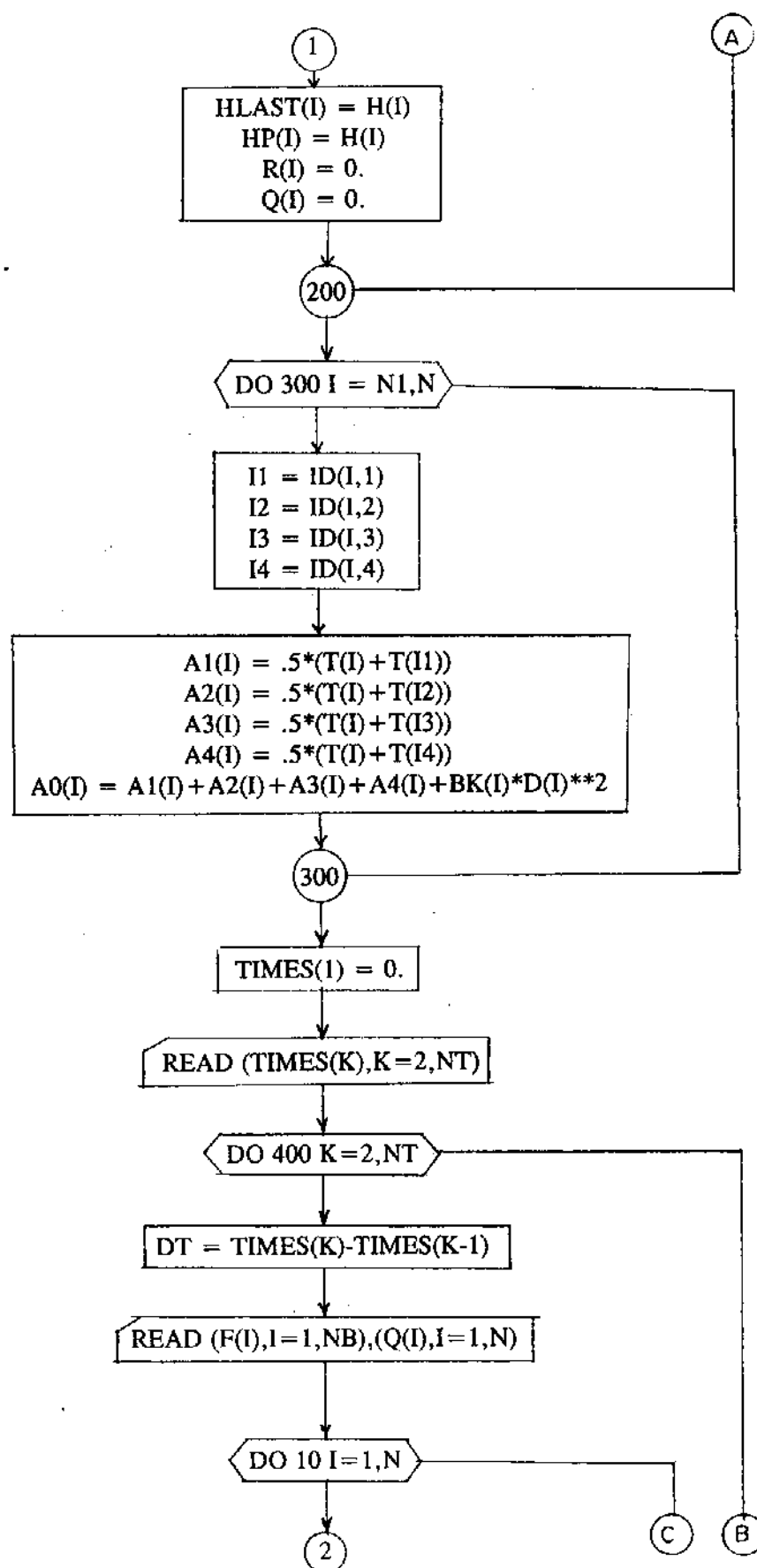
NB	=	จำนวนของ Boundary Points
N	=	จำนวนของจุดทั้งหมดบน Solution Domain
ERR	=	Error Criteria
ID(I,J)	=	หมายเลขประจำ Nodes
ALPHA	=	α ในสมการ (3.10)
BETA	=	β ในสมการ (3.10)
F(I)	=	F_p ในสมการ (3.14)
ZXN	=	N_x ในสมการ (3.15)
YZN	=	N_y ในสมการ (3.15)
DP	=	δ ในสมการ (3.15)
D(I)	=	Δ ในสมการ (3.15) หรือ สมการ(3.9)
T(I)	=	T ในสมการ (3.7)
BK(I)	=	$\frac{K'}{b'}$ ในสมการ (3.7)
HP(I)	=	h_0' ในสมการ (3.7)
R(I)	=	R ในสมการ (3.7)
Q(I)	=	Q_0 ในสมการ (3.7)
H(I)	=	h
S(I)	=	Storage Coefficient ที่จุด I
HLAST	=	ค่าของ H(I) ที่ได้จากช่วงเวลาสุดท้าย
TIMES(K)	=	ค่าของแต่ละช่วงเวลา
NT	=	จำนวนของช่วงเวลา
DT	=	ช่วงเวลา = TIMES(K)-TIMES(K-1)

3.2.2 Flowchart ในการคำนวณ

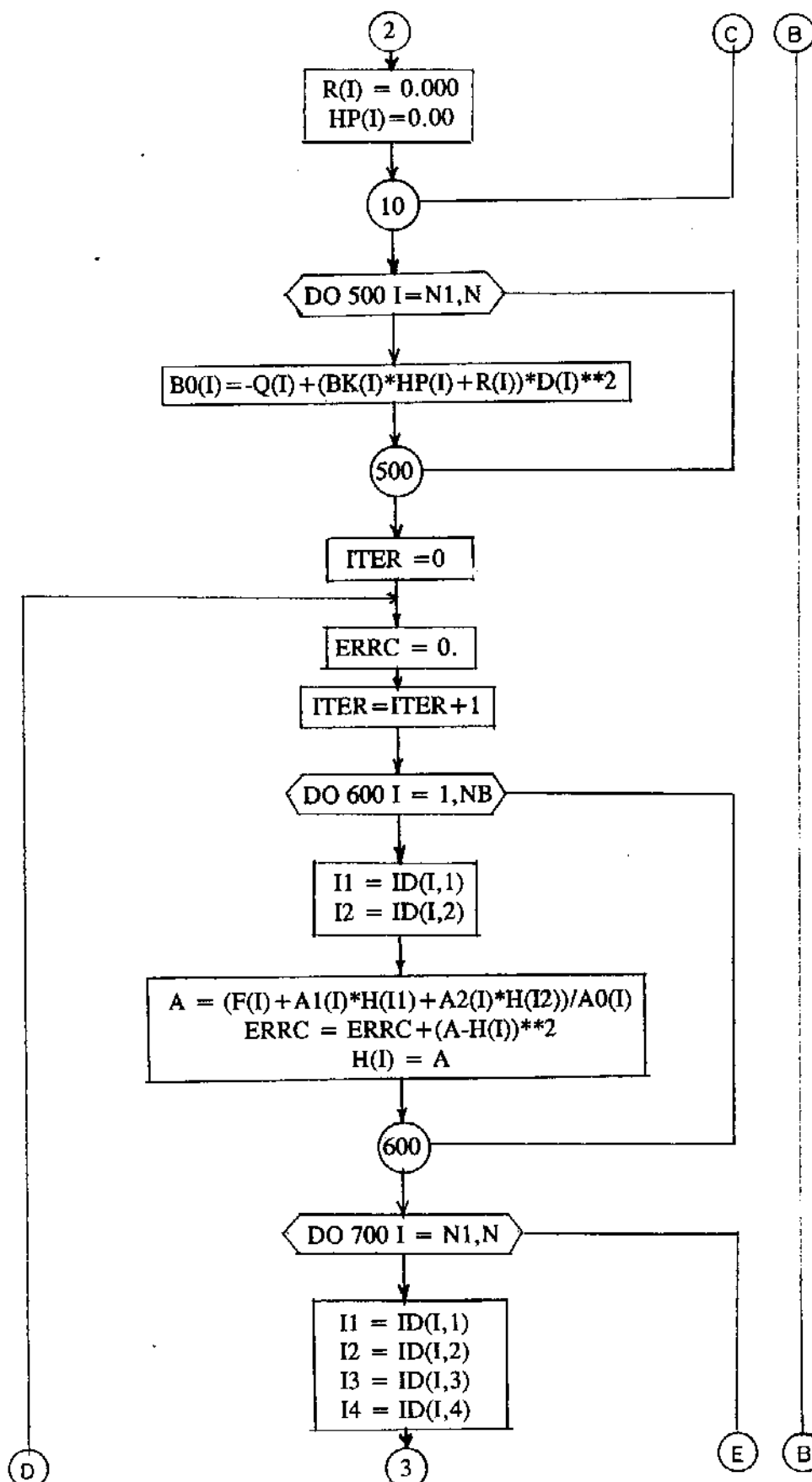
Flowchart การคำนวณใน Computer Program แสดงดังภาพที่ 3.5



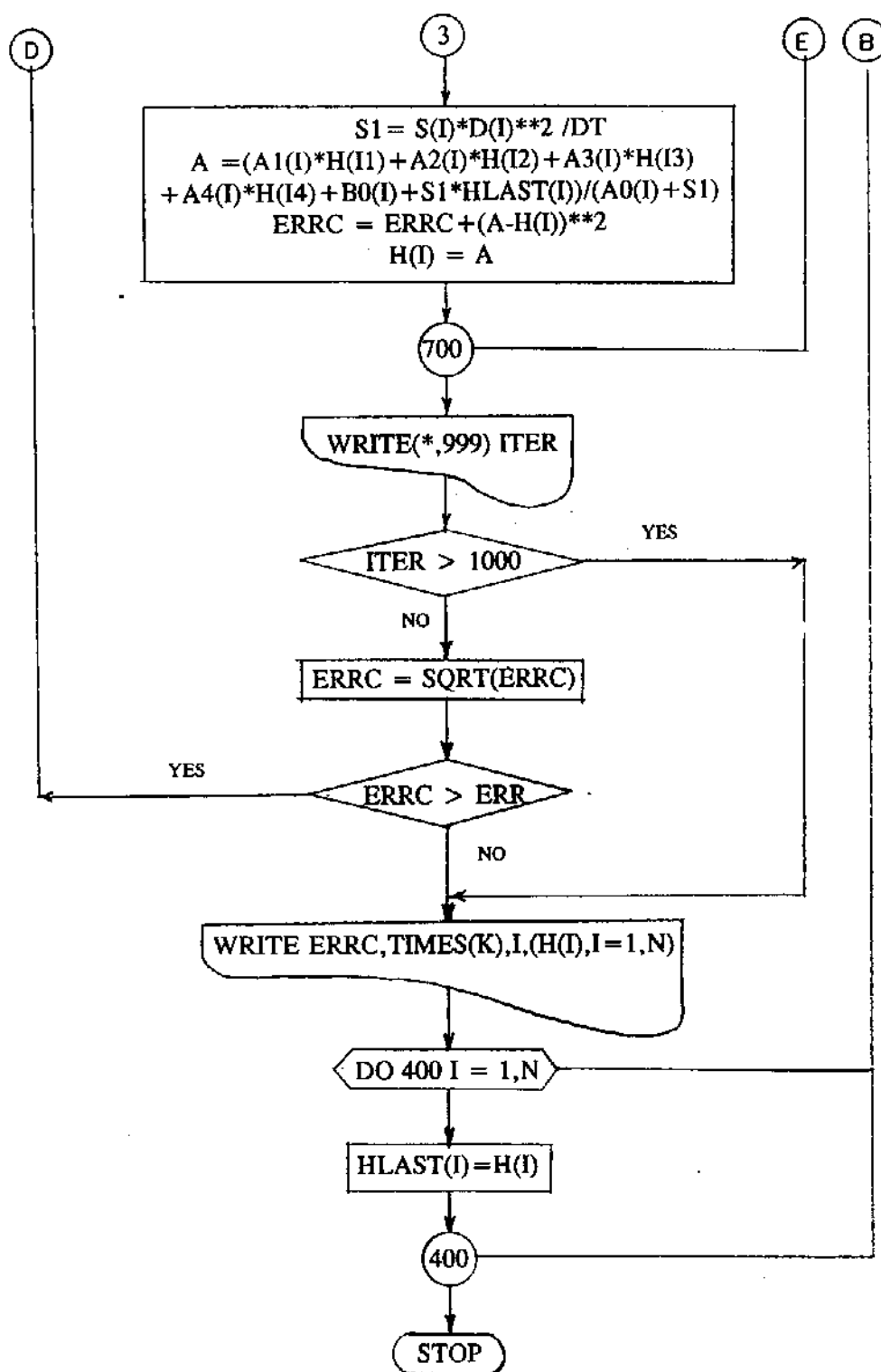
ภาพที่ 3.5 แสดง Flowchart ของโปรแกรมการคำนวณ



ภาพที่ 3.5 แสดง Flowchart ของโปรแกรมการคำนวณ (ต่อ)



ภาพที่ 3.5 แสดง Flowchart ของโปรแกรมการคำนวณ (ต่อ)



ภาพที่ 3.5 แสดง Flowchart ของโปรแกรมการคำนวณ (ต่อ)

3.2.3 โปรแกรมคอมพิวเตอร์

รายละเอียดโปรแกรมคอมพิวเตอร์ GW.FOR แสดงดังภาพที่ 3.6

```

DIMENSION ID(550,4),D(550),T(550),H(550),BK(550)
DIMENSION HLAST (550),HP(550),R(550),Q(550),F(550)
DIMENSION S(550),TIMES(550),A1(550),A2(550),A3(550)
DIMENSION A4(550),A0(550),B0(550)
READ(1,*) NB,N,NT,ERR
WRITE(2,*) NB,N,NT,ERR
DO 100 I = 1,NB
  READ(1,*)ID(I,1),ID(I,2),ALPHA,BETA,ZXN,ZYN,DP,
  *   D(I),T(I),H(I),F(I)
30  FORMAT (1X,2I4,2F6.1,2F7.2,2F8.2,3F6.2)
  A = (ALPHA+BETA*DP)/D(I)
  PI= 3.1415926536
  A1(I) = A*COS(PI*ZYN/180)
  A2(I) = A*COS(PI*ZXN/180)
  A0(I) = A1(I)+A2(I)+BETA
  HLAST(I) = H(I)
100  CONTINUE
  N1 = NB+1
  DO 200 I =N1,N
    READ(1,*)ID(I,1),ID(I,2),ID(I,3),ID(I,4),T(I),D(I),
    *   BK(I),S(I),H(I)
40  FORMAT (1X,4I8,5F7.2)
    HLAST(I) = H(I)
    HP(I) = H(I)
    R(I) = 0.
    Q(I) = 0.
200  CONTINUE
    DO 300 I = N1,N
      I1 = ID(I,1)
      I2 = ID(I,2)
      I3 = ID(I,3)
      I4 = ID(I,4)
      A1(I) = .5*(T(I)+T(I1))
      A2(I) = .5*(T(I)+T(I2))
      A3(I) = .5*(T(I)+T(I3))
      A4(I) = .5*(T(I)+T(I4))
      A0(I) = A1(I)+A2(I)+A3(I)+A4(I)+BK(I)*D(I)**2
300  CONTINUE
      TIMES(1) = 0.

```

ภาพที่ 3.6 แสดงรายละเอียดโปรแกรมคอมพิวเตอร์

```

READ(1,*) (TIMES(K),K=2,NT)
DO 400 K=2,NT
DT = TIMES(K)-TIMES(K-1)
READ(1,*)(F(I),I=1,NB)
READ(1,*)(Q(I),I=1,N)
DO 10 I=1,N
R(I)=0.000
HP(I)=0.00
10 CONTINUE
DO 500 I=N1,N
B0(I)=-Q(I)+(BK(I)*HP(I)+R(I))*D(I)**2
500 CONTINUE
11 ITER =0
50 ERRC = 0.
ITER=ITER+1
DO 600 I = 1,NB
I1 = ID(I,1)
I2 = ID(I,2)
A = (F(I)+A1(I)*H(I1)+A2(I)*H(I2))/A0(I)
ERRC = ERRC+(A-H(I))**2
H(I) = A
600 CONTINUE
DO 700 I = N1,N
I1 = ID(I,1)
I2 = ID(I,2)
I3 = ID(I,3)
I4 = ID(I,4)
S1 = S(I)*D(I)**2 /DT
A = (A1(I)*H(I1)+A2(I)*H(I2)+A3(I)*H(I3)
* +A4(I)*H(I4)+B0(I)+S1*HLAST(I))/(A0(I)+S1)
ERRC = ERRC+(A-H(I))**2
H(I) = A
700 CONTINUE
WRITE(*,999) ITER
999 FORMAT(1X,'ITERATION = ',I4)
IF(ITER .GT.1000) GOTO 1500
ERRC = SQRT(ERRC)
IF(ERRC.GT.ERR) GOTO 50
1500 WRITE (2,1000) (ERRC,TIMES(K),I,H(I),I=1,N)
1000 FORMAT(2X,2F20.6,I6,F20.6)
DO 400 I = 1,N
HLAST(I)=H(I)
400 CONTINUE
222 STOP
END

```

ภาพที่ 3.6 แสดงรายละเอียดโปรแกรมคอมพิวเตอร์ (ต่อ)

3.3 วิธีการใช้งานโปรแกรมคอมพิวเตอร์

3.3.1 การเตรียมไฟล์ข้อมูล (Data File)

การเตรียมข้อมูลสำหรับใช้คำนวณในโปรแกรม GW.EXE มีรูปแบบของไฟล์เป็นตามภาพที่

3.7

NB	N	NT	ERR									
ID (1,1)	ID (1,2)	α	β	ZXN	ZYN	δ	Δ	T	H	F		
ID (2,1)	ID (2,2)	α	β	ZXN	ZYN	δ	Δ	T	H	F		
ID (3,1)	ID (3,2)	α	β	ZXN	ZYN	δ	Δ	T	H	F		
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
ID (NB,1)	ID (NB,2)	α	β	ZXN	ZYN	δ	Δ	T	H	F		
ID (NB+1,1)	ID (NB+1,2)	ID (NB+1,3)	ID (NB+1,4)	T	Δ	BK	S	H				
ID (NB+2,1)	ID (NB+2,2)	ID (NB+2,3)	ID (NB+2,4)	T	Δ	BK	S	H				
ID (NB+3,1)	ID (NB+3,2)	ID (NB+3,3)	ID (NB+3,4)	T	Δ	BK	S	H				
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
ID (N,1)	ID (N,2)	ID (N,3)	ID (N,4)	T	Δ	BK	S	H				
TIMES(1)	TIMES(2)	TIMES(3)		TIMES(NT)								
F(1)	F(2)	F(3)		F (NB)								
Q(1)	Q(2)	Q(3)		Q(N)								
F(1)	F(2)	F(3)		F(NB)								
Q(1)	Q(2)	Q(3)		Q(N)								
.	.	.										
.	.	.										
.	.	.										
(NT-1) ชุด												

ภาพที่ 3.7 แสดงตัวอย่างรูปแบบของไฟล์ข้อมูลที่ใช้ในโปรแกรมคำนวณ

3.3.2 วิธีการใช้โปรแกรม

โปรแกรมคำนวณ คือ GW.EXE ซึ่งได้จากการ Compile ของโปรแกรม GW.FOR ซึ่งแสดงในภาพที่ 3.6 สำหรับข้อมูลจริง (Data File) ที่ใช้ในการคำนวณ คือ "GW1.DAT" แสดงในภาคผนวก จ. เริ่มต้นการใช้งานโดยการพิมพ์ชื่อโปรแกรม คือ GW แล้วกด ENTER จากนั้นโปรแกรมจะถามชื่อ Data File โดยแสดงหน้าจอว่า UNIT 1? ให้ตอบโดยการพิมพ์ชื่อ Data File คือ GW1.DAT กด ENTER โปรแกรมจะถาม UNIT 2? หมายถึง Output File ในที่นี้ใช้ GW1.OUT จากนั้นกด ENTER โปรแกรมจะทำการคำนวณจนกระทั่งได้ผลลัพธ์ หน้าจอจะแสดงคำว่า "Stop-Program Terminated" จะได้ผลลัพธ์โดยใช้ชื่อว่า "GW1.OUT" ซึ่งแสดงในภาคผนวก จ.