CHAPTER FOUR

COMPUTATIONAL PROCEDURES FOR IMAGE RECONSTRUCTION

4.1 COMPUTER IMPLEMENTATION

The reconstruction procedure that is most commonly used on CT scanners is based on the filtered-backprojection algorithm. This algorithm has been shown to be extremely accurate and amenable to fast implementation. It is presented here for the case of parallel projection data

This algorithm is based on the relationships

$$g(x,y) = \int_{0}^{\pi} Q_{\theta}(x\cos\theta + y\sin\theta)\,d\theta \qquad (4.1)$$

where $Q_{\theta}(t)$, called filtered projections, are related to the projections $P_{\theta}(t)$ by

$$Q_{\theta}(t) = \int_{-\infty}^{\infty} P_{\theta}(z)\,h_{1}(t-z)\,dz, \qquad (4.2)$$

where $P_{\theta}(z)$ is the ray-sums of projection and the filter impulse response $h_{1}(t)$ is the inverse Fourier transform of the $|w|$ function in the frequency domain over the bandwidth of the system,

$$h_{1}(t) = \int_{-W}^{W} |w| \exp(j2\pi wt)\,dw \qquad (4.3)$$

Here W is the frequency beyond which the spectral energy in any projection can be assumed to be zero. These equations suggest the following steps for a digital implementation of the algorithm.

Step A Filtering

Each projection is sampled with a sampling interval of a cm. In order that the sampled projections do not suffer from aliasing distortion, this implies that W = 1/2a. Substituting this value of W in equation (4.3), the impulse response will be

$$h_1(t) = \frac{1}{2a^2}\left[\frac{(\sin(2\,t/2a))}{2\pi/2a}\right] - \frac{1}{4a^2}\left[\frac{\sin(\pi t/2a)}{\pi t/2a}\right]^2 \quad (4.4)$$

since the data are measured with a sampling interval of a and are presumably bandlimited, for digital processing the impulse response need only be known with the same bandwidth and,hence,the same sampling interval. We obtain from equation (4.4)[117,118]

$$h_1(na) = -\frac{4}{a^2\pi}\left[\frac{1}{(4n^2-1)}\right] \quad n=0,\pm1,\ldots \quad (4.5)$$

which is called Shepp-Logan filter,where n takes both negative and positive integer values. At the sampling points (na) the filtered projection values may be obtained from equation (4.2)

$$Q_\theta(na) = \sum_{m=-\infty}^{\infty} P_\theta(ma)h_1[(n-m)a] \quad (4.6)$$

Step B Backprojection

The second step deals with reconstruction the image from the filtered projections using a digital approximation to the integral in equation (4.2). When the number of projections M_{proj} is large and uniformly distributed over 180°, equation (4.2) may be approximated by

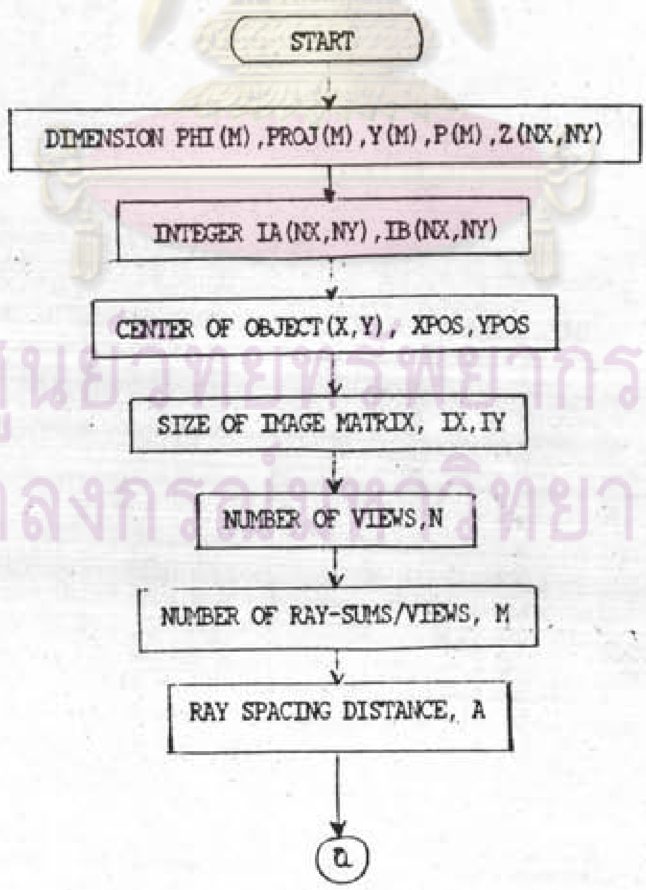
$$g'(x,y) = \frac{\pi}{M_{proj}} \sum_{i=1}^{M_{proj}} Q_{\theta i}(x\cos\theta_i + y\sin\theta_i) \qquad (4.7)$$

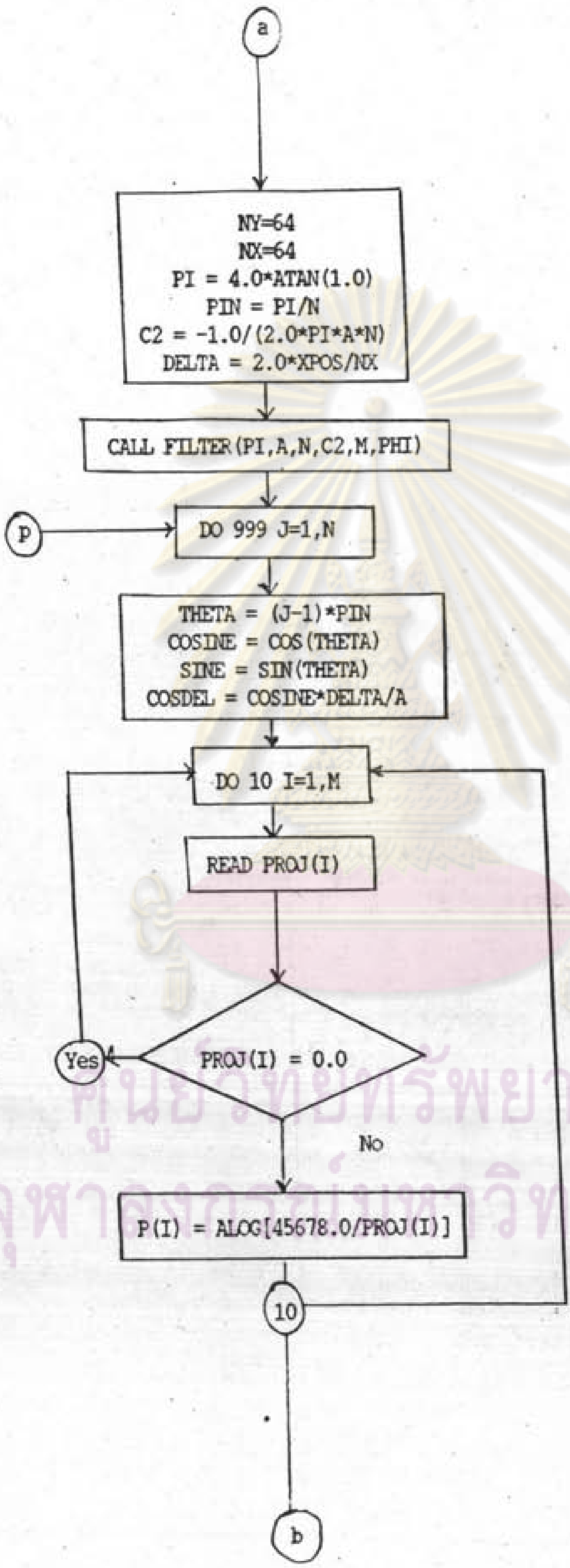
This equation calls for each filtered projection $Q_{\theta i}$ to be backprojected.

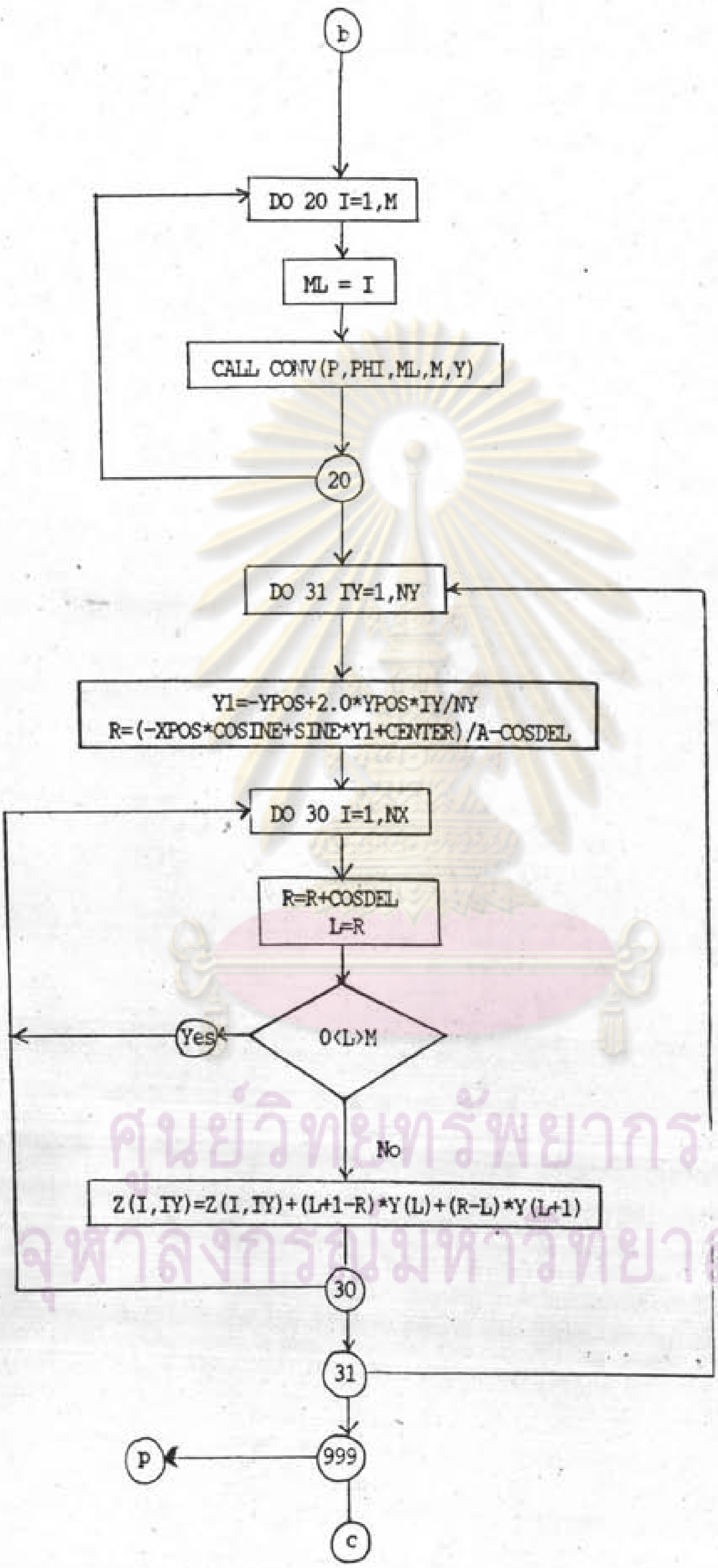
4.2 FLOW CHART

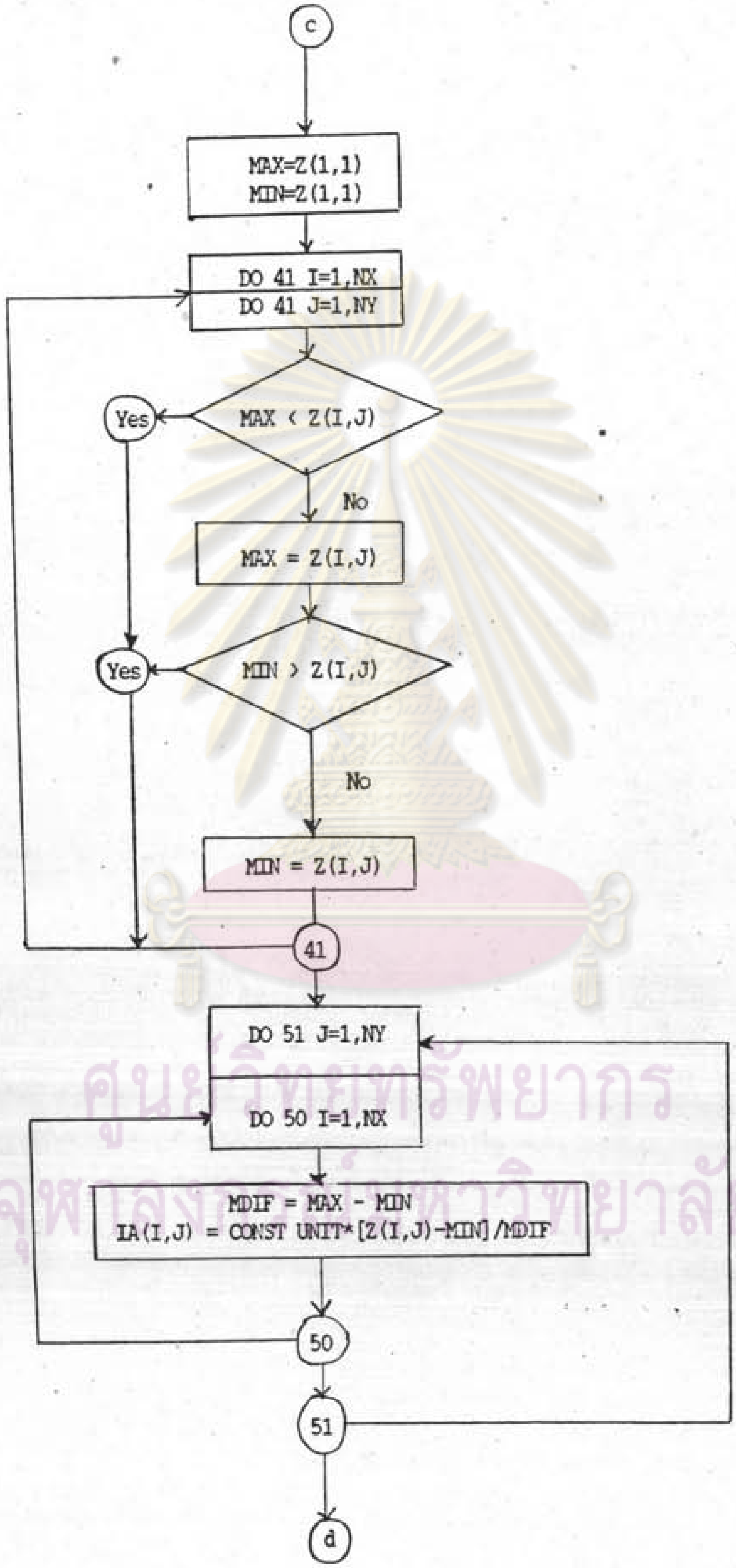
A computer program was developed in FORTRAN-IV using the algorithm illustrated in 4.1. The program consists of main program and 3 subprograms. Detials of the computer program are as follow.

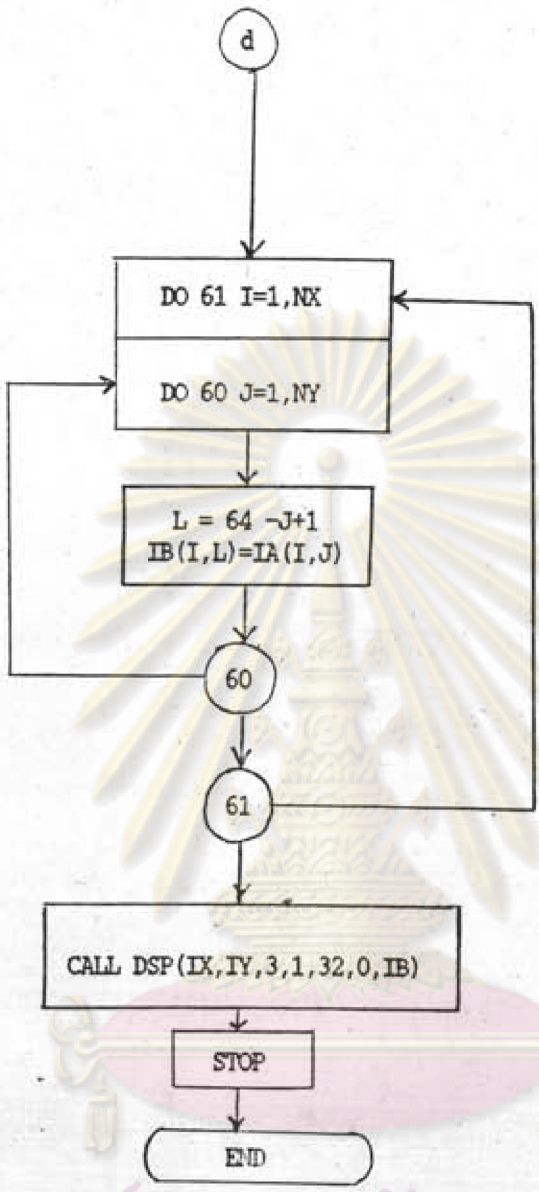
4.2.1 Flow Chart for Main Program




a
NY=64
NX=64
PI = 4.0*ATAN(1.0)
PIN = PI/N
C2 = -1.0/(2.0*PI*A*N)
DELTA = 2.0*XPOS/NX
CALL FILTER(PI,A,N,C2,M,PHI)
P
DO 999 J=1,N
THETA = (J-1)*PIN
COSINE = COS(THETA)
SINE = SIN(THETA)
COSDEL = COSINE*DELTA/A
DO 10 I=1,M
READ PROJ(I)
PROJ(I) = 0.0
Yes
No
P(I) = ALOG[45678.0/PROJ(I)]
10
b

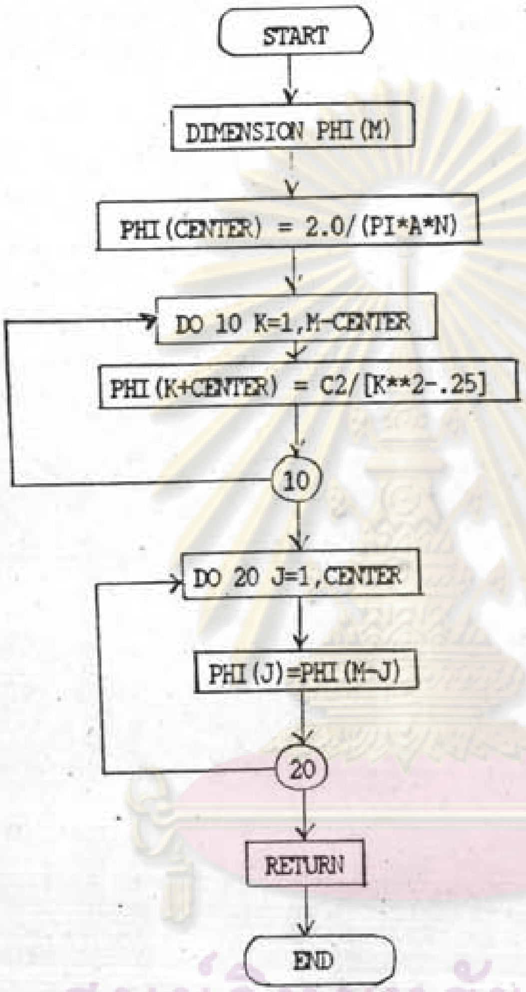

b
DO 20 I=1,M
ML = I
CALL CONV(P,PHI,ML,M,Y)
20
DO 31 IY=1,NY
Y1=-YPOS+2.0*YPOS*IY/NY
R=(-XPOS*COSINE+SINE*Y1+CENTER)/A-COSDEL
DO 30 I=1,NX
R=R+COSDEL
L=R
0<L>M
Yes
No
Z(I,IY)=Z(I,IY)+(L+1-R)*Y(L)+(R-L)*Y(L+1)
30
31
999
P
c


c
MAX=Z(1,1)
MIN=Z(1,1)
DO 41 I=1,NX
DO 41 J=1,NY
MAX < Z(I,J)
Yes
No
MAX = Z(I,J)
MIN > Z(I,J)
Yes
No
MIN = Z(I,J)
41
DO 51 J=1,NY
DO 50 I=1,NX
MDIF = MAX - MIN
IA(I,J) = CONST UNIT*[Z(I,J)-MIN]/MDIF
50
51
d


d
DO 61 I=1,NX
DO 60 J=1,NY
L = 64 -J+1
IB(I,L)=IA(I,J)
60
61
CALL DSP(IX,IY,3,1,32,0,IB)
STOP
END

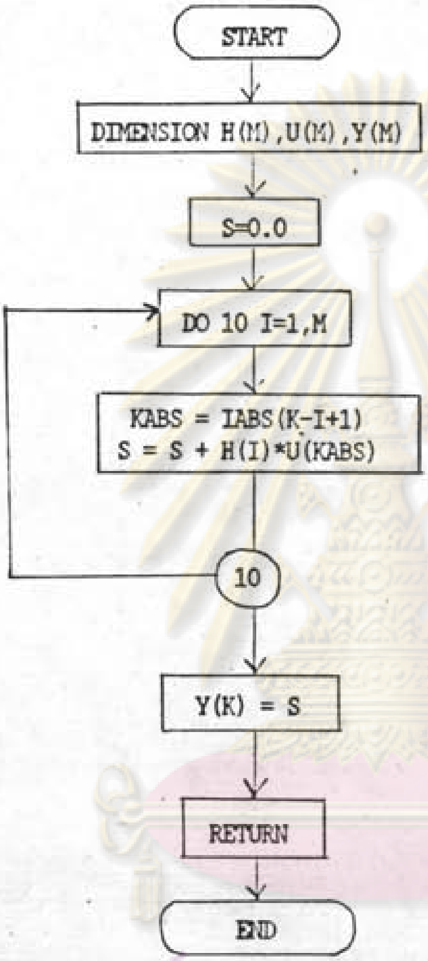
4.2.2 Flow Chart for subprogram FILTER

FILTER(PI,A,N,C2,M,PHI)



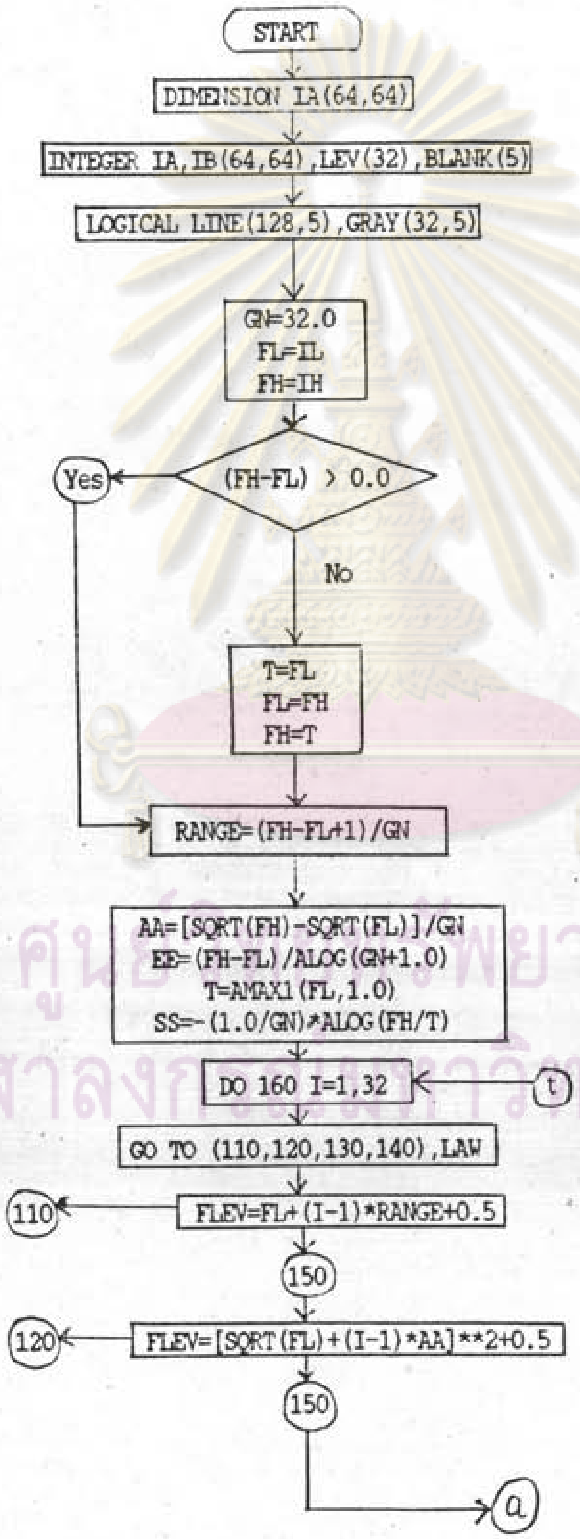
4.2.3 Flow Chart for subprogram CONVOLUTION

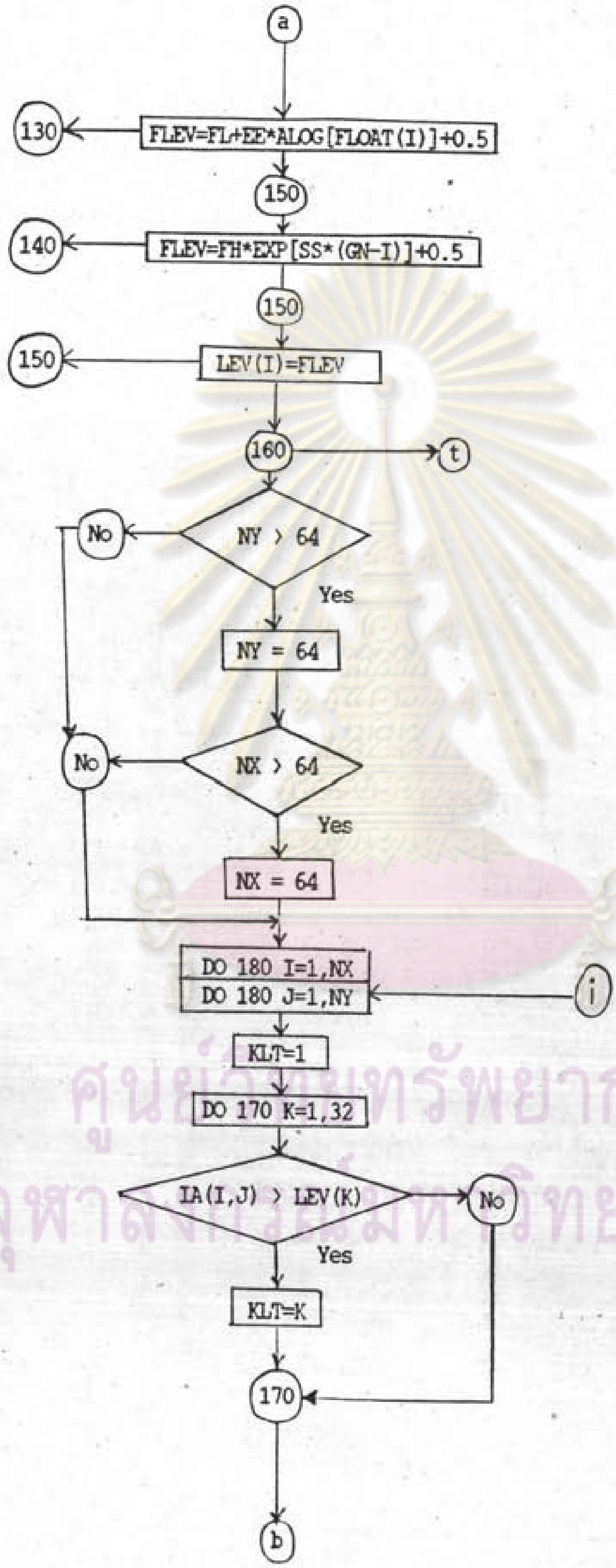
CONV(H,U,K,M,Y)

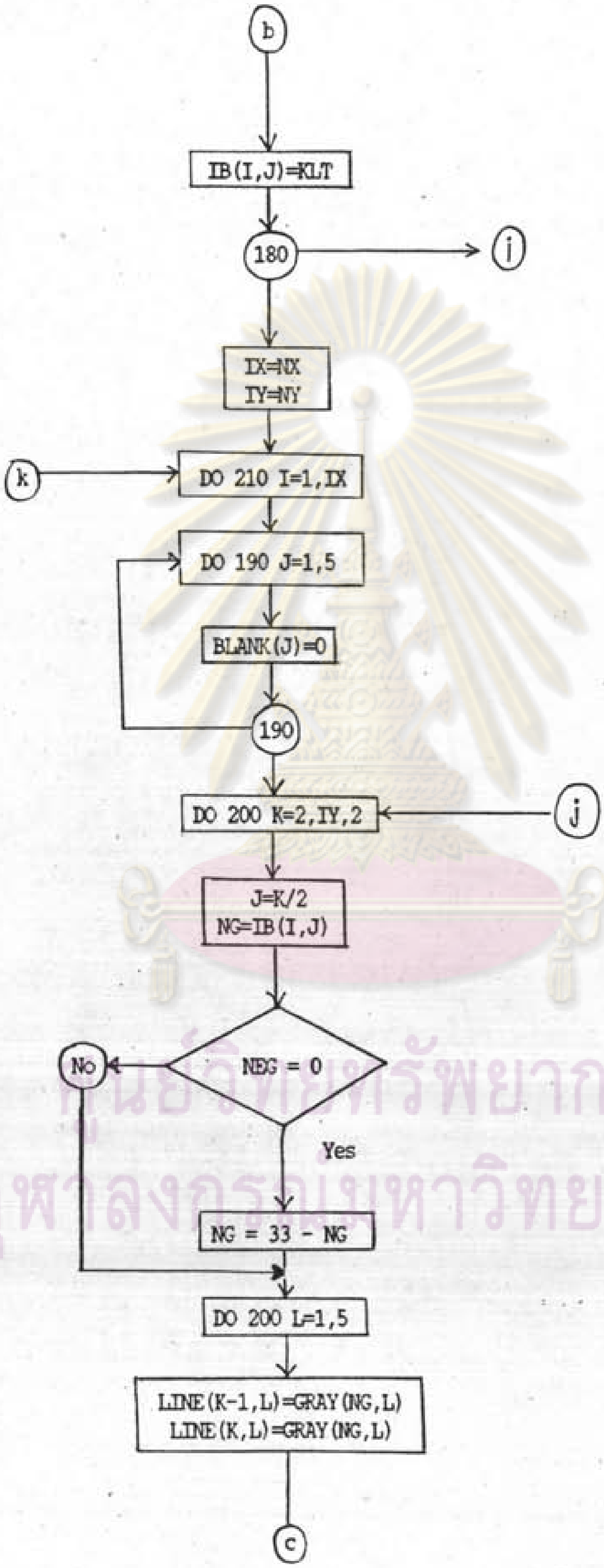


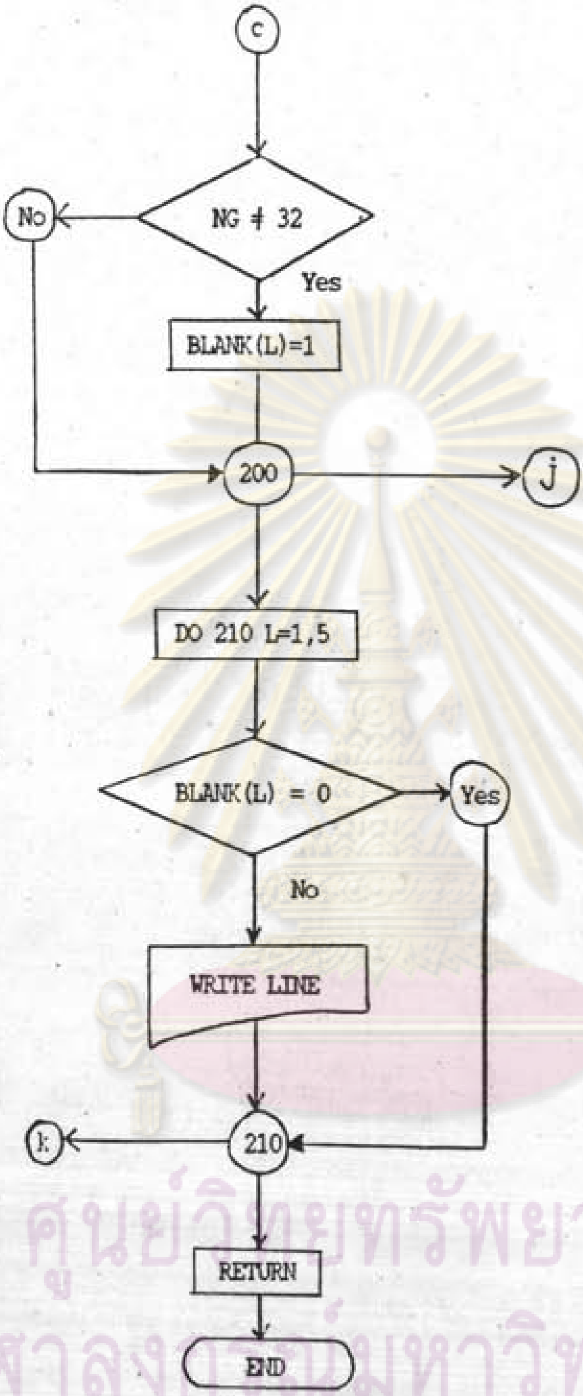
4.2.4 Flow Chart for subprogram DISPLAY IMAGE

DSP(NX,NY,LAW,IL,IH,NEG,IA)




a
130
FLEV=FL+EE*ALOG[FLOAT(I)]+0.5
150
140
FLEV=FH*EXP[SS*(GN-I)]+0.5
150
150
LEV(I)=FLEV
160
t
NY > 64
No
Yes
NY = 64
NX > 64
No
Yes
NX = 64
DO 180 I=1,NX
DO 180 J=1,NY
i
KLT=1
DO 170 K=1,32
IA(I,J) > LEV(K)
No
Yes
KLT=K
170
b


b
IB(I,J)=KLT
180
i
IX=NX
IY=NY
k
DO 210 I=1,IX
DO 190 J=1,5
BLANK(J)=0
190
DO 200 K=2,IY,2
j
J=K/2
NG=IB(I,J)
NEG = 0
No
Yes
NG = 33 - NG
DO 200 L=1,5
LINE(K-1,L)=GRAY(NG,L)
LINE(K,L)=GRAY(NG,L)
c



c
NG ≠ 32
No
Yes
BLANK(L)=1
200
j
DO 210 L=1,5
BLANK(L) = 0
Yes
No
WRITE LINE
210
k
RETURN
END

4.3 SUMMARY OF THE CALCULATION PROCEDURE

4.3.1 Input Quantities

The input data required for initializing the calculation procedure are as follow

a. XPOS,YPOS = center of object(x,y)

b. IX,IY = size of image matrix

c. N = number of views

d. M = number of ray-sums/view

e. A = ray spacing distance

The data to be processed are the ray-sums projection which have to be input to the program.

4.2.2 Order Of Calculations

a. Compute filter using equation (4.5) by subprogram FILTER

b. Read ray-sums from the file

c. Compute convolution between filter and ray-sums using equation (4.6) by subprogram CONV

d. Reconstruction of the image matrix using equation (4.7)

e. Check to see if all of the $O = 180^{\circ}$. If so, stop computation. If not, repeat step b. to step d.

4.2.3 Output Quantities

a. IB = image matrix

b. Display IB by subprogram DSP, to be printed out on paper.