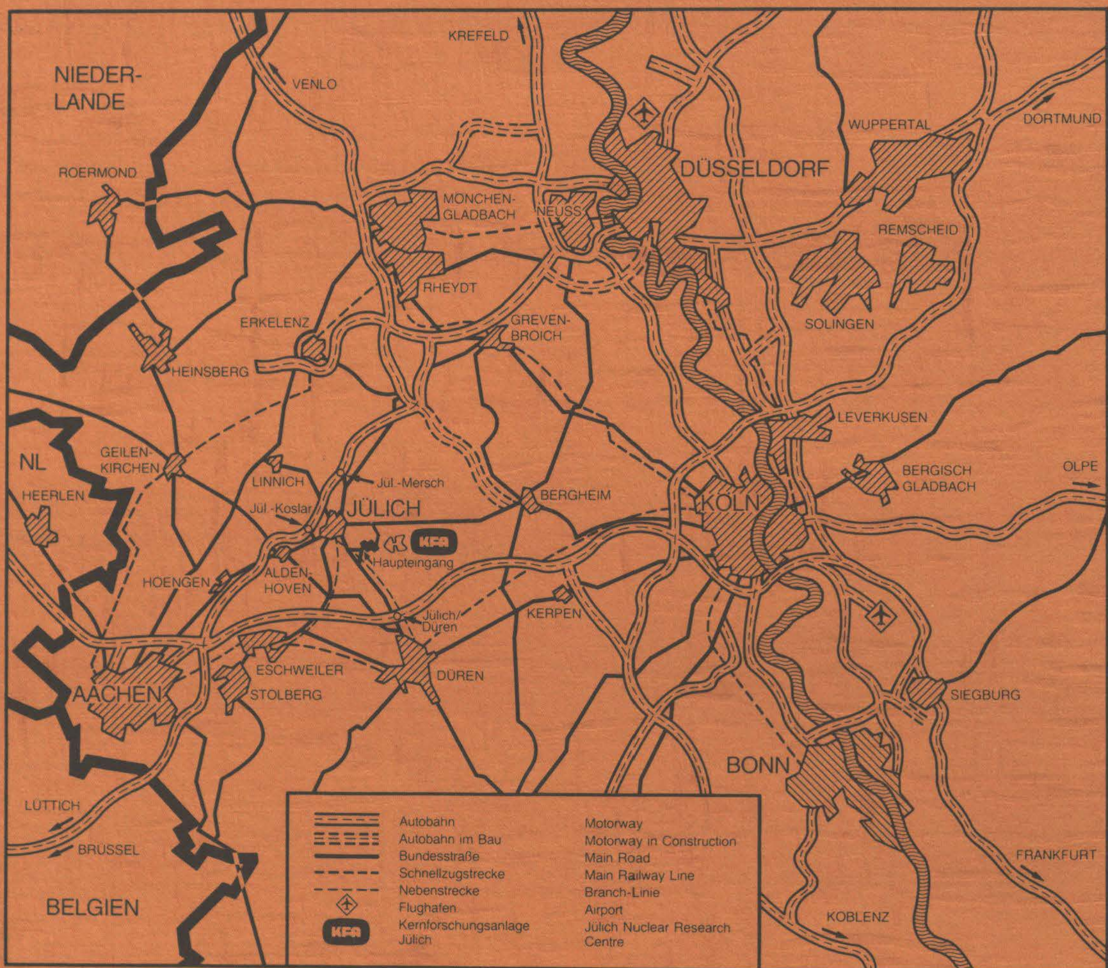
KERNFORSCHUNGSANLAGE JÜLICH GmbH
Institut für Medizin

Computerprogram for the Determination of Minimal Cardiac Transit Times

by
P. Bosiljanoff, H. Herzog, A. Schmid,
D. Sommer, K. Vyska, L.E. Feinendegen

Jül - Spez - 175
Oktober 1982
ISSN 0343-7639



Als Manuskript gedruckt

Spezielle Berichte der Kernforschungsanlage Jülich – Nr. 175

Institut für Medizin Jül - Spez - 175

Zu beziehen durch: ZENTRALBIBLIOTHEK der Kernforschungsanlage Jülich GmbH
Postfach 1913 · D-5170 Jülich (Bundesrepublik Deutschland)
Telefon: 02461/610 · Telex: 833556 kfa d

Computerprogram for the Determination of Minimal Cardiac Transit Times

by
P. Bosiljanoff, H. Herzog, A. Schmid,
D. Sommer, K. Vyska, L.E. Feinendegen

Institute of Medicine
Nuclear Research Center Jülich GmbH
and
Department of Nuclear Medicine
University Düsseldorf

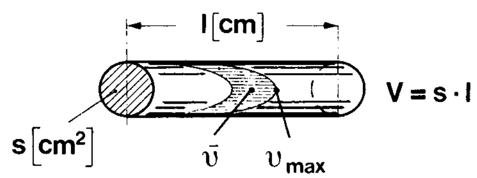
Table of Contents

1. Introduction

Blood flow is an important parameter of cardiac performance and can be determined by measuring circulation times (1). For that purpose, usually an Anger-Type gamma-camera is used to register the first pass of a radioactive tracer of blood flow through the heart. The acquired data are processed by a suitable computer program yielding time-activity curves for sequential heart segments, which are selected by the region of interest technique. Appropriate smoothing of the curves facilitates the evaluation of data and also the automatic detection of activity arrival (2,3,4). Sequential activity arrivals describe the fastest flow or minimal transit time.

2. Theory of Minimal Cardiac Transit Times

Minimal cardiac transit times (MTTs) are the differences of appearance times of a radioactive indicator in sequential segments of the heart. Figure 1 shows the relationship between the ratio of volume/flow and the flow velocity.



$$\frac{V}{F} = \frac{s \cdot l}{F} = \frac{cm^2 \cdot l}{\frac{cm^3}{sek}} = \frac{l}{\bar{\upsilon}}$$

$$\bar{\upsilon} \cdot K = \upsilon_{max}$$

$$\bar{\upsilon}_{max} = \frac{l}{t_{min}}$$

$$\bar{\upsilon} = \frac{l}{t_{min} \cdot K}$$

$$\boxed{\frac{V}{F} = t_{min} \cdot K}$$

(K for cardiac flow = 1.2)

Figure 1 Relationship between volume (V)/flow (F) and the flow velocity (υ)

The mean flow velocity is generally proportional to the maximal flow velocity. This proportionality is expressed by a coefficient K, which has been experimentally defined for the cardiopulmonary circulation and has the value of 1.2 (5). After rearrangement and substitution of the equations in figure 1, it can be clearly seen that the MTTs are directly proportional to the ratio of volume/flow. Theory and experiments show moreover that this relation is independent of the type of flow, be it laminar or turbulent (6). Since MTTs are measured by the arrival times of the tracer and do not directly relate to mean transit times, they are not influenced by factors that are external to the flow system, such as the duration of tracer injection or the regional field size of measurement (7).

The MTTs are inversely related to heart rate. The equation expressing the correlation is different for data obtained from patients in seated or supine position (2,8).
For the case of pulsating flow in the heart, theory and experiment indicate that, in the absence of aneurisms with flow vortices, the MTTs are inversely proportional to the ejection fraction (9), and this is a well-known cardiological parameter. The relationship expressed in figure 2 was found valid when compared to measurements of the left ventricular ejection fraction (LV-EF), by biplane angiography (10), or by the gated blood pool technique.

$$EF = \frac{t_s}{MTT - nt_d}$$

EF = ventricular ejection fraction
MTT = ventricular minimal transit time
t_s = time of systole
t_d = time of diastole
n = number of heart cycles during the ventricular transit time

Figure 2 Relationship between ejection fraction EF and minimal transit time

3. Performance of MTT-measurement

MTTs are measured by any fast gamma-camera in anterior projection, with the patient in either the supine or seated position. The pulse rate of the patient's heart has to be registered during the measurement, best by ECG, which also may serve as a physiological trigger, in case the gated list mode is chosen. It is up to the investigator to decide whether to use the frame mode or the gated list mode. Data analysis requires a dynamic study with 10 frames/sec over a period of 20 sec; a 32 x 32 word matrix is sufficient. The computer used for this study is a PDP 11/34 with a GAMMA-11-system. The time-activity curves are displayed on a VSVØ1 display unit including 4 bit maps. Depending on the type of camera and collimator Tc-99m-DTPA, Tc-99m-labelled red blood cells or In-113m-DTPA may be taken as the radioactive indicator in amounts of 3-5 mCi, in a volume of not more than 0.6 ml. The activity is injected into an antecubital vein by a saline flushing technique, as shown schematically in figure 3.

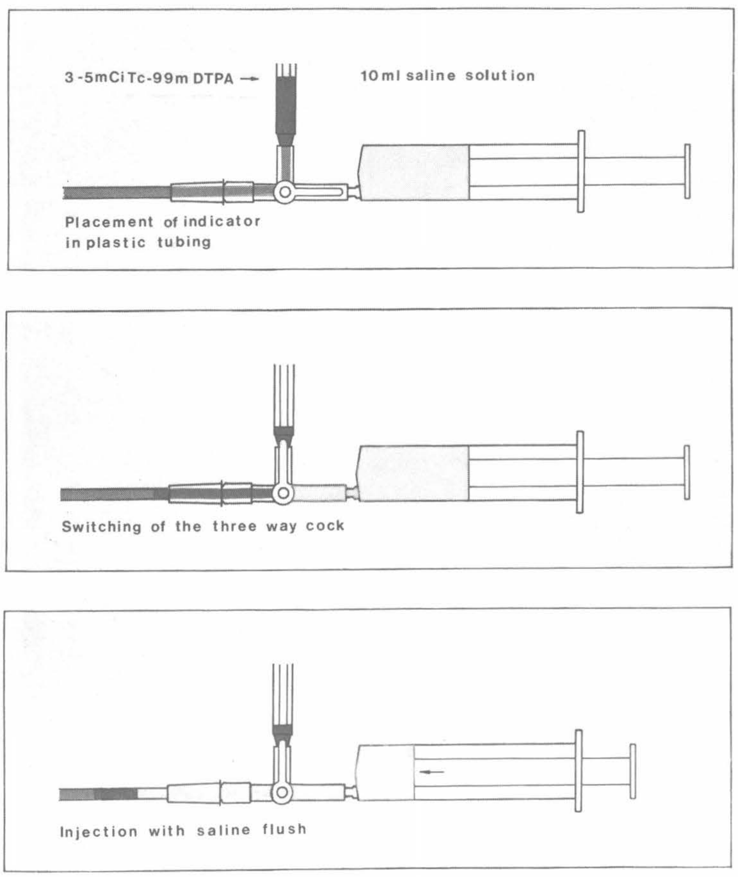


Figure 3 Technique of indicator injection

After insertion of the canule into an antecubital vein, the radioactive indicator is first placed into the plastic tubing. After changing the position of the three-way stop cock, 10-15ml of saline is injected rapidly at a rate of about 5ml/sec, in order to push the radioactive bolus into the venous system at a constant flow rate. Data registration begins with tracer injection. The measurement and data analysis may last less than 10 minutes per patient. The total body exposure is about 50-80mrem per investigation.

4. Evaluation

4.1 Selection of Regions of Interest

After the acquisition of data and initial display of the first pass (dynamic study, 10 frames/sec, total of 200 frames/study), quantitative data analysis starts by first adding 10 frames, which are then interpolated, and 5% of the background is sub-

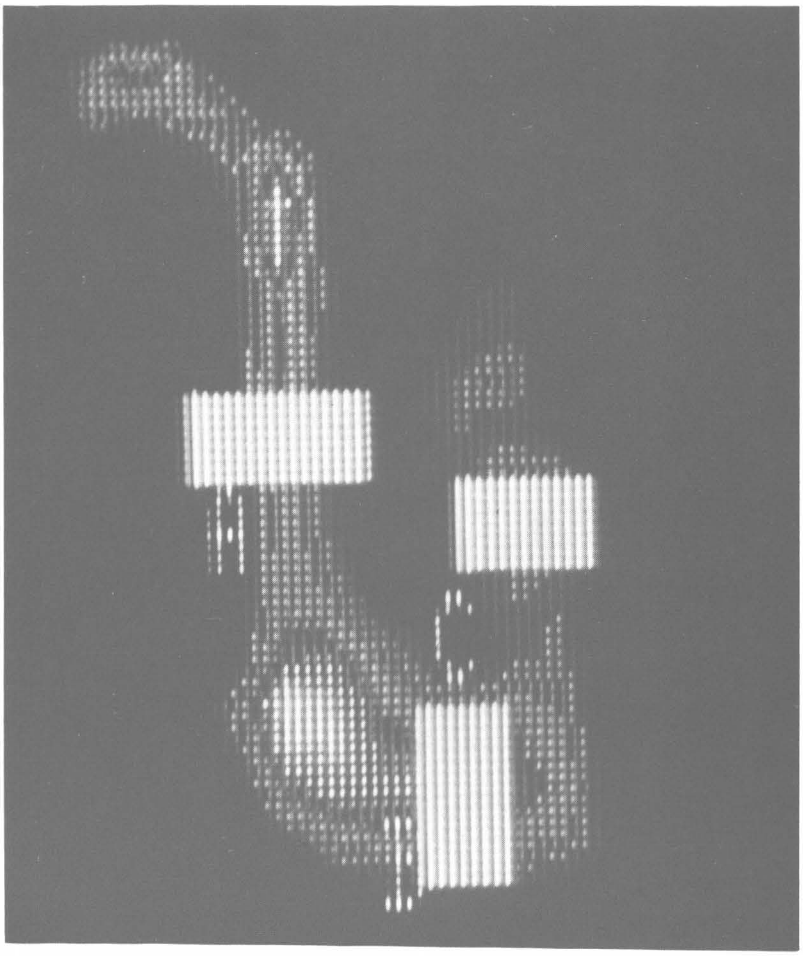


Figure 4a Definition of ROIs in the right heart region

tracted. Then, by means of the step by step addition of frames the right heart is first projected. For MTT measurement, regions of interest (ROIs) are defined for the superior vena cava, right atrium, and pulmonary artery, as shown in figure 4a. Continuation of the stepwise frame summation leads to the projection of the left heart, and ROIs are placed over the left atrium, left ventricle, and aortic root, as can be seen in figure 4b.

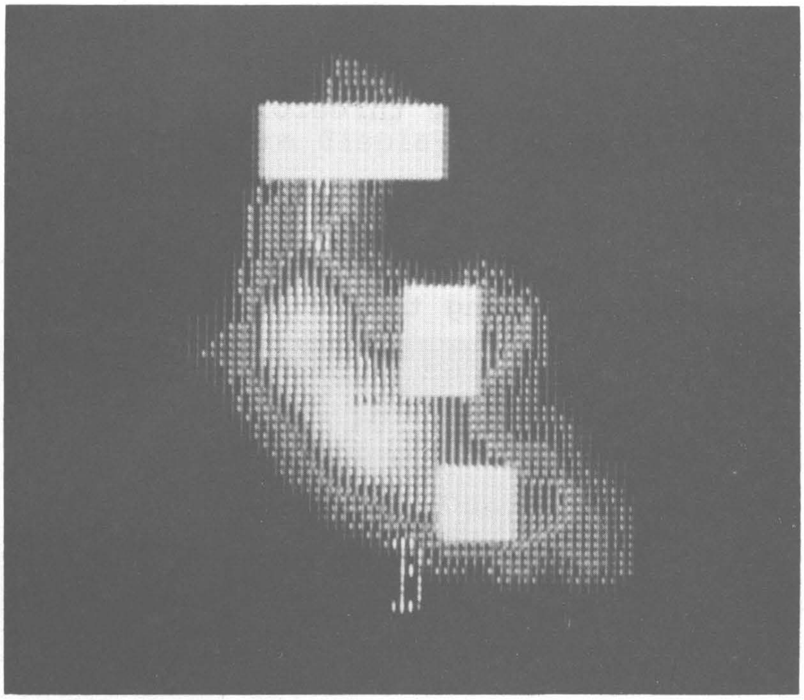


Figure 4b Definition of ROIs in the left heart region

It is important that the neighbouring regions are not situated too close to each other. Thus, the border between the atria and the ventricles and the efferent vessels should not be included in ROI, since there is a risk of ROI-overlapping by heart movement leading to wrong definitions of the heart segments. Therefore, it is useful to ensure that there is a distinct distance between the ROIs and the border line of any particular heart segment. The velocity of blood flow is high enough to make the appearance times of tracer in the various heart segments within the error of measurement independent of exact positions of the ROI within the segment. After delineation of the ROIs time-activity curves are generated and are finally stored into a save-area.

4.2 Determination of Transit Times

4.2.1 Automatic Finding of Indicator's Appearance Times

Data of each time-activity curve corresponding to the six chosen ROIs are read as Y(11)...Y(300) from the save-area. Data values Y(1)...Y(10) are zeroed to get a defined pre-start phase. First, the curves are smoothed three times, using the following equation:

$$\overline{Y}(I) = (Y(I-3)+3xY(I-2)+6xY(I-1)+7xY(I) +Y(I+3)+3xY(I+2)+6xY(I-1)) /27$$

This symmetrically weighted smoothing does not produce any bias. The coefficients result from the threefold application of a normal three-point smoothing

$$\overline{Y}(I) = (Y(I-1)+Y(I)+Y(I+1)) /3$$

For the curves 1 to 3, data of the beginning segment are zeroed up to a certain point belonging to the ascending slope of the curve maximum. The value of this point is slightly greater than 3% of the curve maximum.

The maximum slope of each curve, corresponding to the indicator's appearance time, is found, if the angle α(I) between two regression lines has a relative maximum for the first time. Regression line 1 is calculated from the curve points I, I+1, ..., I+4. Regression line 2 belongs to the curve points I+5, ..., I+9 for the curves 1 to 4 and to the curve points I+5,..., I+14 for the curve 5 and 6 (Figure 5 and Figure 6).

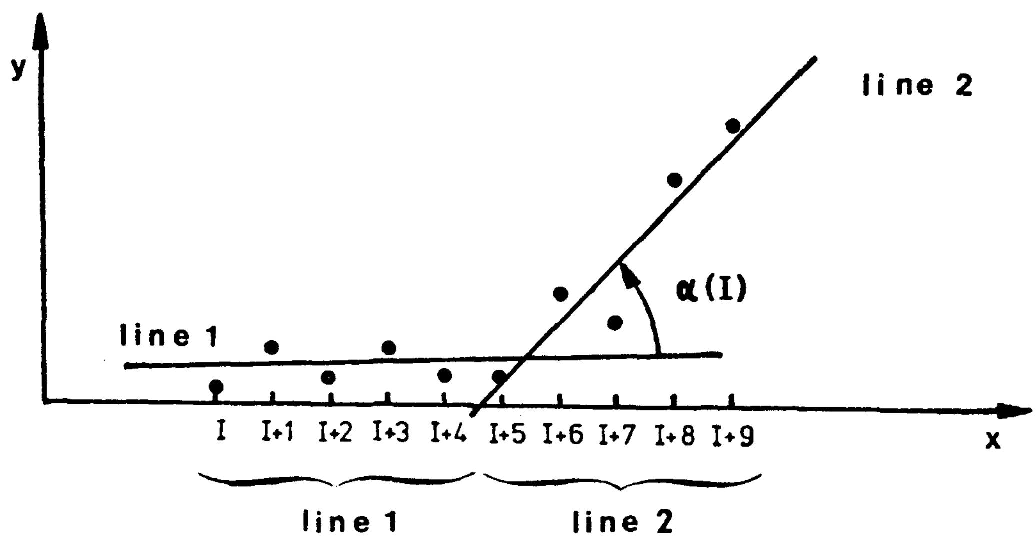


Figure 5 Application of two regression lines for search of indicator's appearance times

Beginning with curve 1 and I=1 the regression lines are shifted by steps on the curve. The maximum angle $\alpha(I_{max})$ is found, if values for n following angles are smaller. For the curves 1 to 3 n is 6 and for the curves 4 to 6 n is 10. The search for the angle $\alpha(I_{max})$ of curve M (M=2,3,5,6) starts at the found slope of curve M-1. In curve 4 the search begins after its first relative maximum. The individual handling of the single curves takes into account their typical courses.

4.2.2 Interaction with the Program

After the automatic determination of the transit times the six time-activity curves are displayed on the screen of the VSVØ1, the appearance times of the indicator being marked by bars (Figure 6).

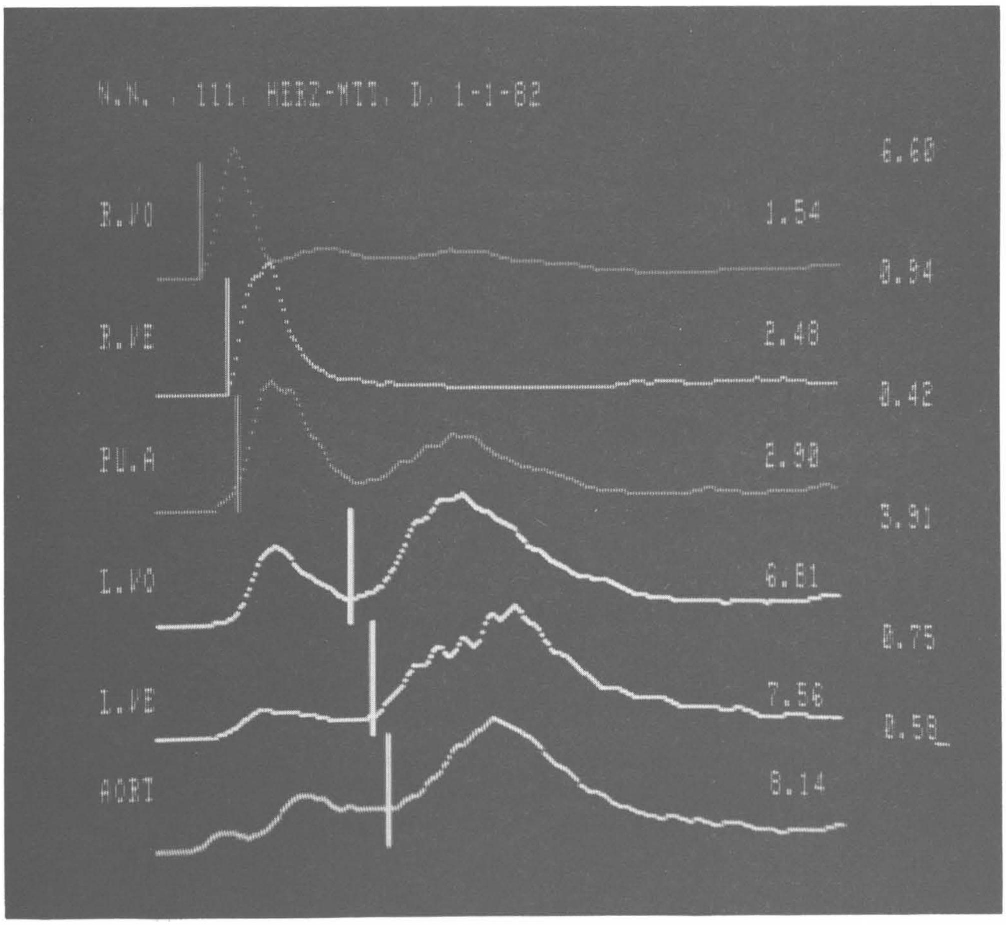


Figure 6 Output of the smoothed time-activity curves, the appearance times marked by bars (coloured display on VSVØ1)

The interaction between user and computer further offers the possibility for correcting the automatically determined appearance times. Experience has shown that especially the region of the left atrium is often not easy to detect.

If the user has chosen "Manual determination of Transit Times" at the start of the program, the automatic evaluation is skipped, and instantly the program enters the correction procedure for manual marking of the appearance times.
After the user agrees to the result of the automatic determination of transit times, the patient's heart rate as well as his position during the data acquisition, and the circumstances of investigation (rest or exercise) have to be defined. If a labelling method with red blood cells has been used, a correction factor needs to be applied.

4.3 Output of Results

Finally, the program prints the minimal cardiac transit times, in terms of total transit times, as well as segmental transit times for the right atrium, right ventricle, lung, left atrium and left ventricle. The measured values are normalized to a rate of 80/min and are compared to normal mean values(Figure 7).

Normal minimal transit times (in sec)

Mean values normalized to a heart rate of 80/min (± standard deviation in %)

	seated	supine
Total MTT	6.6 ± 4	6.06 ± 5
R.A.	0.82 ± 18	0.45 ± 20
R.V.	0.77 ± 18	0.77 ± 19
Lung	3.2 ± 9.7	3.15 ± 8
L.A.	0.92 ± 12	0.68 ± 16
L.V.	0.86 ± 15	1.0 ± 18
Total heart	3.4 ± 8	2.9 ± 10

Figure 7 Normal MTT values of a healthy population

The deviation from the normal mean values is characterized by a coefficient F. Moreover, these findings are qualitatively rated. Figure 8 shows an example printout of results.

```
N.N. , 111, HEART-MTT, D, 1-1-82


MEASUREMENT OF MINIMAL TRANSIT TIMES (MTT)

HEARTRATE: 86

TYPE OF STUDY:   REST                    SUPINE

TRANSIT TIMES (SEC) - G=MEAS.VAL., N=FREQU.NOR., F=FACTOR NORM
-------------
                 G     N       F   NORM.REGION   FINDING

TOTAL          6.60  7.09  1.17  5.05 - 7.05  SLIGHT PROLONG.
R.ATRIUM       0.94  1.01  2.27  0.13 - 0.76  EXTENSIVE PROLONG.
R.VENTR.       0.42  0.45  0.61  0.37 - 1.11  NORMAL
LUNG           3.91  4.20  1.30  2.38 - 4.07  MEDIUM PROLONG.
L.ATRIUM       0.75  0.81  1.21  0.15 - 1.19  NORMAL
L.VENTR.       0.58  0.62  0.65  0.44 - 1.49  NORMAL
```

Figure 8 Example printout of study result

5. Literature

1. Hegglin, R. und W. Rutishauser: In: Kreislaufdiagnostik mit der Farbstoffverdünnungsmethode, Kreislaufzeiten, Herzminutenvolumen, Blutvolumen, Klappeninsuffizienz, Shunt. Thieme, Stuttgart, 1962.

2. Feinendegen, L.E., V. Becker, K. Vyska, H. Schicha, and L. Seipel: Minimal cardiac transit times - Diagnostic radiocardiography in heart disease. J. Nucl. Biol. Med. 16: 211 - 223 (1972).

3. Feinendegen, L.E., V. Becker, K. Vyska, H. Schicha, Chr. Freundlieb, P. Bosiljanoff, B. Lösse, und U. Gleichmann: Bestimmung globaler und regionaler Herzfunktion mit den minimalen Transitzeiten. Verh. Dtsch. Ges. Herz- und Kreislaufforschg. 46, 118-129 (1980).

4. Bosiljanoff, P., A. Schmid, D. Sommer, K. Vyska, L.E. Feinendegen: Klinische Wertigkeit der Messung der minimalen cardialen Transitzeit mittels eines automatischen Rechenprogramms. 60. Dtsch. Röntgenkongr., Köln, Mai 1980 (Abstr.)

5. Rossi, H.H., S.H. Powers and B. Dwork: Measurements of flow in straight tubes by means of dilution technique. Amer. J. Physiol. 173: 103 (1953).

6. Vyska, K.: Theoretische Grundlagen zur Interpretation von Indikator-Dilutions-Kurven. In: Intensiv-Notfallmedizin-Anästhesiologie. Hrsg.: Zindler, Purschke, S. 134-146. Thieme, Stuttgart 1976.

7. Becker, V., H. Breuer, K. Vyska, H. Schicha and L.E. Feinendegen: Experimental measurements of the dependence of indicator dilution curves on extrasystemic parameters. In: Nuklearmedizin - Stand und Zukunft. Hrsg.: H.A.E. Schmidt, M. Woldring, F.K. Schattauer-Verlag Stuttgart-New York, 93-96 (1978).

8. Schicha, H., K. Vyska, W. Knapp, V. Becker, H. Heck, W. Hollmann, L.E. Feinendegen: Minimale kardiale Transitzeiten (MTTs) in der Herzdiagnostik - Messungen mit der Gamma-Retina und Indium-113m. IV. MTTs bei Untrainierten und Leistungssportlern in Ruhe und nach Belastung. Zeitschrift

für Kardiologie, Bd. 63, 733-744, 1974.

9. Vyska, K., M. Profant, H. Schicha, V. Becker, Chr. Freundlieb und L.E. Feinendegen: Theoretische Grundlagen der Anwendung der minimalen kardialen Transitzeiten für die Bestimmung der Ejektionsfraktion in der Herzkammer. In: Nuklearmedizin in klinischer und technologischer Sicht, F.K. Schattauer-Verlag Stuttgart, 68-72 (1975).

10. Sigwart, U., H. Schicha, V. Becker, K. Vyska, H. Schmidt, H.M. Mertens, U. Gleichmann und L.E. Feinendegen: Ventrikuläre Ejektionsfraktion - Isotopenangiographische Untersuchungen und Vergleichsmessungen mit der Cineangiographie. Verhdl. Dtsch. Ges. Inn. Med. 81: 227-230 (1975).

```
      PROGRAM MTTJUE
C
C     ***********************************************************************
C
C     AUTOMATIC DETERMINATION OF MINIMAL TRANSIT TIMES
C     FOR HEART STUDIES
C
C     PROGRAMMERS: A.SCHMID,D.SOMMER,H.HERZOG
C
C     NUCLEAR RESEARCH CENTER
C     INSTITUTE  OF  MEDICINE
C
C     RELEASION DATE 1-SEP-82
C
C     ***********************************************************************
C
      LOGICAL*1 LBLA(512)
      INTEGER*2 NROI,BLNR,AN,PZ,BLA(256),LMAX(6)
      INTEGER*2 IG,IK,IS,IABA(6),IABB(6),IABC(6),ISA,IGA
      INTEGER*2 I,J,K,L,M,N
      INTEGER*2 IXA,IYA,IXL,IYL,MAN
      REAL*4 F(300,6),X(300),Y(300),SCHW,FAK,MAX,MAXA,DET,A,B
      REAL*4 C,D,XMAX(6),XX,YY,CMIN(6),CMINA(6)
C
      COMMON F,X,Y,SCHW,FAK,MAX,MAXA,DET,A,B,C,D,XX,YY,
     1 CMIN,NROI,BLNR,AN,PZ,BLA,IG,IK,IS,IGA,ISA,
     2 I,J,K,L,M,N
      COMMON /MTTA/ XMAX
      COMMON /MTT/ IXA,IYA,IXL,IYL
      COMMON /DAT/ LMAX,CMINA,IABA,IABB,IABC
C
      EQUIVALENCE (BLA,LBLA)
C
```

```
      DATA LMAX / 6,6,6,10,10,10 /
      DATA CMINA /10.,10.,10.,4.,4.,4./
      DATA IABA /0,0,1,0,2,0/
      DATA IABB /0,0,0,1,0,1/
      DATA IABC /0,0,0,3,0,3/
      DATA F /1800*0./
C
      DO 1 I=1,300                    !X-VALUES FOR
1     X(I)=(I-1)*.1                   !10 FRAMES PER SECOND
C
C*****    DETERMINATION OF ARRIVAL TIMES
C
4     CALL MTTBES (MAN)
C
C*****    OUTPUT OF TRANSIT TIMES
C
      CALL MTTAUS (MAN,LBLA)
      CALL CLOSE(1)
      READ(5,500)IDUMMY               !FOR MANUAL FORMFEED
500   FORMAT(I5)
      WRITE(5,501)
501   FORMAT(////' NEXT STUDY ?',$)
      IF(IANT(I).EQ.1) GO TO 4
      END
```

```
C     *********************************************************
C
C     SUBROUTINE FOR DETERMINATION OF MINIMAL TRANSIT TIMES
C
C     *********************************************************
C
C     ***********************
      SUBROUTINE MTTBES (MAN)
C     ***********************
C
      LOGICAL*1 LBLA(512)
      INTEGER*2 NROI,BLNR,AN,PZ,BLA(256),LMAX(6)
      INTEGER*2 IG,IGX,IK,IS,IABA(6),IABB(6),IABC(6),ISA,IGA,IKA
      INTEGER*2 I,J,K,L,M,N,NA,ICH,KK
      INTEGER*2 IXA,IYA,IXL,IYL,MAN,MANWEI,SVNR
      REAL*4 F(300,6),X(300),Y(300),SCHW,FAK,MAX,MAXA,DET,A,B
      REAL*4 C,D,XMAX(6),XX,YY,CMIN(6),CMINA(6),TRAZEI(6),FAX(6)
C
      COMMON F,X,Y,SCHW,FAK,MAX,MAXA,DET,A,B,C,D,XX,YY,
     1 CMIN,NROI,BLNR,AN,PZ,BLA,IG,IK,IS,IGA,ISA,
     2 I,J,K,L,M,N
      COMMON /SVA/ NA,ICH,SVNR
      COMMON /MTTA/ XMAX,TRAZEI
      COMMON /MTT/ IXA,IYA,IXL,IYL
      COMMON /DAT/ LMAX,CMINA,IABA,IABB,IABC
C
      EQUIVALENCE (BLA,LBLA)
C
      DATA FAX / 3*0.03,3*0. /
C
      SVNR=0
      MANWEI=0
      WRITE(5,500)
500   FORMAT(' AUTOMATIC DETERMINATION OF TRANSIT TIMES ?',$)
      IF(IANT(I).EQ.1) GO TO 150
1     WRITE(5,502)
```

```
502   FORMAT(' ENTER'/
     1        ' 0 = MANUAL DETERMINATION OF TRANSIT TIMES'/
     2        ' 1 = DIRECT INPUT OF TRANSIT TIMES')
      READ(5,501) MAN
501   FORMAT(I5)
      IF(MAN.NE.0.AND.MAN.NE.1) GO TO 1
      IF(MAN.EQ.1) RETURN
      MANWEI=1
150   KK=0
      MAN=0
      ICH=1
      AN=3
      IXA=20
      IYA=0
      IXL=256
      IYL=256
C
C*****    READ SAVE-AREA-NR. AND OPEN IT
C
2     CALL SVAR
C
C*****    READ AND TEST TEXTBLOCK
C
      READ(1'1) BLA
      IF(BLA(184).EQ.6) GO TO 3
      WRITE (5,503)
503   FORMAT(' SAVEAREA DOES NOT CONTAIN 6 CURVES'/
     1 ' PLEASE ENTER AGAIN ')
      CALL CLOSE (1)
      GO TO 2
C
C*****    READ 300 VALUES FOR ROI 1-6
C*****    AND STORE THEM IN F(300,6)
C
3     ICHAN =ILUN(ICH)
18    BLNR=5
      DO 10 NROI=1,6
      ICODE=IREADW(290,F(11,NROI),BLNR,ICHAN)
```

```
      IF(ICODE.NE.290)TYPE 504,ICODE
504   FORMAT(' ERRORCODE IN IREADW:',I6)
      BLNR=BLNR+4
10    CONTINUE
C
C     SMOOTH CURVES AN-TIMES AND SET THRESHOLD
C
19    IF(AN .EQ. 0) GOTO 40
      DO 24 M=1,AN
      DO 24 I=1,6
      CALL MAXIM(F(1,I),200,MAX,L)
      CMIN(I)=CMINA(I)*MAX/100.
      SCHW=MAX*FAX(I)
      DO 30 K=1,2
      J=L
28    IF(F(J,I).GT.SCHW) GO TO 25
26    F(J,I)=0.
      J=J-1
      IF(J.NE.0) GO TO 26
      GO TO 27
25    J=J-1
      IF(J.NE.0) GO TO 28
27    IF(K.EQ.2) GO TO 30
      CALL GLAETT(F(1,I),Y,300,5,290)
      DO 29 J=1,300
29    F(J,I)=Y(J)
30    CONTINUE
24    CONTINUE
40    CONTINUE
C
C*****    START VALUE OF ARRIVAL TIMES FOR MANUAL DETERMINATION
C
      DO 153 I=1,6
153   XMAX(I)=1.0
```

```
C
C*****    DETERMINATION OF ARRIVAL TIMES
C
      IG=5
      ISA=0
      IKA=1
69    CONTINUE
C
C*****    INITIALIZATION OF VIRTUAL BIG BIT MAP
C
      CALL VSBBML
      ICLR=1
      CALL VSBBMI(ICLR)
      IF(MANWEI.EQ.1) GOTO 79
      IF(IKA .GT. 6)GOTO 79
      DO 70 IK=IKA,6
      IF(IK.NE.4) GO TO 73
C
C*****    SEARCH MAXIMUM IN CURVE 4
C
      CALL MAXIM(F(ISA,4),30,MAX,L)
      ISA=ISA+L
73    J=ISA
      MAX=0.
      I=0
      L=100
77    I=I+1
76    K=I+J
      CALL REGRES(X(K),F(K,IK),IG,A,B)
      IGX=5
      IF(IK  .GT. 4) IGX=10
      CALL REGRES(X(K+IG),F(K+IG,IK),IGX,C,D)
      XX=0.
      YY=0.
      IF(A.EQ.C) GO TO 71
      XX=-(B-D)/(A-C)
      YY=A*XX+B
71    MAXA=ATAN(C)-ATAN(A)
```

```
      IF(IK.GT.1.AND.XX.LT.XMAX(IK-1)) GOTO 77
      IF(C.LT.CMIN(IK) .AND. MAX .EQ. 0.)GO TO 77
      IF(MAXA.LT.MAX) GO TO 75
      MAX=MAXA
      IS=K
      XMAX(IK)=XX
      L=I+LMAX(IK)
75    I=I+1
      IF(I.GT.ISA+200) GO TO 82
      IF(I.LT.L) GO TO 76
      ISA=IS
      IF(IK.GE.4) ISA=ISA+2
70    CONTINUE
79    CONTINUE
C
C*****    CALCULATION OF TRANSIT TIMES
C
      DO 170 I=2,6
170   TRAZEI(I)=XMAX(I)-XMAX(I-1)
      TRAZEI(1)=XMAX(6)-XMAX(1)
C
C*****    OUTPUT OF PATIENT DATA
C
86    CALL MTBDHE(LBLA(17))
C
C*****    PLOT CURVES
C
      DO 80 I=1,6
      CALL MTPLHE(I,300,F(1,I))
80    CONTINUE
```

```
C
C*****    KK=SWITCH FOR SKIPPING STATEMENT 83
C
      IF(KK.EQ.1) GOTO 89
83    WRITE(5,505)
505   FORMAT(
     1  ' -2=WITHOUT SMOOTHING,'
     2  ,'-1=ONCE MORE SMOOTHING,0=CURVES OK,1=CORRECT CURVE')
      READ(5,501) IK
      AN=1
      IF(IK.LT.-2.OR.IK.GT.1) GO TO 83
      IF(IK .NE. -2) GOTO 85
      AN=0
      GOTO 18
85    IF(IK.EQ.-1) GOTO 19
      IF(IK.EQ.0) GO TO 82
      KK=0
89    WRITE(5,506)
506   FORMAT(' NUMBER OF CURVE? ',$)
      READ(5,501) IK
      IF(IK.LT.1.OR.IK.GT.6) GO TO 83
      KK=1
      CALL XCURHE(XMAX(IK))
      XMAX(IK)=XMAX(IK)+1.
      IKA=IK+1
      ISA=XMAX(IK)*10.+.5
      GO TO 69
82    CONTINUE
      RETURN
      END
```

```
C       SUBROUTINE FOR INPUT SAVEAREA NO. AND OPENING OF THE FILE
C       C
C       ***************
        SUBROUTINE SVAR
C       ***************
C
        LOGICAL*1 FILNAM(12),FIL(10)
        INTEGER*2 I,J,K,ICH,SVNR
C
        COMMON /SVA/ NA,ICH,SVNR
C
        DATA FILNAM / 'S','V','A','R','0','0','.','S','Y','S' ,' ',' '/
        DATA FIL / '0','1','2','3','4','5','6','7','8','9' /
C
        IF(SVNR.EQ.0) GO TO 1          !SVAR NO. IS READ
        J=SVNR                         !SVAR NO. IS TRANSFERRED
        GO TO 4
1       WRITE (5,500)
500     FORMAT (' SAVEAREA NO. ? ',$)
        READ (5,501) J
501     FORMAT(I2)
        IF (J.LT.0) J = 0
4       I = J/10
        K=33
        IF(J.GT.0.AND.J.LT.10) K = 53        !SVAR 1-9
        J = J-10*I
        FILNAM(5) = FIL(I+1)
        FILNAM(6) = FIL(J+1)
        CALL ASSIGN (ICH,FILNAM,10,'OLD','NC')
        DEFINE FILE ICH (K,256,U,NA)
        RETURN
        END
```

```
C       SUBROUTINE FOR SMOOTHING OF CURVE YE (WEIGHTS 1-3-6-7-6-3-1)
C
C       ********************************
        SUBROUTINE GLAETT(YE,YA,N,N1,N2)
C       ********************************
C
C       YE(N) INPUTVECTOR
C       YA(N) OUTPUTVECTOR
C       N1,N2 LIMITS FOR SMOOTHING
C
        INTEGER*2 N,N1,N2,I
        REAL*4 YE(1),YA(1)
C
        IF(N.LE.0)GOTO 10
        IF(N1.LE.0)GOTO10
        IF(N1.GT.N.OR.N1.GT.N2)GOTO 10
        IF(N2.GT.N)GOTO 10
        I=0
1       I=I+1
        IF(I.GT.N)RETURN
        IF(I.GE. N2) GOTO 7
        IF(I.GT.N1)GOTO 3
7       CONTINUE
        K=I
        YA(K)=YE(I)
        GOTO 1
3       IF(I.NE.N1+1.AND. I.NE.N2-1)GOTO 5
        YA(I)=(YE(I-1)+YE(I)+YE(I+1))/3.
        GOTO 1
5       IF(I.NE.N1+2.AND.I.NE.N2-2)GOTO 6
        YA(I)=(YE(I-2)+YE(I+2)+2.*(YE(I-1)+YE(I+1))
     1  +3.*YE(I))/9.
        GOTO 1
```

```
6     CONTINUE
      YA(I)=(YE(I-3)+YE(I+3)+3.*(YE(I-2)+YE(I+2))
     1  +6.*(YE(I-1)+YE(I+1))+7.*YE(I))/27.
      GOTO 1
10    WRITE(5,500) N,N1,N2
500   FORMAT(' WRONG VALUES N,N1,N2:',3I5,
     1   /,' IN SUBROUTINE GLAETT  ***STOP***'//)
      STOP
      END
```

```
C       SUBROUTINE FOR COMPUTATION OF A REGRESSION LINE
C
C       ****************************
        SUBROUTINE REGRES(X,Y,PZ,A,B)
C       ****************************
C
C       DETERMINATION OF THE COEFFICIENTS A,B OF A
C       REGRESSION LINE Y = A * X + B FOR PZ POINTS (X,Y)
C
        INTEGER*2 PZ,I,N
        REAL*4 X(1),Y(1),A,B,SXX,SX,SXY,SY,DET
C
        N=PZ
        SXX=0.
        SX=0.
        SY=0.
        SXY=0.
        IF(N.LT.2) N=2
        DO 1 I=1,N
        SXX=SXX+X(I)*X(I)
        SXY=SXY+X(I)*Y(I)
        SX=SX+X(I)
1       SY=SY+Y(I)
        DET=N*SXX-SX*SX
        IF(ABS(DET).LT.1.E-6) GO TO 2
        A=(SXY*N-SX*SY)/DET
        B=(SXX*SY-SX*SXY)/DET
        RETURN
2       WRITE(5,500) (I,X(I),Y(I),I=1,N)
500     FORMAT(' DETERMINANT 0 AT'/4X,'I',5X,'X',8X,'Y'/
     1  (I5,2F10.3))
        STOP
        END
```

```
C      SUBROUTINE FOR SEARCHING MAXIMUM OF VECTOR X
C
C      *******************************
       SUBROUTINE MAXIM(X,ANZ,MAX,ORT)
C      *******************************
C
C      X   = INPUTVECTOR
C      ANZ = END  OF RANGE TO BE SEARCHED IN
C      MAX = VALUE OF MAXIMUM
C      ORS = INDEX OF MAXIMUM
C
       INTEGER*2 ANZ,ORT,I
       REAL*4 X(1),MAX
C
       MAX=0.
       DO 1 I=1,ANZ
       IF(X(I) .LT. MAX)GOTO 1
       MAX=X(I)
       ORT=I
1      CONTINUE
       RETURN
       END
```

```
C       SUBROUTINE FOR OUTPUT OF RESULT
C
C       ***************************
        SUBROUTINE MTTAUS (MAN,LBLA)
C       ***************************
C
        LOGICAL*1 LBLA(512),S(850),LW
        INTEGER*2 BS,LPOS,DR,BEL,IHFQ,MAN,ERY
        REAL*4 M(20),AUF(2,2),D,E,B,A,F,LOKA(3,6),BEFA(6,8)
        REAL*4 TAB(60,2),RUBE(4,2)
C
        COMMON /MTTA/ XMAX(6)
C
        DATA TAB/
       1  5.771,6.076,7.133,7.926,9.078,10.238,13.210,1.E9,6.605,.034,
       2  -1.E9,.516,1.125,1.396,2*1.806,2*1.E9,.821,.0051,
       3  -1.E9,.492,1.053,1.314,2*1.701,2*1.E9,.773,.003,
       4  -1.E9,2.592,3.845,4.185,4.829,4.829,2*1.E9,3.219,.0127,
       5  -1.E9,.698,1.147,1.569,2*2.031,2*1.E9,.923,.0078,
       6  -1.E9,.598,1.131,1.471,2*1.903,2*1.E9,.865,.0053,
       1  4.751,5.052,7.045,7.259,8.314,9.376,12.098,1.E9,6.049,0.,
       2  -1.E9,.126,.763,.763,2*.979,2*1.E9,.445,0.,
       3  -1.E9,.368,1.113,1.26,2*1.631,2*1.E9,.741,0.,
       4  -1.E9,2.376,4.067,4.188,2*4.832,2*1.E9,3.221,0.,
       5  -1.E9,.146,1.187,1.187,1.468,1.468,2*1.E9,.667,0.,
       6  -1.E9,.437,1.486,1.636,2*2.116,2*1.E9,.962,0./
        DATA RUBE  / 'REST','    ','    ','    ','AFTE','R EX'
       1 ,'ERCI','SE  '/
        DATA AUF   / 'SEAT','ED  ','SUPI','NE  ' /
        DATA LOKA  / 'TOTA','L   ','    ','R.AT','RIUM','    ','R.VE',
       1             'NTR.','.   ','LUNG','    ','    ','L.AT','RIUM',
       2             '    ','L.VE','NTR.','.   ' /
```

```
      DATA  BEFA / 'EXTE','NSIV','E SH','ORT.','    ','    ',
     1             'SLIG','HT S','HORT','.   ','    ','    ',
     2             'NORM','AL  ','    ','    ','    ','    ',
     3             'SLIG','HT P','ROLO','NG. ','    ','    ',
     4             'MEDI','UM P','ROLO','NG. ','    ','    ',
     5             'MEDI','UM P','ROLO','NG. ','    ','    ',
     6             'EXTE','NSIV','E PR','OLON','G.  ','    ',
     7             'SHOR','T.  ','    ','    ','    ','    ' /
      DATA S / '\','#','-','$',846*'200 /
C
C*****    BS= LOG.NO OF DISPLAY UNIT, DR = LOG.NO OF PRINTER
C
      BS=5
      DR=5
      M(10)=XMAX(6)-XMAX(1)
      DO 10 I=11,15
      M(I)=XMAX(I-9)-XMAX(I-10)
10    CONTINUE
      DO 20 I=10,850
20    S(I) = S(5)
      IF(MAN.EQ.1) GO TO 40
      DO 30 I=1,66
30    S(9+I)=LBLA(16+I)
40    WRITE(BS,500)
500   FORMAT(' HEART RATE: ',$)
      READ(BS,501) IHFQ
501   FORMAT (I5)
      IF(IHFQ .LT. 30.OR. IHFQ .GT. 250)GOTO 40
      WRITE(BS,502)
502   FORMAT(' MEASUREMENT IN SUPINE POSITION ?',$)
      LPOS=IANT(I)
      WRITE(BS,503)
503   FORMAT(' EXERCISE STUDY ?',$)
      BEL=IANT(I)
      WRITE(BS,518)
518   FORMAT(' LABELING OF RED BLOOD CELLS ?',$)
```

```
      ERY=IANT(I)
      IF(MAN.EQ.0) GO TO 80
      WRITE(5,519)
519   FORMAT('PATIENT-NAME:',$)
      READ(BS,520) (LBLA(I),I=17,82)
520   FORMAT(66A1)
      DO 70 I=11,15
      IF(I .EQ. 11)WRITE(DR,504)
      IF(I .EQ. 12)WRITE(DR,505)
      IF(I .EQ. 13)WRITE(DR,506)
      IF(I .EQ. 14)WRITE(DR,507)
      IF(I .EQ. 15)WRITE(DR,508)
504   FORMAT(' RIGHT ATRIUM = ',$)
505   FORMAT(' RIGHT VENTR. = ',$)
506   FORMAT(' LUNG         = ',$)
507   FORMAT(' LEFT ATRIUM  = ',$)
508   FORMAT(' LEFT VENTR.  = ',$)
      READ(BS,509) M(I)
509   FORMAT(F6.3)
70    CONTINUE
      M(10)=M(11)+M(12)+M(13)+M(14)+M(15)
80    DO 90,I=1,6
90    M(I)=M(9+I)
      WRITE(DR,510)
510   FORMAT(' PAPER ADJUSTED FOR OUTPUT ?')
      READ(DR,511) LW
511   FORMAT (A1)
      WRITE(DR,512)(LBLA(I),I=17,82)
512   FORMAT(//' ',66A1,/1H )
      WRITE(DR,513)
513   FORMAT(//,' MEASUREMENT OF MINIMAL TRANSIT TIMES (MTT)'//)
      WRITE(5,514) IHFQ
514   FORMAT(' HEARTRATE: ',I3,//)
      WRITE(DR,515) (RUBE(J,BEL+1),J=1,4),(AUF(J,LPOS+1),J=1,2)
515   FORMAT(' TYPE OF STUDY:    ',4A4,5X,2A4/)
```

```
      WRITE(DR,516)
516   FORMAT(// TRANSIT TIMES (SEC) - G=MEAS.VAL., N=FREQU.NOR.,
     1 , 'F=FACTOR NORM',//' -------------',/,15X,'G',4X,'N',6X,
     2  'F',2X,'NORM.REGION',2X,'FINDING',/)
      DO 140 I=1,6
      IF(ERY.EQ.1) M(I)=M(I)*1.18
      B=M(I)
      IF(LPOS .EQ. 0) GOTO 100
      A=M(I)*IHFQ/80.
      GOTO 110
100   A=(IHFQ-80.)*TAB(10*I,LPOS+1)+M(I)
110   D=A/TAB(10*I-1,LPOS+1)
      E=TAB(10*I-8,LPOS+1)+.001
      F=TAB(10*I-7,LPOS+1)
      DO 120 J=1,8
      IF(A .LE. TAB(10*I+J-10,LPOS+1)) GOTO 130
120   CONTINUE
      J=8
130   M(I)=J
      IF(J .GT. 7) J=7
      IF(I .GT. 1 .AND. J .LT. 3) J=8
      WRITE(DR,517)(LOKA(K,I),K=1,3),B,A,D,E,F,(BEFA(K,J),K=1,6)
517   FORMAT(1X,3A4,F5.2,2F6.2,F6.2,' -',F5.2,2X,6A4)
140   CONTINUE
      RETURN
      END
```

```
C      ************************************
C
C
C      SUBROUTINES FOR OUTPUT TO VSV01
C
C      ************************************
C
C      SUBROUTINE FOR OUTPUT OF PATIENT-DATA
C
C      **********************
       SUBROUTINE MTBDHE(TXT)
C      **********************
C
       LOGICAL*1 TXT(66),TXT64(64),BLANK(2)
C
       DATA BLANK /' ',0/
C
       DO 1 I=1,63
1      TXT64(I)=TXT(I)
C
C*****    MARK END OF STRING
C
       TXT64(64)=0
C
C*****    BLANK CHARACTER-MAP OF VSV01
C
       DO 2 K=1,25
       DO 2 L=1,64
       KK=K
       LL=L
2      CALL STROUT(BLANK,KK,LL)
C
C*****    OUTPUT OF PATIENT-DATA
       CALL STROUT (TXT64,1,1)
       RETURN
       END
```

```
C      SUBROUTINE FOR PLOTTING OF TIME-ACTIVITY-CURVES
C
C      ******************************
       SUBROUTINE MTPLHE (IKUR,NFR,Y)
C      ******************************
C
       INTEGER*2 IXA,IYA,IXL,IYL
       REAL*4 Y(1)
C
       COMMON /MTT/ IXA,IYA,IXL,IYL
C
C*****    ORDER OF CURVES FROM TOP TO BOTTOM
C
       YNORM=65.
       ISPACE=38
       INC=(6-IKUR)*ISPACE
C
C*****    OUTPUT OF LABELING AND MTT-MARKING LINES
C
       CALL MTTXHE (IKUR,INC,ISPACE)
C
C*****    OUTPUT OF TIME-ACTIVITY-CURVES
C
       YNORM = FLOAT (IYL)/6.
       YMAX = 0
       DO 1 I=1,NFR
1      YMAX = AMAX1(YMAX,Y(I))
       I1=11
       I2=I1+(IXL-1)-IXA
       DO 2 I=I1,I2
       YY = Y(I)*YNORM/YMAX
       IY = INC + IFIX(YY+0.5)
       IX = I-I1+IXA
       CALL VSBBMP (IKUR,IX,IY)
2      CONTINUE
       RETURN
       END
```

```
C      SUBROUTINE FOR LABELLING OF CURVES
C
C      **********************************
       SUBROUTINE MTTXHE (IKUR,INC,ISPACE)
C      **********************************
C
       LOGICAL*1 BYT(10),STR(5),LNAM(5),RNAM(24)
       INTEGER*2 ICX,IXA
       REAL*4 ZEIT(6),TRAZEI(6),YNORM
C
       COMMON /MTTA/ ZEIT,TRAZEI
       COMMON /MTT/ IXA
C
       DATA STR /'S','E','K','.',0 /
       DATA RNAM /'R','.','V','O','R','.','V','E','P','U','.','A',
      1           'L','.','V','O','L','.','V','E','A','O','R','T'/
C
       DO 1 I=1,4
1      LNAM(I)=RNAM(I+(IKUR-1)*4)
       LNAM(5)=0
C
C*****     CALCULATION OF THE APPROPIATE DISPLAY-ROW
C
       LINE=25-(INC/10+1)
C
C*****     OUTPUT OF LABELLING
C
       CALL STROUT (LNAM,LINE,1)
C
C*****     ABSCISSA OF SLOPE-MARKER
C
       ZEIT1=ZEIT(IKUR)-1.
       ICX=IFIX(ZEIT1*10.+.5)+IXA
```

```
C
C*****    OUTPUT OF SLOPE-MARKER
C
      I1=INC
      I2=INC+ISPACE
      DO 2 I=I1,I2
      II=I
      CALL VSBBMP (IKUR,ICX,II)
2     CONTINUE
C
C*****    CONVERSION OF TIME-VALUES TO ASCII
C
      ENCODE(9,3,BYT) ZEIT1
3     FORMAT(F9.2)
      BYT(10)=0
C
C*****    OUTPUT OF TRANSIT TIMES
C
      CALL STROUT (BYT,LINE,42)
      ENCODE(9,3,BYT) TRAZEI(IKUR)
      CALL STROUT (BYT,LINE-2,50)
      RETURN
      END
```

```
C      SUBROUTINE FOR CROSSHAIR-OUTPUT AND -SHIFTING
C
C      ************************
       SUBROUTINE XCURHE (ZEIT)
C      ************************
C
       LOGICAL*1 ZEICH(2), ZEI(4)
       INTEGER*2 IXEN,IYEN,ICX,IZEI,IXA,IR
       REAL*4 ZEIT
C
       COMMON /MTT/ IXA
C
       DATA ZEI /'R','L','J','M'/
C
       EQUIVALENCE (ZEICH(1),IZEI)
C
C*****    ENABLE X-CROSSHAIR
C
       IXEN=1
       IYEN=0
       IR=1
       CALL CROSEN(IXEN,IYEN)
C
C*****    SET BIT 12 OF JOB STATUS WORD
C
       CALL IPOKE("44,"10000.OR.IPEEK("44))
       ZEIT=ZEIT-1.
1      ICX=IFIX(ZEIT*10.+.5)+IXA
C
C*****    SET X-CROSSHAIR
C
       CALL CROSXY(ICX,1)
```

```
C
C*****    GET CONTROL-LETTER
C*****    READ FROM KEYBOARD IN SPECIAL MODE WITHOUT RETURN
C
30     IZEI=ITTINR()
       IF(IZEI.LT.0) GO TO 30
       IF(ZEICH(1).NE.ZEI(1)) GOTO 10
       ZEIT=ZEIT+0.1
       IR=1
       GOTO 1
10     IF(ZEICH(1).NE.ZEI(2)) GOTO 20
       ZEIT=ZEIT-0.1
       IR=-1
       GOTO 1
20     IF(ZEICH(1).NE.ZEI(3)) GOTO 40
       ZEIT=ZEIT+IR
       GOTO 1
40     IF(ZEICH(1).NE.ZEI(4)) GOTO 1
C
C*****    RESET BIT 12 OF JOB STATUS WORD
C
       CALL IPOKE("44,"167777.AND.IPEEK("44))
C
C*****    DISABLE X-CROSSHAIR
C
       IXEN=0
       CALL CROSEN (IXEN,IYEN)
       RETURN
       END
```

```
C      FUNCTION FOR HANDLING YES/NO ANSWERS
C
C      ****************
       FUNCTION IANT(I)
C      ****************
C
       LOGICAL*1 A,YES,JA,EINS,NO,NULL,BLANK
       DATA YES,JA,EINS,NO,NULL,BLANK/'Y','J','1','N','O',' '/
 1     ACCEPT 2,A
 2     FORMAT(A1)
       IANT=-1
       IF(A.EQ.YES.OR.A.EQ.JA.OR.A.EQ.EINS) IANT=1
       IF(A.EQ.NO.OR.A.EQ.NULL.OR.A.EQ.BLANK) IANT=0
       IF(IANT.GT.-1) RETURN
       TYPE 3
 3     FORMAT(' POSSIBLE ANSWERS: Y  J  1  OR N  O  <RETURN>'//
      1  ' REPLY ONCE MORE:',$)
       GO TO 1
       END
```

```
        .TITLE  VSBBM
;
;       SUBROUTINES FOR I/O-HANDLING WITH VSV01
;       THE FOUR BIT MAPS ARE CONSIDERED AS ONE VIRTUAL BIG BIT MAP
;       THE CALLING FORTRAN PROGRAM REFERS TO THIS BIG BIT MAP ONLY
;
;       AUTHOR: H. HERZOG, KFA JUELICH, INSTITUTE  OF  MEDICINE (9-SEP-81)


        .SBTTL  VSBBMI
;       CALL VSBBMI (ICLEAR)
;
;       VSBBMI:SUBROUTINE FOR POSITIONING OF 4 BIT MAPS
;              IN ORDER TO FORM A VIRTUAL BIG BIT MAP (BBM)
;              TO COVER THE WHOLE DISPLAY SCREEN
;
;              THE BIT MAPS ARE OPTIONALLY CLEARED
;
;       ICLEAR:0 = THE BIT MAPS ARE NOT CLEARED
;              1 = THE BIT MAPS ARE CLEARED
;
;ICLEAR =       R0                      ;CONTAINS CLEAR INFORMATION
;READY  =       R0                      ;CONTAINS THE READY FLAG INFORMATION
;       =       R1                      ;CONTAINS THE MODIFIED STATUS
;       =       R2                      ;CONTAINS THE SAVED MODIFIED STATUS
;       =       R3                      ;CONTAINS MODIFIED FUNCT.CODE F. STATUS REG.
;       =       R4                      ;CONTAINS CODE FOR BIT MAP POSITION
FCBMST  =       172600                  ;FUNCT.CODE OF BIT MAP STAT.REG.
STATUS  =       001400                  ;ORIGINAL STATUS
;
```

```
VSBBMI::
        CLR     R3
        MOV     #2,     R4              ;SET POSITION BITS FOR BIT MAP 2
        MOV     @2(R5), R0              ;GET ICLEAR
        SWAB    R0                      ;PUT ICLEAR INTO THE RIGHT POSITION
        MOV     #STATUS,R1              ;MODIFIED STATUS IN R1
        MOV     R1,     R2              ;SAVE R1 INTO R2
        MOV     #FCBMST,R3              ;MAKE FUNCT.CODE IDENTIFIING BIT MAP 0
SP1:    ADD     #20,    R3              ;MAKE FUNCT.CODE IDENTIFIING BIT MAPS
        MOV     R1,     (R3)            ;LOAD BIT MAP STAT.REG.
WAIT:   MOV     (R3),   R0              ;READ READY BIT FLAG
        BIT     #000200,R0              ;WAIT, UNTIL READY FLAG
        BEQ     WAIT                    ;IS SET AGAIN
        MOV     R2,     R1              ;GET OLD VALUE FROM R2 TO R1
        ADD     R4,     R1              ;PUT POS.CODE INTO STATUS WORD
        ASL     R4                      ;MODIFY
        ASL     R4                      ;POSITION CODE FOR BIT NEXT BIT MAP
        CMP     #128.,  R4              ;CHECK, IF BIT MAP 2 DONE
        BNE     SP1                     ;JUMP BACK, IF NOT YET DONE
        MOV     R2,     R1              ;GET OLD VALUE FROM R2 TO R1
        ADD     #12,    R1              ;PUT IN POS.CODE FOR BIT MAP 3
        MOV     R1,     20(R3)          ;LOAD STATUS BIT MAP 3
WAIT1:  MOV     20(R3), R0              ;READ READY BIT FLAG
        BIT     #000200,R0              ;WAIT, UNTIL READY FLAG
        BEQ     WAIT1                   ;IS SET AGAIN
        RTS     PC
;
;
;
```

```
        .SBTTL VSBBMP
;       CALL VSBBMP (IBIT4,IX,IYC)
;       VSSBMP:SUBROUTINE FOR RANDOM INPUT OF ONE PIXEL
;                IN A VIRTUAL BIG BIT MAP (BBM) CONSISTING OF 4 BIT MAPS
;       IBIT4: COLOUR ADDRESS OF ONE PIXEL
;       IX:    HORIZONTAL POSITION OF PIXEL 0(LEFT)-255.(RIGHT)
;       IYC:   VERTICAL POSITION OF PIXEL 0(BOTTOM)-255.(TOP)
;
;IX     =       R0                      ;HORIZONTAL POSITION
;IYC    =       R1                      ;VERTICAL POSITION
;IY     =       R1                      ;VERTICAL POSITION (INVERTED)
;       =       R2                      ;IDENTIFIER FOR BIT MAPS
FCIDPC  =       172602                  ;FUNCTION CODE OF BIT MAP
                                        ;PROGRAM COUNTER
VSBBMP::
        CLR     R2                      ;CLEAR IDENTIFIER FOR BIT MAP
        MOV     @6(R5), R1              ;GET VERT. POS. DOWN-TOP 0-255.
        COM     R1                      ;PUT VERT. POS. TOP-DOWN 0-255.
        BIC     #177400,R1              ;CLEAR UPPER BYTE
        MOV     @4(R5), R0              ;GET HORIZONTAL POSITION
        CMP     #128.,  R0              ;128-IX=?
        BLE     BITM13                  ;BRANCH, IF IX G.E. 128.
;
;BIT MAP 0 OR 2                         ;HORIZ. POS. UNMODIFIED
        CMP     #128.,  R1              ;128.-IY=?
        BLE     BITM2                   ;BRANCH, IF IY G.E. 128.
;
;BIT MAP 0                              ;VERT. POS. UNMODIFIED
        ADD     #20,    R2              ;BIT MAP 0 IDENTIFIER IN R2
        JMP     SPR
;
;BIT MAP 2
BITM2:  SUB     #128.,  R1              ;MODIFY VERT. POS.
        ADD     #60,    R2              ;BIT MAP 2 IDENTIFIER IN R2
        JMP     SPR
;
;BIT MAP 1 OR 3
BITM13: SUB     #128.,  R0              ;MODIFY HORIZ. POS.
```

```
        CMP     #128.,  R1              ;128.-IY=?
        BLE     BITM3                   ;BRANCH, IF IY G.E. 128.
;
;BIT MAP 1                              ;VERT. POS. UNMODIFIED
        ADD     #40,    R2              ;BIT MAP 1 IBENTIFIER IN R2
        JMP     SPR
;
;BIT MAP 3
BITM3:  SUB     #128.,  R1              ;MODIFY VERT. POS.
        ADD     #100,   R2              ;BIT MAP 3 IDENTIFIER IN R2
;
SPR:    BIS     #FCIDPC,R2              ;MAKE FUNCT.CODE FOR BIT MAP PC
        MOV     R0,     @#IX            ;SAVE R0 TO IX
        ASR     R0                      ;DIVIDE IX
        ASR     R0                      ;BY 4
        ASH     #5,     R1              ;HORIZ. POS. TIMES 32.
        ADD     R0,     R1              ;NUMBER OF BIT MAP PC IN R1,E.H.:
                                        ;R1=IX/4+IY*32.
        MOV     R1,     (R2)            ;OUTPUT OF BIT MAP PC-VALUE
        ADD     #2,     R2              ;MODIFY R2 FROM F.CODE PC INTO
                                        ;F.CODE OF BIT MAP BUFFER
        ASL     R0                      ;MULTIPLY R0
        ASL     R0                      ;TIMES 4
        MOV     @#IX,   R1              ;RETRIEVE HORIZ. POS.
        SUB     R0,     R1              ;COMPUTE POS. OF BYTE IN BIT MAP
                                        ;BUFFER, E.H. R1=IX-(IX/4)*4
        ADD     R1,     R2              ;MODIFY BIT MAP BUFFER F.CODE
                                        ;TO F. CODE OF BIT MAP BUFFER
        MOVB    @2(R5), (R2)            ;OUTPUT OF COLOUR ADDRESS TO
                                        ;BIT MAP BUFFER (BYTE)
        RTS     PC
;
;
IX:     .WORD   .-.                     ;HORIZONTAL POSITION
```

```
        .SBTTL  VSBBML
;       CALL    VSBBML
;
;
;       VSBBML:SUBROUTINE FOR DEFINITION OF A
;              LOOK-UP-TABLES OF 4 BIT MAPS
;
FCLUT   =       172610                  ;START FUNCT.CODE
FCLUT4  =       172710                  ;FUNCT.CODE OF LOOK UP TABLE 4
;
;       =       R0                      ;CONTAINS ACTUAL FUNCT. CODE
;       =       R1                      ;CONTAINS ADR. OF ACTUAL WORD OF
;                                       ;LOOK-UP-TABLE-BUFFER
;
VSBBML::
        MOV     #FCLUT, R0              ;START FUNCT.CODE TO R0
LOOP1:  ADD     #20,    R0              ;COMPUTE ACTUAL FUNCT.CODE
        MOV     #LUT1,  R1              ;GET 1. ADR. OF LOOK-UP-T. BUFFER
LOOP2:  MOV     (R1)+,  (R0)            ;BRING VALUE TO LOOK-UP-TABLE
        CMP     #LUT16, R1              ;CHECK, IF LAST BUFFER
        BNE     LOOP2                   ;VALUE TRANSFERRED
;
        CMP     #FCLUT4,R0              ;CHECK, IF 4.
        BNE     LOOP1                   ;LOOK-UP-TABLE WRITTEN
        RTS     PC
;
```

```
LUT1:     .WORD    000000                     ;BUFFER FOR LOOK-UP-TABLE-CONTENTS
LUT2:     .WORD    000403
LUT3:     .WORD    001014
LUT4:     .WORD    001460
LUT5:     .WORD    002017
LUT6:     .WORD    002474
LUT7:     .WORD    003063
LUT8:     .WORD    003402
LUT9:     .WORD    004010
LUT10:    .WORD    004440
LUT11:    .WORD    005012
LUT12:    .WORD    005450
LUT13:    .WORD    006042
LUT14:    .WORD    006462
LUT15:    .WORD    007070
LUT16:    .WORD    007477
;
;
;
;
```

```
        .SBTTL  CROSEN
;       CALL    CROSEN(IXEN,IYEN)
;
;       SUBROUTINE FOR ENABLING OF CROSSHAIRS
;       IXEN=1  ENABLE X-CROSSHAIR
;       IXEN=0 DISABLE X-CROSSHAIR
;       IYEN=1  ENABLE Y-CROSSHAIR
;       IYEN=0 DISABLE Y-CROSSHAIR
;
FCCRHS  =       172600                  ;FUNCTION-CODE STAT-REG. (CROSSHAIRS)
STATCR  =       0                       ;ORIGINAL STATUS
;
CROSEN::
        MOV     #STATCR,R0              ;GET CROSSHAIR-STATUS IN R0
        MOV     @2(R5), R1              ;GET IXEN IN R1
        BIT     #1,     R1              ;IF IXEN=0,
        BEQ     LIYEN                   ;GO TO LIYEN
        BIS     #010000,R0              ;SET BIT FOR X-CROSSH.ENABLING
LIYEN:  MOV     @4(R5), R1              ;GET IYEN IN R1
        BIT     #1,     R1              ;IF IYEN=0,
        BEQ     WEITER                  ;GO TO WEITER
        BIS     #004000,R0              ;SET BIT FOR Y-CROSSH.ENABLING
WEITER: MOV     R0,     @#FCCRHS                ;OUTPUT OF CROSSHAIR-STATUS
        RTS     PC
;
;
;
```

```
        .SBTTL  CROSXY
;       CALL    CROSXY(IX,IY)
;
;       SUBROUTINE FOR POSITIONING OF CROSSHAIRS
;       IX=POSITION OF X-CROSSHAIR (0-255)
;       IY=POSITION OF Y-CROSSHAIR (0-255)
;
FCHAIR  =       172602                  ;FUNCTION-CODE FOR CROSSHAIR REG.
;
CROSXY::
        MOVB    @4(R5), R1              ;GET Y-POSITION
        SWAB    R1                      ;Y-POSITION TO LEFT BYTE
        ADD     @2(R5), R1              ;GET X-POSITION AND ADD IT TO Y-POS.
        MOV     R1,     @#FCHAIR        ;OUTPUT OF X-CROSSHAIR-POSITION
        RTS     PC
;
;
;
;
;
```

```
        .SBTTL  STROUT
;       CALL STROUT(STR,ILIN,ICOL)
;
;       STROUT:SUBROUTINE FOR OUTPUT OF A ASCII CHAR.
;              STRING TO VSV01
;       STR:   CHARACTER-STRING (LOGICAL ARRAY)
;              LAST CHARACTER IS REPRESENTED BY A DECIMAL ZERO
;              OR IN COLUMN 64.
;       ILIN:  LINE-POSITION OF STRING
;       ICOL:  COLUMN-POSITION OF FIRST CHARACTER
;
FCCURS  =       172604                  ;FUNCTION CODE OF CURSOR POS. REGISTER
FCCHAR  =       172600                  ;FUNCTION CODE OF BUFFER/STAT REGISTER
;       =       R1                      ;CONTAINS ADDRESS OF CHAR. IN STRING ARRAY
;       =       R2                      ;CONTAINS CURSOR POSITION
;
STROUT::
        MOV     2(R5),  R1              ;TRANSFER ADDRESS OF FIRST CHARACTER TO R1
        MOVB    @4(R5), R2              ;TRANSFER LINE NUMBER
        DEC     R2                      ;CORRECT LINE NUMBER (VSV01 LINES 0-24!)
        SWAB    R2                      ;LINE NUMMBER TO LEFT BYTE OF R2
        ADD     @6(R5), R2              ;TRANSFER COLUMN NUMBER
        DEC     R2                      ;CORRECT COLUMN NUMBER (VSV01 COLUMNS 0-63)
                                        ;COMPLETE CURSOR POS. OF FIRST CHAR. NOW IN R2
        MOV     R2,     @#FCCURS        ;POSITIONING OF CURSOR
LAB1:   CMPB    (R1),   #0              ;CHECK FOR
        BEQ     LAB2                    ;TERMINATING ZERO
        CMPB    R2,     #64.            ;CHECK FOR
        BEQ     LAB2                    ;LAST+1 COLUMN
        MOVB    (R1),   @#FCCHAR        ;OUTPUT OF CHARACTER
WAIT2:  BIT     @#FCCHAR,#100000        ;WAIT, UNTIL READY FLAG
        BEQ     WAIT2                   ;IS SET AGAIN
        INC     R1                      ;INCREMENT R1 FOR ADRESS OF NEXT CHARACTER
        INC     R2                      ;INCREMENT CURS.POS. FOR LAST+1 COLUMN CHECK
        BR      LAB1                    ;SAME AGAIN
LAB2:   RTS     PC
;
;
        .END
```