U.S. DEPARTMENT OF COMMERCE
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
NATIONAL WEATHER SERVICE
NATIONAL METEOROLOGICAL CENTER

OFFICE NOTE 180

A Method for Solving the Helmholtz Equation in a Rectangular Region Using Homogeneous Neuman Boundary Conditions

Joseph P. Gerrity, Jr.
Development Division

JUNE 1978

1. Introduction

The Helmholtz equation is

$$\nabla^2\phi - \hat{\lambda}(x,y)\phi = \hat{G}(x,y) \tag{1}$$

where $\hat{\lambda}$ is a real, non-negative function, ∇^2 is the Laplacian, ϕ is the unknown, and $\hat{G}$ is a forcing function. The boundary conditions to be satisfied are $\frac{\partial\phi}{\partial n} = 0$, where $\frac{\partial}{\partial n}$ is the outward normal derivative.

The numerical algorithm presented here for the solution of this equation is a split-pseudo-direct method. We approximate the equation (1) using finite differences. A recurrence, or relaxation, equation is written to develop a sequence of solutions. Unlike the SOR method, we use implicit approximations in the recurrence equation in one coordinate direction. The implicit equation is inverted using Gaussian elimination. After completing the generation of the new approximate solution the process is repeated, but the other coordinate direction is now approximated implicitly in the recurrence equation.

Just two such passes have been found necessary for a test problem.

2. The finite difference equation

Assume that the region is rectangular of size $\big((P+1)\Delta x,(Q+1)\Delta x\big)$. Equation (1) is written,

$$\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} - (4+\lambda_{ij})\phi_{ij} = G_{ij} \tag{2}$$

where

$$\lambda_{ij} = (\Delta x)^2\ \hat{\lambda}\big((i-1)\Delta x,(j-1)\Delta x\big)$$

and

$$G_{ij} = (\Delta x)^2\ \hat{G}\big((i-1)\Delta x,(j-1)\Delta x\big)$$

The recurrence (or relaxation) equation is written in step 1.

$$\phi_{ij}^{n+1} = \phi_{ij}^{n} + \alpha\left(\phi_{i+1,j}^{n+1} + \phi_{i-1,j}^{n+1} + \phi_{ij+1}^{n} + \phi_{ij-1}^{n+1} - (4+\lambda_{ij})\phi_{ij}^{n+1} - G_{ij}\right) \quad (3)$$

In this step the i coordinate is done implicitly. The boundary conditions are evoked wherever they can be used.

The equation (3) is rewritten

$$-\alpha\ \phi_{i-1,j}^{n+1} + \left(1+\alpha(4+\lambda_{ij})\right)\phi_{ij}^{n+1} - \alpha\ \phi_{i+1,j}^{n+1} = R_{ij} \quad (4)$$

where

$$R_{ij} = \phi_{ij}^{n} + \alpha\left(\phi_{ij+1}^{n} + \phi_{ij-1}^{n+1} - G_{ij}\right) \quad (5)$$

The solution of (4) by Gaussian Elimination (cf. Richtmyer and Morton, p. 198) is carried out, augmenting j after each solution. On the row j=2, the variable ϕ_{ij-1}^{n+1} is equal to ϕ_{ij}^{n+1} because of the boundary condition, and this fact is used to modify (4). Similarly when j=Q, the parameter ϕ_{ij+1}^{n} is replaced by ϕ_{ij}^{n+1} because of the boundary condition.

In step 2, the recurrence equation is modified to

$$\phi_{ij}^{n+1} = \phi_{ij}^{n} + \alpha\left(\phi_{ij+1}^{n+1} + \phi_{ij-1}^{n+1} + \phi_{i+1,j}^{n} + \phi_{i-1,j}^{n+1} - (4+\lambda_{ij})\phi_{ij}^{n+1} - G_{ij}\right) \quad (6)$$

The j-coordinate direction is now done implicitly. The modifications of the system are similar to those outlined in step 1.

3. A test problem

In Appendix A, the Fortran code for a test problem is given. A 'true' solution was generated using an IBM random number generator. The Helmholtz term was similarly generated. Using these fields, the forcing function was

calculated for P=52, Q=46. Employing a first guess solution that was everywhere equal to the mean value of the 'true' solution, we ran the code with $\alpha=\frac{1}{4}$. At no point did the numerical solution differ from the 'true' solution by 1 part in 1000.

The average value of the 'true' solution is 5496.098, that of the numerical solution is 5496.094. The Bias was -.321E-2, and the Root-mean-square error of the numerical solution was 0.365E-2.

4. Further work

The method will be explored further using larger domains and different values of λ, including zero.

Reference

Richtmyer and Morton, Difference methods for initial value problems. Interscience, New York, N.Y., 1967.

```
LEVEL 2.2  (SEPT 76)                                  OS/360  FORTRAN H EXTENDED PLUS
REQUESTED OPTIONS. NODECK,NOLIST,OPTIMIZE(0),NOMAP,SIZE(MAX),NOIL,NOXREF,NOTERM,LC(
OPTIONS IN EFFECT. NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF
                        FUNCTIONS INLINE ARE:    NONE

                 C       MAIN PROGRAM FOR THE SOLUTION OF THE HELMHOLTZ EQUATION
                 C
   ISN 0002              DIMENSION Z(53,47),TSOL(53,47),R(53,47),HLM(53,47)
                 C
                 C       Z IS THE NUMERICAL SOLUTION OF THE DIFFERENCE EQUATION
                 C     TSOL IS THE TRUE SOLUTION OF THE DIFFERENCE EQUATION
                 C     R IS THE FORCING FUNCTION FOR TSOL.
                 C       HLM IS THE HELMHOLTZ COEFFICIENT.
                 C
                 C       ALPHA IS THE COEFFICIENT OF RELAXATION.
   ISN 0003              DATA ALPHA/.25/
                 C
   ISN 0004              CALL INITL(R,TSOL,HLM)
                 C
                 C       SET GUESS TO MEAN VALUE OF TSOL.
                 C
   ISN 0005              FACT = 51*45
   ISN 0006              TMEAN = 0.
   ISN 0007              DO 1 I=2,52
   ISN 0008              DO 1 J=2,46
   ISN 0009              TMEAN = TMEAN + TSOL(I,J)/FACT
   ISN 0010            1 CONTINUE
   ISN 0011              DO 2 I=1,53
   ISN 0012              DO 2 J=1,47
   ISN 0013              Z(I,J) = TMEAN
   ISN 0014            2 CONTINUE
                 C
   ISN 0015              PRINT 3, TMEAN
   ISN 0016            3 FORMAT(1H0,'AVERAGE VALUE OF TSOL=',F10.3)
                 C
                 C       STEP1 SOLVES THE IMPLICIT ITERATIVE EQ. IN THE X DIRECTION.
                 C
   ISN 0017         4000 CONTINUE
   ISN 0018              CALL STEP1(Z,R,HLM,ALPHA,IER)
   ISN 0019              CALL DTFOUT(TSOL,Z,1)
   ISN 0020              CALL STEP2(Z,R,HLM,ALPHA,IER)
   ISN 0021              CALL DTFOUT(TSOL,Z,2)
   ISN 0022              IF(IER .EQ. 0) GO TO 4
   ISN 0024              PRINT 5
   ISN 0025            5 FORMAT(1H0,'HAVE TO STOP PROBLEM IN STEP1',)
   ISN 0026              STOP
   ISN 0027            4 CONTINUE
                 C
                 C     CALCULATE STATISTICS OF SOLUTION
                 C
   ISN 0028              CALL STAT(Z,TSOL)
   ISN 0029              STOP
   ISN 0030              END

*OPTIONS IN EFFEC,*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
```

```
 LEVEL 2.2  (SEPT 76)                          OS/360  FORTRAN H EXTENDED PLUS
REQUESTED OPTIONS. NODECK,NOLIST,OPTIMIZE(0),NOMAP,SIZE(MAX),NOIL,NOXREF,NOTRACE,LC(6
OPTIONS IN EFFECT. NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF A
                      FUNCTIONS INLINE ARE:   NONE

  ISN 0002            SUBROUTINE INITL(R,TSOL,XL)
  ISN 0003            DIMENSION R(53,47),TSOL(53,47),XL(53,47)
                C
                C      SET-UP TRUE SOLUTION USING RANDOM PERTURBATION
                C      Y IS A RANDOM NUMBER  0<Y<1
  ISN 0004            IX = 7131
  ISN 0005            DO 10 I=2,52
  ISN 0006            DO 10 J=2,46
  ISN 0007            CALL RDIGIT(IX,IY,Y)
  ISN 0008            TSOL(I,J) = 5000. + 1000. * Y
  ISN 0009            IX = IY
  ISN 0010         10 CONTINUE
                C
                C      ESTABLISH BOUNDARY CONDITIONS
  ISN 0011            DO 11 I=2,52
  ISN 0012            TSOL(I,1) = TSOL(I,2)
  ISN 0013            TSOL(I,47) = TSOL(I,46)
  ISN 0014         11 CONTINUE
  ISN 0015            DO 12 J=1,47
  ISN 0016            TSOL(1,J) = TSOL(2,J)
  ISN 0017            TSOL(53,J) = TSOL(52,J)
  ISN 0018         12 CONTINUE
                C
                C      SET - UP SCALED HELMHOLTZ PARAMETER
  ISN 0019            DX = 2.E5
  ISN 0020            IX = 837
  ISN 0021            DO 13 I=1,53
  ISN 0022            DO 13 J=1,47
  ISN 0023            CALL RDIGIT(IX,IY,Y)
  ISN 0024            XL(I,J) = Y * 10. * (DX**2)
  ISN 0025            IX = IY
  ISN 0026         13 CONTINUE
                C
                C      COMPUTE FORCING FUNCTION (SCALED BY DX**2)
  ISN 0027            DO 14 I=2,52
  ISN 0028            DO 14 J=2,46
  ISN 0029            R(I,J) = TSOL(I+1,J) + TSOL(I-1,J) + TSOL(I,J+1) + TSOL(I,J-1)
                     1  - 4. * TSOL(I,J)
  ISN 0030            R(I,J) = R(I,J) - XL(I,J) * TSOL(I,J)
  ISN 0031         14 CONTINUE
  ISN 0032            RETURN
  ISN 0033            END

*OPTIONS IN EFFEC.*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFEC.*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF A
*OPTIONS IN EFFEC.*   FUNCTIONS INLINE ARE:   NONE
*OPTIONS IN EFFEC.*
```

```
LEVEL 2.2  (SEPT 76)                          OS/360  FORTRAN H EXTENDED PLUS
REQUESTED OPTIONS. NODECK,NOLIST,OPTIMIZE(0),NOMAP,SIZE(MAX),NOIL,NOXREF,NOTERM,LC(6
OPTIONS IN EFFECT. NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF A

                        FUNCTIONS INLINE ARE:   NONE

   ISN 0002             SUBROUTINE RDIGIT(IX,IY,Y)
   ISN 0003             IY = IX * 65539
   ISN 0004             IF(IY) 5,6,6
   ISN 0005           5 IY = IY + 2147483647 + 1
   ISN 0006           6 Y = IY
   ISN 0007             Y = Y * 0.4656613E-9
   ISN 0008             RETURN
   ISN 0009             END
*OPTIONS IN EFFEC,*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFEC,*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF A
*OPTIONS IN EFFEC,*   FUNCTIONS INLINE ARE:   NONE
*OPTIONS IN EFFEC,*
*STATISTICS*      SOURCE STATEMENTS =        8, PROGRAM SIZE =       338, SUBPROGRAM NA
*STATISTICS*  NO  DIAGNOSTICS GENERATED
****** END OF COMPILATION ******                                          70K BYTES OF
```

```
REQUESTED OPTIONS. NODECK,NOLIST,OPTIMIZE(0),NOMAP,SIZE(MAX),NOIL,NOXREF,NOTERM,LC(
OPTIONS IN EFFECT. NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF
                   FUNCTIONS INLINE ARE:   NONE

ISN 0002            SUBROUTINE STEP1(Z,R,HLM,ALPHA,IER)
ISN 0003            DIMENSION Z(53,47),R(53,47),HLM(53,47)
ISN 0004            DOUBLE PRECISION C(51),D(51),E(51),ANS(51)
ISN 0005            DOUBLE PRECISION F(51)
              C      SET UP RELAXATION EQ.
              C      J = 2 ....   START
ISN 0006            DO 10 I=2,52
ISN 0007            F(I-1) = Z(I,2) + ALPHA *( Z(I,3) - R(I,2) )
ISN 0008         10 CONTINUE
ISN 0009            DO 11 I=1,51
ISN 0010            D(I) = 1. + ALPHA * ( 3. + HLM(I+1,2) )
ISN 0011            C(I) = -ALPHA
ISN 0012            E(I) = -ALPHA
ISN 0013            IF( I .EQ. 1  .OR. I .EQ. 51) D(I) = D(I) + C(I)
ISN 0015         11 CONTINUE
              C     SOLVE EQ.
ISN 0016            CALL GELIM(C,D,E,F,ANS,IER,51)
ISN 0017            IF(IER .NE. 0) GO TO 9000
ISN 0019            DO 12 I=2,52
ISN 0020            Z(I,2) = ANS(I-1)
ISN 0021         12 CONTINUE
ISN 0022            Z(1,2) = Z(2,2)
ISN 0023            Z(53,2) = Z(52,2)
ISN 0024            DO 13 I=1,53
ISN 0025            Z(I,1) = Z(I,2)
ISN 0026         13 CONTINUE
              C    J = 2 <.... END
              C
              C      LOOP TO DO J=3 THROUGH 45 ....START
              C
ISN 0027            DO 20 J=3,45
              C      SET UP EQUATION PARAMETERS
ISN 0028            DO 30 I=1,51
ISN 0029            C(I) = - ALPHA
ISN 0030            D(I) = 1. + ALPHA * ( 4. + HLM(I+1,J) )
ISN 0031            E(I) = - ALPHA
ISN 0032            IF( I .EQ. 1  .OR. I .EQ. 51) D(I) = D(I) + C(I)
ISN 0034            F(I) = Z(I+1,J) + ALPHA * ( Z(I+1,J+1) + Z(I+1,J-1) -R(I+1,J))
ISN 0035         30 CONTINUE
              C      SOLVE EQ.
ISN 0036            CALL GELIM(C,D,E,F,ANS,IER,51)
ISN 0037            IF(IER.NE.0) GO TO 9000
ISN 0039            DO 31 I=2,52
ISN 0040            Z(I,J) = ANS(I-1)
ISN 0041         31 CONTINUE
ISN 0042            Z(1,J) = Z(2,J)
ISN 0043            Z(53,J) = Z(52,J)
              C
ISN 0044         20 CONTINUE
              C
```

```
                C       END OF LOOP
                C
                C        J = 46   .... START
                C        SET UP EQUATION
  ISN 0045              DO 40 I=1,51
  ISN 0046              C(I) = - ALPHA
  ISN 0047              D(I) = 1. + ALPHA * ( 3. + HLM(I+1,46) )
  ISN 0048              E(I) = - ALPHA
  ISN 0049              IF( I .EQ. 1  .OR. I .EQ. 51) D(I) = D(I) + C(I)
  ISN 0051           40 CONTINUE
  ISN 0052              DO 41 I=1,51
  ISN 0053              F(I) = Z(I+1,46) + ALPHA * ( Z(I+1,45) - R(I+1,46) )
  ISN 0054           41 CONTINUE
                C        SOLVE EQ.
  ISN 0055              CALL GELIM(C,D,E,F,ANS,IER,51)
  ISN 0056              IF(IER.NE.0) GO TO 9000
  ISN 0058              DO 42 I=2,52
  ISN 0059              Z(I,46) = ANS(I-1)
  ISN 0060              Z(I,47) = Z(I,46)
  ISN 0061           42 CONTINUE
  ISN 0062              Z(1,46) = Z(2,46)
  ISN 0063              Z(53,46) = Z(52,46)
  ISN 0064              Z(1,47) = Z(2,46)
  ISN 0065              Z(53,47) = Z(52,46)
                C
                C        COMPLETES SOLUTION ... ALL PTS
                C
  ISN 0066              IER = 0
  ISN 0067              RETURN
  ISN 0068        9000  IER = 1
  ISN 0069              RETURN
  ISN 0070              END
*OPTIONS IN EFFEC:*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFEC:*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF
*OPTIONS IN EFFEC:*   FUNCTIONS INLINE ARE:    NONE
*OPTIONS IN EFFEC:*
*STATISTICS*     SOURCE STATEMENTS =      69, PROGRAM SIZE =     4444, SUBPROGRAM N
*STATISTICS*  NO DIAGNOSTICS GENERATED
****** END OF COMPILATION ******                                        62K BYTES O
```

REQUESTED OPTIONS. NODECK,NOLIST,OPTIMIZE(0),NOMAP,SIZE(MAX),NOIL,NOXREF,NOTERM,LC

OPTIONS IN EFFECT. NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF

FUNCTIONS INLINE ARE: NONE

```
ISN 0002          SUBROUTINE STEP2(Z,D,HLM,ALPHA,IER)
ISN 0003          DIMENSION Z(53,47),R(53,47),HLM(53,47)
ISN 0004          DOUBLE PRECISION C(51),D(51),E(51),F(51),ANS(51)
            C
            C      SET UP RELAXATION EQ.
            C      I=2   ...   START
ISN 0005          DO 10 J=2,46
ISN 0006          F(J-1) = Z(2,J) + ALPHA * ( Z(3,J) - R(2,J) )
ISN 0007       10 CONTINUE
ISN 0008          DO 11 J=1,45
ISN 0009          D(J) = 1. + ALPHA * ( 3. + HLM(2,J+1) )
ISN 0010          C(J) = - ALPHA
ISN 0011          E(J) = - ALPHA
ISN 0012          IF( J .EQ. 1  .OR.  J .EQ. 45) D(J) = D(J) + C(J)
ISN 0014       11 CONTINUE
            C      SOLVE EQ.
ISN 0015          CALL GELIM(C,D,E,F,ANS,IER,45)
ISN 0016          IF(IER .NE. 0)  GO TO 9000
ISN 0018          DO 12 J=2,46
ISN 0019          Z(2,J) = ANS(J-1)
ISN 0020       12 CONTINUE
ISN 0021          Z(2,1) = Z(2,2)
ISN 0022          Z(2,47) = Z(2,46)
ISN 0023          DO 13 J=1,47
ISN 0024          Z(1,J) = Z(2,J)
ISN 0025       13 CONTINUE
            C      I = 2   ...   END
            C
            C     LOOP TO DO I=3  THRU 51 ... START
            C
ISN 0026          DO 20 I=3,51
            C      SET UP EQUATION PARAMETERS
ISN 0027          DO 30 J=1,45
ISN 0028          C(J) = - ALPHA
ISN 0029          D(J) = 1. + ALPHA * ( 4. + HLM(I,J+1) )
ISN 0030          E(J) = - ALPHA
ISN 0031          IF( J .EQ. 1  .OR. J .EQ. 45 ) D(J) = D(J) + C(J)
ISN 0033          F(J) = Z(I,J+1) + ALPHA * ( Z(I+1,J+1) + Z(I-1,J+1) -R(I,J+1)
ISN 0034       30 CONTINUE
            C     SOLVE EQ.
ISN 0035          CALL GELIM(C,D,E,F,ANS,IER,45)
ISN 0036          IF(IER .NE. 0) GO TO 9000
ISN 0038          DO 31 J=2,46
ISN 0039          Z(I,J) = ANS(J-1)
ISN 0040       31 CONTINUE
ISN 0041          Z(I,1) = Z(I,2)
ISN 0042          Z(I,47) = Z(I,46)
            C
ISN 0043       20 CONTINUE
            C
```

```
                C       END OF LOOP
                C
                C       I=52
                C       SET UP EQUATION
  ISN 0044              DO 40 J=1,45
  ISN 0045              C(J) = - ALPHA
  ISN 0046              D(J) = 1. + ALPHA * ( 3. + HLM(52,J+1) )
  ISN 0047              E(J) = - ALPHA
  ISN 0048              IF( J .EQ. 1 .OR. J .EQ. 45) D(J) = D(J) + C(J)
  ISN 0050           40 CONTINUE
  ISN 0051              DO 41 J=1,45
  ISN 0052              F(J) = Z(52,J+1) + ALPHA * ( Z(51,J+1) - R(52,J+1) )
  ISN 0053           41 CONTINUE
                C
                C       SOLVE EQ.
  ISN 0054              CALL GELIM(C,D,E,F,ANS,IER,45)
  ISN 0055              IF(IER .NE. 0 ) GO TO 9000
  ISN 0057              DO 42 J=2,46
  ISN 0058              Z(52,J) = ANS(J-1)
  ISN 0059              Z(53,J) = Z(52,J)
  ISN 0060           42 CONTINUE
  ISN 0061              Z(52,1) = Z(52,2)
  ISN 0062              Z(52,47) = Z(52,46)
  ISN 0063              Z(53,1) = Z(52,2)
  ISN 0064              Z(53,47) = Z(52,46)
                C
                C       COMPLETES SOLUTION  ALL POINTS
                C
  ISN 0065              IER = 0
  ISN 0066              RETURN
  ISN 0067         9000 IER = 1
  ISN 0068              RETURN
  ISN 0069              END
*OPTIONS IN EFFEC,*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFEC,*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF
*OPTIONS IN EFFEC,*   FUNCTIONS INLINE ARE:   NONE
*OPTIONS IN EFFEC,*
*STATISTICS*      SOURCE STATEMENTS =       68, PROGRAM SIZE =     4410, SUBPROGRAM N
*STATISTICS*  NO  DIAGNOSTICS GENERATED
****** END OF COMPILATION ******                                          62K BYTES O
```

```
REQUESTED OPTIONS. NODECK,NOLIST,OPTIMIZE(0),NOMAP,SIZE(MAX),NOIL,NOXREF,NOTERM,LC(60
OPTIONS IN EFFECT. NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF AL
                   FUNCTIONS INLINE ARE:   NONE

                   SUBROUTINE GELIM(C,D,E,F,ANS,IER,N)
ISN 0002           DOUBLE PRECISION H(51),S(51),G
ISN 0003           DOUBLE PRECISION C(51),D(51),E(51),F(51),ANS(51)
ISN 0004           USE GAUSSIAN ELIMINATION PROCESS TO SOLVE THE MATRIX EQUATION
C                   T * (ANS) = (F), WHERE T IS A TRI-DIAGONAL MATRIX OF ORDER N,
C                  AND ANS  AND F ARE N DIMENSIONAL VECTORS.
C                   C,D,AND E ARE THE ELEMENTS OF THE MATRIX T, NOTE C(1)&E(N) DON
C                  APPEAR.
C
C                  INITIALIZE
C
ISN 0005           NN = N-1
ISN 0006           G = D(1)
ISN 0007           IF ( G .EQ. 0.) GO TO 9000
ISN 0009           H(1) = F(1) / G
ISN 0010           S(1) = F(1) / G
C
C                  ELIMINATION (FORWARD PROCESS)
C
ISN 0011           DO 10 K=2,NN
ISN 0012           G = D(K) - H(K-1) * C(K)
ISN 0013           IF( G .EQ. 0.) GO TO 9000
ISN 0015           H(K) = F(K) / G
ISN 0016           S(K) = ( F(K) - C(K) * S(K-1) ) / G
ISN 0017        10 CONTINUE
C
C                  LAST ONES  ...  NOTE H(N) = 0.
C
ISN 0018           G = D(N) - C(N) * H(N-1)
ISN 0019           IF( G .EQ. 0.) GO TO 9000
ISN 0021           S(N) = ( F(N) - C(N) * S(N-1) ) / G
C
C                  BACK SUBSTITUTION
C
ISN 0022           ANS(N) = S(N)
ISN 0023           DO 20 K=1,NN
ISN 0024           I = N - K
ISN 0025           ANS(I) = S(I) - H(I) * ANS(I+1)
ISN 0026        20 CONTINUE
ISN 0027           IER = 0
ISN 0028           RETURN
ISN 0029      9000 IER = 1
ISN 0030           RETURN
ISN 0031           END
*OPTIONS IN EFFEC.*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFEC.*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF AL
*OPTIONS IN EFFEC.*   FUNCTIONS INLINE ARE:   NONE
```

```
 LEVEL 2.2  (SEPT 76)                                  OS/360  FORTRAN H EXTENDED PLUS
REQUESTED OPTIONS. NODECK,NOLIST,OPTIMIZE(0),NOMAP,SIZE(MAX),NOIL,NOXREF,NOTERM,LC(
OPTIONS IN EFFECT. NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF

                        FUNCTIONS INLINE ARE:   NONE

   ISN 0002             SUBROUTINE STAT(Z,TSOL)
   ISN 0003             DIMENSION Z(53,47),TSOL(53,47)
               C      CALCULATE BIAS AND ROOT MEAN SQUARE ERROR ALSO THE MEAN OF Z
   ISN 0004             FACT = 51 * 45
   ISN 0005             ZMEAN = 0.
   ISN 0006             BIAS = 0.
   ISN 0007             RMSE = 0.
   ISN 0008             DO 10 I=2,52
   ISN 0009             DO 10 J=2,46
   ISN 0010             ZMEAN = ZMEAN + Z(I,J) / FACT
   ISN 0011             D = Z(I,J) - TSOL(I,J)
   ISN 0012             D2 = D*D
   ISN 0013             BIAS = BIAS + D/FACT
   ISN 0014             RMSE = RMSE + D2/FACT
   ISN 0015          10 CONTINUE
   ISN 0016             RMSE = SQRT(RMSE)
   ISN 0017             PRINT 20,ZMEAN,BIAS,RMSE
   ISN 0018          20 FORMAT(1H0,'STATISTICS OF SOLUTION',/,1H0,'MEAN Z=',F10.3,3X,
                     1  'BIAS=',E12.3,5X,'RMSE=',E12.3)
   ISN 0019             RETURN
   ISN 0020             END
*OPTIONS IN EFFEC.*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFEC.*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF
*OPTIONS IN EFFEC.*   FUNCTIONS INLINE ARE:   NONE
*OPTIONS IN EFFEC.*
*STATISTICS*      SOURCE STATEMENTS =       19, PROGRAM SIZE =       756, SUBPROGRAM N
*STATISTICS*  NO  DIAGNOSTICS GENERATED
****** END OF COMPILATION ******                                          70K BYTES O
```

```
LEVEL 2.2  (SEPT 76)                                  OS/360  FORTRAN H EXTENDED PLUS
REQUESTED OPTIONS. NODECK,NOLIST,OPTIMIZE(0),NOMAP,SIZE(MAX),NOIL,NOXREF,NOTERM,LC(
OPTIONS IN EFFECT. NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF
                        FUNCTIONS INLINE ARE:   NONE

  ISN 0002              SUBROUTINE DIFOUT(TSOL,Z,ISCAN)
  ISN 0003              DIMENSION TSOL(53,47),Z(53,47),DIF(53,47)
  ISN 0004              DIMENSION KTL(33),NTAB(9),CON(4),KNUM(4)
  ISN 0005              DATA NTAB/-1,53,47,47,47,1,1,0,53/
  ISN 0006              DATA CON/0.,1.,60.,0./
  ISN 0007              DATA KTL/4H    ,'IMPL','  HLM','HLTZ','  EQ ','SOL ',27*4H    /
  ISN 0008              DATA KNUM/'1 ST','2 ND','  SCA','N   '/
  ISN 0009              DATA KTSOL/'TSOL'/
  ISN 0010              DATA KZ/'  Z  '/
  ISN 0011              DATA KDIF/'DIF '/
  ISN 0012              KTL(9) = KNUM(ISCAN)
  ISN 0013              KTL(10) = KNUM(3)
  ISN 0014              KTL(11) = KNUM(4)
  ISN 0015              DO 1 I=1,53
  ISN 0016              DO 1 J=1,47
  ISN 0017              DIF(I,J) = Z(I,J) - TSOL(I,J)
  ISN 0018            1 CONTINUE
                C
                C        PRINT OUT SOLUTION ERROR FIELD
                C
  ISN 0019              KTL(13) = KTSOL
  ISN 0020              CALL GRDPPT(TSOL,NTAB,CON,KTL,1,1)
  ISN 0021              KTL(13)= KZ
  ISN 0022              CALL GRDPPT(Z,NTAB,CON,KTL,1,1)
  ISN 0023               KTL(13) = KDIF
  ISN 0024              CALL GRDPRT(DIF,NTAB,CON,KTL,1,1)
  ISN 0025              RETURN
  ISN 0026              END
*OPTIONS IN EFFEC,*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFEC,*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF
*OPTIONS IN EFFEC,*   FUNCTIONS INLINE ARE:   NONE
*OPTIONS IN EFFEC,*
*STATISTICS*      SOURCE STATEMENTS =       25, PROGRAM SIZE =    10882, SUBPROGRAM NA
*STATISTICS*  NO  DIAGNOSTICS GENERATED
****** END OF COMPILATION ******                                          66K BYTES OF
```

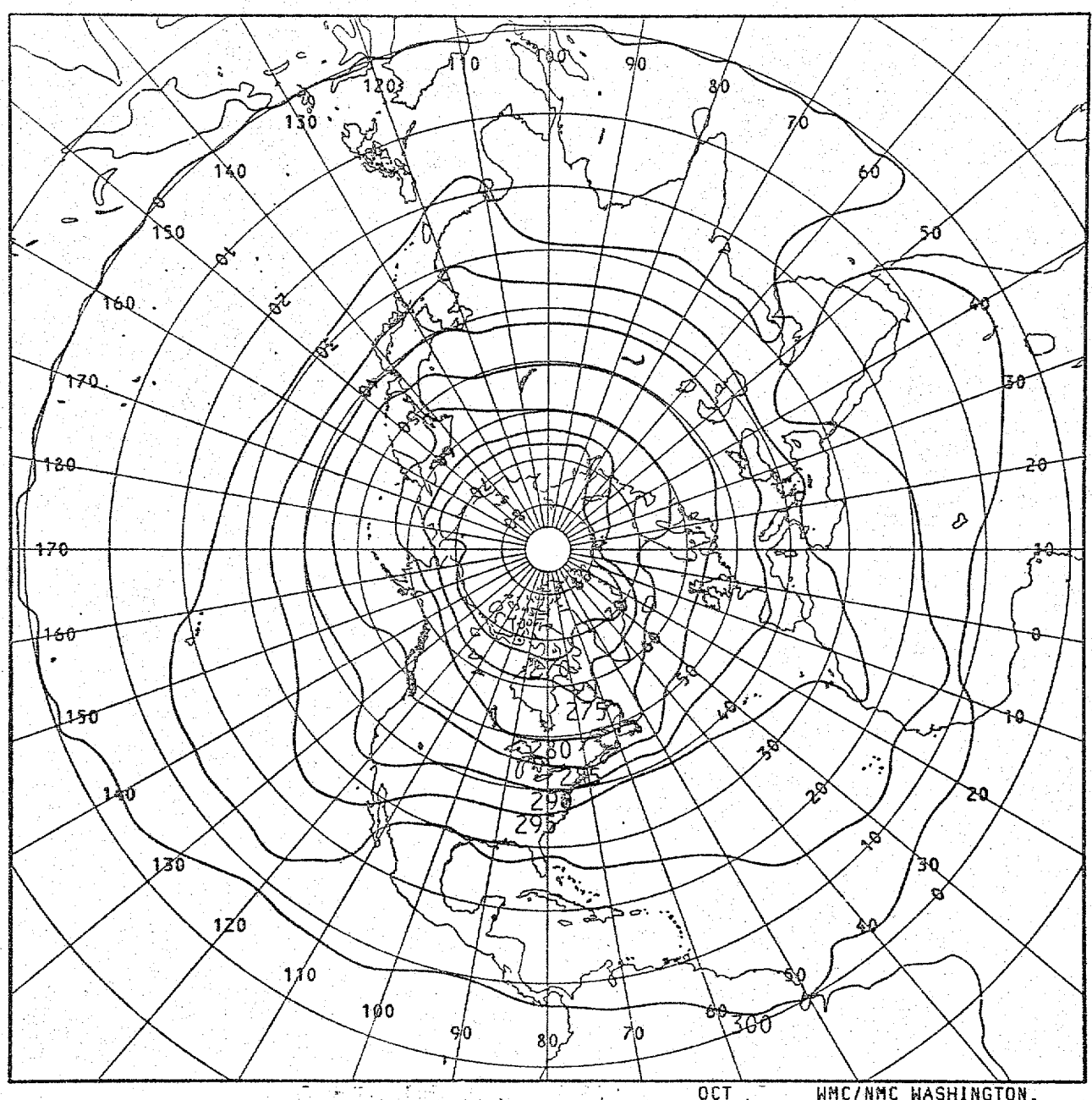

Fig. 11. October average sea surface temperatures °K: Northern Hemisphere. The 270K isotherm approximates the ice edge.

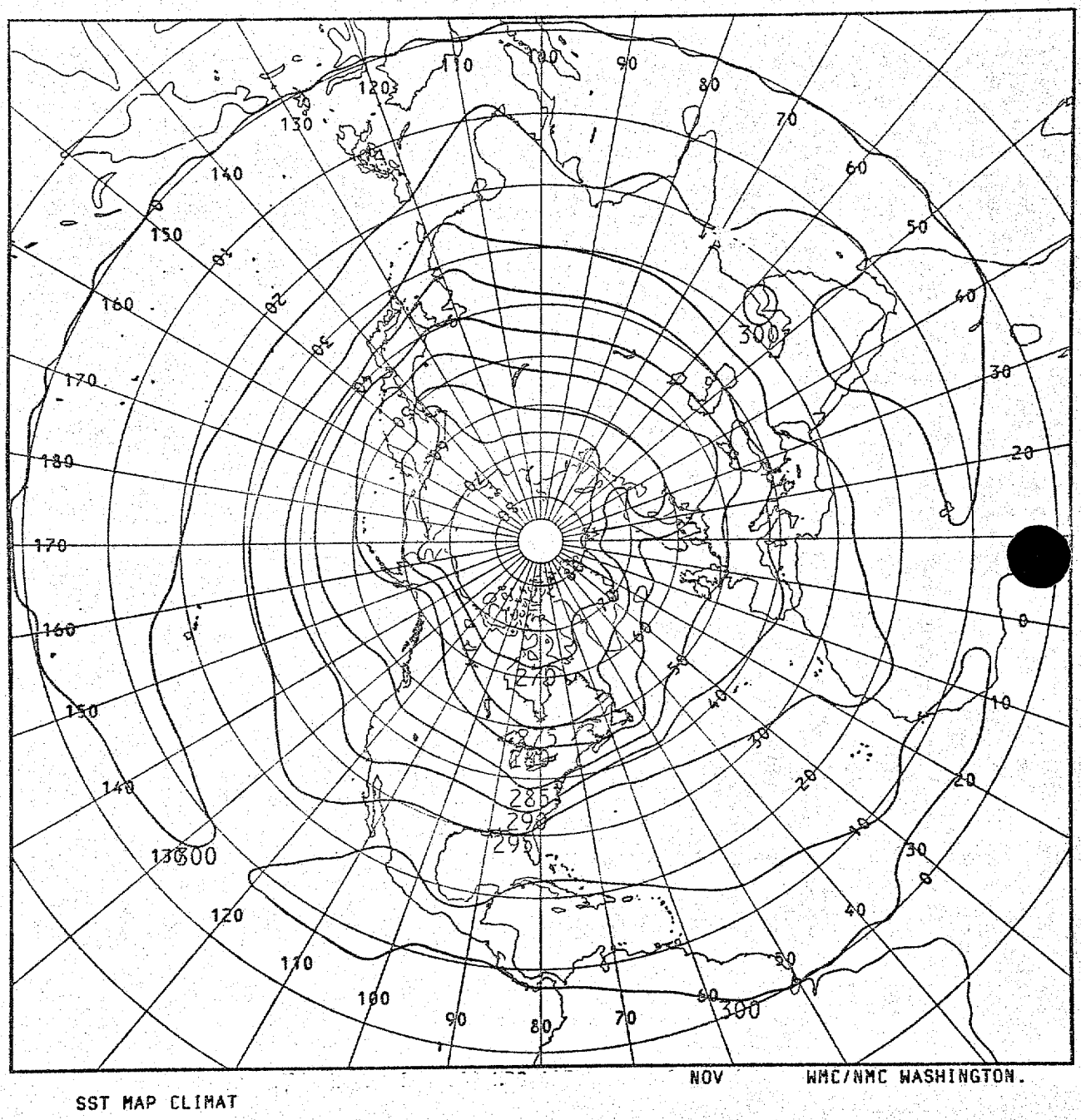

Fig. 12. November average sea surface temperatures °K: Northern Hemisphere. The 270K isotherm approximates the ice edge.

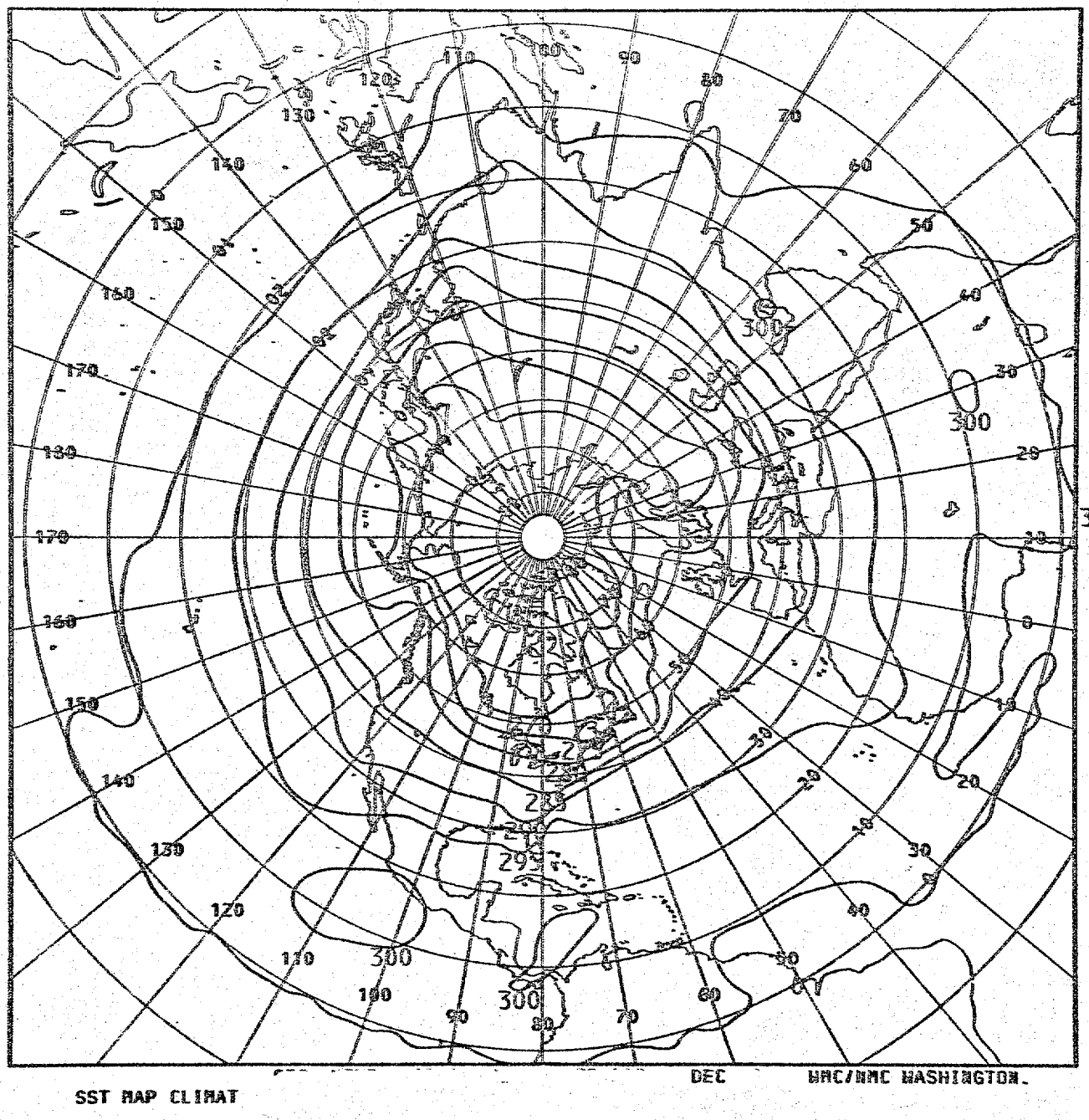

Fig. 13. December average sea surface temperatures ^{o}K: Northern Hemisphere. The 270K isotherm approximates the ice edge.

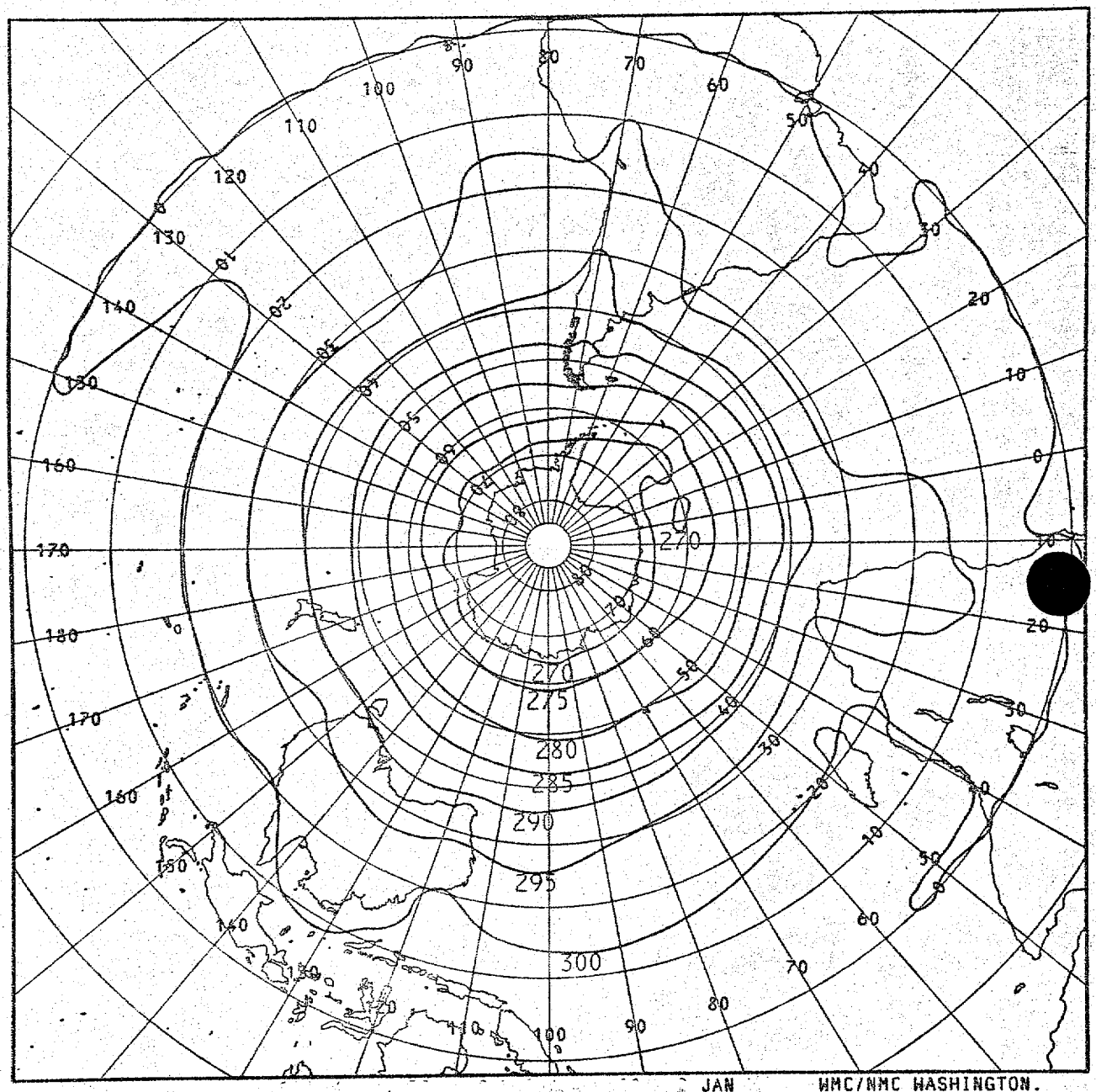

Fig. 14. January average sea surface temperatures ^{o}K: Southern Hemisphere. The 270K isotherm approximates the ice edge.

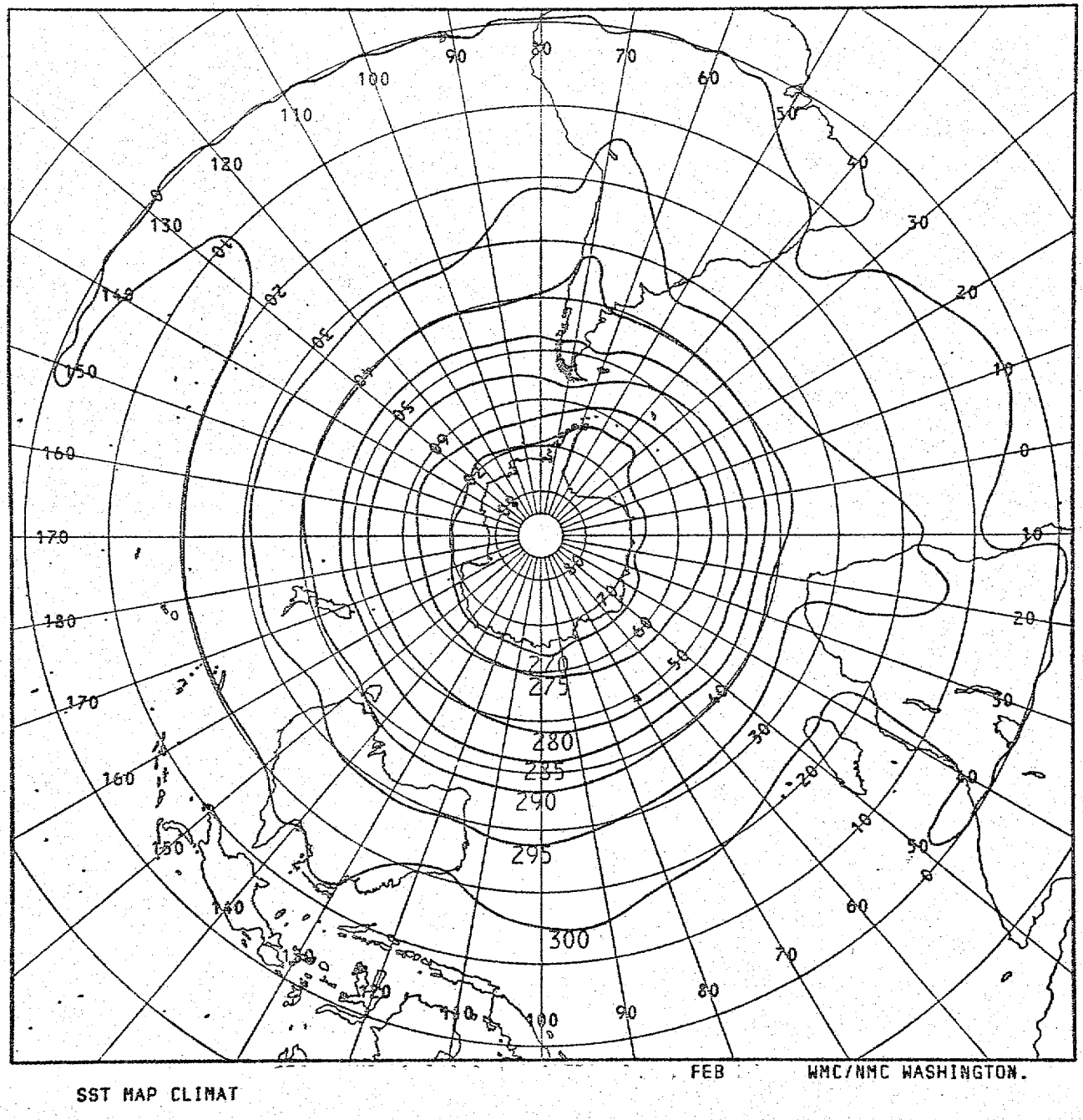

Fig. 15. February average sea surface temperatures ^{o}K: Southern Hemisphere The 270K isotherm approximates the ice edge.

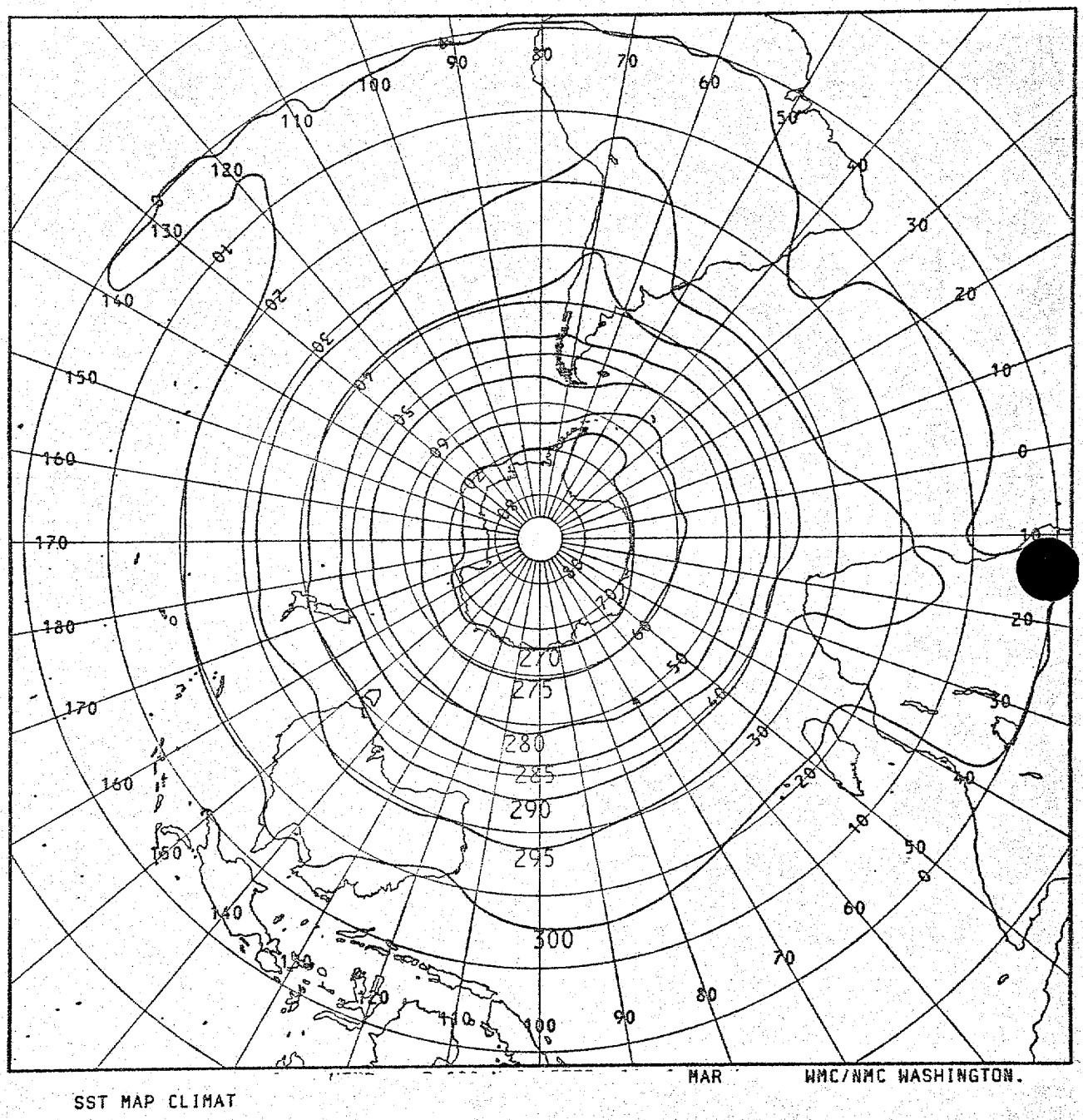

Fig. 16. March average sea surface temperatures ^{o}K: Southern Hemisphere. The 270K isotherm approximates the ice edge.

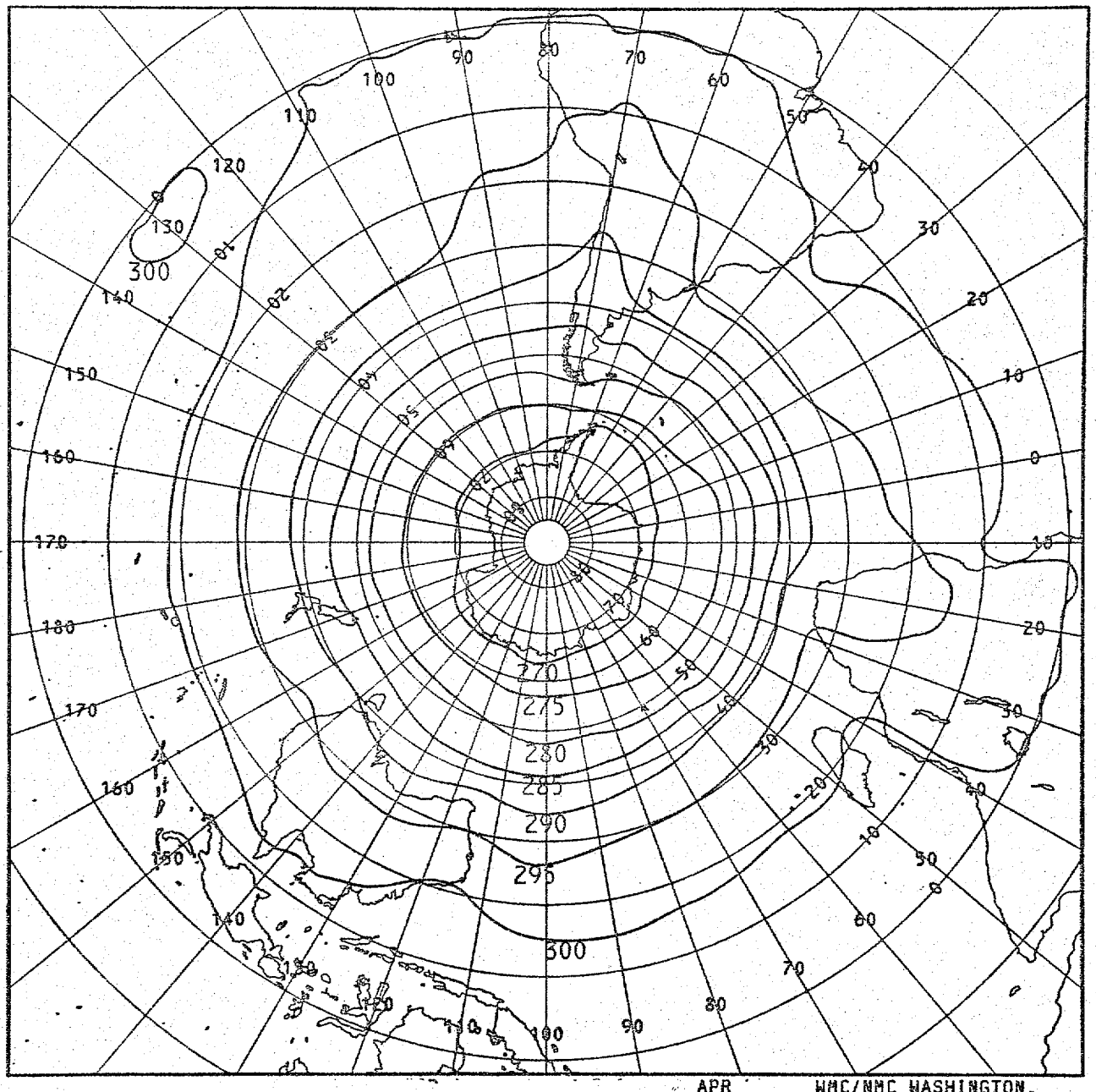

Fig. 17: April average sea surface temperatures ^{o}K: Southern Hemisphere. The 270K isotherm approximates the ice edge.

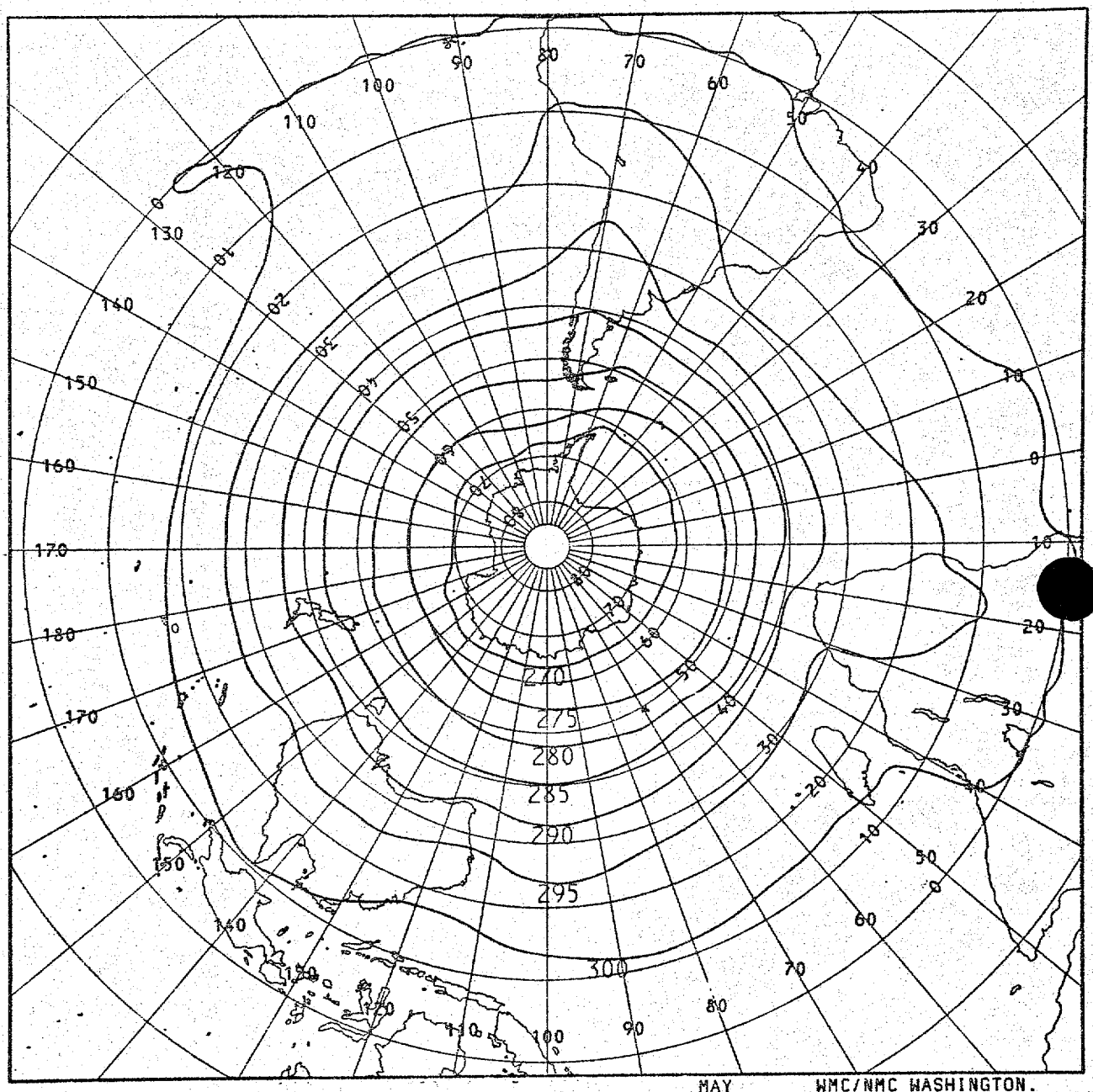

Fig. 18. May average sea surface temperatures °K: Southern Hemisphere. The 270K isotherm approximates the ice edge.

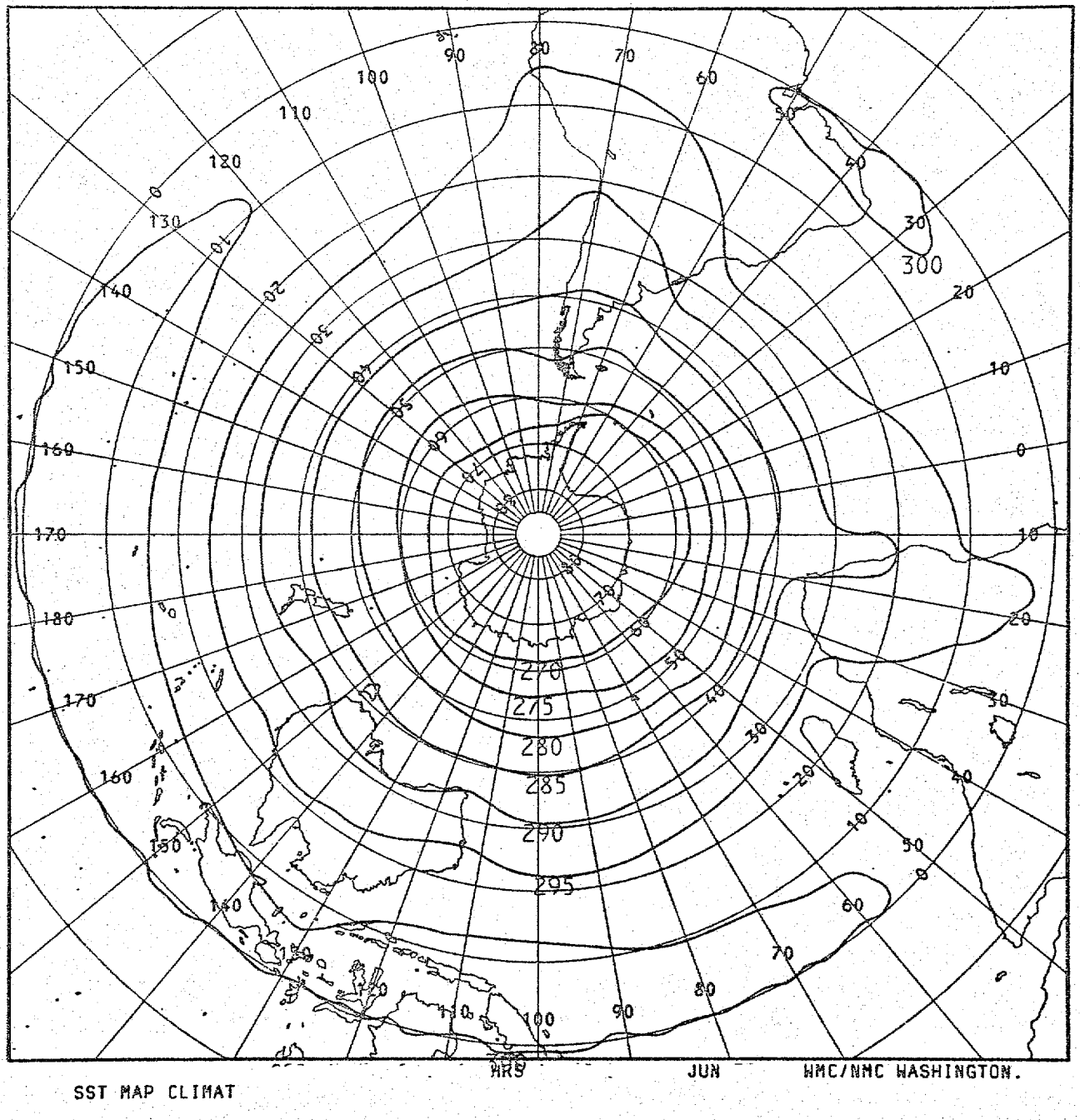

Fig. 19. June average sea surface temperatures ^{o}K: Southern Hemisphere. The 270K isotherm approximates the ice edge.

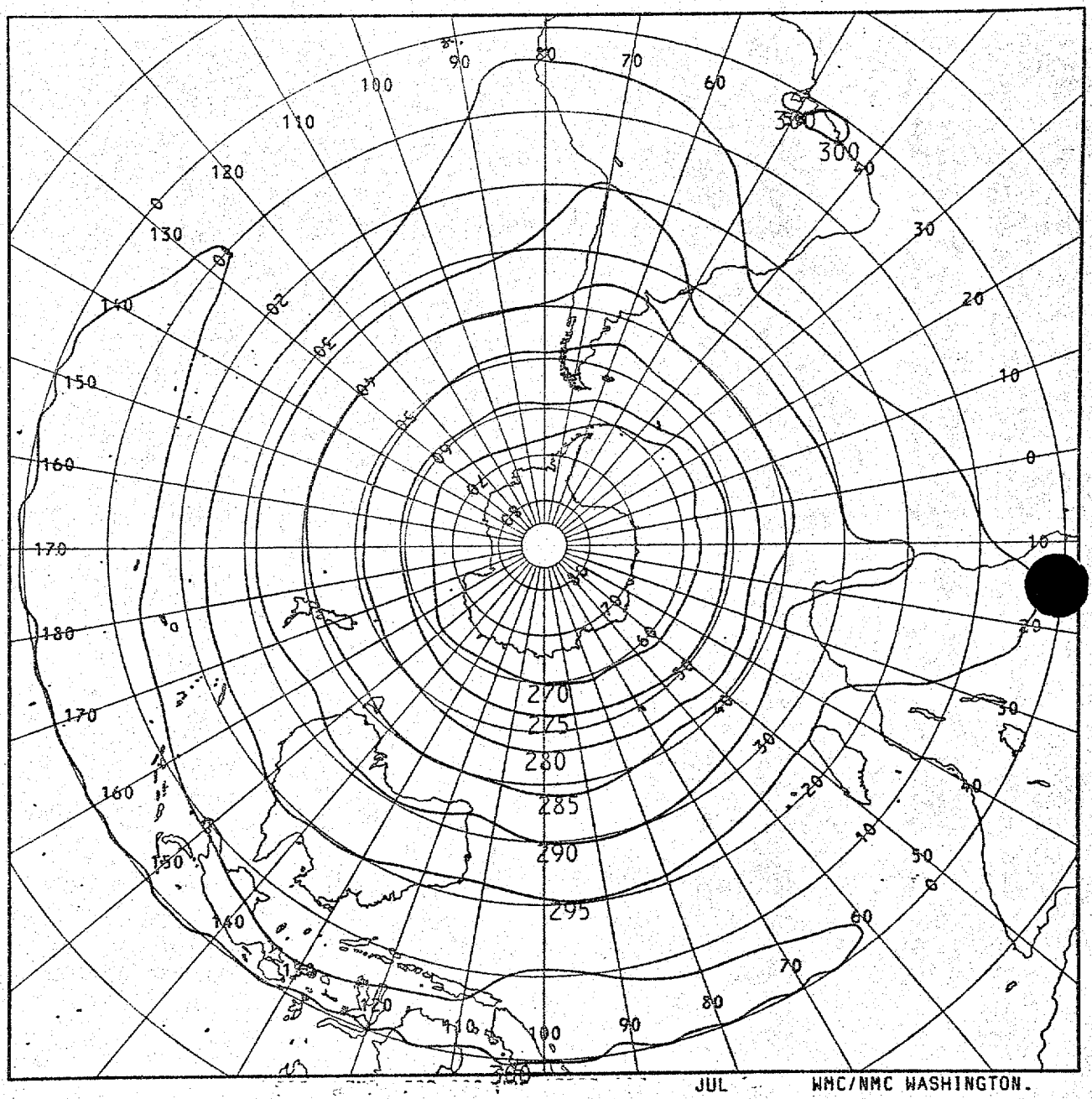

Fig. 20. July average sea surface temperatures ^{o}K: Southern Hemisphere. The 270K isotherm approximates the ice edge

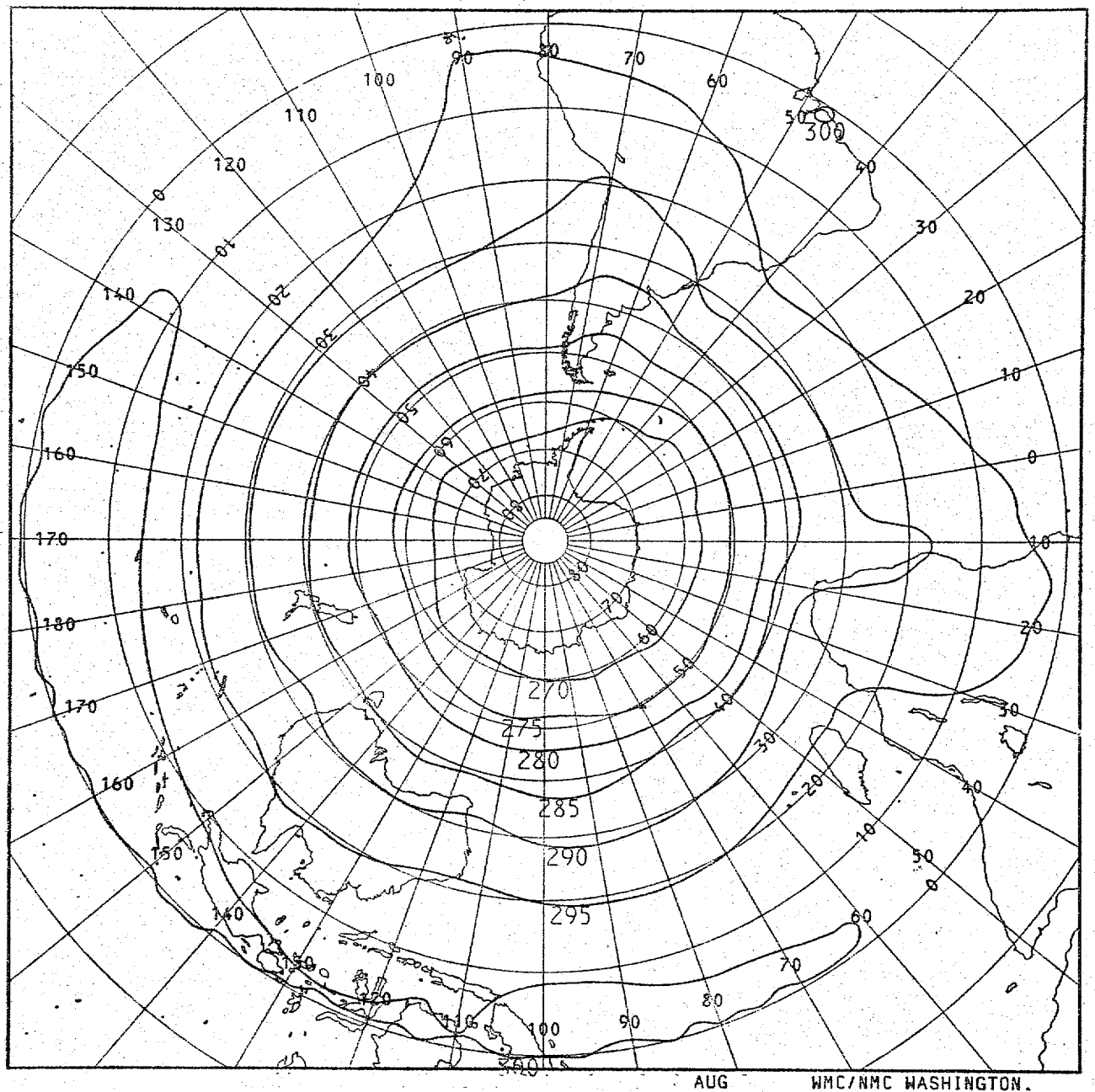

Fig. 21. August average sea surface temperatures ^{o}K: Southern Hemisphere. The 270K isotherm approximates the ice edge.

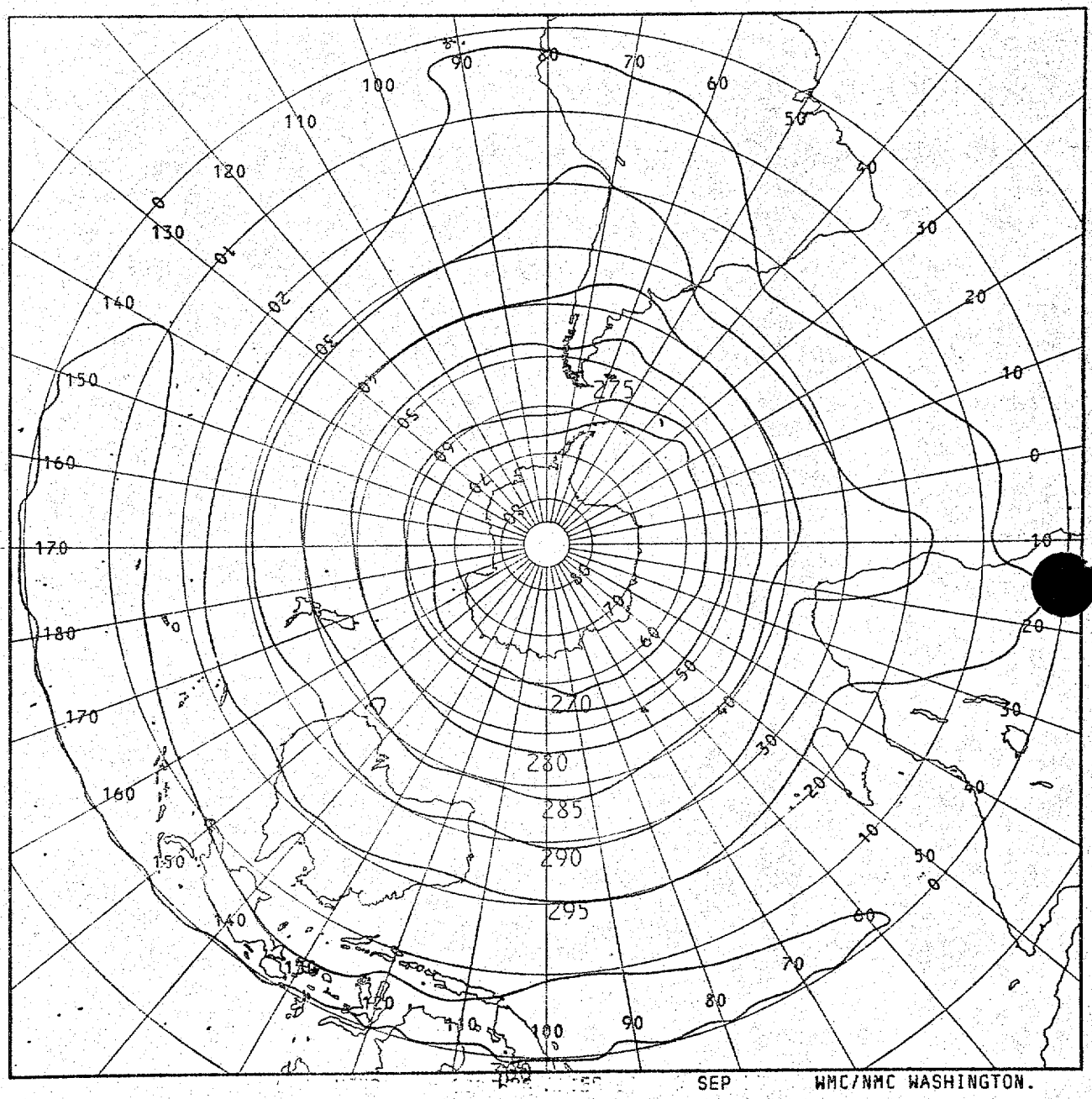

Fig. 22. September average sea surface temperatures °K: Southern Hemisphere. The 270K isotherm approximates the ice edge.

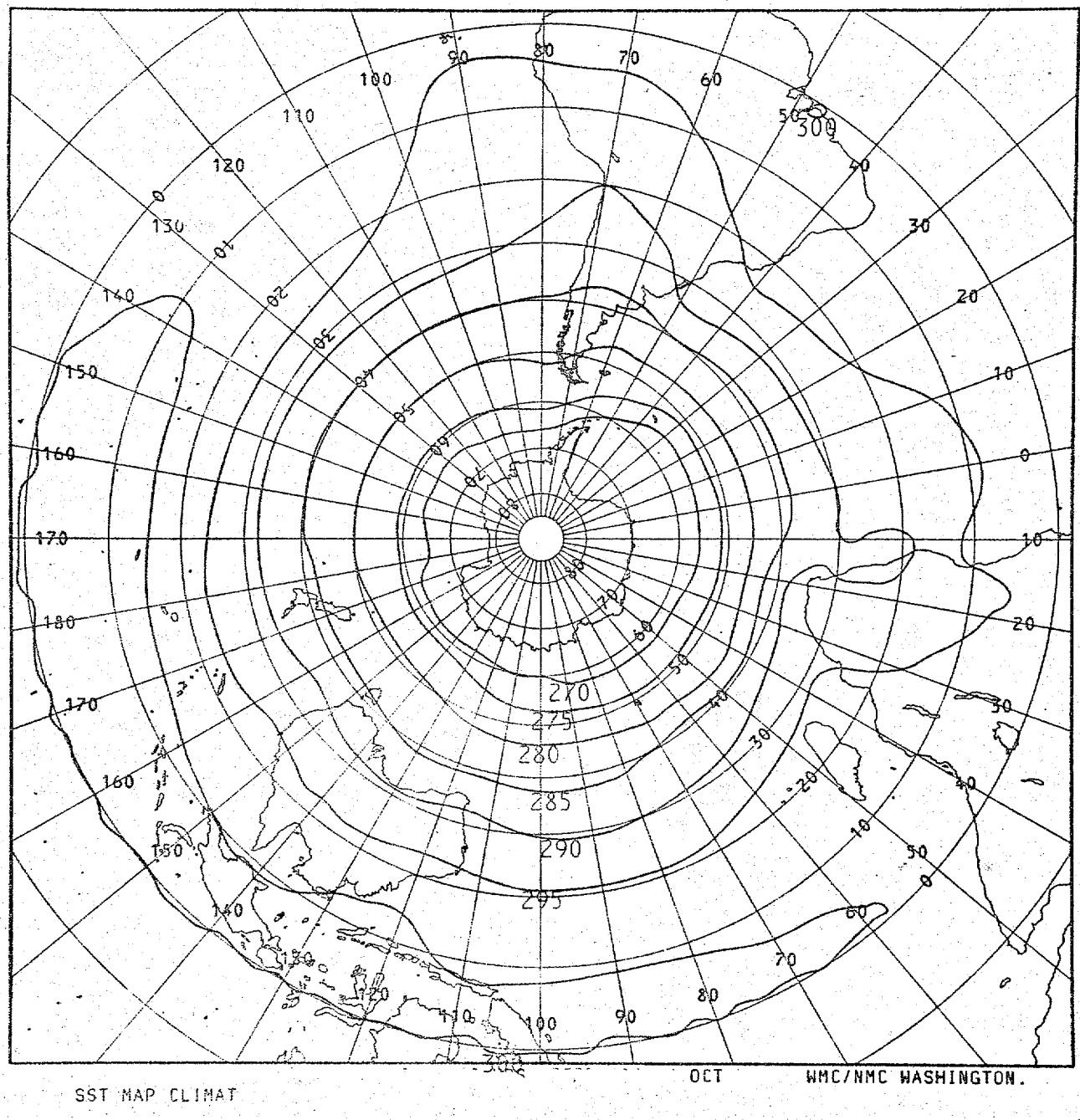

Fig. 23. October average sea surface temperatures ^{o}K: Southern Hemisphere. The 270K isotherm approximates the ice edge.

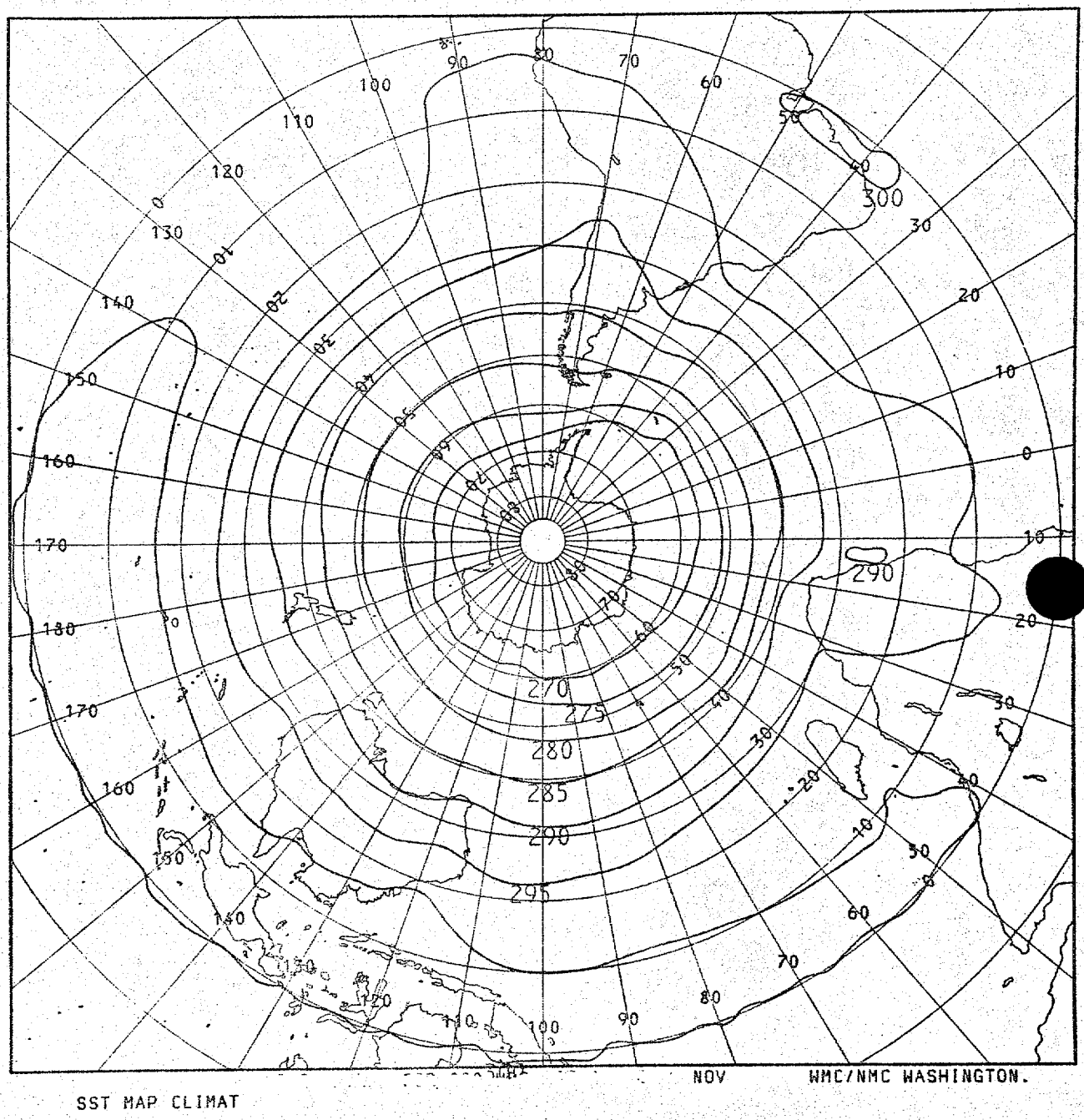

Fig. 24. November average sea surface temperatures ^{o}K: Southern Hemisphere. The 270K isotherm approximates the ice edge.

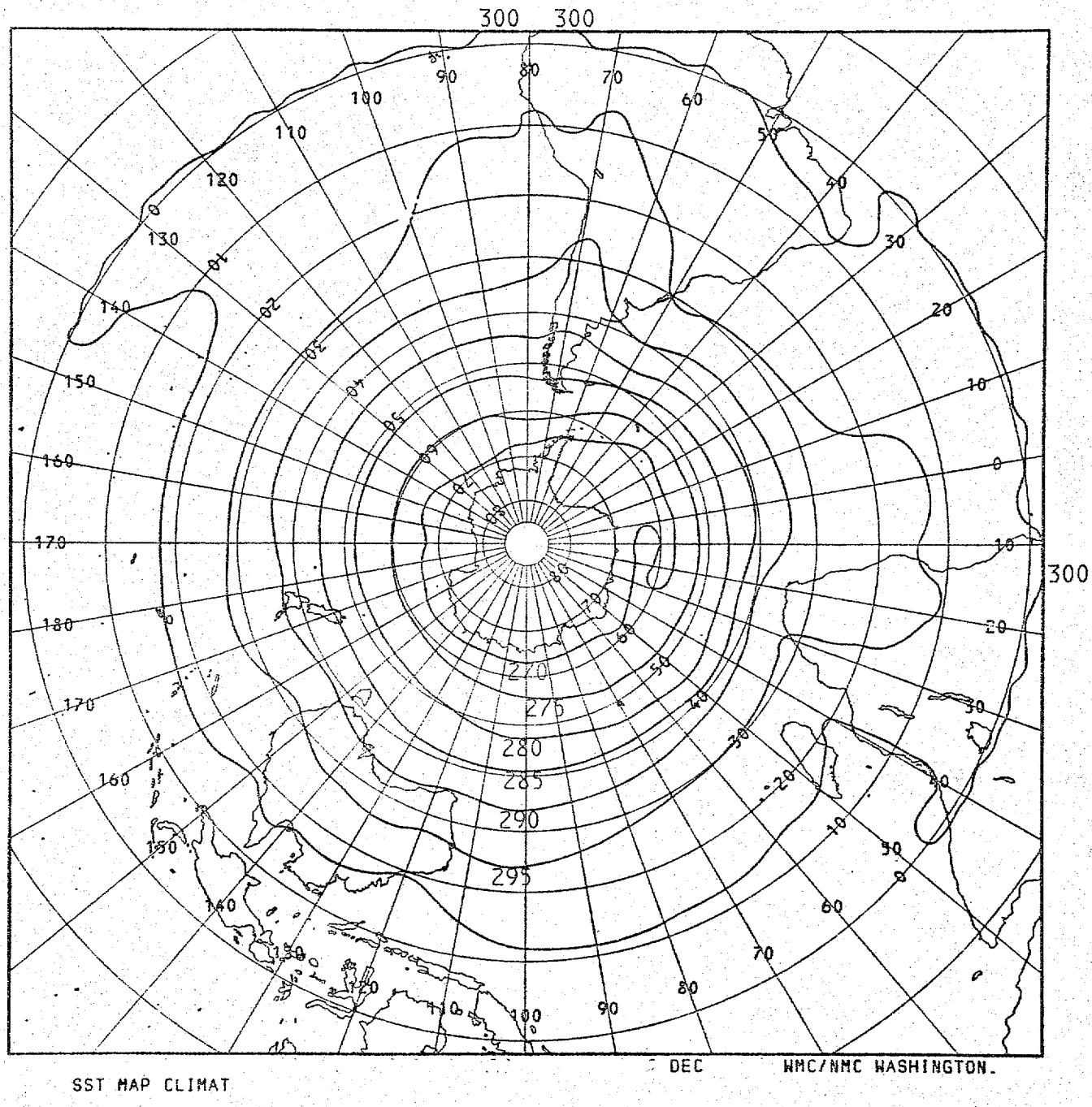

Fig. 25. December average sea surface temperatures ^{o}K: Southern Hemisphere. The 270K isotherm approximates the ice edge.