②

Naval Oceanographic and
Atmospheric Research Laboratory

Technical Note 34
July 1990

Incorporation of Radar Sea Clutter Prediction into Operational Navy Environmental Support Products: Prototype Software Development

D. W. Merdes
NAS
South Weymouth, MA

Prepared for
Forecast Guidance and Naval Systems Support Division
Atmospheric Directorate
Monterey, CA 93943-5006

Approved for public release; distribution is unlimited. Naval Oceanographic and Atmospheric Research Laboratory, Stennis Space Center, Mississippi 39529-5004.

ABSTRACT

Prototype operational FORTRAN software is developed implementing a two-scale microwave sea surface scatterometry model. A program suitable for specific numerical testing, and another program illustrating its potential operational utility in generating graphical visual aids, are also documented. Limitations of the selected scatterometry model are discussed, and suggestions on the direction of future development efforts are offered.

ACKNOWLEDGEMENTS

Program SIGMOD, from which the code in Appendices A and B was derived, was obtained from William J. Plant (NRL, presently WHOI) via Tak Kee Cheung (UCAR) and Larry D. Phegley (NOARL). Tak Kee Cheung assisted the author in the analysis of the SIGMOD code. The staff of NOARL's Atmospheric Directorate were most helpful; Buck Sampson's assistance in using the XTASY graphics software, and Gerard Vogel and Richard Titus' provision of the wind fields used to demonstrate the operation of program DNRCS (Appendix C), deserve explicit mention in that regard.

The support of the sponsor, The Space and Naval Warfare Systems Command (PMW 141), CAPT C. Hoffman, Program Element 63704N, is gratefully acknowledged.

CONTENTS

INCORPORATION OF
RADAR SEA CLUTTER PREDICTION
INTO OPERATIONAL NAVY ENVIRONMENTAL SUPPORT PRODUCTS:
PROTOTYPE SOFTWARE DEVELOPMENT

1. INTRODUCTION

It is well known that the normalized radar cross section[1] of the sea surface displays great variability under different environmental conditions. Data published by Guinard, et al (1971), for example, clearly illustrates this variability; except for a single data set, they also show a fairly regular variation with respect to a single parameter--local wind speed. In Cheung's recent report (1988) and paper (1989), which review the extensive literature relating to estimation of radar sea clutter based on available (known, estimated, or predicted) environmental information, specific reference is made to a number of theoretical models that have been published--some of which have been reduced to computer code. Cheung's report gives reason to expect that operational support software appropriate for execution on small computers with limited storage capacity and processing speed, could be developed to predict radar sea clutter with sufficient fidelity to be tactically useful.

In order to assess the technical feasibility and potential utility of an organic radar sea clutter estimation capability, one or more of the available theoretical models must be implemented in computer code designed to facilitate embedding in prototype, developmental, or even operational computer-based fleet support systems. This study reports the successful coding of a FORTRAN subroutine, termed "NRCS", which implements a two-scale microwave scatterometry model proposed by Plant (1986). Plant's model is based on the so-called "composite roughness" theory that is widely regarded as appropriate for angles of incidence in the range bounded approximately by 30 and 70 degrees ("intermediate" angles). The subroutine performs neither READ nor WRITE operations that would preclude its integration into specialized systems that lack support for ordinary FORTRAN input/output units, so a test program "TNRCS" is provided to facilitate testing for user-specified numerical inputs. Program "DNRCS" is also provided to demonstrate a concept of how the subroutine could be employed to generate a tactically useful computer-generated visual aid.

2. CODING AND TESTING AN EMBEDDED NORMALIZED RADAR CROSS SECTION SUBROUTINE

The author was fortunate to be able to start with the research-oriented FORTRAN program SIGMOD originally developed by Plant and subsequently modified by Plant and Cheung. That program successfully implements the theory devel-

---

[1] The normalized radar cross section σ_0 of a region of the ocean surface is the ratio of radar cross section to its physical area. Thus, if the radar return from a 10 m^2 region were equal to that from a perfect reflector 1 m^2 in area, the normalized radar cross section would equal 0.1, or -10 dB in logarithmic terms (dB = 10 $\log_{10} \sigma_0$).

oped in Plant (1986), but is unsuitable for embedding into operational software; it is, itself, a main program that accepts input data from a keyboard, generates an array of normalized radar cross-section values over a range of parameter values, outputs summary information to the monitor, and writes tabular information to two different disk files.

Subroutine NRCS is designed to return a single value of normalized radar cross section for a given set of parameters describing environmental conditions and microwave radiation characteristics. Because it performs no input/output operations of its own, it can be embedded in any system capable of executing FORTRAN subroutine calls and its execution is controlled by the invoking program. Appendix A lists the subroutine code, which includes comments defining the arguments. Appendix B is a listing of program TNRCS, which was used to test NRCS output against tables generated by program SIGMOD.

Because NRCS does not record intermediate quantities used in the computation, SIGMOD remains the preferred tool for research study of the behavior of Plant's model as a function of various inputs.

3. ILLUSTRATING THE POTENTIAL TACTICAL UTILITY OF SEA CLUTTER ESTIMATION

Program DNRCS was written to demonstrate the potential tactical utility of an organic capability to estimate radar sea clutter. It accesses a data file containing wind fields over a region of the ocean obtained from an operational weather product from the Fleet Numerical Oceanography Center. Calls to NRCS are executed for each of the grid points, and three graphical displays are then generated. Figures 1 and 2 show, respectively, absolute and logarithmic contour plots from DNRCS of normalized radar cross section. Of greater interest would be something like Figure 3, which displays estimated signal-to-noise-ratio of a target of interest with respect to a background of sea clutter. Appendix C is a program listing of DNRCS.

4. ENGINEERING A FLEXIBLE SOFTWARE PACKAGE TO FACILITATE EMBEDDING OF SEA CLUTTER PREDICTION IN OPERATIONAL SYSTEMS

One can imagine a number of operational scenarios in which radar-dependent surveillance sensors and weapons could be affected by sea clutter; hence, the ability to estimate current effects of clutter, and to predict them, could be advantageous to a tactical commander. Appendix D sketches out a simple conceptual software package, Environmental_Sea_Clutter, in pseudocode mimicking the Ada computer language, which would provide three different levels of information on environmentally-related sea clutter. The lowest-level procedure, NRCS, would estimate normalized radar cross section as a function of environmental parameters that could reasonably be expected to be available aboard a naval platform either from direct observation or from routine external environmental support sources. The FORTRAN subroutine of Appendix A can be considered a first prototype of this procedure, but a model of somewhat higher fidelity would probably be needed. Treatment of near-normal and near-grazing incidence angles will also have to be included.

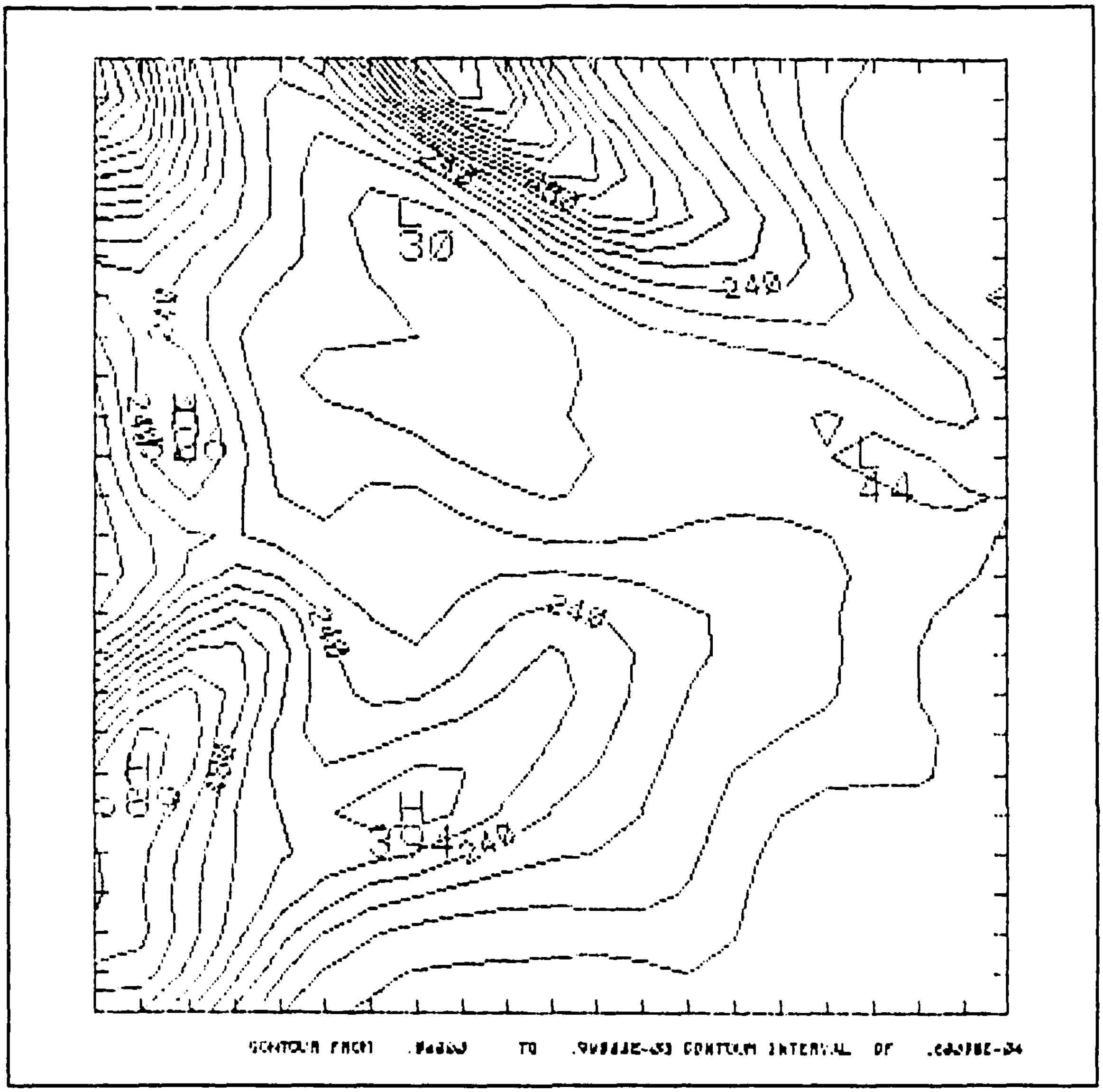

Figure 1. Contour plot of normalized radar cross section σ_0 generated by program DNRCS.

It is unlikely that the output of NRCS would be of direct interest to operational commanders, whose concern would inevitably focus on the capability of a given radar-based system (his own or an adversary's) to perform effectively. A higher-level procedure, ERCS, would be required to estimate environmental radar cross section based on NRCS and parameters characterizing the radar itself (e.g. pulse duration, beamwidth) and its location with respect to a target of interest (e.g. altitude, azimuth). Procedure TSNR would then be employed to estimate signal-to-noise ratio of the target with respect to environmentally-generated clutter, based on the ERCS and the target's own radar cross section. The latter procedure could be used to generate displays along the lines of Figure 3, and could also serve as the basis for higher-

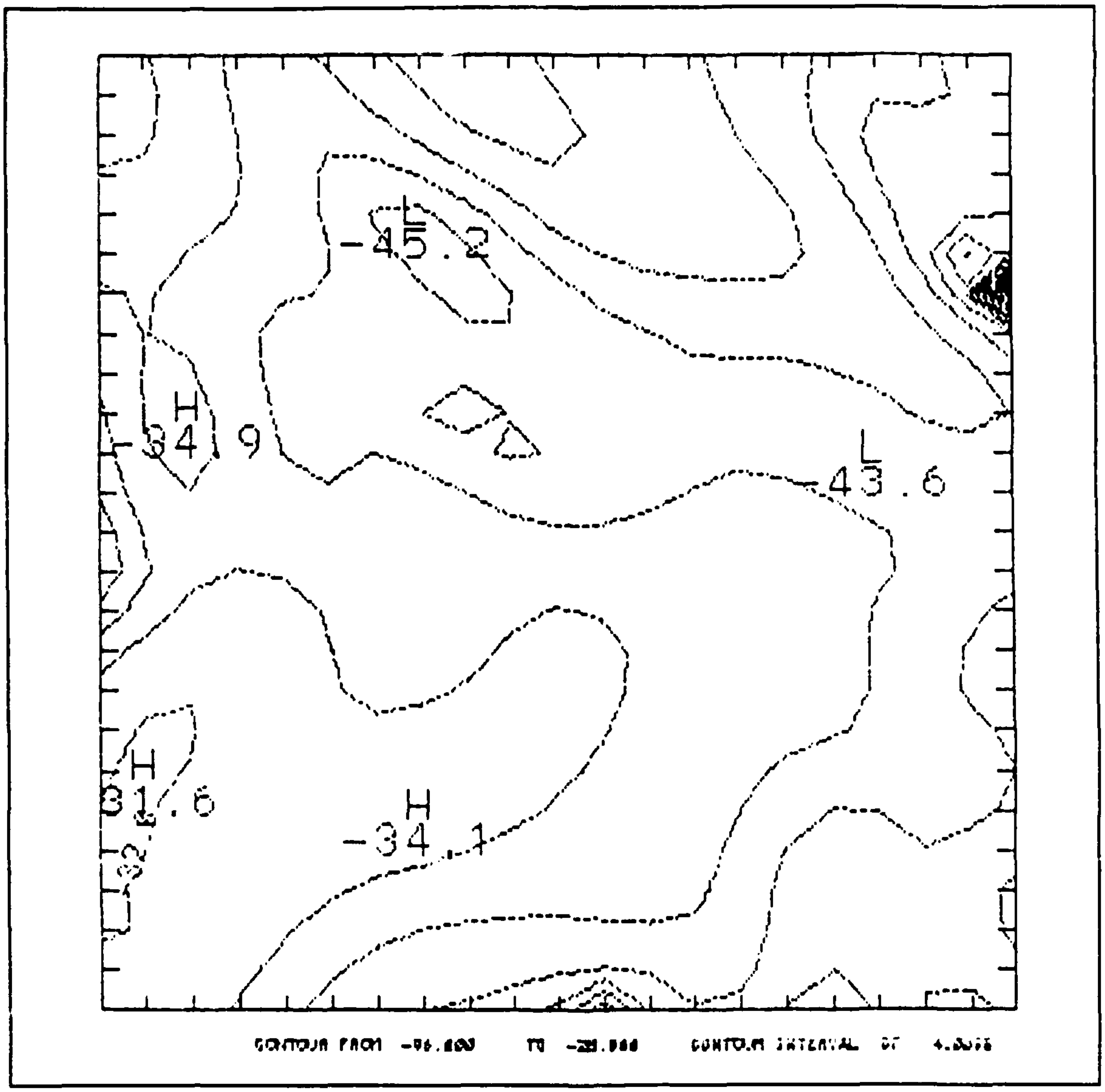

Figure 2. Contour plot of normalized radar cross section in dB (dB = 10 $\log_{10} \sigma_0$). Same field as Figure 1.

level displays of where certain targets would or would not be detectable by the radar in question.

The software engineering effort to develop Environmental_Sea_Clutter would probably identify other modules for that generic "toolbox" package.

5. RECOMMENDATIONS FOR FUTURE STUDY

Efforts to continue the line of development initiated in this study can logically be divided into three categories: (1) identification of where, and to what extent, a Navy need exists for an organic radar sea clutter estimation

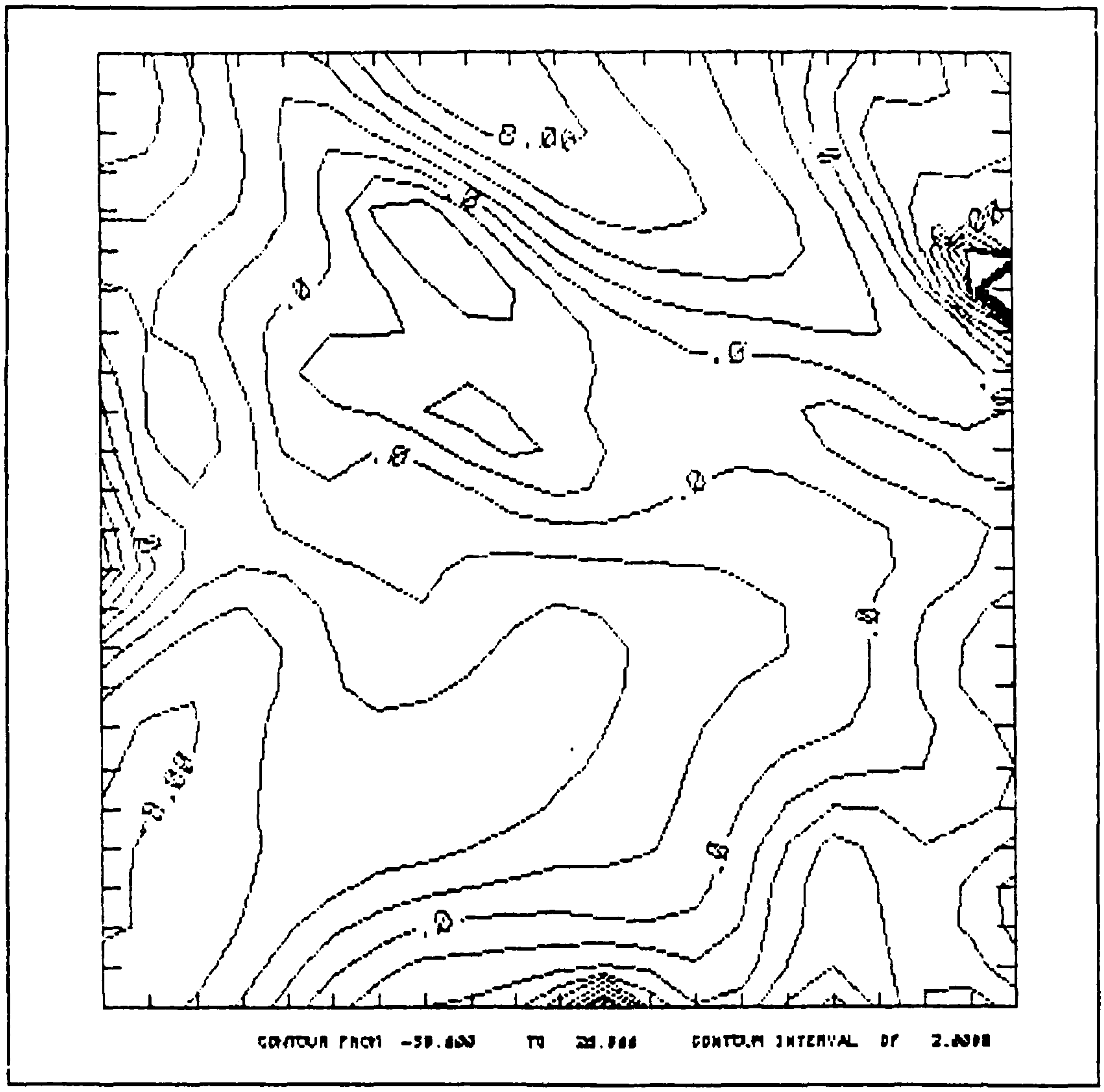

Figure 3. Estimated signal-to-noise ratio (SNR) in dB of a target of interest with respect to a background of sea clutter. Same field as Figure 1.

capability--to include conceptual development of how this kind of information could be displayed, or otherwise used, in existing/developmental/proposed computer-based systems; (2) analysis to identify what requirements the above needs-oriented study imposes on a "toolbox" package along the lines of Appendix D; and (3) development of the lowest-level module "NRCS", with fidelity sufficient to satisfy operational requirements without violating the constraints that only readily available data can be used and that computational load not exceed the organic resources of operational platforms.

The first and second categories of effort together constitute a top-down exploratory development (6.2) program that, if successful, would lead to transitions into advanced development (6.3) programs and/or upgrades to existing fleet systems. Hopefully, the mainline 6.2 effort would find support as part of one of the Office of Naval Technology's exploratory development block programs. Possibly, a case could be made now to develop an application for some particular developmental or operational system, in which case 6.3 (or higher category) funding should be sought from the appropriate sponsor.

In the case of the third category--the effort to implement the required underlying mathematical capability--there is probably a need for a top-down effort to define the required capabilities as well as a bottom-up effort to determine whether adequate theoretical models are available. Requirements definition and selection of models to be implemented would seem to belong in the 6.2 arena, whereas 6.1 funding should be sought for whatever fundamental research is needed to address deficiencies in the current "state-of-the-art". Since it appears highly unlikely that a single model will be valid over the entire possible range of incidence angles (0 to 90 degrees), the module that finally emerges from this effort will need to exhibit smooth behavior in the transition regions between models appropriate to near-normal, intermediate, and grazing incidence angles.

There is a definite liklihood that additional fundamental research will be needed. Plant, 1986, states that his proposed model is incomplete and is mainly intended to indicate which physical variables require additional investigation for algorithm development. The principal deficiency of this model is probably the fact that the long-wave ("swell") component of the local surface wave motion is assumed to have reached equilibrium for the specified wind parameters. (Note that Subroutine NRCS of Appendix A does not allow for specification of the direction of long-wave motion.) Plant addresses this and other deficiencies in the conclusion of his paper. It is worth noting that Plant's work is focused on the problem of estimating the values of certain atmospheric and oceanic parameters based on satellite-mounted scatterometry data, which is the inverse of our desire to estimate sea clutter as a function of environmental parameters; perhaps a research effort explicitly pursued from the latter perspective would be fruitful.

REFERENCES

Cheung, T. K., 1988: The Effects of Better Environmental Inputs in Estimating Sea Clutter, Technical Report TR 88-01, Naval Environmental Prediction Research Facility, Monterey, CA.

Cheung, T. K., 1989: *Can Better Environmental Inputs Improve Sea Clutter Estimation? A Numerical Experiment*, Int. J. Remote Sensing, 10, 23-35.

Guinard, N. W., J. T. Ransone, and J. C. Daley, 1971: *Variation of the NRCS of the Sea with Increasing Roughness*, J. Geophys. Res., 76, 1525-1538.

Plant, W. J., 1986: *A Two-Scale Model of Short Wind-Generated Waves and Scatterometry*, J. Geophys. Res., 91, 10735-10749.

Line# Source Line   Microsoft FORTRAN Optimizing Compiler Version 4.01

```
C   SUBROUTINE NRCS                                       VERS 16 MAR 90
C
C   MODULE TO COMPUTE THE NORMALIZED RADAR CROSS SECTION OF THE OCEAN
C    SURFACE AS A FUNCTION OF ENVIRONMENTAL FACTORS
C
C   ARGUMENTS:  [(R) DENOTES REAL, (I) DENOTES INTEGER]
C    INPUTS:
C     WNM   (R) RADAR WAVENUMBER (INVERSE CM)
C               RANGE:  WNM > 0
C     POL   (I) POLARIZATION (DIMENSIONLESS INDEX)
C               POL = 0 IMPLIES VERTICAL, POL = 1 IMPLIES HORIZONTAL
C     PHI   (R) ANGLE OF INCIDENCE (RADIANS)
C               RANGE:  0 < PHI < PI/2
C     CHI   (R) AZIMUTH WITH RESPECT TO WIND (RADIANS)
C               RANGE:  0 <= CHI < 2*PI
C     SLOPE (R) MEAN LONG-WAVE SLOPE
C               RANGE:  0 < SLOPE [OR SLOPE = -1.0 FOR COMPUTED DEFAULT]
C     WIND  (R) WIND SPEED (M/SEC)
C               RANGE:  0 <= WIND <= 26 M/SEC
C    OUTPUTS:
C     SIGO  (R) NORMALIZED RADAR CROSS SECTION (DIMENSIONLESS)
C               RANGE:  0 < SIGO
C     SIGDB (R) ...(?)...
C     CAODB (R) ...(?)...
C     IRET  (I) RETURN CODE:
C       0 - SUCCESSFUL RETURN
C      -1 - WNM   OUT OF RANGE
C      -2 - POL   VALUE ILLEGAL
C      -3 - PHI   OUT OF RANGE
C      -4 - CHI   OUT OF RANGE
C      -5 - A     OUT OF RANGE
C      -6 - SLOPE OUT OF RANGE
C      -7 - WIND  OUT OF RANGE
C       1 - UW/C0 > 1.7 (INTERMEDIATE COMPUTATION FAILURE CONDITION)
C      99 - UNSPECIFIED FAILURE
C
C    EMPIRICAL PARAMETERS:
C     A     (R) GROWTH RATE PARAMETER OF PAGE 10,737 (DIMENSIONLESS)
C     EPS   (R) CROSSWIND/UPWIND SLOPE RATIO (DIMENSIONLESS)
C     AA    (R) PARAMETER "A0" OF EQ (49)
C     BB    (R) PARAMETER "A1" OF EQ (49)
C     CC    (R) PARAMETER "A2" OF EQ (49)
C
      SUBROUTINE NRCS(WNM,POL,PHI,CHI,SLOPE,WIND,
     1                IRET,SIGO,SIGDB,CAODB)
C
C  ARITHMETIC CONSTANTS:
      REAL PI,HALFPI,THPI,TWOPI
      PARAMETER ( PI = 3.1415927, HPI = PI/2.0,
     1            THPI = 3.0*HPI, TWOPI = 2.0*PI )
C
C  EMPIRICAL CONSTANTS (PER PAGE 10,741):
      REAL A,AA,BB,CC
```

```
      PARAMETER ( A = 0.05, EPS = 0.4, AA = 1.0, BB = 0.05, CC = -1.0 )
C
C  TYPE DECLARATIONS:
      INTEGER POL
      REAL WNM,PHI,CHI,SLOPE,WIND
      INTEGER IRET
      REAL SIGO,SIGDB,CAODB
C
      REAL SSIN,CCOS,TTAN,CCOT,CD,WNV,CWIND,LWSLP
      REAL PMV,PM2V,PMH,PM2H,G,GV,GH,T3
C
C  INITIALIZATION:
      IRET  = 99
      SIGO  = -1.0
      SIGDB = 1000000.0
      CAODB = 1000000.0
C
C  ENSURE INPUTS ARE WITHIN ALLOWED RANGE:
      IF (WNM.LE.0.0) THEN
          IRET = -1
          GOTO 900
       ENDIF
      IF ( (POL.NE.0).AND.(POL.NE.1) ) THEN
          IRET = -2
          GOTO 900
       ENDIF
      IF ( (PHI.LE.0.0).OR.(PHI.GE.HPI) ) THEN
          IRET = -3
          GOTO 900
       ENDIF
      IF ( (CHI.GE.0.0).AND.(CHI.LT.TWOPI) ) THEN
          IF ( (CHI.LT.HPI).OR.(CHI.GT.THPI) ) THEN
              THETA = CHI
            ELSE
              THETA = CHI - PI
           ENDIF
        ELSE
          IRET = -4
          GOTO 900
       ENDIF
      IF (A.LE.0.0) THEN
          IRET = -5
          GOTO 900
       ENDIF
      IF ( (SLOPE.LE.0.0).AND.(SLOPE.NE.-1.0) ) THEN
          IRET = -6
          GOTO 900
       ENDIF
      IF ( (WIND.LT.0.0).OR.(WIND.GE.26.0) ) THEN
          IRET = -7
          GOTO 900
       ENDIF
      SSIN = SIN(PHI)
```

```
      CCOS = COS(PHI)
      TTAN = SSIN/CCOS
      CCOT = 1/TTAN
      CWIND = 100.0*WIND
C
C+++++++++++++++++COMPUTE WIND AND WAVE PARAMETERS+++++++++++++++++++
C
C  DRAG COEFFICIENT "CD" - EQ (46):
      IF (WIND.LE.(4.0)) THEN
                                          CD = 0.00114
        ELSEIF (WIND.LT.10.0) THEN
          CD = 0.00114
        ELSEIF (WIND.LT.(26.0)) THEN
          CD = 0.00049 + 0.000065*WIND
       ENDIF
C
C  FRICTION VELOCITY "USTAR":
      USTAR = CWIND*SQRT(CD)
C
      WNW= 2.0*WNM*SSIN
      CO = SQRT( 981/WNW + 74.0*WNW )
C
C  "UW" - EQ (36):
      UW = 0.6*USTAR - 0.084*USTAR*ALOG( 1.0 + 92.0/WNW )
      IF (UW.GT.(1.7*CO)) THEN
          IRET = 1
          GOTO 900
       ENDIF
C
C  "PMO" - EQ (49):
      PMO = AA + BB*USTAR + CC*WNW
C
      C   = CO + UW*COS(THETA)
      AFR = WNW*C
      GTK = 981.0 + 222.0*WNW*WNW
      CG  = GTK/(2.0*WNW*CO)
     1     + (UW+7.728*USTAR/(WNW+92.0))*COS(THETA)
      DCGDK = -GTK**2/(4.0*WNW**3*CO**3) + 222.0/CO
     1       + 7.728*USTAR*COS(THETA)*(92./(WNW*(WNW+92.)))/(WNW+92.)
      CGC = CG/C
      BETO = 0.04*USTAR**2*AFR/C**2
C
C  "BETW" - EQ (38):
      BETW = 0.04*USTAR**2/(CO+0.5*UW)**2
C
C  "R" - EQ (C4):
      R   = UW/(CO+0.5*UW)
C
C  EXPANSION COEFFICIENTS IN "R" - EQ (C8):
      R1 = 0.64 - 0.36*R + 0.25*R**2
      R2 = 0.42 - 0.58*R + 0.34*R**2
      R4 = 0.08 + 0.08*R - 0.16*R**2
      R6 = 0.04 + 0.04*R + 0.02*R**2
```

```
C
C+++++++++++++++++++++COMPUTE mp & Ti VALUES+++++++++++++++++++++++++++
C
      GH = (CCOS/(0.111*CCOS+1.0))**4
      GV = CCOS**4*(1.0+SSIN**2)**2/(CCOS+0.111)**4
      IF (POL.EQ.0) THEN
          PMV  = 2.0*SIN(2*PHI)/(1.0+SSIN**2) - 4.0*TTAN
     1                + 4.0*SSIN/(CCOS+0.111)
          PM2V = PMV**2 - 4.0*(3.0*CCOS**2-2.0)/(1+SSIN**2)**2
     1                + 4.0*(1.0+0.111*CCOS)/(CCOS+0.111)**2
     2                - 4.0/CCOS**2
          PM2P = PM2V
          PMP  = PMV
          G    = GV
          T3   = 2.0*(SQRT(GH/GV) - 0.5)/SSIN**2
        ELSEIF (POL.EQ.1) THEN
          PMH  = -4.0*TTAN + 0.444*SSIN/(0.111*CCOS+1.0)
          PM2H = PMH**2 - 4.0/CCOS**2
     1                + 0.444*(0.111+CCOS)/(0.111*CCOS+1.0)**2
          PM2P = PM2H
          PMP  = PMH
          G    = GH
          T3   = 2.0*(SQRT(GV/GH) - 0.5)/SSIN**2
        ELSE
          IRET = 99
          GOTO 900
       ENDIF
      T2 = PM2P/2.0 - PMP*CCOT*(2.0*CGC+2.0)
     1     + (4.0*CCOT**2+1.0)*CGC + 3.0*CCOT**2 + 1.0
     2     + (3.0*CGC**2 - WNW*DCGDK/C)*CCOT**2
      T1 = PMP + CCOT*(WNW*CC/PM0-2.0*CGC-2.0)
C
C++++++++++++COMPUTE SLOPE AND COEFFICIENTS Ai++++++++++++++++++++++++++
C
      IF (SLOPE.EQ.-1.0) THEN
          LWSLP = 0.008 + 0.0000156*CWIND
        ELSE
          LWSLP = SLOPE
       ENDIF
      A00 =  BETW*R1
      A01 =  LWSLP*( 0.5*BETW*R1*(T2+T3)
     1      + 0.25*BETW*R2*(1.0-2.0*EPS)*(T2-T3)
     2      - 0.5*CGC*PM0/AFR + 0.5*WNW*CC/AFR )
      A0  =  A00 + A01
      A1  =  LWSLP*( 0.5*PM0*T1*BETW*(R1+0.5*R2) )
      A20 =  BETW*R2
      A21 =  LWSLP*(0.5*BETW*R2*(T2+T3)
     1      + 0.25*BETW*(2.0*R1-R4)*(1.0-2.0*EPS)*(T2-T3)
     2      - (2.0+0.5*CGC)*PM0/AFR + 0.5*WNW*CC/AFR)
      A2  =  A20 + A21
      A2P = -LWSLP*(PM0*UW*SIN(THETA)*0.5/(C*AFR))
      A3  =  LWSLP*(0.25*PM0*T1*BETW*(R2-R4))
      A40 = -BETW*R4
```

```
      A41 =  LWSLP*( -0.5*BETW*R4*(T2+T3)
     1     + 0.25*BETW*(R2+R6)*(1.0-2.0*EPS)*(T2-T3) )
      A4  =  A40 + A41
C
C+++++++++++++++++EVALUATE SIGMA0 AND TAKE LOG++++++++++++++++++++++++++
C
        COEF  = PI*A*G/SSIN**4
        SIG0  = COEF*( A0 + A1*COS(CHI)
     1          + A2 *COS(2.0*CHI) + A3*COS(3.0*CHI)
     2          + A2P*SIN(2.0*CHI) + A4*COS(4.0*CHI) )
        RATIO = ( A01+A1*COS(CHI) + A21*COS(2.0*CHI) + A3*COS(3.0*CHI)
     1            + A41*COS(4.0*CHI) + A2P*SIN(2.0*CHI) )
     2          /( A00 + A20*COS(2.0*CHI) + A40*COS(4.0*CHI) )
        IF (SIG0.GT.0.0) THEN
            SIGDB = 10.0*ALOG10(SIG0)
          ELSE
            SIGDB = -99.0
         ENDIF
        CA0 = COEF*A0
        IF (CA0.GT.0.0) THEN
            CA0DB = 10.0*ALOG10(CA0)
          ELSE
            CA0DB = -99.0
         ENDIF
C
C***********************************************************************
      IRET = 0
  900 CONTINUE
      RETURN
      END
```

```
NRCS  Local Symbols

Name                      Class   Type              Size   Offset

CA0DB . . . . . . . . . . param                              0006
SIGDB . . . . . . . . . . param                              000a
SIG0. . . . . . . . . . . param                              000e
IRET. . . . . . . . . . . param                              0012
WIND. . . . . . . . . . . param                              0016
SLOPE . . . . . . . . . . param                              001a
CHI . . . . . . . . . . . param                              001e
PHI . . . . . . . . . . . param                              0022
POL . . . . . . . . . . . param                              0026
WNM . . . . . . . . . . . param                              002a
A2P . . . . . . . . . . . local   REAL*4               4     0000
R1. . . . . . . . . . . . local   REAL*4               4     0004
C . . . . . . . . . . . . local   REAL*4               4     0008
R2. . . . . . . . . . . . local   REAL*4               4     000c
T1. . . . . . . . . . . . local   REAL*4               4     0010
T2. . . . . . . . . . . . local   REAL*4               4     0014
R4. . . . . . . . . . . . local   REAL*4               4     0018
```

```
NRCS  Local Symbols

Name                             Class   Type            Size   Offset

T3. . . . . . . . . . . . . . . local   REAL*4             4    001c
G . . . . . . . . . . . . . . . local   REAL*4             4    0020
CD. . . . . . . . . . . . . . . local   REAL*4             4    0024
R6. . . . . . . . . . . . . . . local   REAL*4             4    0028
CG. . . . . . . . . . . . . . . local   REAL*4             4    002c
BETO. . . . . . . . . . . . . . local   REAL*4             4    0030
CGC . . . . . . . . . . . . . . local   REAL*4             4    0034
PMO . . . . . . . . . . . . . . local   REAL*4             4    0038
USTAR . . . . . . . . . . . . . local   REAL*4             4    003c
GH. . . . . . . . . . . . . . . local   REAL*4             4    0040
R . . . . . . . . . . . . . . . local   REAL*4             4    0044
LWSLP . . . . . . . . . . . . . local   REAL*4             4    0048
PM2H. . . . . . . . . . . . . . local   REAL*4             4    004c
AFR . . . . . . . . . . . . . . local   REAL*4             4    0050
COEF. . . . . . . . . . . . . . local   REAL*4             4    0054
DCGDK . . . . . . . . . . . . . local   REAL*4             4    0058
GV. . . . . . . . . . . . . . . local   REAL*4             4    005c
PM2P. . . . . . . . . . . . . . local   REAL*4             4    0060
A00 . . . . . . . . . . . . . . local   REAL*4             4    0064
A01 . . . . . . . . . . . . . . local   REAL*4             4    0068
A20 . . . . . . . . . . . . . . local   REAL*4             4    006c
A21 . . . . . . . . . . . . . . local   REAL*4             4    0070
A40 . . . . . . . . . . . . . . local   REAL*4             4    0074
PMH . . . . . . . . . . . . . . local   REAL*4             4    0078
PM2V. . . . . . . . . . . . . . local   REAL*4             4    007c
A41 . . . . . . . . . . . . . . local   REAL*4             4    0080
GTK . . . . . . . . . . . . . . local   REAL*4             4    0084
CCOS. . . . . . . . . . . . . . local   REAL*4             4    0088
CCOT. . . . . . . . . . . . . . local   REAL*4             4    008c
UW. . . . . . . . . . . . . . . local   REAL*4             4    0090
PMP . . . . . . . . . . . . . . local   REAL*4             4    0094
A0. . . . . . . . . . . . . . . local   REAL*4             4    0098
A1. . . . . . . . . . . . . . . local   REAL*4             4    009c
BETW. . . . . . . . . . . . . . local   REAL*4             4    00a0
A2. . . . . . . . . . . . . . . local   REAL*4             4    00a4
CO. . . . . . . . . . . . . . . local   REAL*4             4    00a8
PMV . . . . . . . . . . . . . . local   REAL*4             4    00ac
CA0 . . . . . . . . . . . . . . local   REAL*4             4    00b0
A3. . . . . . . . . . . . . . . local   REAL*4             4    00b4
A4. . . . . . . . . . . . . . . local   REAL*4             4    00b8
CWIND . . . . . . . . . . . . . local   REAL*4             4    00bc
THETA . . . . . . . . . . . . . local   REAL*4             4    00c0
TTAN. . . . . . . . . . . . . . local   REAL*4             4    00c4
WNW . . . . . . . . . . . . . . local   REAL*4             4    00c8
SSIN. . . . . . . . . . . . . . local   REAL*4             4    00cc
RATIO . . . . . . . . . . . . . local   REAL*4             4    00d0
```

```
                                                                  PAGE    7
                                                                  03-16-90
                                                                  15:40:31

                              Microsoft FORTRAN Optimizing Compiler Version 4.01

Symbolic Constant             Type                Value

PI. . . . . . . . . . . . . . REAL*4            3.1415927E+000
THPI. . . . . . . . . . . . . REAL*4            4.7123890E+000
TWOPI . . . . . . . . . . . . REAL*4            6.2831855E+000
HPI . . . . . . . . . . . . . REAL*4            1.5707964E+000
A . . . . . . . . . . . . . . REAL*4            5.0000001E-002
AA. . . . . . . . . . . . . . REAL*4            1.0000000E+000
BB. . . . . . . . . . . . . . REAL*4            5.0000001E-002
CC. . . . . . . . . . . . . . REAL*4           -1.0000000E+000
EPS . . . . . . . . . . . . . REAL*4            4.0000001E-001


Global Symbols

Name                          Class   Type              Size   Offset

NRCS. . . . . . . . . . . . . FSUBRT  ***                ***    0000

Code size = 0d73 (3443)
Data size = 00b4 (180)
Bss size  = 00d4 (212)

No errors detected
```

APPENDIX B. TNRCS LISTING

PAGE 1
03-15-90
22:46:12

```
Line#  Source Line          Microsoft FORTRAN Optimizing Compiler Version 4.01

      PROGRAM TNRCS
C
C  ARITHMETIC CONSTANTS:
      REAL PI,HALFPI,THPI,TWOPI
      PARAMETER ( PI = 3.1415927, HPI = PI/2.0,
     1            THPI = 3.0*HPI, TWOPI = 2.0*PI )
C
      REAL WNM,ANGINC,ANGAZ,SLOPE,WIND
      INTEGER POL
C
      WRITE(6,6000)
 6000 FORMAT(1X,'* * * N R C S  Test Program * * *',/)
C
    1 CONTINUE
      PRINT *,'ENTER THE MICROWAVE NUMBER IN CM-1:'
      READ *, WNM
C
    5 CONTINUE
      PRINT *,'ENTER THE POLARIZATION (V=0,H=1):'
      READ *, POL
      IF (POL.NE.0.AND.POL.NE.1) GO TO 5
C
      PRINT *,'ENTER THE INCIDENCE ANGLE (DEG):'
      READ*, ANGINC
C
      PRINT *,'ENTER THE WIND SPEED (M/S), AZIMUTH ANGLE (DEG):'
      READ *, WIND, ANGAZ
C
      PRINT *,'Enter Mean-Square Long-Wave Slope (or -1 for default):'
      READ *, SLOPE
C
      PHI = PI*ANGINC/180.0
      CHI = PI*ANGAZ/180.0
      CALL NRCS(WNM,POL,PHI,CHI,SLOPE,WIND,
     1          IRET,SIGO,SIGDB,CAODB)
C
      WRITE (*,6001) IRET, SIGO, SIGDB, CAODB
6001  FORMAT(1X, 'IRET =', I3,'  SIGO =',  1P, G12.5, 0P,
     1                        '  SIGDB =', F7.3,
     2                        '  CAODB =', F7.3,/)
  900   CONTINUE
       PRINT *,' RUN MODEL AGAIN? (TYPE 0 TO REPEAT, 1 TO STOP) '
         READ *,NN
          IF (NN.EQ.0) GO TO 1
      END


main  Local Symbols

Name                         Class   Type              Size   Offset

SLOPE . . . . . . . . . . . local   REAL*4               4    0002
SIGO. . . . . . . . . . . . local   REAL*4               4    0006
```

```
                                                                PAGE   2
                                                                03-15-90
                                                                22:46:12
                                Microsoft FORTRAN Optimizing Compiler Version 4.01

      main  Local Symbols

      Name                          Class   Type              Size   Offset

      CHI . . . . . . . . . . . . . local   REAL*4               4     000a
      NN. . . . . . . . . . . . . . local   INTEGER*4            4     000e
      PHI . . . . . . . . . . . . . local   REAL*4               4     0012
      SIGDB . . . . . . . . . . . . local   REAL*4               4     0016
      POL . . . . . . . . . . . . . local   INTEGER*4            4     001a
      ANGINC. . . . . . . . . . . . local   REAL*4               4     001e
      ANGAZ . . . . . . . . . . . . local   REAL*4               4     0022
      WIND. . . . . . . . . . . . . local   REAL*4               4     0026
      WNM . . . . . . . . . . . . . local   REAL*4               4     002a
      IRET. . . . . . . . . . . . . local   INTEGER*4            4     002e
      CAODB . . . . . . . . . . . . local   REAL*4               4     0032

      Symbolic Constant                Type                 Value

      PI. . . . . . . . . . . . . . REAL*4              3.1415927E+000
      THPI. . . . . . . . . . . . . REAL*4              4.7123890E+000
      TWOPI . . . . . . . . . . . . REAL*4              6.2831855E+000
      HPI . . . . . . . . . . . . . REAL*4              1.5707964E+000


      Global Symbols

      Name                          Class   Type              Size   Offset

      NRCS. . . . . . . . . . . . . extern  ***               ***     ***
      main. . . . . . . . . . . . . FSUBRT  ***               ***     0000

      Code size = 0165 (357)
      Data size = 0179 (377)
      Bss size  = 0036 (54)

      No errors detected
```

```
      PROGRAM DNRCS
C                                                      VERS 16 MAR 1990
C
C  ARITHMETIC CONSTANTS:
      REAL PI,HALFPI,THPI,TWOPI
      PARAMETER ( PI = 3.1415927, HPI = PI/2.0,
     1            THPI = 3.0*HPI, TWOPI = 2.0*PI )
C
      REAL WNM,ANGINC,ANGLK,ANGAZ,SLOPE,RDAREA,TGTSIG
      INTEGER POL
C
C Z CONTAINS THE VALUES TO BE PLOTTED.
C
C
      REAL Z(21,25), UWIND(21,25), VWIND(21,25), DZ(21,25), SNR(21,25)
C
C Define some default colors to use
C
      REAL COLS
      INTEGER NCOLI, COLA, NPIC, SLOTS, COLI
      REAL RED(6), BLUE(6), GREEN(6)
      DATA RED/1.,1.,0.,0.,1.,0./
      DATA GREEN/1.,0.,0.,1.,1.,1./
      DATA BLUE/1.,0.,1.,0.,0.,1./
C
C SPECIFY COORDINATES FOR PLOT TITLES.  ON AN ABSTRACT GRID WHERE
C THE INTEGER COORDINATES RANGE FROM 0.0 TO 1.0, THE VALUES TX AND TY
C DEFINE THE CENTER OF THE TITLE STRING.
C
      DATA TX/.3955/, TY/.9765/
C
    1 CONTINUE
      PRINT *,'Enter the Microwave Number in cm-1:'
      READ *, WNM
C
    5 CONTINUE
      PRINT *,'Enter the Polarization (V=0,H=1):'
      READ *, POL
      IF (POL.NE.0.AND.POL.NE.1) GO TO 5
C
      PRINT *,'Enter the Incidence Angle (deg):'
      READ*, ANGINC
C
      PRINT *,'Enter Look Direction (deg):'
      READ *, ANGLK
c
c  open GKS and active the display
c
      call xgbeg
      call xtcopt('OPNWKS', 1.0)
C
C
C  Determine the number of color slots we can fill (max we
```

```
C  want is 5).  Note that some displays do not have loadable
C  color tables and the XTCCOL will have no effect.
C
       CALL XTCOPT('COLAVL',COLS)
       NCOLI = COLS
       IF (NCOLI.EQ.0) THEN
C  Monochrome so dont set any colors
      SLOTS = 0
      ELSE
      SLOTS = MIN0(NCOLI-1,6)
          CALL XTCCOL(1, SLOTS, RED, GREEN, BLUE)
      END IF
C
C  Set the text color to index 1.
C
      CALL XTCOPT('TCOLOR',1.0)
C
C  Set the size of labels
C
      CALL CTROPT('ISIZEL',2.0)
      CALL CTROPT('ISIZEM',3.0)
C
C FILL TWO DIMENSIONAL ARRAY TO BE PLOTTED
C
C  GET WIND FIELDS:
      OPEN (UNIT=7,FILE='UVWIND.DAT',STATUS='OLD')
      CALL WINDIN(UWIND)
      CALL WINDIN(VWIND)
      CLOSE(UNIT=7,STATUS='KEEP')
C
C  USE DEFAULT COMPUTATION OF LONG-WAVE SLOPE:
      SLOPE = -1.0
C
      PHI = PI*ANGINC/180.0
      DO 20 I=1,21
        DO 10 J=1,25
C
C         COMPUTE WIND MAGNITUDE (METERS/SEC) AND DIRECTION (DEG):
          WIND  = SQRT( UWIND(I,J)**2 + VWIND(I,J)**2 )/100.0
          ANGWD = 180.0*ATAN2( UWIND(I,J) , VWIND(I,J) )/PI
C
C         ANGAZ DEPENDS ON LOOK ANGLE AND WIND DIRECTION
          ANGAZ = ANGLK - 180.0 - ANGWD
C
C         MAYBE MUST ADD 360 DEGREES TWICE TO GET 0 < ANGAZ < 360):
          IF (ANGAZ.LT.0.0) ANGAZ = 360.0 + ANGAZ
          IF (ANGAZ.LT.0.0) ANGAZ = 360.0 + ANGAZ
C
C         CONVERT TO RADIANS:
          CHI = PI*ANGAZ/180.0
C
C         GET NRCS AT GRID POINT (I,J):
          CALL NRCS(WNM,POL,PHI,CHI,SLOPE,WIND,
```

```
     1              IRET,SIGO,SIGDB,CAODB)
          Z(I,J)  = SIGO
          DZ(I,J) = SIGDB
C
   10     CONTINUE
   20   CONTINUE
C
C
C  Set text to color index 1
C
      CALL XTCOPT('TCOLOR',1.0)
C
C   set up the positive, negative and text colors
C
      CALL CTROPT('PCOLOR',2.0)
      CALL CTROPT('NCOLOR',3.0)
      CALL CTROPT('TCOLOR',1.0)
C
C  Set the permiter color
C
      CALL GRIOPT('PCOLOR',5.0)
C
C ENTRY EZCNTR REQUIRES ONLY THE ARRAY NAME AND ITS DIMENSIONS
C
C THE TITLE FOR THIS PLOT IS
C
C  DEMONSTRATION PLOT FOR EZCNTR ENTRY OF CONREC
C
      CALL WTSTR ( TX, TY,
     1            '  Normalized Radar Cross Section (absolute)  ',2,0,0 )
      CALL EZCNTR (Z,21,25)
C
      CALL WTSTR ( TX, TY,
     1            '     Normalized Radar Cross Section (dB)     ',2,0,0 )
      CALL EZCNTR (DZ,21,25)
C
C
C ENTRY CONREC ALLOWS USER SPECIFICATION OF PLOT PARAMETERS, IF DESIRED
C
C IN THIS EXAMPLE, THE LOWEST CONTOUR LEVEL (-4.5), THE HIGHEST CONTOUR
C LEVEL (4.5), AND THE INCREMENT BETWEEN CONTOUR LEVELS (0.3) ARE
C SPECIFIED.
C
C THE TITLE FOR THIS PLOT IS
C
C  DEMONSTRATION PLOT FOR CONREC ENTRY OF CONREC
C
C
      WRITE(*,6101)
 6101 FORMAT(1X,'Enter Area of Ocean Surface Corresponding to ',
     1 'Radar Resolution (m**2): ')
      READ*, RDAREA
C
```

```
                                                                   PAGE   4
                                                                   03-16-90
                                                                   15:39:29

 Line#  Source Line          Microsoft FORTRAN Optimizing Compiler Version 4.01

   160        WRITE(*,6102)
   161   6102 FORMAT(1X,'Enter Radar Cross-Section of Target (m**2): ')
   162        READ*, TGTSIG
   163  C
   164        DO 80, I=1,21
   165          DO 81 J=1,25
   166            ASNR = TGTSIG/( RDAREA*Z(I,J) )
   167            IF (ASNR.GT.0.0) THEN
   168                SNR(I,J) = 10.0*ALOG10(ASNR)
   169              ELSE
   170                SNR(I,J) = 49.9
   171                WRITE(*,6902) I, J, Z(I,J), ASNR
   172             ENDIF
   173     81      CONTINUE
   174     80    CONTINUE
   175   6902 FORMAT(1X, 'Z(',I2,',',I2,') =', 1P, E14.4,
   176       1        '   ASNR =', E14.4, 0P)
   177  C
   178        PRINT *, ' -----> Press "ENTER" for SNR display'
   179        CALL FRAME
   180        CALL WTSTR ( TX ,TY,
   181       1            '  Target SNR with Respect to Radar Clutter   ',2,0,0 )
   182  C-    CALL CONREC (Z,21,21,25,-4.5,4.5,.3,0,-1,0)
   183        CALL CONREC (SNR,21,21,25,-50.0,+50.0,2.0,0,-1,0)
   184        CALL FRAME
   185  C
   186        call xgend
   187        stop
   188        END


main  Local Symbols

Name                          Class    Type              Size    Offset

SLOPE . . . . . . . . . . . . local    REAL*4               4     0002
UWIND . . . . . . . . . . . . local    REAL*4            2100     0006
RED . . . . . . . . . . . . . local    REAL*4              24     0172
GREEN . . . . . . . . . . . . local    REAL*4              24     018a
BLUE. . . . . . . . . . . . . local    REAL*4              24     01a2
TX. . . . . . . . . . . . . . local    REAL*4               4     01ba
TY. . . . . . . . . . . . . . local    REAL*4               4     01be
VWIND . . . . . . . . . . . . local    REAL*4            2100     083a
I . . . . . . . . . . . . . . local    INTEGER*4            4     106e
J . . . . . . . . . . . . . . local    INTEGER*4            4     1072
TGTSIG. . . . . . . . . . . . local    REAL*4               4     1076
SIGO. . . . . . . . . . . . . local    REAL*4               4     107a
CHI . . . . . . . . . . . . . local    REAL*4               4     107e
SLOTS . . . . . . . . . . . . local    INTEGER*4            4     1082
Z . . . . . . . . . . . . . . local    REAL*4            2100     1086
DZ. . . . . . . . . . . . . . local    REAL*4            2100     18ba
PHI . . . . . . . . . . . . . local    REAL*4               4     20ee
SIGDB . . . . . . . . . . . . local    REAL*4               4     20f2
```

```
main  Local Symbols

Name                          Class   Type              Size    Offset

POL . . . . . . . . . . . . . local   INTEGER*4            4      20f6
ANGLK . . . . . . . . . . . . local   REAL*4               4      20fa
RDAREA. . . . . . . . . . . . local   REAL*4               4      20fe
ANGINC. . . . . . . . . . . . local   REAL*4               4      2102
ANGWD . . . . . . . . . . . . local   REAL*4               4      2106
COLS. . . . . . . . . . . . . local   REAL*4               4      210a
ANGAZ . . . . . . . . . . . . local   REAL*4               4      210e
WIND. . . . . . . . . . . . . local   REAL*4               4      2112
WNM . . . . . . . . . . . . . local   REAL*4               4      2116
SNR . . . . . . . . . . . . . local   REAL*4            2100      211a
ASNR. . . . . . . . . . . . . local   REAL*4               4      294e
IRET. . . . . . . . . . . . . local   INTEGER*4            4      2952
NCOLI . . . . . . . . . . . . local   INTEGER*4            4      2956
CAODB . . . . . . . . . . . . local   REAL*4               4      295a


Symbolic Constant               Type                Value

PI. . . . . . . . . . . . . . REAL*4            3.1415927E+000
THPI. . . . . . . . . . . . . REAL*4            4.7123890E+000
TWOPI . . . . . . . . . . . . REAL*4            6.2831855E+000
HPI . . . . . . . . . . . . . REAL*4            1.5707964E+000
```

```
      SUBROUTINE WINDIN(WIND)
      REAL WIND(21,25)
      CHARACTER*80 CARD
      CHARACTER*12 BFLD
      CHARACTER*4  RFLD
      CHARACTER*1  C(80)
      EQUIVALENCE(C,CARD)
      EQUIVALENCE(C( 1),BFLD)
      EQUIVALENCE(C(14),RFLD)
C
    1 CONTINUE
      READ(7,7001) CARD
      IF (BFLD.NE.'FIELD TITLE:') GOTO 1
C
      N = 1
    5 CONTINUE
      READ(7,7001) CARD
      IF ( (RFLD.NE.'OW  ').AND.(RFLD.NE.'OE  ') ) THEN
          GOTO 5
        ELSE
          DO 7 I=1,21
            IF       (N.EQ.1) THEN
                READ(7,7005) (WIND(I,J),J= 1, 7)
              ELSEIF (N.EQ.2) THEN
```

```
                                                                    PAGE   6
                                                                    03-16-90
                                                                    15:39:29

 Line#  Source Line           Microsoft FORTRAN Optimizing Compiler Version 4.01

   213                    READ(7,7005) (WIND(I,J),J= 8,14)
   214                  ELSEIF (N.EQ.3) THEN
   215                    READ(7,7005) (WIND(I,J),J=15,21)
   216                  ELSEIF (N.EQ.4) THEN
   217                    READ(7,7005) (WIND(I,J),J=22,25)
   218                 ENDIF
   219      7         CONTINUE
   220              N = N + 1
   221              IF (N.LE.4) GOTO 5
   222           ENDIF
   223          RETURN
   224     7001 FORMAT(A80)
   225     7005 FORMAT(6X, 7F9.2)
   226          END


WINDIN  Local Symbols

Name                          Class   Type                Size   Offset

WIND. . . . . . . . . . . . . param                                0006
I . . . . . . . . . . . . . . local   INTEGER*4             4      295e
J . . . . . . . . . . . . . . local   INTEGER*4             4      2962
N . . . . . . . . . . . . . . local   INTEGER*4             4      2966
CARD. . . . . . . . . . . . . local   CHAR*80              80      296a
BFLD. . . . . . . . . . . . . local   CHAR*12              12      296a
RFLD. . . . . . . . . . . . . local   CHAR*4                4      2977
C . . . . . . . . . . . . . . local   CHAR*1               80      296a


Global Symbols

Name                          Class   Type                Size   Offset

CONREC. . . . . . . . . . . . extern  ***                 ***      ***
CTROPT. . . . . . . . . . . . extern  ***                 ***      ***
EZCNTR. . . . . . . . . . . . extern  ***                 ***      ***
FRAME . . . . . . . . . . . . extern  ***                 ***      ***
GRIOPT. . . . . . . . . . . . extern  ***                 ***      ***
NRCS. . . . . . . . . . . . . extern  ***                 ***      ***
WINDIN. . . . . . . . . . . . FSUBRT  ***                 ***     0736
WTSTR . . . . . . . . . . . . extern  ***                 ***      ***
XGBEG . . . . . . . . . . . . extern  ***                 ***      ***
XGEND . . . . . . . . . . . . extern  ***                 ***      ***
XTCCOL. . . . . . . . . . . . extern  ***                 ***      ***
XTCOPT. . . . . . . . . . . . extern  ***                 ***      ***
main. . . . . . . . . . . . . FSUBRT  ***                 ***     0000

Code size = 097a (2426)
Data size = 02d7 (727)
Bss size  = 29ba (10682)

No errors detected
```

APPENDIX D. Environmental_Sea_Clutter PACKAGE SPECIFICATION

```
package Environmental_Sea_Clutter is

procedure NRCS(in Envir.Surface_Wind ,
                  Envir.Swell        ,
                  Envir.Precipitation,
                  Envir.Salinity     ,
                  Envir.Air_Temp     ,
                  Envir.Sea_Temp     ,
                  Radar.Frequency    ,
                  Radar.Polarization ,
                  Radar.Incidence_Angle;
              out NRCS                 );
-- Yields Normalized Radar Cross Section (NRCS) of sea surface.

procedure ERCS(in Radar.Beam_Specification,
                  NRCS                    ;
              out ERCS                     )
-- Yields radar cross section of environmental clutter for a given
--   radar.  The input information in Radar.Beam_Specification can be
--   expressed in different ways, but must be sufficient to determine a
--   "footprint" over which NRCS should be integrated (e.g. pulse
--   duration beamwidth, and radar altitude).

procedure TSNR(in Target.RCS,
                  ERCS      ;
              out TSNR       );
-- Yields signal-to-noise ratio of a given target with respect to
--   sea clutter, for the radar in question.

end Environmental_Sea_Clutter
```

DISTRIBUTION LIST

NOARL (10)
ATTN: CODE 125L
JCSSC, MS 39529-5004

NAVAL RESEARCH LABORATORY
ATTN: RADAR DIV.
WASHINGTON, DC 20375

OFFICE OF NAVAL RESEARCH
ATTN: CODE 12
800 N. QUINCY ST.
ARLINGTON, VA 22217-5000

COMMANDING OFFICER
NAVAL AIR FACILITY
ATTN: NR NORA 0966
WASHINGTON, DC 20390

OFFICE OF NAVAL RESEARCH
ATTN: CODE 1122 PO, PHYS. OCE
ARLINGTON, VA 22217-5000

FLEET INTELLIGENCE CENTER
ATTN: OPERATIONS OFFICER
(PACIFIC)
PEARL HARBOR, HI 96860

NAVAL POSTGRADUATE SCHOOL
ATTN: LIBRARY, CODE 0142
MONTEREY, CA 93943-5002

CHIEF OF NAVAL OPERATIONS
ATTN: OP-096, OP-0961B
U.S. NAVAL OBSERVATORY
WASHINGTON, DC 20392-1800

NOARL
ATTN: CODE 125P
JCSSC, MS 39529-5004

NAVOCEANCOMFAC
ATTN: OPERATIONS OFFICER
JCSSC, MS 39529

PENNSYLVANIA STATE UNIV. (4)
ATTN: D. MERDES, APPL. RSCH.
P.O. BOX 30
STATE COLLEGE, PA 16801

OFFICE OF NAVAL TECHNOLOGY
ATTN: CODE 22
800 N. QUINCY ST.
ARLINGTON, VA 22217-5000

U.S. NAVAL ACADEMY
ATTN: CHAIRMAN OCEANOG. DEPT.
ANNAPOLIS, MD 21402

SPAWARSYSCOM
ATTN: CODE PMW-141
NAT. CTR. #1
WASHINGTON, DC 20363-5100

NAVAL RESEARCH LABORATORY
ATTN: SPACE SCIENCE DIV.
WASHINGTON, DC 20375

OFFICE OF NAVAL TECHNOLOGY
ATTN: DR. M. BRISCOE, CODE 228
800 N. QUINCY ST.
ARLINGTON, VA 22217-5000

OFFICE CHIEF OF NAVAL RESEARCH
ATTN: CODE 01RM
NAVAL RESERVE TECH MOBIL. PROG
800 N. QUINCY ST.
ARLINGTON, VA 22217

OFFICE OF NAVAL RESEARCH
ATTN: HEAD, OCEAN SCIENCES DIV
CODE 1122
ARLINGTON, VA 22217-5000

FLEET INTELLIGENCE CENTER
ATTN: OPERATIONS OFFICER
(EUROPE & ATLANTIC)
NORFOLK, VI 23522

NAVAL POSTGRADUATE SCHOOL
ATTN: CODE OC (OCEANOGRAPHY)
MONTEREY, CA 93943-5000

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. Agency Use Only *(Leave blank).*

2. Report Date. July 1990

3. Report Type and Dates Covered. Final

4. Title and Subtitle.
Incorporation of Radar Sea Clutter Prediction into
Operational Navy Environmental Support Products:
Prototype Software Development

5. Funding Numbers.
Program Element No. 63704N
Project No. X1596
Task No. 0500
Accession No. DN650752

6. Author(s).
D.W. Merdes

7. Performing Organization Name(s) and Address(es).
Naval Oceanographic and Atmospheric Research Laboratory
Atmospheric Directorate
Monterey, CA 93943-5006

8. Performing Organization Report Number.
NOARL Technical Note 34

9. Sponsoring/Monitoring Agency Name(s) and Address(es).
Space and Naval Warfare Systems Command (PMW-141)
Washington, DC 20363-5000

10. Sponsoring/Monitoring Agency Report Number.
NOARL Technical Note 34

11. Supplementary Notes. CDR D.W. Merdes, USNR-R, a Naval Reserve officer then serving as Commanding Officer of NR NORA 1091, NAS South Weymouth, MA, performed this study during periods of annual training at NOARL in 1989-90. He is a Research Associate at The Pennsylvania State University Applied Research Laboratory.

12a. Distribution/Availability Statement.
Approved for public release; distribution is unlimited.

12b. Distribution Code.

13. Abstract *(Maximum 200 words).*

Prototype operational FORTRAN software is developed implementing a two-scale microwave sea-surface scatterometry model. A program suitable for specific numerical testing, and another program illustrating its potential operational utility in generating graphical visual aids, are also documented. Limitations of the selected scatterometry model are discussed, and suggestions on the direction of future development efforts are offered.

14. Subject Terms.
Radar
Sea clutter

15. Number of Pages. 3030

16. Price Code.

17. Security Classification of Report. UNCLASSIFIED

18. Security Classification of This Page. UNCLASSIFIED

19. Security Classification of Abstract. UNCLASSIFIED

20. Limitation of Abstract. Same as report

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102