FINAL REPORT

MISSILE GEOMETRY PACKAGE

SBIR CONTRACT NUMBER DAAH01-88-C-0942

MARCH 31, 1989

Performed by:
Concept Analysis Corporation
14789 Keel Street
Plymouth, MI 48170

DISTRIBUTION STATEMENT A
Approved for public release: Distribution Unlimited

14789 keel street plymuth, mi 48170 phone 313/455 2340

92 7 29 020
Concept Analysis Corporation (CAC) markets a
general-purpose CAD/CAM system, CADCEPT. CAC has tailored
custom CADCEPT features for Redstone, using the CADCEPT
Macro language, which is a powerful tool for geometric
modelling using parametric design techniques. This system
was used to build a geometric model of a missile.

Task 1 was an initial problem definition task.
Conversations with Mr. Ed Vaughn at Redstone resulted in the
submission of a sketch of a typical (unclassified) missile
and specifications for input format to Redstone's SPIRITS
analysis program. The goal was defined to model this
missile in CADCEPT, and to demonstrate that sections could
be cut and fed into SPIRITS in the proper format. It was
discovered that the primary reason for interest in CADCEPT
was its ability to do faceted geometric modelling, and that
another system of interest, BRL CAD, from the Ballistics
Research Laboratories, was of interest to Redstone
personnel. It was thought at the time that BRL CAD might be
superior to CADCEPT for the purpose of preparing input to
SPIRITS. CAC has since acquired and implemented a copy of
BRL CAD, and will make a case that CADCEPT will be a nice
complement for BRL CAD, especially because it is more
interactive than BRL CAD, it has a drafting capability,
interfaces to machining postprocessors, and a powerful macro
capability. A macro is somewhat like computer software, but
is really the product of an end user with an application,
written in the CADCEPT command language. The medium of
representation is text, and the instructions are interpreted
(not compiled) by the system, inline with other user
instructions. The main difference between macro writing and
other steps in the operation of CADCEPT is that macros can
be saved, and parameters can be varied easily at later
dates. In addition, macros have control structures similar
to structured programming languages. Listings of macros for
modelling the sample missile and for other purposes are
given in Figures 1, 7, 9, 11, 13, 15, 18, 22 and 26.

Task 2 was the preparation of demonstration macros,
which illustrate the capability to handle the geometric
shapes and design problems identified, and to simulate new
software required. Information necessary to commence Task 2
was received on October 31, 1988. It was decided to write
macros for the following components:
- Hemispherical nose
- Cylinder
- Transition piece from cylinder to rectangular
  parallelepiped
- Rectangular parallelepiped
- Group of airfoil stabilizers, modelled as symmetric
  Joukowski profiles in cross-section.
- Holes in the transition piece.
It has turned out that because of a bug in the UNIX version of CADCEPT which we were subsequently forced to use because of the failure of CAC’s VAX computer, and consequently the VAX version, we have not been able to demonstrate the volume subtraction necessary to pierce the holes in the transition piece.

Task two has been further elaborated to output data from CADCEPT into Redstone’s engineering analysis program SPIRITS, and information on this requirement was also furnished on October 31. A program has been written implemented to “thin” section data, in order to prepare it for input to SPIRITS. A conversation with Mr. Vaughn about the subject gave information that sometimes too much data is produced by ray tracing and other programs. CAC’s thinner will preserve the character of sections, while discarding insignificant data. A listing for the FORTRAN source will be found in Figure 24, sample input file in figure 23, and sample output file in figure 25. Figure 22 shows that CADCEPT is capable of interfacing with SPIRITS in the sense that it can produce a file of points in the format given us by Mr. Vaughn’s October 31 submission. Figure 26 shows further the reverse process, that of feeding CADCEPT with points produced by other programs. We might add that this capability is extra to the IGES capability which CADCEPT also has, in case data is produced not by just any arbitrary program, but by a CAD system which also has IGES capability.

Results of Task 2 are displayed in Figures 1-26. It can be seen from perusal of the listings that shapes can be changed considerably by the modification of a few parameters, thus illustrating parametric design, and modelling without drawings. The CADCEPT database is more than just a few interrelated files—many components are connected in logical ways. Macros allow cross checking between dimensions, so that when one dimension is changed, others can be made to change automatically.

Task 3 was the determination of software-hardware requirements. The extent of CAC’s involvement in configuring a turnkey system and being in a consultative relationship was to be determined. We have determined that the target configuration is any VAX computer running ULTRIX, DEC’s version of UNIX, with a Tektronix 4207-compatible terminal. Most recent information from Redstone is that the 4207’s have been replaced with more advanced Tektronix terminals. We have verified that these are upward compatible with the Tektronix 4107, a slightly less powerful version of the 4207. At the time of this writing, we have
not implemented the 4107 support code for UNIX. CAC will do this implementation soon on CAC's UNIX machine, for CAC's own benefit. The use of the Tektronix terminals is straightforward and needs no special support, since they will operate in "4014 emulation" mode. The major benefit of native code support for the 4107-4207 machines as far as modeling capability of CADCEPT is concerned is that this enables a graphics tablet to be used. We understand that Redstone does not use graphics tablets. Much of the CAD world seems to be moving in the direction of iconic menus, which require only the terminal. We plan to implement icons sometime this year.

The verification of whether CADCEPT will run on the VAX under ULTRIX is being performed at the moment of this writing, and we believe that any problems will be minor, since as we understand it, DEC ULTRIX FORTRAN is more compatible with VAX-VMS than f77 FORTRAN which we are currently using. This is because VMS and ULTRIX are manufactured by the same company, Digital Equipment Corporation. The UNIX version is more portable than the VMS version, because we have eliminated the system's dependency on byte ordering. The major VAX-VMS-related system dependencies in the port to UNIX were changing the VMS "Q10" functions to UNIX system calls, which required interfacing FORTRAN routines to UNIX C-functions. These were written by the Phase I Principal Investigator prior to submission of the initial proposal for this SBIR contract.

Also as part of Task 3, we have verified by actual inspection of the code and implementation, as well as by bringing it up on CAC's in-house UNIX computer, that another CAD package of interest to Redstone, BRL CAD, is sufficiently portable and well engineered to run on Redstone's equipment. We have included the knowledge thus acquired in our Phase II proposal.

Task 4 was the preparation of system recommendations. This amounts to our Phase II report. In summary, our system recommendations are:

1. BRL CAD, CADCEPT, and Redstone's analysis programs, such as SPIRITS, should be interfaced together in a tightly coupled fashion.

2. Add new software to integrate the kinds of geometric entities between the packages, and to customize them further for missile geometry.
(3) Provide training and consulting services in the use of the new system.

(4) Add new three-dimensional display devices.

(5) Implement an iconic menusystem throughout.

(6) Interface to the Eagle system, a large integrated system in use at Eglin Air Force Base for Computational Fluid Dynamics. CAC has acquired a copy of this and is developing expertise in its use. CAC personnel, Dr. Glance in particular, have experience in Computational Fluid Mechanics. This, combined with a general knowledge of and commercial practice in, continuum mechanics, show that CAC is well qualified to implement the use of Eagle in connection with all the rest of the software mentioned above.

(7) Interface with CAC's structural mechanics programs, in an intelligent and specialized manner.

Task 5, presentation of results and response to review, has been done remotely, by the installation of CADCEPT on Redstone's equipment and by this Final Report and the Phase II proposal.
The following is a list of figures documenting the Phase I example problem results. “nascad” means "/nexus/nascad/nascad, my". In the following instructions, commas and periods after CADCEPT commands are for these instructions only - not for CADCEPT.

Figure 1: Listing of Macro MISSILE. This is the main macro which invokes other macros and assembles the missile. Found in nascad/repairs/missile.com . cat it.

Figure 2: Isometric view of the result of running MISSILE. Found in nascad/MISSILE.XYZ. LOAD 'MISSILE.XYZ' and SHOW 'ISO'.

Figure 3: Zoomed top view of the result of running MISSILE, showing the Joukowski profile of the fins. While data for Figure 2 is LOADed, do SHOW 'T', FIT, SHOW, ZOOM 20 20 20 -10 -10 -1, SHOW

Figure 4: Top view of the result of running MISSILE. Figure 3, FIT, SHOW

Figure 5: Right view of the result of running MISSILE. Figure 4, SHOW 'R', FIT, SHOW

Figure 6: Front view of the result of running MISSILE. Figure 5, SHOW 'F', FIT, SHOW

Figure 7: Listing of macro HEMI, used to product the nose piece. Found in nascad/hemi.com . cat it.

Figure 8: Isometric of the result of running HEMI. Found in nascad/HEMI.XYZ . LOAD HEMI.XYZ . LOAD HEMI.XYZ, SHOW 'ISO', FIT, SHOW

Figure 9: Listing of macro CYL, used to produce the cylindrical part. Found in nascad/cyl.cc . cat it.

Figure 10: Isometric of the result of running CYL. Found in nascad/CYL.XYZ . LOAD 'CYL.XYZ', SHOW 'ISO', FIT, SHOW

Figure 11: Listing of macro TRANPIECE, used to produce the transition piece. Found in nascad/tranpiece.com. cat it.

Figure 12: Isometric of the result of running TRANPIECE. Found in nascad/TRANPIECE.XYZ . LOAD 'TRANPIECE.XYZ', SHOW 'ISO', FIT, SHOW.
Figure 13: Listing of macro BOX, used to produce the base component. Found in nascad/box.com. cat it.

Figure 14: Isometric of the result of running BOX. Found in nascad/BOX.XYZ. LOAD 'BOX.XYZ', SHOW 'ISO', FIT, SHOW.

Figure 15: Listing of macro FINS, used to produce the set of eight fins. Found in nascad/repairs/fins.com and nascad/repairs/jeffins.com. cat it.

Figure 16: Isometric of the result of running FINS & JEFFINS. Not stored as complete object. Do steps in nascad/repairs/missile.com, just the ones for the fins part, except ATTACH 'MAIN' instead of 'MISSILE'.

Figure 17: Mathematical analysis for Joukowski profile.

Figure 18: Listing of macro PROFILE, for Joukowski profile. Found in nascad/profile.com. cat it.

Figure 19: Typical Joukowski profile, result of running PROFILE. Found in nascad/PROFILE.XYZ. LOAD 'PF4PROFILE.XYZ', FIT, SHOW.

Figure 20: Sections of missile, using PIERCE command. Found in nascad/repairs/MISSILE.PRC. LOAD 'MISSILE.PRC', SHOW 'ISO', FIT, SHOW.

Figure 21: Missile showing section planes. Found in /nascad/repairs/MISSILE.CUT. LOAD 'MISSILE.CUT', SHOW 'ISO', FIT, SHOW.

Figure 22: Listing of macro OUTPNTS showing one method of getting points out of CADCEPT into a file in SPIRITS format. Found in nascad/outpnts.com. cat it.

Figure 23: Listing of the file of points gotten by running macro OUTPNTS on the model produced by PROFILE. Found in nascad/PROFILE.PNTS. cat it.

Figure 24: Listing of thinner program. Found in nascad/thin, nascad/thin.f, nascad/thin.o. A standalone program, which prompts for input and output and thin tolerance.

Figure 25: Listing of set of points produced by using thinner program on file shown in Figure 22. Found in nascad/ROPROFILE.PNTS. cat it.

Figure 26: Listing of macro INPNTS, showing how point files can be read into CADCEPT. Found in nascad/inpnts.com. cat it.
Figure 1. Listing of macro MISSILE. This is the main macro which calls other macros to assemble the missile.
Figure 2. Isometric view of the result of running MISSILE.
Figure 3. Zoomed top view of the result of running MISSILE, showing the Joukowski profile of the fins.
Figure 4. Top view of the result of running MISSILE.
Figure 5. Right view of the result of running MISSILE.
Figure 6. Front view of the result of running MISSILE.
WILL HE PRODUCE HEMISPHERE AT CYLTOP. RADIUS CYLRAD

INITIALIZE PARAMETERS

FLAG 21 'ON' ! MAKE RESULTING DRIVE SURFACE VISIBLE.
NUMSEG=#1 ! NUMBER SEGMENTS IN ALL POLYGONS.
NUMS=19 NUMSEG ! NUMBER SEGMENTS FOR ARC DISPLAYS & POLYGONALIZATION.
SYG 2.001
CYLTOP=0.0,0.0
CYLRAD=4.9
DCC

E HEMI
CUM=1
HEMISTEP! CUM ! GET UP PARAMETERS
UNSA
DEFUPW 'HEMI'
ADDS 1
ADDA CYLTOP CYLRAD

! SHOULD NOW HAVE THE "DRIVE" POLY ON LEVEL 0

UNSA
ADDS 1
! PT1 AND PT2 WILL BE ENDS OF QUARTER CIRCLE ARC
PT1=CYLTOP(1)+CYLRAD
PT12=CYLTOP(2)
PT13=CYLTOP(3)
PT21=CYLTOP(1)
PT22=CYLTOP(2)
PT23=CYLTOP(3)-CYLRAD
PT1=PT11,PT12,PT13
PT2=PT21,PT22,PT23
ADDA PT1 CYLTOP PT2
CLEV 5
NLA
ADDS 0
DRIVE 5 0 10 ! MAKE THE HEMI.
VOLS
INWARD
OVAR 'ALL'
STOR 'HEMI,XYZ'
MSG=' NOW EXIT, RUN INWARDPP.com, GET BACK IN, DO LODVOL & SHOW'
! MSG
E
SDIC 'HEMI.DAT'
EXIT
[nascad] 155

Figure 7. Listing of macro HEMI, used to produce the nose piece.
Figure 8. Isometric of the result of running HEMI.
The macro CYL is defined as follows:

```
MACRO
E CYLSTEP1
 ' THIS FILE IS cyl.com
 ' WILL MAKE RIGHT CIRCULAR CYLINDER.
 ' INITIALIZE PARAMETERS
HI=52.  ' Z- COORD OF BOTTOM OF CYLINDER.
HI=50.  ' Z- COORD OF TOP OF CYLINDER.
CYLRAD=4.  ' RADIUS OF CYLINDER.
CYLBOT=0,0,H1
CYLTOP=0,0,H2
ALONG=CYLTOP-CYLBOT
SV=1,0,0,0

C

E CYL
CYLSTEP1
UNGA
DEFPRW CYL'
ADDS 1
ADDA CYLBOT CYLRAD
CLEV 0  ' NOW HAVE LOWER CIRCLE ON LEVEL 0
COPY
MOVE ALONG
CLEV 5  ' NOW HAVE UPPER CIRCLE ON LEVEL 5
SELA
ADDS 0
RULE 0 5 10  ' MAKE THE CYLINDER.
SELS
SELA
VCLS
INWARD
DVAR 'ALL'
STOR 'CYL.XYZ'
 ' MSG=' NOW EXIT, RUN INWARDPP.com, GET BACK IN, DO LODVOL, SHOW.
 ' MSG
E
E
EDIC 'CYL.DAT'
EXIT
(inascad1 156)
```

Figure 9. Listing of macro CYL, used to produce the cylindrical piece.
Figure 10. Isometric of the result of running macro CYL.
cat tranpiece.com

MACRO
E TRNSPIE:
! INITIALIZE PARAMETERS
H1=30. ! HEIGHT OF CIRCLE
H2=50. ! HEIGHT OF SQUARE
CYLRAE=4. ! RADIUS OF CIRCLE
BOXRAD=7. ! HALF-DIAGONAL OF SQUARE
SCTOP=0,0,H1
CYLOBE=0,0,H1
P1=CYLRAE,0,H1
P2=CtroRAD,0,H1
P3=CYLRAE,0,H1
P4=0,.5*BOXRAD,H1
P5=BOXRAD,0,H2
P6=0,0,5*BOXRAD,H2
P7=0,0,.5*BOXRAD,H2
P8=0,0,.5*BOXRAD,H2
NUMSEG=#1 ! NUMBER SEGMENTS IN ALL POLYGONS.
SYS #13 NUMSEG ! NUMBER SEGMS FOR ARC DISPLAYS.
PALONG=P5-P6
PINC=PALONG/NUMSEG ! INCREMENT FOR LINE SEGMS IN POLY.
PBE=0
SYS 2,0001
DOC
! FORMAT (8CA1.2X,15)

E
E TRNSPIE
! TO MAKE MISSILE TRANSITION PIECE
DUM=#1
TRNSPIE DUM
DEFPW 'TRANPIECE'
CHOPUP ! ADDS STRAIGHT LINE POLY FROM P5 TO P6,
! WITH NUMSEG SEGMENTS
SFLA
COPY
ROTA -90. BOXTOP CYLOBE
COPY
ROTA -90. BOXTOP CYLOBE
COPY
ROTA -90. BOXTOP CYLOBE
SFLA
CONNECT P5
DELS ! GET RID OF THE COMPONENTS
SFLA ! WE NOW HAVE A SQUARE-LookING POLYGON
CLEV 0 ! ON LEVEL 0
UNSA
ADDS 1
SYS #13 4*NUMSEG
ADD P1 CYLOBE P1 'E'
CLEV 3
SFLA
ADDS 0
RULE 0 5 10
DELS
SFLA
VOLS
INWARD
! MSG= 'NOW EXIT, RUN INWARDFP.com, GET BACK IN, ADIC TRNSPIECE'
DVAR 'ALL'
STOR 'TRANPIECE.XYZ'
! MSG
! MSG= 'AND DO STEPS'
Figure 11b. Macro TRANPIECE.
Figure 11c. Listing of macro TRANPIECE, concl.
Figure 12. Isometric of the result of running TRANPIECE.
Figure 13. Listing of macro BOX, used to produce the base of the missile.
Figure 14. Isometric of the result of running BOX.
WILL MAKE A MISSILE FIN

INITIALIZE PARAMETERS

A=0  Z-PRIME COORD OF INSIDE OF BIG FINS.
H2=20. Z-PRIME COORD OF OUTSIDE OF BIG FINS.
W=4 WIDTH
T=.6 THICKNESS

SMALLFIN=0.0,10 CENTER OF SMALL FINS
BIGFIN=0,0,30 CENTER OF BIG FINS
MBIGFIN=0,0,-30 MINUS BIGFIN
CRBI=0,0,H1 -PRIME POINT OF INSIDE OF BIG FINS.
CRBI=0,0,H2 -PRIME POINT OF OUTSIDE OF BIG FINS
XOLTP=10,0,0 AXES FOR ROTATION
XOUT2=20,0,0
YOUT1=W/2,0,0 YOUT2=W/2,0,10
NUMSEG=20 NUMBER SEGMENTS IN ALL POLYGONS.
SYS 118 NUMSEG NUMBER SEGMENTS FOR ARC DISPLAYS & POLYGONALIZATION.
SYS 2 .001

DDC

ZVEC= SMALLFIN-BIGFIN
DOWNX=0,0,-20

E

E FINS
FINSTEP1

UNSA

DEFIN 'FINS'
TO MAKE SYMMETRIC JOUKOWSKI PROFILE.

SO=.12245009 ! SORT(3)/9.0 CADCEPT DOESN'T YET HAVE SORT.
C=W/4.0 CIRCLE RADIUS.
EPS=SO*T/C PROFILE THICKNESS
M=EPS*C
A=M+C
THETA= 0.0
PI=180.
TUPI=2.0*PI
THINC=TUPI/NUMSEG
FNJUMP THETA
PSEG=X,Y,H1
THETA=THETA+THINC
FNJUMP THETA
PEND=X,Y,H1
PLITL=.001,.001,.001
ADD PSEG PEND
BOXL=PEND-PLITL
BOXR=PEND+PLITL
SFLY BOXL BOXR
PSEG=PEND
NTIMES=NUMSEG-1
FOR (MW1=1:NTIMES) FOTHER
SELA
ADD 0
CONNECT
CLEV 0 ON LEVEL 0
COPY
MOVE SPAN
CLEV 3 WE NOW HAVE LOWER POLYGON ON LEVEL 3
SELA
ADD 0
RULE 0.5,10 MAKE THE PRISM

Figure 15a. Listing of macro FINS.
Fig 15b. Macro FINS, cont.
Figure 15c. Macro FINs, concl.
Figure 16. Result of running macro FINS.
Figure 17. Mathematical analysis for Joukowski profile.

As \( \theta \) goes from 0 to \( 2\pi \), point \((x,y)\) given as follows will trace out the profile:

\[
x = a \cos \theta - m + \frac{c^2(a \cos \theta - m)}{a^2 + m^2 - 2am \cos \theta}
\]

\[
y = a \sin \theta - \frac{ca^2 \sin \theta}{a^2 + m^2 - 2am \cos \theta}
\]

where, with \( C \) computed from \( \text{tmax} \) in figure, \( \epsilon = \frac{\text{tmax} \sqrt{3}}{2C} \)

\( m = \epsilon C, \) and \( a = m + C = C(1 + \epsilon) \)
MACRO
E PROFILE
' TO MAKE SYMMETRIC JOUKOWSKI PROFILE.
FORMAT (F10.3)
PROMPT 'ENTER WIDTH, F10.3' W
PROMPT 'ENTER THICKNESS, F10.3' T
NUMSEG=40    ! can't make this very much bigger- no room.
! Note: I3 format doesn't work (for prompt for this)
FD=1.9245009    ! SORT(3)/9.0
C=W/4.0    ' CIRCLE RADIUS.
EPS=SD+T/C    ' PROFILE THICKNESS
M=EPS*D
A=M+C
UNSA
THETA=0.0
F1=1.50
THETA=2.0*PI
THINC=THETA/NUMSEG
NUPT THETA
BEG=X,Y,0
THETA=THETA+THINC
NUPT THETA
PEND=X,Y,0
PLITL=.001,.001,.001
ADDL BEG PEND
BOXL=PEND-PLITL
BOXR=PEND+PLITL
SELV BOXL BOXR
PBEG=PEND
NTIMES=NUMSEG-1
FOR (MW1=1; NTIMES) NOTHER
STOR 'PROFILE.XYZ'
DVAR 'ALL'
'SGB= ' ALL DONE'
'SGB
E
E NOTHER
' ADDS ANOTHER SEGMENT TO THE POLYGON
THETA=THETA+THINC
NUPT THETA
ADDV X,Y,0
E
E NUPT
TH=#1
COST=CD(S(theta)
DENOM=A*A+M*M-2.0*A*M*COST
C2=C*C
C2D=C2/DENOM
PMW=A*COST-M
DVAR TH DENOM PMW COST C2 C2D
E
E
SDIC 'PROFILE.DAT'
EXIT
[nascad] 243

Figure 18. Listing of macro PROFILE, which produces Joukowski profiles.
Figure 19. Typical Joukowksi profile, produced by PROFILE.
Figure 20. Sections of missile, gotten by using the CADCEPT command PIERCE. In general, we may produce input for SPIRITS this way.
Figure 21. Showing locations of cutting planes for sections in fig 20.
Figure 22. Listing of macro OUTPNTS showing one method of getting points out of CADCEPT and into SPIRITS.
Figure 23. Listing of the file of points gotten by running the macro `OUTPUTS` on the file produced by `PROFILE`.

```
cat PROFILE.PNTS
2.500  0.000  0.000
2.464  0.000  0.000
2.359  0.000  0.000
2.187  0.011  0.000
1.955  0.024  0.000
1.672  0.044  0.000
1.346  0.070  0.000
0.937  0.100  0.000
0.607  0.132  0.000
0.315  0.163  0.000
0.179  0.191  0.000
-0.564  0.214  0.000
-0.933  0.223  0.000
-1.277  0.233  0.000
-1.620  0.227  0.000
-1.965  0.210  0.000
-2.097  0.193  0.000
-2.282  0.146  0.000
-2.417  0.102  0.000
-2.498  0.052  0.000
-2.526  0.000  0.000
-2.438  0.052  0.000
-2.417  0.102  0.000
-2.282  0.146  0.000
-2.097  0.193  0.000
-1.965  0.210  0.000
-1.620  0.227  0.000
-1.277  0.233  0.000
-0.933  0.223  0.000
-0.564  0.214  0.000
-0.179  0.191  0.000
 0.215  -0.163  0.000
 0.807  -0.132  0.000
 0.987  -0.100  0.000
 1.346  -0.070  0.000
 1.672  -0.044  0.000
 1.955  -0.024  0.000
 2.137  -0.011  0.000
 2.359  -0.003  0.000
 2.464  0.000  0.000
 2.500  0.000  0.000
```

PROGRAM USETHIN

INTEGER FLUN, EFLUN
REAL*8 X(3), TOL, XMAX
REAL*8 POINTS(5,128)
CHARACTER*32 FNAME, EFNAME
DATA FNAME /*
DATA EFNAME /*
EFLUN=12
FLUN=11
WRITE(6,135)
135 FORMAT('ENTER INPUT FILENAME:')
READ(5,136) FNAME
WRITE(6,137)
137 FORMAT('ENTER OUTPUT FILENAME: ')
READ(5,136)EFNAME
136 FORMAT(A30)
OPEN(UNIT=FLUN,FILE=FNAME(1:30),FORM='FORMATTED',
    STATUS='OLD',ERR=30)
GO TO 32
30 WRITE(6,138)
138 FORMAT('OPEN INPUT FILE ERROR')
GO TO 999
32 MPTS=0
10 READ(FLUN,101,ERR=34,END=998) X
101 FORMAT(3F10.3)
MPTS=MPTS+1
DO 77 MW=1, 3
    POINTS(MW,MPTS)=X(MW)
77 CONTINUE
POINTS(4,MPTS)=0
POINTS(5,MPTS)=0
GO TO 10
998 CONTINUE
TOL= 0.015
XMAX= 100.0
CALL THIN(POINTS,MPTS,TOL, XMAX)
OPEN(UNIT=EFLUN,FILE=EFNAME(1:30),FORM='FORMATTED',
    STATUS='NEW',ERR=31)
GO TO 33
31 WRITE(6,139)
139 FORMAT('OPEN OUTPUT FILE ERROR')
GO TO 999
33 WRITE(EFLUN,115) MPTS
115 FORMAT(I4)
DO 20 I=1,MPTS
20 WRITE(EFLUN,101) (POINTS(MW,I), MW=1, 3)
999 STOP
34 WRITE(6, 667) MPTS
667 FORMAT('GOT READ ERROR. MPTS=', I4)
STOP
END

Figure 24a. Listing of thinner program. A standalone program.
SUBROUTINE THIN(X, MPTS, TOL, XMAX)
C THINS 5D POINT ARRAY X SO THAT CHORD HEIGHT IS GREATER THAN TOL.
C USES NO TRIGONOMETRIC FUNCTIONS OR SQUARE ROOT.
C NPTS IS NUMBER OF POINTS IN ARRAY X.
C THE EXTRA 2D IN X IS AUGMENTATION BY U AND V PARAMETERS.
C WILL NOT ALLOW POINTS TO BE DISCARDED WHICH ARE CHORDALLY
C FARTHER AWAY FROM LAST POINT THAN XMAX. THIS ALMOST MEANS
C THAT THE MAXIMUM CHORDAL POINT SPACING OF THE OUTPUT IS XMAX.
C X AND XMAX ARE REPLACED ON OUTPUT BY THE THINNED CURVE, I. E.:
C THINNING IS DONE "IN PLACE". THIS MAY CAUSE TROUBLE ON CONVERSION
C TO ANOTHER LANGUAGE SUCH AS C, WHERE ARGUMENTS ARE PASSED BY
C VALUE.
C
REAL*8 X(5,128), TOL, XMAX
INTEGER MPTS
REAL*8 D1(3), D2(3), CH2, CHORD, TOLSQ, Q1, Q2, Q3, DSQ
INTEGER NPTS, L, K, LAST, L1, I, M

CH2 IS CONSTANT LENGTH TOLERANCE, SQUARED. NOTE THAT WE DO ALL LENGTH
COMPARISONS ON Squared LENGTHS, SINCE THE SQUARE FUNCTION IS
MONOTONIC. WE THUS AVOID COMPUTATION OF SQUARE ROOTS.
CH2= XMAX * XMAX
NPTS= MPTS

DONT BOTHER TO THIN LESS THAN 11 POINTS.
IF (NPTS .LE. 10) GO TO 999
CHORD= -1.
TOLSQ= TOL**2
L=1
K=2
LAST=1

PROGRAM FROM HERE ON USES FOLLOWING STRATEGY:
INCREMENT K UNTIL MAX CHORD HEIGHT OF RESULTING POLYGON
COMPOSED OF POINTS L THRU K-1 EXCEEDS TOLERANCE.
THEN WE SAVE THE KTH POINT AND START NEW POLYGON
ALSO AT KTH POINT.

5  K= K + 1
   IF (K.GT.NPTS) GO TO 200
   DSQ= -1.
   Q1= 0.
   DO 6 M=1, 3
       D1(M)= X(M, K+1) - X(M, L)
   6  Q1= Q1 + D1(M) *D1(M)
C DONT SAVE POINT IF VERY CLOSE TO LAST ONE.
   IF (DABS(Q1) .LT. 1.E-8) GO TO 5
   L1= LAST +1
C CHORD IS SQUARE OF CHORD LENGTH FROM LAST TO K.
   CHORD= (X(1, K) - X(1, L))**2 + 
           (X(2, K) - X(2, L))**2 + (X(3, K) - X(3, L))**2
   DO 10 I= L1, K
       Q2=0.
       Q3= 0.
   10  DO 7 M=1, 3
       D2(M)= X(M, I) - X(M, L)

Figure 24b. Thinner program, cont.
Q2 = Q2 + D2(M) * D2(M)
Q3 = Q3 + D1(M) * D2(M)

C DSQ IS DISTANCE** OF X(*, K) FROM CHORD (X(*, L), X(*, K + 1)).

DSQ = DMAX1(DSQ, Q2 - Q3 * Q3 / Q1)

C THROW OUT (DONT SAVE) POINT K
IF ((DSQ .LT. TOLSQ) .AND. (CHORD .LT. CH2)) GO TO 5

C SAVE POINT K
L = L + 1
DO 50 N = 1, 5
50 X(N, L) = X(N, K - 1)
LAST = K - 1
GO TO 5

200 L = L + 1
DO 250 N = 1, 5
250 X(N, L) = X(N, NPTS)
MPTS = L

999 RETURN
END

Figure 24c. Thinner program, concl.
Figure 25. Listing of points produced by thinner program used on file of points shown in figure 22.
Figure 26. Listing of macro INPNTS, showing how point files can be read into CADCEPT.