APPENDIX H: Conversion Algorithm for Orbital Parameters

This appendix contains software written in power BASIC that converts IBM double precision, floating point numbers to their decimal equivalent. Between Sept. 8, 1992 - Sept. 21, 1992 and Oct. 21, 1992 - Nov. 15, 1994, the orbital parameters that were added to the Level 1b dataset header were IBM double precision floating point numbers. The conversion software is contained below.

```
'HEXTODEC.BAS converts values from Level 1b header in hex notation to decimal
'  (Print statements for illustration only)
'Notation:
'Default precision: 8 byte floating point
'$ indicates string variable
'% indicates a two byte integer
'# indicates 8 byte floating point
defdbl a-z        'Default precision for all variables 8 byte floating point
'Example values:
'True values
x0=2707.578247 : y0=-1855.599762 : z0=-6455.342772
'Values in hex from Level 1b header
x$="43A939407FED2027" : y$="C373F998A009F622" : z$="C4193757BFE7E1FB"
cls : print "     X                Y                Z"
for n%=1 to 3
  select case n%
    'Select proper variable string
    case 1 : s$=x$ : case 2 : s$=y$ : case 3 : s$=z$
  end select
  hxdg$="&h"+mid$(s$,1,1)   'Take 1st hex digit (4 bits).  "&h" indicates hex
  v=val(hxdg$)           'Find decimal value of hex digit using VAL function
  'If the value is ∃ 8, then 1st bit (sign bit)=1, and value is negative.
  '  If so, subtract 8 (sign bit) from value.
  if v>=8 then sign$="-" : v=v-8 else sign$="+"
  expnt=v*16     'Multiply by 16, since these are high order bits of exponent
  'Evaluate 2nd hex digit, but all bits are significant for value
  hxdg$="&h"+mid$(s$,2,1) : v=val(hxdg$)
  expnt=expnt+v-64     'Add the two values and subtract 64 for exponent value
  'Remaining 14 hex digits are fractional part of number
  '  Divide 1st by 16^1, 2nd by 16^2, etc., and sum
  frct=0#
  for i%=3 to 16
    hxdg$="&h"+mid$(s$,i%,1) : v=val(hxdg$)
    frct=frct+v/16^(i%-2)
  next i%
  'Multiply fractional part by 16^exponent, and make negative if sign bit set
  nmb=16^expnt*frct : if sign$="-" then nmb=-nmb
  select case n%
```

```
    case 1 : x=nmb : locate 2,1 : print x0 : locate 3,1 : print x
    case 2 : y=nmb : locate 2,20 : print y0 : locate 3,20 : print y
    case 3 : z=nmb : locate 2,40 : print z0 : locate 3,40 : print z
  end select
next n%
stop
```

Note: For more information on how values are stored in IBM, refer to an IBM Assembler Language reference manual.