```fortran
!MS2-RC-NamesEtc.f95
!2025.01.17.1700cst JMS- The variables of Min-Steps Rubik's Cube solving.
!              Computer- "T4"/HP-800-G4-Mini/i7-8700T/IntelUHD630/Win10Pro-22H2
!                         ^name  ^Mfgr.Id    ^chipset ^graphics    ^OS
!                            /AbsoftProFortran 21.0.2/OpenGL+Glut3.6
!                             ^compiler                ^Fortran graphics

!"Min-Steps.pdf" introduces this software.

!Table of Contents - ToC:
!Module MS2RCDef

!----------------------------------------------------------------------------7 9
Module MS2RCDef                                                  !2024.07.10
!//////////////////////////////////////////////////////////
!use MS3RCDef, only:
Implicit none
!--- SER Analysis Scope: ------------------------
  character(64)::cTitle  !Title of the Analysis
  integer(4)   ::RubSize !=2: 2x2x2 Cube
                         !=3 !=3: 3x3x3 cube
  integer(4)   ::iType   !=2:Existing- 2x2x2- for visualization
                         !=3:        - 3x3x3- for visualization
                         !=4:LuTGen- based on (nLoMin,nLoMax)
                         !=5:      - based on nTracksTot & Track

  character(64)::EnvNml  !Namelist input file
  character(64)::cFileOut!Name - of the Output file
  character(64)::cLuTOut !     - of the Lookup Table

!--- Emulator (model): -----------------------------
!--- Rubik`s cube Emulator array:
  integer(4)  :: AECAV(0:19,24,27) !nAout<-(nCin,nAin,nVin)

!--- Moves & Choices => Sequences: --------------
!--- Moves:
  integer(4)::Mmax=30 !Moves    - maximum       ...limits an unknown array size
                      !It turns out that Mmax =15 suffices to solve
                      !2x2x2`s, but I wasn't able to predict that.
                      !

!!integer(4)::nM      !Moves  - index - defined locally in subroutines
  integer(4)::Mtot    !       - total
  integer(4)::Mused   !       - total used thusfar
  integer(4)::D       !Level   - the minimum total number of moves required
                      !            to solve a given Result.
                      !            ~ "the height above solved".
  integer(4)::Dmax    !Level   ~ worst case Result

!--- Choices: -----------------------------------
!!integer(4)::nC      !Choices- index - defined locally in subroutines
  integer(4)::Cmax   &!       - max    Rubsize=3:
               = 18   !                 6 faces with -90,+90, 180deg. rotations
  integer(4)::Ctot   &!       - total  Rubsize=2:
               = 9    !                 3 faces with -90,+90, 180deg. rotations
  integer(4)::C(18) &!         - maps nC values to Emulator (model) variables
               = (/ 1,2,5,6,9,10,13,15,17, 0, 0, 0, 0, 0, 0, 0, 0, 0/) !2x2x2
              != (/ 1,2,3,4,5, 6, 7, 8, 9,10,11,12,13,14,15,16,17,18/) !3x3x3
                     !                  Ctot non-zero values are left justified
 integer(4)::nCinv4(0:19)=(/0,2,1,4,3,6,5,8,7,10,9,12,11,13,14,15,16,17,18,19/)
 !--- Sequences:               a Sequence => the Moves scramblins a solved cube
  integer(8)::nS8     !Sequence- #- decimal              note: i*8, can be huge
  integer(4)::MC(30)  !one Sequence- MC(1:Mused) Moves,
                      !                                each with a Choice value
                      !                                in C(nC),nC in [1,nCTot]
  character(30)::cMCsymb!         - ... a character string of MC- for printout
                        !            report in Csymb format (1x,a1)  (1:Mused)
  integer(8)::Stried8 !Sequence- decimal- tried, some with duplicate Results
  integer(8)::Sfound8 !        -        - total number- found thusfar
  integer(8)::Stot8   !        -        -              - of/for all Results
  integer(8)::Smax8   !        -        - 24^Mtot

!--- Voter(s) & Attitudes => Results: ----------    Results <=> cube scrambles
!--- Voters:                                              Voter(s) <=> cell(s)
!!integer(4)::nV      !Voter     - index - defined locally in subroutines
  integer(4)::Vmax  &!          - max    - supported by array sizing
               = 27                                        !20 cells max
  integer(4)::Vtot  &!          - total - to solve for
               = 8                                         !8 cells
  integer(4)::V(27) &!          - maps nV values to Emulator (model) variables
               = (/  1,2,3,4,5,6,7,8,0,0 &
                    ,0,0,0,0,0,0,0,0,0,0 &
                    ,0,0,0,0,0,0,0        /) !2x2x2 cell#`s
                     !                  Vtot non-zero values are left justified
                     !                   number & report in decimal format (i3)
  integer(1)::Vtotshow !VA(1:Vtotshow) are displayed
!              = 8
!--- Attitudes:                                  Attitude(s) => 3Dof rotation(s)
!                                                cell translations are implicit
!!integer(4)::nA      !Attitude - index - defined locally in subroutines
  integer(4)::Atot =24!         - total
!--- Results:                                         Results <=> cube scrambles
  integer(4)::VA(27) !one Result- VA(1:Vtot) Attitudes, each with a numeric
                     !                            Attitude value in [1:Atot]
  character(27)::cVAsymb!       - VA(1:Vtot) as a character string- for sorting
                        !              report in  Asymbol format (a8) (1:Vtot)
                        !Tracking 3x3x3 rotation of the on-axis cells (#21-#26)
                        !  & the center virtual cell (#27) is supported
                        !  & simplifies animation.
  character( 5)::cSpare1!...round to 32 bytes
  character(27)::cVAtest!VA symbolic input test
  character( 5)::cSpare2!...round to 32 bytes
!--- recursive Indexer:
  integer(8)::      H8 !Handle- decimal- # (index)
  real(16)  ::   r16H8 !
  integer(8):: nRaccum8!      -        - recursive accumulator
                       !                   = predicted Rtot8 value
                       !          integer(8) maxes out     at ~<9.2e18
  real(16)  ::r16accum8!    ...used to warn of ^... counts to ~ 5.0e30

  integer(8)::Rfound8!      -        - total number- found-
  integer(8)::Rtot8  !      -        -             - of/for all Results
  integer(4)::iRvalid!      - is new (not found before)

!---- Sequences & Results record(s): -------------
!Sequences- of 30 Moves           - maximum
!Results  - for 20 Voters(s)/cells-
 type ::  SRrec;sequence   !                                           Bytes
  integer(8)  ::nS8        ! index  - of Sequence Sr(nS8)- decimal   ~<9.2e18  8
  integer(8)  ::H8         !"Handle"- of Lookup   Lr(H8 )-                      8
  !--Moves & Choices:
  integer(1)  ::Mused      !Moves  - in use, = %MC(0)                          1
  integer(1)  ::MC(1:30)   !MC(1:Mused)  the values are model "Choices"        30
  integer(1)   ::MC(1:30)  !MC(1:Mused), the values are model  choices         30
                          !values as decimals, report in CEsymb format (x1,a1)
  !--Voters & Attitudes:
  integer(1)   :: Vtot    !Voter(s)- total being analyzed                    1
  character(20)::cVAsymb  !Sortable character version of VEA(1:Vtot)          8
                          !      report in Asymbol format (a20)(1:Vtot)
  integer(1)   :: VA(20)  !VEA(1:Vtot) , the values are "Attitudes"           8
 end type SRrec   !                                                   Total: 64
 type(SRrec)            ::resetSR                                    !Total: 88
 type(SRrec)            ::Srprev
 type(SRrec)            ::Srseed
 type(SRrec)            ::Sr1
 type(SRrec)            ::Srtry
  type(SRrec)           ::Srbase
 type(SRrec)            ::Srn
 type(SRrec)            ::SrTestIn
!type(SRrec),allocatable::Sr(:) !Sequence records
!type(SRrec),allocatable::Rr(:) !Result records

  integer(8)::nSeqs(4,0:40)  !(nSeq|n1st|nLast|count,Sequence#)
  integer(4)::nSeqPrev       !Sequence#- previous
  integer(4)::nSeqPres       !         - present
!------------------------------------------------

!-- "Emulator" initialization Requests record: -----
!           (used by the .nml files):
 type ::  Erec ;Sequence                            !Ein: 2696 bytes
  integer(4)   ::Einsize
  integer(8)   ::Zonebimsize
  integer(8)   ::ZoneFound
  character(60)::cTitle  !Title- of the Analysis
 !Export filenames:
  character(60)::NameRoot!      - Root of run export filenames
 !...these FileNames are expansions of NameRoot:
  character(60)::cSummaryOut   !Name = %Nameroot +"-Summary.txt"- an ASCII file
  character(60)::cRrAsciiOut   !     =           +"-RrASCII.txt"-
  character(60)::cSrBinary     !     =           +"-Sr.bim"     - a binary file
  character(60)::cRrtoSrBinary !     =           +"-RrtoSr.bim" -
  character(60)::cEinBinary    !     =           +"-Ein.bim" -
  character(60)::cRrtoDisBinary!     =           +"-RrToDis.bim" -

  integer(4)   ::iType   !=2:Existing- 2x2x2- for visualization
                         !=3:        - 3x3x3- for visualization
                         !=4:LuTGen- based on (nLoMin,nLoMax)
                         !=5:      - based on nTracksTot & Track

  integer(4)   ::RubSize !=2:2x2x2;=3:3x3x3
 !-- define the Voter(s) = cell numbers:
  integer(4)::Vtot       !Voters- total- analyzed
  integer(4)::V(27)      !V(1:Vtot)- cell#'s being analyzed
                         !            - [1:27]:on [0]:off
                         !  number & report as decimals format (i3)
 !-- define the Attitudes = cell rotations:
  integer(4)   ::VA(27)   !Raid input- numeric  VA(1:Vtot)
  character(27)::cVAtest  !          - symbolic version of VA(1:Vtot)
 !-- define the Choices = face rotations allowed:
  integer(4)::Ctot       !Choices- total-
  integer(4)::C(18)      !C(1:Ctot) face rotations in use
                         !            - [1:18]:on [0]:off
                         !  number as decimals, report in Csymbol format (a1)

 !-- Define the Moves: a Move is any one face rotation:
  integer(4)::Mmax       !Moves  - maximum      ...limits an unknown array size
 !--- Visualization variables (if applicable):
  integer(2)::Mtotvis    !Moves  - total- for visualization
  integer(1)::MCvis(1:1022) ![1:18]  ->print as"iAtt"([1:I])~hex
  integer(1)::Vtotshow   !VA(1:Vtotshow) are displayed
 !integer(1)::Rfirst( 27)!Result- first- values[1:24] ->print as iAtt[a:x]
 !integer(1)::Rlast(  27)!      - last -
 !integer(1)::i1Spare2(7)!
 !integer(8)::RfirstnR   !      - #- first-  applies to 2x2x2's
 !integer(8)::RlastnR    !      -  - last -            (3x3x3: no sort#'s yet)

 !Wisdom Zone variables:
  integer(4)::Ztot
  integer(4)::Z(9)
  integer(4)::nZ
  character(60)::cZrFilename(9)
  integer(4)::ZMaskNeed(20)
  integer(4)::ZMaskfora(20)
  integer(4)::Vsizez(20)
  integer(4)::Mtot
 end type Erec
 type(Erec)::Ein ,Einreset! ----------------------

 !--- Wisdom Zone variables: Each zone is a Wisdom subset:
  integer(4)::Ztot         !Zones- in use-total
  integer(4)::Z(9)         !     - in use [1:nZtot], left justified
 type(Erec) ::EinW(9)      !     - parameters of the each Wisdom rec
  character(60)::cZrFilename(9)
  integer(4)::nZ           !     - currently in use
  integer(4)::iDiscord(    0:9)
  integer(4)::nOrder(      0:9)

  integer(1),allocatable,target::Zone1(:) !   253440 = (24,24,22,20)
  integer(1),allocatable,target::Zone2(:) ! 11022480 = (21,18,15,12, 9,6,3)
  integer(1),allocatable,target::Zone3(:) !185794560 = (18,16,14,12,10,8,6,4,2)
  integer(1),allocatable,target::Zone4(:) !
  integer(1),allocatable,target::Zone5(:) !
  integer(1),allocatable,target::Zone6(:) !
  integer(1),allocatable,target::Zone7(:) !
  integer(1),allocatable,target::Zone8(:) !
  integer(1),allocatable,target::Zone9(:) !
  integer(1)           ,pointer::ZonePnt(:)!

!--- Symbology for counting in base 24. Choices don't use the zero.
!      This adds sequential upper case G through N to the hexadecimal character set
!      while avoiding confusion between the letter "O" and zero:"0".
!      Rubik's Cubes have up to 18 move choices.
  integer(4)::Csymbol(0:23) = (/                            &
 !   0   1   2   3   4   5   6   7   8   9
 !  "0" "1" "2" "3" "4" "5" "6" "7" "8" "9"
    48, 49, 50, 51, 52, 53, 54, 55, 56, 57                  &
 !
 !  10  11  12  13  14  15  16  17  18  19  20  21  22  23
 !  "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N"
  , 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78   /)

!--- Symbology for up to 26 Attitudes: (Rubik's Cubes cells have 24.)
!      Zero indicates that no attitude is defined yet.
  integer(4)::Asymbol(-1:26) =  (/                            &
  !   " "     :=-1 "unused"
  !       "." := 0 "no attitude defined yet"
  !   -1   0   1   2   3   4   5   6   7   8   9  10  11  12    :offset 96
  !   " "  " "  "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l"
  !       .   a   b   c   d   e   f   g   h   i   j   k   l
      32, 46, 97, 98, 99,100,101,102,103,104,105,106,107,108 &
  !
  !   13  14  15  16  17  18  19  20  21  22  23  24  25  26    :offset 96
  !   "m" "n" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"
    ,109,110,111,112,113,114,115,116,117,118,119,120,121,122  /)

!--- For sortable symbology supporting up-to base 53 counting
!      and 52 Choice sequencing: see Utilities.f95

  integer(4)::RaidInit = 0
!---"Result address id" = "Raid" for a group of Voters&Attudes
!   "Attitude locations of Voters" = "AlV", in solved order.
!      the present Attitudes define where each Voter/cell has morphed to.
!      Recursive Function Indexer() realizes the complicated addressing.
 type ::  AlVrec;sequence !
  integer(4)  ::n         !Voter  - #- Ein%V(n) order                       cell#
  integer(4)  ::nV        !Voter  - #                                       cell#
  integer(4)  ::nA        !Attitude-                                    rotation#
  character(1)::Al(24)    !Attitude(s)- in location group order
  integer(4)  ::mdis(24)  !           - moves to solved

  integer(4)  ::nAl(24)   !           - Al order- all - decimal
  integer(4)  ::nAloc(24) !           - A  order- all - locations ***new***
  integer(4)  ::nlused    !location   - used
  integer(4)  ::nAavail   !Attitude(s)- available at each recursion level
  integer(4)  ::nAu(24)   !           - (1:nAavail)
  integer(4)  ::nAused    !           - =
  integer(4)  ::nAcount  !           - = ?????????????????????
  integer(4)  ::lavail(20)!locations  - available
 end type AlVrec          !
 type(AlVrec)::resetAlV
 type(AlVrec)::AlV(20)       !Raid record for each cell
 type(AlVrec)::RVbase(0:20)  !Result Voter(s) sequence- Baseline i.e.: all
 type(AlVrec)::RV    (0:20)  !                          - RV(Ein%V(1:Vtot))
!type(AlVrec)::RVSto (0:20,3)!                          - For voting      !*****

!--- vote groups:                                                    2024.08.27
 type ::  Vgrec;sequence
  integer(4)   ::V(20)
  integer(4)   ::VA(20)
  integer(4)   ::Maskneed(20,3) !must be >0                              !*****
  integer(4)   ::Maskfor1(20,3)    !must be ones                         !*****
  integer(4)   ::iFlag
  integer(4)   ::iDiscord(0:9)  !Discord (0) = total,       =-1: all invalid,
                                !                           = 0: solved
                                !            = (1)*100 + (2) + (3) when they are >=0
                                !        (1) = Group1 *100: [-1,0,..,800]
                                !        (2) = Group2       [-1,0,.., 11]
                                !        (3) = Group3       [-1,0,.., 11]
  character(40)::cVgFileName(3) !data import filenames                   !*****
 end type Vgrec
 type(Vgrec)            ::Vgzero,resetVg
 type(Vgrec)            ::Vg


!------------------------------------------------
!Contains
!---------------------------------------------------------------------------7 9
!//////////////////////////////////////////////////////////
End Module MS2RCDef
!---------------------------------------------------------------------------7 9
```

```

!ms-rc analysis <=> Rubik's Cube Emulator model                      2024.11.13
! Voter(s)   <=> Cell(s): 8 corner cells are solved in a 2x2x2 cube.
!                           #: 1, 2, 3, 4, 5, 6, 7, 8
!                        :12 more edge cells are solved in a 3x3x3 cube.
!                           #: 9,10,11,12,13,14,15,16,17,18,19,20
!                      In addition, face rotations can be accumulated by:
!                        : 6 on-axis cells:
!                           #:21,22,23,24,25,26     &
!                        : 1 imaginary center cell
!                           #:27

! Cell diagram:                +X            Axis Face# Choice()     |->hex:
!                              /              -X:    1   # 1, # 2  #13| 1, 2,  D
!                             /               +X:    2   # 3, # 4  #14| 3, 4,  E
!    (+1,-1,-1)= 2-----11------4 =(+1,+1,-1)  -Y:    3   # 5, # 6  #15| 5, 6,  F
!               /|     /|   . /|              +Y:    4   # 7, # 8  #16| 7. 8,  G
!              9-----*21----12 |              -Z:    5   # 9, #10  #17| 9, A,  H
!             /| |   /| | . /| |              +Z:    6   #11, #12  #18| B, C,  I
!(-1,-1,-1)= 1-----10------3 | |             Rotation:[-90^ +90^ 180^  -  + 180]
!  X  Y  Z   | | |  | | |. | | |
!            | |14--|-|*24-|-|-16
!            | |/|  | |/|  | |/|
!            |*22---|*27---|*25 ....+Y->  Choice(7,8,& G)
!            |/| |  |/| |  |/| |    Indices:  1-thru- 8 are            corners
!           13-----*23-----15| |           :  9-thru-20 are 3x3x3      edges
!            | | |  | | |  | | |           : 21-thru-26 are 3x3x3 face centers
!            | | 6--|-|19--|-|-8 =(+1,+1,+1)          --------------------
!            | |/   | |/   | |/            :     27  accumulates rotations
!            |17----|*26---|-20            :      0   pure Euler rotation
!            |/     |/.    |/                          (face independent)
!(-1,-1,+1)= 5-----18------7 =(-1,+1,+1)
!                     |         Solving a 2x2x2 is nearly as difficult as
!                     |         solving just the corner cells of a 3x3x3.
!                    +Z  Choice(11,12,& I)

! Attitudes <=> 3DoF Rotations: Each cell has 24 possible (roll,pitch,yaw)'s.
!                               DoF: "Degree of Freedom"

!"Data-Attitudes.txt"  JMS  2024.03.20
!A(1:24)  "Cell Attitudes"
!!%nA   %iRPY             %ijk       %PathLen
!symb #  Roll Pitch   Yaw   i   j   k,     +1
!a    1,    0,    0,    0,  0,  0,  0,     +1  = solved
!b    2,  -90,    0,    0, -1,  0,  0,     +1
!c    3,  +90,    0,    0, +1,  0,  0,     +1
!d    4,    0,  -90,    0,  0, -1,  0,     +1
!e    5,    0,  +90,    0,  0, +1,  0,     +1
!f    6,    0,    0,  -90,  0,  0, -1,     +1
!g    7,    0,    0,  +90,  0,  0, +1,     +1
!h    8, +180,    0,    0, +2,  0,  0,     +2
!i    9,    0, +180,    0,  0, +2,  0,     +2
!j   10,    0,    0, +180,  0,  0, +2,     +2
!k   11,  -90,    0,  -90, -1,  0, -1,     +2
!l   12,  -90,    0,  +90, -1,  0, +1,     +2
!m   13,    0,  -90,  -90,  0, -1, -1,     +2
!n   14,    0,  -90,  +90,  0, -1, +1,     +2
!o   15,    0,  +90,  -90,  0, +1, -1,     +2
!p   16,    0,  +90,  +90,  0, +1, +1,     +2
!q   17,  +90,    0,  -90, +1,  0, -1,     +2
!r   18,  +90,    0,  +90, +1,  0, +1,     +2
!s   19,  -90,    0, +180, -1,  0, +2,     +3
```

```
!t   20,    0,  -90, +180,  0, -1, +2,     +3
!u   21,    0,  +90, +180,  0, +1, +2,     +3
!v   22,  +90,    0, +180, +1,  0, +2,     +3
!w   23, +180,    0,  -90, +2,  0, -1,     +3
!x   24, +180,    0,  +90, +2,  0, +1,     +3

! Choices   <-> Face Rotations:
!                 There are 6 faces, each with -90, +90, & 180 deg rotation;
!                               hence there are 18 choices total.

!The heart of the Emulator model is 648 lines of data in "Data-AECAV.txt":
!---------------------------------------------------------------------------7 9

```